actuators

Special Issue Reprint

Nonlinear Control of Mechanical and Robotic Systems

Edited by
Arturo Zavala-Río, Victor Santibañez and Fernando Reyes-Cortés

mdpi.com/journal/actuators

MDPI

Nonlinear Control of Mechanical and Robotic Systems

Nonlinear Control of Mechanical and Robotic Systems

Guest Editors

Arturo Zavala-Río
Victor Santibañez
Fernando Reyes-Cortés

MDPI

Basel • Beijing • Wuhan • Barcelona • Belgrade • Novi Sad • Cluj • Manchester

Guest Editors

Arturo Zavala-Río
Control and Dynamical Systems Division
San Luis Potosi Institute of Scientific and Technological Research
San Luis Potosí
Mexico

Victor Santibañez
La Laguna Institute of Technology
National Technological Institute of Mexico
Torreón
Mexico

Fernando Reyes-Cortés
Faculty of Electronic Sciences
Autonomous University of Puebla
Puebla
Mexico

Editorial Office

MDPI AG
Grosspeteranlage 5
4052 Basel, Switzerland

This is a reprint of the Special Issue, published open access by the journal *Actuators* (ISSN 2076-0825), freely accessible at: https://www.mdpi.com/journal/actuators/special_issues/439D2WGO21.

For citation purposes, cite each article independently as indicated on the article page online and as indicated below:

Lastname, A.A.; Lastname, B.B. Article Title. *Journal Name* **Year**, *Volume Number*, Page Range.

ISBN 978-3-7258-8857-3 (Hbk)
ISBN 978-3-7258-8858-0 (PDF)
https://doi.org/10.3390/books978-3-7258-8858-0

© 2026 by the authors. Articles in this reprint are Open Access and distributed under the Creative Commons Attribution (CC BY) license. The reprint as a whole is distributed by MDPI under the terms and conditions of the Creative Commons Attribution-NonCommercial-NoDerivs (CC BY-NC-ND) license (https://creativecommons.org/licenses/by-nc-nd/4.0/).

Contents

About the Editors . . . . . **vii**

Arturo Zavala-Río, Víctor Santibáñez and Fernando Reyes-Cortes
Nonlinear Control of Mechanical and Robotic Systems
Reprinted from: *Actuators* **2026**, *15*, 415, https://doi.org/10.3390/act15080415 . . . . . **1**

Daniel Gamez-Herrera, Juan Sifuentes-Mijares, Victor Santibañez and Isaac Gandarilla
Composite Adaptive Control of Robot Manipulators with Friction as Additive Disturbance
Reprinted from: *Actuators* **2025**, *14*, 237, https://doi.org/10.3390/act14050237 . . . . . **5**

Cristina Losada-Gutiérrez, Felipe Espinosa, Carlos Cruz and Biel P. Alvarado
Electronic Control Unit and Digital Twin Based on Raspberry Pi 4 for Testing the Remote Nonlinear Trajectory Tracking of a P3-DX Robot
Reprinted from: *Actuators* **2025**, *14*, 376, https://doi.org/10.3390/act14080376 . . . . . **27**

Esmeralda Enriqueta Mascota Muñoz, Oscar González Miranda, Xchel Ramos Soto, Juan Manuel Ibarra Zannatha and Santos Miguel Orozco Soto
Quadratic Programming Vision-Based Control of a Scale-Model Autonomous Vehicle Navigating in Intersections
Reprinted from: *Actuators* **2025**, *14*, 494, https://doi.org/10.3390/act14100494 . . . . . **43**

Luis Angel Martínez-Ramírez, Rafael Isaac Vásquez-Cruz, German Ardul Munoz-Hernandez, Gerardo Mino-Aguilar, Wuiyevaldo Fermín Guerrero-Sánchez, Roberto Carlos Ambrosio-Lázaro and José Fermi Guerrero-Castellanos
Trajectory Control for Car-like Mobile Robots via Frugal Predictive Control with Integrated Disturbance Rejection
Reprinted from: *Actuators* **2026**, *15*, 260, https://doi.org/10.3390/act15050260 . . . . . **64**

Fei Yan, Jianhua Li, Huawei Han, Qiwang Xu and Linfeng Hu
Kinematic Decoupling and α-TDE-NTSM Control for Single-Tendon-Driven Manipulators
Reprinted from: *Actuators* **2026**, *15*, 271, https://doi.org/10.3390/act15050271 . . . . . **101**

Filiberto Muñoz Palacios, Eduardo S. Espinoza, Jorge Said Cervantes-Rojas, Jesus Patricio Ordaz Oliver, Octavio Garcia-Salazar and Luis Rodolfo Garcia Carrillo
Robust Trajectory Tracking Control of an Unmanned Surface Vehicle via a Sliding-Mode Dynamic Neural Network Identifier
Reprinted from: *Actuators* **2026**, *15*, 273, https://doi.org/10.3390/act15050273 . . . . . **123**

Isela G. Carrera, Hector A. Moreno and Jose Luis Ordoñez-Avila
Design and Modeling of a Robot for Rehabilitation of the Sit-to-Stand Movement and Walking
Reprinted from: *Actuators* **2026**, *15*, 323, https://doi.org/10.3390/act15060323 . . . . . **151**

Davide D'Andrea, Giacomo Risitano and Dario Santonocito
Validation of Pneumatic Actuation for Fast Fatigue Testing of Additive-Manufactured Polymers
Reprinted from: *Actuators* **2025**, *14*, 598, https://doi.org/10.3390/act14120598 . . . . . **181**

About the Editors

Arturo Zavala-Río

Arturo Zavala-Río is a Senior Researcher at the Control and Dynamical Systems Division of the San Luis Institute of Scientific and Technological Research, Mexico. His main research interests include the modeling, analysis and control of nonlinear systems with special focus on the control of mechanical and robotic systems. Since 1999, he has been an active member of the National Researcher System from the National Council of Science and Technology, currently the Ministry of Science, Humanities, Technology, and Innovation, Mexico.

Victor Santibañez

Victor Santibañez is from La Laguna Institute of Technology / National Technological Institute of México. His areas of interest include robot, nonlinear, adaptive, fuzzy, and mechatronic system control. He has authored or co-authored over 300 publications, comprising two books published internationally by Springer and Pearson Prentice Hall; 122 papers in international journals indexed by ISI-JCR; seven articles in peer-reviewed journals; six chapters in internationally published books; and 200 conference papers (75 international and 125 national). He has supervised 49 master's and 19 doctoral graduates. As of July 2025, his total SCOPUS citation count stands at 3,213 (3,369 according to research.com), and his h-index is 33 (covering the period from 1996 to July 2026). He holds Level III status in the SECIHTI National System of Researchers, Mexico.

Fernando Reyes-Cortés

Fernando Reyes-Cortés is a professor and researcher in robotics and control at Benemerita Autonomous University of Puebla, Mexico. His main research interests include position and tracking control for robot manipulators and nonlinear dynamic systems. Since 1993, he has been an active member of the National Researcher System from the National Council of Science and Technology, currently the Ministry of Science, Humanities, Technology, and Innovation, Mexico.

actuators

MDPI

Editorial

Nonlinear Control of Mechanical and Robotic Systems

Arturo Zavala-Río [1,*], Víctor Santibáñez [2] and Fernando Reyes-Cortes [3]

[1] Control and Dynamical Systems Division, San Luis Potosi Institute of Scientific and Technological Research, San Luis Potosí 78216, Mexico
[2] La Laguna Institute of Technology, National Technological Institute of Mexico, Torreón 27000, Mexico; vasantibanezd@lalaguna.tecnm.mx
[3] Faculty of Electronic Sciences, Autonomous University of Puebla, Puebla 72570, Mexico; fernando.reyes@correo.buap.mx
* Correspondence: azavala@ipicyt.edu.mx

Position and motion control of robotic and mechanical systems have proven to be challenging tasks, primarily because of their multidimensional, coupled and highly nonlinear dynamics. This, as well as the need for the system to move within a wide range of its motion domain, makes nonlinear control design a necessity. Probably the most *popular* and *ad hoc* design method for robotic and mechanical systems is *Passivity-Based Control* [1], which uses control laws that involve position- and velocity-error correction terms, as well as model-dependent terms, that render the desired trajectory/position as a closed-loop equilibrium; this is given adequate stability properties through the aforementioned correction terms shaping the potential energy function and injecting damping (*via* the position- and velocity-error related terms, respectively). In such a model, it is hard to imagine a robotic or mechanical system being controlled across a wide range of its motion domain without the system model being included in the control algorithm. But this adds further complications. The system's physical parameters are not always exactly known or could change over time. Moreover, there could be unmodeled phenomena; friction components and input saturation are common examples of this. The latter is a consequence of natural restrictions on actuator capabilities, but further constraints on the system's physical configuration or velocities could also need to be considered (velocities being relevant, for instance, in the case of mobile robots). Further complications include the lack of system variable (such as velocity) measurements or underactuation. On the other hand, high-accuracy closed-loop performance is now expected or even demanded. All these aspects have produced the need for refined control design, giving rise to nonlinear control schemes with sophisticated design techniques, such as adaptive [2], robust [3], output-feedback [4], bounded [5], or finite-time control [6], as well as combinations of these, such as robust adaptive [7], bounded output-feedback [8], bounded finite-time [9] or finite-time adaptive controllers [10]. Improvements on control design that take these complications into account are continual active topics in the research community.

This Special Issue collects works that present recent advances in nonlinear control of robotic and mechanical systems. It contains eight articles that deal with the above-mentioned active topics or with aspects that are intimately related to the control design, testing or implementation of such systems, such as modeling or design, construction, and actuation of robots with innovative mechanical structures. The included papers are briefly described below.

Contribution 1 develops an innovative view on handling (often neglected) Coulomb and viscous friction components in the control of robot manipulators. Considering them as additive dissipative disturbance terms with a known linearly parameterizable structure

that proves to be *ad hoc* for control design purposes, the authors successfully design a composite adaptive control scheme that achieves trajectory tracking and, under persistently exciting desired trajectories, estimation of system and friction component parameters. Such an approach contributes a practical control strategy for refined trajectory tracking of robot manipulators. A thorough Lyapunov-function-based closed-loop analysis supports the contributed design, and experimental results on a two-degree-of-freedom robot arm further corroborate the analytical results and the efficiency of the proposed control scheme.

Contribution 2 presents a novel strategy for the validation of nonlinear robotic control systems prior to implementation. The proposed strategy combines Hardware-in-the-Loop (HIL) and Digital Twin (DT) technologies. HIL is described by the authors as a novel testing methodology used within the field of control system development, enhancing interaction between hardware components and simulated or emulated environments. DT is described by the authors as a virtual representation or model of a physical object or process obtained from real-world data. The authors show the efficiency of their proposed methodology through an experimental implementation of a development environment for the nonlinear trajectory tracking of a wheeled differential drive device: the P3-DX robot. The implementation involves a Lyapunov-based designed controller.

Contribution 3 details an optimal control scheme for autonomous vehicles navigating intersection scenarios. The control design is based on the solution of a Quadratic Programming optimization technique. Compared to previous approaches, this work presents a low-level control system that only involves on-board sensors and computers to accurately navigate intersections, avoiding communication between agents and external sensing. The control scheme is supported through a Lyapunov-based closed-loop analysis and experiments on an AutoMiny platform. The proposed control approach further deals with input constraints, solving the tracking problem and avoiding input saturation.

Contribution 4 gives an integral solution to the open challenge of achieving trajectory control for autonomous car-like mobile robots. It solves this complex problem through a modular system that incorporates a multi-level hierarchical control scheme. The modular scheme combines an extended Kalman filter for state estimation, frugal model predictive control for trajectory tracking, and an electronic differential through which each wheel is independently regulated by a nonlinear dynamic control law capable of addressing model uncertainties and external disturbances. The study includes all the analytical details, as well as experimental validation through a simple yet complete experimental platform using open and standardized communication protocols, an on-board embedded system to drive wheel angular velocities and direction, and a MATLAB-ROS (Robot Operating System) 2024B ground station. The experimental results demonstrate that the interval disturbance-compensating low-level control, predictive high-level planning, and sensor-fusion-based state estimation provide an effective solution for reliable autonomous navigation in challenging environments.

Contribution 5 presents the design and implementation of a novel tendon-driven manipulator motivated by weight reduction and increased flexibility for rigid-link manipulators. The study not only covers the technical details of the design but also produces a kinematic model. It further develops a non-singular terminal-sliding-mode control scheme with time-delay estimation by using the weighted average of estimated unknown dynamics at two different delay times. Such a control design is supported through Lyapunov analysis. Through experimental implementation, the constructed prototype is proven to offer reliable performance. The presented study on the implemented prototype thus shows its feasibility for applications, offering advantages in weight reduction, drag minimization and control implementations.

https://doi.org/10.3390/act15080415

Contribution 6 deals with the control of unmanned surface vehicles (USVs) under model uncertainties, unknown parameters, and external disturbances. The proposed approach engages with the underactuated nature of USV dynamics through a suitable change of coordinates. Since the transformed system dynamics involve hydrodynamic coefficients, which are very difficult to determine, the authors include a novel sliding-mode-based dynamic neural network identifier that enables the reconstruction of the transformed coordinates without the need for exact values of the system parameters. Moreover, the design leads to finite-time convergence of the defined sliding-surface variable and errors in the neural weights to zero in finite time. The convergence of the transformed coordinate reconstruction errors is proven to take place asymptotically. The estimated transformed coordinate variables are then transmitted to a sliding-mode-based controller that guides those variables to the desired trajectory. The analysis is supported through Lyapunov-based analysis and corroborated through simulation results.

Contribution 7 presents the design and modeling of a new robotic device constructed to assist therapists in the rehabilitation of sit-to-stand and walking movements. The paper focuses on the technical aspects of the system and provides a detailed mathematical analysis of its mechanics. The development of the designed prototype is described, covering the main aspects considered during the mechanical design, the definition of the control architecture, and the integration of the system components. The paper covers all the details of the inverse-kinematics and -dynamics models.It further highlights the usefulness of these models to address crucial aspects such as the determination of the required position and velocity profiles, providing necessary tools for the control of the robot and the design of the mechanical architecture of the prototype.

Contribution 8 focuses on the validation of pneumatic actuation for fast fatigue testing of additively manufactured polymers. Through mechanical testing machines equipped with control systems, the work validates pneumatic actuators as a viable alternative to servo-hydraulic systems, showing their effectiveness and reliability. This is significant for the robotics community, as the growing interest in compliant and soft robotics has prompted researchers to reconsider the potential of pneumatic actuation for advanced and complex applications.

We believe that the above-described works included in this Special Issue present relevant contributions to the research community on the control, modeling, design, implementation and actuation of robotic and mechanical systems. We are convinced that scholars, researchers and readers interested in the treated topics will find appealing original solutions to interesting open problems in this collection. Future research should focus on the integration of further constraints into control design. For instance, input constraints are not always considered, or their impact may be evaluated through implementation tests that are not necessarily exhaustive. While further integration of these and other types of constraints into control design has not been exhaustively studied for all types of mechanical systems—or is often disregarded in the formulation of considered problems—it has proven to be the only way to ensure avoidance of the potentially undesirable or unexpected effects that result from their omission. We thank all the reviewers involved in reviewing the accepted papers, as well as all the personnel from *Actuators* for their invaluable help in the production of this Special Issue.

Author Contributions: Conceptualization, A.Z.-R., V.S. and F.R.-C.; methodology, A.Z.-R., V.S. and F.R.-C.; investigation, A.Z.-R., V.S. and F.R.-C.; writing original draft preparation, A.Z.-R., V.S. and F.R.-C.; writing review and editing, A.Z.-R., V.S. and F.R.-C.; visualization, A.Z.-R., V.S. and F.R.-C.; supervision, A.Z.-R. All authors have read and agreed to the published version of the manuscript.

Data Availability Statement: Data sharing is not applicable.

https://doi.org/10.3390/act15080415

Conflicts of Interest: The authors declares no conflicts of interest.

List of Contributions

1. Gamez-Herrera, D.; Sifuentes-Mijares, J.; Santibañez, V.; Gandarilla, I. Composite adaptive control of robot manipulators with friction as additive disturbance. *Actuators* **2025**, *14*, 237. https://doi.org/10.3390/act14050237.
2. Losada-Gutiérrez, C.; Felipe E.; Cruz, C.; Alvarado, B.P. Electronic control unit and digital twin based on Raspberry Pi 4 for testing the remote nonlinear trajectory tracking of a P3-DX robot. *Actuators* **2025**, *14*, 376. https://doi.org/10.3390/act14080376.
3. Mascota Muñoz, E.E.; González Miranda, O.; Ramos Soto, X.; Ibarra Zannatha, J.M.; Orozco Soto, S.M. Quadratic programming vision-based control of a scale-model autonomous vehicle navigating in intersections. *Actuators* **2025**, *14*, 494. https://doi.org/10.3390/act14100494.
4. Martínez-Ramírez, L.A.; Vásquez-Cruz, R.I.; Munoz-Hernandez, G.A.; Mino-Aguilar, G.; Guerrero-Sánchez, W.F.; Ambrosio-Lázaro, R.C.; Guerrero-Castellanos, J.F. Trajectory control for car-like mobile robots via frugal predictive control with integrated disturbance rejection. *Actuators* **2026**, *15*, 260. https://doi.org/10.3390/act15050260.
5. Yan, F.; Li, J.; Han, H.; Xu, Q.; Hu, L. Kinematic decoupling and α-TDE-NTSM control for single-tendon-driven manipulators. *Actuators* **2026**, *15*, 271. https://doi.org/10.3390/act15050271.
6. Muñoz Palacios, F.; Espinoza, E.S.; Cervantes-Rojas, J.S.; Ordaz Oliver, J.P.; Garcia-Salazar, O.; Garcia Carrillo, L.R. Robust trajectory tracking control of an unmanned surface vehicle via a sliding-mode dynamic neural network identifier. *Actuators* **2026**, *15*, 273. https://doi.org/10.3390/act15050273.
7. Carrera, I.G.; Moreno, H.A.; Ordoñez-Avila, J.L. Design and modeling of a robot for rehabilitation of the sit-to-stand movement and walking. *Actuators* **2026**, *15*, 323. https://doi.org/10.3390/act15060323.
8. D'Andrea, D.; Risitano, G.; Santonocito, D. Validation of pneumatic actuation for fast fatigue testing of additive-manufactured polymers. *Actuators* **2025**, *14*, 598. https://doi.org/10.3390/act14120598.

References

1. Ortega, R.; Loría, A.; Niclasson, P.J.; Sira-Ramírez, H. *Passivity-Based Control of Euler-Lagrange Systems*; Springer: London, UK, 1998.
2. Slotine, J.J.E.; Li, W. On the adaptive control of robot manipulators. *Int. J. Robot. Res.* **1987**, *6*, 49–59. [CrossRef]
3. Slotine, J.J.E. The robust control of robot manipulators. *Int. J. Robot. Res.* **1985**, *4*, 49–64. [CrossRef]
4. Berghuis, H.; Nijmeijer, H. Global regulation of robots using only position measurements. *Syst. Control Lett.* **1993**, *21*, 289–293. [CrossRef]
5. Spong, M.W.; Thorp, J.S.; Kleinwaks, J.M. The control of robot manipulators with bounded input. *IEEE Trans. Autom. Control* **1986**, *31*, 483–490. [CrossRef]
6. Hong, Y.; Xu, Y.; Huang, J. Finite-time control for robot manipulators. *Syst. Control Lett.* **2002**, *46*, 243–253. [CrossRef]
7. Tao, G. On robust adaptive control of robot manipulators. *Automatica* **1992**, *28*, 803–807. [CrossRef]
8. Santibañez, V.; Kelly, R. On global regulation of robot manipulators: Saturated linear state feedback and saturated linear output feedback. *Eur. J. Control* **1997**, *3*, 104–113. [CrossRef]
9. Zavala-Río, A.; Zamora-Gómez, G.I.; Sanchez, T.; Reyes-Cortes, F.; López-Araujo, D.J. A more exhaustive study on the finite-time and exponential tracking continuous control for constrained-input mechanical systems: Improved design and experiments. *Int. J. Robust Nonlinear Control* **2022**, *32*, 8526–8557. [CrossRef]
10. Cruz-Zavala, E.; Moreno, J.A.; Loría, A.; Arteaga, M.A. On Adaptive Continuous Finite-Time Regulators for Euler-Lagrange Systems. *Int. J. Adapt. Control Signal Process.* **2026**, *40*, 733–751. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

actuators

MDPI

Article

Composite Adaptive Control of Robot Manipulators with Friction as Additive Disturbance

Daniel Gamez-Herrera , Juan Sifuentes-Mijares, Victor Santibañez * and Isaac Gandarilla

Instituto Tecnológico de La Laguna, Tecnológico Nacional de México, Blvd. Revolución y Av. Instituto Tecnológico de La Laguna S/N, Torreón C.P. 27000, Coahuila, Mexico; danielgamezherrera@hotmail.com (D.G.-H.); jsifuentesm@correo.itlalaguna.edu.mx (J.S.-M.); igandarillae@lalaguna.tecnm.mx (I.G.)

* Correspondence: vasantibanezd@lalaguna.tecnm.mx; Tel.: +52-871-705-1324

Abstract

In this paper, an adaptive control scheme composed of an estimated feed-forward compensation and a PD control law with three mutually independent estimators is proposed for the tracking of desired trajectories in joint space for a robotic arm. One of the estimators is used to identify inertial and geometrical parameters, while the others determine the two principal components of the friction phenomenon: the part whose magnitude is position-dependent but velocity-independent and the part whose magnitude is proportional to velocity. Next, the persistently exciting condition is satisfied for each regression matrix of the estimators in a way that is easier to prove than the classical structure. Then, uniform global asymptotic stability can be concluded for the tracking error, regardless of parametric convergence, by applying the direct Lyapunov theorem. This scheme has been applied experimentally for a robotic arm to verify the theoretical results. The experimental results yielded a better performance in both estimating the parameters and tracking, with a much simpler overall analysis than the alternatives consulted.

Keywords: adaptive control; robot control; non-linear control; Lyapunov stability; persistently exciting

1. Introduction

Most applications of robot manipulators require at least partial knowledge of the relation of input variables to output variables. When such a relation is determined analytically, it is established in a set of ordinary differential equations. Such a set is called the *dynamic model*, and its behavior is characterized by the parameters it includes. Non-linear model-based control techniques require full knowledge of these parameters to design a controller. Computed torque is one of the main techniques used to undertake the tracking of desired trajectories in joint space for robot manipulators as it allows for feedback linearization, which consists of a form of transformation that leads to some linear closed-loop dynamics, which are not necessarily stable but are controllable [1]. However, it presents a significant drawback, namely, that all the parameters in the dynamic model must be known, which is not the case; at the modeling stage, only the structure of the dynamic model can be assumed to be known exactly.

A complementary approach in these cases is the so-called adaptive control theory, whose methodologies involve the field of system identification; this theory considers a set of differential equations as a grey box model, assuming its structure a priori knowledge and then estimating the model parameters [2]. Subsequent to the modeling stage is the

parametric identification stage wherein the linear parametrization of robot dynamics is often used, and an identification algorithm is implemented to estimate the unknown parameters [3]. This approach is not only useful in a situation where the constant parameters are not known a priori, but they can vary as the robot dynamics evolves [4]. This implies that at least one parameter may be time-variant and that if the controller is not continuously redesigned, it may not properly control the changing plant.

The last case is the most realistic proposal as even gravitational interaction is not constant; it is a distributed parameter over spacetime. A nominal mid-range value on Earth is defined as it is greater at the poles than at the Equator [5]; however, variation in its magnitude can be neglected as it is small. On the other had, the phenomenon of friction cannot be neglected as its contribution can affect the dynamic behavior of a mechanical system such as a robot manipulator [6]; even in theory, it may change the stability conclusions because of its dissipative effect. Extensive research concerning the friction phenomenon has been carried out since the work of Coulomb and Amontons, who established some dependencies of the coefficient of friction on conditions such as the normal force, the size of the contact area, the sliding velocity, and atmospheric variables [7].

The need for a friction model that takes into account the fact that the friction force at zero velocity is higher than the friction in motion, and that the decrease from static to kinetic friction is a continuous process, led to some more complex models such as the Armstrong–Hélouvry model [8]. This model incorporates the Stribeck effect, which describes the decrease in friction force with an increase in velocity close to null velocity as a non-linear function of velocity, i.e., as a transition between the maximum static friction and the typical Coulomb friction, and as a viscous friction, i.e., a regime proportional to the velocity. A conceptual representation is depicted in Figure 1.

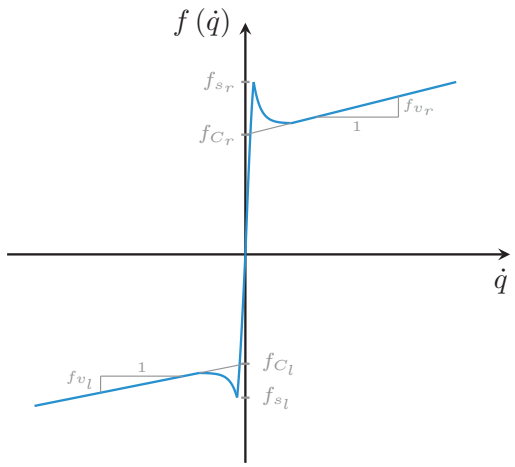


Figure 1. Conceptual representation of friction phenomenon.

Despite the successful application of this kind of static model, it is indisputable that friction as a phenomenon possesses its own dynamics. Some time later, dynamic friction models appeared, such as the Dahl model [9], and a generalization of it, the LuGre model [10], with the former being a one-state variable model whose behavior is similar to that of a non-linear spring in the elastoplastic domain. For small deflections, this is governed by Hooke's law and is approximately linear, but for large deflections, non-reversible changes appear, which reveals a frictional lag between velocity reversals. Its importance lies in the discovery of the existence of a hysteresis between the necessary force before the sliding occurs and the related displacement [11]. The latter is an extension of Dahl model. The LuGre model introduces a velocity-dependent exponential function instead of a constant inside the dynamics of the internal state; since the steady state of

the Dahl model is equivalent to the Coulomb friction, it does not take into account the Stribeck effect [12]. This strategy is not only useful in considering the Stribeck effect; it is fundamental when it comes to handling anisotropy, i.e., the fact that friction force is not uniform in all directions. The LuGre model has been extensively used for friction compensation in control applications because of two key factors: that the model state possesses a simple physical interpretation as the average deflection of a bristle, and its continuous first-order bounded dynamics [13]. Nonetheless, the experimental parameters introduced in this model, the fact that the model's internal states are not measurable, and its time-varying nature suggest that much more a priori knowledge is required [14].

Other main alternative approach based on control theory is to compensate any deviation from the nominal dynamics with some control strategy using the data acquired from sensors. Representative approaches, such as neuronal networks [15], fuzzy logic [16], fault tolerant schemes [17], and Bayesian optimization [18], have been researched throughout this past decade, but they have been found to be too complex in structure, to be computationally inefficient, and to involve more sensors, and their stability has also proven hard to prove. A control scheme that overcomes the drawbacks identified in recent research is proposed in the present work.

Contributions

In this work, a technical proposal for estimating friction coefficients online is presented, using a key modularity feature in the design of the controller. The regression matrix of the identifier achieves some useful boundedness properties independently of the controller. The proposed scheme possesses the following novelties and advantages:

- A decomposition of the robot manipulator model into an adequate linear parametrization for the conservative part and another for the non-conservative part is presented.
- The friction phenomenon is characterized as two independent additive disturbances that can be linearly parametrized in terms of the physical coefficients of friction with a quiet simple regression matrix.
- A separation into three simpler regression matrices is performed, which allows us to deal with them one by one and to find the upper and lower bounds that satisfy the persistent excitation condition for each of them.
- A relaxation in the *persistent excitation condition* is made for the overall regression matrix that arises when the well-known *swapping technique* is applied. Both concepts are presented in detail in Section 3.3.
- Lyapunov analysis is developed to support the fact that the robust control methodology in the designing stage of the controller is compensating for the disturbances while guaranteeing that the tracking position and velocity errors converge asymptotically to zero.
- An effective practical compensation for friction, not involving a cumbersome dynamic model of friction and staying as simple as possible in stability analysis within a passivity framework, is presented, performing the tracking of the desired trajectories in joint space successfully.

2. Problem Formulation

2.1. Robot Manipulator Model

The dynamic model in the joint-space of a general n-degrees-of-freedom robot manipulator with rigid links considering friction in the joints can be written in the compact vectorial form [19,20]:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{f}(\dot{\boldsymbol{q}},t) = \boldsymbol{\tau}\,, \tag{1}$$

where the joint positions, velocities, and acceleration vectors are indicated as $\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}} \in \mathbb{R}^{n\times 1}$, respectively. $\boldsymbol{M} \in \mathbb{R}^{n\times n}$ denotes the inertia matrix, and $\boldsymbol{C} \in \mathbb{R}^{n\times n}$ denotes a centripetal-Coriolis matrix. The vector of generalized forces due to gravity is represented as $\boldsymbol{g} \in \mathbb{R}^{n\times 1}$, the vector of non-conservative forces due to friction is designated as $\boldsymbol{f} \in \mathbb{R}^{n\times 1}$, and $\boldsymbol{\tau} \in \mathbb{R}^{n\times 1}$ denotes the vector of exogenous torques.

In regards to the vector of non-conservative forces due to friction, a time-variant model is taken into account to describe two major contribution components of this phenomenon with non-lumped elements, exhibiting regimes of transition. These include the one that considers the anisotropy of the Coulomb-like friction coefficients and the one that considers the transition threshold due to the non-linearity close to zero velocity in the viscous-like friction coefficients. Thus, the model of $\boldsymbol{f}$ is composed as follows:

$$\boldsymbol{f}(\dot{\boldsymbol{q}}, t) = \boldsymbol{F}_C(t)\,\mathbf{tanh}(k\dot{\boldsymbol{q}}) + \boldsymbol{F}_v(t)\dot{\boldsymbol{q}}\,, \tag{2}$$

where $\boldsymbol{F}_C(t) \in \mathbb{R}^{n\times n}$ is a diagonal matrix whose elements are the coefficients of the Coulomb-like friction, the first term of (2), and $\boldsymbol{F}_v(t) \in \mathbb{R}^{n\times n}$ is a diagonal matrix whose elements are the coefficients of the viscous-like friction, the second term of (2). On the other hand, the hyperbolic tangent function in its vectorial form is denoted as $\mathbf{tanh}(k\dot{\boldsymbol{q}}) \in \mathbb{R}^{n\times 1}$, with $k > 0$, and it is used as a regularization of the typical discontinuous sign function, i.e., as a continuous relation dependent of the joint velocity [21]. Furthermore, the election of time-varying coefficients at the Coulomb-like friction term aims to describe the change in the magnitude of the non-conservative force contribution when the joint changes its direction of rotation; i.e., friction exhibits an anisotropic behavior since its effects do not have the same magnitude in all directions. Moreover, the use of time-varying coefficients at the viscous-like friction term is a technical proposal introduced to facilitate the transition from the static regime to the dynamic regimen.

2.2. Properties of the Dynamic Model

All the dynamics of the specific kind of robotic manipulator previously described have been captured in model (1), i.e., when no other unmodeled dynamic or perturbation is involved in the plant. Some fundamental properties of the dynamic model for this manipulator are presented in this section for its use in the control design stage. All of the properties and their proofs can be consulted in [19,20,22].

Property 1. *The dependence of the conservative part of the manipulator model on the dynamic parameters of each of its individual terms is linearly parametrizable in terms of a new set of unknown parameters* $\boldsymbol{\theta}^* \in \mathbb{R}^{p\times 1}$ *in the form*

$$\boldsymbol{M}(\boldsymbol{q}, \boldsymbol{\theta}^*)\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\theta}^*)\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}, \boldsymbol{\theta}^*) = \boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\boldsymbol{\theta}^*\,, \tag{3}$$

where $\boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) \in \mathbb{R}^{n\times p}$ *is the regressor matrix, which is only dependent on the trajectories.*

Property 2. *The centripetal Coriolis matrix* $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ *is related to the inertia matrix* $\boldsymbol{M}(\boldsymbol{q})$ *by the identity*

$$\boldsymbol{q}^T\left[\dot{\boldsymbol{M}}(\boldsymbol{q}) - 2\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\right]\boldsymbol{q} = 0, \qquad \forall \boldsymbol{q}, \dot{\boldsymbol{q}} \in \mathbb{R}^{n\times 1} \tag{4}$$

since the matrix $\dot{\boldsymbol{M}}(\boldsymbol{q}) - 2\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ *is skew-symmetric.*

Property 3. *The inertia matrix* $\boldsymbol{M} = \boldsymbol{M}^T \in \mathbb{R}^{n\times n}$ *is a symmetric, uniformly positive definite* $\boldsymbol{M}(\boldsymbol{q}) > 0$ *for all* $\boldsymbol{q}$ *and uniformly bounded above and below by*

$$\mu_1\mathbb{I} \leq \boldsymbol{M}(\boldsymbol{q}) \leq \mu_2\mathbb{I} < \infty, \qquad \mu_1, \mu_2 > 0, \mathbb{I} \in \mathbb{R}^{n\times n}\,. \tag{5}$$

Furthermore, in order to exploit the dissipative characteristic of friction, the vector $\boldsymbol{f}(\dot{\boldsymbol{q}}, t)$ in (2) will be treated as an additive disturbance. First, a convenient linear reparametrization in friction coefficients is presented:

$$\boldsymbol{f}(\dot{\boldsymbol{q}}, t) = \operatorname{diag}(\mathbf{tanh}(k\dot{\boldsymbol{q}}))\operatorname{diag}(\boldsymbol{F}_C(t)) + \operatorname{diag}(\dot{\boldsymbol{q}})\operatorname{diag}(\boldsymbol{F}_v(t))\,, \tag{6}$$

where, for the present work, the diagonal operator takes a matrix of $n \times n$ and returns a $n \times 1$ vector, whose elements are the matrix diagonal elements; alternatively, it takes a $n \times 1$ vector and returns a diagonal matrix of $n \times n$, whose elements are the elements of the vector.

Property 4. *For a robot manipulator with anisotropic friction and continuous transition regimens around zero velocity in its joints, a reparametrization in terms of the friction coefficient can be express as*

$$\boldsymbol{f}(\dot{\boldsymbol{q}}, t) = \boldsymbol{Y}_C(\dot{\boldsymbol{q}})\boldsymbol{\theta}_C(t) + \boldsymbol{Y}_v(\dot{\boldsymbol{q}})\boldsymbol{\theta}_v(t)\,, \tag{7}$$

where $\boldsymbol{Y}_C(\dot{\boldsymbol{q}}) \in \mathbb{R}^{n\times n}$ is the regressor matrix associated with the time-varying Coulomb-like friction coefficients $\boldsymbol{\theta}_C(t) \in \mathbb{R}^{n\times 1}$. Similarly, $\boldsymbol{Y}_v(\dot{\boldsymbol{q}}) \in \mathbb{R}^{n\times n}$ is the regressor matrix associated with the time-varying viscous-like friction coefficients $\boldsymbol{\theta}_v(t) \in \mathbb{R}^{n\times 1}$.

2.3. Control Objective and Identification Objective

Consider the robot model (1): the control objective is that the joint positions of the robot $\boldsymbol{q}$ track the desired position trajectories $\boldsymbol{q}_d(t)$ asymptotically, i.e.,

$$\lim_{t\to\infty}[\boldsymbol{q}_d(t) - \boldsymbol{q}(t)] = \mathbf{0} \in \mathbb{R}^n,$$

for all initial configurations $(\boldsymbol{q}(0), \boldsymbol{q}_d(0))$, where $\boldsymbol{q}_d(t)$ is an arbitrarily chosen continuous function and at least once differentiable. Moreover, it is supposed that $\boldsymbol{q}_d(t)$ and $\dot{\boldsymbol{q}}_d(t)$ are known bounded vector functions, but the inertial and geometric parameter vector $\boldsymbol{\theta}^*$, the Coulomb-like friction coefficients vector $\boldsymbol{\theta}_C(t)$, and the viscous-like friction coefficients vector $\boldsymbol{\theta}_v(t)$ are not exactly known. Thus, the identification objective is that the estimation of the parameter vectors $\boldsymbol{\theta}(t)$, $\hat{\boldsymbol{\theta}}_C(t)$, and $\hat{\boldsymbol{\theta}}_v(t)$ track the nominal parameter vectors $\boldsymbol{\theta}^*$, $\boldsymbol{\theta}_C(t)$, and $\boldsymbol{\theta}_v(t)$ asymptotically:

$$\begin{aligned}
\lim_{t\to\infty}[\boldsymbol{\theta}^* - \boldsymbol{\theta}(t)] &= \mathbf{0} \in \mathbb{R}^{p\times 1}\,,\\
\lim_{t\to\infty}\left[\boldsymbol{\theta}_C(t) - \hat{\boldsymbol{\theta}}_C(t)\right] &= \mathbf{0} \in \mathbb{R}^{n\times 1}\,,\\
\lim_{t\to\infty}\left[\boldsymbol{\theta}_v(t) - \hat{\boldsymbol{\theta}}_v(t)\right] &= \mathbf{0} \in \mathbb{R}^{n\times 1}\,.
\end{aligned}$$

2.4. Motivation of the Proposed Scheme

Considering the major drawbacks of existing dynamic friction models, a more practical scheme is desired for robotics applications, i.e., one that does not require the further empirical investigation of any other interdisciplinary fields, such as tribology, materials science, or solid mechanics, but one that is entirely based in control theory and the properties of the robot manipulator as a mechanical system. Inspired by the fact that the friction phenomenon must be characterized by a dissipative behavior, as well as previous results [23,24], the passivity framework seems reasonable to explore. As an alternative to the model-based methods used to describe friction, a straightforward method for friction compensation can be implemented by assuming the basic structure of the friction phenomenon, introducing time-varying coefficients to estimate the amplitude and timing

of the contribution of each of the two major components of friction and determining an identification algorithm that satisfies this purpose using adaptive control theory.

This approach of treating friction allows us to parameterize the dynamic model in a novel way, and, combined with the passivity framework, it allows us to deal with the non-conservative part of the dynamics in the stability analysis and the identification algorithm analysis to guarantee the control objective and parametric convergence. For the former, it allows us to propose simple quadratic Lyapunov functions for each estimator and then only to add them, making the analysis as simple as it can be. For the latter, it simplifies the process of finding the bounds of the regressor matrix of each estimator and the proof that they satisfy the persistent excitation condition. All of this leads to a complete estimated feed-forward compensation that, along with an in-disguise PD controller using a sliding mode variable $\boldsymbol{s}$, is sufficient to prove, to the best of the authors' knowledge, the control objective and the identification objective previously stated in the easiest way, demonstrating quite-good experimental performance with a two-degree-of-freedom robotic arm.

The main benefits from the choice of the sliding mode control approach are the following:

- It makes the control objective straightforward to conclude since the variable $\boldsymbol{s}$ conveys enough information of $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$ in order to show that its boundedness and convergence to zero imply those of $\tilde{\boldsymbol{q}}$ and $\dot{\tilde{\boldsymbol{q}}}$.
- It removes the computational complexity from the computed torque as the inverse inertia matrix does not have to be determined.
- Since it is a robust control method, and since the two major components of friction are considered as disturbances, it rejects the small changes in the disturbances, i.e., the transition between static and dynamic regimes.

The latter phenomenon is always present in most applications with an inherently digital controller as the physical realization of the controller is established in a discrete time domain via two processes: amplitude discretization, referred to as quantization, and time discretization, referred to as sampling [25]. This induces (frequently underestimated) non-linear unmodeled dynamics, ubiquitous in the triggering of all actuators, since every time the actuator is activated, there is small interval of time at which the friction force transitions between a quasistatic state, in which movement is set to stand-by, and a motion state after overcoming a threshold to break the equilibrium of internal conservative forces, i.e., a state where the elastic deformation between the surfaces is present without sliding.

3. Preliminaries

3.1. Lyapunov Stability

Some basic concepts to prove the uniform global asymptotic stability of a dynamic system are presented next [22]. Consider a general dynamical system described by

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t, \boldsymbol{x}(t)), \qquad \boldsymbol{x}(0) \in \mathbb{R}^n, \, \forall t \geq 0\,, \tag{8}$$

where the vector $\boldsymbol{x}(t) \in \mathbb{R}^n$ refers to the state of the dynamical system in (8), and $\boldsymbol{x}(0) \in \mathbb{R}^n$ is called the initial condition. A constant vector $\boldsymbol{x}_e \in \mathbb{R}^n$ is an equilibrium or equilibrium state of the system (8) if

$$\boldsymbol{f}(t, \boldsymbol{x}_e) = \boldsymbol{0} \qquad \forall t \geq 0\,.$$

A function $V : \mathbb{R}_+ \times \mathbb{R}^n \to \mathbb{R}_+$ is a Lyapunov function for the equilibrium $\boldsymbol{x} = \boldsymbol{0}$ of the equation (8) if the following apply:

1. $V(t, \boldsymbol{0}) = 0, \qquad \forall t \geq 0$
2. $V(t, \boldsymbol{x}) > 0, \qquad \forall \boldsymbol{x} \neq \boldsymbol{0} \in \mathbb{R}^n, \, \forall t \geq 0$
3. Its partial derivatives are continuous.

4. Its total time derivative along the trajectories of (8) satisfies $\dot{V}(t, \boldsymbol{x}) \leq 0, \quad \forall \boldsymbol{x} \in \mathbb{R}^n, \forall t \geq 0.$

A function $W : \mathbb{R}^n \to \mathbb{R}$ is said to be radially unbounded if

$$\left[\lim_{\|\boldsymbol{x}\| \to \infty} W(\boldsymbol{x}) \right] \to \infty .$$

A continuous function $V : [0, \infty) \times \mathbb{R}^n \to \mathbb{R}$ is decrescent if there exists a positive definite function $W : \mathbb{R}^n \to [0, \infty)$ such that

$$V(t, \boldsymbol{x}) \leq W(\boldsymbol{x}) \qquad \forall t \geq 0, \forall \boldsymbol{x} \in \mathbb{R}^n .$$

Finally, the origin $\boldsymbol{x} = \mathbf{0} \in \mathbb{R}^n$ is the only globally uniformly asymptotically stable equilibrium state of (8) if there exists a radially unbounded and decrescent Lyapunov function such that its time derivative is negative definite.

3.2. Case of Study

Consider the fully actuated serial-link robot manipulator with two degrees of freedom in Figure 2, whose position $\boldsymbol{q} \in \mathbb{R}^{2\times 1}$ can be described by the measurement of the angles associated with each joint, and whose physical inputs $\boldsymbol{\tau} \in \mathbb{R}^{2\times 1}$ are given by the torques of two actuators, one in each joint of the manipulator. The physical parameters of the robotic arm are summarized in Table 1.

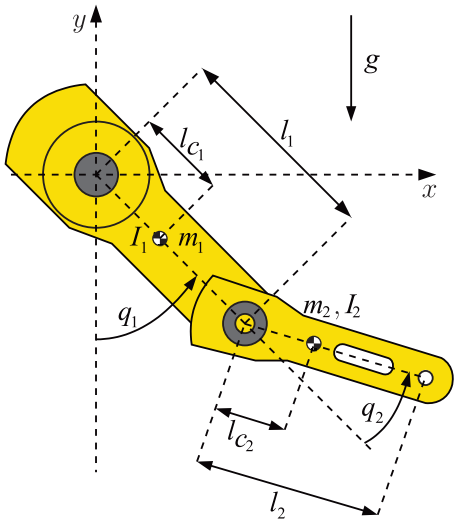


Figure 2. Diagram of the 2-DOF robot manipulator [26].

Table 1. Physical parameters of the 2-DOF robot manipulator.

Description	Notation	Units
Length of link 1	l_1	m
Length of link 2	l_2	m
Distance to the center of mass of link 1	l_{c_1}	m
Distance to the center of mass of link 2	l_{c_2}	m
Mass of link 1	m_1	kg
Mass of link 2	m_2	kg
Inertia related to center of mass of link 1	I_1	$\mathrm{kg} \cdot \mathrm{m}^2$
Inertia related to center of mass of link 2	I_2	$\mathrm{kg} \cdot \mathrm{m}^2$
Acceleration due to gravity	g	$\frac{\mathrm{m}}{\mathrm{s}^2}$

The dynamic model of the robot manipulator in (1) can be explicitly written as

$$\begin{bmatrix} M_{11}(\boldsymbol{q}) & M_{12}(\boldsymbol{q}) \\ M_{21}(\boldsymbol{q}) & M_{22}(\boldsymbol{q}) \end{bmatrix} \ddot{\boldsymbol{q}} + \begin{bmatrix} C_{11}(\boldsymbol{q},\dot{\boldsymbol{q}}) & C_{12}(\boldsymbol{q},\dot{\boldsymbol{q}}) \\ C_{21}(\boldsymbol{q},\dot{\boldsymbol{q}}) & C_{22}(\boldsymbol{q},\dot{\boldsymbol{q}}) \end{bmatrix} \dot{\boldsymbol{q}} + \begin{bmatrix} g_1(\boldsymbol{q}) \\ g_2(\boldsymbol{q}) \end{bmatrix} + \boldsymbol{f}(\dot{\boldsymbol{q}},t) = \boldsymbol{\tau} \tag{9}$$

where

$$\begin{aligned} M_{11}(\boldsymbol{q}) &= m_1 l_{c_1}^2 + m_2[l_1^2 + l_{c_1}^2 + 2l_1 l_{c_2}\cos(q_2)] + I_1 + I_2, \\ M_{12}(\boldsymbol{q}) &= m_2\left[l_{c2}^2 + l_1\, l_{c2}\, \cos\left(q_2\right)\right] + I_2, \\ M_{21}(\boldsymbol{q}) &= m_2\left[l_{c2}^2 + l_1\, l_{c2}\, \cos\left(q_2\right)\right] + I_2, \\ M_{22}(\boldsymbol{q}) &= m_2 l_{c_2}^2 + I_2, \\ C_{11}(\boldsymbol{q},\dot{\boldsymbol{q}}) &= -m_2 l_1 l_{c_2}\sin(q_2)\dot{q}_2, \\ C_{12}(\boldsymbol{q},\dot{\boldsymbol{q}}) &= -m_2 l_1 l_{c_2}\sin(q_2)\left[\dot{q}_1 + \dot{q}_2\right], \\ C_{21}(\boldsymbol{q},\dot{\boldsymbol{q}}) &= m_2 l_1 l_{c_2}\sin(q_2)\dot{q}_1, \\ C_{22}(\boldsymbol{q},\dot{\boldsymbol{q}}) &= 0, \\ g_1(\boldsymbol{q}) &= [m_1 l_{c_1} + m_2 l_1]g\ \sin(q_1) + m_2 l_{c_2} g\ \sin(q_1+q_2), \\ g_2(\boldsymbol{q}) &= m_2 l_{c_2} g\sin(q_1+q_2). \end{aligned}$$

For a two link robot manipulator with only revolute joints, Property 4 in (10) can be explicitly written as follows

$$\boldsymbol{Y}_C(\dot{\boldsymbol{q}})\boldsymbol{\theta}_C(t) + \boldsymbol{Y}_v(\dot{\boldsymbol{q}})\boldsymbol{\theta}_v(t) = \begin{bmatrix} \tanh(k\dot{q}_1) & 0 \\ 0 & \tanh(k\dot{q}_2) \end{bmatrix} \begin{bmatrix} f_{C_1}(t) \\ f_{C_2}(t) \end{bmatrix} + \begin{bmatrix} \dot{q}_1 & 0 \\ 0 & \dot{q}_2 \end{bmatrix} \begin{bmatrix} f_{v_1}(t) \\ f_{v_2}(t) \end{bmatrix}. \tag{10}$$

Thus, by choosing a vector $\boldsymbol{\theta}^* \in \mathbb{R}^{4\times 1}$ of unknown but constant parameters composed of a non-linear combination of physical inertia and geometric parameters, the robot manipulator in (9) can be parameterized as Properties 1 and 4, indicated by

$$\boldsymbol{\tau} = \begin{bmatrix} Y_{11} & Y_{12} & Y_{13} & Y_{14} \\ Y_{21} & Y_{22} & Y_{23} & Y_{24} \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} + \begin{bmatrix} \tanh(k\dot{q}_1) & 0 \\ 0 & \tanh(k\dot{q}_2) \end{bmatrix} \begin{bmatrix} f_{C_1}(t) \\ f_{C_2}(t) \end{bmatrix} + \begin{bmatrix} \dot{q}_1 & 0 \\ 0 & \dot{q}_2 \end{bmatrix} \begin{bmatrix} f_{v_1}(t) \\ f_{v_2}(t) \end{bmatrix} \tag{11}$$

where

$$\begin{aligned} Y_{11} &= \ddot{q}_1, \\ Y_{12} &= l_1\cos(q_2)(2\ddot{q}_1 + \ddot{q}_2) - l_1\sin(q_2)\left(2\dot{q}_1\dot{q}_2 + \dot{q}_2^2\right) + g\sin(q_1+q_2), \\ Y_{13} &= \ddot{q}_2, \\ Y_{14} &= g\sin\left(q_1\right), \\ Y_{21} &= 0, \\ Y_{22} &= l_1\cos(q_2)\ddot{q}_1 + l_1\sin(q_2)\dot{q}_1^2 + g\sin(q_1+q_2), \\ Y_{23} &= \ddot{q}_1 + \ddot{q}_2, \\ Y_{24} &= 0, \\ \theta_1 &= I_1 + m_1 {l_{c_1}}^2 + I_2 + m_2 {l_{c_2}}^2 + m_2 {l_1}^2, \\ \theta_2 &= m_2 l_{c_2}, \\ \theta_3 &= I_2 + m_2 {l_{c_2}}^2, \\ \theta_4 &= m_1 l_{c_1} + m_2 l_1. \end{aligned}$$

It is worth noting that this parametrization has no prior knowledge other than the acceleration due to gravity and the length of the first link, i.e., l_1.

3.3. Identification Algorithm

In order to derive an algorithm to estimate the unknown parameters of Properties 1 and 4, first, the model at (1) can be expressed as

$$\boldsymbol{\tau} = \boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\boldsymbol{\theta}^* + \boldsymbol{Y}_C(\dot{\boldsymbol{q}})\boldsymbol{\theta}_C(t) + \boldsymbol{Y}_v(\dot{\boldsymbol{q}})\boldsymbol{\theta}_v(t)\,. \tag{12}$$

As the first regressor matrix $\boldsymbol{Y}$ depends on the joint acceleration $\ddot{\boldsymbol{q}}$, the swapping technique can be used to avoid the measurement of the acceleration signal in the experimental implementation [27]; this consists in filtering both parts of (12) with a stable first-order filter $w(t) = \lambda_f \mathrm{e}^{-\lambda_f t}$ with positive adaptive gain λ_f. Focusing on the first term in (12), such avoidance can be derived as follows:

$$\begin{aligned}
\boldsymbol{W}(\boldsymbol{q}, \dot{\boldsymbol{q}})\boldsymbol{\theta}^* &= w(t) * [\boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\boldsymbol{\theta}^*]\,, \\
&= \int_0^t \lambda_f \mathrm{e}^{-\lambda_f (t-\sigma)} [\boldsymbol{M}(\boldsymbol{q}(\sigma))\ddot{\boldsymbol{q}}(\sigma) + \boldsymbol{C}(\boldsymbol{q}(\sigma), \dot{\boldsymbol{q}}(\sigma))\dot{\boldsymbol{q}}(\sigma) + \boldsymbol{g}(\boldsymbol{q}(\sigma))] \mathrm{d}\sigma\,, \\
&= \int_0^t \lambda_f \mathrm{e}^{-\lambda_f (t-\sigma)} \left[\frac{\mathrm{d}(\boldsymbol{M}\dot{\boldsymbol{q}})}{\mathrm{d}\sigma} - \dot{\boldsymbol{M}}\dot{\boldsymbol{q}} + \boldsymbol{C}\dot{\boldsymbol{q}} + \boldsymbol{g} \right] \mathrm{d}\sigma\,, \\
&= \int_0^t \lambda_f \mathrm{e}^{-\lambda_f (t-\sigma)} \mathrm{d}(\boldsymbol{M}\dot{\boldsymbol{q}}) + \int_0^t \lambda_f \mathrm{e}^{-\lambda_f (t-\sigma)} \left[-\dot{\boldsymbol{M}}\dot{\boldsymbol{q}} + \boldsymbol{C}\dot{\boldsymbol{q}} + \boldsymbol{g} \right] \mathrm{d}\sigma\,, \\
&= \left[\lambda_f \mathrm{e}^{-\lambda_f (t-\sigma)} \boldsymbol{M}(\boldsymbol{q}(\sigma))\dot{\boldsymbol{q}}(\sigma) \right]_0^t + \int_0^t \lambda_f \mathrm{e}^{-\lambda_f (t-\sigma)} \left[\boldsymbol{g} - \left(\boldsymbol{C}^\top + \lambda_f \boldsymbol{M} \right) \dot{\boldsymbol{q}} \right] \mathrm{d}\sigma\,, \\
&= \lambda_f \boldsymbol{M}\dot{\boldsymbol{q}} + \lambda_f \mathrm{e}^{-\lambda_f t} * \left[\boldsymbol{g} - \left(\boldsymbol{C}^\top + \lambda_f \boldsymbol{M} \right) \dot{\boldsymbol{q}} \right] - \lambda_f \mathrm{e}^{-\lambda_f t} \boldsymbol{M}(\boldsymbol{q}(0))\dot{\boldsymbol{q}}(0)\,, \\
&= [\delta(t) - w] * \lambda_f \boldsymbol{M}(\boldsymbol{q}, \boldsymbol{\theta}^*)\dot{\boldsymbol{q}} + w * \left[\boldsymbol{g}(\boldsymbol{q}, \boldsymbol{\theta}^*) - \boldsymbol{C}^\top(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\theta}^*)\dot{\boldsymbol{q}} - \boldsymbol{M}(\boldsymbol{q}_0)\dot{\boldsymbol{q}}_0 \delta(t) \right].
\end{aligned}$$

where $\boldsymbol{q}(0) = \boldsymbol{q}_0$, $\dot{\boldsymbol{q}}(0) = \dot{\boldsymbol{q}}_0$, and $\delta(t)$ denote the delta distribution. Finally, it can be seen that all the terms depend linearly on $\boldsymbol{\theta}^*$, and the remaining linear operator can be written as a matrix $\boldsymbol{W}$ only depending on $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$. Now, this linear relation may be written in such a way that the new regressor matrix of inertial and geometry parameters is given by

$$\boldsymbol{W}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \lambda_f \mathrm{e}^{-\lambda_f t} * \boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\,. \tag{13}$$

Thus, filtering both sides of (12) in a similar fashion leads to

$$\boldsymbol{y}(t) = \boldsymbol{W}(\boldsymbol{q}, \dot{\boldsymbol{q}})\boldsymbol{\theta}^* + \boldsymbol{W}_C(\dot{\boldsymbol{q}}_r)\boldsymbol{\theta}_C(t) + \boldsymbol{W}_v(\dot{\boldsymbol{q}}_r)\boldsymbol{\theta}_v(t)\,, \tag{14}$$

where the filtered torque is denoted as $\boldsymbol{y} = w(t) * \boldsymbol{\tau}$,

$$\boldsymbol{W}_C(\dot{\boldsymbol{q}}_r) = \lambda_f \mathrm{e}^{-\lambda_f t} * \boldsymbol{Y}_C(\dot{\boldsymbol{q}})\,,$$

$$\boldsymbol{W}_v(\dot{\boldsymbol{q}}_r) = \lambda_f \mathrm{e}^{-\lambda_f t} * \boldsymbol{Y}_v(\dot{\boldsymbol{q}})\,,$$

and $\dot{\boldsymbol{q}}_r$ is a reference velocity resulting from shifting the desired velocities $\dot{\boldsymbol{q}}_d$ according to the position error, as will be formally defined later on. Then, defining the identification error as the difference between the estimate of the filtered torque and the actual filtered torque, it can be expressed as follows:

$$\boldsymbol{e}_i(t) = \boldsymbol{W}\boldsymbol{\theta}(t) + \boldsymbol{W}_C\hat{\boldsymbol{\theta}}_C(t) + \boldsymbol{W}_v\hat{\boldsymbol{\theta}}_v(t) - \boldsymbol{y}(t) = \boldsymbol{W}\tilde{\boldsymbol{\theta}}(t) + \boldsymbol{W}_C\tilde{\boldsymbol{\theta}}_C(t) + \boldsymbol{W}_v\tilde{\boldsymbol{\theta}}_v(t)\,. \tag{15}$$

Furthermore, it is intuitive that the identification error in (15) will decrease if each of its terms is close to zero; for instance, if $\tilde{\boldsymbol{\theta}}(t)$ is zero, then the inertial and geometric estimated parameters will be exactly the nominal parameters since these would exactly generate the response of the plant. Thus, it is essential to find an *update law* that minimizes this error according to some criterion. In this work, the criterion used is the minimization of the integral of the squared error. For the first term, the criterion is

$$J[\boldsymbol{\tau}(t), \boldsymbol{q}(t), t] = \int_0^t \left[\boldsymbol{W}(\tau)\tilde{\boldsymbol{\theta}}(\tau)\right]^2 \mathrm{d}\tau\,. \tag{16}$$

Then, it can be proven that the update law of the parametric estimates that minimizes the functional in (16) is the gradient update law:

$$\dot{\boldsymbol{\theta}}(t) = -\gamma \boldsymbol{W}^{\mathsf{T}}(t)\boldsymbol{W}(t)\tilde{\boldsymbol{\theta}}(t)\,, \tag{17}$$

since $\dot{\tilde{\boldsymbol{\theta}}}(t) = \dot{\boldsymbol{\theta}}(t)$, defining $\boldsymbol{A}(t) = \gamma \boldsymbol{W}(t)\boldsymbol{W}^{\mathsf{T}}(t)$ with $\gamma > 0$ and $\boldsymbol{A}(t) \geq 0 \in \mathbb{R}^{p\times p}$, $\forall t > 0$. Then, equation (17) can be expressed as the time-varying first-order differential equation:

$$\dot{\tilde{\boldsymbol{\theta}}}(t) = -\boldsymbol{A}(t)\tilde{\boldsymbol{\theta}}(t)\,. \tag{18}$$

By way of linear time-varying system theory, it can be concluded that if the time-varying coefficient $\boldsymbol{A}(t)$ in (18) can be bounded above and below by a positive definite constant matrix over a finite interval, then the system described by (18) is asymptotically stable; this leads to the fundamental definition of the persistent excitation of a matrix given in [28]. A matrix $\boldsymbol{W} : [0, \infty) \to \mathbb{R}^{n\times p}$ is persistently exciting if there exists $\alpha_1, \alpha_2, \delta_1 > 0$ such that

$$\alpha_1 \mathbb{I} \leq \int_{t_0}^{t_0+\delta_1} \boldsymbol{W}^{\mathsf{T}}(t)\boldsymbol{W}(t)\mathrm{d}t \leq \alpha_2 \mathbb{I}, \quad \forall t_0 \geq 0\,, \tag{19}$$

where $\mathbb{I} \in \mathbb{R}^{n\times n}$.

Similar arguments can be applied to the rest of terms in (15), and this leads to the following conditions. A matrix $\boldsymbol{W}_C : [0, \infty) \to \mathbb{R}^{n\times n}$ is persistently exciting if there exists $\beta_1 \geq 0$ and $\beta_2, \delta_2 > 0$ such that

$$\beta_1 \mathbb{I} \leq \int_{t_0}^{t_0+\delta_2} [\boldsymbol{W}_C(\dot{\boldsymbol{q}}_r(t))]^{\mathsf{T}} \boldsymbol{W}_C(\dot{\boldsymbol{q}}_r(t))\mathrm{d}t \leq \beta_2 \mathbb{I}, \quad \forall t_0 \geq 0\,. \tag{20}$$

While a matrix $\boldsymbol{W}_v : [0, \infty) \to \mathbb{R}^{n\times n}$ is persistently exciting if there exists $\gamma_1 \geq 0$ and $\gamma_2, \delta_3 > 0$ such that

$$\gamma_1 \mathbb{I} \leq \int_{t_0}^{t_0+\delta_3} [\boldsymbol{W}_v(\dot{\boldsymbol{q}}_r(t))]^{\mathsf{T}} \boldsymbol{W}_v(\dot{\boldsymbol{q}}_r(t))\mathrm{d}t \leq \gamma_2 \mathbb{I}, \quad \forall t_0 \geq 0\,. \tag{21}$$

For a two-rigid-links robot manipulator with only revolute joints, some boundaries for the regressor matrices associated with the friction parameters can be obtained as follows:

$$\boldsymbol{W}_C(\dot{\boldsymbol{q}}_r) \leq \begin{bmatrix} \|\lambda_f \mathrm{e}^{-\lambda_f t} * \tanh(k\dot{q}_1)\| & 0 \\ 0 & \|\lambda_f \mathrm{e}^{-\lambda_f t} * \tanh(k\dot{q}_2)\| \end{bmatrix} \leq \lambda_f \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{22}$$

$$\boldsymbol{W}_v(\dot{\boldsymbol{q}}_r) \leq \begin{bmatrix} \|\dot{q}_{d_1} - \lambda_f \tilde{q}_1\| & 0 \\ 0 & \|\dot{q}_{d_2} - \lambda_f \tilde{q}_2\| \end{bmatrix} \leq \begin{bmatrix} |\dot{q}_{d_1}| + \lambda_f & 0 \\ 0 & |\dot{q}_{d_2}| + \lambda_f \end{bmatrix}. \tag{23}$$

This results in

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \leq \int_{t_0}^{t_0+\delta_2} [\boldsymbol{W}_C(\dot{\boldsymbol{q}}_r(t))]^{\mathsf{T}} \boldsymbol{W}_C(\dot{\boldsymbol{q}}_r(t))\mathrm{d}t \leq {\lambda_f}^2 \delta_2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{24}$$

and

$$\lambda_f{}^2\delta_3\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} \leq \int_{t_0}^{t_0+\delta_3}[\boldsymbol{W}_v(\dot{\boldsymbol{q}}_r(t))]^{\mathsf{T}}\boldsymbol{W}_v(\dot{\boldsymbol{q}}_r(t))\mathrm{d}t \leq \begin{bmatrix}\left(c_1+\lambda_f\right)^2\delta_3 & 0\\ 0 & \left(c_2+\lambda_f\right)^2\delta_3\end{bmatrix}, \quad (25)$$

since $\|\dot{\boldsymbol{q}}_d\|$ is bounded by some finite number by assumption. Thus, conditions (20) and (21) can be satisfied.

4. Controller Design

Given the robot manipulator model in (1), explicitly determined in (9) and (10), it is clear that its structure is non-linear and coupled as the dynamic of the first link affects the dynamic of the second one, and vice versa. Nevertheless, feedback linearization can be performed to obtain a transformation that leads to linear closed-loop dynamics by canceling the non-linearities of the model. This technique is frequently called computed torque, and it is indeed a global diffeomorphism guaranteed by Property 3, as proven in [29]. A general computed-torque control law is

$$\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{q})\boldsymbol{\nu} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{f}(\dot{\boldsymbol{q}},t)\,, \quad (26)$$

with $\boldsymbol{\nu} = \ddot{\boldsymbol{q}}_d - 2\boldsymbol{\Lambda}\dot{\tilde{\boldsymbol{q}}} - \boldsymbol{\Lambda}^2\tilde{\boldsymbol{q}}$. Then, the tracking error $\tilde{\boldsymbol{q}} = \boldsymbol{q} - \boldsymbol{q}_d$ satisfies the following closed-loop equation:

$$\ddot{\tilde{\boldsymbol{q}}} + 2\boldsymbol{\Lambda}\dot{\tilde{\boldsymbol{q}}} + \boldsymbol{\Lambda}^2\tilde{\boldsymbol{q}} = \boldsymbol{0}, \qquad \boldsymbol{\Lambda} \in \mathbb{R}^{n\times n} > 0 \quad (27)$$

Therefore, $\tilde{\boldsymbol{q}}$ converges to zero exponentially. Clearly, it is implicitly assumed that the dynamics in the computed torque at (26) are known perfectly, which is not the case in adaptive control theory. To overcome this parametric uncertainty while guaranteeing zero steady-state position errors, the residual tracking errors are restricted to lying on a sliding surface, as reported in [30].

A new variable $\boldsymbol{s}$, called virtual velocity, is defined as the weighted sum of the position error and the velocity error.

$$\boldsymbol{s} = \dot{\tilde{\boldsymbol{q}}} + \boldsymbol{\Lambda}\tilde{\boldsymbol{q}}, \qquad \boldsymbol{\Lambda} = \boldsymbol{\Lambda}^{\mathsf{T}} > 0\,. \quad (28)$$

As this variable can be seen as a filtered tracking error if it converges to zero, $\tilde{\boldsymbol{q}}$ and $\dot{\tilde{\boldsymbol{q}}}$ will both tend to zero as $t \to \infty$, as proven in [30]. In order to achieve this, $\dot{\boldsymbol{q}}$ will be replaced by $\boldsymbol{s}$ for the design of the controller, which implies that the desired velocity $\dot{\boldsymbol{q}}_d$ needs to be replaced by the virtual *reference trajectory:*

$$\dot{\boldsymbol{q}}_r = \dot{\boldsymbol{q}}_d - \boldsymbol{\Lambda}\tilde{\boldsymbol{q}}\,. \quad (29)$$

4.1. Control Law and Composite Update Law by Lyapunov Stability Analysis

As the objective of this controller is to enhance performance in asymptotically tracking trajectories for robot manipulators in joint space, the central idea of control law design is to make use of the information of three independent estimators in both cases of parameter estimation, i.e., constant nominal parameters and time-varying parameters, while each parametric estimation error is bounded. First, the dynamical model of a robot manipulator (1) without non-conservative forces $\boldsymbol{f}(\dot{\boldsymbol{q}},t)$ is considered, which is the conservative part:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}\,. \quad (30)$$

Further, in the next subsection, additive disturbances will be employed to determine the stability of (1). Now, applying Property 1 to (30), the design of the adaptive mechanism for this subsystem involves obtaining an update law for the estimation of the unknown constant vector $\boldsymbol{\theta}^*$. For such a purpose, the Lyapunov method can be applied, considering the Lyapunov function:

$$V(\boldsymbol{s},t) = \frac{1}{2}\left[\boldsymbol{s}^{\mathsf{T}}\boldsymbol{M}\boldsymbol{s} + \tilde{\boldsymbol{\theta}}^{\mathsf{T}}{\boldsymbol{\Gamma}_1}^{-1}\tilde{\boldsymbol{\theta}}\right], \qquad \boldsymbol{\Gamma}_1 > 0\,. \tag{31}$$

Taking the derivative of (31) with respect to time results,

$$\dot{V}(\boldsymbol{s},t) = \boldsymbol{s}^{\mathsf{T}}\boldsymbol{M}\dot{\boldsymbol{s}} + \frac{1}{2}\boldsymbol{s}^{\mathsf{T}}\dot{\boldsymbol{M}}\boldsymbol{s} + \dot{\boldsymbol{\theta}}^{\mathsf{T}}{\boldsymbol{\Gamma}_1}^{-1}\tilde{\boldsymbol{\theta}}\,. \tag{32}$$

Now, from (28) and (29), it can be shown that $\dot{\boldsymbol{s}} = \ddot{\boldsymbol{q}} - \ddot{\boldsymbol{q}}_r$; from this fact, in (30), it follows that

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} = \boldsymbol{\tau} - \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} - \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})[\boldsymbol{s} + \dot{\boldsymbol{q}}_r] - \boldsymbol{g}(\boldsymbol{q})\,,$$

and introducing this manipulation in (32) leads to

$$\dot{V} = \boldsymbol{s}^{\mathsf{T}}[\boldsymbol{\tau} - \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}}_r - \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}_r - \boldsymbol{g}(\boldsymbol{q})] + \frac{1}{2}\boldsymbol{s}^{\mathsf{T}}\left[\dot{\boldsymbol{M}}(\boldsymbol{q}) - 2\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\right]\boldsymbol{s} + \dot{\boldsymbol{\theta}}^{\mathsf{T}}{\boldsymbol{\Gamma}_1}^{-1}\tilde{\boldsymbol{\theta}}\,,$$

but by Property 2, it can be written as

$$\dot{V}(\boldsymbol{s},t) = \boldsymbol{s}^{\mathsf{T}}[\boldsymbol{\tau} - \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}}_r - \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}_r - \boldsymbol{g}(\boldsymbol{q})] + \dot{\boldsymbol{\theta}}^{\mathsf{T}}{\boldsymbol{\Gamma}_1}^{-1}\tilde{\boldsymbol{\theta}}\,. \tag{33}$$

Now, applying Property 1 in a similar fashion to (33), and using the fact that $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}_r = \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}_r)\dot{\boldsymbol{q}}$, which implies that $\dot{\boldsymbol{q}}$ can be replace by $\dot{\boldsymbol{q}}_r$, a linear relationship in terms of its unknown parameters and the same structure of the regression matrix in (3) can be expressed as

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}}_r + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}_r)\dot{\boldsymbol{q}}_r + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}}_r,\ddot{\boldsymbol{q}}_r)\boldsymbol{\theta}^*\,. \tag{34}$$

Thus, the control law that determines the torques of the actuators of the robot manipulator can be chosen, being composed of an estimated feed-forward compensation and a PD controller as follows:

$$\boldsymbol{\tau} = \boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}}_r,\ddot{\boldsymbol{q}}_r)\boldsymbol{\theta}(t) - \boldsymbol{K}_D\boldsymbol{s}\,. \tag{35}$$

Finally, substituting (34) and the control law (35) yields the derivative of the Lyapunov function (33), expressed by

$$\dot{V}(\boldsymbol{s},t) = \boldsymbol{s}^{\mathsf{T}}\boldsymbol{Y}\tilde{\boldsymbol{\theta}} - \boldsymbol{s}^{\mathsf{T}}\boldsymbol{K}_D\boldsymbol{s} + \dot{\boldsymbol{\theta}}^{\mathsf{T}}{\boldsymbol{\Gamma}_1}^{-1}\tilde{\boldsymbol{\theta}}\,. \tag{36}$$

4.2. Composite Adaptation

A composite adaptation law extracts information about parameters not only from the tracking errors but also from the identification errors to achieve the parameter adaptation. Such a law was originally proposed in [1]. The following variant is proposed for the first estimator:

$$\dot{\boldsymbol{\theta}} = -\boldsymbol{\Gamma}_1\left[\boldsymbol{Y}^{\mathsf{T}}\boldsymbol{s} + \boldsymbol{W}^{\mathsf{T}}\boldsymbol{\kappa}\boldsymbol{e}_{i_1}\right], \tag{37}$$

where $\alpha > 0$, $\boldsymbol{\kappa} = \boldsymbol{\kappa}^{\mathsf{T}} > 0 \in \mathbb{R}^{n\times n}$ denotes the weighting matrix of the parametric identification information based on the identification error, and $\boldsymbol{\Gamma}_1 = {\boldsymbol{\Gamma}_1}^{\mathsf{T}} > 0 \in \mathbb{R}^{p\times p}$ denotes the gain matrix of the composite gradient algorithm.

Then, substituting (37) in (36) leads to

$$\dot{V}(\boldsymbol{s},t) = -\boldsymbol{s}^{\mathsf{T}}\boldsymbol{K}_D\boldsymbol{s} - \left(\boldsymbol{\kappa}\boldsymbol{e}_{i_1}\right)^{\mathsf{T}}\boldsymbol{W}\tilde{\boldsymbol{\theta}}\,. \tag{38}$$

Since $\boldsymbol{e}_{i_1} = \boldsymbol{W}\tilde{\boldsymbol{\theta}} \Rightarrow \boldsymbol{e}_{i_1}{}^{\mathsf{T}} = \tilde{\boldsymbol{\theta}}^{\mathsf{T}}\boldsymbol{W}^{\mathsf{T}}$, the derivative of the Lyapunov function (38) becomes

$$\dot{V}(\boldsymbol{s},t) = -\boldsymbol{s}^{\mathsf{T}}\boldsymbol{K}_D\boldsymbol{s} - \boldsymbol{e}_{i_1}{}^{\mathsf{T}}\boldsymbol{\kappa}\boldsymbol{e}_{i_1} < 0, \quad \forall \boldsymbol{s} \neq \boldsymbol{0}, \forall \boldsymbol{e}_i \neq \boldsymbol{0}\,. \tag{39}$$

This implies that $V(\boldsymbol{s}(t),t) \leq V(\boldsymbol{s}(0),0)$; therefore, $\boldsymbol{s}$ and $\tilde{\boldsymbol{\theta}}$ are bounded above by the construction of (31), which indicates that $V(t)$ will decrease whenever either the tracking error $\boldsymbol{s}$ or the identification error $\boldsymbol{e}_{i_1}$ is non-zero.

5. Additive Disturbances and Lyapunov Stability

Lyapunov theory, as a formalism, generalizes the notion of energy in physical systems to any abstract dynamic set of differential equations via the so-called Lyapunov functions. Of course, this generalization preserves some properties, such as the total energy in a physical system being the sum of the individual energy in each of the subsystems that compose such a system. While passivity formalism established laws by which to obtain such a generalization of energy by conveniently performing combinations of the Lyapunov functions considered for each these subsystems. When a non-conservative force is considered, it can be treated as a bounded (not necessarily continuous) disturbance in feedback control. The fundamental result of such an approach is the fact that a Lyapunov function for a general system can be determined by solely adding the Lyapunov functions describing each of its subsystems. This approach is often referred to as an additive disturbance, and in this work, friction effects will be dealt with via two independent subsystems that identify time-varying coefficients, as described by (7). Now, considering the system in (1) with non-conservative forces due to friction model (2), as well as the results in the previous subsection, the proposed control law is given by

$$\boldsymbol{\tau} = \boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}}_r,\ddot{\boldsymbol{q}}_r)\boldsymbol{\theta}(t) + \boldsymbol{Y}_C(\dot{\boldsymbol{q}}_r)\hat{\boldsymbol{\theta}}_C(t) + \boldsymbol{Y}_v(\dot{\boldsymbol{q}}_r)\hat{\boldsymbol{\theta}}_v(t) - \boldsymbol{K}_D\boldsymbol{s}\,. \tag{40}$$

with update laws (37) for the first term and

$$\dot{\hat{\boldsymbol{\theta}}}_C(t) = -\boldsymbol{\Gamma}_2\boldsymbol{Y}_C{}^{\mathsf{T}}(\dot{\boldsymbol{q}}_r)\boldsymbol{s}\,, \qquad \boldsymbol{\Gamma}_2 = \boldsymbol{\Gamma}_2{}^{\mathsf{T}} > 0\,, \tag{41}$$

$$\dot{\hat{\boldsymbol{\theta}}}_v(t) = -\boldsymbol{\Gamma}_3\boldsymbol{Y}_v{}^{\mathsf{T}}(\dot{\boldsymbol{q}}_r)\boldsymbol{s}\,, \qquad \boldsymbol{\Gamma}_3 = \boldsymbol{\Gamma}_3{}^{\mathsf{T}} > 0\,, \tag{42}$$

for the second and third terms, respectively. Then, the election of the Lyapunov function is as follows:

$$V(\boldsymbol{s},t) = \frac{1}{2}\left[\boldsymbol{s}^{\mathsf{T}}\boldsymbol{M}\boldsymbol{s} + \tilde{\boldsymbol{\theta}}^{\mathsf{T}}\boldsymbol{\Gamma}^{-1}\tilde{\boldsymbol{\theta}} + \tilde{\boldsymbol{\theta}}_C^{\mathsf{T}}\boldsymbol{\Gamma}_2^{-1}\tilde{\boldsymbol{\theta}}_C + \tilde{\boldsymbol{\theta}}_v^{\mathsf{T}}\boldsymbol{\Gamma}_3^{-1}\tilde{\boldsymbol{\theta}}_v\right]. \tag{43}$$

The time derivative of (43) along the trajectories of the system results in

$$\dot{V}(\boldsymbol{s},t) = \boldsymbol{s}^{\mathsf{T}}[\boldsymbol{\tau} - \boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}}_r,\ddot{\boldsymbol{q}}_r)\boldsymbol{\theta}^* - \boldsymbol{f}(\dot{\boldsymbol{q}},t)] + \dot{\boldsymbol{\theta}}^{\mathsf{T}}\boldsymbol{\Gamma}_1{}^{-1}\tilde{\boldsymbol{\theta}} + \dot{\boldsymbol{\theta}}_C^{\mathsf{T}}\boldsymbol{\Gamma}_2{}^{-1}\tilde{\boldsymbol{\theta}}_C + \dot{\boldsymbol{\theta}}_v^{\mathsf{T}}\boldsymbol{\Gamma}_3{}^{-1}\tilde{\boldsymbol{\theta}}_v\,. \tag{44}$$

Since $\boldsymbol{f}(\dot{\boldsymbol{q}},t;\dot{\boldsymbol{q}}_d) = \boldsymbol{Y}_C(\dot{\boldsymbol{q}}_r)\boldsymbol{\theta}_C(t) + \boldsymbol{Y}_v(\dot{\boldsymbol{q}}_r)\boldsymbol{\theta}_v(t)$, using (37), (41), and (42) leads to

$$\dot{V}(\boldsymbol{s},t) = -\boldsymbol{s}^{\mathsf{T}}[\boldsymbol{K}_D + \boldsymbol{F}_C(t) + \boldsymbol{F}_v(t)]\boldsymbol{s} - \boldsymbol{e}_{i_1}{}^{\mathsf{T}}\boldsymbol{\kappa}\boldsymbol{e}_{i_1} < 0, \quad \forall \boldsymbol{s} \neq \boldsymbol{0}, \forall \boldsymbol{e}_i \neq \boldsymbol{0}\,. \tag{45}$$

Of course, the friction effects in (2) are physically bounded above and below by some finite constant because Coulomb-like friction effects are, at most equal, to the force causing movement. Mathematically, this translates into a saturation with a non-symmetric threshold given by

$$\boldsymbol{F}_C(t) = \begin{cases} \boldsymbol{F}_{C_1}, & \dot{\boldsymbol{q}}(t) > 0, \\ \boldsymbol{0}, & \dot{\boldsymbol{q}}(t) = 0, \\ \boldsymbol{F}_{C_2}, & \dot{\boldsymbol{q}}(t) < 0. \end{cases} \tag{46}$$

On the other hand, viscous-like friction effects are, at most, equal to a factor of the normal force that ensures static equilibrium, and are at least equal to the force sufficient to hold the movement of an object. Every value in between depends on the direction of movement, if it was already in movement or at rest, and the rate of velocity. Mathematically, the dependence of the state of a system on its history translates into a rate-dependent hysteresis. Thus, each time variant friction coefficient in (45) is bounded as follows:

$$\boldsymbol{k}_1 = \min\{\boldsymbol{F}_{C_1}, \boldsymbol{F}_{C_2}\} \leq \boldsymbol{F}_C(t) \leq \max\{\boldsymbol{F}_{C_1}, \boldsymbol{F}_{C_2}\} = \boldsymbol{k}_2 \tag{47}$$

$$\boldsymbol{k}_3 = \inf_{t_0 \leq t < \infty} \{\boldsymbol{F}_v(t)\} \leq \boldsymbol{F}_v(t) \leq \sup_{t_0 \leq t < \infty} \{\boldsymbol{F}_v(t)\} = \boldsymbol{k}_4 \tag{48}$$

Therefore, the time derivative in (39) satisfies

$$\dot{V}(\boldsymbol{s}, t) \leq -W_3(\boldsymbol{s}) = -\left[\boldsymbol{s}^{\mathsf{T}}(\boldsymbol{K}_D + \boldsymbol{k}_2 + \boldsymbol{k}_4)\boldsymbol{s}\right] < 0\,. \tag{49}$$

Finally, since $0 < W_1(\boldsymbol{s}) \leq V(\boldsymbol{s}, t) \leq W_2(\boldsymbol{s})$ and $\dot{V}(\boldsymbol{s}, t) \leq -W_3(\boldsymbol{s}) < 0$, the virtual velocity $\boldsymbol{s}$ tends to $\mathbf{0}$ as $t \to \infty$. Now, from definition (28), $\tilde{\boldsymbol{q}}$ is exponentially bounded and will tend to $\mathbf{0}$ as $t \to \infty$ as well as $\dot{\tilde{\boldsymbol{q}}}$. This is sufficient to demonstrate that system

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\boldsymbol{q} \\ \dot{\boldsymbol{q}}\end{bmatrix} = \begin{bmatrix}\dot{\boldsymbol{q}} \\ \boldsymbol{M}^{-1}(\boldsymbol{q})[\boldsymbol{\tau} - \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \boldsymbol{g}(\boldsymbol{q}) - \boldsymbol{f}(\dot{\boldsymbol{q}}, t)]\end{bmatrix}, \tag{50}$$

is uniformly asymptotically stable.

6. Results

Experimental Test Implementation

The two-degree-of-freedom robot manipulator shown in Figure 3 was used as the physical plant for the experimental tests carried out using the following setup. All the links are actuated by direct-drive motors from Yokogawa Electric Corporation, model DM1A-200G for link 1 and model DM1B-030G for link 2. The motors are configured in torque mode, with a maximum torque of $\pm 200\,\mathrm{N \cdot m}$ for link 1 and $\pm 30\,\mathrm{N \cdot m}$ for link 2. The motors are equipped with relative encoders with a resolution of 1,024,000 and 655,360 pulses per revolution, respectively.

The digital implementation of the proposed controller and update law runs in MATLAB/Simulink using the Simulink Desktop Real-Time tool with a sampling period of 1 ms on a PC equipped with a Core i5 14,600K CPU, 32 gb of RAM, a RTX 3060 GPU, a 512 gb SSD M.2 for storage, and Windows 10 OS. The connection between the PC and the robot was carried out by a data acquisition card (DAQ) from Sensoray model 626 mounted on a PCI port on the motherboard of the PC and the drivers of each link from PARKER COMPUMOTOR, model DMG3-1200A for link 1 and DMG3-1030B for link 2. The DAQ sends a voltage signal representing each torque computed by the control law into analog inputs in the drivers. On the other hand, the drivers send a pulse train that replicates the signal from the encoders inside the motors to digital inputs in the DAQ. It is worth noting that the only custom-built devices in this setup are the links and the base stand of the robot; these were made by Centro de Investigación Científica y de Educación Superior de Ensenada for academic purposes.



Figure 3. Two-degree-of-freedom robotic arm and motor drivers used for the experimental tests.

The desired trajectories used in the experimental implementation tests are defined as follows:

$$\begin{bmatrix} q_{d_1} \\ q_{d_2} \end{bmatrix} = \begin{bmatrix} b_1\left[1-\mathrm{e}^{-d_1 t^3}\right] + c_1\left[1-\mathrm{e}^{-d_1 t^3}\right]\sin(\omega_1 t) + \cos(3\omega_1 t) + \sin(5\omega_1 t) \\ b_2\left[1-\mathrm{e}^{-d_2 t^3}\right] + c_2\left[1-\mathrm{e}^{-d_2 t^3}\right]\sin(\omega_2 t) + \cos(3\omega_2 t) + \sin(5\omega_2 t) \end{bmatrix} \tag{51}$$

where

$$b_1 = \frac{\pi}{3}[\mathrm{rad}], \quad b_2 = \frac{\pi}{3}[\mathrm{rad}], \quad c_1 = \frac{\pi}{12}[\mathrm{rad}], \quad c_2 = \frac{\pi}{16}[\mathrm{rad}]$$

$$d_1 = 0.3\left[\frac{1}{\mathrm{s}^3}\right], \quad d_2 = 0.4\left[\frac{1}{\mathrm{s}^3}\right], \quad \omega_1 = 1.5\left[\frac{\mathrm{rad}}{\mathrm{s}}\right], \quad \omega_2 = 1\left[\frac{\mathrm{rad}}{\mathrm{s}}\right].$$

The choice on desired trajectories has two purposes: the first one is to provide a rather complicated tracking task and still show good performance compared to other controllers; the second one is to satisfy the persistent excitation for the first estimator in (19) by following the insight that three harmonics—i.e., at least three periodic functions with a common factor but different frequencies—are enough to excite all of the natural modes of a 2-DOF mechanical system; indeed, this is the physical meaning of persistent excitation. Of course, this excitation needs to be held for a finite period, but sine functions are periodic and harmonic; thus, experimentally, this condition is satisfied trivially after certain cycles. All the constant parameters in controller design are shown below:

$$\boldsymbol{\Lambda} = \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}, \quad \lambda_f = 130, \quad \boldsymbol{\Gamma}_1 = \begin{bmatrix} 0.23 & 0 & 0 & 0 \\ 0 & 0.86080155 & 0 & 0 \\ 0 & 0 & 0.65305335 & 0 \\ 0 & 0 & 0 & 0.3 \end{bmatrix}.$$

$$\boldsymbol{\kappa} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \boldsymbol{\Gamma}_2 = \begin{bmatrix} 48 & 0 \\ 0 & 18.48 \end{bmatrix}, \quad \boldsymbol{\Gamma}_3 = \begin{bmatrix} 36 & 0 \\ 0 & 6.15 \end{bmatrix}$$

Physical initial conditions were $\boldsymbol{q}(0) = \mathbf{0}$ and $\dot{\boldsymbol{q}}(0) = \mathbf{0}$, and the only prior information of the system is that $l_1 = 0.313\,[\mathrm{m}]$. The angular positions of each joint are shown in Figure 4 and Figure 5, respectively.

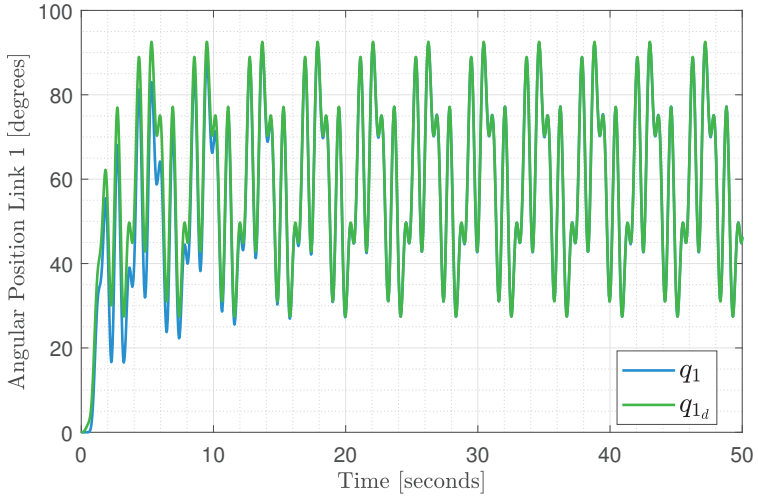


Figure 4. Angular position for joint 1.

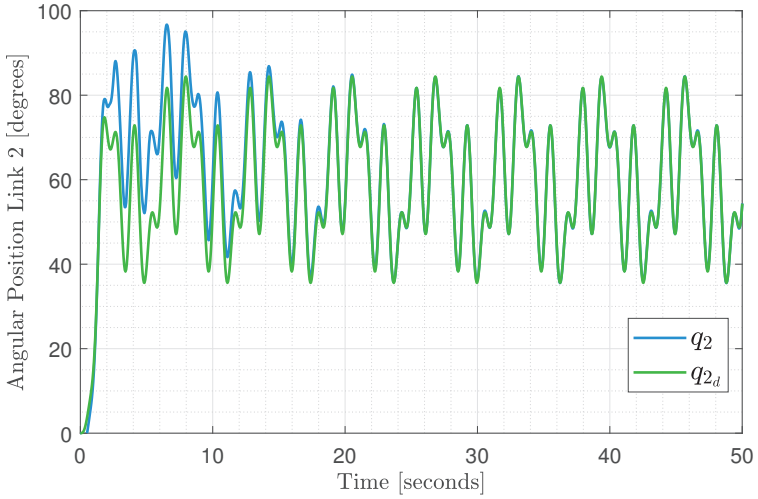


Figure 5. Angular position for joint 2.

The angular position errors of each joint are shown in Figure 6 and Figure 7, respectively.

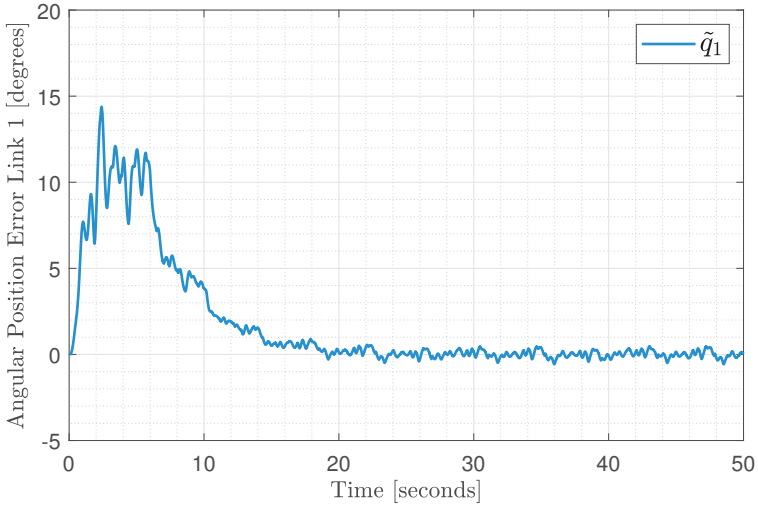


Figure 6. Angular position error for joint 1.

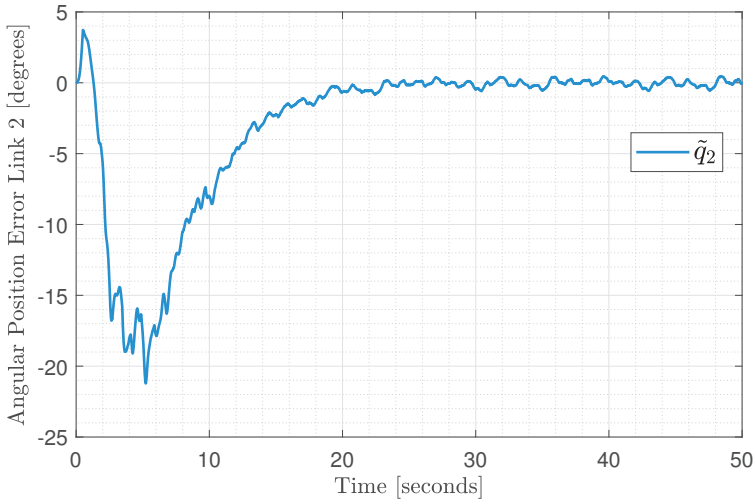


Figure 7. Angular position error for joint 2.

The torques applied at each joint are shown in Figure 8 and Figure 9, respectively.

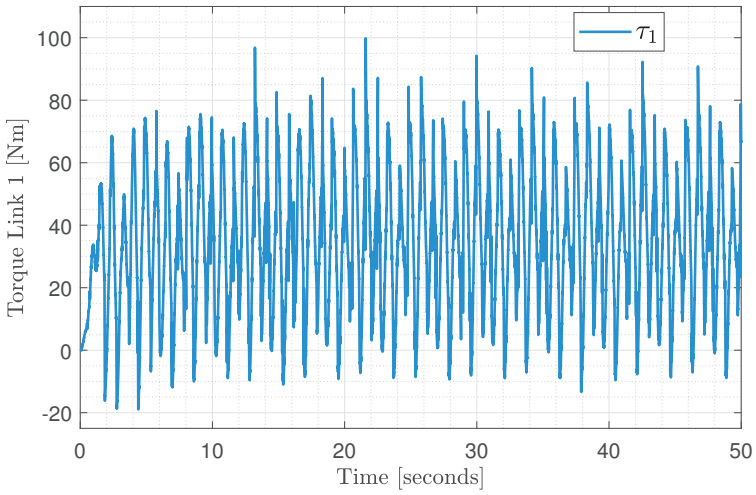


Figure 8. Torque applied at joint 1.

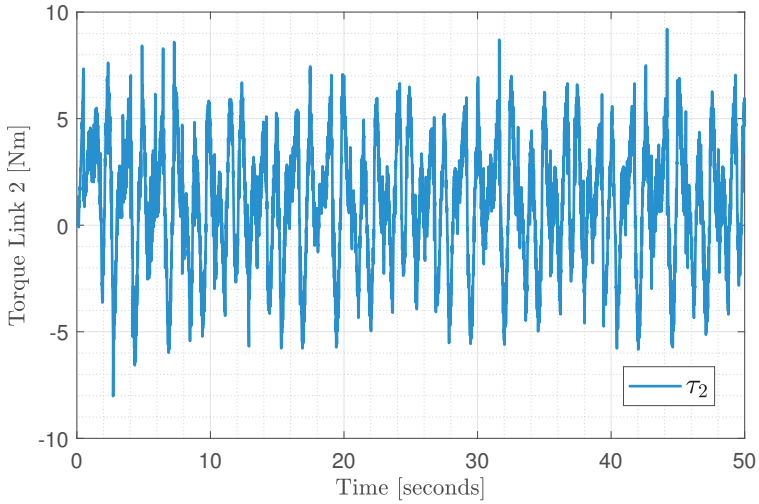


Figure 9. Torque applied at joint 2.

Finally, the nominal vector of physical inertia and geometric parameters, i.e.,

$$\boldsymbol{\theta}^* = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} = \begin{bmatrix} I_1 + m_1 {l_{c_1}}^2 + I_2 + m_2 {l_{c_2}}^2 + m_2 {l_1}^2 \\ m_2 l_{c_2} \\ I_2 + m_2 {l_{c_2}}^2 \\ m_1 l_{c_1} + m_2 l_1 \end{bmatrix}, \tag{52}$$

was estimated. The parameter estimates are shown in Figure 10, Figure 11, Figure 12 and Figure 13, respectively.

As seen in previous figures, all the elements in (52) tend to stay around their nominal constant values, respectively, since the desired trajectories in (51) were selected to satisfy (19), and their derivatives were bounded above by some finite number. Of course, the physical real parameters where known but were supposed to be unknown while designing the controller. On the other hand, friction coefficients were not exactly known a priori; instead, according to way the robot dynamics evolve when tracking the specific desired trajectories, both parts of (7) were determined by two independent estimators with their own update laws. The parameter estimates of Coulomb-like friction and viscous-like friction are shown in Figure 14 and Figure 15, respectively.

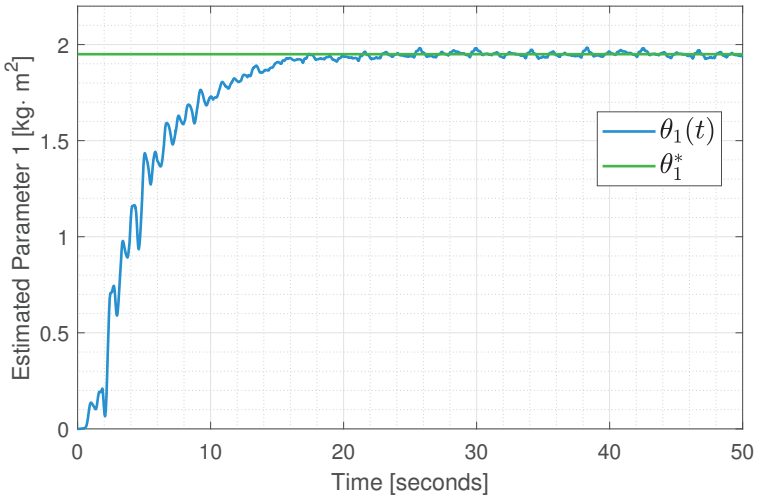


Figure 10. Evolution of the estimated parameter θ_1.

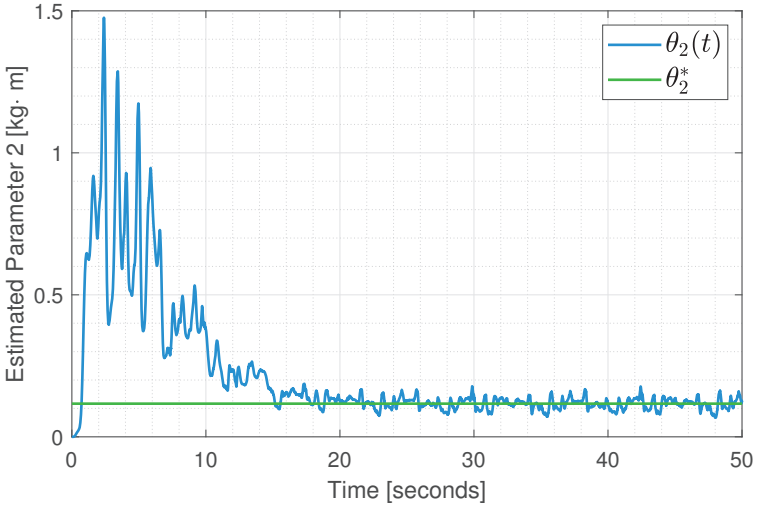


Figure 11. Evolution of the estimated parameter θ_2.

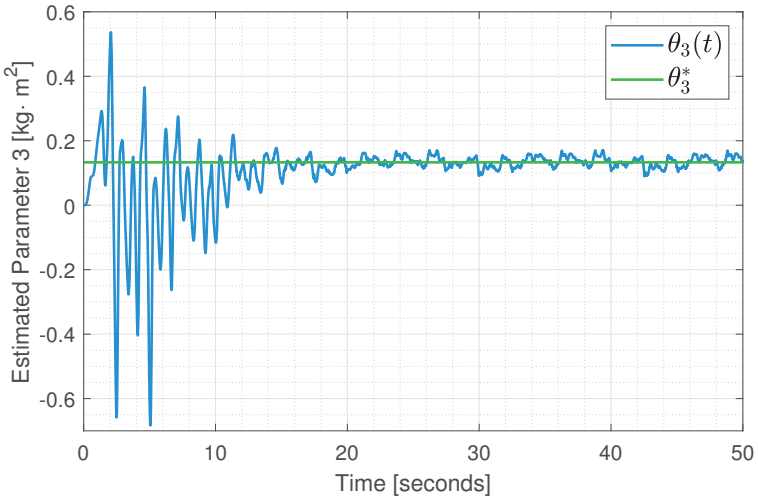


Figure 12. Evolution of the estimated parameter θ_3.

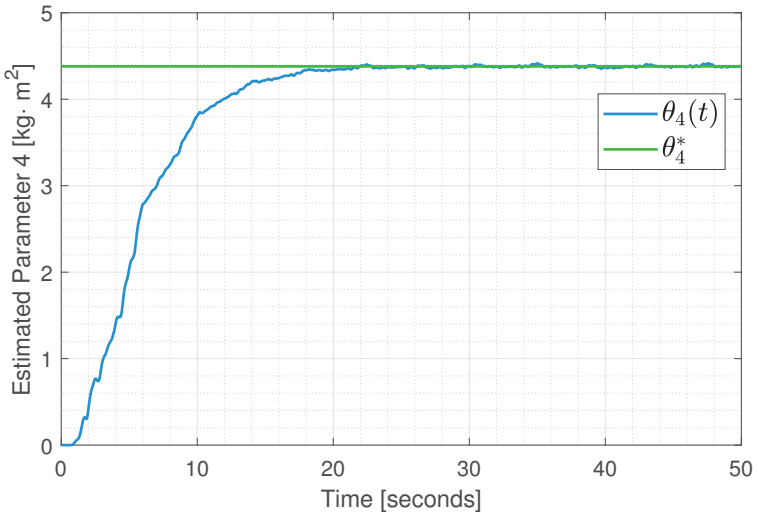


Figure 13. Evolution of the estimated parameter θ_4.

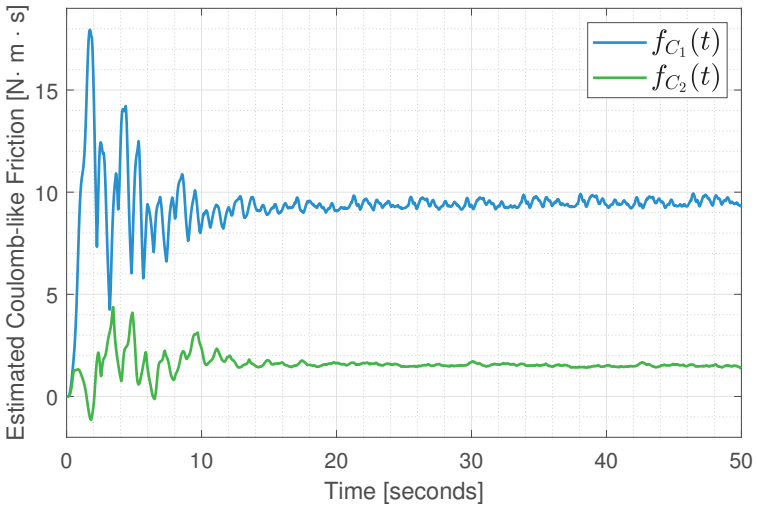


Figure 14. Evolution of the estimated Coulomb-like friction coefficient f_{C_1} and f_{C_2}.

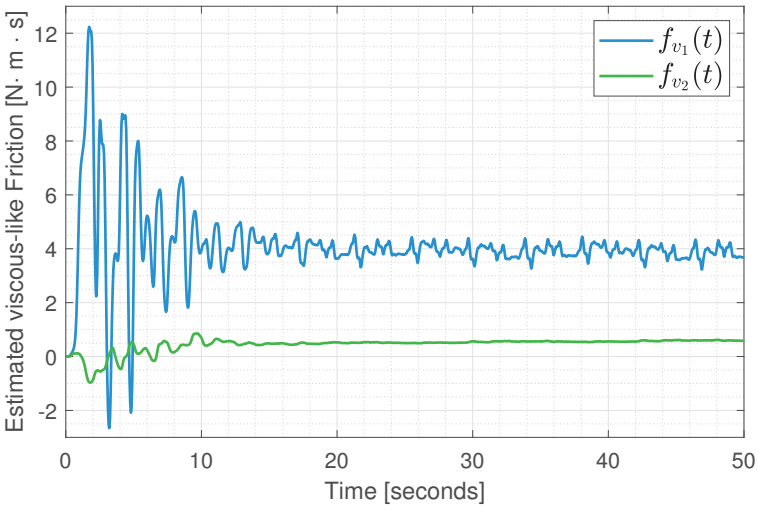


Figure 15. Evolution of the estimated viscous-like friction coefficient f_{v_1} and f_{v_2}.

In both of the estimates, the coefficients associated with link 1 present a larger variation while tracking, this reflects the fact that the link has a significant threshold between the static regime compared to the dynamic regime due to a larger apparent area of contact that increases the adhesion between surfaces, while the second link only presents significant variation when a change in direction occurs at the maximum velocity of the link. A video of the experimental test was uploaded to YouTube: https://youtu.be/9AA2FeD3SRs (accessed on 20 March 2025).

7. Conclusions

In this manuscript, two important (but often underestimated) non-ideal non-linearities of friction phenomena are addressed to overcome the lack of a precise dynamic model of non-conservative forces in robot manipulators while preserving the passive mapping between the adaptation law and the parameter estimation error via additive disturbances theory. We put forward a technical proposal for two independent passive systems to take into account the anisotropy of typical Coulomb friction and the rate-dependent hysteresis generated in transition from one regime of viscous friction. Mathematical support for the decomposition of the conservative and non-conservative part into a novel linear parametrization is presented. This allows us to identify simple bounds for each regressor matrix and to satisfy the persistent excitation condition quite easily. Furthermore, simple quadratic Lyapunov functions for each estimator can be used in the stability analysis. The term of the adaptive computed torque used as control law, which can be seen in the in-disguise PD controller, uses the robust nature of the sliding mode approach to reject small changes in the additive disturbances, leading to a quite-good tracking performance in experimental tests. It is crucial to point out that in control theory applications, with the digital implementation of the controller, the discrete time domain is inherently induced and therefore a state transition between (quasi-)static and dynamic regimes. Finally, the proposed controller keeps the Lyapunov stability analysis simple as both friction phenomena are taken into account independently of each other and treated as additive disturbances that translate into an enhanced damping effect.

Author Contributions: Conceptualization, D.G.-H., J.S.-M. and V.S.; Software, D.G.-H. and I.G.; Validation, D.G.-H., J.S.-M. and V.S.; Formal analysis, D.G.-H.; Writing—original draft, D.G.-H.; Supervision, J.S.-M., V.S. and I.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially supported by Tecnologico Nacional de Mexico TecNM Projects 22483.25-P and LAGU-PYR-2025-23646, the net RICCA of TecNM, and PH.D. Student SECIHTI grant, CONACYT grant number 134534, postdoctoral fellow SECIHTI grant I1200/311/2023.

Data Availability Statement: Data are contained within the article.

Acknowledgments: This work was developed in the framework of the international network "Red internacional de control y cómputo aplicados" supported by Tecnológico Nacional de México (TecNM). This research was supported by "Secretaría de Ciencia, Humanidades, Tecnología e Innovación" and partially supported by TecNM Projects.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Slotine, J.J.E.; Li, W. Composite adaptive control of robot manipulators. *Automatica* **1989**, *25*, 509–519. [CrossRef]
2. Nielsen, H.A.; Madsen, H. Modelling the heat consumption in district heating systems using a grey-box approach. *Energy Build.* **2006**, *38*, 63–71. [CrossRef]
3. Slotine, J.J.E.; Li, W. On the Adaptive Control of Robot Manipulators. *Int. J. Robot. Res.* **1987**, *6*, 49–59. [CrossRef]
4. Hastie, T.; Tibshirani, R. Varying-Coefficient Models. *J. R. Stat. Soc. Ser. B (Methodol.)* **1993**, *55*, 757–779. [CrossRef]
5. Stokes, G. On the Variation of Gravity at the Surface of the Earth. In *Mathematical and Physical Papers*; Cambridge University Press: Cambridge, UK, 2009; pp. 131–171.
6. Marques, F.; Flores, P.; Pimenta Claro, J.C.; Lankarani, H.M. A survey and comparison of several friction force models for dynamic analysis of multibody mechanical systems. *Nonlinear Dyn.* **2016**, *86*, 1407–1443. [CrossRef]
7. Popova, E.; Popov, V.L. The research works of Coulomb and Amontons and generalized laws of friction. *Friction* **2015**, *3*, 183–190. [CrossRef]
8. Armstrong-Hélouvry, B. *Control of Machines with Friction*, 1st ed.; Kluwer Academic Publishers: London, UK, 1991; pp. 24–35.
9. Dahl, P.R. Solid Friction Damping of Mechanical Vibrations. *AIAA J.* **1976**, *14*, 1675–1682. [CrossRef]
10. Olsson, H.; Åström, K.J.; Canudas de Wit, C.; Gäfvert, M.; Lischinsky, P. Friction Models and Friction Compensation. *Eur. J. Control* **1998**, *4*, 176–195. [CrossRef]
11. Armstrong-Hélouvry, B.; Dupont, P.; Canudas de Wit, C. A Survey of Models, Analysis Tools and Compensation Methods for the Control of Machines with Friction. *Automatica* **1994**, *30*, 1083–1138. [CrossRef]
12. Åström, K.J.; Canudas de Wit, C. Revisiting the LuGre Friction Model. *IEEE Control Syst. Mag.* **2008**, *28*, 101–114.
13. Marques, F.; Woliński, Ł.; Wojtyra, M.; Flores, P.; Lankarani, H.M. An Investigation of a Novel LuGre-based Friction Force Model. *Mech. Mach. Theory* **2021**, *166*, 104493. [CrossRef]
14. Niranjan, P.; Karinka, S.; Sairam, K.V.S.S.S.S.; Upadhya, A.; Shetty, S. Friction modeling in servo machines: A review. *Int. J. Dynam. Control* **2018**, *6*, 893–906. [CrossRef]
15. Grami, S.; Okonkwo, P.C. Friction Compensation in Robot Manipulator Using Artificial Neural Network. In *Advances in Automation, Signal Processing, Instrumentation, and Control*; Komanapalli, V.L.N., Sivakumaran, N., Hampannavar, S., Eds.; Springer: Singapore, 2021; pp. 641–650.
16. Sun, Y.; Liang, X.; Wan, Y. Tracking Control of Robot Manipulator with Friction Compensation Using Time-Delay Control and an Adaptive Fuzzy Logic System. *Actuators* **2023**, *12*, 184. [CrossRef]
17. Ali K.; Mehmood A.; Iqbal J. Fault-tolerant scheme for robotic manipulator—Nonlinear robust back-stepping control with friction compensation. *PLoS ONE* **2021**, *16*, e0256491. [CrossRef] [PubMed]
18. Han, J.; Cheng, H.; Shan, X.; Liu, H.; Xiao, J.; Huang, T. A novel multi-pulse friction compensation strategy for hybrid robots. *Mech. Mach. Theory* **2024**, *201* 105726. [CrossRef]
19. Lewis, F.L.; Dawson, D.M.; Abdallah, C.T. *Robot Manipulator Control: Theory and Practice*, 2nd ed.; CRC Press: Boca Raton, FL, USA, 2003; pp. 125–136.
20. Spong, M.W.; Hutchinson, S.; Vidyasagar, M. *Robot Dynamics and Control*, 2nd ed.; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2006; pp. 216–219.
21. Rill, G.; Schaeffer, T.; Schuderer, M. LuGre or not LuGre. *Multibody Syst. Dyn.* **2024**, *60*, 191–218. [CrossRef]
22. Kelly, R.; Santibáñez, V.; Loría, A. *Control of Robot Manipulators in Joint Space*, 1st ed.; Springer: London, UK, 2005; pp. 19–57. 95–101.
23. Barahanov, N.; Ortega, R. Necessary and Sufficient Conditions for Passivity of the LuGre Friction Model. *IEEE Trans. Autom. Control* **2000**, *45*, 830–832. [CrossRef]
24. Canudas-de-Wit, C.; Kelly, R. Passivity Analysis of a Motion Control for Robot Manipulators with Dynamic Friction. *Asian J. Control* **2007**, *9*, 30–36. [CrossRef]

25. Istepanian, R.S.H.; Whidborne, J.F. Finite-Precision Computing for Digital Control Systems: Current Status and Future Paradigms. In *Digital Controller Implementation and Fragility: A Modern Perspective*; Istepanian, R.S.H., Whidborne, J.F., Eds.; Springer: London, UK, 2001; pp. 1–12.
26. Soto I.; Campa R. Two-Loop Control of Redundant Manipulators: Analysis and Experiments on a 3-DOF Planar Arm. *Int. J. Adv. Robot. Syst.* **2013**, *10*, 85. [CrossRef]
27. Sastry, S.S.; Boyd, M. *Adaptive Control: Stability, Convergence and Robustness*, 1st ed.; Prentice-Hall, Inc.: Englewood Cliffs, NJ, USA, 1989; pp. 141.
28. Boyd, S.; Sastry, S.S. Necessary and Sufficient Conditions for Parameter Convergence in Adaptive Control. *Automatica* **1986**, *22*, 629–639. [CrossRef]
29. Gilbert, E.G.; Ha, I.J. An approach to nonlinear feedback control with applications to robotics. In Proceedings of the 22nd IEEE Conference on Decision and Control, San Antonio, TX, USA, 14–16 December 1983; pp. 879–884.
30. Slotine, J.J.; Sastry, S.S. Tracking Control of Non-linear Systems Using Sliding Surfaces with Application to Robot Manipulators. In Proceedings of the 1983 American Control Conference, San Francisco, CA, USA, 22–24 June 1983; pp. 132–135.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

actuators

MDPI

Article

Electronic Control Unit and Digital Twin Based on Raspberry Pi 4 for Testing the Remote Nonlinear Trajectory Tracking of a P3-DX Robot

Cristina Losada-Gutiérrez, Felipe Espinosa *, Carlos Cruz and Biel P. Alvarado

Electronics Department, Engineering School, University of Alcala, 28805 Alcalá de Henares, Spain; cristina.losada@uah.es (C.L.-G.); carlos.cruzt@uah.es (C.C.); biel.alvarado@uah.es (B.P.A.)
* Correspondence: felipe.espinosa@uah.es

Abstract

The properties of Hardware-in-the-Loop (HIL) for the development of controllers, together with electronic emulation of physical process by Digital Twins (DT) significantly enhance the optimization of design and implementation in nonlinear control applications. The study emphasizes the use of the Raspberry Pi (RBP), a low-cost and portable electronic board for two interrelated goals: (a) the Electronic Control Unit (ECU-RBP) implementing a Lyapunov-based Controller (LBC) for nonlinear trajectory tracking of P3DX wheeled robots, and (b) the Digital Twin (DT-RPB) emulating the real robot behavior, which is remotely connected to the control unit. ECU-RBP, DT-RBP and real robot are connected as nodes within the same wireless network, enhancing interaction between the three physical elements. The development process is supported by the Matlab/Simulink environment and the associated packages for the specified electronic board. Following testing of the real robot from the ECU-RBP in an open loop, the model is identified and integrated into the DT-RBP to replicate its functionality. The LBC solution, which has also been validated through simulation, is implemented in the ECU-RBP to examine the closed-loop control according to the HIL strategy. Finally, the study evaluates the effectiveness of the HIL approach by comparing the results obtained from the application of the LBC, as implemented in the ECU-RBP to both the real robot and its DT.

Keywords: hardware in the loop; electronic control unit; digital twin; Raspberry Pi; Lyapunov-based control; nonlinear trajectory tracking; P3-DX robot; wireless network control

1. Introduction

The increasing complexity of modern control systems, particularly in nonlinear and real-time applications, has driven the need for more efficient and reliable development and validation methodologies. Hardware-in-the-Loop (HIL) represents a novel testing methodology used within the field of control system development, enhancing interaction between hardware components and simulated or emulated environments. This approach enables control engineers to validate and optimize their designs prior to deployment in real systems. Consequently, the HIL approach reduces the risks associated with real-world failures, enables the early identification of design flaws, and supports the testing of edge cases that would otherwise be difficult or high cost to reproduce in physical systems. Numerous examples of HIL applications are present in various industrial domains,

including the automotive sector [1–3], academia [4–6], and power electronics [7–9] where practical validation of theoretical concepts is essential for education and prototyping.

In parallel with the evolution of HIL strategies, the concept of the Digital Twin (DT) has gained considerable interest. A DT can be defined depending on the context of application: education and training, service and maintenance, or production planning and control. Basically, it is a virtual representation or model of a physical object or process obtained from real-world data. More precisely, the physical entity and its virtual implementation are interconnected through a synchronized connection allowing bidirectional data exchange to complete the DT concept [10–12].

DT is currently a crucial element of smart manufacturing, cyber–physical systems, and Industry 4.0 architectures. Recent reviews and surveys highlight their expanding role across sectors, underlining their utility for system monitoring, forecasting, and control validation [13–16]. Despite the rapid progress in both HIL and DT technologies, their combined use in nonlinear control applications on low-cost, embedded platforms remain underexplored. Consequently, the paper focuses on the electronic implementation of HIL-DT for robotic applications.

1.1. Related Works

Recent advances in DT technology are increasingly applied in mobile robotics to improve control accuracy, adaptability, and real-time decision making. Studies focus on DT-based modeling and control strategies, highlighting dynamic adaptation, virtual-physical integration, and environmental interaction. For instance, a DT-based parameter compensation method was introduced in [17] to enhance the accuracy of the modeling of mobile robot dynamics under variable conditions. The approach uses real-time physical robot feedback to continuously update the virtual model and improve control performance. Furthermore, a DT-based tracking control for car-like robots is presented in [18], coordinating virtual simulations with real-world execution. The system combines trajectory tracking with environmental perception in dynamic scenarios. DT's role is examined in [19] for the optimization of robot operations, presenting a DT approach to enhance autonomous mobile robots' working environments, with a case study in automated intralogistics systems. In a more recent contribution, ref. [20] focuses on enabling real-time interaction between industrial mobile robots and their DTs during motion. It proposes using multiple intelligent gateways to support digital twin pre-migration and identifies the optimal timing for smooth migration during robot operation.

Regarding hardware resources for testing the mentioned design strategies in control engineering, a historical overview of HIL simulators for real-time control applications is presented in [21], exemplified by platforms such as dSPACE, Typhoon HIL, or Speedgoat. These platforms enhance realistic testing of hardware, specifically electronic controllers, implemented in field programmable gate arrays (FPGA) or digital signal processors (DSP) units. Among the commercially available electronic platforms, the Raspberry Pi 4 (RBP), model B, is widely used in both industrial and academic settings [22–30]. It was released in 2019 and their detailed features (CPU, GPU, memory, ports, connectivity) are described in [31]. The RBP has been widely adopted in recent literature as a cost-effective and versatile embedded platform for implementing a range of control strategies aimed at achieving successful system performance. The validation by simulation (Matlab/Simulink) of a three-phase brushless DC motor implements the algorithm code in a RBP as virtual-platform in-the-loop approach is presented in [32]. In addition, the use of RBP as Electronic Control Unit (ECU) for the Adaptive Model Predictive Control of a vehicle simulated on a HIL test bench is presented in [33]. From a control design perspective, Lyapunov theory is extensively deployed to develop stable nonlinear controllers. Such is the scenario

in robotic trajectory-tracking problems, where control regulation is essential to drive the error in posture towards zero asymptotically. Concerning wheeled robots, existing Lyapunov-based Controller (LBC) solutions can be categorized into (a) alternatives that address trajectory errors described in polar coordinates [34,35] and (b) those using Cartesian coordinates [36,37].

1.2. Research Gaps and Contributions

Although significant advances have been made in the fields of nonlinear control and DT technologies for robotics, several important research gaps remain unaddressed. Firstly, most existing DT and HIL implementations rely on high-performance or licensed hardware platforms, which limit their applicability in educational or resource-constrained settings [4,5,38]. Secondly, while HIL methodologies have proven effective in validating control strategies, their integration with nonlinear controllers and DTs remains limited. Many controls validation setup operate under assumption of ideal communication conditions, such as negligible latency and perfect synchronization between plant and controller [39]. As a result, the transition from simulation to physical implementation can reveal instabilities. Finally, the separation of controller and plant models across different hardware platforms introduces unnecessary complexity and increases implementation costs. Few studies have explored the feasibility of integrating both components into a single low-cost embedded system, which could streamline design and optimize resources [40]. As shown in Table 1, the current literature lacks a comprehensive examination of using platforms like the Raspberry Pi for implementing DT in nonlinear control of mobile robots.

Table 1. Summary of the relevant literature.

Ref.	Focus Area	Research Objective	Sorting
[1–3,29]	Automotive	HIL for ESC systems	No
[4,5]	Education	Robotic HIL	No
[8,9]	Power Systems	HIL DC and Grid	No
[13–15]	Control Modeling and Optimization	DT in control	No
[30–37]	Robotics	Trajectory Tracking	No
This research	Electronic Control	HIL, DT	Yes

In this context, the paper proposes a complete methodology for the design and validation of nonlinear controllers within a HIL-DT framework. The use of a RBP as a unified platform for hosting both the controller and the virtual process offers a low-cost and scalable solution, enabling greater access to advanced control techniques. The goal is to test the tracking of nonlinear path trajectories of the wheeled differential driven robot P3-DX. The electronic control solution is validated through a three-stage process: simulation, DT testing, and real-world experimentation.

The main contribution relates to the application of the cost-effective yet robust electronic platform RBP, serving dual purposed: (a) electronic control unit (ECU-RBP) connected wirelessly to the plant under examination and (b) DT-RBP emulating the robot behaviour as a realistic approach for validating the remote closed-loop control prior to the implementation of the real robot, following the HIL-DT concept. The methodology incorporates practical uncertainties, such as variations in battery levels, mechanical backlash, and wireless communication delays, into the experimental workflow. This enhances the realism and robustness of the validation process. Furthermore, the integration of Simulink support packages for RBP streamlines code generation and deployment, facilitating rapid prototyp-

ing and reducing development time. Together, these contributions advance the state of the art in accessible, realistic, and robust validation of nonlinear robotic control systems.

The remainder of this paper is structured as follows. The resources and methodology are described in Section 2; Section 3 presents the experimental results; and finally, the discussion and conclusions of this study are presented in Sections 4 and 5, respectively.

2. Resources and Methodology

The section includes a description of the tools, the physical elements, and the methodology proposed to achieve the goals. The architecture of the development environment for the nonlinear trajectory tracking of the P3-DX robot includes two key components: the Electronic Control Unit (ECU-RBP) and the Robot Digital Twin (DT-RBP), both implemented on RBP 4 Model B boards, as illustrated in Figure 1.

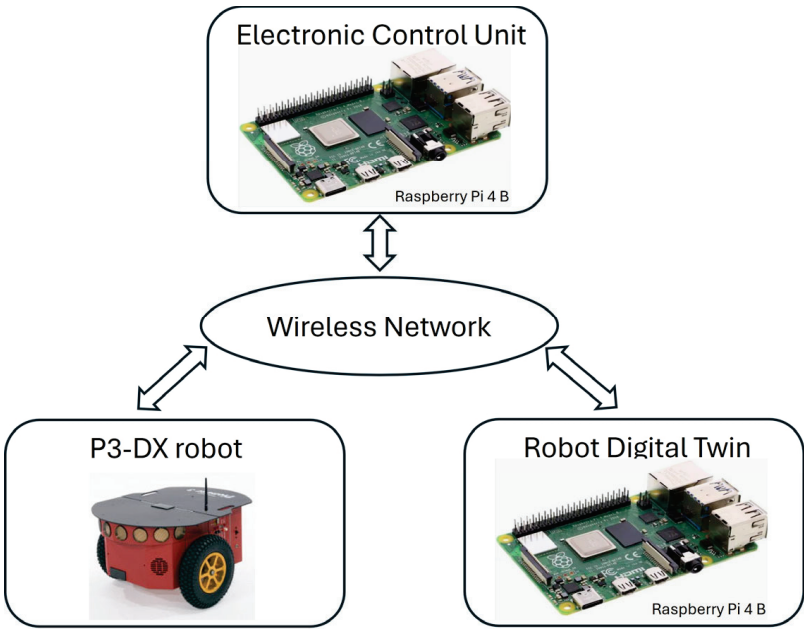


Figure 1. Key elements of the development environment used by authors for the nonlinear trajectory tracking of the P3-DX robot. The Electronic Control Unit (ECU-RBP) and the Robot Digital Twin (DT-RBP) are implemented in RBP 4 (model B).

2.1. Hardware and Software Resources

This research is supported by the interaction between P3-DX robotic units and RBP computing platforms. The core hardware component used in this work is the RBP 4 Model B, a compact and cost-effective electronic board with the following notable specifications: a Broadcom BCM2711 quad-core Cortex-A72 (ARM v8) 64-bit SoC running at 1.8 GHz; a VideoCore VI GPU compatible with OpenGL ES 3.1; 8 GB of LPDDR4-3200 SDRAM; and robust connectivity options including dual-band (2.4 GHz and 5.0 GHz) IEEE 802.11ac wireless and Gigabit Ethernet. Its computational capabilities, extensive library support, active development community, and affordability make it well-suited for electronic control system design. To complete the development environment, the Raspbian operating system was installed along with the specific MATLAB 2024a toolboxes: MATLAB Support Package for RBP Hardware and Simulink Support Package for RBP Hardware.

For the robot under test, the P3-DX was selected [41]. It is a wheeled differential drive robot widely adopted in both education and research contexts. Although the robot is equipped with multiple sensors, this work focuses solely on odometry data to capture its current linear and angular velocities. The robot is integrated with an additional RBP as a

digital platform that enables wireless communication with the control node (ECU-RBP) and provides an electronic interface to the robot's firmware.

For Computer Aided Control Design Tool, Matlab/Simulink was used. Simulink Coder [42] deserves special mention, as it enables the automatic generation of C and C++ code from Simulink models. This capability makes it a widely used tool in embedded system development, especially for applications such as hardware-in-the-loop (HIL) testing.

2.2. Methodology for Obtaining the Robot Digital Twin

Before using any electronic device described previously, it is imperative that a methodology to create the DT is developed. This methodology is outlined in Figure 2 and described as follows:

- Comprehension of system dynamics: Understand the principle of operation of the process to be modelled.
- Electronic interfacing: Setup the electronic interface that allows the excitation and recording of the real process response.
- Excitation design and data acquisition: Design the excitation signals (input vector) and perform the experiments to record the corresponding output vector.
- System response analysis: Analyze the response of the plant to identify region of nonlinear (saturation, dead zone) behaviour and linear characteristics (dynamic and static).
- Parametric modeling: Propose a parametric model for the linear zone and identify the corresponding parameters using the associated tools (e.g., Matlab Identification Toolbox).
- Model validation: Validate the proposed model using a different input test vector from the one used for identification.
- Model refinement or code generation: If the validation results are unsatisfactory, revisit and revise the proposed model (return to the modeling step). If the results are acceptable, generate code for implementation on the target digital platform (e.g., using Simulink Coder).
- DT validation: Evaluate the DT implemented on the electronic platform by comparing the real and virtual process (DT) with new test inputs applied via the same electronic interface.

2.2.1. Robot Modeling and Identification

Once the ECU-RBP is configured to communicate bidirectionally with the P3-DX robot (see Figure 1), a series of open-loop tests are conducted. A bilateral UDP wireless link enables reliable data exchange, supporting model identification and validation. The nonlinear components (dead-zone and saturation) and the linear ones (state-space linear model) of the P3-DX robotic unit are obtained, where u'_k is the input vector including linear ($V'_{i,k}$) and angular ($W'_{i,k}$) velocities, and y_k is the output vector with velocities ($V_{o,k}$, $W_{o,k}$) obtained from odometry following the study presented in [43]. These elements are used to complete the Simulink model depicted in Figure 3. The identified model parameters, determined using a sampling time of 25 ms, are detailed in Appendix A.1. Additionally, remote access to the robot from the ECU-RBP via the wireless network introduces a delay of q samples, which directly influences the dimensionality of the state vector in the discrete-time linear model.

Simulink Coder [42] enables the deployment of the Simulink-based robot model, depicted in Figure 4, onto the DT-RBP platform. This allows the virtual robot to be controlled, running on the electronic board, ready to work as a DT of the real process.

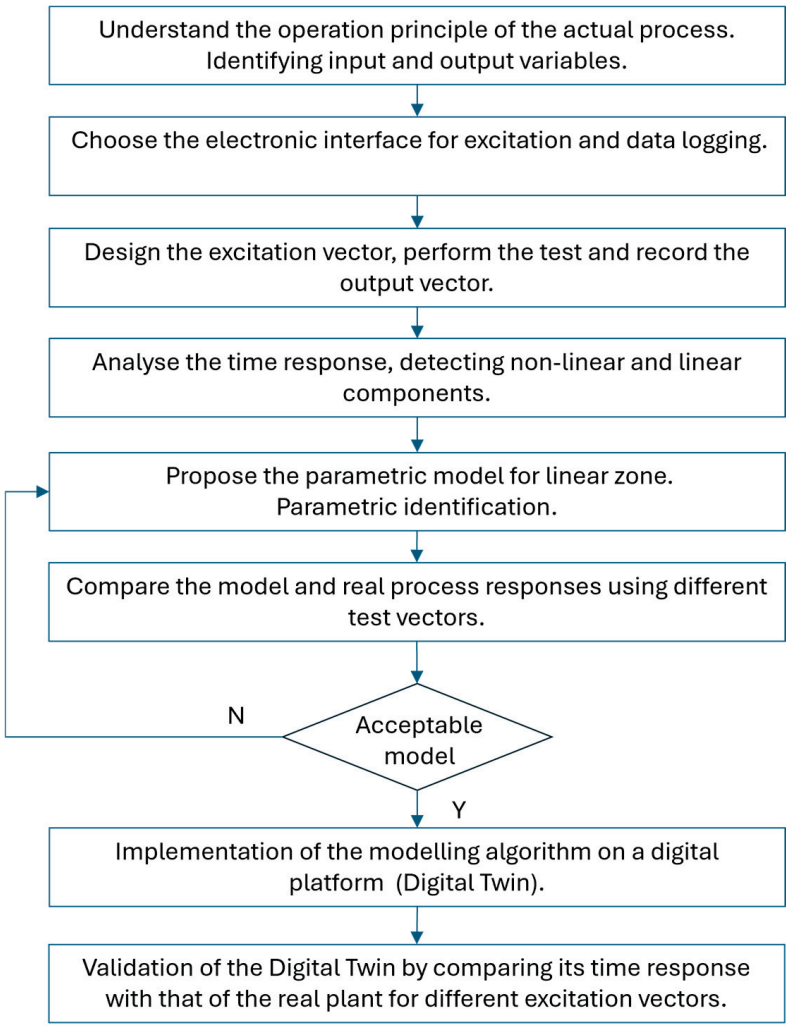


Figure 2. Development of methodology for DTs, from testing the real process to validate the virtual one.

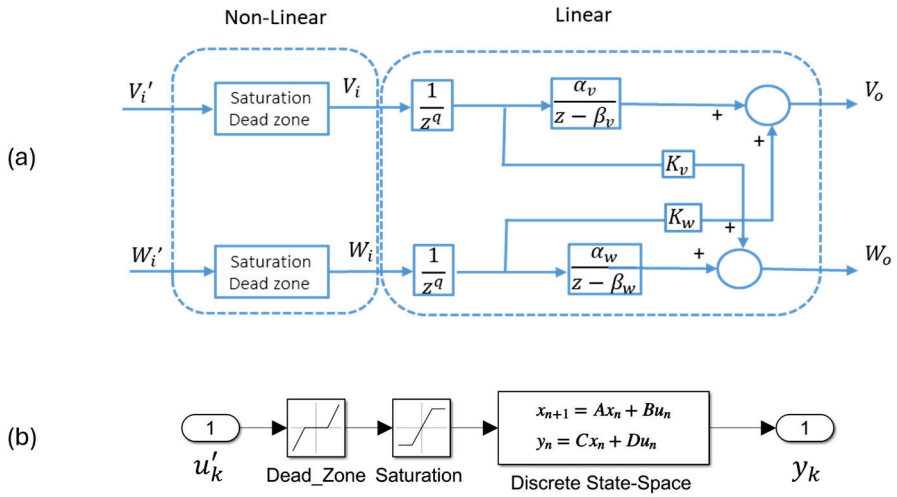


Figure 3. Linear and nonlinear components of the identified model of the P3-DX robot (**a**). Variable relationship and Simulink model (**b**).

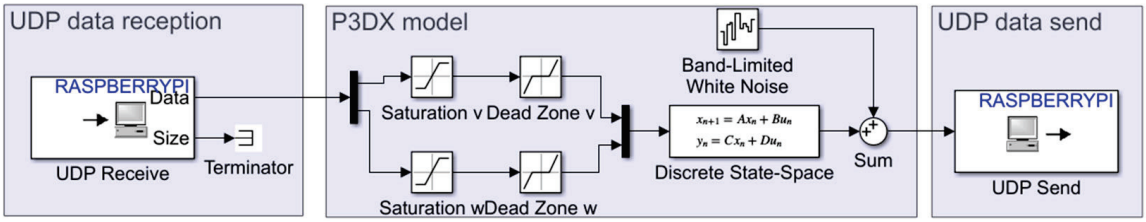


Figure 4. Simulink project to be downloaded on the DT-RBP working as a DT of the identified robotic unit.

2.2.2. Nonlinear Trajectory Tracking: Lyapunov-Based Controller

For the nonlinear trajectory tracking, the strategy developed by Alcala et al. in the Elektra project [36] is applied for the control of an autonomous vehicle. The design process follows a structured sequence of steps, summarized as follows.

(a) Consider the kinematics model of a unicycle mobile (Equation (1))

$$\begin{aligned} \dot{\theta} &= w, \\ \dot{x} &= v{\cdot}cos(\theta), \\ \dot{y} &= v{\cdot}sin(\theta), \end{aligned} \tag{1}$$

being v the linear velocity, w the angular velocity, $x,\ y\ and\ \theta$ the robot pose in Cartesian space, and the upper dot means the time derivative.

(b) Define the pose error e (Cartesian space): $x_{e,}$, $y_e\ and\ \theta_e$, respect to the desired values.
(c) Obtain the time derivative of the pose error $\dot{e}$.
(d) Propose the Lyapunov Function $V(e)$, in this case:

$$V(e) = \frac{1}{2}\, x_e^2 + \frac{1}{2}\, y_e^2 + \frac{1}{2}\, \theta_e^2 \tag{2}$$

(e) Evaluate the time derivative of the Lyapunov Function $\dot{V}(e)$, where the control vector (v, w) and the desired velocities (v_d , w_d) according to the trajectory reference are involved,

$$\dot{V}(e) = K_2 x_e(v_d cos\theta_e - v) + \theta_e(w_d - w) + K_2 y_e v_d sin\theta_e \tag{3}$$

(f) Propose a control law that guarantees stability of the control system, that means $\dot{V}(e) < 0$. In this case, the control law is:

$$\begin{bmatrix} v \\ \\ w \end{bmatrix} = \begin{bmatrix} k_1\, x_e + v_d\, cos\theta_e \\ \\ w_d + k_2\, v_d\, y_e + k_3\, \theta_e \end{bmatrix} \tag{4}$$

being $k_{1,}$, $k_2\ and\ k_3$ positive control gains.

(g) Adjust the gains involved in the control law according to the desired trajectory tracking (see Appendix A.2).

At this stage of the design process, the dynamics related to the velocity servosystem (as described in [43]) is incorporated to validate the complete Simulink model, illustrated in Figure 5. The Kalman Filter provides the estimated state $\hat{x}$ and the filtered output $\hat{y}$ to close the servosystem loop. All the control gains involved in the global control solution depicted in Figure 5 are detailed in Appendix A.2.

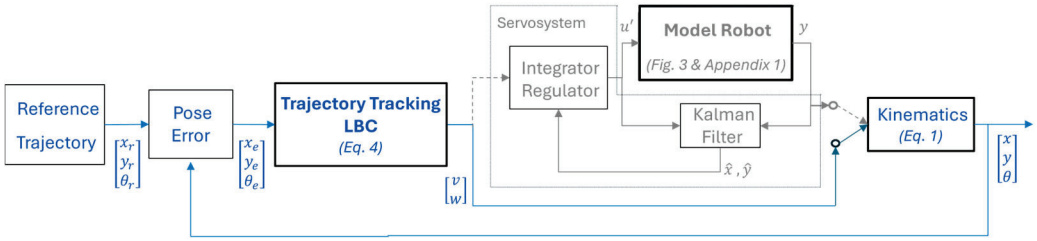


Figure 5. Block diagram to validate by simulation the global control solution for nonlinear trajectory tracking of the robotic unit.

Finally, following the same approach used for the implementation of DT, all control components shown in Figure 5—excluding the Robot Model—are deployed to the ECU-RBP using the Simulink Coder tool [36], allowing the acquisition of experimental results.

3. Experimental Results

In previous sections, the following aspects were presented: (a) theoretical ones supporting the algorithms running in the ECU-RBP and the DT-RBP, and (b) the hardware and software resources required to complete the test.

The P3-DX is a differential drive robot developed by MobileRobots Inc. It is made of an aluminium body (44 × 38 × 22 cm) with 16.5 cm diameter drive wheels. The two DC motors use 38.3:1 gear ratio and contain 500-tick encoders. The differential drive platform is non-holonomic and can rotate in place moving both wheels, or it can swing around a stationery wheel in a circle of 32 cm radius. A rear caster is included for balancing the robot. On flat floor, the P3-DX can move at speeds of 1.6 m/s. At slower speeds it can carry payloads up to 23 kg. In addition to motor encoders, the P3DX base includes eight ultrasonic transducer (range-finding sonar) sensors arranged to provide 180-degree forward coverage [41].

As a case study, the tracking of two different trajectories for the P3-DX is proposed.

The closed-loop solution comprising the ECU-RBP and the DT-RBP (i.e., virtual robot), is initially evaluated following the configuration shown in Figure 1 (right side). Subsequently, as indicated in Figure 1 (left side), the same electronic control unit (ECU-RBP) is applied to the physical robotic platform. In both configurations, the ECU-RBP and DT-RBP are connected to a host laptop, enabling testing and result evaluation through the Simulink graphical interface. Nevertheless, each RBP platform can execute its respective functions, control (ECU-RBP) and modeling (DT-RBP), independently in stand-alone mode. The experimental setup corresponding to this configuration is illustrated in Figure 6 and the real work scenario in the Engineering School at University of Alcala is shown in Figure 7.

In the first experiment, an eight-form trajectory reference is defined, starting from the coordinates (0, 0, 0) to the final target (0, 2, π). Figure 8a shows the results obtained from the DT (red line) and the robot (yellow line) with respect to the reference (blue line). Linear and angular velocities registered from the real (Robot) and virtual (DT) mobiles are depicted in Figure 8b. The highest difference is accumulated at the end of the trajectory.

To gain deeper insight into the advantages and limitations of the proposed experiment, the tracking errors shown in Figure 9 are analyzed. These include the position $(e_x,\ e_y)$ and orientation (e_θ) errors of the robot (red line) and the DT (blue line) along the trajectory. Compared to Figure 8, an increase in pose errors is observed during transitions in reference signals. However, the overall errors remain within acceptable margins, under 5 cm in X and Y, and under 0.05 rads in θ.



Figure 6. Set-up for obtaining experimental results working with the same electronic controller ECU-RBP applied to the virtual process (ECU-DT) or the real one (P3-DX robot).



Figure 7. Real work scenario (laboratories corridor) for validating nonlinear trajectory tests with the P3-DX robotic platform remotely controlled by the ECU-RBP.

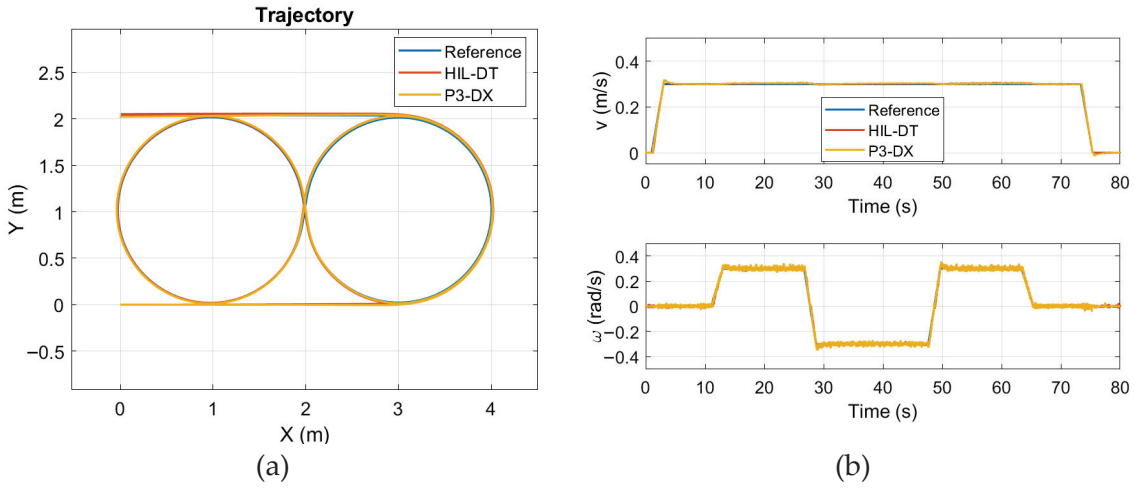


Figure 8. (**a**). Comparison of reference and followed trajectories, by DT-RBP (red line) and P3-DX (yellow line) (**b**). Linear and angular velocities obtained from the real robot and its DT.

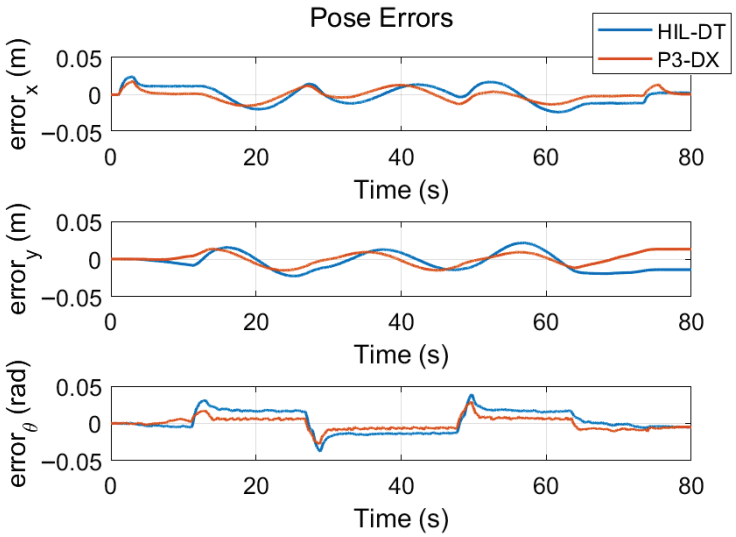


Figure 9. Comparison of pose (position and orientation) errors once the control signals are applied from the ECU-RBP to the DT-RBP (blue line) and to the P3-DX (red line). Eight-form trajectory experiment.

A longer trajectory, including a simulated disturbance at the plant input has also been tested. As shown in Figure 10, the trajectory starts at the door of the laboratory L03 (red dot) with a initial pose $(0, 0, \pi/2)$ and ends at the door of the OL5 (green dot) with a pose $(3.80, 6.80, \pi/2)$, following the path represented in blue color. The measures of the corridor are also included.

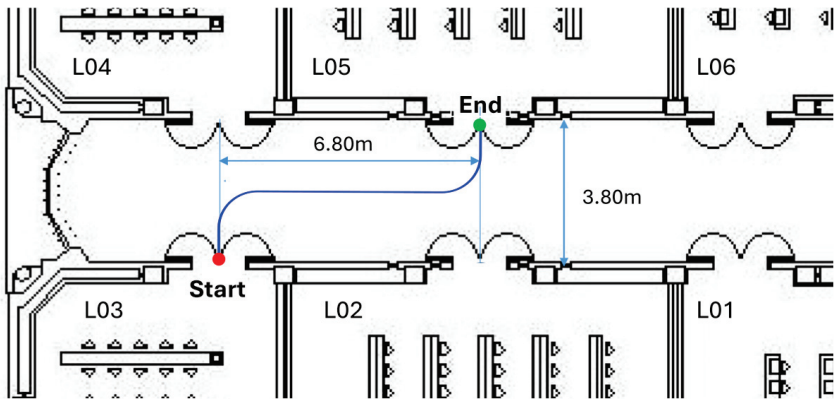


Figure 10. Layout of the laboratories corridor where the experiments are performed, including the robot's reference path (blue line) and the start (red) and end (green) points of the path.

The simulated disturbance, shown in Figure 11, is added to the angular velocity plant input to evaluate the behaviour of the control system.

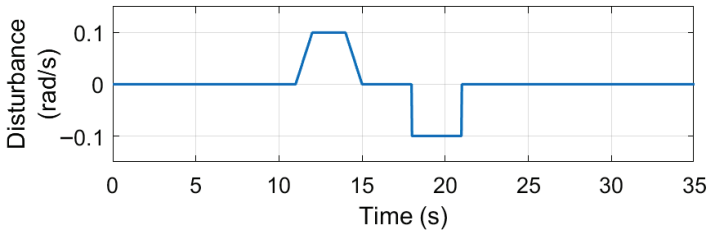


Figure 11. Disturbance added to the angular velocity plant input.

Figure 12a shows the results obtained from the DT (red line) and the robot (yellow line) with respect to the reference (blue line), and Figure 12b presents the linear and angular

velocities registered. The effects of the disturbance can be clearly observed from second 11 to 21, with changes in the output angular velocities and the trajectory followed, which are quickly corrected.

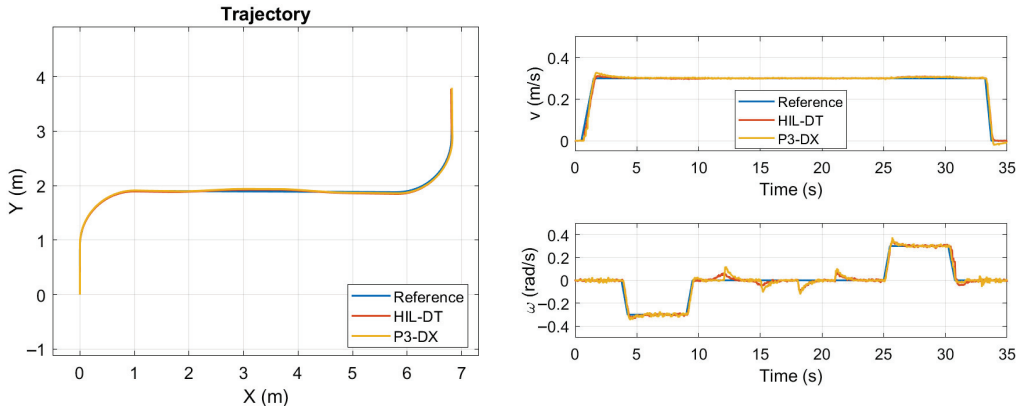


Figure 12. (**a**). Comparison of reference and followed trajectories, by DT-RBP (red line) and P3-DX (yellow line) (**b**). Output linear and angular velocities obtained from the real robot and its DT.

Finally, the tracking errors are also shown in Figure 13. As in the previous experiment, the values remain acceptable.

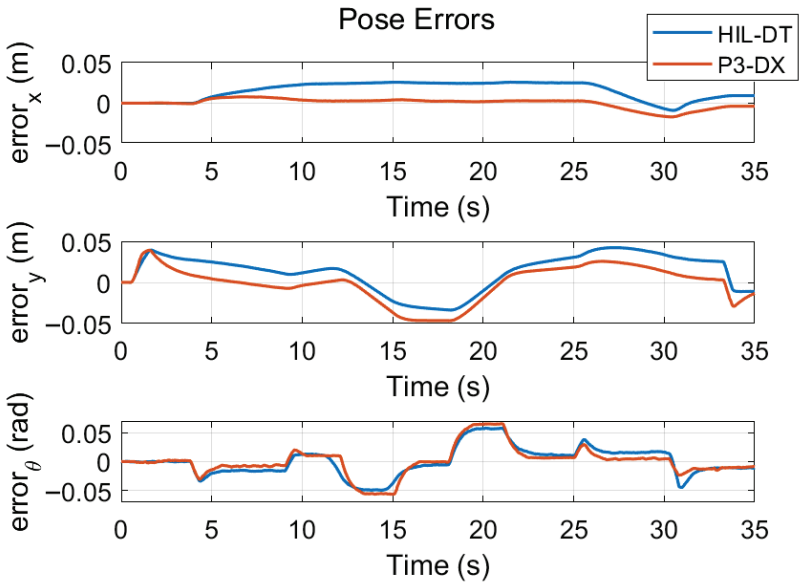


Figure 13. Comparison of pose (position and orientation) errors once the control signals are applied from the ECU-RBP to the DT-RBP (blue line) and to the P3-DX (red line). S-form trajectory experiment.

4. Discussion

The experimental results show the proposed low-cost HIL architecture effectively validating nonlinear control strategies for mobile robotics. It uses a Lyapunov-based Controller on an ECU-RBP integrated with a DT-RBP and the physical P3-DX robot via a wireless network. The control system consistently performs across DT and physical platforms, as evidenced in Figures 8a and 12a. Both platforms accurately track the trajectories even with velocity disturbance. Velocity profiles in Figures 8b and 12b confirm the DT's dynamic behavior mirrors the physical robot, validating the DT model's fidelity and the controller's adaptability.

The implemented control architecture integrates both controller and plant (real and DT) through a wireless network, addressing not only theoretical challenges but also practical considerations that affect overall system performance. Firstly, although the plant model derived through a system identification process (as shown in Figure 3) includes inherent delays due to the wireless communication channel, these delays are not embedded in the DT-RBP model. This omission is intentional, as delays are already present in the current communication between the Electronic Control Unit (ECU-RBP) and the Robot-Based Platform (DT-RBP) and duplicating them would result in an unrealistic control response within the HIL context. Secondly, the wireless communication channel used during the experimental validation exhibited non-deterministic behavior. The observed variability in transmission delay led to fluctuations in the system response, highlighting the importance of accounting for such non-idealities when transitioning from simulation to physical implementation. Thirdly, minor adjustments of control gains were required for approaching control errors, from simulation to emulation with DT, and finally to implementation on the physical robot. As it is known, the behaviour of the process under control is slightly time variant due to changes in physical factors such as battery charge level, wheel pressure, or mechanical backlash, among others, as well as environmental conditions like floor surface and wireless signal quality. These practical aspects contribute to the slight discrepancies observed between the responses shown in Figures 8 and 9. Despite these differences, the overall performance remains consistent and validates the proposed control strategy.

5. Conclusions

This study demonstrates the feasibility and effectiveness of using Hardware-in-the-Loop combined with DT (HIL-DT) to design and validate nonlinear control solutions for mobile robots with portable and low-cost hardware platforms. The proposed methodology shows that high-cost instrumentation is not required for developing and testing advanced control strategies. By leveraging widely available tools and open-source platforms, such as the Raspberry Pi (RBP) 4, both the controller and the process model (as a digital twin) can be deployed on the same embedded system.

The integration of Simulink support packages for RBP greatly simplifies the code generation process and enhances the portability of control solutions from simulation to embedded deployment. This feature enhances fast prototyping and reduces the complexity of the design workflow. The same electronic board allows the designer to implement the process model working as a virtual process. This way, the electronic resources are optimized and the methodology of design simplified. Once the global control solution is validated by simulation, an alternative to design onboard control and model code is to take advantage of the add-ons offered by some Computer-Aided Control System Design tools, that is the case of the support packages for RBP hardware available on Simulink. The use of a Lyapunov-based controller for nonlinear trajectory tracking was successfully implemented and tested on a P3-DX mobile robot. The control solution demonstrated robust performance despite system uncertainties and variable communication delays.

Overall, the presented case study validates the proposed methodology and highlights the potential of DT applications using RBP platforms in both academic and industrial contexts. Considering that the real process and the RBP-DT are nodes of the same wireless network, they could bidirectionally and periodically interchange data flow allowing the parameters self-adaptation of the Digital Twin model. This way a new research line is open for future work. Besides, scalability to multi-agent systems, real-time diagnostics, or adaptive control strategies based on DT may be explored as future research lines.

Author Contributions: Conceptualization, F.E.; methodology, C.L.-G.; formal analysis, F.E. and B.P.A.; experiment C.L.-G. and C.C.; writing; review, C.C. and B.P.A.; and supervision F.E. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the University of Alcala through the projects: Development of digital twin for test bench in control engineering formation (Ref UAH/EV-1545, UAH/EV1633) and Integration of Demand Response in Prosumer Energy Management (Ref PIUAH24/IA-020).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

DT	Digital Twin
HIL	Hardware-in-the-Loop
RBP	Raspberry Pi
ECU	Electronic Control Unit
LBC	Lyapunov-based Controller
FPGA	Field Programmable Gate Array
DSP	Digital Signal Processor
CPU	Central Process Unit
GPU	Graphic Processing Unit
DC	Direct Current motor
UDP	User Datagram Protocol

Appendix A.

Appendix A.1. Identification Parameters

In Section 2.2.1 we focused on the interest of identifying the robot model from data registered from a real unit excited by velocity references running in the ECU-RBP, both wirelessly linked. The odometry of the robot allows the corresponding velocities according to the process dynamics. The values of the parameters appearing in Figure 3, related to the nonlinear and linear components of the P3-DX, are given Table A1.

Table A1. Parameters related to the identified model of the P3-DX.

Parameter	Value
Ts	25 ms
Delay q	3 samples
Input vector size	2
Output vector size	2
State vector size	8
Linear velocity saturation	-0.7701 m/s, 0.7487 m/s
Angular velocity saturation	-1.7700 rad/s, 1.7550 rad/s
Linear velocity dead-zone	-0.0151 m/s, 0.0149 m/s
Angular velocity dead-zone	-0.0300 rad/s, 0.0750 rad/s
α_v	0.3984
α_w	0.2287
β_v	0.6014
β_w	0.7706
K_v	-57×10^{-6}
K_w	-36×10^{-6}

Taking into account these parameters, the discrete-time linear model in the state space is represented by the following characteristic matrices:

$$\mathrm{A} = \begin{pmatrix} \beta_v & \alpha_v & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \beta_w & \alpha_w & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \mathrm{B} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix},$$

$$\mathrm{C} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & K_w & 0 & 0 \\ 0 & K_v & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}, \mathrm{D} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}.$$

Appendix A.2. Control Gains of the Global Solution

In Section 2.2.2 we deal with the global control solution according to the scheme shown in Figure 5. The control gains obtained from the design process are included in Table A2.

Table A2. Control gains resulting from the global control design.

Parameter	Value
Servo: Gain regulator Kr	$\begin{bmatrix} 0.75 & 0.31 & 0.34 & 0.14 & -0.43 & -0.09 & -0.07 & -0.09 \\ -0.25 & -0.09 & -0.08 & -0.12 & 2.98 & 0.65 & 0.64 & 0.44 \end{bmatrix}$
Servo: Integral Gain Ki	$\begin{bmatrix} -0.29 & 0.12 \\ 0.10 & -0.76 \end{bmatrix}$
LBC gains	$K_1 = 1.21$, $K_2 = 1.30$, $K_3 = 1.53$

References

1. Tumasov, A.V.; Vashurin, A.S.; Trusov, Y.; Evgeny, T. The Application of Hardware-in-the-Loop (HIL) Simulation for Evaluation of Active Safety of Vehicles Equipped with Electronic Stability Control Systems. *Procedia Comput. Sci.* **2019**, *150*, 309–315. [CrossRef]
2. Lin, J.S.; Wang, Y.C.; Lin, C.L.; Ma, L.S. Adaptive Cruise Control System Combined with AIoT. In Proceedings of the International Conference on Consumer Electronics—Taiwan (ICCE-Taiwan), Taichung, Taiwan, 9–11 July 2024. [CrossRef]
3. Suh, M.W.; Chung, J.H.; Seok, C.S.; Kim, Y.J. Hardware-in-the-loop simulation for ABS based on PC. *Int. J. Veh. Des.* **2004**, *24*, 157–170. [CrossRef]
4. Temeltas, H.; Gokasan, M.; Bogosyan, S. Hardware in the Loop Robot Simulators for On-site and Remote Education in Robotics. *Int. J. Eng. Educ.* **2006**, *22*, 815–828.
5. Grega, W. Hardware-in-the-loop simulation and its application in control education. In Proceedings of the FIE'99 Frontiers in Education. 29th Annual Frontiers in Education Conference. Designing the Future of Science and Engineering Education, San Juan, PR, USA, 10–13 November 1999; ISBN 0-7803-5643-8.
6. Ahmed, A.H.; Alsharif, A.O.; Aburas, O.E. Low Cost Hardware-in-the-Loop Control Systems Using Microcontrollers. *J. Alasmarya Univ. Basic Appl. Sci.* **2021**, *6*, 587–606. [CrossRef]
7. Zheng, J.L.; Zeng, Y.B.; Xu, H.; Liu, W.C.; Mou, D.; Zhao, Z.M. An Event-Based Synchronization Framework for Controller Hardware-in-the-loop Simulation of Electric Railway Power Electronics Systems. *arXiv* **2023**, arXiv:2311.07036.
8. Sanz, M.; Santamargarita, D.; Huerta, F.; Ochoa, D.; Lázaro, A.; Barrado, A. Reduced-Order Model of Power Converters to Optimize Power Hardware-In-the-Loop Technology in Dc-Distributed Systems. In Proceedings of the 2020 IEEE Applied Power Electronics Conference and Exposition (APEC), New Orleans, LA, USA, 15–19 March 2020. [CrossRef]
9. Flack, C.; Ucer, E.; Smith, C.P.; Kisacikoglu, M. Controller Hardware-in-the-Loop Testing of Decentralized EV-Grid Integration. In Proceedings of the 2022 IEEE Power & Energy Society General Meeting (PESGM), Denver, CO, USA, 17–21 July 2022. [CrossRef]

10. Wagg, D.; Worden, K.; Barthorpe, R.; Gardner, P. Digital Twins: State-of-The-Art Future Directions for Modelling and Simulation in Engineering Dynamics Applications. *ASCE-ASME J. Risk Uncertain. Eng. Syst. Part B Mech. Eng.* **2020**, *6*, 030901. [CrossRef]
11. Segovia, M.; Garcia-Alfaro, J. Design, Modeling and Implementation of Digital Twins. *Sensors* **2022**, *22*, 5396. [CrossRef]
12. Magalhães-Lima, L.G.; Soares-Ferreira, J.M.; Nascimento-Coelho, D.; Regis De Melo, R.; Souza Da Silva, W. Development and Validation of a Digital Twin for a 5-Revolute Joint Robotic Manipulator. In Proceedings of the 2025 Brazilian Conference on Robotics (CROS), Belo Horizonte, Brazil, 28–30 April 2025.
13. Birk, W.; Hostettler, R.; Razi, M.; Atta, K.; Tammia, R. Automatic generation and updating of process industrial digital twins for estimation and control—A review. *Front. Control. Eng.* **2022**, *3*, 954858. [CrossRef]
14. Jørgensen, B.N.; Howard, D.A. Digital Twins: Benefits, Applications and Development Process. In Proceedings of the EPIA Conference on Artificial Intelligence, Faial Island, Portugal, 5–8 September 2023. [CrossRef]
15. Thelen, A.; Zhang, X.; Fink, O.; Lu, Y.; Ghosh, S.; Youn, B.D.; Todd, M.D.; Mahadevan, S.; Hu, C.; Hu, Z. A comprehensive review of digital twin—Part 1: Modeling and twinning enabling technologies. *Struct. Multidiscip. Optim.* **2022**, *65*, 354. [CrossRef]
16. Thelen, A.; Zhang, X.; Fink, O.; Lu, Y.; Ghosh, S.; Youn, B.D.; Todd, M.D.; Mahadevan, S.; Hu, C.; Hu, Z. A comprehensive review of digital twin—Part 2: Roles of uncertainty quantification and optimization, a battery digital twin, and perspectives. *Struct. Multidiscip. Optim.* **2023**, *66*, 1. [CrossRef]
17. Li, R.; Shang, X.; Wang, Y.; Liu, C.; Song, L.; Zhang, Y.; Gu, L.; Zhang, X. Research on Parameter Compensation Method and Control Strategy of Mobile Robot Dynamics Model Based on Digital Twin. *Sensors* **2024**, *24*, 8101. [CrossRef]
18. Zhao, L.; Nie, Z.; Xia, Y.; Li, H. Virtual–Physical Tracking Control for a Car-Like Mobile Robot Based on Digital Twin Technology. *IEEE Trans. Ind. Electron.* **2024**, *71*, 16348–16356. [CrossRef]
19. Stączek, P.; Pizoń, J.; Danilczuk, W.; Gola, A. A digital twin approach for the improvement of an autonomous mobile robots (AMR's) operating environment—A case study. *Sensors* **2021**, *21*, 7830. [CrossRef]
20. Wang, Y.; Zhao, X. A digital twin dynamic migration method for industrial mobile robots. *Robot. Comput.-Integr. Manuf.* **2025**, *92*, 102864. [CrossRef]
21. Mihalic, F.; Truntic, M.; Hren, A. Hardware-in-the-Loop Simulations: A Historical Overview of Engineering Challenges. *Electronics* **2022**, *11*, 2462. [CrossRef]
22. Arif, S.; Khan, A.; Rehman, S. A Lightweight Received Signal Strength Indicator Estimation Model for Low-Power Internet of Things Devices in Constrained Indoor Networks. *Appl. Sci.* **2025**, *15*, 3535. [CrossRef]
23. Bruno, S.; Trifan, I.D.; Vita, L.; Loprencipe, G. Development of Low-Cost Monitoring and Assessment System for Cycle Paths Based on Raspberry Pi Technology. *Infrastructures* **2025**, *10*, 50. [CrossRef]
24. Bueno, G.; Sanchez-Vargas, L.; Diaz-Maroto, A.; Ruiz-Santaquiteria, J.; Blanco, M.; Salido, J.; Cristobal, G. Real-Time Edge Computing vs. GPU-Accelerated Pipelines for Low-Cost Microscopy Applications. *Electronics* **2025**, *14*, 930. [CrossRef]
25. Nichols, K.M.; Roembke, R.A.; Adamczyk, P.G. Real-Time Motor Control Using a Raspberry Pi, ROS, and CANopen over EtherCAT, with Application to a Semi-Active Prosthetic Ankle. *Actuators* **2025**, *14*, 84. [CrossRef]
26. Alvarez-Oviedo, A.; Mamani-Villanueva, J.F.; Echaiz-Espinoza, G.A.; Mauricio-Villanueva, J.M.; Ortiz-Salazar, A.; Llanos-Villarreal, E.R. Design of a System for Driver Drowsiness Detection and Seat Belt Monitoring Using Raspberry Pi 4 and Arduino Nano. *Designs* **2025**, *9*, 11. [CrossRef]
27. Dimitrov, K.; Chivarov, N.; Chivarov, S.; Paunova-Krasteva, T.; Filipov, E.; Daskalova, A. Concept of a Cyber–Physical System for Control of a Self-Cleaning Aquaponic Unit. *AgriEngineering* **2024**, *6*, 3843–3874. [CrossRef]
28. Fazylova, A.; Tultayev, B.; Iliev, T.; Stoyanov, I.; Kabasheva, M.; Kosunalp, S. Experimental Study of an Industrial Data Transmission Network in the Automatic Control System of a Wind Turbine. *Machines* **2024**, *12*, 746. [CrossRef]
29. Turlykozhayeva, D.; Temesheva, S.; Ussipov, N.; Bolysbay, A.; Akhmetali, A.; Akhtanov, S.; Tang, X. Experimental Performance Comparison of Proactive Routing Protocols in Wireless Mesh Network Using Raspberry Pi 4. *Telecom* **2024**, *5*, 1008–1020. [CrossRef]
30. Drăgoi, M.V.; Nisipeanu, I.; Frimu, A.; Tălîngă, A.M.; Hadăr, A.; Dobrescu, T.G.; Suciu, C.P.; Manea, A.R. Real-Time Home Automation System Using BCI Technology. *Biomimetics* **2024**, *9*, 594. [CrossRef]
31. Rsapberry Pi 4. Available online: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/ (accessed on 20 March 2025).
32. Walter, J.; Fakih, M.; Grüttner, K. Hardware-Based Real-Time Simulation on the Raspberry Pi. In Proceedings of the 2nd Workshop on High-Performance and Real-Time Embedded Systems (HiRES 2014), Vienna, Australia, 20–22 January 2014.
33. Landolfi, E.; Salvi, A.; Troiano, A.; Natale, C. Hardware-in-the-loop validation of an adaptive model predictive control on a connected and automated vehicle. *Int. J. Adapt. Control. Signal Process.* **2023**, *37*, 1459–1491. [CrossRef]
34. Amoozar, M.H.; Zhang, Y.M. Trajectory tracking of Wheeled Mobile Robots: A kinematical approach. In Proceedings of the 2012 IEEE/ASME 8th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications, Suzhou, China, 8–10 July 2012. [CrossRef]

35. Zambelli, M.; Karayiannidis, Y.; Dimarogonas, D.V. Posture regulation for unicycle-like robots with prescribed performance guarantees. *IET Control. Theory Appl.* **2015**, *9*, 192–202. [CrossRef]
36. Alcala, E.; Puig, V.; Quevedo, J.; Escobet, T.; Comasolivas, R. Autonomous vehicle control using a kinematic Lyapunov-based technique with LQR-LMI tuning. *Control. Eng. Pract.* **2018**, *73*, 1–12. [CrossRef]
37. Vazquez, M.; Ardito-Proulx, M.; Wadoo, S. Lyapunov Based Trajectory Tracking Dynamic Control for a QBOT-2. In Proceedings of the 2020 IEEE Integrated STEM Education Conference (ISEC), Princeton, NJ, USA, 1 August 2020. [CrossRef]
38. Liu, M.; Fang, S.; Dong, H.; Xu, C. Review of digital twin about concepts, technologies, and industrial applications. *J. Manuf. Syst.* **2021**, *58*, 346–361. [CrossRef]
39. Vered, Y.; Elliott, S.J. The use of digital twins to remotely update feedback controllers for the motion control of nonlinear dynamic systems. *Mech. Syst. Signal Process.* **2023**, *185*, 109770. [CrossRef]
40. Saad, A.; Faddel, S.; Mohammed, O. IoT-based digital twin for energy cyber-physical systems: Design and implementation. *Energies* **2020**, *13*, 4762. [CrossRef]
41. P3-DX Robot. Available online: https://www.generationrobots.com/media/Pioneer3DX-P3DX-RevA.pdf?srsltid=AfmBOopigKPz8ste_THqrR6Gy0mbhTVBrQ8hLlfLf8HZTO2f8OTK4PxR (accessed on 20 March 2025).
42. Simulink Coder. Available online: https://ch.mathworks.com/products/simulink-coder.html (accessed on 20 March 2025).
43. Cristina Losada-Gutiérrez, C.; Espinosa, F.; Santos-Pérez, C.; Marrón-Romera, M.; Rodriguez-Ascariz, J.M. Remote control of a robotic unit: A case study for control engineering formation. *IEEE Trans. Educ.* **2020**, *63*, 246–256. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

actuators

MDPI

Article

Quadratic Programming Vision-Based Control of a Scale-Model Autonomous Vehicle Navigating in Intersections

Esmeralda Enriqueta Mascota Muñoz [1], Oscar González Miranda [2], Xchel Ramos Soto [1], Juan Manuel Ibarra Zannatha [2] and Santos Miguel Orozco Soto [3,*]

[1] Science and Technology College, Autonomous University of Mexico City, Mexico City 06720, Mexico
[2] Automatic Control Department, Research and Advanced Studies Center of the National Polytechnic Institute, Mexico City 07360, Mexico; jibarra@cinvestav.mx (J.M.I.Z.)
[3] Faculty of Engineering, Free University of Bolzano, 39100 Bolzano, Italy
* Correspondence: santos.orozco@uacm.edu.mx

Abstract

This paper presents an optimal control for autonomous vehicles navigating in intersection scenarios. The proposed controller is based on solving a Quadratic Programming optimization technique to provide a feasible control signal respecting actuator constraints. The proposed controller was implemented in a scale-sized vehicle and is executed using only on-board perception and computing systems to retrieve the state dynamics, i.e., an inertial measurement unit and a monocular camera, to compute the estimated states through intelligent computer vision algorithms. The stability of the error signals of the closed-loop system was proved both mathematically and experimentally, using standard performance indices for ten trials. The proposed technique was compared against LQR and MPC strategies, showing 67% greater accuracy than the LQR approach and 53.9% greater accuracy than the MPC technique, while turning during the intersection. Moreover, the proposed QP controller showed significantly greater efficiency by reducing the control effort by 63.3% compared to the LQR, and by a substantial 78.4% compared to the MPC. These successful results proved that the proposed controller is an effective alternative for autonomously navigating within intersection scenarios.

Keywords: autonomous vehicles; self-driving vehicles; tracking control; QP control; vision-based control; intersection scenario

1. Introduction

Over the past decade, autonomous driving systems have attracted increasing interest from both industry and research communities. These systems are a cornerstone for reducing traffic accidents, improving traffic efficiency, and enabling new forms of mobility. A key factor in the development of autonomous driving systems is the design of controllers capable of safely and accurately following desired trajectories, especially in complex urban environments [1].

Most of the advances in the development of self-driving vehicles have started as research projects using scale-sized prototypes, as is the case of the project reported in this work, which was developed in academic laboratory conditions using a scale-sized car, where the road surface is flat and there are no adverse atmospheric conditions; this work is focused in the problem of controlling this car to manage the navigation in standard urban intersections.

Intersections are particularly critical points in traffic networks, often leading to delays and accidents. For this reason, Autonomous Intersection Management (AIM) is an active area of research, where path planning is performed for each vehicle in cruising scenarios, and vehicle movements are executed through automation [2,3].

Considering the literature reported so far, the research gaps found are the following: (1) Control systems for navigating in intersections require oversimplified models and complex sensing and communication schemes. (2) Several strategies neglect the lower-level vehicle controller implementation details and integration. (3) There is a substantial predominance on simulation-based validation. (4) The use of off-board computing and sensing reduces the reliability on real-world scenarios. (5) The prevalently employed controllers are not capable to handle real-world constraints, such as actuator limits.

Aiming to fill out the aforementioned research gaps, this work presents an autonomous control architecture that enables lane following and intersection navigation using only an inertial measurement unit (IMU), a monocular camera, and computer vision algorithms. The system integrates visual processing for lane and intersection detection, vehicle state estimation, and a Quadratic Programming (QP)-based controller that determines the vehicle's actions in real time, which in contrast to standard optimal controllers such as the linear quadratic regulator (LQR) or predictive algorithms, allows the handling of inequality constraints with less computational burden, directly computing the control input without a prediction horizon, as is presented in [4]. The proposed architecture consists of several modules: a visual perception module that processes images captured by a forward-facing camera to identify lane markings; a Convolutional Neural Network (CNN) to detect cruising scenarios; a state estimation module that interprets visual and inertial data to define the vehicle's current and desired states; and a switching controller module that implements an optimal control scheme based on QP to compute steering and velocity commands, enabling the vehicle to achieve navigation objectives while satisfying the imposed actuator constraints.

This paper is organized as follows. Section 2 presents the related work, while Section 3 presents dynamic model of the vehicle and the hardware/software architecture employed. Section 4 details the visual feedback pipeline, covering road perspective correction, state estimation, and CNN-based scene classification. Section 5 describes the proposed QP controller and its stability. Section 6 reports the experimental results obtained in real-world navigation scenarios, comparing the proposed QP controller against LQR and Model-Predictive Control (MPC) baselines. Finally, Section 7 summarizes the main conclusions and outlines future research directions.

2. Related Work

The navigation control of autonomous vehicles is taken for granted in navigation scenarios, so most research is instead dedicated to traffic monitoring and control, thereby leaving aside the potential implications of the lower-level controller implementation. Nonetheless, some authors have attempted to solve the navigation in intersections control problem, so diverse schemes have been reported in the literature so far, such as PID, Optimal Control, and MPC. However, most studies in the literature develop control laws based on overly simplified models, requiring a constant communication between vehicles and others agents, also assuming that many vehicle states are measured. In addition, a substantial majority of these results are obtained solely through numerical simulations, leaving aside the feedback state measurement or estimation algorithms [1,3,5–9].

For example, Vitale et al. [9], propose a centralized control method to manage traffic at road intersections. Their scheme solves an optimization problem where the states include the position and velocity of the vehicles, and the output is an acceleration profile for each

vehicle within the intersection. They present simulation results showing that vehicles can cross safely without collisions, maintaining a safety distance of six meters. However, their system requires constant and perfect communication between vehicles, and the model assumes that vehicles cannot make turns. These limitations hinder its applicability in real-world environments.

Another strategy is presented by Ding et al. [8], where a control system for autonomous vehicles is used to perform turning maneuvers at intersections. In their work, vehicles communicate with Road Side Units and with each other to obtain the position and velocity of nearby vehicles. Using a Model-Predictive Control (MPC) scheme, each vehicle calculates its acceleration and steering angle to execute left turns, right turns, or U-turns while cruising—adjusting its speed accordingly and maintaining a safe distance. Nevertheless, this system relies on a communication infrastructure that is subject to latency, data loss, and limited coverage. Additionally, high precision in vehicle positioning is required.

Similarly, Meng and Cassandras [7] propose a real-time optimal control method for autonomous vehicles approaching signalized intersections under free-flow conditions. The goal is to minimize both energy consumption and travel time while avoiding idle stops. By leveraging vehicle-to-infrastructure (V2I) communication, their algorithm dynamically adjusts vehicle speed profiles to enable non-stop intersection crossings. The control formulation assumes a single autonomous vehicle operating in free-flow mode, excluding interaction effects, lateral maneuvers, and lane constraints. The framework also depends on precise V2I signal timing, and potential discomfort caused by acceleration spikes near the onset of green lights may require additional buffering strategies. Furthermore, the method assumes straight-line motion without turns, and does not address the feasibility of lane changes.

In contrast, the navigation of autonomous vehicles in cruising scenarios without external communication systems has been less explored in the literature [10–13]. For example, Pourjafari et al. [13] propose a predictive scheme for navigating unsignalized road intersections. Their objective is to predict the behavior of other vehicles, determine the order of passage at collision points, and compute an acceleration profile for the ego vehicle. The proposed control method considers multiple constraints, while the states of other vehicles are estimated using Long Short-Term Memory (LSTM) and Graph Neural Networks. Ultimately, they present simulation results based on 250 real-world scenarios, demonstrating that the vehicle can navigate safely and, in some cases, outperform human drivers. However, the method relies heavily on the precision of neural network data and has not been experimentally validated.

On the other hand, Zannatha et al. [12,14] have presented a modular navigation system for scaled autonomous vehicles operating in urban environments. Their approach integrates route planning using Dijkstra's algorithm, geometric trajectory generation adapted to intersection geometry, and a hybrid navigation controller that combines odometry-based Stanley steering with trapezoidal velocity profiling. All modules were validated in a ROS–Gazebo simulator within a 2D urban environment featuring 90° intersections and a uniform road layout. However, the method assumes idealized urban geometry—excluding roundabouts, diagonal roads, or variable curvatures. Only a single vehicle is considered, with no dynamic agents or multi-agent coordination. Their controller relies heavily on odometry, and lane perception is limited to geometric assumptions.

Sezer et al. [11] present a decision-making system for autonomous vehicles that infers the intentions of other drivers and makes safe and efficient decisions to cross or merge at unsignalized T-intersections. To achieve this, they use a probabilistic model known as a Mixed Observability Markov Decision Process (MOMDP). The system was implemented on an autonomous vehicle (a modified golf cart) and tested at a real intersection on the campus

of the National University of Singapore. In their experiments, the system successfully detected changes in the intentions of aggressive drivers, avoided collisions, and chose safe actions. However, the model assumes that the positions of both the autonomous and human-driven vehicles are perfectly observable, which may not hold in practice due to sensor errors, occlusions, or adverse conditions. Additionally, the approach is computationally expensive.

Additionally, control strategies based on Quadratic Programming are attractive because they yield optimal solutions while simultaneously accounting for state and control input constraints. However, this approach has been primarily used for trajectory tracking [15–19], and mostly present only simulation results. In Lee [19], a soft-constrained variant of the constrained iterative Linear Quadratic Regulator (CILQR) algorithm is proposed for lane-keeping control in autonomous vehicles. The controller introduces slack variables to relax state and input constraints within a barrier-function-based optimization framework, aiming to suppress steering jitter without relying on external filters. Performance is validated through numerical simulations and vision-based experiments in perturbed environments.

Similarly, Yuan and Zhao [17] present a hierarchical trajectory-tracking controller for autonomous vehicles that couples lateral and longitudinal dynamics using a hybrid Linear LQR and MPC framework. The architecture integrates an Extended Kalman Filter (EKF) observer to estimate driving states (e.g., velocity and yaw rate), aiming to improve accuracy under state uncertainty. The framework is validated using CarSim–Simulink co-simulation across various speed scenarios and benchmarked against a classical MPC-only control scheme. However, the EKF's accuracy depends on well-tuned noise parameters, which may vary across scenarios.

Considering the published results regarding autonomous vehicles control in intersections so far, this paper presents a low-level control system that only uses on-board sensors and computers to accurately navigate in intersections, avoiding communication between agents and external sensing in contrast to the works found in the literature. Moreover, the proposed control scheme takes into account actuator constraints and it is implemented on a real scale-model vehicle, which are clear advantages over the state of the art.

3. Autonomous Vehicle Description

3.1. Vehicle Dynamics

The motion of autonomous vehicles is typically modeled using dynamic equations that capture the behavior of the car's body and its interaction with the environment. A widely adopted approach is the so-called *bicycle* model, which simplifies the car dynamics while preserving key characteristics such as turning radius, slip angle, etc. [20]. This model is crucial for designing and implementing predictive control algorithms. Consider Figure 1, where both the longitudinal and lateral velocities, V_x and V_y, are shown, as well as the geometric parameters of the vehicle detailed in Table 1. The lateral dynamics of the vehicle are governed by the following force and moment equations,

$$\begin{aligned} m\dot{V}_y(t) &= F_{yf} + F_{yr}\,, \\ I_z\ddot{\psi}(t) &= l_f F_{yf} - l_r F_{yr}\,, \end{aligned} \tag{1}$$

where F_{yf} and F_{yr} are the lateral tire forces at the front and rear wheels, respectively, which in turn are given by

$$\begin{aligned} F_{yf} &= 2C_{cf}(\delta - \theta_f)\,, \\ F_{yr} &= 2C_{cr}(-\theta_r)\,. \end{aligned} \tag{2}$$

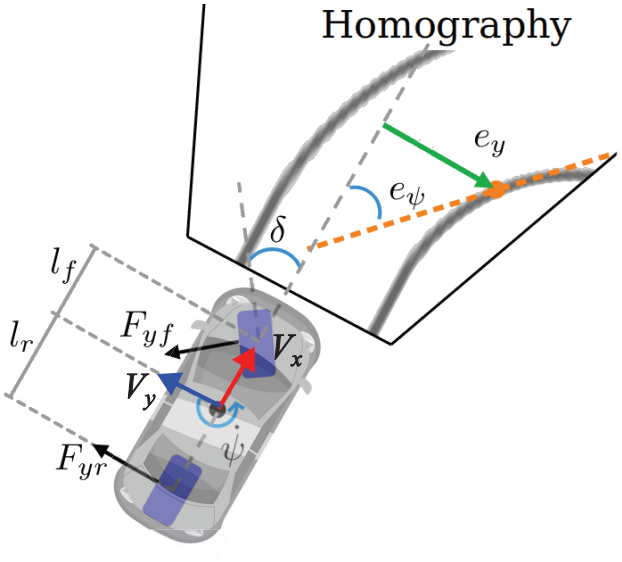


Figure 1. Diagram of parameters involved in the simplified vehicle model. Red and Blue arrows denote the vehicle's reference frame axes.

Table 1. Parameter description.

Symbol	Description
C_{cf}, C_{cr}	Cornering stiffness of front and rear tires
ℓ_f, ℓ_r	Distance from Center of Mass to front and rear axles
m	Mass of the vehicle
I_z	Yaw moment of inertia
V_x	Longitudinal velocity

The angles θ_f and θ_r, known as the tire slip angles. After some algebra and identities it follows that

$$\begin{aligned} \tan(\theta_f) &= \frac{V_y + l_f\dot{\psi}}{V_x}\,, \\ \tan(\theta_r) &= \frac{V_y - l_r\dot{\psi}}{V_x}\,, \end{aligned} \tag{3}$$

so the lateral dynamics can be linearized as follows about small values of θ_f and θ_r

$$\begin{aligned} \dot{V}_y(t) &= -\left(\frac{2C_{cf}+2C_{cr}}{mV_x}\right)V_y - \left(V_x + \frac{2C_{cf}l_f + 2C_{cr}l_r}{mV_x}\right)\dot{\psi} + \left(\frac{2C_{cf}}{m}\right)\delta\,, \\ \ddot{\psi}(t) &= -\left(\frac{2C_{cf}l_f - 2C_{cr}l_r}{I_zV_x}\right)V_y - \left(\frac{2C_{cf}l_f^2 + 2C_{cr}l_r^2}{I_zV_x}\right)\dot{\psi} + \left(\frac{2C_{cf}l_f}{I_z}\right)\delta\,. \end{aligned} \tag{4}$$

Afterwards, defining the tracking errors

$$\begin{aligned} e_y &= y - y_{ref}\,, \\ e_\psi &= \psi - \psi_{ref}\,, \\ \Rightarrow \dot{e}_y &= V_y + V_x e_\psi \quad \text{with: } \dot{y}_{ref} = -V_x e_\psi\,, \\ \dot{e}_\psi &= \dot{\psi} - \dot{\psi}_{ref}\,, \end{aligned}$$

and folding the dynamic equations into them it yields

$$\begin{aligned}\ddot{e}_y &= -\left(\frac{2C_{cf}+2C_{cr}}{mV_x}\right)V_y - \left(V_x + \frac{2C_{cf}l_f + 2C_{cr}l_r}{mV_x}\right)\dot{\psi} + \left(\frac{2C_{cf}}{m}\right)\delta + V_x\dot{e}_\psi \\ &= -\left(\frac{2C_{cf}+2C_{cr}}{mV_x}\right)(\dot{e}_y - V_x e_\psi) - \left(V_x + \frac{2C_{cf}l_f + 2C_{cr}l_r}{mV_x}\right)(\dot{e}_\psi + \dot{\psi}_{ref}) \\ &\quad + \left(\frac{2C_{cf}}{m}\right)\delta + V_x\dot{e}_\psi \,,\end{aligned}$$

$$\begin{aligned}\ddot{e}_\psi &= -\left(\frac{2C_{cf}l_f - 2C_{cr}l_r}{I_z V_x}\right)V_y - \left(\frac{2C_{cf}l_f^2 + 2C_{cr}l_r^2}{I_z V_x}\right)\dot{\psi} + \left(\frac{2C_{cf}l_f}{I_z}\right)\delta - \ddot{\psi}_{ref} \\ &= -\left(\frac{2C_{cf}l_f - 2C_{cr}l_r}{I_z V_x}\right)(\dot{e}_y - V_x e_\psi) - \left(\frac{2C_{cf}l_f^2 + 2C_{cr}l_r^2}{I_z V_x}\right)(\dot{e}_\psi + \dot{\psi}_{ref}) \\ &\quad + \left(\frac{2C_{cf}l_f}{I_z}\right)\delta - \ddot{\psi}_{ref} \,.\end{aligned}$$

Finally, considering V_x constant, the derivatives of the reference signals are set to zero (position regulation), and after rearranging terms, the state-space representation of the vehicle model becomes

$$\begin{pmatrix}\dot{e}_y\\ \ddot{e}_y\\ \dot{e}_\psi\\ \ddot{e}_\psi\end{pmatrix} = \underbrace{\begin{pmatrix}0 & 1 & 0 & 0\\ a_{21} & 0 & a_{23} & a_{24}\\ 0 & 0 & 0 & 1\\ a_{41} & 0 & a_{43} & a_{44}\end{pmatrix}}_{A}\underbrace{\begin{pmatrix}e_y\\ \dot{e}_y\\ e_\psi\\ \dot{e}_\psi\end{pmatrix}}_{x} + \underbrace{\begin{pmatrix}0\\ b_2\\ 0\\ b_4\end{pmatrix}}_{B}\delta \,. \tag{5}$$

The expressions of matrices A and B elements are listed in Table 2.

Table 2. Elements of matrices A and B.

Element	Value	Element	Value
a_{21}	$\frac{2C_{cf}+2C_{cr}}{mV_x}$	a_{43}	$\frac{2C_{cf}\ell_f - 2C_{cr}\ell_r}{I_z}$
a_{23}	$\frac{2C_{cf}+2C_{cr}}{m}$	a_{44}	$\frac{-2C_{cf}\ell_f^2 + 2C_{cr}\ell_r^2}{I_z V_x}$
a_{24}	$\frac{-2C_{cf}\ell_f + 2C_{cr}\ell_r}{mV_x}$	b_2	$\frac{2C_{cf}}{m}$
a_{41}	$\frac{-2C_{cf}\ell_f + 2C_{cr}\ell_r}{I_z V_x}$	b_4	$\frac{2C_{cf}\ell_f}{I_z}$

Then, the discrete-time version of (5) is given by

$$x_{k+1} = A_d x_k + B_d \delta_k \,, \tag{6}$$

where $A_d = e^{hA}$ and $B_d = \int_0^h e^{sA}B\, ds$ for a given sampling period h, with x_0 denoting the initial measured errors vector within the range -7.2 cm $\leq e_{y_0} \leq 15.1$ cm and -0.37 rad $\leq e_{\psi_0} \leq 0.25$ rad. Such a model is useful to design the control system in discrete time, which in turn will be executed as a program within an embedded computer.

3.2. The AutoMiny 4.0

The experimental platform in study is the AutoMiny 4.0, which is a miniature autonomous vehicle designed for research and educational purposes. It integrates

multiple sensors and actuators that are necessary for autonomous navigation (https://autominy.github.io/AutoMiny/, accessed on 30 July 2025) [14]. The AutoMiny vehicle is illustrated in Figure 2. It is equipped with an ELP USBFHD01M fisheye camera (Ailipu Technology Inc., Shenzhen, China) to provide a wide field of view, a 360° A2M8 LiDAR (SLAMTEC, Shanghai, China) for obstacle detection, and a BNO055 IMU (Bosch Sensortec, Reutlingen, Germany) for orientation and motion sensing. For actuation, it uses a TowerPro SG-5010 servomotor (Tower Pro, Kowloon, Hong Kong) for steering and a 2232B 012 BX4 SC (Faulhaber, Schönaich, Germany) brushless DC motor for traction. Moreover, an Arduino Nano manages the low-level motor control. Additionally, the perception, control, and navigation algorithms are executed by an Intel NUC8i5BEK mini-PC (Intel, Santa Clara, CA, USA), running Ubuntu 18.04 with the Robot Operating System (ROS). All components are mounted on a rigid Alucobond chassis. Such components are shown in Figure 3, whereas Figure 4 illustrates the system block diagram representing the signal interchange between on-board electronic devices.

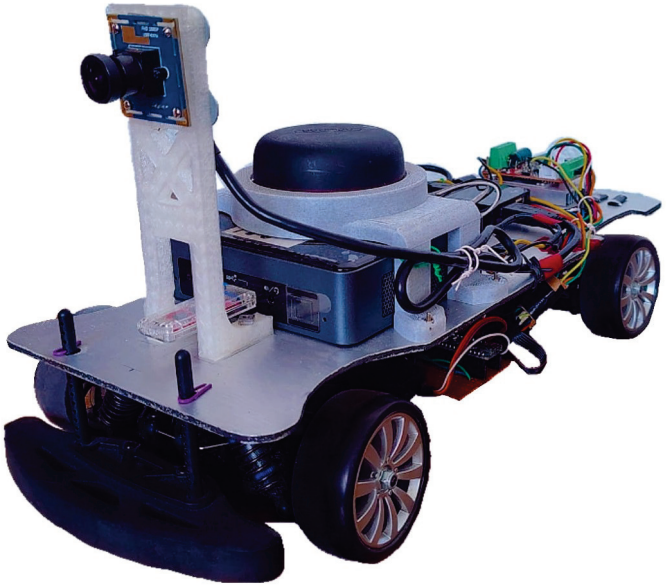


Figure 2. Assembled AutoMINY platform.

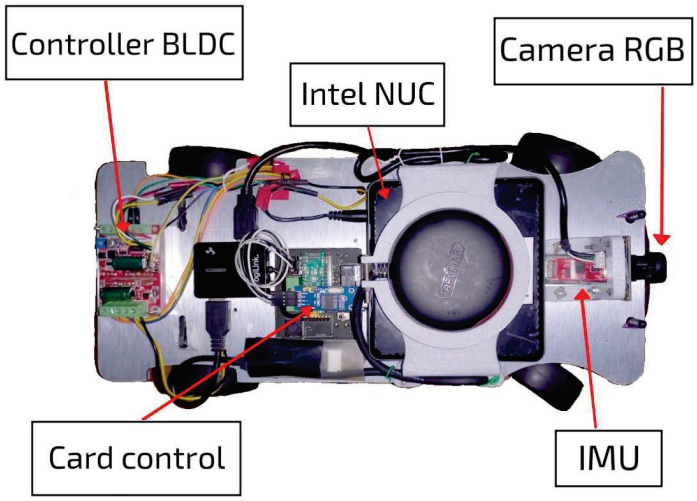


Figure 3. AutoMINY with labeled components including sensors, actuators, and on-board computer.

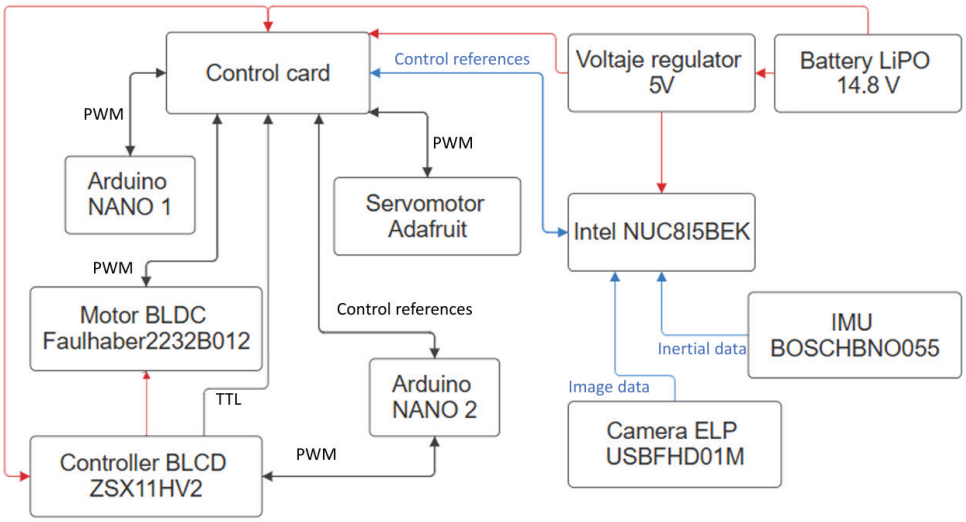


Figure 4. System diagram of the AutoMINY platform. Red arrows indicate power connections (5 V and 14.8 V); blue arrows represent USB communication links; black arrows denote other connections between components.

3.3. ROS Software Configuration

The ROS Melodic framework provides a robust infrastructure for distributed integration and communication among sensors, actuators, and control algorithms. Furthermore, it leverages a publisher-subscriber communication model, enabling data exchange between nodes through topics, allowing both temporal and spatial decoupling between components. The nodes implemented to execute the control of the AutoMiny are as follows:

- `camera_ELP`: Captures images from the front-facing fisheye lens camera at a frequency of 30 Hz and publishes them on the topic `Image_Raw`.
- `bno_055`: Collects inertial data from the BNO055 IMU sensor, including acceleration, angular velocity, and orientation (quaternion or Euler angles). The processed data is published on the topic `bno_055/output`.
- `QP_control`: Receives camera and IMU data from `Image_Raw` and `bno_055/output`, respectively. It implements a convolutional neural network (CNN) to classify the scene as either "in intersection" or "out of intersection". Based on this classification, it switches between two QP-based controllers, each tailored to one of the cases. At the end, it publishes the steering angle δ on `steering`, and the desired longitudinal speed V_x on `speed`.
- `gps_pose`: Receives ground-truth camera images and performs visual processing, including color segmentation, homography transformation, and geometric feature extraction. It then provides absolute position and orientation data for experimental validation. The 2D pose is published on the topic `pose_car`, and the error states e_y and e_ψ are published on `error_car`.
- `LQR_control` and `MPC_control`: These nodes implement alternative control strategies for trajectory tracking using LQR and MPC, respectively. Both nodes receive reference trajectories and vehicle state information, and output control commands. The steering angle δ and desired speed V_x are published on the topics `steering` and `speed`, respectively.

The interaction between the aforesaid ROS nodes is presented in Figure 5 for the LQR and MPC approaches, and in Figure 6 for the QP controller, where it can be clearly shown that the proposed QP does not depend on a ground-truth external measurement (`gps_pose`), but only gets feedback from the on-board sensors.

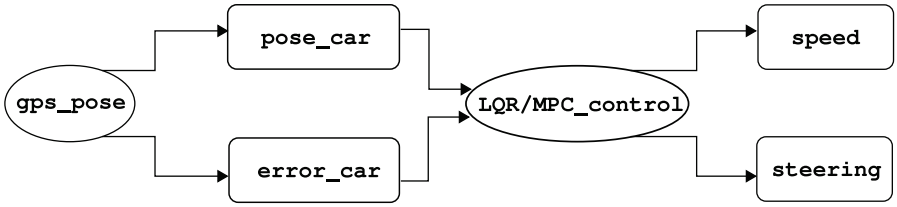


Figure 5. ROS-based modular architecture using LQR or MPC control.

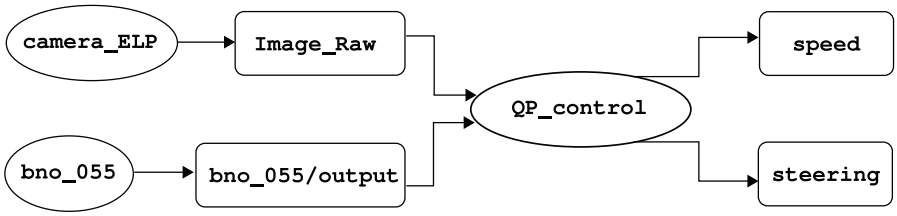


Figure 6. ROS-based modular architecture using QP control.

4. Visual Feedback

4.1. Road Perspective Correction

Consider the 2D plane captured by the on-board camera, also known as the image space, denoted by

$$I_{sp} \triangleq \left\{ (u, v) \in \mathbb{Z}_+^2 \;\middle|\; u \in [1, p],\ v \in [1, q] \right\} \tag{7}$$

where u and v denote the horizontal and vertical pixel coordinates, and $p \times q$ is the image resolution. Also consider the coordinate frame associated with the road, denoted by $\{W\}$. The trajectory coordinates of the self-driving car are defined in the road frame $\{W\}$ and expressed in meters. However, in the corresponding image, this trajectory appears distorted due to the projection model of the camera and is expressed in pixels. Since the vehicle control is computed in the image space, it is necessary to establish a model that relates the coordinates in the road frame $\{W\}$ to those in the image space I_{sp} (in pixels).

Such a model corresponds to the camera projection applied to a 2D plane (the road surface), rather than the general 3D workspace model. This reduced model is known as a homography, and it is obtained by projecting planar world points into the image using the camera model, which is defined by three matrices: the intrinsic calibration matrix, the perspective projection, and the homogeneous transformation representing the pose of the camera with respect to the world frame. The full projection model is given by

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} k_x & 0 & c_x \\ 0 & k_y & c_y \\ 0 & 0 & 1 \end{bmatrix}}_{K} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R_{3\times3} & t_{3\times1} \\ 0_{1\times3} & 1 \end{bmatrix} \begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix}, \tag{8}$$

where k_x, k_y, c_x, and c_y are real scalars representing the intrinsic camera parameters, $R \in \mathcal{SO}(3)$ is the rotation matrix, and $t \in \mathbb{R}^3$ is the translation vector, both defining the pose of the camera with respect to the world frame. It is important to note that the world frame axes are selected so that axis z is pointed down to make the correspondence between the coordinates in the world and image frames easier. Now, if the points of interest lie on the plane defined by $(X', Y', 0, 1)^T$, the (8) projection model is reduced to an homography as

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} k_x & 0 & c_x \\ 0 & k_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_{2\times2} & t_{2\times1} \\ 0_{1\times2} & 1 \end{bmatrix} \begin{bmatrix} X' \\ Y' \\ 1 \end{bmatrix}, \quad = \begin{bmatrix} k_x & 0 & c_x \\ 0 & k_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta) & \sin(\theta) & t_x \\ \cos(\theta) & -\sin(\theta) & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X' \\ Y' \\ 1 \end{bmatrix} = H \begin{bmatrix} X' \\ Y' \\ 1 \end{bmatrix}, \tag{9}$$

where (t_x, t_y, θ) is the vehicle pose and H is the homography matrix. Given a set of points on the road surface with known world coordinates (in meters) and their corresponding image coordinates (in pixels), obtained from a known pose (used to initialize the car odometry), it is possible to estimate the inverse homography matrix used to reconstruct the workspace corresponding only to the road surface

$$\begin{bmatrix} X' \\ Y' \\ 1 \end{bmatrix} = H^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}. \tag{10}$$

Homography matrix and its inverse are functions of the actual vehicle pose (t_x, t_y, θ). Even if these matrices are identified in a specific part of the workspace, i.e., for a specific pose $(R_{2\times2}, t_{2\times1})$, it can be used all over the road surface because the absolute coordinates of the features on the road surface are not needed and only their relative position respect to the car is. In addition, the image analysis to find features in the workspace (e.g., lane markings, intersection, horizontal signage, etc.) is applied only in a relevant area of the reconstructed space in front of the car, so only a rectangle of 3.5 m $\times$ 2.1 m in front of the vehicle is used. Thus, features in the workspace (e.g., lane markings, intersection, horizontal signage) can be localized and measured and lateral errors can be evaluated through some image processing and analysis algorithms, as described in the following sections. In Figure 7, the steps to obtain the homography transformation of the road are shown.

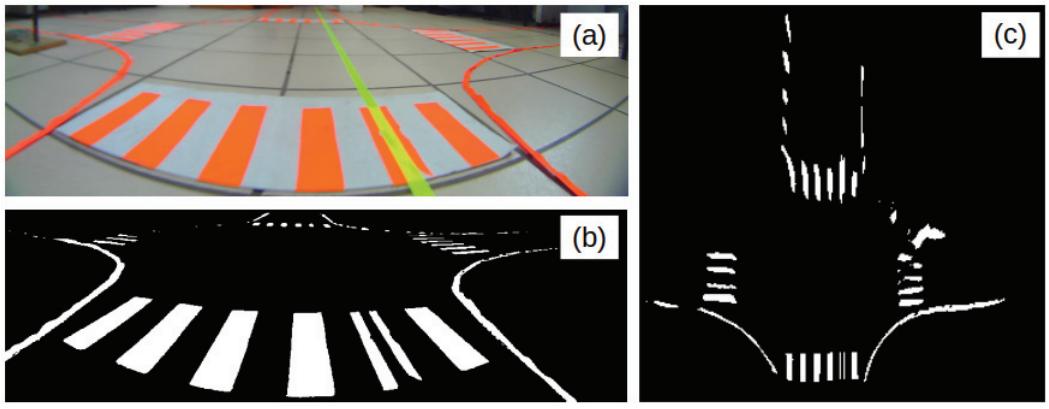


Figure 7. Perspective correction of the road. (**a**) Initial RGB image of the road. The orange lines are the road signs. The yellow line is a part of the experimental setup used for other works regardless this paper. (**b**) Color segmentation. (**c**) Homography of the road.

4.2. Ground-Truth System

Ground truth refers to accurate and verified information used as a reference to evaluate the performance of computer vision models. It plays a critical role in a model's ability to generalize and make correct decisions in real-world scenarios by reducing errors and biases, leading to a balanced decision-making and improving prediction accuracy.

Figure 8 illustrates the system's processing pipeline, which begins with image acquisition using a fisheye camera mounted on the ceiling. The captured images are corrected

to eliminate tangential lens distortion and subsequently transformed into a top-down (bird's-eye) view using the homography transformation (10). Afterwards, a color-based segmentation is applied to identify markers placed on the vehicle, and finally, image moments are computed from these segmented regions to determine the vehicle's position and orientation within the environment by locating the centroids of each marker color.

Figure 9 shows the colored markers placed on the top of the vehicle (e.g., green and blue patches) to facilitate visual localization via segmentation. The left image shows the vehicle in the physical environment, while the right image displays the homography-transformed top-down view, where the vehicle's position and orientation can be clearly observed. This system is essential for quantifying tracking errors and validating the performance of the implemented controllers.

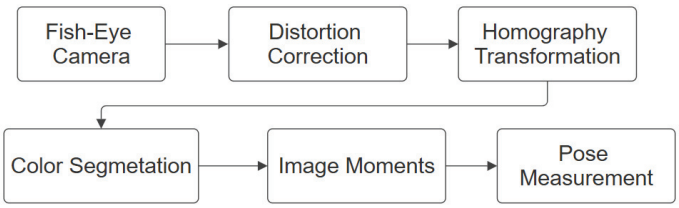


Figure 8. Processing pipeline of the ground-truth system. The pipeline includes image acquisition from a ceiling-mounted fisheye camera, distortion correction, homography transformation to obtain a bird's-eye view, color segmentation for marker detection, and pose estimation using image moments.

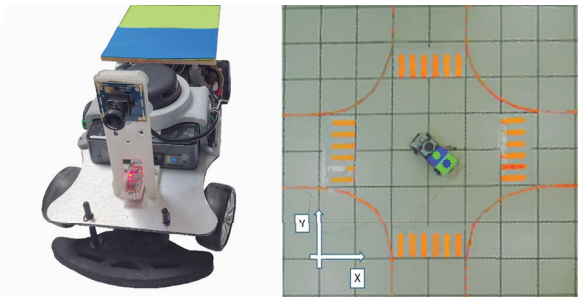


Figure 9. Visual localization of the vehicle using colored markers. Left: image of the physical scene showing the top of the vehicle with green and blue markers. Right: top-down view obtained through homography, enabling precise estimation of vehicle position and orientation.

4.3. Convolutional Neural Network

In order to select the type of controller that the AutoMiny must follow, a Convolutional Neural Network (CNN) was trained to detect whether the car is inside or outside the intersection. The input to the CNN is an homography-cropped image of size $180 \times 300 \times 1$, which is classified into one of two categories: **In** or **Out**. The architecture of the network is shown in Figure 10, where it can be appreciated that the input image is the segmented homography. Moreover, the training dataset was built by recording road images while the car navigated through the intersection. In each image, the lane markings were segmented, and an homography transformation was applied. A total of 5636 images were used for training, and 628 were used for validation. Figure 11 shows three examples of each class. After training, the network achieved an accuracy of 97.9%. The confusion matrix for the validation set is shown in Figure 12.

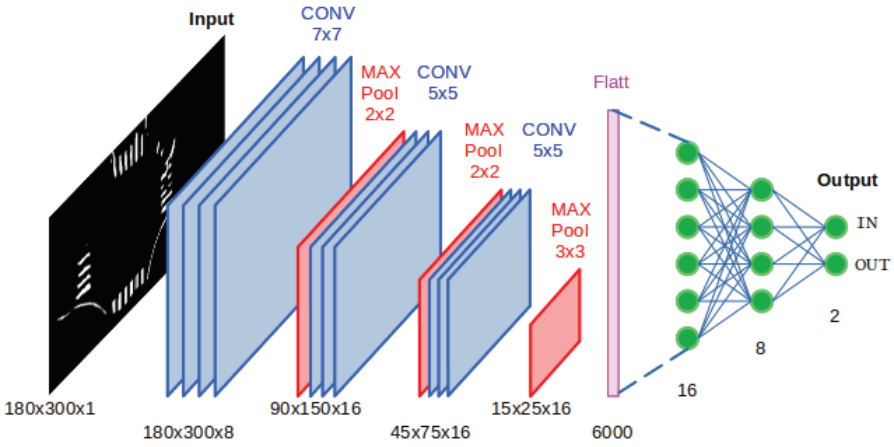


Figure 10. Architecture of the CNN.

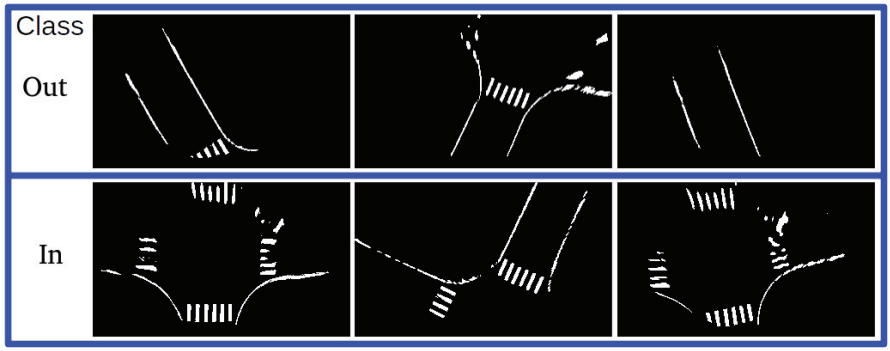


Figure 11. Examples from the dataset used to train the CNN. The first row corresponds to the **Out** class, and the second to the **In** class.

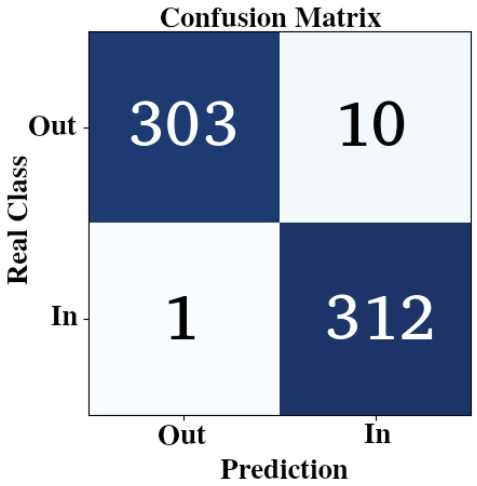


Figure 12. Confusion matrix for the validation of a 313 images dataset. The network classifies images with an accuracy of 97.9%, since 303 were correctly classified and 10 missclassified for the OUT class, while 312 were correctly classified and 1 missclassified for the case of the IN class.

4.4. States Computation Through Visual Feedback

Recalling the error dynamics (6), the lateral error e_y and heading error e_ψ are measured using the homography of the road. Specifically, each image captured by the on-board camera undergoes an homography transformation, followed by a color segmentation step to extract the lane markings. Then, two points on the right lane line are identified using a line detector. These points, denoted as (x_1, y_1) and (x_2, y_2), are used to compute the slope m of a secant line. The lateral and heading errors are then calculated as $e_y = y_1 - y_{ref}$

and $e_\psi = -\arctan(m)$, respectively. These variables are illustrated in Figure 13, where its relation with the visual information can be appreciated. Notice that the states e_y and e_ψ are according to the image frame as is explained in Section 4.

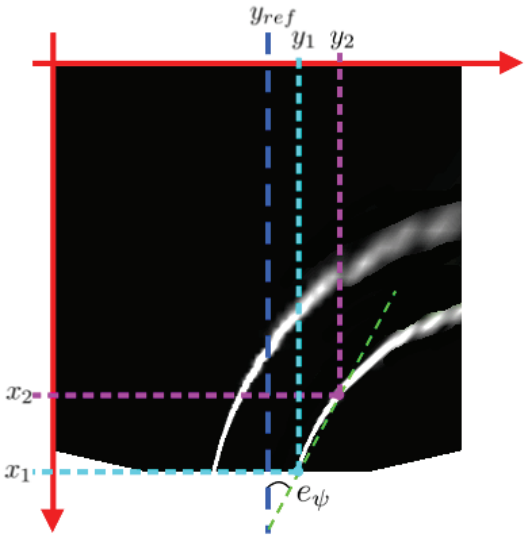


Figure 13. Measurement of the states e_y and e_ψ using the homography of the road. The heading error e_ψ is defined as the angle between the secant line of the right lane marking (in green) and the optical axis (in blue). The lateral error is given by $e_y = y_1 - y_{ref}$.

5. Quadratic-Programming Control

Quadratic Programming (QP) is a type of mathematical optimization problem in which the objective function is quadratic and the constraints are linear. It is a widely used control strategy because it allows for the explicit handling of constraints on system dynamics, states, and actuator limits, which cannot be directly handled by traditional optimal control approaches like the Linear Quadratic Regulator (LQR). The objective function typically reflects goals such as minimizing control effort or tracking error, while the constraints can include the system dynamics, actuator saturation, state bounds, and safety requirements.

Considering the error dynamics (6), and the control problem of leading to zero the states, a cost objective function can be defined as

$$\begin{cases} J(\delta_k) = \sum_{k=0}^{\infty} x_k^T Q x_k + R\delta_k^2 + g(\delta_k)\,, \\ g(\delta_k) = \rho \cdot \max(0,\ |\delta_k| - 1)^2\,, \end{cases} \tag{11}$$

where $Q \in \mathbb{R}^{4\times 4}$ is a positive definite diagonal matrix and $R, \rho \in \mathbb{R}$ are positive scalars. The penalty constant ρ is obtained by varying its value while solving the optimization problem in simulation, it can be later implemented in the real vehicle. Moreover, $g(\delta_k)$ is a differentiable quadratic function that penalizes violations of the actuator constraints. This approach is referred to as a soft constraint [21,22], which introduces a weighted penalty whenever the control input δ_k exceeds the admissible range, which in this case is $\delta_k \in [-1.5, 1.5]$, resulting from the admissible values of the ROS topic `steering`, which in turn maps the corresponding PWM value to move the servomotor within a -0.4 to 0.4 radians or 67.8 to 112.2 deg (0 deg steering corresponds to 90 deg servomotor shaft).

Given a control horizon N_h, at iteration k, it is possible to estimate future states $x_{k+1},\ x_{k+2}, \ldots, x_{k+N_h}$ by applying (6) iteratively. Then, the estimated immediate cost from time step k is

$$J_k = \sum_{j=1}^{N_h} x_{k+j}^T Q x_{k+j} + R\delta_{k+j}^2 + g(\delta_{k+j}) \,. \tag{12}$$

Assuming that this has the form $\delta_{k+j} = -K_{k+j}x_{k+j}$, it is possible to find a set of optimal gains K_{k+j} that minimize (12) using a numerical solver, such as the Broyden–Fletcher–Goldfarb–Shanno (BFGS) (https://docs.scipy.org/doc/scipy/reference/optimize.minimize-bfgs.html, accessed on 2 October 2025) [23]. Then, the control problem formulated as a QP is

$$\begin{array}{ll} \min\limits_{x} & \sum_{j=1}^{N_h} x_{k+j}^T Q x_{k+j} + R\delta_{k+j}^2 + g(\delta_{k+j}) \\ s.t. & -1.5 \leq \delta(k+j) \leq 1.5 \,. \end{array} \tag{13}$$

In practice, two types of controllers are used for navigation. The first is a vision-based controller for lane keeping, using the lane markings as reference. From the images captured by the on-board camera, it is possible to measure the states e_y and e_ψ (as detailed in Section 4.4). In general, this is sufficient to follow the road. However, at an intersection, there are three possible paths and insufficient floor markings to reliably estimate the vehicle's position and orientation. For this reason, a second controller with IMU-based feedback was designed. This controller records the vehicle's yaw angle ψ_0 just before entering the cruise and computes the orientation error e_ψ as follows:

$$e_\psi = \psi - \psi_{ref}, \quad \text{where} \quad \psi_{ref} = \begin{cases} \psi_0 & \text{if going straight (case 01)} \\ \psi_0 + \frac{\pi}{2} & \text{if turning left (case 10)} \\ \psi_0 - \frac{\pi}{2} & \text{if turning right (case 11)} \end{cases} \tag{14}$$

After computing K_{imu} and K_{vis} by minimizing the immediate cost in (12), a controller switching strategy was implemented as follows:

$$\delta_k = \underbrace{(-\zeta_k K_{\text{imu}} + (1-\zeta_k)K_{\text{vis}})}_{K_k} x_k \,, \tag{15}$$

where $\zeta_k = \frac{1}{1+e^{-a(kh-b)}} - \frac{1}{1+e^{-a(kh-c)}}$. Such a function is defined as the difference between two sigmoid functions, as shown in Figure 14. When $\zeta_k \approx 0$, the vision-based controller dominates, whereas when $\zeta_k \to 1$, the IMU-based controller takes over the steering. Figure 15 illustrates the controller switching strategy and how each path in the cruise is labeled.

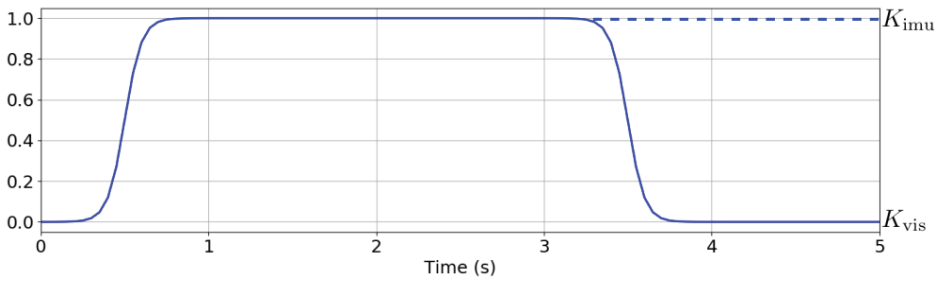


Figure 14. Shape of the function ζ_k defined as the difference between two sigmoid curves. This function creates a smooth transition between controllers over time.

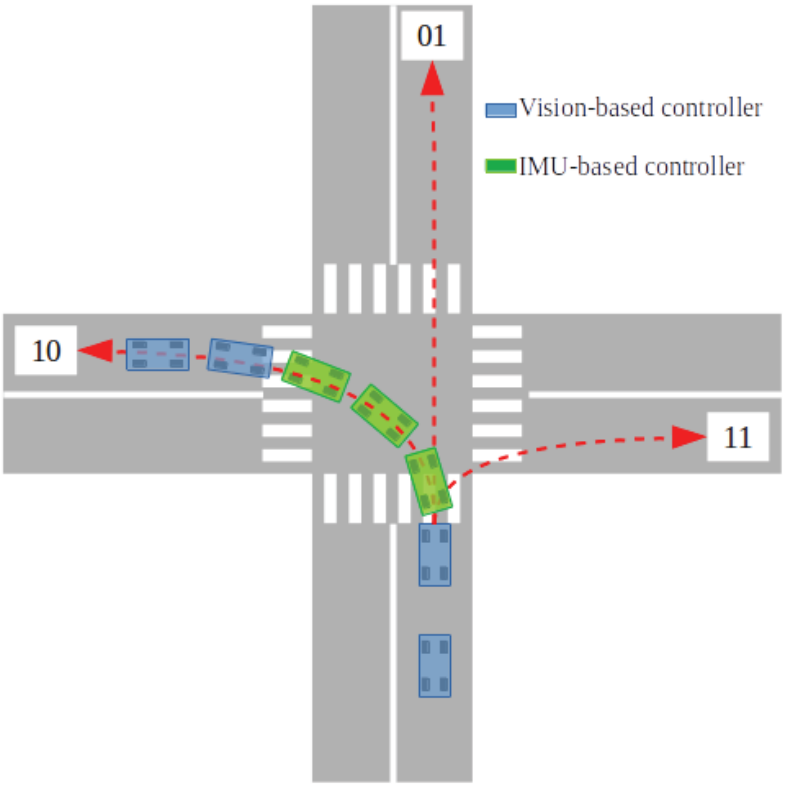


Figure 15. Two controllers were used during navigation: a vision-based controller for lane keeping (in blue), and an IMU-based controller for navigating the cruise section (in green). The three possible paths were labeled as 01, 10, and 11.

Stability

Consider the discrete error dynamics (6). The control problem is to lead $x_k \to 0$ also during the transition between controllers, so the QP controller (15) is proposed, ensuring a smooth transition. Thus the following is assumed:

A1. The longitudinal velocity satisfies $V_x > 0 \; \forall \; k > 0$, hence the pair (A_k, B_k) is controllable.
A2. The QP is convex, despite the linear constraints, since the cost function is convex, so all the solutions are feasible.
A3. The states are always measured at a higher rate than the discrete dynamics and the vehicle has enough power to perform the control action.
A4. The AutoMINY vehicle navigates in a flat surface, so roll and pitch angles are neglected.
A5. The road is always within the camera field of view.

The closed-loop system composed by the error dynamics and the proposed controller can be expressed as

$$\dot{x}_k = (A_k - B_k K_k) x_k \; . \tag{16}$$

The following theorem proves the stability of its solutions x_k.

Theorem 1. *Under Assumptions A1 to A5, the solutions of the state equations describing the closed-loop system are stable, i.e., the solutions* $x_k = (A_k - B_k K_k)^k x_0$ *are stable.*

Proof. To demonstrate the stability of the solutions for the differences Equation (16), the QP optimization problem must be analyzed. Firstly, it is worth remembering that the cost function matrix Q in (13) is symmetric and a positive definite. Afterwards, from the definition of optimal control (Pontryagin's Maximum Principle), one component of the Hamiltonian's gradient yields the optimal controller

$$u_k^* = -K_k^* x_k = -(B_k^T P_{k+1} B + R_k)^{-1} B_k^T P_{k+1} A_k x_k \,,$$

where R is the same proposed in the cost function (13), and $P_{k+1} \in \mathbb{R}^{4\times4}$ is symmetric and positive definite, which is obtained, by

$$P_k = A_k^T P_{k+1} A_k + Q_k - (A_k^T P_{k+1} B_k)(B_k^T P_{k+1} B_k + R_k)^{-1}(B_k^T P_{k+1} A_k) \,.$$

The solution of this finite-horizon problem is obtained by recursively solving the QP as P_{k+1} is obtained; thus, the following Lyapunov candidate function is proposed

$$V_k(x_k) = \frac{1}{2} x_k^T P_k x_k \,,$$

whose difference equation is

$$\Delta V_k(x_k) = V_{k+1}(x_{k+1}) - V_k(x_k) = -(Q_k + K_k^T R_k K_k) x_k \,.$$

Since $\Delta V_k(x_k)$ is negative definite, then the stability of the trajectories of x_k is guaranteed. □

6. Experimental Results

In order to assess the performance of the controller (15) developed in this work, a series of experiments were conducted in which the car followed one of three possible paths in an intersection scenario. For each path, the initial pose of the car was varied. Ten experiments were carried out in total, and during each trial, the Root Mean Square (RMSE) of the lateral error e_y, the Integral Square Error (ISE) of the heading error e_ψ, and the Total Control Effort (TCE) of δ were computed using the following:

$$\text{RMSE } e_y = \sqrt{\frac{1}{N}\sum_{k=1}^{N} e_{y_k}^2} \,, \tag{17}$$

$$\text{ISE } e_\psi = \sum_{k=1}^{N} e_{\psi_k}^2 \,, \tag{18}$$

$$\text{TCE } \delta = \sum_{k=1}^{N} |\delta_k| h \,, \tag{19}$$

where h is the sampling period.

Figure 16 shows both the reference trajectory and the mean trajectory followed by the car during the 10 trials using the proposed controller (15), where it can be shown that the trajectories were tracked with a few centimeters error, which is acceptable for a prototype scaled vehicle. A video of the experiments is available at https://youtu.be/Tnf-_E8l6jo?si=pVEqnw_1i7ZdY6mW (accessed on 30 July 2025).

In order to compare the performance of this controller with other approaches, LQR and MPC controllers were also implemented to follow the ideal trajectory based on ground-truth information. As before, ten experiments were carried out for each case, the initial pose of the car was varied in each trial, and the RMSE of e_y, the ISE of e_ψ, and the TCE of δ were measured. It is worth mentioning that both LQR and MPC require the use of ground-truth data to be implemented; in contrast, QP control is exclusively fedback with the on-board sensors. Table 3 reports the mean values of these metrics over ten experiments for each case. In these experiments, the vehicle navigated within an intersection, following the path of the case 01. The different controllers results are plotted as follows: QP control in Figure 17, LQR control in Figure 18, and MPC control in Figure 19.

Although the best $\overline{\text{RMSE}}$ e_y was achieved by the LQR controller in all cases, it is important to note that the QP controller (15) obtained comparable results using only on-board information from the IMU and the camera, which demonstrates its superiority in more realistic sensing conditions. Moreover, the QP controller resulted in the lowest ISE and TCE across all cases. It can also be observed in Figures 17–19 where the errors and control signals were plotted. It can be clearly observed that the control signal supplied by the QP control is effectively constrained within the desired range, and also smoother than those obtained through the other compared controllers, which may potentially contribute to improve the passenger comfort and increase the life cycle of actuators.

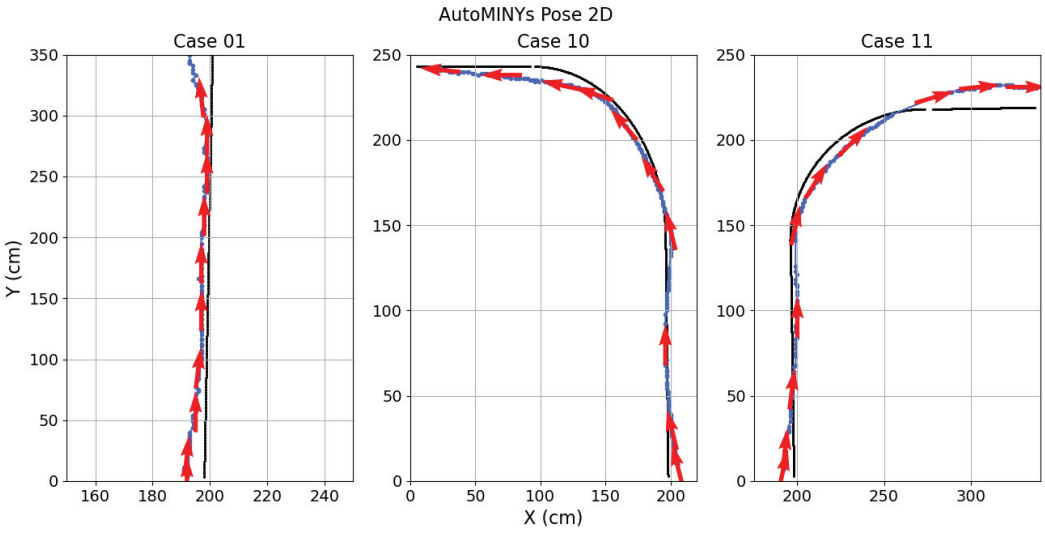


Figure 16. The trajectory of the car measured by the Ground-Truth system. Each path was labeled as 01, 10, and 11. Black line is the desired trajectory. Blue line is the average of performed trajectories. Red arrows denote the orientation of the car and the sense of the motion.

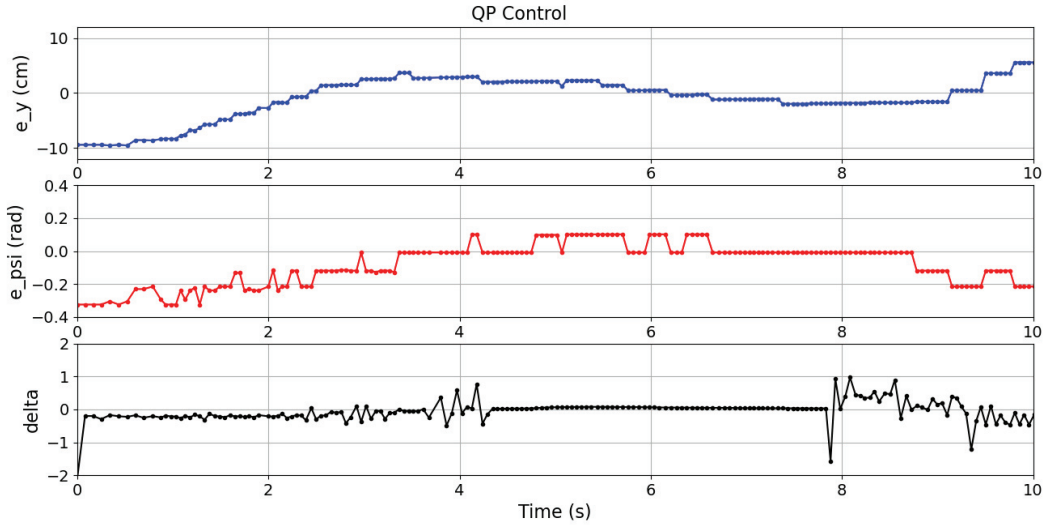


Figure 17. Time evolution of the lateral error e_y (blue), heading error e_ψ (red), and steering angle δ (black) during intersection navigation using the proposed QP-based control strategy.

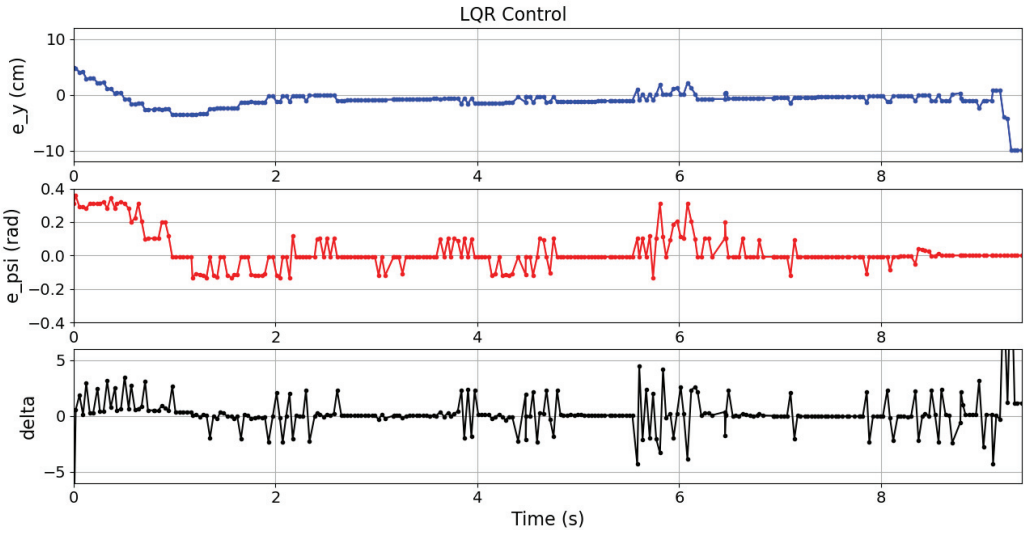


Figure 18. Time evolution of the lateral error e_y (blue), heading error e_ψ (red), and steering angle δ (black) during intersection navigation using the LQR control strategy.

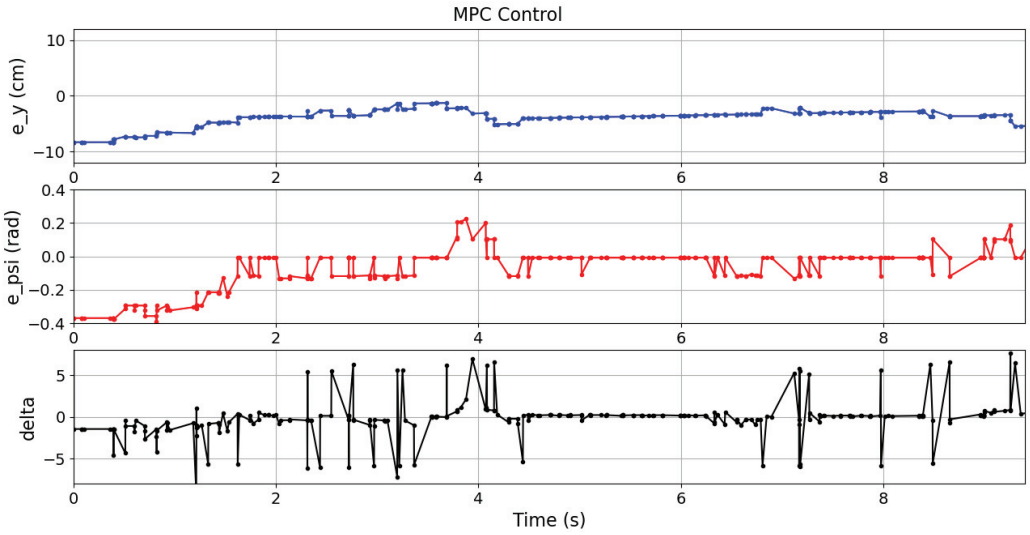


Figure 19. Time evolution of the lateral error e_y (blue), heading error e_ψ (red), and steering angle δ (black) during intersection navigation using the MPC strategy.

Table 3. RMSE, ISE, and TCE mean of the QP, LQR and MPC signals. The lowest values for each case are highlighted in bold.

Controller	Case	$\overline{\text{RMSE}}e_y$	$\overline{\text{ISE}}e_\psi$	$\overline{\text{TCE}}_\delta$
QP	01	6.0422	**2.7277**	**2.4320**
	10	10.0003	**4.9394**	**8.2269**
	11	9.2055	**6.3084**	**6.9541**
	Avg.	8.4160	**4.6585**	**5.8710**
LQR with GT	01	**4.9287**	8.5903	11.0374
	10	**4.8819**	13.6381	17.0680
	11	**6.2626**	20.1073	19.9529
	Avg.	**5.3577**	14.1119	16.0194
MPC with GT	01	7.6805	4.4789	20.5100
	10	6.8492	8.1341	29.4932
	11	6.3466	17.6865	31.4997
	Avg.	6.9588	10.0998	27.1676

7. Conclusions

In this work, a Quadratic Programming (QP)-based controller was developed and evaluated for the autonomous navigation of a scale-size autonomous vehicle in intersection scenarios. The execution of the experiments was performed using computer vision techniques, namely the computation of the dynamic states by means of an homography-based estimation. Lane markings were also segmented, enabling accurate measurement of the vehicle's lateral and heading errors.

A ground-truth system was implemented using a ceiling-mounted camera with colored markers, which proved to be highly useful for validating and quantifying the performance of the controllers. Additionally, a Convolutional Neural Network (CNN) was used to detect whether the vehicle was inside or outside an intersection with high accuracy, facilitating the correct switching between straight driving and intersection navigation controllers.

The proposed QP control was validated both mathematically via a Lyapunov method analysis and also experimentally in a real scale-model vehicle. The experimental results showed that the QP controller, which relied solely on on-board sensor information, achieved a competitive performance for lateral lane tracking compared to the LQR and MPC controllers that used ground-truth data. However, the proposed QP controller is significantly more accurate for curve tracking, demonstrating an average 67.0% lower ISE compared to the LQR and an average 53.9% lower ISE compared to the MPC. Moreover, when considering the TCE, the QP controller proved to be significantly more energy-efficient, reducing the required effort by an average of 63.3% compared to the LQR and by an average of 78.4% compared to the MPC, suggesting a less energy consumption and smoother operation, which can be potentially translated into a comfortable passenger experience, actuators preservation, and gentle maneuvers. In addition, it was proved that the control signals supplied by the QP control are correctly constrained to the selected range in contrast to the other compared techniques. These findings validate the feasibility of combining vision-based techniques with QP control for autonomous vehicle applications in controlled environments.

On the one hand, one potential limitation of the proposed controller is that it is designed to work efficiently in flat-surface navigation, so that roll and pitch motions can be negligible. On the other hand, the proposed model excels for scale-sized vehicles, but a more complex model must be implemented for real-sized ones.

As future work, the control system could be extended to more complex scenarios and include additional sensors to improve the controller's robustness and adaptability. Additionally, the visual perception system can be improved by detecting the lanes and other signals trough semantic segmentation with CNN, which is beneficial to discard disturbing objects on the road, for instance, splashed liquids, garbage, or hail.

Author Contributions: Conceptualization, E.E.M.M., O.G.M. and S.M.O.S.; methodology, E.E.M.M., O.G.M. and S.M.O.S.; software, E.E.M.M., O.G.M. and X.R.S.; validation, E.E.M.M., O.G.M. and X.R.S.; formal analysis, E.E.M.M., O.G.M. and J.M.I.Z.; investigation, E.E.M.M., O.G.M. and J.M.I.Z.; resources, J.M.I.Z.; data curation, E.E.M.M., O.G.M. and X.R.S.; writing—original draft preparation, all; writing—review and editing, all; visualization, E.E.M.M., O.G.M. and X.R.S.; supervision, O.G.M., S.M.O.S. and J.M.I.Z.; project administration, O.G.M. and J.M.I.Z.; funding acquisition, J.M.I.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available upon request from the author.

Acknowledgments: The authors would like to thank the Ministry of Science, Humanities, Technology and Innovation (SECIHTI) for the support given to the National Researcher with CVU No. 319148.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Bichiou, Y.; Rakha, H.A. Developing an optimal intersection control system for automated connected vehicles. *IEEE Trans. Intell. Transp. Syst.* **2018**, *20*, 1908–1916. [CrossRef]
2. Li, S.; Shu, K.; Chen, C.; Cao, D. Planning and decision-making for connected autonomous vehicles at road intersections: A review. *Chin. J. Mech. Eng.* **2021**, *34*, 133. [CrossRef]
3. Zhong, Z.; Nejad, M.; Lee, E.E. Autonomous and semiautonomous intersection management: A survey. *IEEE Intell. Transp. Syst. Mag.* **2020**, *13*, 53–70. [CrossRef]
4. Bouyarmane, K.; Chappellet, K.; Vaillant, J.; Kheddar, A. Quadratic programming for multirobot and task-space force control. *IEEE Trans. Robot.* **2018**, *35*, 64–77. [CrossRef]
5. Alonso, J.; Milanés, V.; Pérez, J.; Onieva, E.; González, C.; de Pedro, T. Autonomous vehicle control systems for safe crossroads. *Transp. Res. Part C Emerg. Technol.* **2011**, *19*, 1095–1110. [CrossRef]
6. Riegger, L.; Carlander, M.; Lidander, N.; Murgovski, N.; Sjöberg, J. Centralized MPC for Autonomous Intersection Crossing. In Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, 1–4 November 2016; pp. 1372–1377. [CrossRef]
7. Meng, X.; Cassandras, C.G. Optimal Control of Autonomous Vehicles for Non-Stop Signalized Intersection Crossing. In Proceedings of the 2018 IEEE Conference on Decision and Control (CDC), Miami, FL, USA, 17–19 December 2018; pp. 6988–6993. [CrossRef]
8. Ding, Z.; Sun, C.; Zhou, M.; Liu, Z.; Wu, C. Intersection vehicle turning control for fully autonomous driving scenarios. *Sensors* **2021**, *21*, 3995. [CrossRef] [PubMed]
9. Vitale, C.; Kolios, P.; Ellinas, G. Autonomous intersection crossing with vehicle location uncertainty. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 17546–17561. [CrossRef]
10. Morimoto, E.; Suguri, M.; Umeda, M. Vision-based Navigation System for Autonomous Transportation Vehicle. *Precis. Agric.* **2005**, *6*, 239–254. [CrossRef]
11. Sezer, V.; Bandyopadhyay, T.; Rus, D.; Frazzoli, E.; Hsu, D. Towards autonomous navigation of unsignalized intersections under uncertainty of human driver intent. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 3578–3585. [CrossRef]
12. Barrera-Ramírez, C.; González-Miranda, O.; Ibarra-Zannatha, J.M. Sistema de planeación y control de navegación para un vehículo autónomo en un entorno urbano. *Pädi Boletín Científico de Ciencias Básicas e Ingenierías del ICBI* **2022**, *10*, 27–35. [CrossRef]
13. Pourjafari, N.; Ghafari, A.; Ghaffari, A. Navigating unsignalized intersections: A predictive approach for safe and cautious autonomous driving. *IEEE Trans. Intell. Veh.* **2023**, *9*, 269–278. [CrossRef]
14. Zannatha, J.M.I.; Miranda, O.G.; Ramírez, C.B.; Miranda, L.A.L.; Aguilar, S.R.A.; Osuna, L.Á.D. Integration of perception, planning and control in the autominy 4.0. In Proceedings of the 2022 19th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE), Mexico City, Mexico, 9–11 November 2022; pp. 1–6. [CrossRef]
15. Alcalá, E.; Puig, V.; Quevedo, J.; Escobet, T.; Comasolivas, R. Autonomous vehicle control using a kinematic Lyapunov-based technique with LQR-LMI tuning. *Control Eng. Pract.* **2018**, *73*, 1–12. [CrossRef]
16. Hu, J.; Xiong, S.; Zha, J.; Fu, C. Lane Detection and Trajectory Tracking Control of Autonomous Vehicle Based on Model Predictive Control. *Int. J. Automot. Technol.* **2020**, *21*, 285–295. [CrossRef]
17. Yuan, T.; Zhao, R. LQR-MPC-Based Trajectory-Tracking Controller of Autonomous Vehicle Subject to Coupling Effects and Driving State Uncertainties. *Sensors* **2022**, *22*, 5556. [CrossRef] [PubMed]
18. Bousskoul, A.E.; Ouachtouk, I.; Ait Elmahjoub, A. Path Tracking for Self-driving Cars: Stanley and LQR Controllers Comparison. In *Innovative Technologies on Electrical Power Systems for Smart Cities Infrastructure (ICESST), Proceedings of the 1st International Conference on Electrical Systems and Smart Technologies, Dakhla, Morocco, 11–13 December 2024*; Sustainable Civil Infrastructures; Springer: Cham, Switzerland, 2025; pp. 337–346. [CrossRef]
19. Lee, D.H. Lane-Keeping Control of Autonomous Vehicles Through a Soft-Constrained Iterative LQR. *IEEE Trans. Veh. Technol.* **2025**, *74*, 5610–5623. [CrossRef]
20. Rajamani, R. *Vehicle Dynamics and Control*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2011.
21. Kulkarni, A.J. *Optimization Methods and Algorithms*; Springer: Singapore, 2025.

22. Hao, T.; Wen, J.; Xiong, J. Maximum Principle of Stochastic Optimal Control Problems with Model Uncertainty. *J. Optim. Theory Appl.* **2025**, *207*, 27. [CrossRef]
23. Abdalla, M.; Alsafartee, J.H. Optimized Model Reduction Using Reducibility Matrix. In Proceedings of the 2025 5th International Conference on Artificial Intelligence and Industrial Technology Applications (AIITA), Xi'an, China, 28–30 March 2025; pp. 1285–1289.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

actuators

MDPI

Article

Trajectory Control for Car-like Mobile Robots via Frugal Predictive Control with Integrated Disturbance Rejection

Luis Angel Martínez-Ramírez , Rafael Isaac Vásquez-Cruz, German Ardul Munoz-Hernandez, Gerardo Mino-Aguilar, Wuiyevaldo Fermín Guerrero-Sánchez , Roberto Carlos Ambrosio-Lázaro and José Fermi Guerrero-Castellanos *

Faculty of Electronics Sciences, Benemérita Universidad Autónoma de Puebla (BUAP), Cd Universitaria, Puebla 72570, Mexico; luis.martinezrmz@correo.buap.mx (L.A.M.-R.); vc225570356@alm.buap.mx (R.I.V.-C.); germanardul.munoz@correo.buap.mx (G.A.M.-H.); gerardo.mino@correo.buap.mx (G.M.-A.); willi@fcfm.buap.mx (W.F.G.-S.); roberto.ambrosio@correo.buap.mx (R.C.A.L.)
* Correspondence: fermi.guerrero@correo.buap.mx

Abstract

This paper presents a hierarchical control architecture for high-precision trajectory tracking of a car-like mobile robot (CLMB) operating under external disturbances arising from normal and tangential wheel forces. The proposed solution addresses the critical challenge of simultaneously rejecting disturbances and accurately following a predefined path at a determined cruise velocity. Since the vehicle is equipped with an electronic differential at the low level, a nonlinear dynamic control (NDC) scheme is implemented to regulate the speed in each wheel. This controller actively estimates and compensates for differential traction losses and other lumped disturbances in real time, ensuring robust wheel velocity tracking across varying terrain conditions. The compensated system is then governed by a high-level frugal model predictive controller (FMPC) that leverages a dynamic vehicle model to compute optimal steering and velocity commands, thereby minimizing future trajectory-tracking errors. To achieve a precise and reliable state estimation necessary for feedback control, an Extended Kalman Filter (EKF) is designed to fuse high-frequency data from wheel encoders with absolute pose measurements from a motion capture system, mitigating the drift inherent in odometry alone. Experimental results on a physical robotic platform demonstrate tracking accuracy and robust disturbance rejection under different operating conditions.

Keywords: model predictive control; Extended Kalman Filter; nonlinear dynamic control; bicycle kinematic model; electronic differential

1. Introduction

1.1. Preliminaries

Mobile robotics has undergone an unprecedented evolution, driven by the need to automate tasks in sectors as diverse as logistics, precision agriculture, and autonomous transportation [1–4]. However, achieving full autonomy remains an open challenge, constrained by the vehicles' ability to perceive, localize, and navigate safely in dynamic and uncertain environments [5,6]. This challenge extends beyond general mobile robotics into the automotive sector, specifically regarding autonomous electric vehicles [7–9]. In this context, vehicles with car-like kinematic configurations represent a critical case study due to their widespread adoption in industry and their non-holonomic constraints, which introduce additional complexity to the planning and execution of maneuvers [10–12].

Since its initial formulations, the problem of optimal estimation in dynamic systems has been addressed through recursive models that combine sensor data with system predictions. The seminal work of Kalman [13] established the foundation for the filter that bears his name, offering an optimal solution for linear systems with Gaussian noise, characterized by its computational efficiency and recursive nature, Subsequently, this concept has been extended to accommodate the complexities of nonlinear systems [14], this is fundamental for modern robotics where sensor models and vehicle kinematics rarely follow linear assumptions. The recent literature identifies that robust navigation in complex scenarios—where global positioning signals are often unreliable—demands the integration of multisensory architectures capable of mitigating uncertainty [2,6,10]. In this field, sensor data fusion through the Kalman Filter (KF) and its extensions for nonlinear systems, such as the Extended Kalman Filter (EKF), has consolidated as the de facto standard for ensuring consistent state estimation and precise obstacle avoidance [1,3]. Current studies demonstrate that the combination of trilateral localization techniques with advanced filtering allows mobile robots to operate successfully even in chaotic and semi-open environments, laying the groundwork for next-generation industrial and service applications [15]. Alternatively, for outdoor environments, Ref. [16] presents a method to combine Global Navigation Satellite System (GNSS) and BeiDou Navigation Satellite System (BeiDou) data with a low-cost Inertial Measurement Unit (IMU) sensor using an EKF to enhance positioning accuracy. Integrating multiple GNSS systems with a Micro–Electro-Mechanical System Inertial Measurement Unit (MEMS IMU) facilitates achieving centimeter-level precision, which is essential for precision agriculture, autonomous vehicles, and surveying. Recently, the literature has explored the use of deep learning techniques, such as Recurrent Neural Networks (RNNs), to model complex dynamic uncertainties that linear filters fail to capture [17]; however, Wang et al. [18] present a comparative study demonstrating that the EKF maintains a critical superiority in terms of computational efficiency and predictability for real-time vehicle control. In contrast, AI-based approaches [19] often require significant computational load and large volumes of training data.

Once the positioning problem has been resolved, the next challenge consists of precise trajectory tracking. The robot is subject to non-holonomic constraints, and the robotic system is nonlinear and underactuated. These characteristics increase the control design challenge [20]. Various control strategies have been developed to solve this problem, ranging from classical controllers and geometric algorithms to optimal and robust controllers, and, recently, the incorporation of artificial intelligence [21]. In [22,23], control structures based on PID are proposed to achieve trajectory tracking; however, the optimality of the control signal is a factor to consider. In [24], an LQR controller is proposed that better adapts to the lateral tracking control problem of intelligent vehicles under complex operating conditions. The robustness of the control against external perturbations or unmodeled dynamics is crucial in practical implementations; based on that assumption, Ref. [25] presents a backstepping second-order sliding mode control so that the kinematic error states, after a nonlinear transformation, tend to zero in finite time. Given the difficulty of establishing a precise vehicle model, the perturbations to which it may be subjected, and the dynamics it undergoes, control alternatives based on observers [26–30] and the use of AI [31,32]) have emerged. Recently, Model Predictive Control (MPC) has emerged as a robust solution thanks to its ability to anticipate future states and explicitly manage state and input constraints [33]. Various hybrid control schemes based on MPC have been developed [34–36] to address deficiencies in the MPC model. Nevertheless, a trade-off exists between the use of nonlinear models (NMPC), which offer a more realistic representation but incur high computational cost and optimization complexity, and linearized models, which fa-

https://doi.org/10.3390/act15050260

cilitate real-time implementation and guaranty more efficient convergence in practical applications [37].

To optimize the implementation of Model Predictive Control (MPC), this work utilizes the concept of feedback linearization, which involves a change in the coordinates applied to the kinematic model of a car-like robot. This approach has been previously explored in the development of various control strategies [38–42]. By employing this method, the system can be simplified to a structure consisting of two simple integrators for the trajectory coordinates in the plane [12,43,44]. This decoupled model enables the MPC optimizer to address the control problem [45]. Subsequently, the calculated control signals are transformed into the vehicle's physical variables using the inverse coordinate transformation, ensuring correspondence between kinematic prediction and execution.

Translating trajectory control signals into traction system involves careful consideration of the vehicle's mechanics, such as whether it is two-wheel or four-wheel drive [46]. For example, rear-wheel differential systems [47] use Ackermann–Jeantand geometry to determine the reference velocities for each motor. This electronic differential maintains a consistent velocity difference between the inner and outer wheels during turns, helping prevent slippage and maintaining motion integrity. In terms of alternative methods, Ref. [48] proposes a mechanical system of coupled pedals that enables independent control of wheel speed. In [49], a differential torque control method is proposed; however, the complexity of modeling both the vehicle and the motor remains a significant challenge. To tackle this issue, the work presented in [50] introduces a control law that focuses on regulation error. This approach utilizes low-pass and high-pass filtering of the control signal to enhance performance.

Effective displacement towards target setpoints within a trajectory-tracking framework is achieved by coordinating steering control with the regulation of traction motor angular velocity. To improve system performance in the face of environmental uncertainties, a disturbance estimator can be integrated into the control law [51–55]. This integrated approach enhances robustness, ensuring stable and precise trajectory tracking even under challenging real-world operating conditions.

The experimental phase and validation of theoretical results are crucial in research. The engineering challenge involves developing and effectively integrating systems to close the gap between theory and practice. In the field of mobile robotics, Ref. [56] outlines the fabrication process for various mobile robots platforms. Additionally, Ref. [57] presents an integrated framework for indoor localization along with experimental validation using an autonomous ADAS-based model car. This vehicle determines its global position by merging data from fiducial markers, inertial sensors, and wheel odometry. Moreover, Ref. [58] introduces a testbed designed to evaluate control strategies under outdoor conditions, offering a cost-efficient and repeatable alternative to full-scale vehicle testing.

1.2. Contribution

The primary contribution of this work is the solution's frugality, which transforms a complex problem—specifically, the trajectory tracking of a car-like mobile robot equipped with an electronic differential—into a modular system incorporating a multi-level hierarchical control scheme. This approach presents a well-considered reformulation that enables the resolution of a nonlinear tracking problem while maintaining the computational efficiency of the quadratic programming (QP) problem associated with linear Model Predictive Control (MPC). Additionally, to decrease the number of decision variables in the QP-problem, a control parameterization method is utilized. This choice decreases the complexity of the optimization problem without compromising the performance of the closed-loop system.

https://doi.org/10.3390/act15050260

The following list summarizes our contributions:

- **Hierarchical integration:** We present a modular framework that combines an Extended Kalman Filter (EKF), a Frugal Model Predictive Control (FMPC), and an electronic differential. In this framework, the velocity of each wheel is independently regulated by a nonlinear dynamic control law capable of addressing model uncertainties and external disturbances.
- **Nonlinear-to-linear structure via coordinate transformation:** Although the kinematics of the car-like mobile robot are governed by a nonlinear model with non-holonomic constraints, we propose controlling the position of a virtual point ("nose") located at a distance p along the line that is normal to the wheel axis. The kinematics associated with the "nose" position are holonomic, so we can apply a coordinate transformation that effectively decouples the system into two first-order integrators. Controlling the nose position is practically relevant, as this location can accommodate a suite of sensors and a camera.
- **MPC based on an integrator mode:** Unlike conventional nonlinear MPC (NMPC) the predictive model used in the present work consists of two pure integrators. This transformation leads to a quadratic programming (QP) problem with minimal computational burden, retaining the advantages of linear MPC (e.g., convexity and efficiency) while handling the nonlinear system.
- **Control parametrization:** To reduce the number of decision variables in the quadratic programming (QP) problem, a reduced control parametrization is adopted. With such a choice, the optimization problem's complexity is reduced without compromising the closed loop's performance.
- **Experimental validation:** To experimentally validate the proposed strategy, a simple yet complete experimental platform is developed using: (1) open and standardized communication protocols, (2) a on-board embedded system to drive wheel angular velocities and direcction, and (3) a MATLAB-ROS (Robot Operating System) ground station.
- **Realiability analysis:** The reliability of the control strategy was assessed by evaluating its ability to track the desired trajectory, reject disturbances, and meet constraints related to both the state and control. Additionally, we examined the capability of the Extended Kalman Filter (EKF) to integrate relative measurements (odometry) and absolute measurements (position data obtained from motion capture) at various sampling times. Performance was quantified using three well-established indices: the Integral of the Squared Error (ISE), the Integral of the Absolute Error (IAE), and the Integral of Time-weighted Absolute Error (ITAE). The results highlight the effectiveness, reliability, and significance of the proposed strategy.

The remainder of this paper is organized as follows: Section 2 presents the mathematical preliminaries and the problem statement. Section 3 focuses on the control design and estimation design. Sections 4–6 describe the development of the experimental platform and its main components, as well as the corresponding results. Finally, Section 7 presents the conclusions.

2. Mathematical Preliminaries and Problem Statement

2.1. Car-like Mobile Robot

Consider a car-like mobile robot as shown in Figure 1. It is important to note the reduction in the geometric representation from four to two wheels (bicycle model): one for traction (rear wheel) and the other for direction (front wheel). To establish the mathematical

https://doi.org/10.3390/act15050260

equations for the kinematic model, the reference point for vehicle position is located at the midpoint of the rear axle. Finally, the kinematic bicycle model is defined as follows:

$$\Sigma_q := \begin{cases} \dot{x} = v_x \cos(\theta) \\ \dot{y} = v_x \sin(\theta) \\ \dot{\theta} = \dfrac{v_x}{\bar{L}} \tan(\delta) \end{cases} \tag{1}$$

where v_x is the longitudinal velocity of the vehicle, δ is the steering angle, and $\bar{L}$ corresponds to the wheelbase. Furthermore, let us define the angular velocity of the frame $\mathbf{E}^b$ with respect to $\mathbf{E}^f$ by $\omega := \dot{\theta}$.

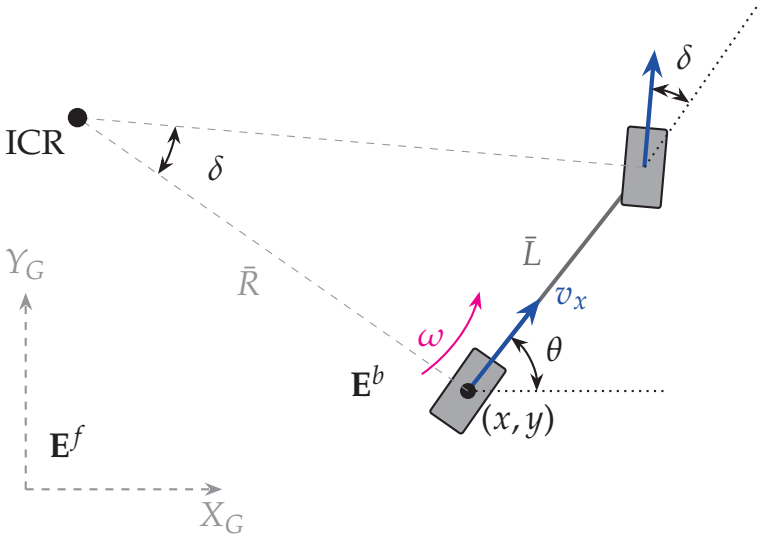


Figure 1. Kinematic bicycle model for car-like mobile robot.

In trajectory-tracking control for a car-like mobile robot using the kinematics model, it is well known that the system's control signals are the longitudinal velocity v_x and the steering angle δ. From Figure 1 the front wheel angle δ can be defined as a function of v_x and ω (vehicle's angular velocity) in the form

$$\delta = \tan^{-1}\left(\frac{\omega \bar{L}}{v_x}\right) \tag{2}$$

In this work, the mobile robot configuration uses Ackermann steering and differential traction on the rear axle, i.e., the left and right wheels are coupled to independent electric motors. Let us define the notation for the angular velocity of each wheel as

$$\omega_i, \text{ with } i \in \{L, R\} \tag{3}$$

Using this traction configuration, each wheel can rotate at a different speed without a mechanical differential; additionally, the tire torque can be controlled independently, providing greater stability during various maneuvers and under different road conditions. The longitudinal displacement velocity of the mobile robot is established as a function of the angular velocities of the wheels:

$$v_x = f(\omega_L, \omega_R) = \frac{r}{2}(\omega_L + \omega_R) \tag{4}$$

where r denotes the radius of the wheels of the mobile robot.

https://doi.org/10.3390/act15050260

2.2. Electronic Differential Based on Ackermann Geometry

Based on the Ackermann steering geometry, we can derive the angular velocities of the inner and outer wheels for curved trajectories. This kinematic model is designed for low-speed operation and assumes that lateral forces and longitudinal tire slip can be neglected. As illustrated in Figure 2, the model relates the steering angles (δ_L and δ_R) to the radius of rotation ($\bar{R}$). This relationship allows us to determine the appropriate angular speeds (ω_L and ω_R) for the drive wheels.

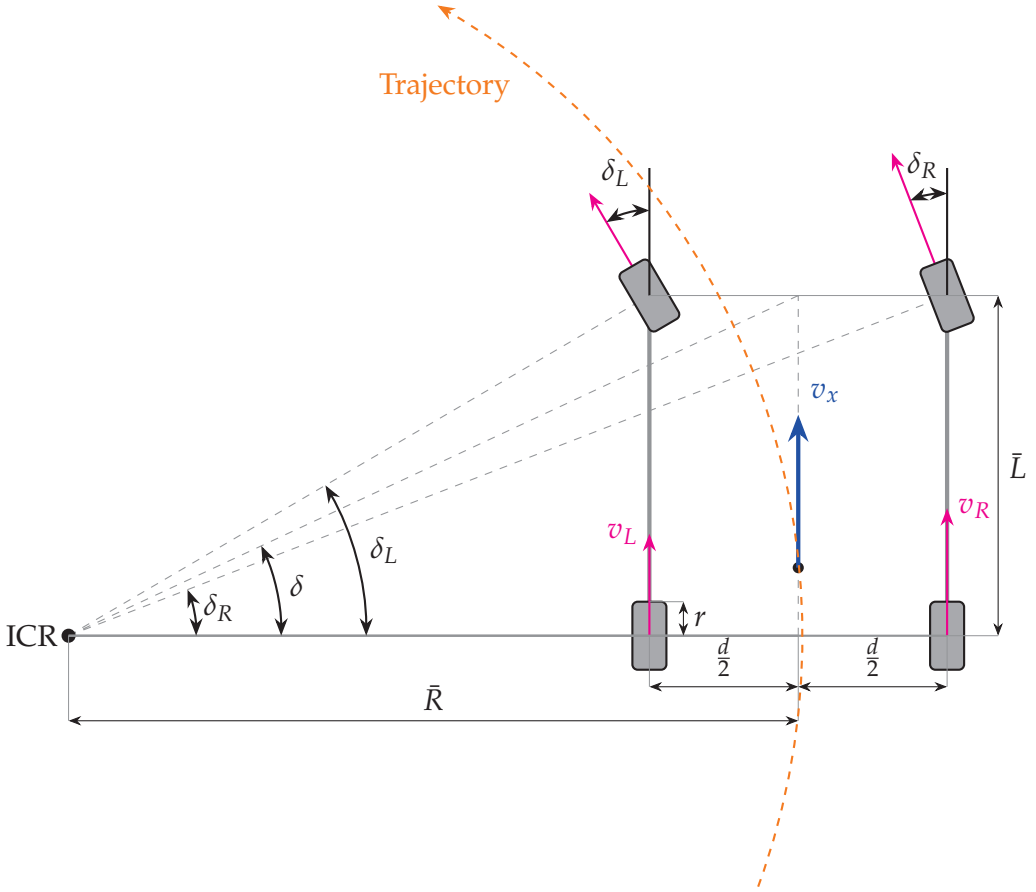


Figure 2. Ackermann geometry with electronic differential [59].

Steering angles can be analytically expressed as functions of the turning radius $\bar{R}$ and the wheelbase $\bar{L}$ by means of vehicle kinematic constraints through the following trigonometric relationships:

$$\delta_L = \tan^{-1}\left(\frac{\bar{L}}{\bar{R} + \frac{d}{2}}\right), \quad \delta_R = \tan^{-1}\left(\frac{\bar{L}}{\bar{R} - \frac{d}{2}}\right) \tag{5}$$

where d corresponds to the width of the track. Under the small-angle approximation, the left and right steering angles can be represented by a single nominal parameter, hereafter referred to as the average Ackermann steering angle (δ):

$$\delta = \frac{\delta_L + \delta_R}{2}, \qquad \tan(\delta) = \frac{\bar{L}}{\bar{R}} \tag{6}$$

The angular velocity of each drive wheel is defined as a function of the vehicle's longitudinal velocity v_x and the Ackermann steering angle δ. This mathematical relationship

https://doi.org/10.3390/act15050260

enables the implementation of an electronic differential, a control strategy that provides independent speed regulation for the rear traction wheels to ensure coordinated cornering.

$$\omega_L = \frac{v_x}{r}\left(1 - \left(\frac{d}{2}\frac{\tan(\delta)}{\bar{L}}\right)\right) \tag{7}$$

$$\omega_R = \frac{v_x}{r}\left(1 + \left(\frac{d}{2}\frac{\tan(\delta)}{\bar{L}}\right)\right) \tag{8}$$

The calculated angular velocities, ω_L and ω_R, serve as reference setpoints for low-level motor controllers. This coordinated control architecture is essential for preventing tire scrub and ensuring the stability of the vehicle's trajectory before the state estimates are refined by the Extended Kalman Filter.

Assumption 1. *In this work, the geometric parameters of the wheelbase $\bar{L}$, wheel radius r, and track width d are considered known and constant.*

2.3. Frugal Model Predictive Control (FMPC)

The FMPC is a specialized MPC framework designed for real-time execution on platforms with limited on-board computational power. Unlike traditional MPC, which can be computationally expensive due to high-dimensional optimization, FMPC utilizes a "frugal" design flow that combines simplified internal models with aggressive control parametrization to drastically reduce the number of decision variables in the underlying optimization problem [60,61].

2.3.1. Core Methodology

At each sampling instant $k \in \mathbb{N}$, the FMPC strategy computes the control sequence

$$\tilde{\mathbf{u}}(k) \triangleq \left[\mathbf{u}_p^T(k),\ \mathbf{u}_p^T(k{+}1),\ \ldots,\ \mathbf{u}_p^T(k{+}N{-}1)\right]^T \in \mathbb{R}^{Nn_u}, \tag{9}$$

that minimizes a quadratic cost over a finite prediction horizon $N \in \mathbb{N}_+$. The internal predictor is a discrete-time linear integrator-chain model of the car-like mobile robot, decoupled along its translational degrees of freedom. The cost is defined by

$$J\left(\tilde{\mathbf{u}}_p(k)\right) = \sum_{i=1}^{N} \|\mathbf{C}_r\,\mathbf{x}(k{+}i) - \mathbf{y}_d(k{+}i)\|_{\mathbf{R}_w}^2 + \sum_{i=0}^{N-1} \|\mathbf{u}_p(k{+}i) - \mathbf{u}_d\|_{\mathbf{Q}_w}^2, \tag{10}$$

where $\mathbf{x}(k{+}i) \in \mathbb{R}^n$ is the predicted state, $\mathbf{C}_r \in \mathbb{R}^{n_r \times n}$ selects the regulated outputs, $\mathbf{y}_d(k{+}i) \in \mathbb{R}^{n_r}$ is the reference trajectory, $\mathbf{u}_d \in \mathbb{R}^{n_u}$ is the steady-state input, and $\mathbf{R}_w \succ 0$ and $\mathbf{Q}_w \succ 0$ are the tracking-error and control-effort weighting matrices, respectively. The minimization of J is subject to:

- *Model dynamics*;
- *Actuator saturation*: $\mathbf{u}_p^{\min} \le \mathbf{u}_p(k{+}i) \le \mathbf{u}_p^{\max},\ i = 0, \ldots, N{-}1$;
- *Output safe set*: $\mathbf{y}^{\min} \le \mathbf{C}_r\,\mathbf{x}(k{+}i) \le \mathbf{y}^{\max},\ i = 1, \ldots, N$.

The pointwise saturation bounds $\mathbf{u}_p^{\min}, \mathbf{u}_p^{\max} \in \mathbb{R}^{n_u}$ and the safe-set bounds $\mathbf{y}^{\min}, \mathbf{y}^{\max} \in \mathbb{R}^{n_r}$ vary by the operating envelope of the situation.

2.3.2. Frugal Implementation via Control Parametrization

The online solution of the resulting Quadratic Optimization Problem (QOP) must complete within the available computation budget $T_c < T_s$, where T_s denotes the controller sampling period and T_c the maximum admissible solver time. To enforce this constraint, FMPC adopts a reduced control parametrization: rather than optimizing the $N \cdot n_u$ entries

https://doi.org/10.3390/act15050260

of $\tilde{\mathbf{u}}(k)$ independently, the entire sequence is expressed as an affine map of a compact decision vector $\mathbf{p}_f(k) \in \mathbb{R}^{n_f}$ through a parametrization matrix $\mathbf{\Pi}_f \in \mathbb{R}^{N \cdot n_u \times n_f}$,

$$\tilde{\mathbf{u}}(k) = \mathbf{\Pi}_f \, \mathbf{p}_f(k), \qquad n_{p_f} \ll N \cdot n_u.$$

Using a change of variable and the constraints transforms the original QOP into a reduced form of dimension n_f, which can be evaluated with a fraction of the original arithmetic cost while preserving tracking performance. As reported by Vásquez-Cruz et al. [61] for a quadrotor simulation, this reduction can exceed an order of magnitude—for instance, from $N \cdot n_u = 90$ to $n_q = 6$ decision variables—without measurable degradation of the closed-loop response.

2.3.3. Feedforward Integration

To minimize steady-state trajectory-tracking error, a feedforward term is incorporated into the control signals. This pre-compensation is derived directly from the desired trajectory dynamics, specifically from its derivative. This term is integrated into the FMPC cost function to ensure that the physical constraints of the actuator are respected despite the additional feedforward action. Consequently, the computational overhead of the controller is reduced, allowing the optimization process to focus primarily on position error correction while maintaining overall computational efficiency.

2.4. Kalman Filter

In the field of autonomous systems, pose estimation (cartesian coordinates and angular orientation) for ground-based robotic platforms represents a fundamental challenge in state estimation. The Extended Kalman Filter (EKF) provides a primary recursive mechanism for harmonizing the a priori state estimate derived from internal kinematic models (odometry) with a posteriori stochastic observations obtained from heterogeneous sensor system. For a car-like mobile robot [62], the state vector is typically defined as $\mathbf{q} \in \mathbb{R}^3$, where $\mathbf{q} = [x, y, \theta]^T$. The EKF propagates the state and its associated uncertainty by linearizing the system dynamics via a first-order Taylor series expansion. In general, the EKF consists of two stages: prediction and correction.

2.5. Problem Statement

As described above, the platform for this research is a car-like mobile robot with differential traction. The main objective is to develop a control strategy to solve the trajectory-tracking problem, ensuring the system follows a parameterized route $\mathbf{q}_d(t)$ over time while maintaining its dynamic constraints on the state and the magnitude of the control signals. Let $\mathbf{q}_p = (x_p(t)\ y_p(t))^T \in \mathbb{R}^2$ be the coordinates of the point P along the axis $\mathbf{X}_B$ in the body frame $\mathbf{E}^b$, but expressed in the inertial frame $\mathbf{E}^f$. Moreover, let $\mathbf{u}_p(t) = (u_x(t)\ u_y(t))^T \in \mathbb{R}^2$ be the velocity of $\mathbf{q}_p(t)$. Then, the control objective is to find $\mathbf{u}_p$ such that

$$\mathbf{q}_e(t) = \mathbf{q}_d(t) - \mathbf{q}_p(t) \to \mathbf{0} \ \text{ when } t \to \infty \tag{11}$$

subject to the following constraints:

$$\text{Position:} \qquad \mathbf{q}_p^{\min} \leq \mathbf{q}_p(t) \leq \mathbf{q}_p^{\max}, \quad \forall t > 0, \tag{12}$$

$$\text{Velocity:} \qquad \mathbf{u}_p^{\min} \leq \mathbf{u}_p(t) \leq \mathbf{u}_p^{\max}, \quad \forall t > 0, \tag{13}$$

$$\text{Velocity rate:} \quad \Delta\mathbf{u}_p^{\min} \leq \Delta\mathbf{u}_p(t) \leq \Delta\mathbf{u}_p^{\max}, \quad \forall t > 0. \tag{14}$$

Furthermore, the state of the system must be estimated from relative and absolute measurements using an Extended Kalman Filter.

https://doi.org/10.3390/act15050260

For that, one proposes the strategy depicted in Figure 3. The schema is composed of three main blocks:

1. A high-level control block is charged to generate the linear and angular velocity profiles while maintaining the specified constraints on the state and maximum velocities given by the user. The core of this stage is a Frugal Model Predictive Control.
2. A low-level dynamic control law is charged to achieve and maintain the desired angular velocity in each motor, which is dictated by the high-level control. This control law must reject the disturbances and uncertainties in the traction system model.
3. An Extended Kalman Filter fuses the odometry, sampled at Δt s, and the position measurement obtained by the motion capture system, sampled at $\gamma \Delta t$, with $\gamma \in \mathbb{N}_+$.

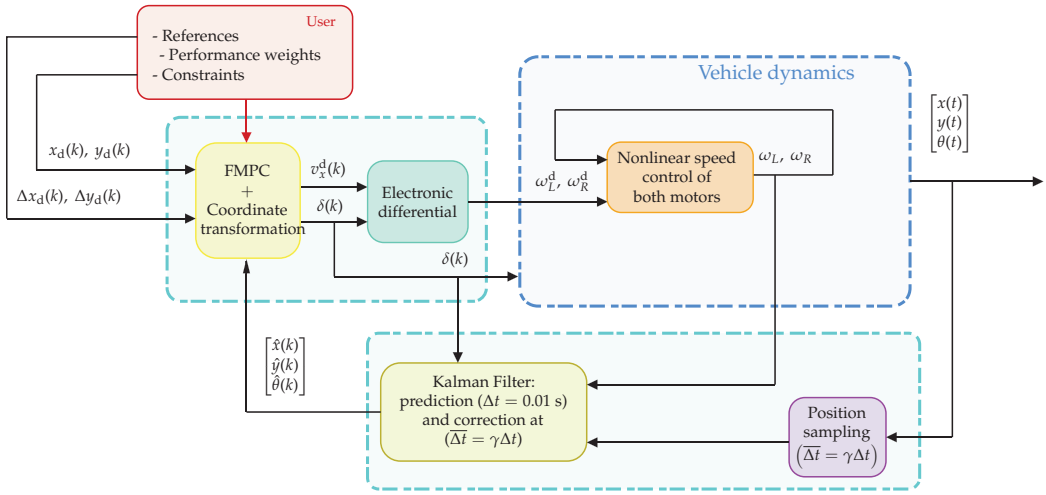


Figure 3. Proposed control architecture.

3. Control and State Estimation Strategy

In this section, each block of the general schema shown in Figure 3 is described in detail.

3.1. FMPC Applied to the Car-like Mobile Robot

The kinematic model of a car-like mobile robot defined in (1) is a nonlinear system of equations; movement of the body is described by trigonometric functions of the orientation angle θ. Besides, because the center of mass is set at the rear axle, the vehicle cannot have instantaneous lateral movement (non-holonomic restriction). To avoid this limitation, the reference point is moved p along the vehicle. The objective is to obtain a function that describes the relationship between the new reference point P and the state of the system $\mathbf{q} = [x\ y\ \theta]^T$. The coordinates of P with respect to the inertial frame $\mathbf{E}^f$ are given by

$$\mathbf{q}_p = \begin{bmatrix} x_p \\ y_p \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + p \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix} \tag{15}$$

the new point depends on the orientation θ. To determine how that point moves over time, the new position expressions are differentiated. From Figure 4, the displacement of the new point P on the inertial frame is described by

$$\dot{\mathbf{q}}_p(t) = \begin{bmatrix} \dot{x}_p(t) \\ \dot{y}_p(t) \end{bmatrix} = \underbrace{\begin{bmatrix} \cos(\theta(t)) & -p\sin(\theta(t)) \\ \sin(\theta(t)) & p\cos(\theta(t)) \end{bmatrix}}_{\mathbf{J}_p} \begin{bmatrix} v_x(t) \\ \omega(t) \end{bmatrix} \tag{16}$$

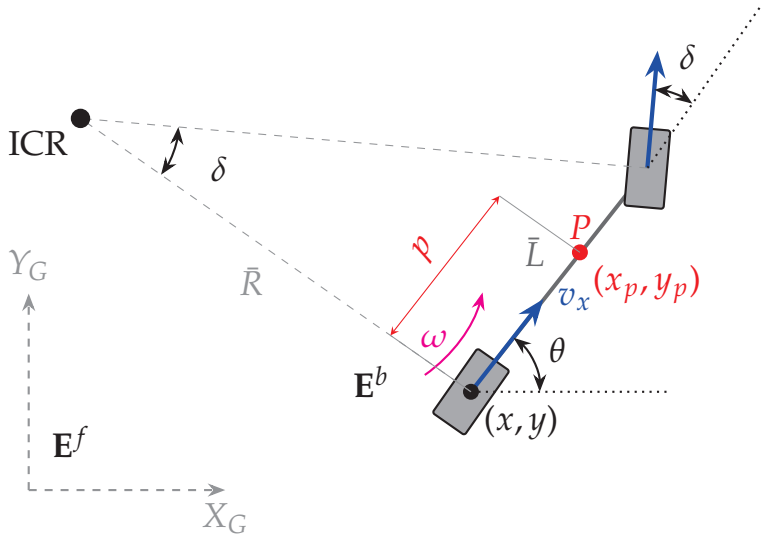


Figure 4. Reference point displaced p distance from rear wheel for feedback linearization.

Let us define the following coordinate change:

$$\begin{bmatrix} v_x(t) \\ \omega(t) \end{bmatrix} = \underbrace{\begin{bmatrix} \cos(\theta(t)) & \sin(\theta(t)) \\ -\frac{1}{p}\sin(\theta(t)) & \frac{1}{p}\cos(\theta(t)) \end{bmatrix}}_{\mathbf{J}_p^{-1}} \underbrace{\begin{bmatrix} u_x(t) \\ u_y(t) \end{bmatrix}}_{\mathbf{u}_p} \tag{17}$$

where $\mathbf{u}_p = [u_x \; u_y]^T$ is a control vector that will later be designed to track the reference position. Note that $\mathbf{u}_p$ represents the linear velocity of the point P expressed in the inertial frame. Using (17) in (16), gives

$$\Sigma_{\dot{q}_p} := \begin{cases} \dot{x}_p(t) = u_x(t) \\ \dot{y}_p(t) = u_y(t) \end{cases} \tag{18}$$

This representation provides a decoupled framework that facilitates the application of linear control techniques while maintaining the essential fidelity of the original kinematic model.

Assumption 2. *The distance p of the virtual point, called "nose," is constant.*

3.1.1. Plant Model for the FMPC

Define the state and control vectors as

$$\mathbf{x}(t) = \begin{bmatrix} x_p(t) & y_p(t) \end{bmatrix}^\top \in \mathbb{R}^2, \qquad \mathbf{u}_p(t) = \begin{bmatrix} u_x(t) & u_y(t) \end{bmatrix}^\top \in \mathbb{R}^2, \tag{19}$$

The plant (18) is then rewritten in continuous-time state-space form as

$$\dot{\mathbf{x}}(t) = \mathbf{A}_c\, \mathbf{x}(t) + \mathbf{B}_c\, \mathbf{u}_P(t), \tag{20}$$

with

$$\mathbf{A}_c = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \qquad \mathbf{B}_c = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{21}$$

Note that the pair $(\mathbf{A}_c, \mathbf{B}_c)$ is controllable and the system (20) has the structure on which the FMPC design is based [60].

https://doi.org/10.3390/act15050260

3.1.2. Discretization by Zero-Order Hold

Let $T_s > 0$ be the sampling period. Applying a zero-order hold (ZOH) discretisation scheme to (20) yields the discrete-time model

$$\mathbf{x}(k+1) = \mathbf{A}\,\mathbf{x}(k) + \mathbf{B}\,\mathbf{u}_P(k), \tag{22}$$

where the discrete matrices are

$$\mathbf{A} = e^{\mathbf{A}_c T_s} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad \mathbf{B} = \int_0^{T_s} e^{\mathbf{A}_c \tau}\,\mathrm{d}\tau\ \mathbf{B}_c = \begin{bmatrix} T_s & 0 \\ 0 & T_s \end{bmatrix}. \tag{23}$$

3.1.3. Predictive Scheme

Starting from the model (22), the N-step-ahead state prediction is constructed following the procedure in [60,63]. Concatenating predictions for $i = 1, \ldots, N$ yields the compact form

$$\tilde{\mathbf{x}}(k) = \mathbf{\Phi}\,\mathbf{x}(k) + \mathbf{\Psi}\,\tilde{\mathbf{u}}(k), \tag{24}$$

where $\tilde{\mathbf{x}}(k) \triangleq \left(\mathbf{x}(k+1)^\top, \ldots, \mathbf{x}(k+N)^\top\right)^\top$ and $\tilde{\mathbf{u}}(k) \triangleq \left(\mathbf{u}_p(k)^\top, \ldots, \mathbf{u}_p(k+N-1)^\top\right)^\top \in \mathbb{R}^{2N}$ are the stacked state and input sequences over the horizon, and

$$\mathbf{\Phi} \triangleq \begin{bmatrix} \mathbf{A} \\ \mathbf{A}^2 \\ \vdots \\ \mathbf{A}^N \end{bmatrix}, \qquad \mathbf{\Psi} \triangleq \begin{bmatrix} \sum_{j=1}^{1} \mathbf{A}^{1-j}\mathbf{B}\,\mathbf{\Pi}_j^{(2,N)} \\ \sum_{j=1}^{2} \mathbf{A}^{2-j}\mathbf{B}\,\mathbf{\Pi}_j^{(2,N)} \\ \vdots \\ \sum_{j=1}^{N} \mathbf{A}^{N-j}\mathbf{B}\,\mathbf{\Pi}_j^{(2,N)} \end{bmatrix}, \tag{25}$$

with $\mathbf{\Pi}_j^{(2,N)} \in \mathbb{R}^{2\times 2N}$ being the block-selection matrix that extracts the j-th input from $\tilde{\mathbf{u}}(k)$, i.e., $\mathbf{\Pi}_j^{(2,N)}\tilde{\mathbf{u}}(k) = \mathbf{u}_p(k+j-1)$.

The regulated and constrained outputs are the Cartesian positions x_p and y_p, selected by means of the following:

$$\mathbf{C}_r = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{2\times 2}, \quad \mathbf{C}_c = \begin{bmatrix} 1 & 0 \end{bmatrix} \in \mathbb{R}^2 \tag{26}$$

The quadratic cost function penalizing position-tracking error and control effort over the horizon N is as was presented in the Equation (10) where $\mathbf{y}_d(k+i) \in \mathbb{R}^2$. Expanding (10) using (24) and dropping terms that are constant with respect to $\tilde{\mathbf{u}}$ yields the standard quadratic form [63]:

$$J(\tilde{\mathbf{u}}(k)) = \frac{1}{2}\,\tilde{\mathbf{u}}(k)^T\mathbf{H}\,\tilde{\mathbf{u}}(k) + \underbrace{\left(\mathbf{F}_1\mathbf{x}(k) + \mathbf{F}_2\tilde{\mathbf{y}}_\mathrm{d} + \mathbf{F}_3\mathbf{u}_\mathrm{d}\right)^T}_{\mathbf{F}^T(k)}\tilde{\mathbf{u}}(k), \tag{27}$$

where

$$\mathbf{H} \triangleq 2\sum_{i=1}^{N}\left[\mathbf{\Psi}_i^T\mathbf{C}_r^T\mathbf{R}_w\mathbf{C}_r\mathbf{\Psi}_i + \left(\Pi_i^{(n_\mathrm{u},N)}\right)^T\mathbf{Q}_w\Pi_i^{(n_\mathrm{u},N)}\right], \tag{28}$$

https://doi.org/10.3390/act15050260

$$\mathbf{F}_1 \triangleq 2\sum_{i=1}^{N}\left[\mathbf{\Psi}_i^T\mathbf{C}_r^T\mathbf{R}_w\mathbf{C}_r\mathbf{\Phi}_i\right], \tag{29}$$

$$\mathbf{F}_2 \triangleq -2\sum_{i=1}^{N}\left[\mathbf{\Psi}_i^T\mathbf{C}_r^T\mathbf{R}_w\Pi_i^{(n_u,N)}\right], \tag{30}$$

$$\mathbf{F}_3 \triangleq -2\sum_{i=1}^{N}\left[\left(\Pi_i^{(n_u,N)}\right)^T\mathbf{Q}_w\right]. \tag{31}$$

Note that $\mathbf{H}$ is computed offline once for all, while $\mathbf{F}(k)$ is updated at every sampling instant k using the current state $\mathbf{x}(k)$ and the future reference sequence $\tilde{\mathbf{y}}_d(k)$.

The constrained optimal control sequence is the solution of the quadratic programming problem (QOP),

$$\tilde{\mathbf{u}}^\star(k) = \arg\min_{\tilde{\mathbf{u}}(k)} \; J(\tilde{\mathbf{u}}(k)), \tag{32}$$

subject to the hard constraints listed in Table 1:

$$\text{Position:} \quad \mathbf{y}^{\min} \le \mathbf{C}_c\,\mathbf{x}(k+i) \le \mathbf{y}^{\max}, \quad i=1,\dots,N, \tag{33}$$

$$\text{Velocity:} \quad \mathbf{u}^{\min} \le \mathbf{u}_p(k+i-1) \le \mathbf{u}^{\max}, \quad i=1,\dots,N, \tag{34}$$

$$\text{Velocity rate:} \quad \Delta\mathbf{u}_p^{\min} \le \mathbf{u}_p(k+i-1) - \mathbf{u}_p(k+i-2) \le \Delta\mathbf{u}_p^{\max}, \quad i=1,\dots,N. \tag{35}$$

The receding-horizon law applies only the first element of the optimal sequence:

$$\mathbf{u}_p(k) = \mathbf{\Pi}_1^{(2,N)}\,\tilde{\mathbf{u}}^\star(k). \tag{36}$$

Table 1. Operational constraints for the car-like mobile robot.

Variable	Lower Bound	Upper Bound
$x_p(k+i)$	$x_p^{\min}$	$x_p^{\max}$
$y_p(k+i)$	$y_p^{\min}$	$y_p^{\max}$
$u_x(k+i-1)$	$u_x^{\min}$	$u_x^{\max}$
$u_y(k+i-1)$	$u_y^{\min}$	$u_y^{\max}$

3.1.4. Frugal Control Parametrization

For a prediction horizon N and $n_u = 2$ inputs, the full decision vector $\tilde{\mathbf{u}}(k) \in \mathbb{R}^{2N}$ would require solving a QOP with $2N$ variables at every sampling instant. For embedded real-time use, this may exceed the available budget $T_c < T_s$.

To reduce the computational load while preserving tracking performance, a *frugal* control parametrization is adopted [60,63]. Instead of optimizing independently all $2N$ components, the input sequence is expressed as a linear function of a compact decision vector $\mathbf{p}_f(k) \in \mathbb{R}^{n_f}$ through a parametrization matrix $\mathbf{\Pi}_f \in \mathbb{R}^{2N\times n_f}$:

$$\tilde{\mathbf{u}}(k) = \mathbf{\Pi}_f\,\mathbf{p}_f(k). \tag{37}$$

A two-node piecewise-constant profile is adopted: the first node value $\mathbf{p}_{f,1}(k) \in \mathbb{R}^2$ is held constant over the first half of the horizon, and the second value $\mathbf{p}_{f,2}(k) \in \mathbb{R}^2$ over the second half. For N steps split at $n_s = \lceil N/2 \rceil$, the parametrization reads as

$$\mathbf{u}_p(k+i-1) = \begin{cases} \mathbf{p}_{f,1}(k), & i \in \{1,\dots,n_s\}, \\ \mathbf{p}_{f,2}(k), & i \in \{n_s+1,\dots,N\}, \end{cases} \tag{38}$$

https://doi.org/10.3390/act15050260

so that $n_q = n_f \cdot n_u = 2 \times 2 = 4$, i.e., the number of decision variables is reduced from $2N$ to just 4, independently of the chosen horizon length. The resulting parametrization matrix is

$$\mathbf{\Pi}_f = \begin{bmatrix} \mathbf{I}_2 & \mathbf{0}_2 \\ \vdots & \vdots \\ \mathbf{I}_2 & \mathbf{0}_2 \\ \mathbf{0}_2 & \mathbf{I}_2 \\ \vdots & \vdots \\ \mathbf{0}_2 & \mathbf{I}_2 \end{bmatrix} \in \mathbb{R}^{2N \times 4}, \tag{39}$$

where the upper $2n_s$ block-rows correspond to node 1 and the lower $2(N - n_s)$ block-rows to node 2.

Substituting (37) into (27) and the inequality constraints transforms the original QOP into the reduced problem,

$$\mathbf{p}_f^\star(k) = \arg\min_{\mathbf{p}_f(k)} \frac{1}{2}\, \mathbf{p}_f(k)^\top \mathbf{H}_f\, \mathbf{p}_f(k) + \mathbf{F}_f(k)^\top \mathbf{p}_f(k), \tag{40}$$

subject to $\mathbf{A}_{\text{ineq}}^{(f)}\, \mathbf{q}(k) \leq \mathbf{B}_{\text{ineq}}^{(f)}(k)$, where the reduced matrices are

$$\mathbf{H}_f \triangleq \mathbf{\Pi}_f^\top \mathbf{H}\, \mathbf{\Pi}_f \in \mathbb{R}^{4\times 4}, \qquad \mathbf{F}_f(k) \triangleq \mathbf{\Pi}_f^\top \mathbf{F}(k) \in \mathbb{R}^4, \tag{41}$$

and $\mathbf{A}_{\text{ineq}}^{(f)}$, $\mathbf{B}_{\text{ineq}}^{(f)}$ are obtained by mapping the constraints (33)–(35) through $\mathbf{\Pi}_f$ [60]. $\mathbf{H}_f$ is computed offline; only $\mathbf{F}_f(k)$ and $\mathbf{B}_{\text{ineq}}^{(f)}(k)$ need to be updated online. Finally, the receding-horizon control law is recovered as

$$\mathbf{u}_p(k) = \mathbf{\Pi}_1^{(2,N)}\, \mathbf{\Pi}_f\, \mathbf{p}_f^\star(k). \tag{42}$$

Remark 1. *Note that the control $\mathbf{u}_p(k)$ represents the desired linear velocity at the instant k such that the point P in the mobile frame follows the desired trajectory. Then, $\mathbf{u}_p(k)$ must be mapped to the desired angular velocities of each wheel. For that, at the instant k, the vector $[v_x(k)\ \omega(k)]^T$ is computed via ${\mathbf{J}_\mathbf{p}}^{-1}$ in* (17)*, and the steering angle $\delta(k)$ is later derived from* (2)*. Then, using* (7) *and* (8) *gives*

$$\omega_L^d(k) = \frac{v_x^d(k)}{r}\left(1 - \left(\frac{d}{2}\frac{\tan(\delta(k))}{\bar{L}}\right)\right) \tag{43}$$

$$\omega_R^d(k) = \frac{v_x^d(k)}{r}\left(1 + \left(\frac{d}{2}\frac{\tan(\delta(k))}{\bar{L}}\right)\right) \tag{44}$$

The complete FMPC structure for the car-like mobile robot is summarized in Figure 3. The user supplies the position constraints, weighting matrices $\mathbf{R}_w$ and $\mathbf{Q}_w$, and the reference trajectory; the FMPC block computes $\mathbf{u}_p(k)$ from the reduced QOP at every sampling instant k and issues the corresponding velocity and steering commands to the vehicle.

3.2. Nonlinear Dynamic Control for the Wheel Speed

The wheels in the mobile robot are actuated by DC-motors. Then, the angular velocity dynamics, in continuous-time, for each wheel is given by

$$\dot{\omega}_i(t) = -a\omega_i(t) + bu_{m_i}(t) + d_i(t), \ \text{ with } i \in \{L, R\} \tag{45}$$

https://doi.org/10.3390/act15050260

a and b are constants that depend on the electrical and mechanical parameters of the motor and will be assumed to be uncertain in this work. u_m is the applied voltage (control) and $d_i \in \mathbb{R}$ represents a signal that concentrates external disturbances, originating from the normal and tangential forces present in the wheels of the vehicle and reflected as a moment of forces around the rotational axis. In the following, the temporary variable t will be omitted unless the context requires it.

To establish a simple but robust dynamic controller, the dynamics (45) can be further written as

$$\dot{\omega}_i = \alpha(u_{m_i} - \xi_i) \tag{46}$$

where α and $\xi_i \in \mathbb{R}$ are assumed unknown. The control objective is to guaranty that $\omega_i \to \omega_i^d$, when $t \to \infty$ regardless of uncertainties in the model denoted by α, and the disturbance ξ which represents the sum of endogenous and exogenous disturbances and can be a time variant.

Before stating the main result, it is important to establish the following considerations:

1. $\alpha > 0$ such that $\alpha \in [\alpha_{\min}, \alpha_{\max}]$;
2. $\xi_i \in [\xi_{\min}, \xi_{\max}]$;
3. $u_{m_i} \in [-M, M], \forall\, M > 0$;
4. $|\frac{d\xi_i}{dt}| \leq K_0$.

Definition 1 (Saturation function)**.** *Given a positive constant M, the continuous and non decreasing function $\sigma_M : \mathbb{R} \to \mathbb{R}$ is defined as*

$$\begin{array}{l} (1)\ \sigma_M = s \text{ if } |s| < M; \\ (2)\ \sigma_M = M \cdot \text{sign}(s); \textit{ in other case} \end{array} \tag{47}$$

Proposition 1. *Consider the system* (46) *with the dynamic control law*

$$\begin{aligned} \dot{z}_i &= \lambda_f(u_{m_i} - z_i) \\ u_{m_i} &= \sigma_M(\lambda(\omega_i^d - \omega_i) + z_i) \end{aligned} \tag{48}$$

with $\lambda, \lambda_f \in \mathbb{R}^+$. Let us define $e_x := \omega_i - \omega_i^d$ and $e_z := z_i - \xi_i$ as the tracking error and the estimation error, respectively. In addition, consider $X_e = (e_x\ e_z)^T$ as the state vector in the error's coordinates. Then, after a long enough time, the error vector converges to the neighborhood of the origin.

$$\|X_e(t)\| \leq \frac{K_0}{\lambda_M} \tag{49}$$

K_0 was previously defined and λ_M is the highest eigenvalue of the matrix $A_s = \frac{1}{2}(A_e + A_e^T)$, where A_e is given by

$$A_e = \begin{pmatrix} -\alpha\lambda & \alpha \\ -\lambda_f\lambda & 0 \end{pmatrix} \tag{50}$$

The proof is given in Appendix A.

Remark 2. *Note that this low-level control law,* (48)*, is designed in continuous-time but implemented in discrete-time in an embedded microprocessor running to a frequency higher than the high-level control given by the FMPC.*

3.3. Extended Kalman Filter for State Estimation

In navigation applications, whether indoors or outdoors, it is common to utilize multiple sensors (such as encoders, cameras, laser sensors, ultrasonic sensors, IMUs, and

https://doi.org/10.3390/act15050260

GPS) to measure or estimate the vehicle's position. To reduce trajectory-tracking errors, it is essential for the mobile robot's current position to be accurate, as this accuracy allows the tracking controller to generate optimal control commands.

The Extended Kalman Filter (EKF) is a version of the Kalman filter designed for nonlinear systems [64]. While the standard filter only works optimally in linear systems, the EKF allows the state of more complex systems to be estimated via linearization. The fusion of sensory data with the Extended Kalman Filter (EKF) is enabled by a two-step iterative process that combines a mathematical model of the system with actual measurements from multiple sensors, assigning a weight to each measurement based on its uncertainty. The algorithm employs the Kalman gain ($\mathbf{K}_{kf}$) to dynamically balance model prediction and sensor measurement based on their respective uncertainties. The state and uncertainty are projected forward using a nonlinear transition model based on the system's physical laws. This distinguishes it from the standard Kalman Filter, which relies on a linear state-space representation. The discrete-time kinematic bicycle model (1) based on wheel speeds (4), which defines the state $\mathbf{q}_{k+1}$, is given by

$$\bar{\mathbf{q}}(k) = \hat{\mathbf{q}}(k-1) + \begin{bmatrix} [r\cos(\theta(k))][\omega_L(k)+\omega_R(k)]\Delta t/2 \\ [r\sin(\theta(k))][\omega_L(k)+\omega_R(k)]\Delta t/2 \\ [r\tan(\theta(k))][\omega_L(k)+\omega_R(k)]\Delta t/2\bar{L} \end{bmatrix} + \mathbf{w}(k) \tag{51}$$

where Δt denotes the sampling interval between updates and $\mathbf{w}(k)$ is the process noise vector, accounting for modeling inaccuracies and external disturbances in the robot's motion.

Since the EKF operates on a linear approximation of the system dynamics, we linearize at each time step around the current operating point. This allows for the forward propagation of the state uncertainty. The corresponding Jacobian is expressed as

$$\mathbf{F}_{kf}(k) = \frac{\partial f[\hat{\mathbf{q}}(k-1), \mathbf{u}_{kf}(k-1)]}{\partial \mathbf{q}} = \begin{bmatrix} 1 & 0 & -[r\sin(\theta(k-1))][\omega_L(k-1)+\omega_R(k-1)]\Delta t/2 \\ 0 & 1 & [r\cos(\theta(k-1))][\omega_L(k-1)+\omega_R(k-1)]\Delta t/2 \\ 0 & 0 & 1 \end{bmatrix} \tag{52}$$

An OptiTrack system provides direct state observations, yielding a linear measurement model. Under the assumption of full state observability, the observation function $h(\mathbf{q}_k)$ is defined by

$$\mathbf{z}_{kf}(k) = h(\mathbf{q}(k)) + \mathbf{v}(k) = \begin{bmatrix} x(k) \\ y(k) \\ \theta(k) \end{bmatrix} + \mathbf{v}(k) \tag{53}$$

where $\mathbf{z}_{kf}(k)$ represents the actual measurements at time k, while $\mathbf{v}(k)$ accounts for the additive measurement noise. To facilitate filter update, the measurement model is linearized about the current predicted state $\hat{\mathbf{q}}_k$ as follows:

$$\mathbf{H}_{kf} = \frac{\partial h(\mathbf{q}(k))}{\partial \mathbf{q}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{54}$$

The stochastic components of the model, namely the process noise $\mathbf{w}_k$ and measurement noise $\mathbf{v}_k$, are assumed to be zero-mean additive White Gaussian Noise (WGN). These are characterized by their respective covariance matrices $\mathbf{Q_{kf}} \in \mathbf{R}^{3\times3}$ and $\mathbf{R_{kf}} \in \mathbb{R}^{3\times3}$. In this study, the measurement noise covariance $\mathbf{R}_{kf}$ is empirically determined on the basis of the static precision of the OptiTrack system:

https://doi.org/10.3390/act15050260

$$\mathbf{R}_{kf} = \begin{bmatrix} \sigma^2_{opt\_x} & 0 & 0 \\ 0 & \sigma^2_{opt\_y} & 0 \\ 0 & 0 & \sigma^2_{opt\_\theta} \end{bmatrix} \tag{55}$$

The covariance matrix represents the uncertainties in the kinematic model, incorporating errors such as wheel slippage and surface irregularities. Following the approach in [65], this is defined based on the expected variability of control inputs and motion model noise. Under the assumption that the noise in the steering angle and wheel speeds are independent and Gaussian, the matrix is expressed as

$$\mathbf{Q}_{kf} = \begin{bmatrix} \sigma^2_x & 0 & 0 \\ 0 & \sigma^2_y & 0 \\ 0 & 0 & \sigma^2_\theta \end{bmatrix} \tag{56}$$

where σ^2_x, σ^2_y and σ^2_θ are the variance of the prediction states (x, y, θ). The EKF algorithm operates iteratively through the following two main phases.

3.3.1. Prediction

In the first stage, the state vector is projected forward using the nonlinear motion model:

$$\bar{\mathbf{q}}(k) = f(\hat{\mathbf{q}}(k-1), \mathbf{u}_{kf}(k)) + \mathbf{w}(k) \tag{57}$$

The state covariance is propagated through the linearized system dynamics:

$$\bar{\mathbf{P}}(k) = \mathbf{F}_{kf}(k)\hat{\mathbf{P}}(k-1)\mathbf{F}^T_{kf}(k) + \mathbf{Q}_{kf} \tag{58}$$

3.3.2. Update

The innovation vector $\mathbf{y}(k)$ represents the discrepancy between the actual sensor measurement and the predicted observation:

$$\mathbf{y}(k) = \mathbf{z}_{kf}(k) - \mathbf{H}_{kf}\bar{\mathbf{q}}(k) \tag{59}$$

Once a measurement is available, the Kalman gain $\mathbf{K}_{kf}$ is computed to weigh the reliability of the sensors:

$$\mathbf{K}_{kf}(k) = \bar{\mathbf{P}}(k)\mathbf{H}^T_{kf}(\mathbf{H}_{kf}\bar{\mathbf{P}}(k)\mathbf{H}^T_{kf} + \mathbf{R}_{kf})^{-1} \tag{60}$$

In the final step, the posteriori state estimate and its associated covariance are refined by means of the calculated Kalman gain:

$$\hat{\mathbf{q}}(k) = \bar{\mathbf{q}}(k) + \mathbf{K}_{kf}\mathbf{y}(k) \tag{61}$$

$$\hat{\mathbf{P}}(k) = (\mathbf{I} - \mathbf{K}_{kf}\mathbf{H}_{kf})\bar{\mathbf{P}}(k) \tag{62}$$

By iteratively applying these prediction and correction phases, the EKF minimizes the estimation error covariance, providing a robust tracking of the vehicle's trajectory.

4. Experimental Platform Description

To validate the control strategy, an ecosystem was developed for the experimental deployment and analysis of the results. This ecosystem consists of three main components: (1) a real-time localization system (RTLS), (2) a car-like robot equipped with an open-architecture autopilot, and (3) a control station responsible for (i) processing information from the RTLS, (ii) executing the control strategy, and (iii) sending the desired steer angle

https://doi.org/10.3390/act15050260

and the desired angular velocity to each motor of the vehicle so that deployment tasks are executed. Figure 5 shows the described ecosystem.

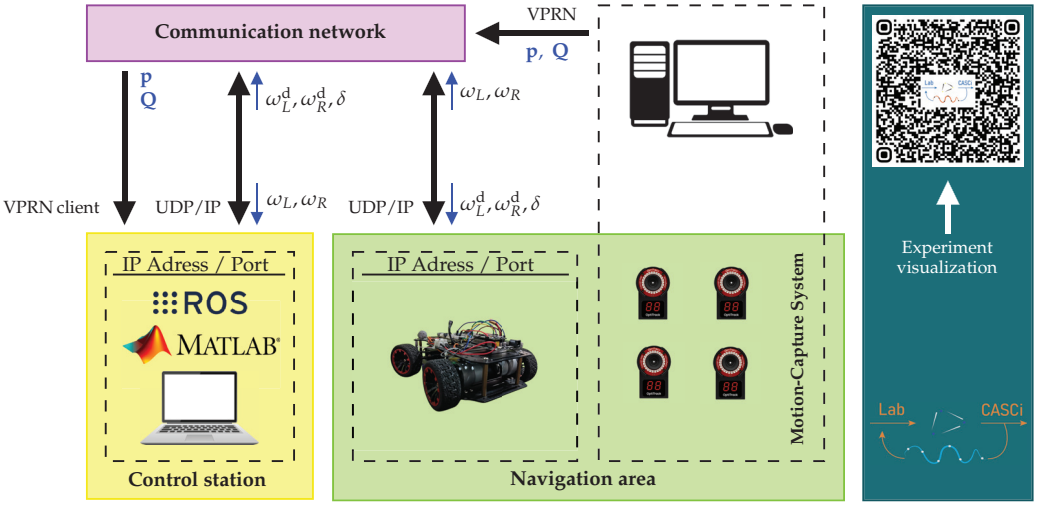


Figure 5. The experimental platform is divided into four key workspaces: the Wi-Fi communication network, the ground control station, the motion capture (MoCap) system, and the navigation test area. The main components include a local router, a ground station for executing control algorithms, MoCap equipment, and the mobile robot. To view a video of the experiment, please visit https://www.youtube.com/watch?v=MQM11T2g1CE or scan the QR code. Accessed on 1 May 2026.

The ground station features are a 12th-generation Core i7-1255U 1.7 GHz processor, 12 GB RAM, Ubuntu 22.04 LTS, and MATLAB 2024B. The onboard controller operates under the ESP32 Devkit v1 together with the board based on a STM32H7 microcontroller, allowing operation in a 100 Hz control loop.

4.1. RTLS

The localization stage is based on a motion capture (MoCap) system comprising a network of infrared cameras and specialized software that allows real-time monitoring of an object's movement from light reflected by markers attached to the vehicle. In this case, the Optitrack system was used, which provides the pose, a 6-DoF tracking vector: 3 DoF for position and 3 DoF for orientation of the vehicle. In our case, only 2 DoF for position and 1 DoF for orientation are used.

4.2. Robotic Platform

The physical agent is a car-like mobile robot, characterized by non-holonomic constraints (Ackermann-steering vehicle). It is equipped with two 12V DC encoder motors on the rear axle and a controller on board for speed control and wireless communication. Table 1 details the platform characteristics.

4.3. Control Station

The control station (CS) acts as the system's central hub, utilizing ROS2 Humble to manage data distribution. It publishes the vehicle's pose to a high-level FMPC trajectory-tracking controller within the MATLAB 2024B environment. Furthermore, the CS processes wheel speed data through an Extended Kalman Filter (EKF), fusing it with kinematic odometry to provide a robust pose estimation for the car-like robot. All system variables are monitored in real-time through the control station.

4.4. Extended Kalman Filtering Tuning

For computational efficiency, the process noise covariance matrix, $\mathbf{Q}_{kf}$, is assumed to be diagonal, where each element represents the error variance of the corresponding state component (Equation (56)). The terms σ_x^2 and σ_y^2 are modeled as proportional to the traveled distance and velocity, whereas σ_θ^2 accounts for uncertainties during rotational maneuvers. Experimental calibration is performed through straight-line motion at a constant velocity to characterize the position error variances σ_x^2 and σ_y^2. Analogously, σ_θ^2 is determined through controlled rotation tests. A high-magnitude $\mathbf{Q}_{kf}$ de-weights the odometry model in favor of external correction sensors, whereas a low-magnitude $\mathbf{Q}_{kf}$ prioritizes the dead reckoning estimates.

To ensure the reproducibility of the estimation stage, the EKF parameters are provided. The process noise covariance matrix $\mathbf{Q}_{kf}$ was set to

$$\mathbf{Q}_{kf} = \begin{bmatrix} 0.01 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.01 \end{bmatrix} \tag{63}$$

The measurement noise covariance matrix was characterized based on the OptiTrack motion capture system's precision (with an experimental *error* < 2 mm):

$$\mathbf{R}_{kf} = \begin{bmatrix} 0.00001 & 0 & 0 \\ 0 & 0.00001 & 0 \\ 0 & 0 & 0.00001 \end{bmatrix} \tag{64}$$

Regarding the prediction stage, the geometric parameters required for odometry and state estimation are summarized in Table 2.

Table 2. Car-like mobile robot platform parameters.

Parameter	Notation	Value
Wheel base	$\bar{L}$	0.175 m
Wheel radius	r	0.033 m
Track width	d	0.18 m
Virtual point	p	0.08 m
Maximum steering angle	δ_{max}	0.69 rad
Maximum speed of wheels	ω_{imax}	37.69 rad/s
Maximum linear velocity	v_{xmax}	1.24 m/s
Maximum angular velocity	ω_{max}	0.94 rad/s

4.5. Experiment Description

To validate the proposed control scheme and evaluate the integrated subsystems, five experiments were designed, each repeated 10 times to assess reliability. In all cases, the objective was to track a time-parameterized lemniscate trajectory. The trajectory is defined by the following parametric equations:

$$\begin{aligned} x_d(t) &= \frac{\bar{a}\cos(\omega_o t)}{\sin^2(\omega_o t)+1} \\ y_d(t) &= \frac{\bar{a}\cos(\omega_o t)\sin(\omega_o t)}{\sin^2(\omega_o t)+1} \end{aligned} \tag{65}$$

where $\bar{a}$ and ω_o represent the amplitude of the trajectory and the angular frequency, respectively.

https://doi.org/10.3390/act15050260

The proposed trajectory-tracking strategy is based on a kinematic bicycle model, focusing exclusively on planar translation velocities and the yaw rate ($\dot{\theta}$). Given that the high-level Model Predictive Control (MPC) employs a kinematic model linearized through input–output feedback linearization (shifted output), dynamic effects such as lateral and rolling forces are inherently excluded from the formulation. Consequently, the operational range is restricted to low-speed maneuvers, where the non-slip assumption remains valid and dynamic perturbations can be effectively mitigated by the low-level controllers, which incorporate disturbance observers for the traction units. Furthermore, the electronic differential algorithm remains applicable under these conditions. To ensure the physical consistency of the zero-slip hypothesis during experimental validation, high-density EVA foam was utilized to provide a high-friction contact surface. The actuator constraints and the steering limits of the car-like mobile robot are summarized in Table 2.

- Experiment 1. The objective of this test is to evaluate the performance of the FMPC relative to a Linear Quadratic Regulator (LQR) using identical cost matrices. Furthermore, trajectory-tracking precision is assessed at a control frequency of 100 Hz, while the EKF-based sensor fusion is validated using a 1 Hz correction rate to test the system's robustness under low-frequency updates.
- Experiment 2. Leveraging the enhanced capabilities of the EKF-based framework, this test evaluates how state estimation bolsters the robustness of trajectory tracking. The architecture facilitates multi-rate operation, decoupling the high-frequency control execution from the lower-frequency pose measurements. Specifically, the EKF prediction stage and the control loop operate at a 10 ms sampling period, whereas the EKF correction stage is triggered at a 4 s interval to assess the system's dead-reckoning reliability during sensor latency.
- Experiment 3. A fundamental advantage of Model Predictive Control (MPC) is its inherent capability to incorporate explicit state and control input constraints. This experiment validates this feature by imposing the state constraint $x_{min} \leq x_k \leq x_{max}$, ensuring the car-like robot's pose remains within a predefined safe operating range. The boundaries for the linear trajectory were set to ± 0.85 m, while for the circular path, they were defined as ± 1.1 m. To further challenge the controller's constraint-handling capabilities, the trajectory amplitude $\bar{a}$ was increased to 1.3 m, exceeding the 1.2 m used in previous benchmarks.
- Experiment 4. The robustness of the control scheme against external disturbances was evaluated through the low-level controller. This controller features a nonlinear structure integrated with an input-based disturbance observer. To test its ability to regulate wheel speed under variable conditions, an additional payload of 896 g was added to the vehicle. Within the trajectory-tracking framework, the vehicle is required to track a specific velocity profile to ensure spatial accuracy at every time step. Therefore, the low-level control must effectively compensate for the added mass and maintain the velocity within the bounds defined by the high-level trajectory-tracking algorithm.
- A central contribution of this work is the computational efficiency of the control scheme. Specifically, motor control, data fusion for pose estimation, and the trajectory-tracking algorithm are executed within a 10 ms sampling interval, necessitating a low computational overhead. To evaluate the impact of the proposed FMPC parameterization on both execution time and tracking precision, its performance was benchmarked against a conventional, non-parameterized MPC.

https://doi.org/10.3390/act15050260

5. Experimental Results

5.1. Experiment 1: DLQR vs. FMPC

To validate the proposed control scheme and ensure a fair comparison, a Digital Linear Quadratic Regulator (DLQR) was developed, as both approaches are categorized as optimal control strategies and were tested under identical conditions. The resulting time-series trajectory tracking for each state is presented in Figure 6. The dashed black line represents the reference, while solid lines and shaded areas denote the mean (μ) and confidence intervals ($\pm 2\sigma$) for DLQR (orange) and FMPC (blue), maintaining consistency across all comparative results. It is observed that the FMPC (blue line) exhibits a slightly more pronounced initial transient in the y-axis—an expected behavior due to the imposed constraints—yet it successfully converges to the reference smoothly. Both controllers demonstrate excellent tracking capabilities relative to the reference signals. Furthermore, the uncertainty bands ($\mu \pm 2\sigma$) suggest that the FMPC maintains a competitive variance against the DLQR, which is noteworthy given that the FMPC solves a constrained optimization problem at each time step. The shaded regions in the θ plot indicate uncertainty peaks during abrupt changes in direction; however, the FMPC effectively manages these high-dynamic regions with a controlled standard deviation.

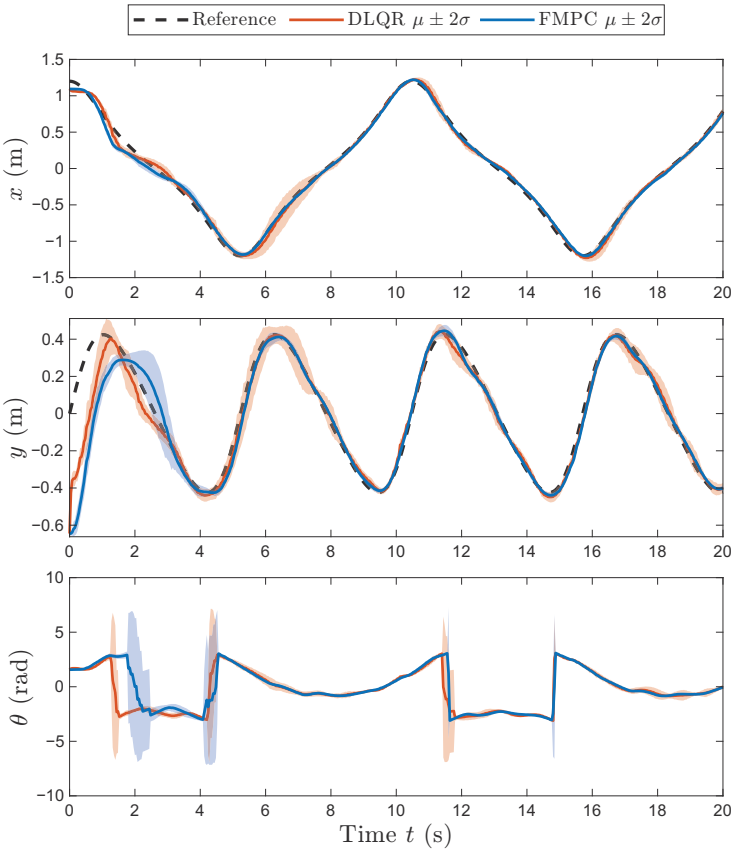


Figure 6. DLQR vs. FMPC with correction at 1 s.

The Figure 7 illustrates the control effort for the traction motors (ω_L, ω_R) and the steering angle (δ). The DLQR exhibits higher variance and more aggressive oscillations, which could lead to mechanical fatigue or actuator damage in a physical implementation. Furthermore, the motor velocity profiles for the FMPC are smoother, indicating a more energy-efficient control action compared to the DLQR, while maintaining competitive tracking performance.

https://doi.org/10.3390/act15050260

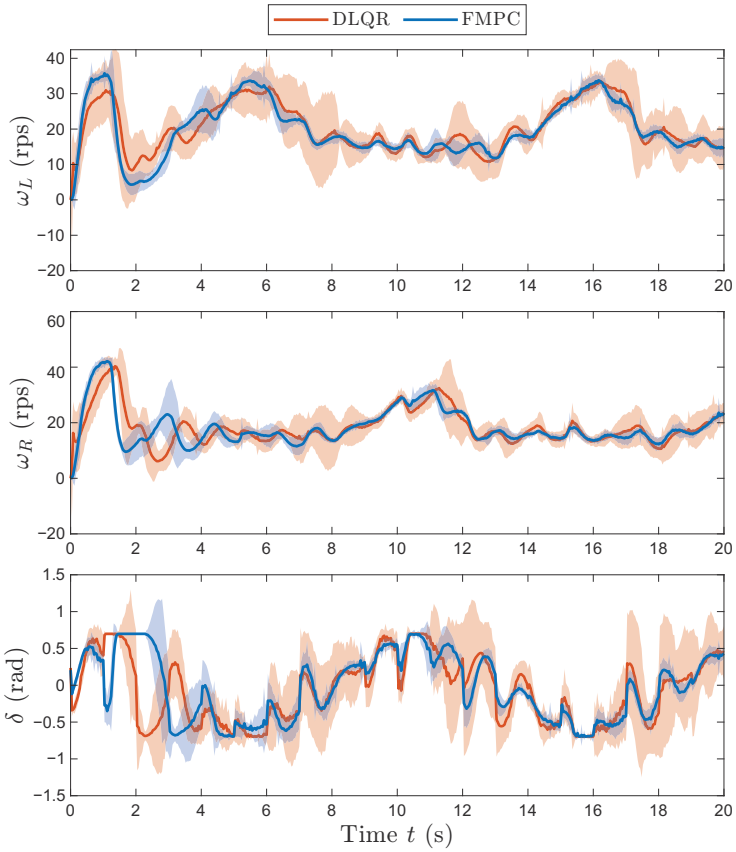


Figure 7. Control signals in DLQR vs. FMPC comparison.

The trajectory-tracking performance in the phase space (x-y plane) is shown in Figure 8. As previously identified in the time-series state dynamics, both approaches achieve successful tracking of the reference trajectory, despite the EKF pose correction occurring at a low frequency of 1 Hz. Notably, the FMPC exhibits more reliable behavior due to its iterative optimization process at each sampling instant; consequently, the steady-state covariance once the trajectory is reached is minimal. Furthermore, its predictive horizon allows the system to anticipate the desired path, resulting in a smoother initial response. In contrast, while the DLQR is also capable of ensuring correct trajectory following, its repeatability is lower. This is attributed to its reactive nature when responding to positioning errors, despite its inherent optimal formulation.

To quantitatively evaluate the tracking performance, three integral error criteria were computed: Integral Square Error (ISE), Integral Absolute Error (IAE), and Integral Time-weighted Absolute Error (ITAE). The resulting metrics are presented in Table 3. It is important to note that these metrics were calculated after an initial transient period ($t > 5$ s) to ensure that the comparison reflects the controllers' ability to maintain the trajectory rather than their initial convergence speed. As shown in Figure 9, the FMPC consistently outperforms the DLQR across all metrics. The significantly lower ISE and ITAE values demonstrate that the FMPC achieves higher tracking fidelity and superior convergence stability, effectively mitigating cumulative errors that the reactive DLQR fails to suppress over time.

https://doi.org/10.3390/act15050260

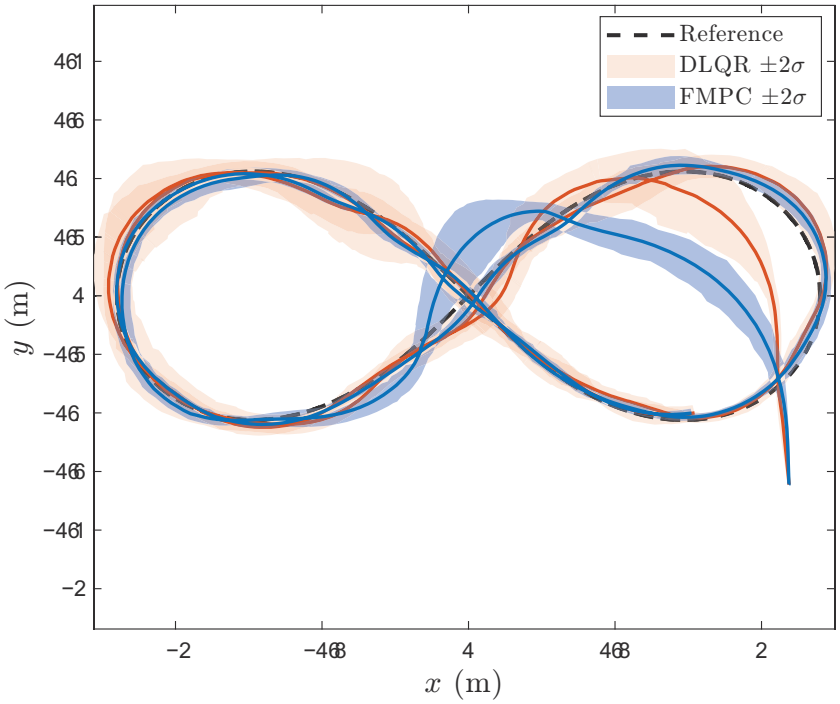


Figure 8. The trajectory-tracking performance in the phase space (x-y) plane for DLQR vs. FMPC.

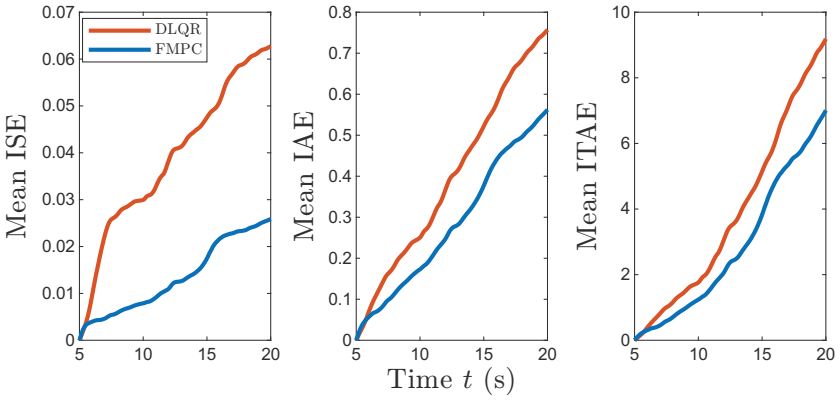


Figure 9. ISE, IAE, and ITAE evolution for both DLQR and FMPC with correction at 1 s.

Table 3. Quantitative evaluation of tracking precision for Experiment 1. The table summarizes ISE, IAE, and ITAE values for both DLQR and FMPC implementations.

Metric	DLQR	FMPC
Mean ISE	0.0627	0.0259
Mean IAE	0.7571	0.5622
Mean ITAE	9.1860	7.0081

5.2. Experiment 2: Validation of Pose Estimation Under Multi-Rate EKF Correction Frequencies

The complete control architecture is where all the processes are synchronized at a high-frequency sampling rate of 100 Hz (10 ms). This rapid execution cycle ensures seamless interaction between the EKF-based state estimation and the FMPC law. As evidenced by the time-series plots in Figure 10, the Extended Kalman Filter provides a robust, low-latency pose estimate, effectively mitigating sensor noise while maintaining high-fidelity tracking of the vehicle's rapid dynamics. The actuator responses for the traction motors (ω_L, ω_R) and the steering angle (δ) demonstrate near-perfect alignment with the required setpoints, even during aggressive maneuvers (as shown in Figure 11).

https://doi.org/10.3390/act15050260

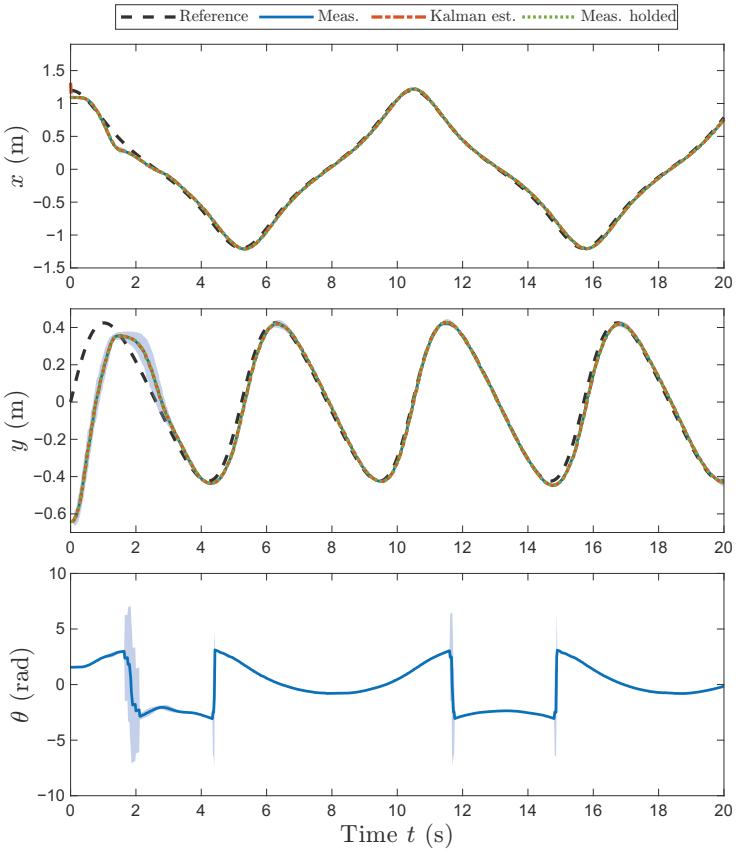


Figure 10. States evolution along the time of FMPC with EKF correction at 10 ms.

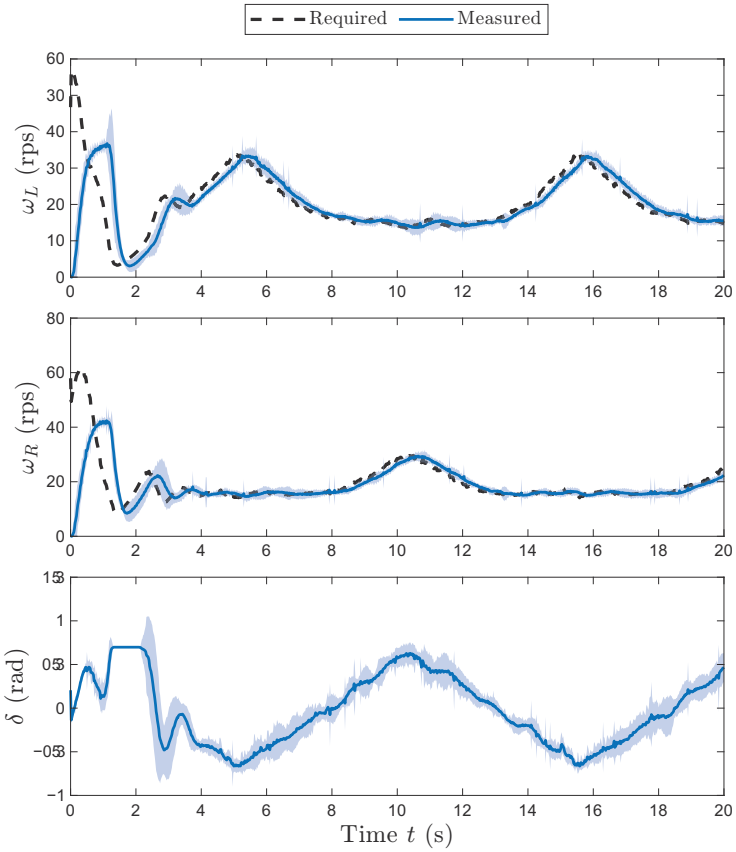


Figure 11. Control signals for FMPC with EKF correction at 10 ms.

https://doi.org/10.3390/act15050260

Phase-space analysis confirms exceptional trajectory repeatability, characterized by minimal steady-state variance captured in Figure 12. These results definitively validate the real-time feasibility and computational efficiency of the proposed hierarchical scheme on an embedded platform.

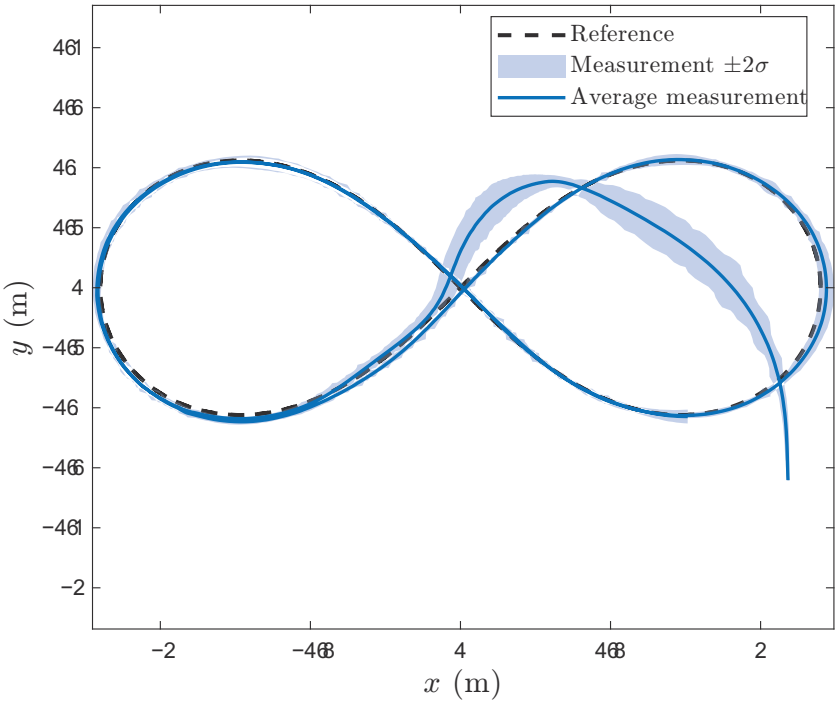


Figure 12. Trajectory-tracking performance of control schema with EKF correction at 10 ms.

To evaluate the system's reliability during sensor degradation or low-frequency positioning updates, the EKF correction stage was throttled to a 4-s interval. As illustrated in Figures 13–15, the hierarchical scheme maintains trajectory integrity through extended periods of dead-reckoning. The time-series analysis reveals that the EKF prediction stage successfully bridges the 4-s gaps between measurements, preventing the controller from becoming unstable. Although the phase-space plots show a marginal increase in the confidence intervals ($\pm 2\sigma$)—typical of cumulative odometric drift—the FMPC effectively manages this uncertainty, resulting in a smooth path that converges back to the reference upon each sensor update. These results confirm the high robustness of the proposed estimator–controller pair under extreme feedback latency. Furthermore, this experiment effectively simulates a GPS-denied environment or a scenario with significant sensor occlusion, where the vehicle must rely solely on its internal kinematic model and odometry for extended periods. Despite the 4-s latency, the FMPC demonstrates remarkable smoothness during the state-correction instances. Rather than exhibiting abrupt 'jumps' or erratic steering behavior when a new measurement is received, the controller leverages its optimization-based nature to seamlessly re-integrate the updated pose into the predictive plan. This ability to maintain dynamic stability under extreme feedback sparsity highlights the reliability of the proposed hierarchical framework for real-world autonomous missions where sensor reliability cannot be guaranteed.

https://doi.org/10.3390/act15050260

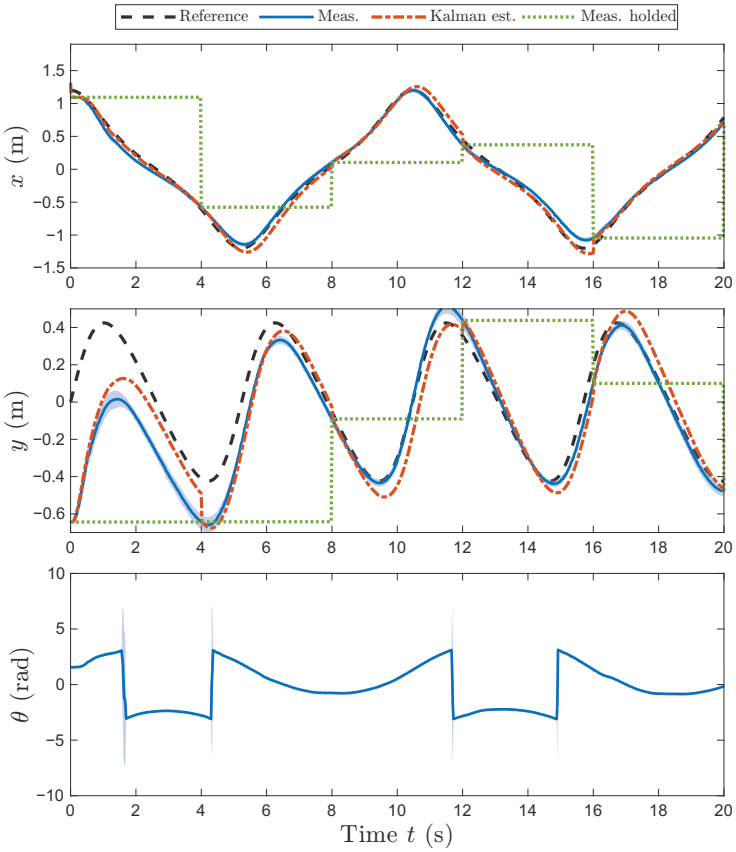


Figure 13. x, y and θ states evolution at 4 s correction rate.

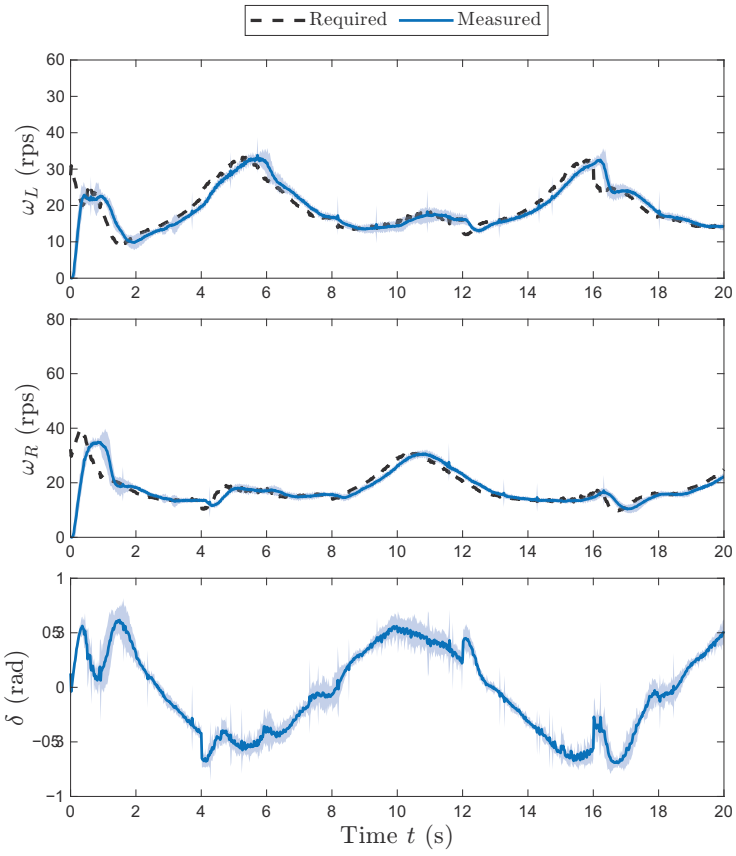


Figure 14. Car-like mobile robot control signals (wheels' speeds and steering angle) at 4 s rate correction.

https://doi.org/10.3390/act15050260

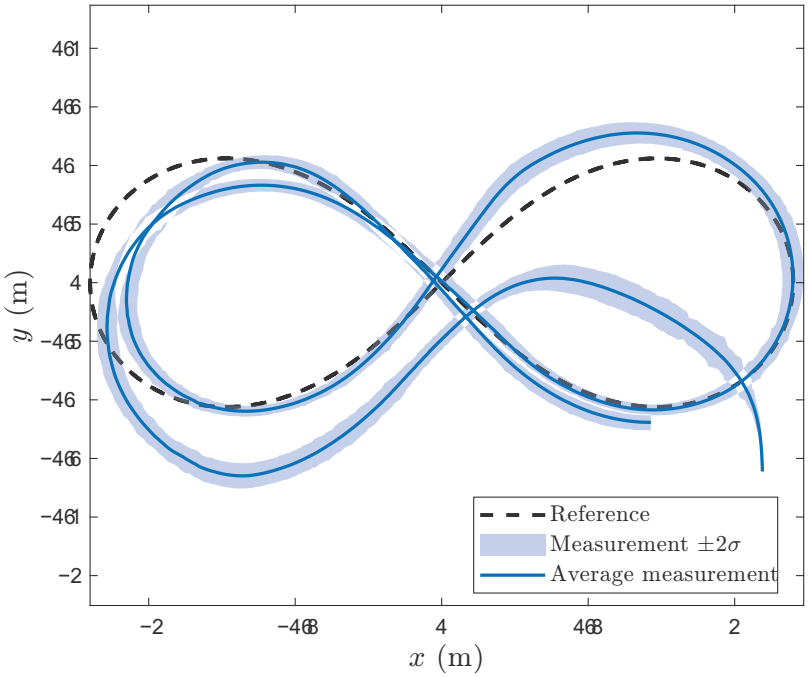


Figure 15. Trajectory-tracking performance of mobile robot at 4 s rate correction.

The comparative performance analysis under variable correction rates is presented in Figure 16 and Table 4, the cumulative error metrics (ISE, IAE, and ITAE) for three distinct EKF correction intervals are 10 ms, 1 s, and 4 s.

A key finding is the minimal performance gap between the 10 ms and 1 s correction rates, indicating that the EKF effectively bridges the feedback latency without significant degradation in tracking precision. While the 4 s correction interval (green line) exhibits higher cumulative error—attributable to the prolonged periods of open-loop odometric prediction—it maintains a stable and predictable trend. The fact that the ITAE slope remains bounded even under such sparse feedback conditions validates the robustness of the FMPC-EKF pair, confirming its suitability for autonomous navigation in environments with intermittent sensor availability.

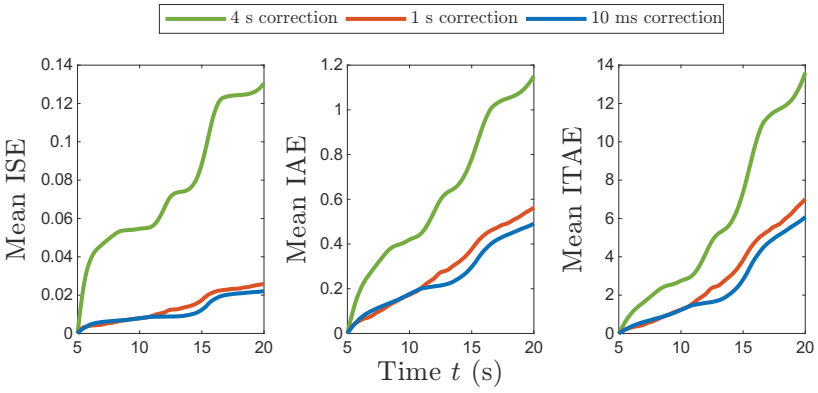


Figure 16. Performance for EKF each correction time.

Table 4. Performance metrics (ISE, IAE, and ITAE) of the FMPC-EKF scheme across different pose correction sampling periods.

Metric	4 s Correction	1 s Correction	10 ms Correction
Mean ISE	0.1305	0.0259	0.0221
Mean IAE	1.1510	0.5622	0.4914
Mean ITAE	13.6049	7.0081	6.0795

https://doi.org/10.3390/act15050260

5.3. Experiment 3: State Constraint Handling

Regarding the operational constraints, Figure 17 demonstrates the FMPC's ability to enforce strict operational boundaries, specifically the lateral state constraint $x_{min} \leq x_k \leq x_{max}$. In both the linear (a) and circular (b) tracking scenarios, the FMPC successfully confines the robot's pose within the predefined safe zones (indicated by the red-shaded regions). As observed in the $x(t)$ plots, whenever the reference trajectory exceeds the safety limits, the controller prioritizes constraint satisfaction over reference tracking, effectively saturating the state at the boundary. The phase-space plots further confirm this behavior, showing how the vehicle's trajectory is 'compressed' to stay within the allowable x-range. This inherent safety feature is critical for autonomous navigation in constrained environments, proving that the proposed hierarchical scheme ensures collision avoidance and regulatory compliance through its active optimization-based framework.

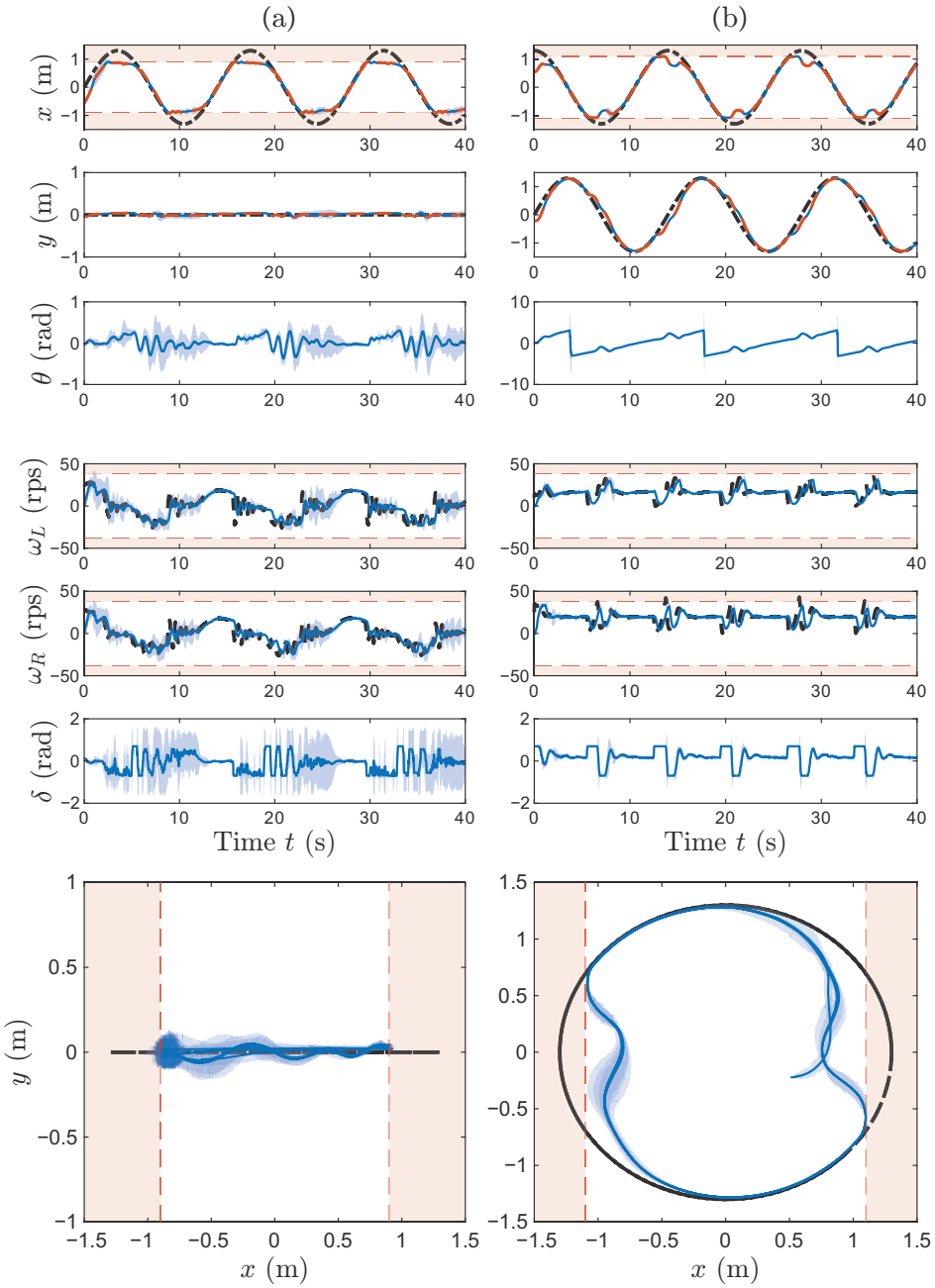


Figure 17. Constrained scenarios. (**a**) Straight line as a reference, and (**b**) circle as a reference.

https://doi.org/10.3390/act15050260

5.4. Experiment 4: Load Disturbance Compensation

In trajectory tracking, precise longitudinal speed control is critical due to the time-dependent nature of the reference trajectory. As illustrated in the upper plots of Figure 18, the wheel angular velocities successfully track the reference signals generated by the electronic differential, even under an 896 g payload. Despite the dynamic effects introduced by the additional mass, the tracking error remains within acceptable margins. This performance is attributed to the low-level controller, which incorporates a control-input-based disturbance observer and explicitly accounts for actuator saturation limits. The lower plots of Figure 18 depict the evolution of the disturbance estimation for both the left and right wheel motors. The observer effectively compensates for the added mass, which is reflected in the increased magnitude of the control signals. Notably, this increase is more pronounced in the left motor; although the payload was intended to be centered, its final placement fell slightly toward the left side. The control scheme successfully identified and compensated for this experimental asymmetry, maintaining system stability. The narrow variance ($\mu \pm 2\sigma$) shown in the shaded regions of Figure 19 indicates high repeatability. Furthermore, while the performance indices (Figure 20) show a cumulative increase in tracking error—particularly the ITAE, which rises more sharply during segments of maximum curvature, the state-phase diagram confirms that this deviation is not significant. Ultimately, the vehicle maintains high fidelity to the lemniscate geometry, demonstrating that the proposed control scheme is robust against both significant payloads and unmodeled mass distributions. Table 5 summarizes ISE, IAE, and ITAE values for both nominal and loaded conditions.

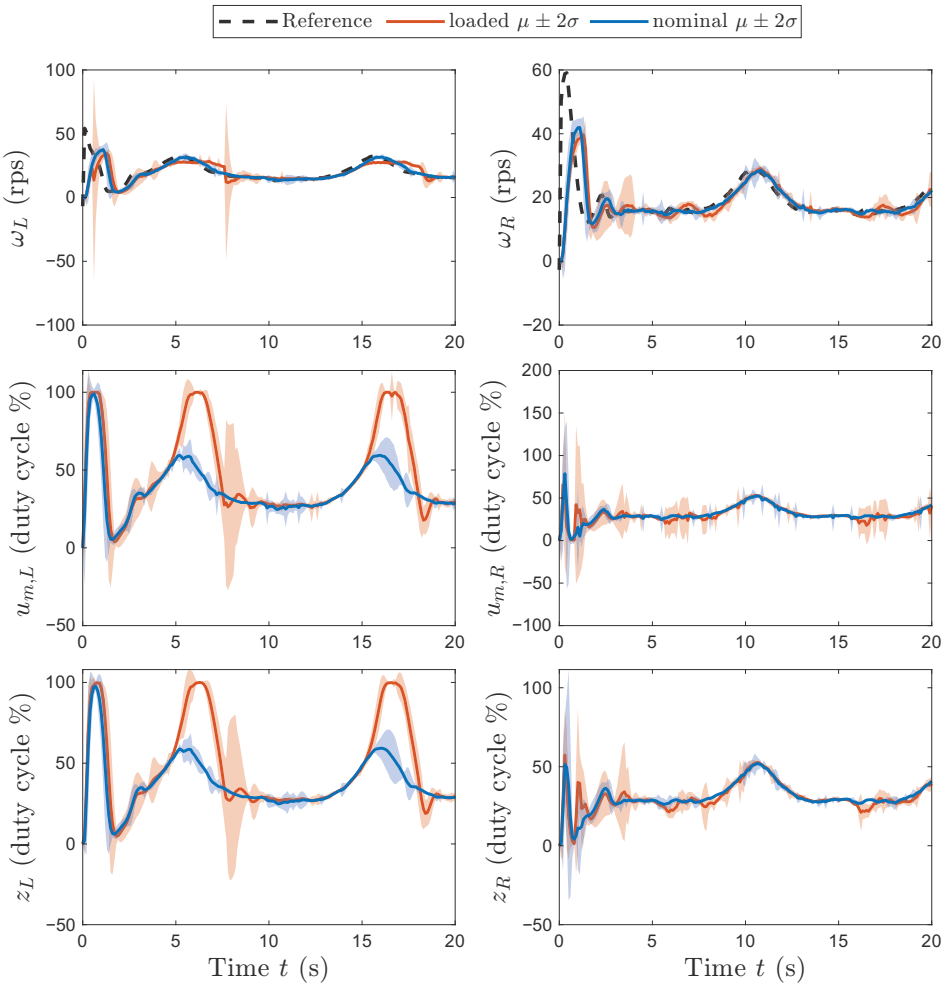


Figure 18. Comparison of low-level control signals under nominal and loaded conditions.

https://doi.org/10.3390/act15050260

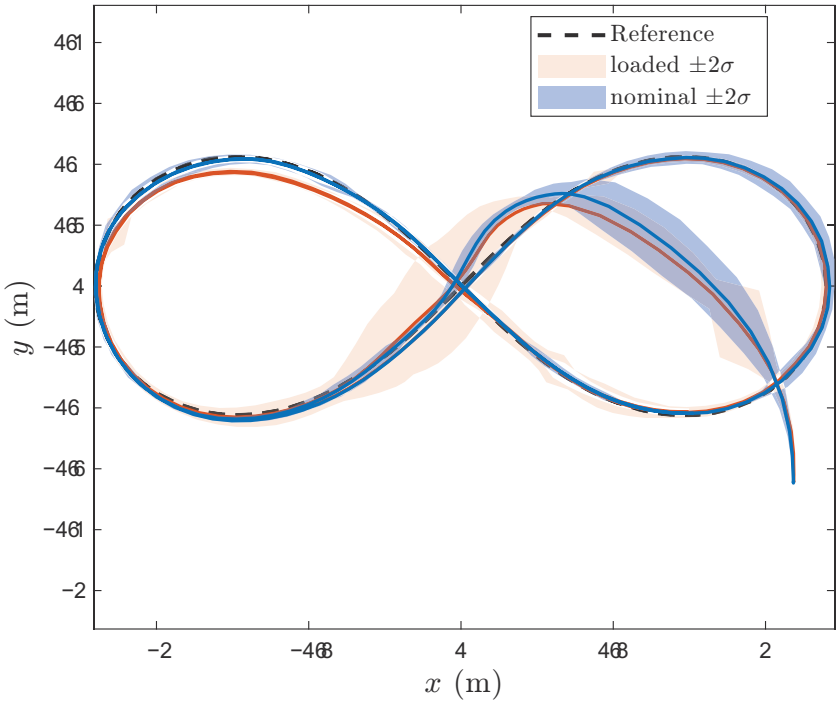


Figure 19. Comparison of trajectory-tracking performance: nominal vs. loaded conditions.

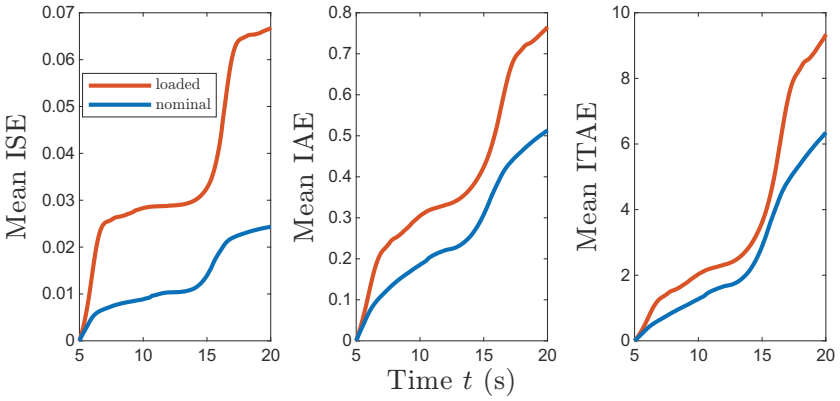


Figure 20. Tracking error metrics comparison for nominal and loaded experimental scenarios.

Table 5. Quantitative evaluation of tracking error for Experiment 4 (Load Disturbance Compensation).

Metric	Nominal	Loaded
Mean ISE	0.0243	0.0668
Mean IAE	0.5134	0.7656
Mean ITAE	6.3483	9.3463

5.5. Experiment 5: MPC vs. FMPC (Computational Cost)

The practical feasibility of the proposed hierarchical control scheme is central to its application in real-world environments. The architecture aims to achieve precise and robust trajectory tracking under strict constraints within a 10 ms sampling interval. This enables the potential deployment of the algorithm on embedded platforms with limited computational resources. To validate the optimization of the computational load, the proposed FMPC was benchmarked against a conventional, non-parameterized MPC. Both controllers utilized the same double-integrator model, identical constraints, cost-function weighting matrices, and numerical solver. As shown in Figure 21 and Table 6, the FMPC significantly reduces computational overhead while simultaneously improving tracking

https://doi.org/10.3390/act15050260

precision, yielding lower error values across all performance indices (ISE, IAE, and ITAE). Regarding execution time (Figure 22), the conventional MPC operates near the sampling limit, with its variance ($\mu \pm 2\sigma$) frequently exceeding the 10 ms threshold. Conversely, the FMPC consistently solves the optimization problem well below the sampling period with negligible variance.

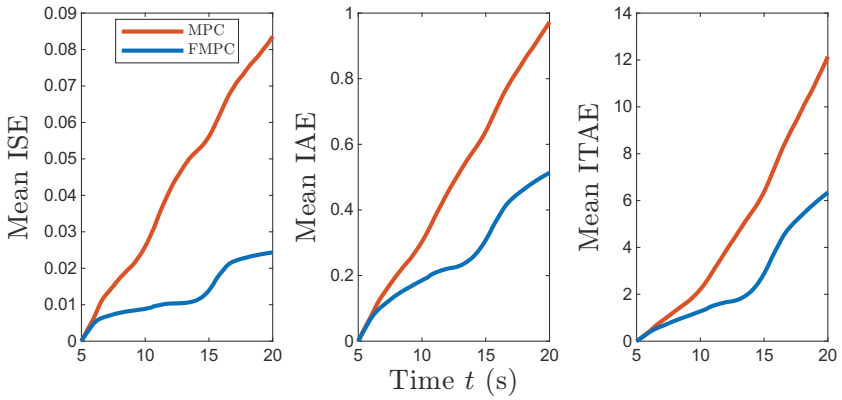


Figure 21. First figure of the calculation time comparison.

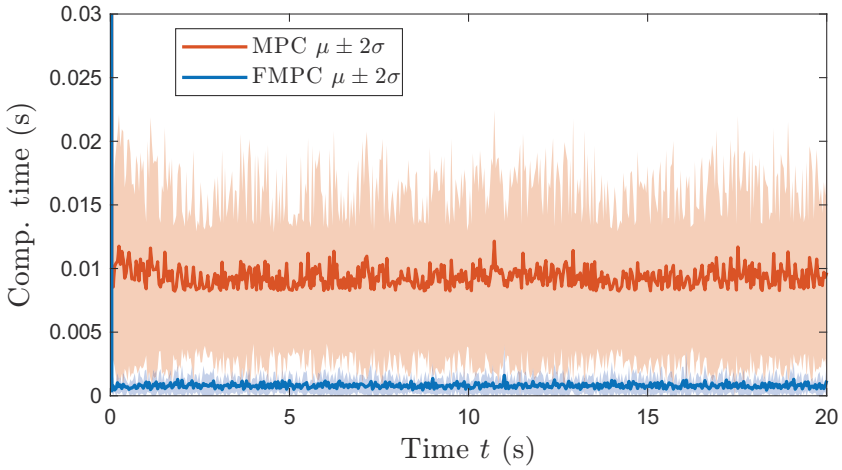


Figure 22. Second figure of the calculation time comparison.

These results highlight the superior real-time performance of the FMPC and its suitability for fully embedded implementation.

Table 6. MPC vs. FMPC metrics.

Metric	MPC	FMPC
Mean ISE	0.0836	0.0243
Mean IAE	0.9728	0.5134
Mean ITAE	12.1433	6.3483
Mean computation time	9.4647 ms	0.8652 ms

6. Empirical Stability Analysis via Cost-Function Evolution

To provide empirical evidence of the closed-loop stability properties of the proposed FMPC algorithm, the temporal evolution of the cost function $J(k)$ and its first-order increment $\Delta J(k) = J(k) - J(k-1)$ were evaluated along ten independent experimental runs as described in experiments 1, 2, and 4. At each sampling instant, $J(k)$ was computed according to the standard quadratic form (10) with $N = 30$, $T_s = 10$ ms, and the steady-state input u_d obtained from the standard rank condition associated with the augmented system

matrix, in agreement with the classical reference-tracking formulation. The evaluation was performed on measured states and applied control inputs, so that the resulting trajectory of $J(k)$ reflects the actual closed-loop behavior of the plant rather than the predicted optimum of the underlying quadratic program.

The upper subplot of the Figure 23 reveals three distinct phases. During the initial transient ($t \in [0, 2]$ s), the cost decreases by nearly two orders of magnitude, dropping from $J(0) \approx 7000$ to a residual value below 150. The subsequent regime exhibits a bounded steady-state behavior in which $J(k)$ remains confined within a small neighborhood of zero, punctuated by mild periodic excursions of approximately 250 units occurring at $t \approx 5$, 10, and 15 s. These excursions are consistent with the geometry of the reference: the lemniscate exhibits regions of maximal curvature at its central crossing and outer apices, where the tracking task becomes most demanding due to abrupt changes in the desired velocity vector. The periodicity, magnitude, and phase of these features are reproducible across the ten experimental runs, as evidenced by the narrow $\pm 2\sigma$ confidence bands surrounding the mean trajectory.

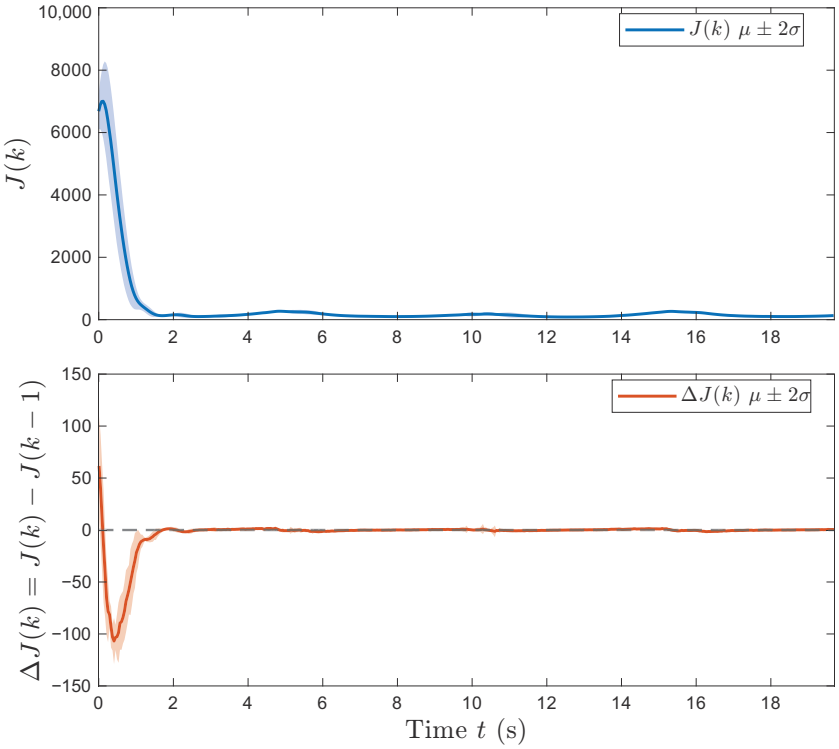


Figure 23. Cost function evolution.

The increment $\Delta J(k)$, displayed in the lower subplot, provides the more discriminating diagnostic. A short-lived positive lobe is observed during the first $\sim$0.4 s of operation, attributable to the physical inertia of the actuators; as the reference begins to evolve from rest, the cart cannot instantaneously match the prescribed velocity profile, causing the tracking error to grow over a small number of samples before the controller asserts effective authority. This phenomenon is intrinsic to any sampled-data implementation operating on real hardware and does not constitute a violation of the Lyapunov-type descent property of MPC, which is rigorously established for the value function $V(x(k)) = J^\star(x(k))$ associated with the optimal control sequence rather than for the realized cost computed from measured signals. Once the actuators reach an operating regime compatible with the reference dynamics, $\Delta J(k)$ becomes strongly negative, attaining values of approximately -110 during the dominant decay phase. After convergence to the steady-state regime, $\Delta J(k)$ oscillates in

https://doi.org/10.3390/act15050260

a tight neighborhood of zero, with both positive and negative excursions remaining below 5% of the peak descent rate observed during the transient.

The aggregate behavior is fully consistent with the Lyapunov-based stability arguments that are standard in the predictive control literature. The cost function exhibits an overall monotonic decrease over the transient phase, ultimate boundedness in the steady-state phase, and a contraction ratio of $J(t_f)/J(0) \ll 1$ that quantifies the controller's regulating action. The minor positive excursions of $\Delta J(k)$ in the steady-state regime are explained by two well-understood mechanisms: first, the reference is not an equilibrium of the system but a periodic, non-stationary trajectory, so the cost cannot decay to zero but only to a residual level dictated by the curvature-induced infeasibility of perfect tracking; second, the cost is evaluated on the realized closed-loop trajectory rather than on the optimal predicted one, introducing a non-negative gap whose fluctuation is consistent with observed measurement noise and unmodeled dynamics. Critically, the standard deviation across the ten runs remains an order of magnitude smaller than the mean throughout the entire experiment, confirming that the observed convergence behavior is a structural property of the closed loop rather than a fortunate artifact of any particular trial. Taken together, these observations constitute strong empirical evidence that the proposed FMPC algorithm enforces practical asymptotic stability of the tracking error and exhibits the descent behavior expected of a properly tuned predictive control law.

7. Conclusions

This work presents a hierarchical control architecture for high-precision trajectory tracking of a car-like mobile robot operating under significant disturbances originating from the normal and tangential forces present in the wheels of the vehicle. The proposed framework integrates a low-level nonlinear dynamic controller with a high-level frugal model predictive controller, enabling simultaneous disturbance rejection and accurate path tracking at a prescribed cruise velocity. Experimental results demonstrate that the integration of disturbance-compensating low-level control, predictive high-level planning, and sensor-fusion-based state estimation provides an effective solution for reliable autonomous navigation in challenging environments. Future work will focus on extending the framework to fully onboard sensing systems, adapting the predictive layer to accommodate varying terrain conditions, and exploring cooperative implementations in multi-vehicle robotic systems.

A significant limitation of the current control scheme is its reliance on a kinematic bicycle model, which fails to guarantee stability or feasibility in high-speed maneuvers. To address this, future research will incorporate the bicycle's dynamic model into the MPC framework. This will allow the system to compensate for lateral forces and varying tire-road friction, ensuring stability at higher velocities. Additionally, we aim to investigate a path-following reformulation where the trajectory is parameterized in the spatial domain. By representing the prediction model as a chain of integrators or decoupled double integrators, the MPC can be solved as a quadratic programming (QP) problem, significantly enhancing real-time computational efficiency.

Author Contributions: Conceptualization, L.A.M.-R., R.I.V.-C. and J.F.G.-C.; methodology, J.F.G.-C. and W.F.G.-S.; software, L.A.M.-R.; validation, L.A.M.-R. and R.I.V.-C.; formal analysis, J.F.G.-C.; investigation, G.A.M.-H.; resources, G.M.-A.; data curation, R.I.V.-C.; writing—original draft preparation, L.A.M.-R. and R.I.V.-C.; writing—review and editing, J.F.G.-C. and R.C.A.-L.; visualization, R.I.V.-C.; supervision, R.C.A.-L. and G.M.-A.; project administration, W.F.G.-S.; funding acquisition, G.A.M.-H., G.M.-A. and J.F.G.-C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by Vicerrectoría de Investigación y Estudios de Posgrado de la BUAP (VIEP-BUAP) grant number 00593-PV/2025.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: The first and second authors acknowledge support from the Secretariat of Science, Humanities, Technology, and Innovation (SECIHTI) in Mexico, for supporting their Ph.D. studies at BUAP through a scholarship.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Proof of Proposition 1

Proof. First, note that the variable z_i given in the first equation of (48) has bounded trajectories:

$$|z_i(t)|_\infty \leq M \tag{A1}$$

Now, consider the dynamics of the error variables:

$$\begin{aligned} \dot{e}_x &= \dot{\omega}_i - \dot{\omega}_i^d = \alpha[-\sigma_M(\lambda e_x - z_i) - \xi_i] \\ \dot{e}_z &= \dot{z}_i - \dot{\xi}_i = \lambda_f[-\sigma_M(\lambda e_x - z_i) - z_i] - \dot{\xi}_i \end{aligned} \tag{A2}$$

Since z_i is bounded, it is assured that e_z remains bounded. Now, the trajectories of e_x will be analyzed. For this, the following candidate Lyapunov function is proposed:

$$V = \frac{1}{2}e_x^2 \tag{A3}$$

We derive along the system trajectory,

$$\begin{aligned} \dot{V} &= e_x\dot{e}_x = e_x(-\alpha\sigma_M(\lambda e_x - z_i) - \alpha\xi_i) \\ &= -e_x\alpha\sigma_M(\lambda e_x - z_i) - e_x\alpha\xi_i \end{aligned} \tag{A4}$$

Assuming that $|\lambda e_x| > 2z_i$ then $|\lambda e_x - z_i| \geq z_i + \varepsilon$ with ε small enough. Due to the boundedness of z_i we can discern that

$$|z_i(t)|_\infty < M \tag{A5}$$

and for this reason $|\lambda e_x - M| \geq M + \varepsilon$. This means that $\lambda e_x - M$ has the same sign as λe_x, i.e., the same sign as e_x. Now, considering that $|\xi_i(t)| \leq \xi_{max}$ it must be true that

$$\dot{V} \leq -\alpha|e_x|\sigma_M(M + \varepsilon) + \alpha|e_x|\xi_{max} \tag{A6}$$

Assume that:

$$\xi_{max} < \min(M + \varepsilon, M) = M \tag{A7}$$

It can be ensured that the function decreases, i.e., $\dot{V} < 0$; consequently, the variable e_x enters the set:

$$\Phi = \{e_x : |\lambda e_x| < 2M\} \tag{A8}$$

Upon entering this set, e_x evolves in the linear part of σ_M, therefore the system (A2) becomes

$$\begin{aligned} \dot{e}_x &= -\alpha[\lambda e_x - e_z] \\ \dot{e}_z &= -\lambda_f\lambda e_x - \dot{\xi}_i \end{aligned} \tag{A9}$$

https://doi.org/10.3390/act15050260

Rewriting the system in matrix form, we have

$$\begin{pmatrix} \dot{e}_x \\ \dot{e}_z \end{pmatrix} = \underbrace{\begin{pmatrix} -\alpha\lambda & \alpha \\ -\lambda_f\lambda & 0 \end{pmatrix}}_{A_e} \begin{pmatrix} e_x \\ e_z \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \dot{\xi}_i \tag{A10}$$

To determine the stability of (A10), the eigenvalues of A_e must be analyzed:

$$|s\mathbb{I} - A_e| = \left| \begin{pmatrix} s & 0 \\ 0 & s \end{pmatrix} \begin{pmatrix} -\alpha\lambda & \alpha \\ -\lambda_f\lambda & 0 \end{pmatrix} \right| = \left| \begin{pmatrix} s+\alpha\lambda & -\alpha \\ \lambda_f\lambda & s \end{pmatrix} \right|$$

Thus, the characteristic polynomial is defined as

$$|s\mathbb{I} - A_e| = s^2 + s\alpha\lambda + \alpha\lambda_f\lambda = 0 \tag{A11}$$

Since $\alpha, \lambda, \lambda_f > 0$, then A_e is a Hurwitz matrix. Thus, for the system given by

$$\dot{X}_e = A_e X_e + B\dot{\xi} \tag{A12}$$

the trajectories of the error vector can be bounded by

$$X_e(t) = e^{A_e(t-t_0)} X_e(t_0) + \int_{t_0}^{t} e^{A_e(t-t_0)} B(\dot{\xi}_i) d\tau \tag{A13}$$

$$\begin{aligned} \therefore \|X_e(t)\| \leq & \|e^{A_e(t-t_0)}\| \|X_e(t_0)\| \\ & + \int_{t_0}^{t} \|e^{A_e(t-t_0)}\| \|B\dot{\xi}_i(t)\| dt \\ \leq & \frac{K_0}{\lambda_M} \end{aligned} \tag{A14}$$

where λ_M is the highest eigenvalue $\frac{1}{2}(A_e + A_e^T)$ □

References

1. Chen, Z.; Yin, J.; Farhan, S.M.; Liu, L.; Zhang, D.; Zhou, M.; Cheng, J. A comprehensive review of obstacle avoidance for autonomous agricultural machinery in multi-operational environment. *Artif. Intell. Agric.* **2026**, *16*, 139–163. [CrossRef]
2. Yang, J.; Zulkarnain, N.; Faisal Ibrahim, M.; Najaa Aimi Mohd Nordin, I.; Abdollah Vaghefi, S. A Comprehensive Review of Agricultural Ground Automatic Navigation Systems Based on Multi-Sensor Fusion. *IEEE Access* **2025**, *13*, 168159–168182. [CrossRef]
3. Khodarahmi, M.; Maihami, V. A Review on Kalman Filter Models. *Arch. Comput. Methods Eng.* **2023**, *30*, 727–747. [CrossRef]
4. Alatise, M.B.; Hancke, G.P. A Review on Challenges of Autonomous Mobile Robot and Sensor Fusion Methods. *IEEE Access* **2020**, *8*, 39830–39846. [CrossRef]
5. Shit, R.C. Precise localization for achieving next-generation autonomous navigation: State-of-the-art, taxonomy and future prospects. *Comput. Commun.* **2020**, *160*, 351–374. [CrossRef]
6. Wang, S.; Ahmad, N.S. A Comprehensive Review on Sensor Fusion Techniques for Localization of a Dynamic Target in GPS-Denied Environments. *IEEE Access* **2025**, *13*, 2252–2285. [CrossRef]
7. Ahmed Amin, A.; Mubarak, A.; Manzoor, H.U. Design of intelligent vehicular and sensor communication network: A comprehensive survey. *Syst. Sci. Control Eng.* **2025**, *13*, 2529187. [CrossRef]
8. Souza De Oliveira, C.; Toledo, R.D.S.; Tulux Oliveira Victorio, V.H.; Von Wangenheim, A. Trajectory Planning for Autonomous Cars in Low-Structured and Unstructured Environments: A Systematic Review. *IEEE Access* **2025**, *13*, 48841–48871. [CrossRef]
9. Erfianti, R.; Asfihani, T.; Suhandri, H.F. GNSS/IMU Sensor Fusion Performance Comparison of a Car Localization in Urban Environment Using Extended Kalman Filter. *IOP Conf. Ser. Earth Environ. Sci.* **2023**, *1127*, 012006. [CrossRef]
10. Teng, X.; Shen, Z.; Huang, L.; Li, H.; Li, W. Multi-sensor fusion based wheeled robot research on indoor positioning method. *Results Eng.* **2024**, *22*, 102268. [CrossRef]
11. Siegwart, R. *Introduction to Autonomous Mobile Robots*; Intelligent Robots and Autonomous Agents; MIT Press: Cambridge, MA, USA, 2004.

12. Lynch, K.M.; Park, F.C. *Modern Robotics: Mechanics, Planning, and Control*, 1st ed.; Cambridge University Press: Cambridge, UK, 2017. [CrossRef]
13. Kalman, R.E. A New Approach to Linear Filtering and Prediction Problems. *J. Basic Eng.* **1960**, *82*, 35–45. [CrossRef]
14. Crassidis, J.L.; Junkins, J.L. *Optimal Estimation of Dynamic Systems*; Number 2 in Chapman & Hall CRC Applied Mathematics and Nonlinear Science Series; Chapman & Hall/CRC: Boca Raton, FL, USA, 2004.
15. Zeng, L.; Guo, S.; Zhu, M.; Duan, H.; Bai, J. An Improved Trilateral Localization Technique Fusing Extended Kalman Filter for Mobile Construction Robot. *Buildings* **2024**, *14*, 1026. [CrossRef]
16. Babri, U.M.; Irfan Haider Jillani, M.; Ahmad, R.; Alam, M.M. Tight Coupling of GNSS and IMU using Extended Kalman Filtering. In Proceedings of the 2024 19th Biennial Baltic Electronics Conference (BEC), Tallinn, Estonia, 2–4 October 2024; pp. 1–6. [CrossRef]
17. Huang, Z.; Ye, G.; Yang, P.; Yu, W. Application of multi-sensor fusion localization algorithm based on recurrent neural networks. *Sci. Rep.* **2025**, *15*, 8195. [CrossRef]
18. Wang, Y. Comparison of the effectiveness of Extended Kalman Filter and machine learning algorithms in sensor fusion. In Proceedings of the 4th International Conference on Computer, Internet of Things and Control Engineering, Wuhan, China, 1–3 November 2024; pp. 18–24. [CrossRef]
19. Kumar Reddy Damagatla, R.; Atia, M. Improving EKF-Based IMU/GNSS Fusion Using Machine Learning for IMU Denoising. *IEEE Access* **2024**, *12*, 114358–114369. [CrossRef]
20. Yan, K.; Ma, B. Global Posture Stabilization for the Kinematic Model of a Rear-Axle Driven Car-Like Mobile Robot Considering Obstacle Avoidance. *IEEE Robot. Autom. Lett.* **2023**, *8*, 5568–5575. [CrossRef]
21. Yao, Q.; Tian, Y.; Wang, Q.; Wang, S. Control Strategies on Path Tracking for Autonomous Vehicle: State of the Art and Future Challenges. *IEEE Access* **2020**, *8*, 161211–161222. [CrossRef]
22. Siradjuddin, I.; Prayoga, D.A.; Wibowo, S.; Syaifudin, Y.W.; Wicaksono, R.P.; Achmadiah, M.N. A PID Control Algorithm for Path Tracking of a Nonholonomic Car-Like Robot. In Proceedings of the 2024 IEEE International Symposium on Consumer Technology (ISCT), Kuta, Indonesia, 13–16 August 2024; pp. 305–310. [CrossRef]
23. Trinh, K.L.T.; Nguyen, H.T. Path tracking control for car-like robots by pid controller with time-varying parameters. *Sci. Technol. Dev. J. Eng. Technol.* **2022**, *5*, 1642–1650. [CrossRef]
24. Yang, T.; Bai, Z.; Li, Z.; Feng, N.; Chen, L. Intelligent Vehicle Lateral Control Method Based on Feedforward + Predictive LQR Algorithm. *Actuators* **2021**, *10*, 228. [CrossRef]
25. Mireles-Perez, C.; Cruz-Ortiz, D.; Salgado, I.; Chairez, I. Backstepping second order sliding mode control for a car-like robot. In Proceedings of the 2022 8th International Conference on Control, Decision and Information Technologies (CoDIT), Istanbul, Turkey, 17–20 May 2022; pp. 463–467. [CrossRef]
26. Zhang, D.; Huang, Q.; Lu, Q.; Zhang, H. Prescribed-Time Extended State Observer-Based Model Predictive Control for Wheeled Mobile Robots. *IEEE Trans. Ind. Electron.* **2026**, *73*, 3187–3198. [CrossRef]
27. Wang, Z.; Gong, B. ESO-LQR-Based Path Tracking Control for Car-Like Robots. In Proceedings of the 2025 44th Chinese Control Conference (CCC), Chongqing, China, 28–30 July 2025; pp. 3465–3470. [CrossRef]
28. Zhu, C.; Li, B.; Zhao, C.; Wang, Y. Trajectory Tracking Control of Car-like Mobile Robots Based on Extended State Observer and Backstepping Control. *Electronics* **2024**, *13*, 1563. [CrossRef]
29. Miranda-Colorado, R. Observer-based finite-time control for trajectory tracking of wheeled mobile robots with kinematic disturbances. *ISA Trans.* **2024**, *148*, 64–77. [CrossRef]
30. Miranda-Colorado, R. Sliding-mode control of perturbed wheeled mobile robots with matched disturbances compensation. *Int. J. Dyn. Control* **2025**, *13*, 289. [CrossRef]
31. Al Mahmud, S.; Kamarulariffin, A.; Ibrahim, A.M.; Mohideen, A.J.H. Advancements and Challenges in Mobile Robot Navigation: A Comprehensive Review of Algorithms and Potential for Self-Learning Approaches. *J. Intell. Robot. Syst.* **2024**, *110*, 120. [CrossRef]
32. Shirzadeh, M.; Shojaeefard, M.H.; Amirkhani, A.; Behroozi, H. Adaptive fuzzy nonlinear sliding-mode controller for a car-like robot. In Proceedings of the 2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI), Tehran, Iran, 28 February–1 March 2019; pp. 686–691. [CrossRef]
33. Camacho, E.F.; Bordons, C. *Model Predictive Control*; Advanced Textbooks in Control and Signal Processing; Springer: London, UK, 2007. [CrossRef]
34. Zhu, Z.; Yao, Q.; Shi, Y. Intelligent Vehicle Adaptive Robust Path Tracking Control Strategy Based on MPC and ADRC. *Int. J. Automot. Technol.* **2026**, *27*, 651–670. [CrossRef]
35. Kordi, F.; Mobayen, S.; Rezaalikhani, H.; Nikoukar, J. Hybrid trajectory tracking control of wheeled mobile robots using predictive kinematic control and dynamic robust control. *Asian J. Control* **2025**, *27*, 1203–1223. [CrossRef]
36. Yeom, K. Design of Deep Neural Network Based Model Predictive Controller for a Car-like Mobile Robot. *Int. J. Mech. Eng. Robot. Res.* **2022**, *11*, 606–613. [CrossRef]

37. Bai, G.; Meng, Y.; Liu, L.; Luo, W.; Gu, Q.; Liu, L. Review and Comparison of Path Tracking Based on Model Predictive Control. *Electronics* **2019**, *8*, 1077. [CrossRef]
38. Rosas-Vilchis, A.; De Loza, A.F.; Aguilar, L.T.; Cieslak, J.; Henry, D.; Montiel-Ross, O. Trajectory tracking control for an AutoNOMO vehicle using a decoupling approach. In Proceedings of the 2020 28th Mediterranean Conference on Control and Automation (MED), Saint-Raphaël, France, 15–18 September 2020; pp. 375–380. [CrossRef]
39. Bascetta, L.; Farina, M.; Gabrielli, A.; Matteucci, M. A feedback linearisation algorithm for single-track models with structural stability properties. *Control Eng. Pract.* **2022**, *128*, 105318. [CrossRef]
40. Tiriolo, C.; Lucia, W. A Set-Theoretic Control Approach to the Trajectory Tracking Problem for Input–Output Linearized Wheeled Mobile Robots. *IEEE Control Syst. Lett.* **2023**, *7*, 2347–2352. [CrossRef]
41. Mohan Rayguru, M.; Rajesh Elara, M.; Ramalingam, B.; Muthugala, M.A.V.J.; Samarakoon, S.M.B.P. A Path Tracking Strategy for Car Like Robots with Sensor Unpredictability and Measurement Errors. *Sensors* **2020**, *20*, 3077. [CrossRef]
42. Yan, K.; Ma, B. A unified controller of global trajectory tracking and posture regulation for a car-like mobile robot. *Int. J. Robust Nonlinear Control* **2024**, *34*, 8590–8604. [CrossRef]
43. De Luca, A.; Oriolo, G.; Samson, C. Feedback control of a nonholonomic car-like robot. In *Robot Motion Planning and Control*; Thoma, M., Laumond, J.P., Eds.; Springer: Berlin/Heidelberg, Germany, 1998; Volume 229, pp. 171–253. [CrossRef]
44. Siciliano, B.; Sciavicco, L.; Villani, L.; Oriolo, G. *Robotics*; Advanced Textbooks in Control and Signal Processing; Springer: London, UK, 2009. [CrossRef]
45. Pereira, B.S.; Santos, T.L.; Conceicao, A.G. A robust feedback-linearization MPC with artificial target for UGVs. *Mechatronics* **2025**, *110*, 103358. [CrossRef]
46. Yildirim, M.; Oksuztepe, E.; Kurum, H. Design of Electronic Differential System for an Electric Vehicle with Four Independently Controlled In-Wheel PMSM. *Adv. Electr. Comput. Eng.* **2024**, *24*, 23–32. [CrossRef]
47. Julio-Rodríguez, J.D.C.; Santana-Díaz., A.; Ramirez-Mendoza, R.A. Individual Drive-Wheel Energy Management for Rear-Traction Electric Vehicles with In-Wheel Motors. *Appl. Sci.* **2021**, *11*, 4679. [CrossRef]
48. Reséndiz Barrón, A.J.; Flores, Y.J.; García-Rodríguez, F.J.; Medina, A.; Serrano Huerta, D.A. Instrumentation of an Electronic–Mechanical Differential for Electric Vehicles with Hub Motors. *World Electr. Veh. J.* **2025**, *16*, 179. [CrossRef]
49. Sun, H.; Wang, H. Research on Electric Vehicle Differential System Based on Vehicle State Parameter Estimation. *Vehicles* **2025**, *7*, 80. [CrossRef]
50. Daya, J.L.F.; Sanjeevikumar, P.; Blaabjerg, F.; Wheeler, P.W.; Ojo, J.O. Implementation of Wavelet-Based Robust Differential Control for Electric Vehicle Application. *IEEE Trans. Power Electron.* **2015**, *30*, 6510–6513. [CrossRef]
51. Wang, D.; Gao, Y.; Wei, W.; Yu, Q.; Wei, Y.; Li, W.; Fan, Z. Sliding mode observer-based model predictive tracking control for Mecanum-wheeled mobile robot. *ISA Trans.* **2024**, *151*, 51–61. [CrossRef]
52. Abadi, A.; Ayeb, A.; Labbadi, M.; Fofi, D.; Bakir, T.; Mekki, H. Robust Tracking Control of Wheeled Mobile Robot Based on Differential Flatness and Sliding Active Disturbance Rejection Control: Simulations and Experiments. *Sensors* **2024**, *24*, 2849. [CrossRef]
53. Peng, J.; Xiao, H.; Lai, G. Nonlinear Disturbance Observer Incorporated Model Predictive Strategy for Wheeled Mobile Robot's Trajectory Tracking Control. *Int. J. Control Autom. Syst.* **2024**, *22*, 2251–2262. [CrossRef]
54. Abbasi, S.J.; Lee, S. Enhanced Trajectory Tracking via Disturbance-Observer-Based Modified Sliding Mode Control. *Appl. Sci.* **2023**, *13*, 8027. [CrossRef]
55. Sariyildiz, E. Analysis and Synthesis of Disturbance Observer-Based Digital Robust Motion Control Systems in State Space. *IEEE Trans. Ind. Electron.* **2025**, *72*, 8376–8385. [CrossRef]
56. Balbuena, J.; Sinche, J.; Quiroz, D.; Arce, D.; Villota, E. PlatROB: An open-source, modular, and low-cost hardware platform for mobile robotics and AI education. *HardwareX* **2026**, *25*, e00747. [CrossRef]
57. Li, X.; Patel, S.; Stronzek-Pfeifer, D.; Büskens, C. Indoor Localization for an Autonomous Model Car: A Marker-Based Multi-Sensor Fusion Framework. In Proceedings of the 2023 IEEE 19th International Conference on Automation Science and Engineering (CASE), Auckland, New Zealand, 26–30 August 2023; pp. 1–8. [CrossRef]
58. Biondo, L.; Vella, A.D.; Vigliani, A. Development, Implementation and Experimental Assessment of Path-Following Controllers on a 1:5 Scale Vehicle Testbed. *Machines* **2025**, *13*, 1116. [CrossRef]
59. Magallan, G.; De Angelo, C.; Bisheimer, G.; Garcia, G. A neighborhood electric vehicle with electronic differential traction control. In Proceedings of the 2008 34th Annual Conference of IEEE Industrial Electronics, Orlando, FL, USA, 10–13 November 2008; pp. 2757–2763. [CrossRef]
60. Vásquez-Cruz, R.I.; Castellanos-Velasco, E.; Guerrero-Castellanos, J.F. Frugal model predictive control and active disturbance rejection for laser beam steering systems. *Control Theory Technol.* **2025**, *23*, 513–528. [CrossRef]
61. Vásquez-Cruz, R.; Castellanos-Velasco, E.; González-Díaz, V.; Castañeda-Camacho, J.; Guerrero-Castellanos, J. Frugal MPC for quadrotor trajectory control. *Mem. Congr. Nac. Control Autom.* **2025**, *8*, 439–444. [CrossRef]

62. Martínez-Ramírez, L.; Guerrero-Castellanos, J.; Ambrosio-Lázaro, R.; Mino-Aguilar, G.; Sánchez-Santana, J.; Munoz-Hernandez, G. Trajectory tracking control with data fusion and geometric algorithm. *Mem. Congr. Nac. Control Autom.* **2025**, *8*, 433–438. [CrossRef]
63. Alamir, M. *Stabilization of Nonlinear Systems Using Receding-Horizon Control Schemes*; Lecture Notes in Control and Information Sciences; Springer: London, UK, 2006; Volume 339. [CrossRef]
64. Brown, R.G.; Hwang, P.Y.C. *Introduction to Random Signals and Applied Kalman Filtering: With MATLAB Exercises*, 4th ed.; John Wiley & Sons: Hoboken, NJ, USA, 2012.
65. Nugraha, M.H.; Abdul, F.; Bramantyo, L.; Rijanto, E.; Saputra, R.P.; Mahendra, O. Mobile Robot Localization via Indoor Positioning System and Odometry Fusion. In Proceedings of the 2024 IEEE International Conference on Smart Mechatronics (ICSMech), Yogyakarta, Indonesia, 19–21 November 2024; pp. 1–6. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

actuators

MDPI

Article

Kinematic Decoupling and α-TDE-NTSM Control for Single-Tendon-Driven Manipulators

Fei Yan *, Jianhua Li, Huawei Han, Qiwang Xu and Linfeng Hu

School of Electromechanical and Automotive Engineering, Yantai University, Yantai 264005, China; 15098679801@s.ytu.edu.cn (J.L.); hulinfeng@s.ytu.edu.cn (L.H.)
* Correspondence: yanfei@ytu.edu.cn

Abstract

Tendon-driven manipulators possess obvious advantages compared to rigid-link manipulators, such as lighter weight, greater flexibility, and adaptability to confined spaces. To solve the problems of backlash and improve the accuracy of motion in specific application environments, this paper proposes a novel single-tendon-driven design for each joint of the manipulator. Kinematic modeling of the manipulator is systematically derived. Then, a decoupling algorithm is designed to mitigate motion coupling effects and enable accurate mapping between motor inputs and joint motions. Moreover, to improve the accuracy of trajectory tracking control for the tendon-driven manipulator, this paper proposes a nonsingular terminal sliding mode (NTSM) control scheme based on time-delay estimation (TDE). TDE is used to estimate unknown disturbances. An adjustable parameter was introduced based on TDE technology, which can enhance the system's robustness against uncertainties and external disturbances. The stability of the closed-loop control system is verified through Lyapunov stability theory. Finally, decoupling experiments are conducted to validate the kinematic model and the feasibility of the proposed design. And comparative experiments are performed to prove the advantages of the proposed control scheme.

Keywords: tendon-driven manipulator; motion coupling; kinematic modeling; trajectory tracking control; nonsingular terminal sliding mode (NTSM); time-delay estimation (TDE)

1. Introduction

The tendon-driven technique is widely applied in wearable robots [1–3], parallel robots [4–6] and serial robots [7–9]. Cables are usually lightweight and suitable for long-distance motion transmission, properties that provide them greater potential in robot design [10,11]. When designing wearable robots, designers can place motors or other driving units on the base to free users from the load of the driving motors [12,13]. Compared with rigid links, cables are more convenient for driving parallel robots than rigid links, and consequently, they are indispensable for parallel robots to some extent [14]. For serial robots, the robot arm will be big and heavy when motors are installed on the robot arm, because motors and speed reducers take up large amounts of space and weight. Continuum robots are a special kind of serial robot whose shape resembles that of a snake. Space is greatly limited in continuum robots, so they can only place motors at the base with tendon-driven techniques applied [15]. Various types of tendon-driven serial robots have been designed for different purposes. For instance, Martin Varga et al. investigated three different structures of continuum robots, namely, one type of elastic module and two types

of rigid modules. They carried out experimental research and showed the differences in motions between these structures [16].

Multi-vertebrae continuum structure is commonly used in continuum manipulator design. The constant curvature assumption is usually used in the modeling of continuum robots, which is, however, not theoretically precise. Multi-vertebrae continuum robots (MVCRs) are also called multi-backbone continuum robots, whose curvature is constant. In other words, one can precisely calculate the dynamic model of a multi-vertebrae continuum robot without theoretical error. MVCRs offer more compliance than rigid-link ones and improve reliability over soft ones. Xinge Li et al. designed an MVCR with a large extension ratio. They connected two extensible sections to build a series, with each section equipped with intermediate constraints. The structure was verified theoretically and experimentally [17]. A multi-vertebrae continuum robot can be made from a super-elastic NITI alloy. To improve the unevenness of the corners, elastic beams are used on each joint of the manipulator. After building the kinetic model, geometric constraint equations and force balance equations, such a Continuum Robot, can be used as minimally invasive surgical instruments [18]. A serial S-shaped backbone structure of a continuum robot was proposed by Kaidi Zhu et al., which can be manufactured by 3D-printing techniques. A serial S-shaped backbone can greatly reduce the deformation of the continuum robot. Experiments have been done to verify the effectiveness of the S-shaped backbone structure [7].

Extensible single-section soft robots are also found in the existing literature [19–21]. This type of robot is bio-inspired, featuring a honeycomb pattern similar to that of a cell or a honeycomb. This type possesses an important advantage in mechanical performance: it has high energy absorption potential. Ibrahim A. Seleem et al. provided a practical example of extensible single-section soft robots. Their robot is designed using a flat metal sheet whose pattern is distributed vertically and horizontally. The robot can perform compression, extension and bending under the contraction of cables [22]. The magnetic continuum robot is a novel idea in continuum robot design [23,24]. Bentao Zou et al. designed an inchworm-shaped magnetic micro-robot for gastric operations. They enhanced the structure by optimizing the batch fabrication process of pH-responsive hydrogels. An inchworm-shaped magnetic microrobot was designed for gastric operations [25]. Another example of a magnetic continuum robot is the one designed by Tianxiang Liu et al. They proposed a real-time static model to cope with nonlinearity. Moreover, they employed a point dipole model to represent the magnetic field to describe the robot's nonlinear deformation more precisely [26]. Chunbo Wang et al. designed a pre-set stiffness continuum robot for minimally invasive surgery. The robot had an integrated structure to obtain a certain pre-set stiffness with bidirectional push-pull capabilities [27]. Single-tendon actuation makes precise distal manipulation possible. Single-tendon actuation is a promising approach in tendon-driven robots for improving precision in distal control. Previous work used single-tendon-sheath transmission to build a precise model [28]. Inspired by these ideas, this work will use single-tendon actuation to design a rigid-link tendon-driven robot. When it comes to a soft continuum robot, stiffness is very important for its working performance. Jung-Che Chang et al. designed a stiffness-adjustable continuum robot with a 6-DOF continuum section. The robot had a long continuum arm and a 6-DOF continuum tip for industrial applications [29]. Pengyuan Wang et al. designed a continuum manipulator based on a helical structure whose stiffness could be changed. They used soft pneumatic actuators to improve the flexibility and payload-to-weight ratio [30].

Some scholars and engineers are devoted to solving motion decoupling in tendon-driven robots in different ways [31]. Noncircular pulleys provide one approach to compensating for motion coupling with small errors. Based on this idea, one can characterize the pulley profile and design the mechanical structure of the decoupling system [32]. Another

https://doi.org/10.3390/act15050271

way to handle the motion-coupling phenomenon is by using movable pulleys. Translational compensation and rotational compensation are designed for different types of rotation joints, including roll, pitch and yaw. Driving cables are used to ascertain the position of movable pulleys and bear the axial force exerted on them [33]. Tendon-driven modules are also used as a linear decoupling mechanism. By carefully designing the relationship between the motor and the cable, linear transmission between the motors and the joints can be realized. Motion deformation in the driving cables can be compensated for by establishing a static model of the manipulator [34]. Regardless of the decoupling method used, the primary principle is to avoid entanglement, as it can lead to friction and even damage the cable transmission system. Existing decoupling units avoid potential entanglement and effectively solve the coupling in tendon-driven robots. However, decoupling mechanisms or units add extra mass to the robot arm, which significantly increases the moving mass of the tendon-driven system. Heavy arms decrease the payload-to-weight ratio and increase the energy consumption of tendon-driven robots.

Many scholars use flexible dynamics to set up systems, which can provide high tracking control performance. In most experiments, the time-delay control (TDC) scheme is a useful method to utilize past system data to estimate current uncertainties [35]. The time-delay estimation (TDE) technique, as the core of TDC, not only provides a model-free structure but also approximates the unknown dynamics and uncertainties of the robot [36,37]. In order to improve the effectiveness of systems, many experiments have been made with the TDE technique. To reduce the tracking error of a three-axis parallel mechanism with strong coupling, nonlinearity and time-varying uncertainties, Zhou et al. designed a model-free decoupled and robust repetitive controller using time-delay estimation (TDE) [38]. Alsaied et al. designed a model-free control approach using time-delay estimation (TDE) combined with an optimized Type-2 fuzzy logic controller, and the control approach is used for gait trajectory tracking of a lower-limb exoskeleton robot [39]. Alqaisi et al. integrated time-delay estimation (TDE) with feedback linearization and sliding mode control, and the TDE is used to estimate nonlinear disturbances [40]. For the purpose of reducing contour errors of machine tools, Dai et al. designed a time-delay-based sliding mode control method incorporating contour error pre-compensation [41]. In [42], a novel adaptive sliding mode control scheme was proposed for a modular self-reconfigurable spacecraft. The control scheme compensates for coupling terms and uncertainties by using TDE. To control the position of a coaxial rotor drone with unknown model dynamics, an optimal model-free fuzzy control algorithm based on TDE was proposed. TDE is used to estimate the unknown dynamic function [43]. Furthermore, the TDE technique has been widely applied in the field of robotic manipulator control [44–46].

In this paper, a novel single-tendon-driven manipulator is designed. To solve the motion coupling problems for the manipulator, this paper builds a kinematic model by developing a coupling model and a decoupling algorithm. Then, a nonsingular terminal sliding mode control scheme is proposed based on α-TDE for trajectory tracking. We will try to make some contributions as follows:

1. Design a tendon-driven manipulator and describe how the proposed manipulator's tendons keep a positive tension force.
2. Calculate the kinematic model and propose a decoupling algorithm.
3. Design a novel nonsingular terminal sliding mode control scheme based on α-TDE.
4. Validate the feasibility and effectiveness of the above method through experiments.

The rest of the structure of this paper is as follows. The method of manipulator design is given in Section 2. Section 3 describes the theoretical knowledge of decoupling. The system dynamics and main control theory results are given in Section 4. Afterward, the

https://doi.org/10.3390/act15050271

decoupling experiment and comparative experiment are presented in Section 5. Finally, Section 6 concludes this paper.

2. Manipular Design

2.1. Overall Structure

As shown in Figure 1, the manipulator is mounted on a vehicle that houses all the driving units and supporting equipment. For joint 1, a tendon stretches out of the vehicle and drives the winch of joint 1, which drives link 1 directly. The situation becomes more complex for joint 2 because the tendon should pass through joint 1 before driving the winch of joint 2. Some pulleys are set on the axle of joint 1 to locate tendons that pass through it. A tendon is set to stretch out of the vehicle and pass through joint 1, and finally drive the winch of joint 2. A tendon goes the same way to drive the winch of joint 3. Two pulleys are set on the axle of joint 1 and joint 2, respectively. A tendon stretches out of the vehicle, passes through the two pulleys, and eventually drives the winch of joint 3. The winch drives the corresponding axle and then drives link 3 to finish the whole force transmission.

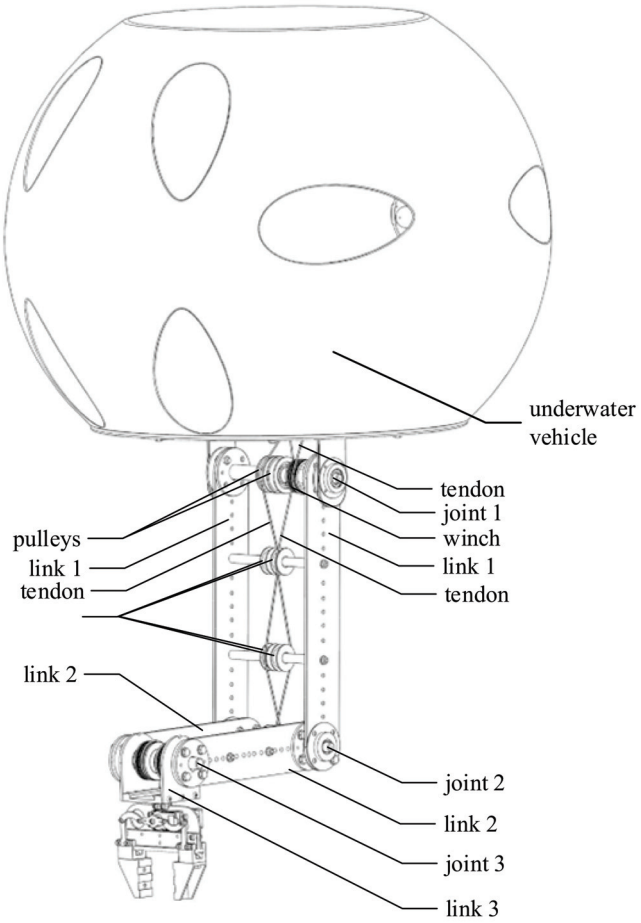


Figure 1. Overall structure of the newly designed tendon-driven manipulator.

2.2. Tendon Routing

A diagram of cable routing is presented in Figure 2, where some parts of the tendon-driven manipulator are moved for clarity. As shown, joint 1 is driven by a single tendon from the winch of motor 1. The tendon can drive joint 1 to rotate in a clockwise direction when the motor is energized to pull the tendon. Meanwhile, assuming gravity is downward, the first joint tends to rotate counter-clockwise under gravity, which ensures the tendon of joint 1 remains in positive tension.

https://doi.org/10.3390/act15050271

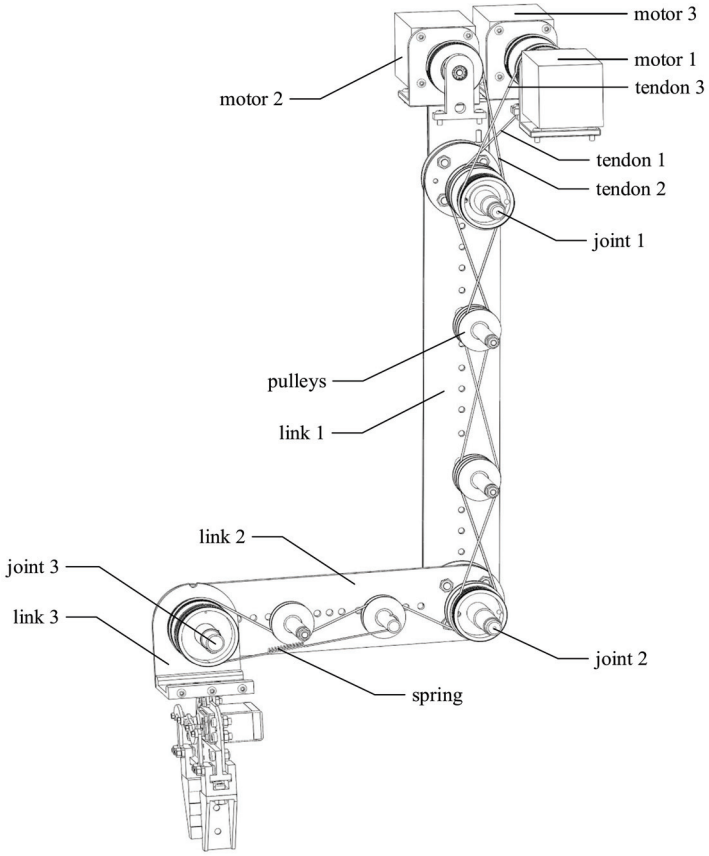


Figure 2. Diagram of cable routing of tendon-driven manipulator.

Consider a simplified 2-DOF tendon-driven manipulator, as shown in Figure 3. Based on the above analysis, the tendon of the first joint is always in positive tension when the arm is on the left side. However, gravity alone is often insufficient to maintain constant positive tension in the tendons when subjected to external disturbances. Fortunately, the driving tendon of joint 2 helps to keep tendon 1 in positive tension.

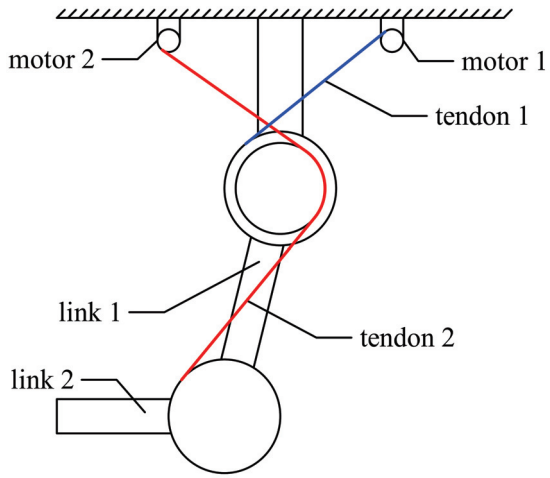
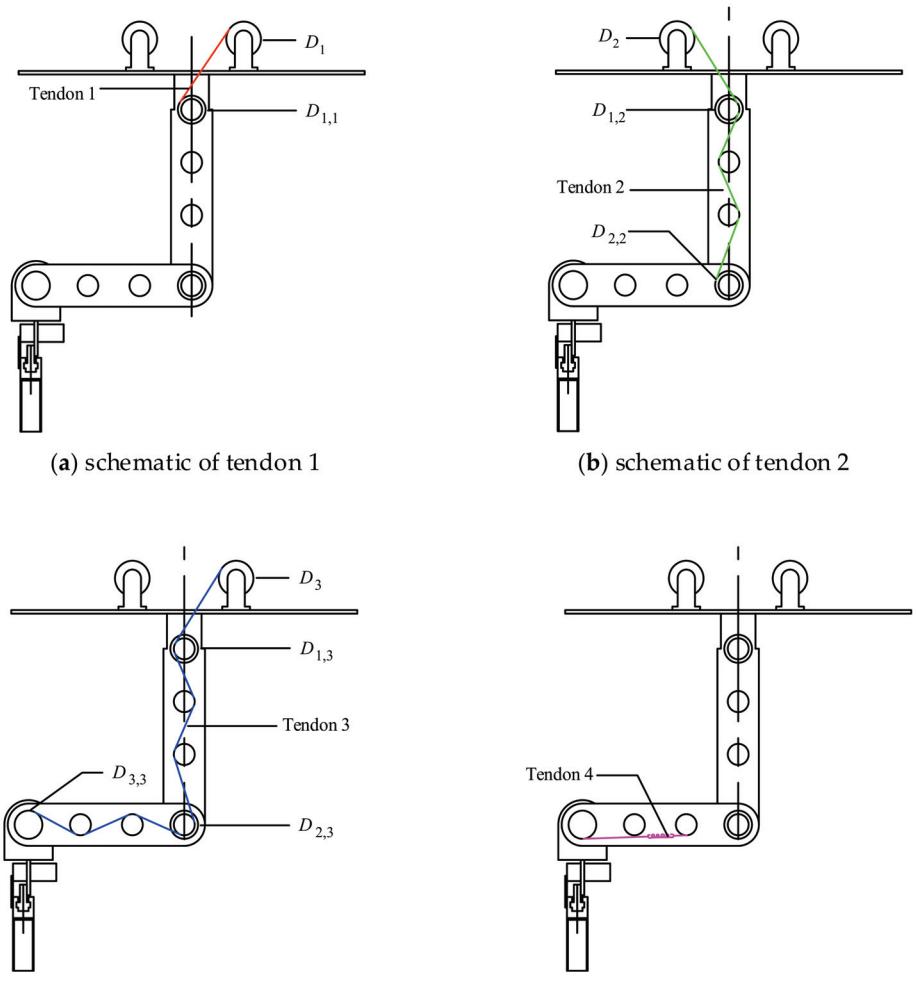


Figure 3. An example of a simplified 2-DOF tendon-driven manipulator.

For joint 2 and tendon 2, suppose the hinge of joint 1 is locked, and motor 2 is energized to pull the tendon. Under these circumstances, tendon 2 tends to shorten and wind onto the winch of motor 2, which can be achieved in two ways. The first way is straightforward: joint 2 rotates clockwise. The second way is that joint 1 rotates counter-clockwise. In other words, pulling tendon 2 provides joint 1 a rotating tendency in a counter-clockwise direction.

https://doi.org/10.3390/act15050271

To discuss tendon routing and how to keep all tendons in positive tension, each tendon in Figure 2 is picked out and plotted in Figure 4. From previous analysis, tendon 1 is driven by motor 1, which can drive joint 1 to rotate in a clockwise direction. Tendon 2 passes through joint 1 and causes it to rotate counter-clockwise when pulled by the motor. Therefore, these two tendons provide joint 1 with two opposing torques, which guarantee that tendon 1 remains in positive tension. The same thing happens on other tendons and joints, but slightly differently on the last joint. Joint 2 is driven by tendon 2, which makes joint 2 rotate clockwise. Similarly, tendon 3, which passes through joint 2, makes joint 2 rotate counterclockwise. Therefore, two torques in different directions ensure the feasibility of joint 2. When it comes to joint 3, however, no tendon passes through this joint because it is the distal end. Tendon 3 drags joint 3 to rotate in a clockwise direction, and tendon 4, with a spring, is used to drag joint 3 to rotate in the opposite direction. A spring has a notable disadvantage: its force varies significantly during operation. Too much preload burdens the motors and increases energy consumption. Therefore, this work finds an alternative to tighten the tendons that uses another motor and tendon instead of the spring, as shown in Figure 5. Tendon 4 passes through joint 1 and joint 2 and connects to a driving motor, which can provide a consistent force. This can be regarded as an improvement because the preload can be adjusted freely under different working requirements. In addition, gravity also plays a role during motion. By limiting the working envelope of the tendon-driven manipulator, the gravity torque is opposite to the driving torque and helps to keep the tendons in positive tension.



(**a**) schematic of tendon 1 (**b**) schematic of tendon 2

(**c**) schematic of tendon 3 (**d**) schematic of tendon 4

Figure 4. Schematic of tendon routing of tendon-driven manipulator.

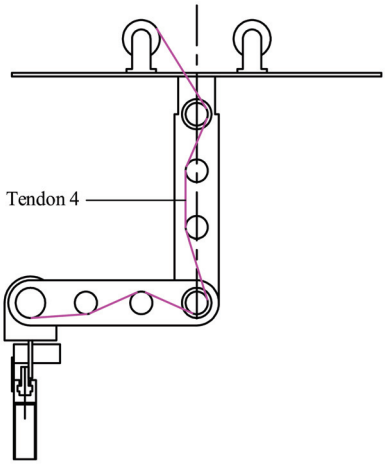


Figure 5. An alternative way to tighten the tendons (schematic of tendon 4).

In this paper, the workspace is divided into two halves, as the manipulator always works in the left area to keep the tendons tight. The left side is referred to as the 'positive' side to describe the cable routing for a multi-DOF tendon-driven manipulator. Tendon 1 winds onto joint 1 on the left of joint 1, the same side as the work area of the manipulator. In this work, tendon 1 is on the positive side of joint 1. In contrast, tendon 2 passes through joint 1 from the opposite side, so we think tendon 2 passes through joint 1 on the negative side. Similarly, tendon 2 is on the positive side of joint 2, and tendon 3 passes through joint 2 on the negative side.

3. Kinematic Model

3.1. Description of Motion-Coupling

A kinematic model is essential for trajectory planning and practical use for a tendon-driven manipulator. Motion-coupling is a unique phenomenon existing in a tendon-driven manipulator, which should be analyzed before establishing the kinematic model. First of all, let us make the following assumptions.

Assumption 1. *Tendons remain at constant length during operation.*

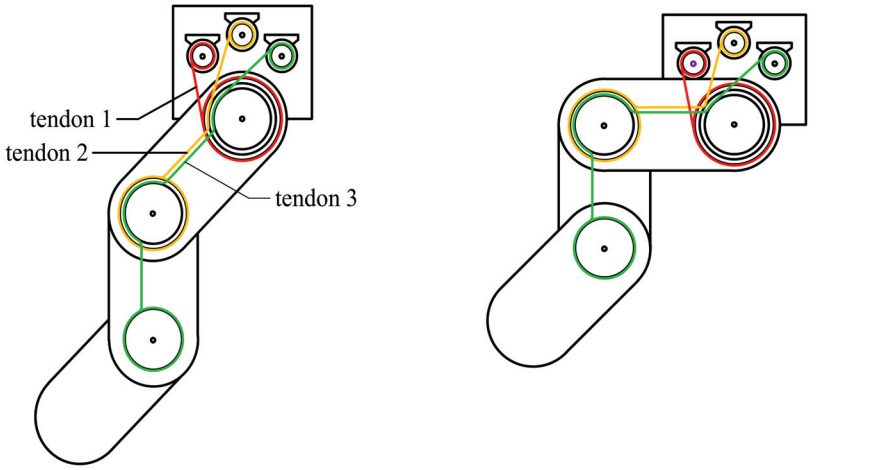
Assumption 2. *Tendons maintain positive tension during operation and never disengage from the pulleys.*

Assumption 3. *The links of the tendon-driven manipulator will not change their shape during work.*

Assumption 4. *If a tendon winds onto a winch, the tendon will not slip relative to the winch.*

For a traditional direct-driven manipulator, one joint's rotation will not change another joint's position. For a tendon-driven manipulator, however, joints near the base will influence the joints near the distal end. As shown in Figure 6, tendons 1, 2 and 3 are depicted in red, orange and green, respectively. From the posture in Figure 6a to that in Figure 6b, only motor 1 rotates to pull tendon 1. Motor 2 holds stable in this procedure, but joint 2 rotates in the opposite direction, which makes link 2 seem to only translate without rotating. This phenomenon is called motion-coupling of a tendon-driven manipulator.

https://doi.org/10.3390/act15050271



(**a**) one posture of the manipulator (**b**) another posture of the manipulator

Figure 6. Posture change after a motor rotation for tendon-driven manipulator.

3.2. Notations

The aim of dynamic modeling is to calculate the position of the end-effector using the rotation angles of the motors. First, the rotational relationship between the motors and the joints should be established. The first joint, numbered joint 1, will not be influenced by other joints and is only determined by the rotation angle of motor 1. For the sake of simplicity and without loss of generality, when a motor pulls the tendon and makes the tendon wind onto the winch, the rotating direction of the winch is considered positive.

Let D_1 denote the diameter of the winch connected to motor 1, D_2 as that for motor 2, and D_3 as that for motor 3. In order to give a general solution of an n-DOF tendon-driven manipulator, D_n is used to represent the diameter of the winch of motor n. There are several pulleys and a winch on the axle of joint 1. The winch is used to drive joint 1, whose diameter is denoted by $D_{1,1}$. Tendon 2 passes through joint 1, and a pulley is required to guide it, whose diameter is denoted by $D_{1,2}$. Here, 1 means the pulley is located at joint 1, and 2 represents that tendon 2 is guided by the pulley. The diameter of other pulleys and winches will be denoted as $D_{p,q}$ by the same rule. The first subscript, p, indicates the joint at which the pulley is located, and the second subscript, q, indicates the tendon that is guided by the pulley. Specifically, when p is equal to q, $D_{p,p}$ is the diameter of the winch that drives joint p.

For an n-DOF tendon-driven manipulator, the rotation angles of motors are denoted by $\mathbf{Q} = [\ q_1 \quad q_2 \quad \cdots \quad q_n\ ]^{\mathrm{T}}$, and those of joints are denoted by $\mathbf{\Theta} = [\ \theta_1 \quad \theta_2 \quad \cdots \quad \theta_n\ ]^{\mathrm{T}}$. Let q_i represent the rotation angle of motor i, and θ_i represent the rotation angle of joint i.

3.3. Mathematical Expression of Motion-Coupling

Referring back to Figure 4, first consider joint 1. It can be found that $q_1 D_1 = \theta_1 D_{1,1}$, because the speed of the tendon is constant everywhere. Forwardly, the rotation angle of joint 2 is not only decided by motor 1, but also influenced by joint 1, i.e., the rotation angle of joint 2 is the sum of the above-mentioned two factors. From Figure 4a,b, we can analyze the length variation in tendon 2 caused by joint 1. The length of the tendon between two pulleys remains constant because tendons are tangent lines of two pulleys. The distance between two pulleys will not change when joints and links rotate. Therefore, when joint 1 rotates, the length variation in tendon 2 is only the amount that tendon winds onto or out of the corresponding pulley. The length change in tendon 2 caused by the rotation of joint 1 is proportional to the rotation angle of joint 1. In addition, we should consider whether the variation is positive or negative. We can see that tendon 1 and tendon 2 wind in opposite

https://doi.org/10.3390/act15050271

directions with respect to joint 1. When tendon 1 moves in a positive direction, i.e., motor 1 drags tendon 1, tendon 2 winds onto the pulley, and tendon 2 drags joint 2. Consequently, when joint 1 moves in a positive direction, joint 2 moves in a positive direction, so $c_{1,2} = 1$ depicts a positive correlation. From Figure 4c, tendon 3 passes through joint 1 from its left side, so $c_{1,3} = -1$ describes a negative correlation. Mathematically, $\Delta_{\text{tendon 2}} = c_{1,2}\theta_1 D_{1,2}$ is used to show the coupling amount of tendon 2 when it passes through joint 1. If a joint rotates clockwise from the view of Figure 4, we think the joint rotates in a positive direction. As shown in this figure, when tendons are pulled, all joints rotate clockwise and $c_{1,1} = c_{2,2} = c_{3,3} = 1$ are given to describe it. For the joint n of a tendon-driven manipulator, the following relationship exists.

$$q_n D_n + \sum_{i=1}^{n-1} c_{i,n}\theta_i D_{i,n} = c_{n,n}\theta_n D_{n,n} \tag{1}$$

The motion-coupling can be re-expressed in matrix form, i.e., $\boldsymbol{\Theta} = \mathbf{SQ}$. Usually, we hope to obtain the joints' rotation angles to locate the end-effector, and the motion-coupling matrix will be helpful. The contents of $\mathbf{S}$ can be obtained by rearranging the above-mentioned formulas and are given as follows.

$$\mathbf{S}^{-1} = \begin{bmatrix} \frac{c_{1,1}D_{1,1}}{D_1} & 0 & \cdots & 0 \\ -\frac{c_{1,2}D_{1,2}}{D_2} & \frac{c_{2,2}D_{2,2}}{D_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{c_{1,n}D_{1,n}}{D_n} & -\frac{c_{2,n}D_{2,n}}{D_n} & \cdots & \frac{c_{n,n}D_{n,n}}{D_n} \end{bmatrix} \tag{2}$$

$\mathbf{S}^{-1}$ is a lower triangular matrix whose elements on the diagonal are $c_{n,n}D_{n,n}/D_n$, which presents the transmission ratio between a motor and the joint it drives. $c_{n,n}D_{n,n}/D_n$ is non-zero, so $\mathbf{S}$ can be obtained by matrix inversion, which is also a lower triangular matrix. The elements of $\mathbf{S}^{-1}$ below the diagonal represent the influence of motion coupling, which decreases with D_n increasing and $D_{i,n}$ decreasing ($i < n$).

Remark 1. *We have already taken the manufacturing tolerances of the robotic arm's pulley into account during the modeling process; the diameter of the pulley used can be accurately measured with a vernier caliper, with a very small tolerance.*

4. NTSM Scheme Design with TDE

4.1. System Dynamics Scheme

The dynamic model for an n-DOF single-tendon-driven manipulator can be described as

$$\boldsymbol{\tau}_m(t) - \boldsymbol{\tau}_s(t) = \mathbf{I}\ddot{\boldsymbol{\theta}}(t) + \mathbf{D}_m\dot{\boldsymbol{\theta}}(t) \tag{3}$$

$$\begin{aligned} \boldsymbol{\tau}_s(t) = \quad & \mathbf{M}(\mathbf{q}(t))\ddot{\mathbf{q}}(t) + \mathbf{C}(\mathbf{q}(t), \dot{\mathbf{q}}(t))\dot{\mathbf{q}}(t) \\ & + \mathbf{G}(\mathbf{q}(t)) + \mathbf{F}(\mathbf{q}(t), \dot{\mathbf{q}}(t)) + \boldsymbol{\tau}_d(t) \end{aligned} \tag{4}$$

$$\boldsymbol{\tau}_s(t) = \mathbf{K}_s(\boldsymbol{\theta}(t) - \mathbf{q}(t)) + \mathbf{D}_s(\dot{\boldsymbol{\theta}}(t) - \dot{\mathbf{q}}(t)) \tag{5}$$

where $\mathbf{q}(t)$ and $\boldsymbol{\theta}(t)$ stand for the positions of the joints and motors. $\mathbf{I}$ and $\mathbf{D}_m$ are the motor's inertia and damping matrices. $\mathbf{M}(\mathbf{q}(t))$ is the mass-inertia matrix of the manipulator. $\mathbf{C}(\mathbf{q}(t), \dot{\mathbf{q}}(t))$ is the Coriolis matrix. $\mathbf{G}(\mathbf{q}(t))$ and $\mathbf{F}(\mathbf{q}(t), \dot{\mathbf{q}}(t))$ stand for the gravitational vector and friction vector. $\boldsymbol{\tau}_d(t)$ is the lumped unknown disturbance. $\boldsymbol{\tau}_m(t)$ is the motor input torque vector. $\boldsymbol{\tau}_s(t)$ is the joint compliance torque vector. $\mathbf{K}_S$ and $\mathbf{D}_S$ are the joint stiffness and damping matrices.

https://doi.org/10.3390/act15050271

Combining (3) with (4) and using a constant matrix $\overline{\mathbf{M}}$, the following equation can be obtained.

$$\overline{\mathbf{M}}\ddot{\mathbf{q}}(t) + \mathbf{H}(t) = \boldsymbol{\tau}_m \tag{6}$$

where $\overline{\mathbf{M}}$ is a constant parameter. $\mathbf{H}(t)$ describes the unknown dynamics of the single-tendon-driven manipulator as

$$\begin{aligned} \mathbf{H}(t) = \quad & (\mathbf{M}(\mathbf{q}(t)) - \overline{\mathbf{M}})\ddot{\mathbf{q}}(t) + \mathbf{C}(\mathbf{q}(t), \dot{\mathbf{q}}(t)) \\ & + \mathbf{G}(\mathbf{q}(t)) + \mathbf{F}(\mathbf{q}(t), \dot{\mathbf{q}}(t)) \\ & + \mathbf{I}\ddot{\theta}(t) + \mathbf{D}_m(t) + \boldsymbol{\tau}_d(t) \end{aligned} \tag{7}$$

As shown in (7), $\mathbf{H}(t)$ is very complex and difficult to obtain using traditional methods. To obtain the information of $\mathbf{H}(t)$, the TDE technique is employed as follows

$$\hat{\mathbf{H}}(t) \cong \mathbf{H}(t) = \boldsymbol{\tau}_m(t - L) - \overline{\mathbf{M}}\ddot{\mathbf{q}}(t - L) \tag{8}$$

where L is the delay time.

First, we employ the TDC scheme with TDE to set up a control law as follows

$$\boldsymbol{\tau}_m(t) = \overline{\mathbf{M}}(\ddot{\mathbf{q}}_d(t) + K_d\dot{\mathbf{e}}(t) + K_p\mathbf{e}(t)) + \hat{\mathbf{H}}(t) \tag{9}$$

where $\mathbf{e}(t) = \mathbf{q}_d(t) - \mathbf{q}(t)$ is the tracking error. K_P and $K_d > 0$ are control parameters. From Equation (9), a model-free scheme is designed without using system dynamics.

Substituting (9) into (6), the results are as follows

$$\mathbf{H}(t) - \hat{\mathbf{H}}(t) = \ddot{\mathbf{e}}(t) + K_d\dot{\mathbf{e}}(t) + K_p\mathbf{e}(t) \tag{10}$$

If using the appropriate K_P and K_d, the control errors $\mathbf{e}$ and $\dot{\mathbf{e}}$ can be confined to a certain range.

4.2. NTSM Control Design

Although the TDC control scheme has good results in many practical systems, it also has shortcomings that need to be further improved. In order to have good control performance, the NTSM surface is proposed as follows

$$\mathbf{s} = \dot{\mathbf{e}} + \eta\varphi(\mathbf{e}) \tag{11}$$

where $\eta = diag(\eta_1, \eta_2)$ are the positive control parameters.

The $\varphi(\mathbf{e})$ is designed as

$$\varphi(\mathbf{e}_i) = \begin{cases} \mathrm{sig}(\mathbf{e}_i)^{b_i}, & \widetilde{\mathbf{s}}_i = 0 \vee \widetilde{\mathbf{s}}_i \neq 0,\ |\mathbf{e}_i| \geq \Delta_i \\ \gamma_{1i}\mathbf{e}_i + \gamma_{2i}\mathbf{e}_i^2, & \widetilde{\mathbf{s}}_i \neq 0,\ |\mathbf{e}_i| \leq \Delta_i \end{cases} \tag{12}$$

where $\widetilde{\mathbf{s}}_i = \dot{\mathbf{e}}_i + \eta_i sig(\mathbf{e}_i)^{b_i}$, $i = 1 \sim n$, $0 < b_i < 1$, and $\Delta_i > 0$. γ_{1i} and γ_{2i} are given as follows

$$\begin{aligned} \gamma_{1i} &= (2 - b_i)\Delta_i^{b_i - 1} \\ \gamma_{2i} &= (b_i - 1)\Delta_i^{b_i - 2} \end{aligned} \tag{13}$$

Afterwards, in order to reduce the influence of external disturbances during the reaching phase, the linear reaching law we utilize is as follows

$$\dot{\mathbf{s}} = -\lambda\mathbf{s} \tag{14}$$

https://doi.org/10.3390/act15050271

where $\lambda = diag(\lambda_1, \lambda_2)$ are the positive control parameters.

Compared to the TDE scheme, we proposed the α-TDE scheme. The α-TDE design uses data from time $t - L$ and time $t - 2L$ and a parameter α to estimate the overall uncertainty. The parameter α can improve estimation accuracy and enhance the system's robustness against uncertainty. The formula is as follows

$$\begin{aligned} \hat{\mathbf{H}}(t) \cong \quad & \mathbf{H}(t) = (1-\alpha)[\boldsymbol{\tau}_m(t-L) - \overline{\mathbf{M}}\ddot{\mathbf{q}}(t-L)] \\ & +\alpha[\boldsymbol{\tau}_m(t-2L) - \overline{\mathbf{M}}\ddot{\mathbf{q}}(t-2L)] \end{aligned} \tag{15}$$

where $\alpha \in [0, 1]$ is a constant parameter.

$\ddot{\mathbf{q}}_{t-L}$ and $\ddot{\mathbf{q}}_{t-2L}$ are usually obtained as follows

$$\begin{cases} \ddot{\mathbf{q}}_{t-L} = (\mathbf{q}_{t-L} - 2\mathbf{q}_{t-2l} + \mathbf{q}_{t-3l})/L^2 \\ \ddot{\mathbf{q}}_{t-2L} = (\mathbf{q}_{t-2L} - 2\mathbf{q}_{t-3L} + \mathbf{q}_{t-4L})/L^2 \end{cases} \tag{16}$$

Because acceleration encoders are not commonly used in manipulator control, most experiments obtain the angle signal from a position encoder and then, through numerical differentiation, obtain the angular acceleration. This method has been widely adopted in TDE-based control schemes.

From Equation (16), numerical differentiation often introduces a large noise effect, so choosing an appropriate small control parameter $\overline{\mathbf{M}}$ can effectively solve the problems above.

Combining the designed NTSM surface and reaching law, the proposed control scheme is given as

$$\boldsymbol{\tau}_m(t) = \overline{\mathbf{M}}\mathbf{u}(t) + \hat{\mathbf{H}}(t) \tag{17}$$

$$\mathbf{u}(t) = \ddot{\mathbf{q}}_d + \eta\dot{\varphi}(\mathbf{e}) + \lambda\mathbf{s} \tag{18}$$

The overall control scheme of α-TDE-based NTSM is shown in Figure 7.

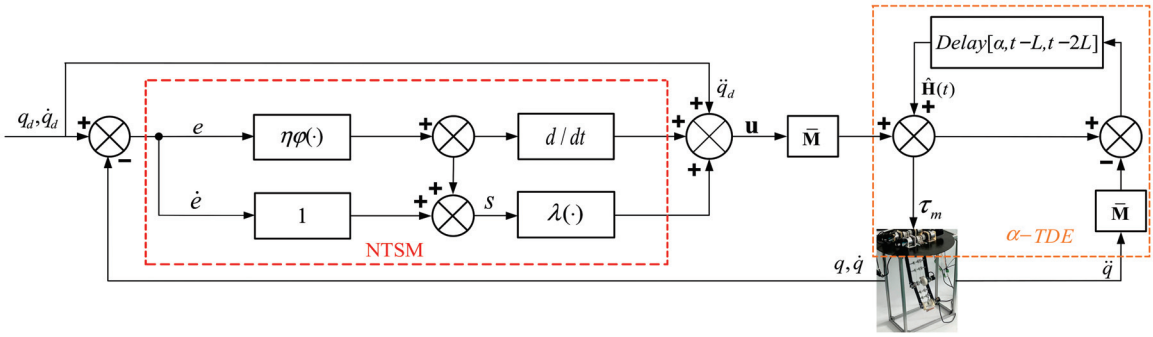


Figure 7. Overall control scheme of α-TDE-based NTSM.

4.3. Stability Analysis

Substituting (18) into (6), we obtain the following

$$\varepsilon = \ddot{\mathbf{e}} + \eta\dot{\varphi}(\mathbf{e}) + \lambda\mathbf{s} \tag{19}$$

where $\varepsilon = \overline{\mathbf{M}}^{-1}(\mathbf{H}(t) - \hat{\mathbf{H}}(t))$ is the estimation error of TDE, which is bounded by $|\varepsilon| \leq \delta$ [47]. The boundedness of the TDE error has been applied to many experimental results.

Choosing the Lyapunov function as

$$V = \frac{1}{2}\mathbf{s}^T\mathbf{s} \tag{20}$$

https://doi.org/10.3390/act15050271

and differentiating (20) along time provides the following

$$\dot{V} = \mathbf{s}^T \dot{\mathbf{s}} = \mathbf{s}^T (\ddot{\mathbf{e}} + \eta \dot{\varphi}(\mathbf{e})) \tag{21}$$

Combining (19) with (21), we have

$$\begin{aligned} \dot{V} &= \mathbf{s}^T (\varepsilon - \lambda \mathbf{s}) \\ &\leq -|\mathbf{s}|^T (\lambda |\mathbf{s}| - |\delta|) \end{aligned} \tag{22}$$

If $\lambda|\mathbf{s}| - |\delta| > 0$ and $\mathbf{s} \neq 0$, Equation (22) will remain $\dot{V} < 0$. Therefore, the manifold of the system converges as

$$|\mathbf{s}| \leq \frac{\delta}{\lambda} = \Omega \tag{23}$$

Case 1: For $\mathbf{s}_i \neq 0$, $|\mathbf{e}_i| \leq \Delta_i$; in this case, the tracking error e is bounded by $|\mathbf{e}_i| \leq \Delta_i = \Delta_e$ and $\varphi(\mathbf{e}_i) = \gamma_{1i}\mathbf{e}_i + \gamma_{2i}{\mathbf{e}_i}^2$.

Then, combining (11) with (23), we obtain the follwing

$$\mathbf{s}_i = \dot{\mathbf{e}}_i + \eta_i(\gamma_{1i}\mathbf{e}_i + \gamma_{2i}{\mathbf{e}_i}^2), |\mathbf{s}_i| \leq \Omega_i \tag{24}$$

Equation (24) can be further derived as

$$\begin{aligned} \left|\dot{\mathbf{e}}_i\right| &= \left|\mathbf{s}_i - \eta_i(\gamma_{1i}\mathbf{e}_i + \gamma_{2i}{\mathbf{e}_i}^2)\right| \\ &\leq |\mathbf{s}_i| + \eta_i\gamma_{1i}|\mathbf{e}_i| + \eta_i\gamma_{2i}|\mathbf{e}_i|^2 \\ &\leq \Omega_i + \eta_i\gamma_{1i}|\Delta_i| + \eta_i\gamma_{2i}|\Delta_i|^2 \end{aligned} \tag{25}$$

Then, substituting (13) into (25), the $\left|\dot{\mathbf{e}}_i\right|$ will converge to

$$\left|\dot{\mathbf{e}}_i\right| \leq \Omega_i + \eta_i {\Delta_e}^b \tag{26}$$

Case 2: For $\mathbf{s} \neq 0$, $|\mathbf{e}_i| \geq \Delta_i$; in this case, $\varphi(\mathbf{e}_i) = sig(\mathbf{e}_i)^{b_i}$, and Equation (24) becomes

$$\mathbf{s}_i = \dot{\mathbf{e}}_i + \eta_i sig(\mathbf{e}_i)^{b_i}, |\mathbf{s}_i| \leq \Omega_i \tag{27}$$

The following equation can be written from (27) as

$$\dot{\mathbf{e}}_i + (\eta_i - s_i[sig(\mathbf{e}_i)^{b_i}]^{-1})sig(\mathbf{e}_i)^{b_i} = 0 \tag{28}$$

When $\eta_i - \mathbf{s}_i[sig(\mathbf{e}_i)^{b_i}]^{-1} > 0$, Equation (28) will remain the NTSM surface. Considering $|\mathbf{s}_i| \leq \Omega_i$, $|\mathbf{e}_i|$ will converge to the following

$$|\mathbf{e}_i| \leq \left(\Omega_i {\eta_i}^{-1}\right)^{\frac{1}{b_i}} \tag{29}$$

Finally, considering the two situations, $|\mathbf{e}_i|$ will converge to the following

$$|\mathbf{e}_i| \leq \Delta_e = \min\left\{\Delta_i, (\Omega_i {\eta_i}^{-1})^{\frac{1}{b_i}}\right\} \tag{30}$$

and $\left|\dot{\mathbf{e}}_i\right|$ will converge to the following

$$\left|\dot{\mathbf{e}}_i\right| \leq \Omega_i + \eta_i {\Delta_e}^b \tag{31}$$

https://doi.org/10.3390/act15050271

5. Experiments

In this section, there are two experiments, including a decoupling experiment and a comparative controller experiment. The decoupling experiment was performed to verify the feasibility and effectiveness of the proposed design for the kinematic model. Then, the comparative controller experiments have been conducted to prove the advantages of the proposed NTSM control scheme based on α-TDE.

5.1. Experimental Setup

The experiments use the manipulator named 'Lumidouce II', which has an SDGA-01C11BD motor. The motor has a rated torque of 0.32 N · m under 3000 rpm rated speed, and it connects to a harmonic reducer whose reduction ratio is 1:100. We collect feedback information through an E6B2-CWZ1X4096P/R encoder, and the resolution of it is 0.022°. Then, we use a Simulink real-time system with an NI PCI-6229 board to run all the controllers. The sampling time is set to 1 ms.

Remark 2. *The sampling period used in this experiment is 1 millisecond, which is commonly used in manipulator control experiments. A smaller sampling period improves the accuracy of time-delay estimation but increases the computational burden on the controller. Conversely, a larger sampling period reduces the computational burden but decreases estimation accuracy.*

5.2. Decoupling Experiment Verification

The manipulator illustrated in Figure 2 was manufactured and is presented in Figure 8. In this design, an extra motor is used to provide tension force instead of a spring. The spring force varies when it changes its length, which makes it impossible to keep a constant tension. In this study, it is assumed that the links of the manipulator do not bend and that the joint hinges have no clearance. However, links change their shape when the applied force increases. Therefore, we try to minimize the influence of force changing by using a motor instead of a spring.

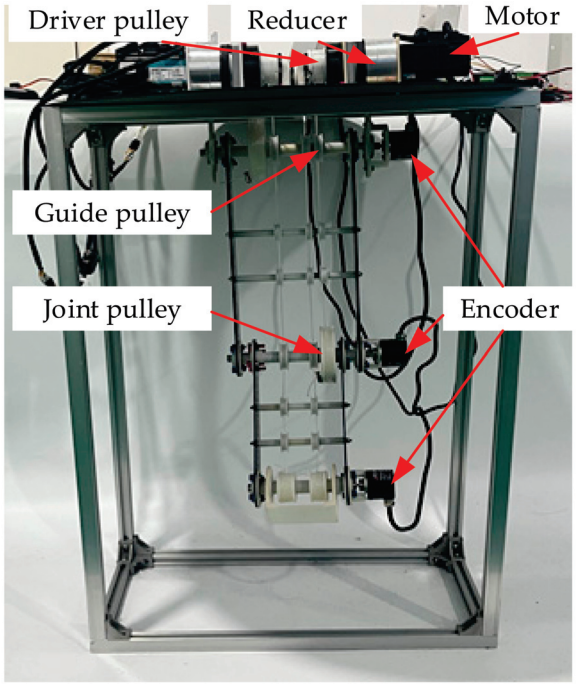


Figure 8. Prototype of designed tendon-driven manipulator.

The elements of the coupling matrix were measured and calculated, as listed in Table 1. The diameters of tendons and pulleys were both taken into account. Using the data in Table 1, the coupling matrix of Equation (2) can be calculated as follows

$$\mathbf{S}^{-1} = \begin{bmatrix} \frac{77}{37} & 0 & 0 \\ -\frac{35}{37} & \frac{77}{37} & 0 \\ \frac{35}{37} & -\frac{35}{37} & 1 \end{bmatrix} \tag{32}$$

Table 1. Elements of the coupling matrix.

Elements of Coupling Matrix	Measure Result
$D_{1,1}/D_1$	77 mm/37 mm
$D_{1,2}/D_2$	35 mm/37 mm
$D_{2,2}/D_2$	77 mm/37 mm
$D_{1,3}/D_3$	35 mm/37 mm
$D_{2,3}/D_3$	35 mm/37 mm
$D_{3,3}/D_3$	37 mm/37 mm

The angular position of the motors is shown in Figure 9, which is used to estimate the position of the joints. The estimated results of joint positions and actual values are both shown in Figures 10–12. The position of joints can be calculated by $\mathbf{\Theta} = \mathbf{SQ}$, in which $\mathbf{S}$ is given by

$$\mathbf{S} = \begin{bmatrix} 0.4805 & 0 & 0 \\ 0.2184 & 0.4805 & 0 \\ -0.2479 & 0.4545 & 1 \end{bmatrix} \tag{33}$$

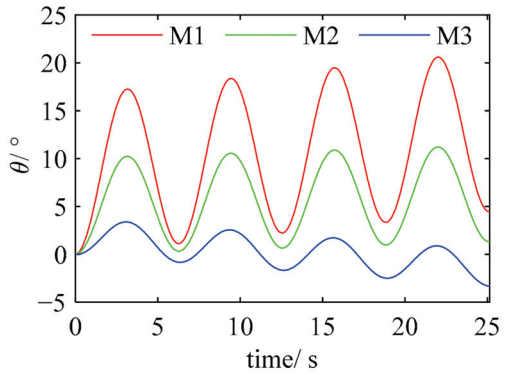


Figure 9. The angular position of motors.

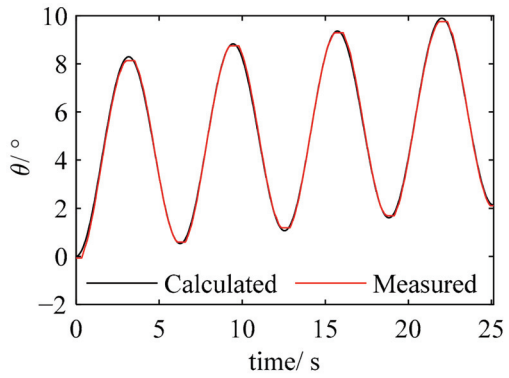


Figure 10. The angular position of joint 1.

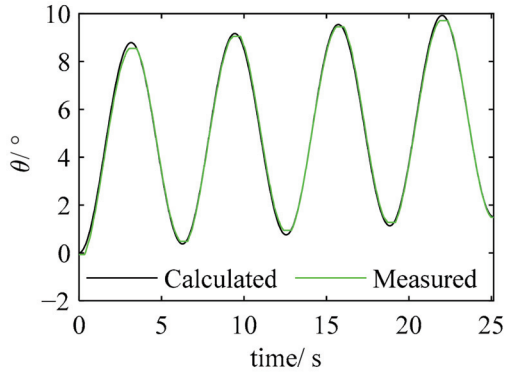


Figure 11. The angular position of joint 2.

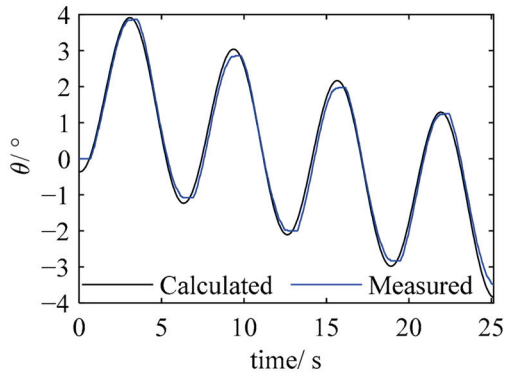


Figure 12. The angular position of joint 3.

As shown in Figures 10–12, the trajectories calculated from the kinematic model and those obtained from the encoders are essentially identical. This indicates that the theoretical model and experimental measurements have high consistency. The difference in the trajectory of joint 1 is tiny because joint 1 is totally decided by its driving motor and free of the influence of other joints, so joint 1 can achieve precise and independent control. The trajectory of joint 2 is decided by its driving motor and the rotation angle of joint 1, which introduces a small coupling effect. However, due to the effective compensation and coordination that occurred between the joints, the trajectory of joint 2 is still small. The trajectory of joint 3 is determined by its driving motor, as well as by joints 1 and 2. Due to assembly clearance and elasticity, joint 3, as the distal end, is greatly influenced. Consequently, the trajectory error of joint 3 is slightly larger than that of the other joints. However, the calculated trajectory and the measured one have a similar trend in growth, which shows the correctness of the theory. Overall, the experimental results support the correctness of the kinematic model presented in this paper.

To further verify the correctness of the proposed kinematic model, additional analyses were conducted based on the data from the comparative controller experiments. The joint rotation angle was obtained from the encoder. Then, substituting the angle joint angles into $\mathbf{Q} = \mathbf{S}^{-1}\boldsymbol{\Theta}$, the experimental motor angles were derived. Using the same method, the theoretical motor rotation angles, which are associated with the expected joint angles, were obtained. As indicated in Figure 13, when joint 2 and joint 3 have the same expected angle, the angles of motor 2 and motor 3 differ significantly. The reason for it is that the coupling angle of joint 3 is almost equal to the expected angle, so motor 3 only needs to rotate a small angle when the joint rotates to the expected angle. By further analysis of Figure 13, the theoretical and experimental results of the motor angle are essentially identical. Due to external disturbances and joint reversals, the experimental joint angles exhibit some changes, which lead to changes in the experimental motor angles. Nevertheless, the

https://doi.org/10.3390/act15050271

theoretical and experimental results of motor angles have similar change trends, thereby verifying the correctness of the proposed kinematic model.

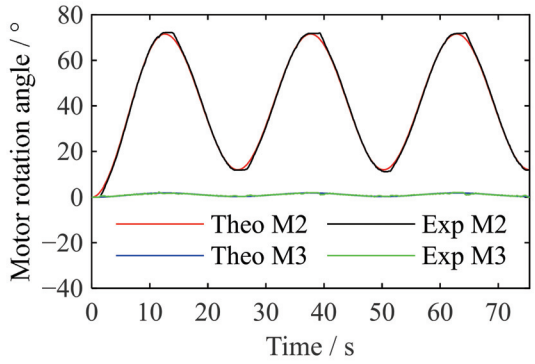


Figure 13. Motor rotation angle.

5.3. Comparative Experiment Studies

Controller 1 (proposed)**.** *A nonsingular terminal sliding mode control scheme based on α-TDE.*

Controller 2. *A nonsingular terminal sliding mode control scheme based on TDE. The sliding mode surface adopted in Controller 2 is identical to that of the proposed controller 1.*

$$\begin{aligned} \boldsymbol{\tau}_m = \quad & \overline{\mathbf{M}}(\ddot{\mathbf{q}}_d + \dot{\varphi}(\mathbf{e}) + \lambda \mathbf{s}) \\ & + \boldsymbol{\tau}_m(t-L) - \overline{\mathbf{M}}\ddot{\mathbf{q}}(t-L) \end{aligned} \tag{34}$$

Controller 3. *A terminal sliding mode control scheme based on TDE.*

$$\mathbf{s} = \dot{\mathbf{e}} + \beta \mathbf{e} \tag{35}$$

$$\begin{aligned} \boldsymbol{\tau}_m = \quad & \overline{\mathbf{M}}(\ddot{\mathbf{q}}_d + \beta \dot{\mathbf{e}} + \lambda \mathbf{s}) \\ & + \boldsymbol{\tau}_m(t-L) - \overline{\mathbf{M}}\ddot{\mathbf{q}}(t-L) \end{aligned} \tag{36}$$

For our proposed control scheme based on α-TDE, the parameters are set as $b = diag(0.9, 0.9)$, $\eta = diag(1.5, 1.5)$, $\lambda = diag(8, 8)$, $\overline{\mathbf{M}} = diag(0.01, 0.01)$, $\Delta = diag(0.2, 0.2)$, and $\alpha = diag(0.1, 0.2)$. The control parameters for Controller 2 are the same as ours. The control parameters for Controller 3 are set as $\beta = diag(2, 2)$, and the others are chosen to be the same as ours.

In the comparative experiments, the tendon-driven manipulator moves to track a trajectory, which is shown in Figure 14a,b. The tracking errors of the two joints are shown in Figure 14c,d. As indicated in Figure 14c,d, it can be observed that the control performances of the three controllers achieve good results. However, the proposed Controller 1 achieves the smallest tracking errors of the three controllers. Especially during the rapid acceleration and deceleration phases, the superiority of Controller 1 is most prominent. Then, the mean absolute error (MAE) and root mean square error (RMSE) were calculated based on the tracking error data, and the calculated results are shown in Table 2. In order to make the comparisons quantitative, we further compared the data from Table 2. For MAE, the values of the proposed Controller 1 can achieve 0.261° and 0.219° for joint 2 and joint 3, corresponding to 90.3% and 91.2% of those achieved by Controller 2, whereas those of Controller 3 are 85.5% and 82.3%. For RMSE, the values of the proposed Controller 1 can achieve 0.311° and 0.261° for joint 2 and joint 3, corresponding to 82.9% and 89.3% of those achieved by Controller 2, whereas those of Controller 3 are 71.3% and 74.7%. Overall, the comparison of MAE and RMSE indicates that the proposed Controller 1 is superior to the other controllers in terms of tracking accuracy.

https://doi.org/10.3390/act15050271

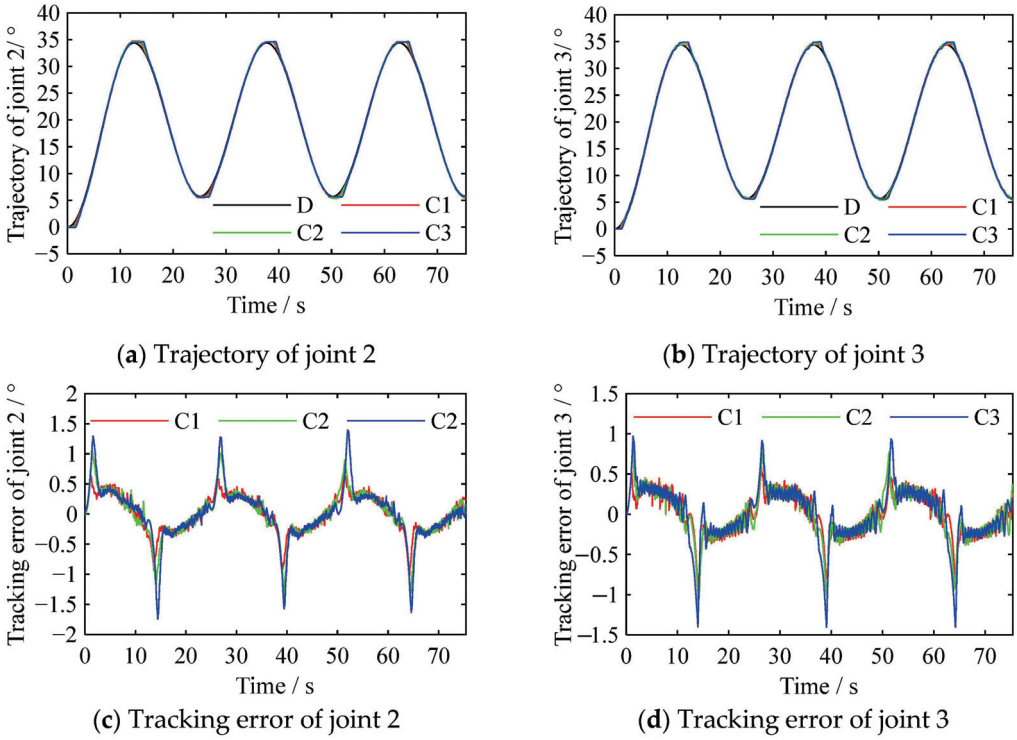


Figure 14. Trajectory tracking performance and tracking error.

Table 2. MAE and RMSE of trajectory tracking performance.

(Degree)	MAE (Joint 2/3)	RMSE (Joint 2/3)
Proposed	0.261/0.219	0.311/0.261
Controller 2	0.289/0.240	0.375/0.292
Controller 3	0.305/0.266	0.436/0.349

The results of the experiments are shown in Figure 15. As indicated in Figure 15a,b, the sliding mode surfaces of the three controllers are similar, indicating that all controllers maintain a stable sliding regime. Taking further analysis of Figure 15c,d, when the tendon-driven manipulator moves upward, the $\hat{\mathbf{H}}(t)$ value of the proposed controller is smaller than that of the other two controllers. Owing to the effect of gravity, the pulling force of the link increases, leading to greater nonlinear disturbances. The proposed Controller 1 uses the α-TDE technique, which utilizes a weighted average of two moments ($t - L$, $t - 2L$) along with an adjusting parameter α, which can improve the accuracy of the time-delay estimation. This achieves better tracking performance of the single-tendon-driven manipulator. When the tendon-driven manipulator moves downward, gravity can assist the manipulator to move downward, and in this case, the $\hat{\mathbf{H}}(t)$ value of the proposed Controller 1 is similar to the other two controllers.

As shown in Figure 15e,f, the control torque of the proposed Controller 1 is significantly smaller than that of the other two controllers during the upward phase. This reduction is attributed to the influence of the α-TDE; this explains that the α-TDE-NTSM controller can effectively solve the nonlinear dynamics of the tendon-driven manipulator. Furthermore, the smoother torque of the proposed Controller 1 reduces mechanical vibration, which is beneficial for trajectory tracking. Overall, the proposed Controller 1 achieves superior tracking performance, which uses lower control torque. This further indicates the effectiveness and practicality of our proposed Controller 1 in practical applications.

https://doi.org/10.3390/act15050271

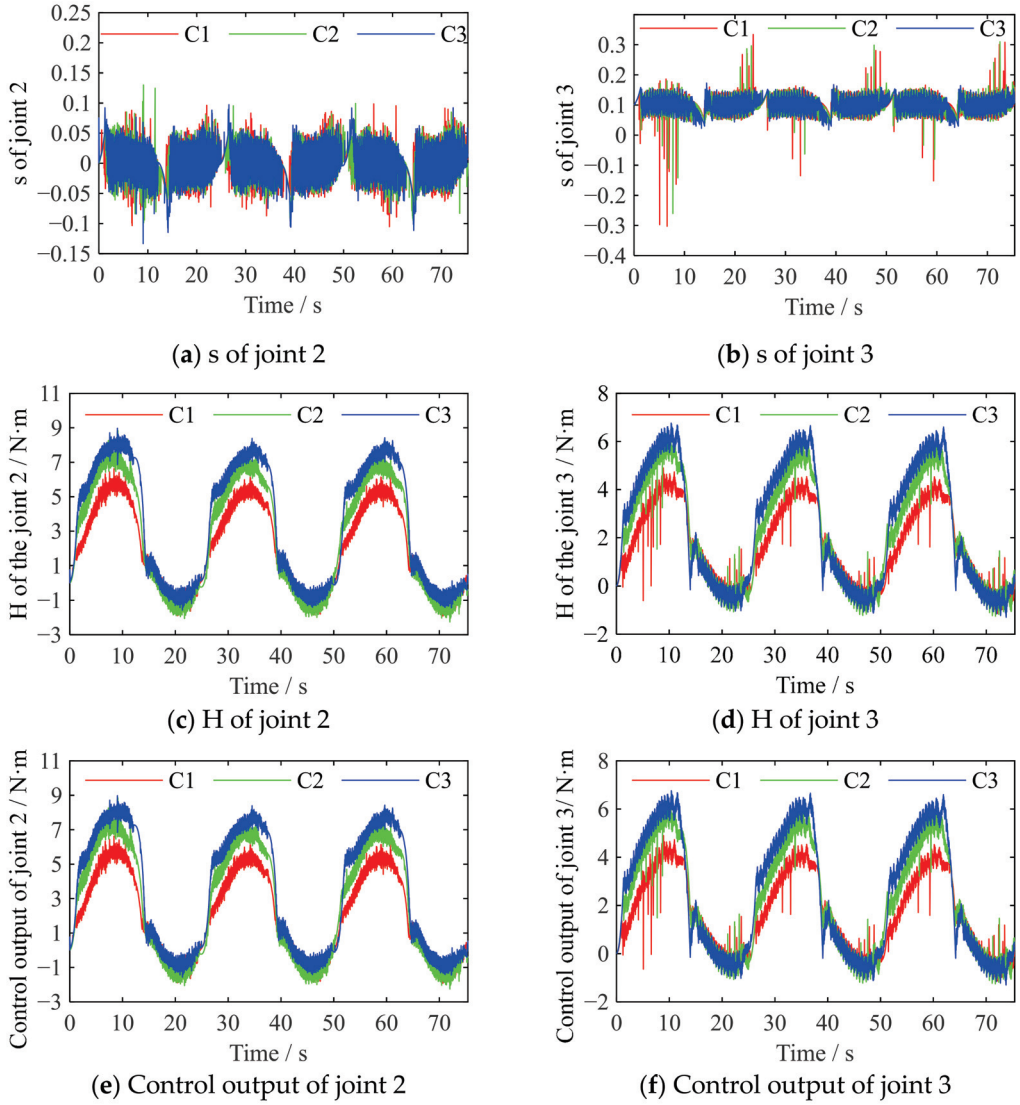


Figure 15. Experiment results.

Overall, all control schemes employed in the experiments provide good tracking control for the single-tendon-driven manipulator. However, experimental results indicate that our proposed control scheme exhibits superior comprehensive control performance.

5.4. Control Experiments with and Without Load

To demonstrate the robustness of our proposed control scheme against unknown uncertainties, an additional 0.2 kg payload was added to the end-effector of the robotic manipulator. The experiment used the same desired trajectory and period as the previous comparative experiments. The results of the experiment are depicted in Figure 16, and the calculated RMSE and MAE values from the experimental data are summarized in Table 3.

As indicated in Figure 16a,b, our proposed controller is still able to maintain accurate tracking performance under load disturbances. For MAE, the values increase by 1.1% and 3.7% for joints 1 and 2, respectively. For RMSE, the values increase by 4.5% and 1.5% for joints 1 and 2, respectively. Summarizing the above experimental results, it can be seen that the strong robustness of our newly proposed control scheme is demonstrated.

https://doi.org/10.3390/act15050271

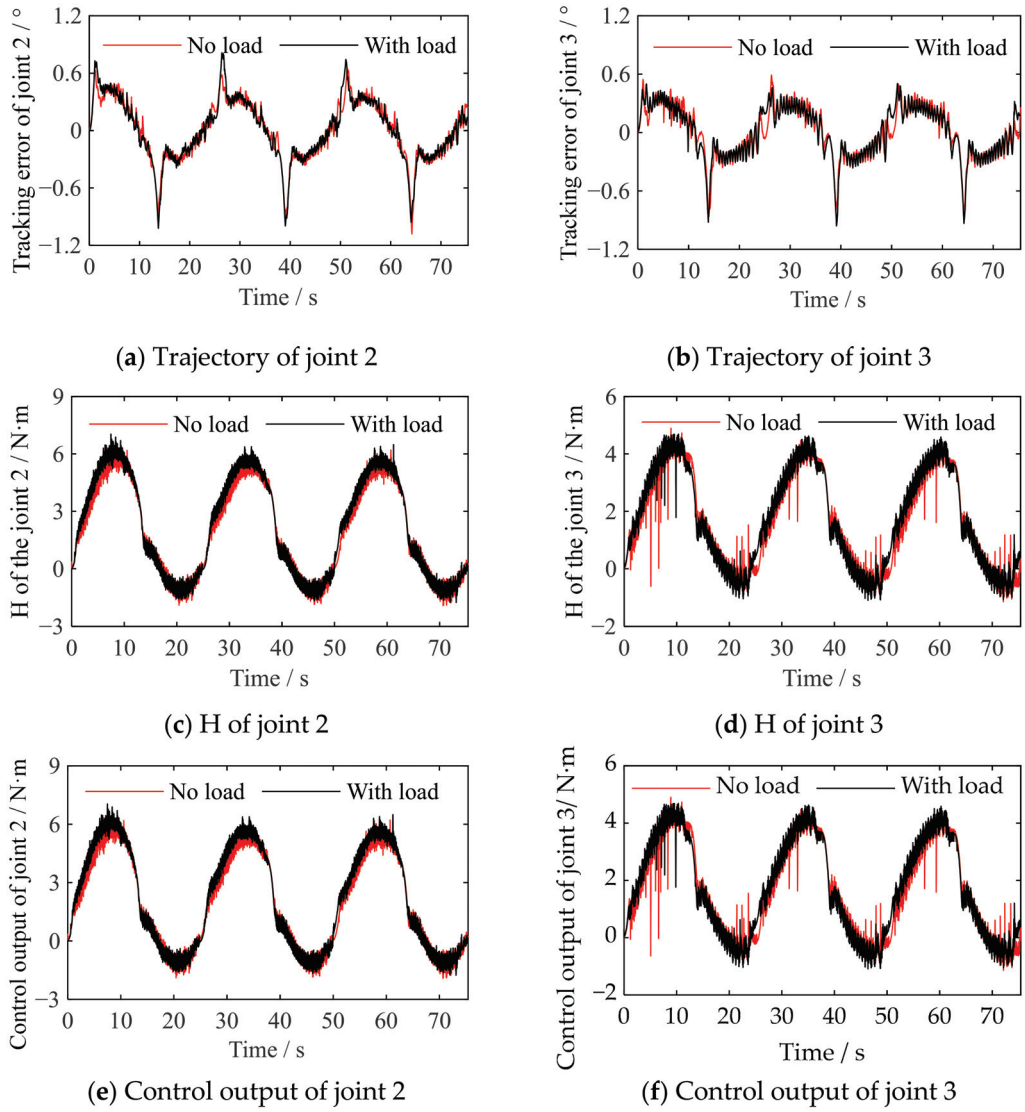


Figure 16. Experimental results with a 0.2 kg payload vs. no payload.

Table 3. Control performance with a 0.2 kg payload vs. no payload.

(Degree)	MAE (Joint 2/3)	RMSE (Joint 2/3)
No load	0.261/0.219	0.311/0.261
With load	0.264/0.227	0.325/0.265

6. Conclusions

This paper presents the design, modeling, and experimental validation of a single-tendon-driven manipulator. The proposed design utilizes a single-tendon actuation method for each joint to reduce backlash and improve motion precision, while an additional tendon and motor serve as a tensioner to maintain positive tendon tension throughout the manipulator's workspace. Then, we designed an NTSM control scheme based on α-TDE for the trajectory tracking control for a single-tendon-driven manipulator, and conducted a comparative experiment to verify the superiority of the proposed controller.

1. The development of a tendon routing strategy that ensures all tendons remain under positive tension. This is achieved by leveraging opposing torques and an active tensioning mechanism, where a motor-driven tendon replaces traditional springs to allow consistent preload adjustment.

https://doi.org/10.3390/act15050271

2. The derivation of a kinematic model that explicitly accounts for motion coupling between joints—a distinctive characteristic of tendon-driven systems. The coupling matrix was formulated to relate motor inputs to joint angles, enabling accurate forward kinematics.
3. The NTSM control design enables the single-tendon-driven manipulator to achieve better trajectory tracking. This is accomplished by employing the α-TDE technique, which estimates external disturbances, and by extending the sampling data period to mitigate the overshoot problems caused by TDE. The stability of the control system is analyzed using the Lyapunov method.
4. The fabrication of a prototype and experimental validation of the kinematic model. Results show close agreement between the predicted and measured joint trajectories, with minor deviations attributed to assembly tolerances and structural elasticity, particularly at the distal end.
5. Compared with the two TDE control schemes, the α-TDE scheme achieves a smaller tracking error and a smoother curve. Furthermore, comparative experiments with and without load further validate the proposed controller's reliability under strong robust performance requirements.

The proposed design and modeling approach demonstrates feasibility for applications, offering advantages in weight reduction, drag minimization, and decoupled control. And our proposed control scheme achieves great performance on trajectory tracking for a single-tendon-driven manipulator. Future work may focus on dynamic modeling, closed-loop control, and extending the design to manipulators with more degrees of freedom.

Author Contributions: Conceptualization, F.Y.; Writing, F.Y.; Data curation, J.L.; Software, J.L.; Formal analysis, H.H.; Funding acquisition, F.Y., H.H.; Validation, Q.X.; Methodology, L.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 52305136, Yantai Science and Technology Innovation Development Plan, grant number 2024JCYJ093 and Major scientific and technological innovation projects in Shandong Province, grant number 2023CXGC010407.

Data Availability Statement: The original data presented in the study are openly available in Zenodo repository at https://zenodo.org/records/19467931 (accessed on 5 April 2026).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

NTSM	Nonsingular terminal sliding mode
TDE	Time-delay estimation

References

1. Bales, I.; Zhang, H. A six degrees-of-freedom cable-driven robotic platform for head–neck movement. *Sci. Rep.* **2024**, *14*, 8750. [CrossRef]
2. Khomami, A.M.; Najafi, F. A survey on soft lower limb cable-driven wearable robots without rigid links and joints. *Rob. Auton. Syst.* **2021**, *144*, 103846. [CrossRef]
3. Doss, A.S.A.; Lingampally, P.K.; Nguyen, G.M.T.; Schilberg, D. A comprehensive review of wearable assistive robotic devices used for head and neck rehabilitation. *Results Eng.* **2023**, *19*, 101306. [CrossRef]
4. Peñaloza, E.A.G.; Caballero, F.D.B.; Leston, L.A.; de Lima, S. Development of a four-cable-driven parallel robot for execution of complex tasks in closed environments. *Int. J. Dyn. Control* **2025**, *13*, 83. [CrossRef]
5. Ha, S.-H.; Woo, S.-W.; Kim, M.-C.; Piao, J.; Kim, G.-H.; Sarker, M.S.; Park, B.-J.; Kim, S.; Jung, M.; Hong, K.-S. A Cable-driven Parallel Robot for High-rack Logistics Automation. *Int. J. Control Autom. Syst.* **2024**, *22*, 3329–3340. [CrossRef]

6. Pannequin, R.; Jouaiti, M.; Boutayeb, M.; Lucas, P.; Martinez, D. Automatic tracking of free-flying insects using a cable-driven robot. *Sci. Robot.* **2020**, *5*, eabb2890. [CrossRef] [PubMed]
7. Zhu, K.; Lueth, T.C.; Sun, Y. Design of FDM-printable tendon-driven continuum robots using a serial S-shaped backbone structure. *Biomim. Intell. Robot.* **2025**, *5*, 100188. [CrossRef]
8. Shi, K.; Yang, J.; Tong, Y.; Hou, Z.; Yu, H. Design and analysis of a cable-driven gravity compensation mechanism for spatial multi-DoF robotic systems. *Mech. Mach. Theory* **2023**, *190*, 105452. [CrossRef]
9. Pigani, L.; Gallina, P. Cable-direct-driven-robot (CDDR) with a 3-link passive serial support. *Robot. Comput. Integr. Manuf.* **2014**, *30*, 265–276. [CrossRef]
10. Gueners, D.; Chanal, H.; Bouzgarrou, B. Design and implementation of a cable-driven parallel robot for additive manufacturing applications. *Mechatronics* **2022**, *86*, 102874. [CrossRef]
11. Song, D.; Lu, M.; Zhao, L.; Sun, Z.; Wang, H.; Zhang, L. A novel real-time tension distribution method for cable-driven parallel robots. *Robotica* **2024**, *42*, 3692–3708. [CrossRef]
12. Banala, S.K.; Agrawal, S.K.; Scholz, J.P. Active Leg Exoskeleton (ALEX) for Gait Rehabilitation of Motor-Impaired Patients. In *IEEE International Conference on Rehabilitation Robotics*; IEEE: Noordwijk, The Netherlands, 2007. [CrossRef]
13. Brackbill, E.A.; Mao, Y.M.Y.; Agrawal, S.K.; Annapragada, M.; Dubey, V.N. Dynamics and control of a 4-dof wearable cable-driven upper arm exoskeleton. In *IEEE International Conference on Robotics & Automation*; IEEE: Kobe, Japan, 2009. [CrossRef]
14. Pott, A. *Cable-Driven Parallel Robots: Theory and Application*; Springer Tracts in Advanced Robotics; Springer: Berlin/Heidelberg, Germany, 2018. [CrossRef]
15. Troncoso, D.A.; Robles-Linares, J.A.; Russo, M.; Elbanna, M.A.; Wild, S.; Dong, X.; Mohammad, A.; Kell, J.; Nor-ton, A.D.; Axinte, D. A Continuum Robot for Remote Applications: From Industrial to Medical Surgery with Slender Continuum Robots. *IEEE Robot. Autom. Mag.* **2022**, *30*, 94–105. [CrossRef]
16. Varga, M.; Sinčák, P.J.; Merva, T.; Marcinko, P.; Miková, L.; Prada, E.; Schnitzer, M.; Virgala, I. Design and analyses of tendon-driven continuum robots for minimally invasive surgery. *IFAC-PapersOnLine* **2024**, *58*, 25–30. [CrossRef]
17. Li, X.; Wang, Y.; Wu, K.; Xu, Y.; Cao, X.; Feng, T.; Chen, T.; Jin, P.; Wang, Z.; Xue, Y.; et al. Design and modeling of a multi-backbone continuum robot with a large extension ratio. *Mech. Mach. Theory* **2025**, *206*, 105935. [CrossRef]
18. Zhang, T.; Zhang, G.; Zhang, X.; Du, F.; Song, R. Design and Modeling of Multi-vertebrae Continuum Robot for Minimally Invasive Surgery. *Procedia Comput. Sci.* **2023**, *226*, 85–91. [CrossRef]
19. Rus, D.; Tolley, M.T. Design, fabrication and control of soft robots. *Nat. News* **2015**, *521*, 467–475. [CrossRef] [PubMed]
20. Abozaid, Y.A.; Aboelrayat, M.T.; Fahim, I.S.; Radwan, A.G. Soft robotic grippers: A review on technologies, materials, and applications. *Sens. Actuators Phys.* **2024**, *372*, 115380. [CrossRef]
21. Vatanaprasan, P.; Chumnanvej, S.; Suthakorn, J.; Treratanakulchai, S. Design Optimization of a Soft Robotic Manipulator for Intraventricular Hemorrhage Evacuation. In Proceedings of the 16th Hamlyn Symposium on Medical Robotics, London, UK, 25–28 June 2024; pp. 53–54.
22. Seleem, I.A.; Naeem, M.A.; Takemura, H.; Ishii, H. Tendon-driven based extensible soft robot with variable stiffness. *Rob. Auton. Syst.* **2025**, *190*, 104988. [CrossRef]
23. Dreyfus, R.; Boehler, Q.; Lyttle, S.; Gruber, P.; Lussi, J.; Chautems, C.; Gervasoni, S.; Berberat, J.; Seibold, D.; Ochsenbein-Kölble, N. Dexterous helical magnetic robot for improved endovascular access. *Sci. Robot.* **2024**, *9*, eadh0298. [CrossRef]
24. Kim, Y.; Genevriere, E.; Harker, P.; Choe, J.; Balicki, M.; Regenhardt, R.W.; Vranic, J.E.; Dmytriw, A.A.; Patel, A.B.; Zhao, X. Telerobotic neurovascular interventions with magnetic manipulation. *Sci. Robot.* **2025**, *7*, eabg9907. [CrossRef] [PubMed]
25. Zou, B.; Liu, H.; Fen, X.; Gao, Z.; Ge, Z.; Yang, W. Design, optimization, and validation of a magnetic mother-child robot system for targeted drug delivery. *Mater. Des.* **2025**, *253*, 113827. [CrossRef]
26. Liu, T.; Zhang, G.; Du, F.; Zhang, X.; He, C.; Li, Y.; Song, R. Magnetic continuum robot: Design, modeling and verification. *Int. J. Mech. Sci.* **2025**, *301*, 110460. [CrossRef]
27. Wang, C.; Jin, T.; Ren, H.; Zhang, H.; Zhao, J. Modeling and design of pre-set stiffness continuum robot. *Int. J. Mech. Sci.* **2025**, *295*, 110224. [CrossRef]
28. Shao, Z.; Wu, Q.; Chen, B.; Wu, H.; Zhang, Y. Modeling and inverse control of a compliant single-tendon-sheath artificial tendon actuator with bending angle compensation. *Mechatronics* **2019**, *63*, 102262. [CrossRef]
29. Chang, J.-C.; Dai, H.; Wang, X.; Axinte, D.; Dong, X. Development of a continuum robot with inflatable stiffness-adjustable elements for in-situ repair of aeroengines. *Robot. Comput. Integr. Manuf.* **2025**, *95*, 103018. [CrossRef]
30. Wang, P.; Zang, J.; Dong, Z.; Xing, Z.; Zhao, J. Design and statics model of an extensible hybrid-driven continuum robot with variable stiffness. *Sens. Actuators Phys.* **2025**, *384*, 116177. [CrossRef]
31. Sai, X.U.; Honglei, Z. Design of a 6-DoF cable-driven parallel robot experimental platform. *Exp. Technol. Manag.* **2025**, *42*, 188–195.
32. Cheng, J.; Shen, T. Motion decoupling for cable-driven serial robots based on a noncircular pulley. In *2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*; IEEE: New Youk, NY, USA, 2023; pp. 739–745.

33. Xu, F.; Zi, B.; Yu, Z.; Zhao, J.; Ding, H. Design and implementation of a 7-DOF cable-driven serial spray-painting robot with motion-decoupling mechanisms. *Mech. Mach. Theory* **2024**, *192*, 105549. [CrossRef]
34. Wang, F.; Dong, H.; Yan, L.; Xu, W.; Liang, B. Development of a linear decoupling cable-driven manipulator with independent driving joints. *Mechatronics* **2024**, *100*, 103192. [CrossRef]
35. Seo, D.H.; Lee, J.W.; An, H.M.; Lee, S.Y. An Adaptive Sliding Mode Control Using a Novel Adaptive Law Based on Quasi-Convex Functions and Average Sliding Variables for Robot Manipulators. *Electronics* **2024**, *13*, 3940. [CrossRef]
36. Duong, T.T.C.; Nguyen, C.C.; Tran, T.D. Experimental Investigation of Motion Control of a Closed-Kinematic Chain Robot Manipulator Using Synchronization Sliding Mode Method with Time Delay Estimation. *Appl. Sci.* **2025**, *15*, 5206. [CrossRef]
37. Truong, T.N.; Vo, A.T.; Kang, H.J. A Novel Time Delay Nonsingular Fast Terminal Sliding Mode Control for Robot Manipulators with Input Saturation. *Mathematics* **2025**, *13*, 119. [CrossRef]
38. Zhou, Z.; Gao, J.; Zhang, L. A model-free decoupled and robust repetitive controller for trajectory tracking performance of a three-axis parallel mechanism. *J. Mech. Sci. Technol.* **2025**, *39*, 857–875. [CrossRef]
39. Alsaied, A.; Li, K. An Optimized Type 2 Fuzzy Control with Time Delay Estimation for Gait Trajectory Tracking in Lower-Limb Exoskeleton Robot. *Int. J. Fuzzy Syst.* **2025**. [CrossRef]
40. Alqaisi, W.; Soliman, M.; Badawi, A.; El-Bayeh, C. Detection and tracking quadrotor using surf and feedback linearization sliding mode control. *ROBOMECH J.* **2025**, *12*, 36. [CrossRef]
41. Dai, J.; Wan, M.; Qin, X.-B.; Zhang, W.-H. A time-delay-based sliding mode controlling method for reducing contour errors of machine tools. *J. Manuf. Process* **2023**, *106*, 407–421. [CrossRef]
42. Li, X.; Zhang, Z.; An, J.; Zhou, X.; Hu, G.; Zhang, G.; Man, W. Adaptive sliding mode control of modular self-reconfigurable spacecraft with time-delay estimation. *Def. Technol.* **2022**, *18*, 2170–2180. [CrossRef]
43. Glida, H.E.; Chelihi, A.; Abdou, L.; Sentouh, C.; Perozzi, G. Trajectory tracking control of a coaxial rotor drone: Time-delay estimation-based optimal model-free fuzzy logic approach. *ISA Trans.* **2023**, *137*, 236–247. [CrossRef]
44. Boudjedir, C.E. A widely-practical model-free prescribed time control for trajectory tracking of n-DOF robot manipulators. *Eur. J. Control* **2025**, *86*, 101396. [CrossRef]
45. Vo, A.T.; Truong, T.N.; Kang, H.-J.; Nguyen, N.H.A. Prescribed performance model-free sliding mode control using time-delay estimation and adaptive technique applied to industrial robot arms. *Inf. Sci.* **2025**, *702*, 121911. [CrossRef]
46. Choi, J.; Kwon, W.; Lee, Y.S.; Han, S. Adaptive time-delay estimation error compensation for application to robot manipulators. *Control Eng. Pract.* **2024**, *151*, 106029. [CrossRef]
47. Meng, D.; Xu, H.; Xu, H.; Sun, H.; Liang, B. Trajectory tracking control for a cable-driven space manipulator using time-delay estimation and nonsingular terminal sliding mode. *Control Eng. Pract.* **2023**, *139*, 105649. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

actuators

MDPI

Article

Robust Trajectory Tracking Control of an Unmanned Surface Vehicle via a Sliding-Mode Dynamic Neural Network Identifier

Filiberto Muñoz Palacios [1,2], Eduardo S. Espinoza [1,2], Jorge Said Cervantes-Rojas [1,2], Jesus Patricio Ordaz Oliver [3], Octavio Garcia-Salazar [4] and Luis Rodolfo Garcia Carrillo [5,*]

[1] Department of Research and Multidisciplinary Studies, Center for Research and Advances Studies of the National Polytechnic Institute, Mexico City 07360, Mexico; fmunoz@secihti.mx (F.M.P.); eduardo.espinoza@cinvestav.mx (E.S.E.); jcervantes@secihti.mx (J.S.C.-R.)
[2] Researchers for Mexico, Secretariat of Science, Humanities, Technology and Innovation, Mexico City 03940, Mexico
[3] Institute of Basic Sciences and Engineering, Autonomous University of the Hidalgo State, Mineral de la Reforma 42184, Mexico; jesus_ordaz@uaeh.edu.mx
[4] Aerospace Engineering Research and Innovation Center, Faculty of Mechanical and Electrical Engineering, Autonomous University of Nuevo Leon, Apodaca 66616, Mexico; octavio.garciasl@uanl.edu.mx
[5] Klipsch School of Electrical and Computer Engineering, New Mexico State University, Las Cruces, NM 88003, USA
* Correspondence: luisillo@nmsu.edu; Tel.: +1-(575)-646-3173

Abstract

The trajectory tracking problem of underactuated unmanned surface vehicles (USVs) with unknown physical parameters arising from hydrodynamic effects is addressed using a robust control strategy based on a sliding-mode dynamic neural network identifier. To handle the unknown physical parameters, a dynamic neural network identifier with a novel structure is developed, enabling the construction of an equivalent mathematical model of the USV dynamics. To compensate for the underactuated nature of the system, a coordinate transformation is introduced. Using this transformation, together with the proposed identifier, a nonsingular sliding-mode controller is designed. Lyapunov-based analysis establishes finite-time convergence of the neural weight estimation errors to zero and convergence of the identification errors to a bounded neighborhood of zero. Furthermore, once the identification errors enter this bounded region, they asymptotically converge to zero. In addition, the closed-loop stability analysis guarantees finite-time convergence of the tracking errors. The effectiveness of the proposed identifier–controller framework is validated through simulation studies that incorporate explicit actuator saturation constraints and external disturbances to emulate realistic operating conditions. These results demonstrate the practical applicability of the proposed control strategy, as the commanded inputs remain within the physical limits of the propulsion system. Comparative results with a state-of-the-art model-based super-twisting controller show that the proposed approach achieves comparable tracking performance while eliminating the need for prior knowledge of the system's dynamic parameters.

Keywords: dynamic neural network identifier; nonsingular sliding-mode; unmanned surface vehicle; free-parameter controller.

1. Introduction

Autonomous maneuvering of unmanned surface vehicles (USVs) requires control of the heading angle and the horizontal position to carry out accurate trajectory tracking

tasks [1]. This type of vehicle is subject to complex aquatic environments, internal nonlinearities, actuator saturation effects and dead zones, modeling and parametric uncertainties, and external disturbances including wind, waves, and currents, among others [2]. In this context, one of the most challenging issues is determining the so-called hydrodynamic coefficients, which are the physical parameters related to hydrodynamic phenomena. These numerical values characterize the damping and added-mass effects experienced by the USV.

Hydrodynamic coefficients are commonly estimated on the basis of a set of simplifying modeling and physical assumptions. Several procedures and techniques have been developed to deal with this issue, for example, computational fluid dynamics, analytical and semi-empirical methods, or captive and free model testing [3]. However, in most cases, the more accurate and complete the USV model is, the more complicated it is to perform a parametric identification of these coefficients.

The authors in [4] presented the identification of a maneuvering unmanned marine vehicle. Such identification is based on the use of an extended Kalman filter and a support vector machine, avoiding an experimental method. In [5], the authors deal with the identification of the hydrodynamic coefficients of an underwater vehicle. The identification is based on a simplified identification model considering low complexity approximations of the damping effects.

In most cases, it is difficult to realize a parametric identification due to complex mathematical representations of mechanical systems (including coupled, underactuated, or lumped parameter dynamics) or due to a priori nonlinear models not being available. In this case, an approximation of complex and/or unknown nonlinear expressions containing parametric uncertainties can be carried out using nonparametric identification. This scheme constructs an approximation directly from the input–output signals, without presuming any fixed or simple mathematical form.

The study in [6] introduces a novel nonparametric identification strategy of a ship maneuvering motion based on a combination of Gaussian process regression and a genetic algorithm. In the work reported in [7], a fixed-time extended state observer is designed to estimate environmental disturbances and model uncertainties affecting multiple surface vehicles in a distributed cooperative control problem. In [8], a finite-time disturbance observer is designed to approximate a nonlinear expression with uncertain damping and added mass coefficients of a USV model. Other studies have used optimal and intelligent strategies to deal with the identification of unknown USV dynamics. The strategy presented in [9] makes use of an optimal method to approximate the uncertainties and hydrodynamic couplings of the mathematical model of a USV. This strategy combines the particle swarm optimization algorithm with the interior-point algorithm.

The work in [10] proposes an adaptive fuzzy controller to address the path-tracking problem for a USV. The proposed scheme leverages the approximation capabilities of a fuzzy inference system to capture nonlinear dynamics, enabling the identification and compensation of model uncertainties and time-varying disturbances. Similarly, in [11], the authors develop a course-tracking control strategy for USVs that incorporates actuator input quantization. A fuzzy logic system is used to approximate system uncertainties, while closed-loop stability is rigorously ensured through a Lyapunov-based analysis. It is worth noting that an adequate estimation of the hydrodynamic interactions represents a key step in obtaining a faithful mathematical description to predict the USV motion performance as well as to design model-based control strategies to compensate for their influence and achieve robust trajectory tracking.

Another important issue related to the USV's dynamic nature is its underactuated motion behavior. In this case, the lateral motion cannot be directly regulated because the vehicle is only equipped with actuators for surge and yaw movements; the lateral motion

https://doi.org/10.3390/act15050273

control requires a coordinated combination of the yaw and surge motions. This makes designing tracking controllers for underactuated USVs more difficult than for fully actuated USVs. In this respect, a model-based control strategy that has recently proven to be effective in addressing underactuated USV tracking problems is the so-called hand position point (HPP) approach [12–17]. This method makes use of a change of coordinates applied to the underactuated dynamic system to transform it into an actuated one, allowing the design of direct control strategies instead of the use of cascaded-based methods.

Nonlinear and classical controllers are designed based on such a simplified mathematical structure. In the study reported in [15], the authors focus on the design of a trajectory tracking and path following controller using the HPP strategy for marine vehicles. In [18], a time-varying version of the HPP is introduced for underactuated underwater vehicles that demonstrate its effectiveness in solving trajectory tracking problems. Similarly, in [16,17], sliding-mode controllers were implemented on HPP-based models to address trajectory tracking problems in USVs.

To address the approximation problem associated with parameter and modeling uncertainties, current studies employ dynamic neural network (DNN) architectures with higher structural complexity, for instance, the Hopfield-type state-space neural networks [19,20]. The continuous time evolution of this class of DNNs has proven to be effective in identifying unknown dynamics and uncertainties online in a broad variety of complex nonlinear systems [21–26]. Such DNNs have two key features: their tunable parameters (the synaptic weights) represent linear elements in the neural network structure, and their training is carried out using an online updating procedure that solves a set of differential equations.

In these studies, the identification stage is characterized by an asymptotic behavior of the identification error to an ultimately bounded region. Recent works based on such an identification scheme have succeeded in guaranteeing convergence rate conditions for the identification problem. In the work studied in [27], an exponential function-based learning law is designed for an identifier with the structure of the DNN described above. The DNN weights are updated by two adaptive algorithms that feature a specified exponential rate of convergence. The identification error is guaranteed to converge to an invariant set defined by a minimal ellipsoid. In addition, a finite-time convergence of the tracking error can be reached. The work in [28] proposes an adaptive sliding-mode controller with a DNN to identify and compensate for unknown hydrodynamic effects for an underwater vehicle. The DNN compensates for the unknown dynamics, and the sliding-mode control forces a finite-time convergence of the tracking error to zero. In this context, the identification error is only guaranteed to converge asymptotically to a bounded region, highlighting a key limitation in existing approaches. Addressing this issue remains an open problem and serves as a primary motivation for the developments presented in this work.

The main contributions of this manuscript are summarized as follows:

- A mathematical representation of the dynamics of the USV is developed using a novel nonparametric structure termed the sliding-mode-based dynamic neural network identifier (SM-DNNI). The proposed identifier structure is based on the DNN with a Hopfield-type state-space architecture reported in [21–28] with enhanced convergence properties provided by nonsingular terminal sliding-mode terms, and is driven by a sliding surface constructed from the identification errors between the measured USV states and the SM-DNNI outputs. This structure enables robust and adaptive online identification of the system dynamics without requiring prior knowledge of physical parameters. This identifier architecture, which integrates robustness and adaptation features, has not been previously reported in the literature.

https://doi.org/10.3390/act15050273

- For the proposed SM-DNNI, a rigorous Lyapunov-based stability analysis is carried out for the identification error dynamics. This analysis establishes finite-time convergence of both the sliding surface and the weight estimation errors of the neural network to zero. Moreover, the identification errors between the SM-DNNI and the actual USV dynamics are shown to vanish in finite time. In contrast to the existing dynamic neural network identifiers reported in [21–23,28], which guarantee only asymptotic convergence to a bounded neighborhood of the origin, and the approach in [27], which ensures exponential convergence to a similar bounded region, the proposed method achieves finite-time convergence to zero.
- Building upon the proposed SM-DNNI framework, a nonsingular terminal sliding-mode controller is developed to address the trajectory tracking problem of USVs. A rigorous stability analysis demonstrates that the tracking errors between the SM-DNNI states and the reference trajectory converge to zero in finite time, while the tracking errors of the actual USV asymptotically approach zero. In particular, the proposed control strategy does not require prior knowledge of the physical parameters of the USV. Although a related adaptive closed-loop control architecture is presented in [28], where finite-time convergence of tracking error is achieved via sliding-mode control, the associated identification errors are only guaranteed to converge asymptotically to a bounded set. In contrast, the proposed approach guarantees finite-time convergence to zero for both the identification and tracking errors.

The remainder of the paper is organized as follows. Section 2 introduces the mathematical model of the USV and presents the preliminaries required for the stability analysis of the proposed methods. Section 3 describes the development of the robust nonparametric dynamic neural network identifier. Section 4 details the design of the control strategy. Simulation results under various scenarios, including actuator saturation and external disturbances, are presented in Section 5. Finally, Section 6 concludes the article and outlines directions for future research.

2. Preliminaries and Dynamic Model

This section introduces the mathematical concepts used in the stability analysis, as well as those that support the development of the nonparametric robust dynamic neural network identifier and the derivation of the proposed control strategy. The dynamic model of the underactuated surface vehicle is also described. Furthermore, based on the hand position point technique, introduced in [15], a mathematical model that represents the vehicle dynamics in the inertial frame for the coordinates x and y is derived.

2.1. Preliminaries

In the following, useful lemmas employed in the stability analysis for the identification and control strategies proposed in this manuscript are presented.

Lemma 1 ([29]). *Let* $\xi_1, \xi_2, \ldots, \xi_n \geq 0$ *and* $0 < \epsilon \leq 1$. *Then*

$$\sum_{i=1}^{n} {\xi_i}^{\epsilon} \geq \left(\sum_{i=1}^{n} \xi_i \right)^{\epsilon} \tag{1}$$

Lemma 2 ([29]). *Let* $\xi_1, \xi_2, \ldots, \xi_n \geq 0$ *and* $\epsilon > 1$. *Then*

$$\sum_{i=1}^{n} {\xi_i}^{\epsilon} \geq n^{1-\epsilon} \left(\sum_{i=1}^{n} \xi_i \right)^{\epsilon} \tag{2}$$

https://doi.org/10.3390/act15050273

Lemma 3 ([30,31]). *Assume that there are positive constants $c > 0$, $k > 0$ and $0 < \alpha < 1$. Consider that the Lyapunov function $V(\chi)$ satisfies*

$$\dot{V}(\chi) \leq -kV(\chi) - cV^{\alpha}(\chi), \quad \forall \chi \in \mathcal{U} \backslash \{0\} \tag{3}$$

Then, $V(\chi)$ converges to zero in finite time, and the convergence time is given by

$$\mathcal{T} \leq \frac{\ln\left(1 + \frac{k}{c} V_0^{1-\alpha}\right)}{k(1-\alpha)} \tag{4}$$

where V_0 is the initial condition of $V(\chi)$. Moreover, $V(\chi)$ converges globally to zero in finite time if and only if $\mathcal{U} = \mathbb{R}^n$ and $V(\chi)$ is radially unbounded.

2.2. Dynamic Model for the Surface Vehicle

The following set of equations expresses a simplified mathematical model of the USV with three degrees of freedom (see Figure 1a): surge, sway, and yaw rate, introduced in [32–34]:

$$\begin{aligned}
\dot{x} &= u\cos(\psi) - v\sin(\psi) \\
\dot{y} &= u\sin(\psi) + v\cos(\psi) \\
\dot{\psi} &= r \\
\dot{u} &= f_u(u) + g_u(u)\tau_u + \Delta_u \\
\dot{v} &= f_v(v) + \Delta_v \\
\dot{r} &= f_r(r) + g_r(r)\tau_r + \Delta_r
\end{aligned} \tag{5}$$

where x and y denote the position of the center of gravity (CoG) of the vehicle with respect to the inertial frame, while the yaw angle is denoted by ψ. The variables u and v represent the *surge* and *sway* translational velocities in the body frame, respectively. Additionally, r denotes the angular yaw velocity (yaw rate). The external disturbances, primarily generated by the waves, are denoted by Δ_u, Δ_v, and Δ_r. The control signals for the surge and yaw velocities are given by τ_u and τ_r, respectively. The functions $f_u(u)$, $f_v(v)$, $f_r(r)$ as well as $g_u(u)$ and $g_r(r)$, are defined as

$$\begin{aligned}
f_u(u) &= \frac{1}{m - X_{\dot{u}}}[(m - Y_{\dot{v}})vr + \bar{X}_u u|u| + X_u u] \\
f_v(v) &= \frac{1}{m - Y_{\dot{v}}}[(-m + X_{\dot{u}})ur + \bar{Y}_v v|v| + Y_v v] \\
f_r(r) &= \frac{1}{I_z - N_{\dot{r}}}[(-X_{\dot{u}} + Y_{\dot{v}})vu + \bar{N}_r r|r| + N_r r] \\
g_u(u) &= \frac{1}{m - X_{\dot{u}}}, \quad g_r(r) = \frac{1}{I_z - N_{\dot{r}}}.
\end{aligned} \tag{6}$$

where m is the mass of the vehicle and I_z denotes the moment of inertia about the Z-axis. The hydrodynamic parameters associated with the linear damping are denoted by X_u, Y_v, N_r, while the nonlinear (quadratic) damping coefficients are given by $\bar{X}_u$, $\bar{Y}_v$, $\bar{N}_r$. Finally, the added mass coefficients are defined as $X_{\dot{u}}$, $Y_{\dot{v}}$ and $N_{\dot{r}}$. Given the slow dynamics of the proposed USV, characterized by velocities on the order of 1 m/s, and the considered sampling interval of 0.001 s, the system exhibits a clear separation of time scales; therefore, a continuous-time model was adopted for the design of the control strategy.

https://doi.org/10.3390/act15050273

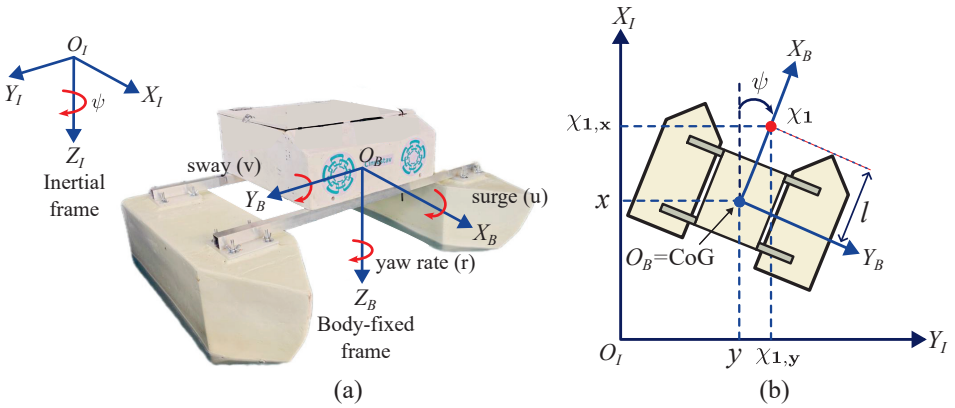


Figure 1. USV diagrams. (**a**) Reference frames, (**b**) hand position point χ_1 in the horizontal plane.

To address the issue of underactuation evidenced by the dynamic model presented in (5), the HPP method used in [15–17] is applied. This method consists of defining the location of the HPP in the vehicle's body (see Figure 1b), which corresponds to the transformation vector $\boldsymbol{\chi_1}$ given by

$$\boldsymbol{\chi_1} = \begin{bmatrix} \chi_{1_x} \\ \chi_{1_y} \end{bmatrix} = \begin{bmatrix} x + l\cos\psi \\ y + l\sin\psi \end{bmatrix} \tag{7}$$

where $l > 0$ is a distance that enables us to represent a pivot point along the longitudinal axis. With this, the new variables to be controlled are χ_{1_x} and χ_{1_y}. The time derivative of $\boldsymbol{\chi_1}$ is given as

$$\dot{\boldsymbol{\chi}}_1 = \begin{bmatrix} u\cos\psi - v\sin\psi - lr\sin\psi \\ u\sin\psi + v\cos\psi + lr\cos\psi \end{bmatrix} \tag{8}$$

Assuming the change of variables, let $\boldsymbol{\chi_2} = \dot{\boldsymbol{\chi}}_1$. Then, $\dot{\boldsymbol{\chi}}_2$ is given by

$$\dot{\boldsymbol{\chi}}_2 = \boldsymbol{R}(\boldsymbol{\psi}) \begin{bmatrix} \dot{u} - v\dot{\psi} - lr\dot{\psi} \\ l\dot{r} + u\dot{\psi} + \dot{v} \end{bmatrix} \tag{9}$$

where the rotation matrix $\boldsymbol{R}(\boldsymbol{\psi})$ is defined as

$$\boldsymbol{R}(\boldsymbol{\psi}) = \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix}. \tag{10}$$

By substituting the dynamic model presented in (5), one obtains the following result

$$\dot{\boldsymbol{\chi}}_2 = \boldsymbol{R}(\boldsymbol{\psi}) \begin{bmatrix} f_u(u) - vr - lr^2 \\ lf_r(r) + f_v(v) + ur \end{bmatrix} + \boldsymbol{R}(\boldsymbol{\psi}) \begin{bmatrix} g_u(u) & 0 \\ 0 & lg_r(r) \end{bmatrix} \begin{bmatrix} \tau_u \\ \tau_r \end{bmatrix} + \boldsymbol{R}(\boldsymbol{\psi}) \begin{bmatrix} \Delta_u \\ l\Delta_r + \Delta_v \end{bmatrix}. \tag{11}$$

Finally, for the surface vehicle, the state space dynamics are expressed in terms of $\boldsymbol{\chi_1}$ and $\boldsymbol{\chi_2}$ as

$$\begin{aligned} \dot{\boldsymbol{\chi}}_1 &= \boldsymbol{\chi_2} \\ \dot{\boldsymbol{\chi}}_2 &= \boldsymbol{R}(\psi) f(\boldsymbol{\chi_1}, \boldsymbol{\chi_2}) + \boldsymbol{R}(\psi) g(\boldsymbol{\chi_1}, \boldsymbol{\chi_2}) \boldsymbol{u} + \boldsymbol{\Delta}_\chi \end{aligned} \tag{12}$$

where

https://doi.org/10.3390/act15050273

$$\begin{aligned} f(\chi_1,\chi_2) &= \begin{bmatrix} f_u(u) - vr - lr^2 \\ lf_r(r) + f_v(v) + ur \end{bmatrix} \\ g(\chi_1,\chi_2) &= \begin{bmatrix} g_u(u) & 0 \\ 0 & lg_r(r) \end{bmatrix} \\ \boldsymbol{u} &= \begin{bmatrix} \tau_u \\ \tau_r \end{bmatrix} \\ \boldsymbol{\Delta}_\chi &= \boldsymbol{R}(\boldsymbol{\psi}) \begin{bmatrix} \Delta_u \\ l\Delta_r + \Delta_v \end{bmatrix}. \end{aligned} \tag{13}$$

The dynamic model can be rewritten in modeling-error form as

$$\begin{aligned} \dot{\boldsymbol{\chi}}_1 &= \boldsymbol{\chi}_2 \\ \dot{\boldsymbol{\chi}}_2 &= \boldsymbol{R}(\psi)\boldsymbol{W_{01}}^\top \boldsymbol{\Phi}(\boldsymbol{\chi_1},\boldsymbol{\chi_2}) + \boldsymbol{R}(\psi)\boldsymbol{W_{02}}^\top \boldsymbol{\Theta}(\boldsymbol{\chi_1},\boldsymbol{\chi_2})\boldsymbol{u} + \tilde{\boldsymbol{f}}_\chi \end{aligned} \tag{14}$$

where $\boldsymbol{W_{01}} \in \mathbb{R}^{4\times 2}$ and $\boldsymbol{W_{02}} \in \mathbb{R}^{8\times 2}$ represent the initial weights, and have the structure

$$\begin{aligned} \boldsymbol{W_{01}} &= [\boldsymbol{W_{01_1}}\ \boldsymbol{W_{01_2}}],\ \boldsymbol{W_{01_1}}, \boldsymbol{W_{01_2}} \in \mathbb{R}^{4\times 1} \\ \boldsymbol{W_{02}} &= [\boldsymbol{W_{02_1}}\ \boldsymbol{0_{4\times 1}};\ \boldsymbol{0_{4\times 1}}\ \boldsymbol{W_{02_2}}],\ \boldsymbol{W_{02_1}}, \boldsymbol{W_{02_2}} \in \mathbb{R}^{4\times 1} \end{aligned} \tag{15}$$

The vector activation functions are composed of sigmoidal functions and have the following structure: $\boldsymbol{\Phi}(\boldsymbol{\chi_1},\boldsymbol{\chi_2}) = [\phi_i] \in \mathbb{R}^{4\times 1}$ and $\boldsymbol{\Theta}(\boldsymbol{\chi_1},\boldsymbol{\chi_2}) = \left[\bar{\boldsymbol{\Theta}}(\boldsymbol{\chi_1},\boldsymbol{\chi_2})\ \boldsymbol{0_{4\times 1}};\ \boldsymbol{0_{4\times 1}}\ \bar{\boldsymbol{\Theta}}(\boldsymbol{\chi_1},\boldsymbol{\chi_2})\right] \in \mathbb{R}^{8\times 2}$, where $\bar{\boldsymbol{\Theta}}(\boldsymbol{\chi_1},\boldsymbol{\chi_2}) = [\theta_i] \in \mathbb{R}^{4\times 1}$. Let us define $\boldsymbol{\iota} = \left[\boldsymbol{\chi_1}^\top\ \boldsymbol{\chi_2}^\top\right]^\top = [\iota_i] \in \mathbb{R}^{4\times 1}$. The elements of the vector of activation functions are defined as

$$\begin{aligned} \phi_i(\iota_i) &= \frac{b_{\varphi_i}}{1+\exp^{-d_{\varphi_i}\iota_i}} - c_{\varphi_i}, \quad i = 1,\ldots,4 \\ \theta_i(\iota_i) &= \frac{b_{\phi_i}}{1+\exp^{-d_{\phi_i}\iota_i}} - c_{\phi_i}, \quad i = 1,\ldots,4 \end{aligned} \tag{16}$$

where $b_{\varphi_i}, b_{\phi_i}, d_{\varphi_i}, d_{\phi_i}, c_{\varphi_i}, c_{\phi_i}, i = 1,\ldots,4$ are parameters of the activation functions. Finally, the modeling error term, denoted by $\tilde{\boldsymbol{f}}_\chi$, is given as

$$\tilde{\boldsymbol{f}}_\chi = \boldsymbol{R}(\psi)f(\boldsymbol{\chi_1},\boldsymbol{\chi_2}) + \boldsymbol{R}(\psi)g(\boldsymbol{\chi_1},\boldsymbol{\chi_2})\boldsymbol{u} + \boldsymbol{\Delta}_\chi - \boldsymbol{R}(\psi)\boldsymbol{W_{01}}^\top \boldsymbol{\Phi}(\boldsymbol{\chi_1},\boldsymbol{\chi_2}) - \boldsymbol{R}(\psi)\boldsymbol{W_{02}}^\top \boldsymbol{\Theta}(\boldsymbol{\chi_1},\boldsymbol{\chi_2})\boldsymbol{u}$$

Assumption 1. *The elements of the modeling error vector* $\tilde{\boldsymbol{f}}_\chi = \left[f_{\chi_x}\ f_{\chi_y}\right]^\top$ *are bounded as* $|f_{\chi_x}| \le {f_{\chi_x}}^+$ *and* $|f_{\chi_y}| \le {f_{\chi_y}}^+$, *where* ${f_{\chi_x}}^+$ *and* ${f_{\chi_y}}^+$ *are positive constants.*

2.3. Identification Sliding Surface

Let us define the identification errors as $\tilde{\boldsymbol{\chi}}_1 = \boldsymbol{\chi}_1 - \hat{\boldsymbol{\chi}}_1$, $\tilde{\boldsymbol{\chi}}_2 = \boldsymbol{\chi}_2 - \hat{\boldsymbol{\chi}}_2$, where $\hat{\boldsymbol{\chi}}_1$ and $\hat{\boldsymbol{\chi}}_2$ represent the identification of $\boldsymbol{\chi}_1$ and $\boldsymbol{\chi}_2$, respectively. Moreover, as outlined in [35], the following sliding surface for the identification errors is defined

$$\boldsymbol{s}_\chi = \tilde{\boldsymbol{\chi}}_2 + \boldsymbol{\kappa}_{\chi_1}\tilde{\boldsymbol{\chi}}_1 + \boldsymbol{\kappa}_{\chi_2}\boldsymbol{\mu}(\tilde{\boldsymbol{\chi}}_1) \tag{17}$$

where $\boldsymbol{s}_\chi = \left[s_{\chi_x}\, s_{\chi_y}\right]^\top$, $\boldsymbol{\kappa}_{\chi_1} = \mathrm{diag}\left(\kappa_{\chi_{1x}}, \kappa_{\chi_{1y}}\right)$, and $\boldsymbol{\kappa}_{\chi_2} = \mathrm{diag}\left(\kappa_{\chi_{2x}}, \kappa_{\chi_{2y}}\right)$ are positive definite gain matrices. The function $\boldsymbol{\mu}(\tilde{\boldsymbol{\chi}}_1) = \left[\mu_x(\tilde{\boldsymbol{\chi}}_1)\ \mu_y(\tilde{\boldsymbol{\chi}}_1)\right]^\top$ is defined as

$$\mu_i(\tilde{\boldsymbol{\chi}}_1) = \begin{cases} \mathrm{sig}(\tilde{\chi}_{1_i})^{\alpha_{\tilde{\chi}_1}}, & |\tilde{\chi}_{1_i}| \ge \tilde{\gamma}_i \\ l_{\tilde{\chi}_{1_i}}\tilde{\chi}_{1_i} + l_{\tilde{\chi}_{2_i}}\mathrm{sig}(\tilde{\chi}_{1_i})^{\alpha_{\tilde{\chi}_2}}, & |\tilde{\chi}_{1_i}| < \tilde{\gamma}_i \end{cases} \tag{18}$$

https://doi.org/10.3390/act15050273

for $i = x, y$, where $0.5 < \alpha_{\tilde{\chi}_1} < 1$ and $1 < \alpha_{\tilde{\chi}_2} < 2$. Let us define $\text{sig}(\tilde{\chi}_{1_i})^{\alpha_{\tilde{\chi}_1}} = |\tilde{\chi}_{1_i}|^{\alpha_{\tilde{\chi}_1}} \text{sgn}(\tilde{\chi}_{1_i})$, where $\text{sgn}(\cdot)$ is the sign function. The time derivative of the sliding surface is given as

$$\dot{s}_\chi = \dot{\tilde{\chi}}_2 + \kappa_{\chi_1}\dot{\tilde{\chi}}_1 + \kappa_{\chi_2}\dot{\mu}(\tilde{\chi}_1) \tag{19}$$

where

$$\dot{\mu}_i(\tilde{\chi}_1) = \begin{cases} \alpha_{\tilde{\chi}_1}|\tilde{\chi}_{1_i}|^{\alpha_{\tilde{\chi}_1}-1}\tilde{\chi}_{2_i}, & |\tilde{\chi}_{1_i}| \geq \tilde{\gamma}_i \\ l_{\tilde{\chi}_{1_i}}\tilde{\chi}_{2_i} + l_{\tilde{\chi}_{2_i}}\alpha_{\tilde{\chi}_2}|\tilde{\chi}_{1_i}|^{\alpha_{\tilde{\chi}_2}-1}\tilde{\chi}_{2_i}, & |\tilde{\chi}_{1_i}| < \tilde{\gamma}_i \end{cases} \tag{20}$$

Solving the system of equations:

$$\begin{aligned} \text{sig}(\tilde{\chi}_{1_i})^{\alpha_{\tilde{\chi}_1}}\Big|_{\tilde{\chi}_{1_i}=\tilde{\gamma}_i} &= \left(l_{\tilde{\chi}_{1_i}}\tilde{\chi}_{1_i} + l_{\tilde{\chi}_{2_i}}\text{sig}(\tilde{\chi}_{1_i})^{\alpha_{\tilde{\chi}_2}}\right)\Big|_{\tilde{\chi}_{1_i}=\tilde{\gamma}_i} \\ \alpha_{\tilde{\chi}_1}|\tilde{\chi}_{1_i}|^{\alpha_{\tilde{\chi}_1}-1}\Big|_{\tilde{\chi}_{1_i}=\tilde{\gamma}_i} &= \left(l_{\tilde{\chi}_1}\tilde{\chi}_{2_i} + l_{\tilde{\chi}_2}\alpha_{\tilde{\chi}_2}|\tilde{\chi}_{1_i}|^{\alpha_{\tilde{\chi}_2}-1}\right)\Big|_{\tilde{\chi}_{1_i}=\tilde{\gamma}_i} \end{aligned} \tag{21}$$

allows us to obtain the values of parameters $l_{\tilde{\chi}_{1_i}}$ and $l_{\tilde{\chi}_{2_i}}$ as

$$\begin{aligned} l_{\tilde{\chi}_{1_i}} &= \frac{\alpha_{\tilde{\chi}_2} - \alpha_{\tilde{\chi}_1}}{\alpha_{\tilde{\chi}_2} - 1}\gamma_i^{\alpha_{\tilde{\chi}_1}-1} \\ l_{\tilde{\chi}_{2_i}} &= \frac{\alpha_{\tilde{\chi}_1} - 1}{\alpha_{\tilde{\chi}_2} - 1}\gamma_i^{\alpha_{\tilde{\chi}_1}-\alpha_{\tilde{\chi}_2}} \end{aligned} \tag{22}$$

With the values computed for $l_{\tilde{\chi}_{1_i}}$ and $l_{\tilde{\chi}_{2_i}}$, the sliding surface (17) remains continuous.

3. Nonparametric Robust Dynamic Neural Network Identifier

To identify the dynamics of the USV described in (14), the following SM-DNNI is proposed

$$\begin{aligned} \dot{\hat{\chi}}_1 =& \hat{\chi}_2 + \mathbf{\Upsilon}\tilde{\chi}_1 \\ \dot{\hat{\chi}}_2 =& R(\psi)\hat{W}_1^\top \mathbf{\Phi}(\chi_1, \chi_2) + R(\psi)\hat{W}_2^\top \mathbf{\Theta}(\chi_1, \chi_2)u \\ &+ \kappa_{\chi_1}(\tilde{\chi}_2 - \mathbf{\Upsilon}\tilde{\chi}_1) + \kappa_{\chi_2}\dot{\mu}(\tilde{\chi}_1) + \mathbf{\Lambda}_{\chi_1}s_\chi \\ &+ \mathbf{\Lambda}_{\chi_2}\mathbf{sig}(s_\chi)^{\sigma_\chi} + \mathbf{\Lambda}_{\chi_3}\mathbf{sgn}(s_\chi) \end{aligned} \tag{23}$$

where $\mathbf{sig}(s_\chi)^{1/2} \triangleq \left[|s_{\chi_x}|^{1/2}\text{sgn}(s_{\chi_x})\ |s_{\chi_y}|^{1/2}\text{sgn}\left(s_{\chi_y}\right)\right]^\top$, and $0 < \sigma_\chi < 1$. The neural weights $\hat{W}_1$ and $\hat{W}_2$ have a structure similar to the initial weights defined in (15), i.e., $\hat{W}_1 = \left[\hat{W}_{1_1}\ \hat{W}_{1_2}\right] \in \mathbb{R}^{4\times 2}$, with $\hat{W}_{1_1}, \hat{W}_{1_2} \in \mathbb{R}^{4\times 1}$, and $\hat{W}_2 = \left[\hat{W}_{2_1}\ \mathbf{0}_{4\times 1};\ \mathbf{0}_{4\times 1}\ \hat{W}_{2_2}\right] \in \mathbb{R}^{8\times 2}$, with $\hat{W}_{2_1}, \hat{W}_{2_2} \in \mathbb{R}^{4\times 1}$. The matrices $\mathbf{\Upsilon}, \mathbf{\Lambda}_{\chi_1}, \mathbf{\Lambda}_{\chi_2}, \mathbf{\Lambda}_{\chi_3} \in \mathbb{R}^{2\times 2}$ are positive diagonal matrices.The function $\mathbf{sgn}(s_\chi)$ is defined element-wise as $\mathbf{sgn}(s_\chi) \triangleq [\text{sgn}(s_{\chi_1})\ \text{sgn}(s_{\chi_2})]^\top$.

From the system dynamics and identifier dynamics, the dynamics of the identification error are obtained as

$$\begin{aligned} \dot{\tilde{\chi}}_1 =& \tilde{\chi}_2 - \mathbf{\Upsilon}\tilde{\chi}_1 \\ \dot{\tilde{\chi}}_2 =& R(\psi)\tilde{W}_1^\top \mathbf{\Phi}(\chi_1, \chi_2) + R(\psi)\tilde{W}_2^\top \mathbf{\Theta}(\chi_1, \chi_2)u \\ &- \kappa_{\chi_1}(\tilde{\chi}_2 - \mathbf{\Upsilon}\tilde{\chi}_1) - \kappa_{\chi_2}\dot{\mu}(\tilde{\chi}_1) - \mathbf{\Lambda}_{\chi_1}s_\chi \\ &- \mathbf{\Lambda}_{\chi_2}\mathbf{sig}(s_\chi)^{\sigma_\chi} - \mathbf{\Lambda}_{\chi_3}\mathbf{sgn}(s_\chi) + \tilde{f}_\chi \end{aligned} \tag{24}$$

where the identification errors of the neural weights are defined as $\tilde{W}_1 = W_{01} - \hat{W}_1$ and $\tilde{W}_2 = W_{02} - \hat{W}_2$. Moreover, by substituting (24) into (19), we obtain the following:

$$\begin{aligned} \dot{s}_\chi =& R(\psi)\tilde{W}_1^\top \mathbf{\Phi}(\chi_1, \chi_2) + R(\psi)\tilde{W}_2^\top \mathbf{\Theta}(\chi_1, \chi_2)u \\ &- \mathbf{\Lambda}_{\chi_1}s_\chi - \mathbf{\Lambda}_{\chi_2}\mathbf{sig}(s_\chi)^{\sigma_\chi} - \mathbf{\Lambda}_{\chi_3}\mathbf{sgn}(s_\chi) + \tilde{f}_\chi \end{aligned} \tag{25}$$

https://doi.org/10.3390/act15050273

Theorem 1. *Let Assumption 1 be satisfied. Consider the nonparametric neural network identifier defined in* (23) *for the transformed dynamics* (12) *of an unmanned surface vehicle. If the adaptive gains for the neural weights are chosen as*

$$\dot{\hat{W}}_1^\top = \Gamma_1 \Big[R(\psi)^\top s_\chi \Phi(\chi_1, \chi_2)^\top + \Sigma_1 \tilde{W}_1^\top + \Omega_1 SGN(\tilde{W}_1)^\top \Big] \tag{26}$$

$$\dot{\hat{W}}_2^\top = \Gamma_2 \Big[R(\psi)^\top s_\chi u^\top \Theta(\chi_1, \chi_2)^\top + \Sigma_2 \tilde{W}_2^\top + \Omega_2 SGN(\tilde{W}_2)^\top \Big] \tag{27}$$

then, the sliding surface s_χ and the estimation errors in the neural weights converge to zero in a finite time given by $\mathcal{T}_1$ defined in (48)*. From the sliding surface definition, the identification errors $\tilde{\chi}_1$ and $\tilde{\chi}_2$ converge in finite time to an arbitrarily small set bounded by $\tilde{\gamma}^+$ in a finite time given by $\mathcal{T}_\succ = \mathcal{T}_1 + \mathcal{T}_2$, where $\mathcal{T}_2$ is defined in (55). Once inside this set, the identification errors converge asymptotically to zero.*

Proof. For the update laws (26) and (27), the matrices Γ_i, Σ_i and Ω_i, $i = 1, 2$, are positive definite diagonal matrices. The functions $\mathbf{SGN}(\tilde{W}_1)^\top$ and $\mathbf{SGN}(\tilde{W}_2)^\top$ are defined element-wise as

$$\mathbf{SGN}(\tilde{W}_1) = \begin{bmatrix} \mathrm{sgn}(\tilde{w}_{11}) & \cdots & \mathrm{sgn}(\tilde{w}_{14}) \\ \mathrm{sgn}(\tilde{w}_{21}) & \cdots & \mathrm{sgn}(\tilde{w}_{24}) \end{bmatrix} \tag{28}$$

$$\mathbf{SGN}(\tilde{W}_2) = \begin{bmatrix} \mathrm{sgn}(\tilde{w}_{11}) & 0 \\ \cdots & \cdots \\ \mathrm{sgn}(\tilde{w}_{14}) & 0 \\ 0 & \mathrm{sgn}(\tilde{w}_{11}) \\ \cdots & \cdots \\ 0 & \mathrm{sgn}(\tilde{w}_{11}) \end{bmatrix} \tag{29}$$

where $\tilde{w}_{1i}$ are the elements of $(\tilde{W}_1)$ and $\tilde{w}_{2i}$ are the elements of $(\tilde{W}_2)$, $i = 1, \ldots, 4$. The stability proof of the theorem is divided into two steps. The first is for the sliding phase, and the second is for the reaching phase.

Step 1. Sliding phase. Consider the following Lyapunov candidate function

$$V_1 = \frac{1}{2} s_\chi^\top s_\chi + \frac{1}{2} \mathrm{tr}\Big(\tilde{W}_1 \Gamma_1^{-1} \tilde{W}_1^\top \Big) + \frac{1}{2} \mathrm{tr}\Big(\tilde{W}_2 \Gamma_2^{-1} \tilde{W}_2^\top \Big) \tag{30}$$

The time derivative is given as

$$\begin{aligned} \dot{V}_1 = & - s_\chi^\top \Lambda_{\chi_1} s_\chi - s_\chi^\top \Lambda_{\chi_2} \mathbf{sig}(s_\chi)^{\sigma_\chi} - s_\chi^\top \Lambda_{\chi_3} \mathbf{sgn}(s_\chi) + s_\chi^\top \tilde{f}_\chi \\ & + s_\chi^\top R(\psi) \tilde{W}_1^\top \Phi(\chi_1, \chi_2) + \mathrm{tr}\Big(\tilde{W}_1 \Gamma_1^{-1} \dot{\tilde{W}}_1^\top \Big) \\ & + s_\chi^\top R(\psi) \tilde{W}_2^\top \Theta(\chi_1, \chi_2) u + \mathrm{tr}\Big(\tilde{W}_2 \Gamma_2^{-1} \dot{\tilde{W}}_2^\top \Big) \end{aligned} \tag{31}$$

Notice that

$$\begin{aligned} s_\chi^\top R(\psi) \tilde{W}_1^\top \Phi(\chi_1, \chi_2) + \mathrm{tr}\Big(\tilde{W}_1 \Gamma_1^{-1} \dot{\tilde{W}}_1^\top \Big) &= \mathrm{tr}\Big(s_\chi^\top R(\psi) \tilde{W}_1^\top \Phi(\chi_1, \chi_2) \Big) - \mathrm{tr}\Big(\tilde{W}_1 \Gamma_1^{-1} \dot{\hat{W}}_1^\top \Big) \\ &= \mathrm{tr}\Big(\tilde{W}_1 \Big[R(\psi)^\top s_\chi \Phi(\chi_1, \chi_2)^\top - \Gamma_1^{-1} \dot{\hat{W}}_1^\top \Big] \Big) \end{aligned}$$

and

$$s_\chi^\top R(\psi)\tilde{W}_2^\top \Theta(\chi_1,\chi_2)u + \mathrm{tr}\left(\tilde{W}_2\Gamma_2^{-1}\dot{\tilde{W}}_2^\top\right) = \mathrm{tr}\left(s_\chi^\top R(\psi)\tilde{W}_2{}^\top \Theta(\chi_1,\chi_2)u\right) - \mathrm{tr}\left(\tilde{W}_2\Gamma_2^{-1}\dot{\hat{W}}_2^\top\right)$$
$$= \mathrm{tr}\left(\tilde{W}_2\left[R(\psi)^\top s_\chi u^\top \Theta(\chi_1,\chi_2)^\top - \Gamma_1^{-1}\dot{\hat{W}}_1^\top\right]\right)$$

Then, Equation (31) yields

$$\begin{aligned}\dot{V}_1 \leq & -s_\chi^\top \Lambda_{\chi_1} s_\chi - s_\chi^\top \Lambda_{\chi_3}\mathbf{sgn}(s_\chi) + s_\chi^\top \tilde{f}_\chi + \mathrm{tr}\left(\tilde{W}_1\left[R(\psi)^\top s_\chi \Phi(\chi_1,\chi_2)^\top - \Gamma_1^{-1}\dot{\hat{W}}_1^\top\right]\right)\\ & + \mathrm{tr}\left(\tilde{W}_2\left[R(\psi)^\top s_\chi u^\top \Theta(\chi_1,\chi_2)^\top - \Gamma_1^{-1}\dot{\hat{W}}_1^\top\right]\right)\end{aligned} \tag{32}$$

By substituting the adaptive laws defined in equations (26) and (27), the following is obtained:

$$\begin{aligned}\dot{V}_1 \leq & -s_\chi^\top \Lambda_{\chi_1} s_\chi - s_\chi^\top \Lambda_{\chi_3}\mathbf{sgn}(s_\chi) + s_\chi^\top \tilde{f}_\chi - \mathrm{tr}\left(\tilde{W}_1\Sigma_1\tilde{W}_1^\top + \tilde{W}_1\Omega_1\mathbf{SGN}(\tilde{W}_1)^\top\right)\\ & - \mathrm{tr}\left(\tilde{W}_2\Sigma_2\tilde{W}_2^\top + \tilde{W}_2\Omega_2\mathbf{SGN}(\tilde{W}_2)^\top\right)\end{aligned} \tag{33}$$

Notice that for the term $-s_\chi^\top \Lambda_{\chi_3}\mathbf{sgn}(s_\chi) + s_\chi^\top \tilde{f}_\chi$, and based on the Assumption 1, it follows that

$$\begin{aligned}& -s_\chi^\top \Lambda_{\chi_3}\mathbf{sgn}(s_\chi) + s_\chi^\top \tilde{f}_\chi \leq -|s_1|\left(\lambda_{\chi_{31}} - f_{\chi_x}^+\right)\\ & -|s_2|\left(\lambda_{\chi_{32}} - f_{\chi_y}^+\right) \leq -s_\chi^\top \Lambda_3^+\mathbf{sgn}(s_\chi)\end{aligned} \tag{34}$$

where $\Lambda_{\chi_3}^+ = \mathrm{diag}\left(\lambda_{\chi_{31}}^+, \lambda_{\chi_{32}}^+\right)$, with $\lambda_{\chi_{31}}^+ = \lambda_{\chi_{31}} - f_{\chi_x}^+ > 0$ and $\lambda_{\chi_{32}}^+ = \lambda_{\chi_{32}} - f_{\chi_y}^+ > 0$. Then, $\dot{V}_1$ yields

$$\begin{aligned}\dot{V}_1 \leq & -s_\chi^\top \Lambda_{\chi_1} s_\chi - \mathrm{tr}\left(\tilde{W}_1\Sigma_1\tilde{W}_1^\top\right) - \mathrm{tr}\left(\tilde{W}_2\Sigma_2\tilde{W}_2^\top\right)\\ & - s_\chi^\top \Lambda_{\chi_3}^+\mathbf{sgn}(s_\chi) - \mathrm{tr}\left(\tilde{W}_1\Omega_1\mathbf{SGN}(\tilde{W}_1)^\top\right)\\ & - \mathrm{tr}\left(\tilde{W}_2\Omega_2\mathbf{SGN}(\tilde{W}_2)^\top\right)\end{aligned} \tag{35}$$

Notice that V_1 can be rewritten as

$$V_1 = \frac{1}{2}\sum_{i=1}^{2} s_i + \frac{1}{2}\sum_{i=1}^{2}\Gamma_{1i}^{-1}\tilde{w}_{1i}^\top\tilde{w}_{1i} + \frac{1}{2}\sum_{i=1}^{2}\Gamma_{2i}^{-1}\tilde{w}_{2i}^\top\tilde{w}_{2i}, \tag{36}$$

then

$$\begin{aligned}& -\frac{1}{2}s_\chi^\top s_\chi - \frac{1}{2}\lambda_{\max}(\Gamma_1)\mathrm{tr}\left(\tilde{W}_1\tilde{W}_1^\top\right)\\ & \qquad - \frac{1}{2}\lambda_{\max}(\Gamma_2)\mathrm{tr}\left(\tilde{W}_2\tilde{W}_2^\top\right) \leq -V_1.\end{aligned} \tag{37}$$

Let us define the first three terms of (35) as follows

$$\dot{V}_\chi = -s_\chi^\top \Lambda_{\chi_1} s_\chi - \mathrm{tr}\left(\tilde{W}_1\Sigma_1\tilde{W}_1^\top\right) - \mathrm{tr}\left(\tilde{W}_2\Sigma_2\tilde{W}_2^\top\right), \tag{38}$$

https://doi.org/10.3390/act15050273

where (38) yields

$$\begin{aligned}\dot{V}_\chi \leq &-\lambda_{\min}(\mathbf{\Lambda_{\chi_1}})\boldsymbol{s}_\chi^\top \boldsymbol{s}_\chi - \lambda_{\min}(\mathbf{\Sigma_1})\mathrm{tr}\left(\tilde{\boldsymbol{W}}_\mathbf{1}\tilde{\boldsymbol{W}}_\mathbf{1}^\top\right)\\ &-\lambda_{\min}(\mathbf{\Sigma_2})\mathrm{tr}\left(\tilde{\boldsymbol{W}}_\mathbf{2}\tilde{\boldsymbol{W}}_\mathbf{2}^\top\right)\end{aligned} \tag{39}$$

Then, for inequalities (37) and (39), the following is obtained: $\dot{V}_\chi \leq -\alpha V_1$ where

$$\alpha = \min\left(2\lambda_{\min}(\mathbf{\Lambda_{\chi_1}}), \frac{2\lambda_{\min}(\mathbf{\Sigma_1})}{\lambda_{\max}(\mathbf{\Gamma_1})}, \frac{2\lambda_{\min}(\mathbf{\Sigma_2})}{\lambda_{\max}(\mathbf{\Gamma_2})}\right) \tag{40}$$

and $\dot{V}_1$ yields

$$\begin{aligned}\dot{V}_1 \leq &-\alpha V_1 - \boldsymbol{s}_\chi^\top \mathbf{\Lambda}_{\chi_3}^+ \mathbf{sgn}(\boldsymbol{s}_\chi) - \mathrm{tr}\left(\tilde{\boldsymbol{W}}_\mathbf{1}\mathbf{\Omega_1}\mathbf{SGN}(\tilde{\boldsymbol{W}}_\mathbf{1})^\top\right)\\ &-\mathrm{tr}\left(\tilde{\boldsymbol{W}}_\mathbf{2}\mathbf{\Omega_2}\mathbf{SGN}(\tilde{\boldsymbol{W}}_\mathbf{2})^\top\right)\end{aligned} \tag{41}$$

Next, using V_1 defined in (30) and inequality (37), we obtain

$$\begin{aligned}V_1 \leq &\frac{1}{2}\boldsymbol{s}_\chi^\top \boldsymbol{s}_\chi + \frac{1}{2}\lambda_{\max}(\mathbf{\Gamma_1})\mathrm{tr}\left(\tilde{\boldsymbol{W}}_\mathbf{1}\tilde{\boldsymbol{W}}_\mathbf{1}^\top\right)\\ &+\frac{1}{2}\lambda_{\max}(\mathbf{\Gamma_2})\mathrm{tr}\left(\tilde{\boldsymbol{W}}_\mathbf{2}\tilde{\boldsymbol{W}}_\mathbf{2}^\top\right)\\ V_1 \leq &\beta_2 \mathbf{\Pi}\end{aligned} \tag{42}$$

where

$$\begin{aligned}\mathbf{\Pi} =&\boldsymbol{s}_\chi^\top \boldsymbol{s}_\chi + \mathrm{tr}\left(\tilde{\boldsymbol{W}}_\mathbf{1}\tilde{\boldsymbol{W}}_\mathbf{1}^\top\right) + \mathrm{tr}\left(\tilde{\boldsymbol{W}}_\mathbf{2}\tilde{\boldsymbol{W}}_\mathbf{2}^\top\right)\\ \beta_2 =&\max\left[\frac{1}{2}, \frac{1}{2}\lambda_{\max}(\mathbf{\Gamma_1}), \frac{1}{2}\lambda_{\max}(\mathbf{\Gamma_2})\right]\end{aligned}$$

then, for $\mathbf{\Pi}$

$$-\mathbf{\Pi}^{1/2} \leq -\frac{1}{\beta_2^{1/2}} V_1^{1/2} \tag{43}$$

Now, for $\dot{V}_1$ in (41)

$$\begin{aligned}\dot{V}_1 \leq &-\alpha V_1 - \lambda_{\min}\left(\mathbf{\Lambda}_{\chi_3}^+\right)\boldsymbol{s}_\chi^\top \mathbf{sgn}(\boldsymbol{s}_\chi)\\ &-\lambda_{\min}(\mathbf{\Omega_1})\mathrm{tr}\left(\tilde{\boldsymbol{W}}_\mathbf{1}\mathbf{SGN}(\tilde{\boldsymbol{W}}_\mathbf{1})^\top\right)\\ &-\lambda_{\min}(\mathbf{\Omega_2})\mathrm{tr}\left(\tilde{\boldsymbol{W}}_\mathbf{2}\mathbf{SGN}(\tilde{\boldsymbol{W}}_\mathbf{2})^\top\right)\\ \leq &-\alpha V_1 - \beta_1\left[\boldsymbol{s}_\chi^\top \mathbf{sgn}(\boldsymbol{s}_\chi) + \mathrm{tr}\left(\tilde{\boldsymbol{W}}_\mathbf{1}\mathbf{SGN}(\tilde{\boldsymbol{W}}_\mathbf{1})^\top\right)\right.\\ &\left.+\mathrm{tr}\left(\tilde{\boldsymbol{W}}_\mathbf{2}\mathbf{SGN}(\tilde{\boldsymbol{W}}_\mathbf{2})^\top\right)\right]\end{aligned} \tag{44}$$

where

$$\beta_1 = \min\left(\lambda_{\min}\left(\mathbf{\Lambda}_{\chi_3}^+\right), \lambda_{\min}(\mathbf{\Omega_1}), \lambda_{\min}(\mathbf{\Omega_2})\right)$$

https://doi.org/10.3390/act15050273

Notice that

$$\begin{aligned} &\boldsymbol{s}_{\chi}^{\top}\mathbf{sgn}(\boldsymbol{s}_{\chi}) = |s_1| + |s_2| \\ &\mathrm{tr}\Big(\tilde{\boldsymbol{W}}_{\mathbf{1}}\mathbf{SGN}\big(\tilde{\boldsymbol{W}}_{\mathbf{1}}\big)^{\top}\Big) = |w_{11_1}| + |w_{11_2}| + |w_{11_3}| + \\ &|w_{12_1}| + |w_{12_2}| + |w_{12_3}| \\ &\mathrm{tr}\Big(\tilde{\boldsymbol{W}}_{\mathbf{2}}\mathbf{SGN}\big(\tilde{\boldsymbol{W}}_{\mathbf{2}}\big)^{\top}\Big) = |w_{21_1}| + |w_{21_2}| + |w_{22_1}| + |w_{22_2}| \end{aligned} \tag{45}$$

Then, from (45) and Lemma 1 the following is obtained for $\dot{V}_1$

$$\begin{aligned} \dot{V}_1 \leq &-\alpha V_1 - \beta_1\Big[\boldsymbol{s}_{\chi}^{\top}\mathbf{sgn}(\boldsymbol{s}_{\chi}) + \mathrm{tr}\Big(\tilde{\boldsymbol{W}}_{\mathbf{1}}\mathbf{SGN}\big(\tilde{\boldsymbol{W}}_{\mathbf{1}}\big)^{\top}\Big) \\ &+ \mathrm{tr}\Big(\tilde{\boldsymbol{W}}_{\mathbf{2}}\mathbf{SGN}\big(\tilde{\boldsymbol{W}}_{\mathbf{2}}\big)^{\top}\Big)\Big] \\ \leq &-\alpha V_1 - \beta_1\Big[\boldsymbol{s}_{\chi}^{\top}\boldsymbol{s}_{\chi} + \mathrm{tr}\Big(\tilde{\boldsymbol{W}}_{\mathbf{1}}\tilde{\boldsymbol{W}}_{\mathbf{1}}^{\top}\Big) + \mathrm{tr}\Big(\tilde{\boldsymbol{W}}_{\mathbf{2}}\tilde{\boldsymbol{W}}_{\mathbf{2}}^{\top}\Big)\Big]^{1/2} \end{aligned} \tag{46}$$

Then, from inequality (43)

$$\dot{V}_1 \leq -\alpha V_1 - \frac{\beta_1}{\beta_2^{1/2}}V_1^{1/2} \tag{47}$$

Finally, from Lemma 3, it is concluded that the sliding surface $\boldsymbol{s}_{\chi}$, and the errors in the neural weights $\tilde{\boldsymbol{W}}_{\mathbf{1}}$ and $\tilde{\boldsymbol{W}}_{\mathbf{2}}$ converge globally within a finite time $\mathcal{T}_1$, defined as

$$\mathcal{T}_1 \leq \frac{2\ln\left(1 + \frac{\alpha\beta_2^{1/2}}{\beta_1}V_1(0)^{1/2}\right)}{\alpha} \tag{48}$$

Step 2. Reaching phase. Having guaranteed $\boldsymbol{s}_{\chi} = \mathbf{0}$, the performance of the system enters the reaching phase, i.e.,

$$\tilde{\boldsymbol{\chi}}_{\mathbf{2}} = -\boldsymbol{\kappa}_{\chi_1}\tilde{\boldsymbol{\chi}}_{\mathbf{1}} - \boldsymbol{\kappa}_{\chi_2}\boldsymbol{\mu}(\tilde{\boldsymbol{\chi}}_{\mathbf{1}}) \tag{49}$$

From Equation (24), we obtain $\tilde{\boldsymbol{\chi}}_{\mathbf{2}} = \dot{\tilde{\boldsymbol{\chi}}}_{\mathbf{1}} + \mathbf{Y}\tilde{\boldsymbol{\chi}}_{\mathbf{1}}$. By substituting this into (49), the following differential equation is obtained for the identification error $\tilde{\boldsymbol{\chi}}_{\mathbf{1}}$:

$$\dot{\tilde{\boldsymbol{\chi}}}_{\mathbf{1}} = -\mathbf{Y}\tilde{\boldsymbol{\chi}}_{\mathbf{1}} - \boldsymbol{\kappa}_{\chi_1}\tilde{\boldsymbol{\chi}}_{\mathbf{1}} - \boldsymbol{\kappa}_{\chi_2}\boldsymbol{\mu}(\tilde{\boldsymbol{\chi}}_{\mathbf{1}}) \tag{50}$$

Notice from Equations (17) and (63) that (50) is a continuous piecewise function. Therefore, the following two cases are analyzed.

Case A. In this case, the performance of (50) when $\|\boldsymbol{\mu}(\tilde{\boldsymbol{\chi}}_{\mathbf{1}})\| \geq \tilde{\gamma}^{+}$, with $\tilde{\zeta}^{+} = \sqrt{\tilde{\gamma}_x + \tilde{\gamma}_y}$, is analyzed, resulting in

$$\dot{\tilde{\boldsymbol{\chi}}}_{\mathbf{1}} = -\mathbf{Y}\tilde{\boldsymbol{\chi}}_{\mathbf{1}} - \boldsymbol{\kappa}_{\chi_1}\tilde{\boldsymbol{\chi}}_{\mathbf{1}} - \boldsymbol{\kappa}_{\chi_2}\mathbf{sig}(\tilde{\boldsymbol{\chi}}_{\mathbf{1}})^{\alpha_{\tilde{\chi}_1}} \tag{51}$$

where $\mathbf{sig}(\tilde{\boldsymbol{\chi}}_{\mathbf{1}})^{\alpha_{\tilde{\chi}_1}} = \Big[|\tilde{\chi}_{1_x}|^{\alpha_{\tilde{\chi}_1}}\mathrm{sgn}(\tilde{\chi}_{1_x})\ |\tilde{\chi}_{1_y}|^{\alpha_{\tilde{\chi}_1}}\mathrm{sgn}\Big(\tilde{\chi}_{1_y}\Big)\Big]^{\top}$ and $0.5 < \alpha_{\tilde{\chi}_1} < 1$. To analyze the stability of (51), the following Lyapunov candidate function is proposed

$$V_2 = \frac{1}{2}\tilde{\boldsymbol{\chi}}_{\mathbf{1}}^{\top}\tilde{\boldsymbol{\chi}}_{\mathbf{1}} \tag{52}$$

https://doi.org/10.3390/act15050273

The time derivative is obtained as

$$\begin{aligned}\dot{V}_2 =& -\tilde{\boldsymbol{\chi}}_1^\top(\mathbf{Y}+\boldsymbol{\kappa}_{\chi 1})\tilde{\boldsymbol{\chi}}_1 - \tilde{\boldsymbol{\chi}}_1^\top\boldsymbol{\kappa}_{\chi 2}\mathbf{sig}(\tilde{\boldsymbol{\chi}}_1)^{\alpha_{\tilde{\chi}_1}}\\ \leq& -\lambda_{\min}(\mathbf{Y}+\boldsymbol{\kappa}_{\chi 1})\tilde{\boldsymbol{\chi}}_1^\top\tilde{\boldsymbol{\chi}}_1 - \lambda_{\min}(\boldsymbol{\kappa}_{\chi 2})\times\\ & \tilde{\boldsymbol{\chi}}_1^\top\mathbf{sig}(\tilde{\boldsymbol{\chi}}_1)^{\alpha_{\tilde{\chi}_1}}\end{aligned} \tag{53}$$

From the inequality defined in Lemma 1, we obtain the following:

$$\dot{V}_2 \leq -2\lambda_{\min}(\mathbf{Y}+\boldsymbol{\kappa}_{\chi 1})V_2 - 2^{\frac{1+\alpha_{\tilde{\chi}_1}}{2}}\lambda_{\min}(\boldsymbol{\kappa}_{\chi 2})V_2^{\frac{1+\alpha_{\tilde{\chi}_1}}{2}} \tag{54}$$

Notice that the exponent $\frac{1+\alpha_{\tilde{\chi}_1}}{2}$ is restricted as $0.75 < \frac{1+\alpha_{\tilde{\chi}_1}}{2} < 1$. Therefore, from Lemma 3, we conclude that V_2 converges to zero in finite time $\mathcal{T}_2$, defined as

$$\mathcal{T}_2 \leq \frac{\ln\left(1+2^{\frac{1-\alpha_{\tilde{\chi}_1}}{2}}\frac{\lambda_{\min}(\mathbf{Y}+\boldsymbol{\kappa}_{\chi 1})}{\lambda_{\min}(\boldsymbol{\kappa}_{\chi 2})}V_2(0)^{\frac{1-\alpha_{\tilde{\chi}_1}}{2}}\right)}{(1-\alpha_{\tilde{\chi}_1})\lambda_{\min}(\mathbf{Y}+\boldsymbol{\kappa}_{\chi 1})} \tag{55}$$

Furthermore, from the condition for $\mu_i(\tilde{\boldsymbol{\chi}}_1)$ defined in (63), it is also concluded that the identification error $\tilde{\boldsymbol{\chi}}_1$ reaches an arbitrarily small set bounded by $\tilde{\zeta}^+$ in a finite time given by $\mathcal{T}_\chi = \mathcal{T}_1 + \mathcal{T}_2$.

Case B. In this case, the performance of the differential Equation (50) when $\|\boldsymbol{\mu}(\tilde{\boldsymbol{\chi}}_1)\| < \tilde{\gamma}^+$, is analyzed, resulting in

$$\dot{\tilde{\boldsymbol{\chi}}}_1 = -\mathbf{Y}\tilde{\boldsymbol{\chi}}_1 - \boldsymbol{\kappa}_{\chi 1}\tilde{\boldsymbol{\chi}}_1 - \boldsymbol{\kappa}_{\chi 2}\boldsymbol{l}_{\chi 1}\tilde{\boldsymbol{\chi}}_1 - \boldsymbol{\kappa}_{\chi 2}\boldsymbol{l}_{\chi 2}\mathbf{sig}(\tilde{\boldsymbol{\chi}}_1)^{\alpha_{\tilde{\chi}_2}}, \tag{56}$$

where $1 < \alpha_{\tilde{\chi}_2} < 2$. To analyze the stability of the differential Equation (56), the following Lyapunov function is proposed

$$V_3 = \frac{1}{2}\tilde{\boldsymbol{\chi}}_1^\top\tilde{\boldsymbol{\chi}}_1 \tag{57}$$

The time derivative is obtained as

$$\begin{aligned}\dot{V}_3 =& -\tilde{\boldsymbol{\chi}}_1^\top(\mathbf{Y}+\boldsymbol{\kappa}_{\chi 1}+\boldsymbol{\kappa}_{\chi 2}\boldsymbol{l}_{\chi 1})\tilde{\boldsymbol{\chi}}_1 - \tilde{\boldsymbol{\chi}}_1^\top\boldsymbol{\kappa}_{\chi 2}\boldsymbol{l}_{\chi 2}\mathbf{sig}(\tilde{\boldsymbol{\chi}}_1)^{\alpha_{\tilde{\chi}_2}}\\ \leq& -\lambda_{\min}(\mathbf{Y}+\boldsymbol{\kappa}_{\chi 1}+\boldsymbol{\kappa}_{\chi 2}\boldsymbol{l}_{\chi 1})\tilde{\boldsymbol{\chi}}_1^\top\tilde{\boldsymbol{\chi}}_1 - \lambda_{\min}(\boldsymbol{\kappa}_{\chi 2}\boldsymbol{l}_{\chi 2})\times\\ & \tilde{\boldsymbol{\chi}}_1^\top\mathbf{sig}(\tilde{\boldsymbol{\chi}}_1)^{\alpha_{\tilde{\chi}_2}}\end{aligned} \tag{58}$$

From the inequality defined in Lemma 2, we obtain the following:

$$\begin{aligned}\dot{V}_3 \leq& -2\lambda_{\min}(\mathbf{Y}+\boldsymbol{\kappa}_{\chi 1}+\boldsymbol{\kappa}_{\chi 2}\boldsymbol{l}_{\chi 1})V_3\\ & -2\lambda_{\min}(\boldsymbol{\kappa}_{\chi 2}\boldsymbol{l}_{\chi 1})V_3^{\frac{1+\alpha_{\tilde{\chi}_2}}{2}}\end{aligned} \tag{59}$$

Notice that the exponent $\alpha_{\tilde{\chi}_2}$ is restricted as $1 < \frac{1+\alpha_{\tilde{\chi}_2}}{2} < 1.5$. Therefore, for $\dot{V}_3$, we obtain the following:

$$\dot{V}_3 \leq -2\lambda_{\min}(\mathbf{Y}+\boldsymbol{\kappa}_{\chi 1}+\boldsymbol{\kappa}_{\chi 2}\boldsymbol{l}_{\chi 1})V_3 \tag{60}$$

Thus, it is concluded from (60) that the identification error $\tilde{\boldsymbol{\chi}}_1$ converges asymptotically to zero. Furthermore, from (49), the identification error $\tilde{\boldsymbol{\chi}}_2$ also converges to zero asymptotically. □

4. Control Strategy Design

Considering that the identification errors converge in finite time to a small bounded set close to zero, for control design purposes, the mathematical model represented by the

https://doi.org/10.3390/act15050273

identifier given in (23) is used. Let us define the tracking errors as $\boldsymbol{\zeta_1} = \boldsymbol{\chi_{1_d}} - \hat{\boldsymbol{\chi}}_{\mathbf{1}}$ and $\boldsymbol{\zeta_2} = \dot{\boldsymbol{\chi}}_{\mathbf{1}_d} - \dot{\hat{\boldsymbol{\chi}}}_{\mathbf{1}}$. The dynamics of the tracking errors are defined as follows

$$\begin{aligned}\dot{\boldsymbol{\zeta}}_{\mathbf{1}} =& \dot{\boldsymbol{\chi}}_{\mathbf{1}_d} - \hat{\boldsymbol{\chi}}_{\mathbf{2}} - \boldsymbol{\Upsilon}\tilde{\boldsymbol{\chi}}_{\mathbf{1}} \\ \dot{\boldsymbol{\zeta}}_{\mathbf{2}} =& \ddot{\boldsymbol{\chi}}_{\mathbf{1}_d} - \boldsymbol{R}(\psi)\hat{\boldsymbol{W}}_{\mathbf{1}}^{\top}\boldsymbol{\Phi}(\boldsymbol{\chi_1}, \boldsymbol{\chi_2}) - \boldsymbol{R}(\psi)\hat{\boldsymbol{W}}_{\mathbf{2}}^{\top}\boldsymbol{\Theta}(\boldsymbol{\chi_1}, \boldsymbol{\chi_2})\boldsymbol{u} \\ & - \boldsymbol{\kappa}_{\chi_1}(\tilde{\boldsymbol{\chi}}_{\mathbf{2}} - \boldsymbol{\Upsilon}\tilde{\boldsymbol{\chi}}_{\mathbf{1}}) - \boldsymbol{\kappa}_{\chi_2}\dot{\boldsymbol{\mu}}(\tilde{\boldsymbol{\chi}}_{\mathbf{1}}) - \boldsymbol{\Lambda}_{\chi_1}\boldsymbol{s}_{\chi} \\ & - \boldsymbol{\Lambda}_{\chi_2}\mathbf{sgn}(\boldsymbol{s}_{\chi}) - \boldsymbol{\Upsilon}(\tilde{\boldsymbol{\chi}}_{\mathbf{2}} - \boldsymbol{\Upsilon}\tilde{\boldsymbol{\chi}}_{\mathbf{1}})\end{aligned} \tag{61}$$

The sliding surface for the tracking errors is defined as

$$\boldsymbol{s}_{\zeta} = \boldsymbol{\zeta_2} + \boldsymbol{\kappa}_{\zeta_1}\boldsymbol{\zeta_1} + \boldsymbol{\kappa}_{\zeta_2}\boldsymbol{\mu}(\boldsymbol{\zeta_1}) \tag{62}$$

where $\boldsymbol{s}_{\zeta} = \begin{bmatrix} s_{\zeta_x} & s_{\zeta_y} \end{bmatrix}^{\top}$. The function $\boldsymbol{\mu}(\boldsymbol{\zeta_1}) = \begin{bmatrix} \mu_x(\boldsymbol{\zeta_1}) & \mu_y(\boldsymbol{\zeta_1}) \end{bmatrix}^{\top}$ is defined as

$$\mu_i(\boldsymbol{\zeta_1}) = \begin{cases} \mathrm{sig}(\zeta_{1_i})^{\alpha_{\zeta_1}}, & |\zeta_{1_i}| \geq \eta_i \\ l_{\zeta_{1_i}}\zeta_{1_i} + l_{\zeta_{2_i}}\mathrm{sig}(\zeta_{1_i})^{\alpha_{\zeta_2}}, & |\zeta_{1_i}| < \eta_i \end{cases} \tag{63}$$

for $i = x, y$, where $0.5 < \alpha_{\zeta_1} < 1$ and $1 < \alpha_{\zeta_2} < 2$. The temporal derivative of (62) is obtained as

$$\dot{\boldsymbol{s}}_{\zeta} = \dot{\boldsymbol{\zeta}}_{\mathbf{2}} + \boldsymbol{\kappa}_{\zeta_1}\dot{\boldsymbol{\zeta}}_{\mathbf{1}} + \boldsymbol{\kappa}_{\zeta_2}\dot{\boldsymbol{\mu}}(\boldsymbol{\zeta_1}) \tag{64}$$

Substituting the tracking error dynamics (61) yields

$$\begin{aligned}\dot{\boldsymbol{s}}_{\zeta} =& \ddot{\boldsymbol{\chi}}_{\mathbf{1}_d} - \boldsymbol{R}(\psi)\hat{\boldsymbol{W}}_{\mathbf{1}}^{\top}\boldsymbol{\Phi}(\boldsymbol{\chi_1}, \boldsymbol{\chi_2}) - \boldsymbol{R}(\psi)\hat{\boldsymbol{W}}_{\mathbf{2}}^{\top}\boldsymbol{\Theta}(\boldsymbol{\chi_1}, \boldsymbol{\chi_2})\boldsymbol{u} \\ & - \boldsymbol{\kappa}_{\chi_1}(\tilde{\boldsymbol{\chi}}_{\mathbf{2}} - \boldsymbol{\Upsilon}\tilde{\boldsymbol{\chi}}_{\mathbf{1}}) - \boldsymbol{\kappa}_{\chi_2}\dot{\boldsymbol{\mu}}(\tilde{\boldsymbol{\chi}}_{\mathbf{1}}) - \boldsymbol{\Lambda}_{\chi_1}\boldsymbol{s}_{\chi} - \boldsymbol{\Lambda}_{\chi_2}\mathbf{sgn}(\boldsymbol{s}_{\chi}) \\ & - \boldsymbol{\Upsilon}(\tilde{\boldsymbol{\chi}}_{\mathbf{2}} - \boldsymbol{\Upsilon}\tilde{\boldsymbol{\chi}}_{\mathbf{1}}) + \boldsymbol{\kappa}_{\zeta_1}(\dot{\boldsymbol{\chi}}_{\mathbf{1}_d} - \hat{\boldsymbol{\chi}}_{\mathbf{2}} - \boldsymbol{\Upsilon}\tilde{\boldsymbol{\chi}}_{\mathbf{1}}) + \boldsymbol{\kappa}_{\zeta_2}\dot{\boldsymbol{\mu}}(\boldsymbol{\zeta_1})\end{aligned} \tag{65}$$

Then, the following control signal is proposed

$$\begin{aligned}\boldsymbol{u} =& \left[\boldsymbol{R}(\boldsymbol{\psi})\hat{\boldsymbol{W}}_{\mathbf{2}}{}^{\top}\boldsymbol{\Theta}(\boldsymbol{\chi_1}, \boldsymbol{\chi_2})\right]^{-\mathbf{1}}\Big(- \boldsymbol{R}(\psi)\hat{\boldsymbol{W}}_{\mathbf{1}}^{\top}\boldsymbol{\Phi}(\boldsymbol{\chi_1}, \boldsymbol{\chi_2}) \\ & - \boldsymbol{\kappa}_{\chi_1}(\tilde{\boldsymbol{\chi}}_{\mathbf{2}} - \boldsymbol{\Upsilon}\tilde{\boldsymbol{\chi}}_{\mathbf{1}}) - \boldsymbol{\kappa}_{\chi_2}\dot{\boldsymbol{\mu}}(\tilde{\boldsymbol{\chi}}_{\mathbf{1}}) - \boldsymbol{\Lambda}_{\chi_1}\boldsymbol{s}_{\chi} - \boldsymbol{\Lambda}_{\chi_2}\mathbf{sgn}(\boldsymbol{s}_{\chi}) \\ & - \boldsymbol{\Upsilon}(\tilde{\boldsymbol{\chi}}_{\mathbf{2}} - \boldsymbol{\Upsilon}\tilde{\boldsymbol{\chi}}_{\mathbf{1}}) + \ddot{\boldsymbol{\chi}}_{\mathbf{1}_d} + \boldsymbol{\kappa}_{\zeta_1}(\dot{\boldsymbol{\chi}}_{\mathbf{1}_d} - \hat{\boldsymbol{\chi}}_{\mathbf{2}} - \boldsymbol{\Upsilon}\tilde{\boldsymbol{\chi}}_{\mathbf{1}}) \\ & + \boldsymbol{\kappa}_{\zeta_2}\dot{\boldsymbol{\mu}}(\boldsymbol{\zeta_1}) + \boldsymbol{\Lambda}_{\zeta_1}\boldsymbol{s}_{\zeta} + \boldsymbol{\Lambda}_{\zeta_2}\mathbf{sig}(\boldsymbol{s}_{\zeta})^{\sigma_{\zeta}}\Big)\end{aligned} \tag{66}$$

where $0 < \sigma_{\zeta} < 1$, and $\boldsymbol{\Lambda}_{\zeta_1}, \boldsymbol{\Lambda}_{\zeta_2} \in \mathbb{R}^{2\times 2}$ are positive diagonal matrices.

Remark 1. *It is important to emphasize that the control law is designed based on the SM-DNNI defined in Equation* (23)*, while it is applied to the USV dynamic model given in Equation* (5)*. Accordingly, the control signal is formulated using the identification errors, the sliding surface* $\boldsymbol{s}_{\chi}$ *associated with the identification errors, and the sliding surface* $\boldsymbol{s}_{\zeta}$ *associated with the tracking errors.*

Substituting the control signal defined in (66) into (65) yields

$$\dot{\boldsymbol{s}}_{\zeta} = -\boldsymbol{\Lambda}_{\zeta_1}\boldsymbol{s}_{\zeta} - \boldsymbol{\Lambda}_{\zeta_2}\mathbf{sig}(\boldsymbol{s}_{\zeta})^{\sigma_{\zeta}} \tag{67}$$

In order to assess the stability of the tracking errors between the reference signal and the SM-DNNI, the following Lyapunov candidate function is proposed:

$$V_4 = \frac{1}{2}\boldsymbol{s}_{\zeta}^{\top}\boldsymbol{s}_{\zeta} \tag{68}$$

https://doi.org/10.3390/act15050273

The temporal derivative is obtained as

$$\begin{aligned}\dot{V}_4 =& -\boldsymbol{s}_{\boldsymbol{\zeta}}^{\top}\boldsymbol{\Lambda}_{\boldsymbol{\zeta_1}}\boldsymbol{s}_{\boldsymbol{\zeta}} - \boldsymbol{s}_{\boldsymbol{\zeta}}^{\top}\boldsymbol{\Lambda}_{\boldsymbol{\zeta_2}}\mathbf{sig}(\boldsymbol{s}_{\boldsymbol{\zeta}})^{\sigma_\zeta} \\ \leq& -\lambda_{\min}(\boldsymbol{\Lambda}_{\boldsymbol{\zeta_1}})\boldsymbol{s}_{\boldsymbol{\zeta}}^{\top}\boldsymbol{s}_{\boldsymbol{\zeta}} - \lambda_{\min}(\boldsymbol{\Lambda}_{\boldsymbol{\zeta_2}})\boldsymbol{s}_{\boldsymbol{\zeta}}^{\top}\mathbf{sig}(\boldsymbol{s}_{\boldsymbol{\zeta}})^{\sigma_\zeta}\end{aligned} \tag{69}$$

From Lemma 1 the following expression is obtained for $\dot{V}_4$:

$$\dot{V}_4 \leq -2\lambda_{\min}(\boldsymbol{\Lambda}_{\boldsymbol{\zeta_1}})V_4 - 2^{\frac{1+\sigma_\zeta}{2}}\lambda_{\min}(\boldsymbol{\Lambda}_{\boldsymbol{\zeta_2}})V_4^{\frac{1+\sigma_\zeta}{2}} \tag{70}$$

Then, from Lemma 3, it is concluded that the tracking error between the reference signal and the SM-DNNI converges to zero in a finite time

$$\mathcal{T}_4 \leq \frac{\ln\left(1 + 2^{\frac{1-\sigma_\zeta}{2}}\frac{\lambda_{\min}(\boldsymbol{\Lambda}_{\boldsymbol{\zeta_1}})}{\lambda_{\min}(\boldsymbol{\Lambda}_{\boldsymbol{\zeta_2}})}V_4(0)^{\frac{1-\sigma_\zeta}{2}}\right)}{(1-\sigma_\zeta)\lambda_{\min}(\boldsymbol{\Lambda}_{\boldsymbol{\zeta_1}})} \tag{71}$$

Remark 2. *For the simulation analysis, the tracking error between the reference signal and the transformed coordinate* $\boldsymbol{\chi}$ *is defined by* $\tilde{\boldsymbol{\zeta}}_{\mathbf{1}} = [\tilde{\zeta}_{1_x}\ \tilde{\zeta}_{1_y}]^{\top}$ *as*

$$\tilde{\boldsymbol{\zeta}}_{\mathbf{1}} = \boldsymbol{\chi}_{\mathbf{1}_d} - \boldsymbol{\chi}_{\mathbf{1}} = \begin{bmatrix}\chi_{1_{dx}} \\ \chi_{1_{dy}}\end{bmatrix} - \begin{bmatrix}\chi_{1_x} \\ \chi_{1_y}\end{bmatrix} \tag{72}$$

5. Simulation Results

To observe the effectiveness of the nonparametric dynamic neural network identifier and the nonsingular terminal sliding-mode controller, simulations under saturation constraints in the control signal were conducted using the USV model given in Equation (5). Furthermore, scenarios with periodic external disturbances and with external disturbances generated by waves were considered to highlight the robustness of the proposed controller, thereby emphasizing the robust capability of the proposed strategy. Table 1 summarizes the USV's parameters used in the simulation sets, as reported in [32].

Table 1. Underwater surface vehicle parameters.

Parameter	Value	Parameter	Value/Equation
m	30 kg	X_u	-25, $(u > 1.2)$ 64.55
I_z	4.1 kg m^2	Y_v	$f(v)$*
L	1.01 m	N_r	$-\pi\rho\sqrt{u^2+v^2}T^2L^2$
T	0.09 m	$X_{u\|u\|}$	0, $(u > 1.2)$ -70.92
B_h	0.27 m	$Y_{v\|v\|}$	-99.99
$X_{\dot{u}}$	-2.25	$Y_{\dot{v}}$	-23.13
$N_{\dot{r}}$	-2.79	$N_{r\|r\|}$	-3.49

*$f(v) = -40\rho|v|\left[1.1 + 0.0045\frac{L}{T} - 0.1\frac{B_h}{T} + 0.016\left(\frac{B_h}{T}\right)^2\right]\left(\frac{\pi TL}{2}\right)$.

Table 2 lists the gains employed for the robust neural network identifiers, while Table 3 presents the gains used for the sliding-mode controller. The identifier gains were obtained using a low-performance controller, i.e., one designed from a coarse process model to collect identification data. Once the identifier gains yielded acceptable performance, the controller gains were subsequently tuned heuristically. A brief description of the tuning procedure for the SM-DNNI gains listed in Tables 2 and 3 is provided in the next remark.

Note for practitioners: regarding the SM-DNNI algorithm tuning procedure, a set of parameters is adjusted by trial and error to perform effective identification and control

https://doi.org/10.3390/act15050273

of the unknown dynamics. The gains $\boldsymbol{k_{\chi_1}}, \boldsymbol{k_{\chi_2}}, \alpha_{\tilde{\chi}_1}, \alpha_{\tilde{\chi}_2}$ determine mostly the identifier performance in the reaching phase. There exists a trade-off when selecting its magnitude; it increases the convergence velocity to the sliding surface for the identification error, but also increments the chattering generated over the sliding surface. A similar procedure is followed for the gains $\boldsymbol{k_{\zeta_1}}, \boldsymbol{k_{\zeta_2}}, \alpha_{\zeta_1}, \alpha_{\zeta_2}, \eta_x, \eta_y$ to define sliding surface for the tracking error. Gains $\boldsymbol{\Upsilon}$, $\boldsymbol{\Lambda_{\chi_1}}$, $\boldsymbol{\Lambda_{\chi_2}}$, $\boldsymbol{\Lambda_{\chi_3}}$ weigh the linear terms associated with the identification error vectors during the reaching phase and the sliding phase. Greater values contribute to the convergence of the identification errors, mainly in the sliding phase. The gains $\boldsymbol{\Gamma_1}$ and $\boldsymbol{\Gamma_2}$ are related to the learning rate. If the learning rate increments its value, the DNN learns faster, but it can introduce oscillations in the learning process; if it has lower values, it reflects slow learning, affecting the identifier's performance. The matrix gains $\boldsymbol{\Sigma_1}$ and $\boldsymbol{\Sigma_2}$ multiply the identification errors of neural weights; their magnitude weighs these correction terms to force the weights to converge to their nominal values. The gains $\boldsymbol{\Omega_1}$ and $\boldsymbol{\Omega_2}$ weight the discontinuous terms to force the neural weights to the nominal ones in finite time. If the values are too large, they may lead to abrupt oscillations of the neural weights during the reaching phase. The elements of the initial weights $\boldsymbol{W_{02}}$ were selected to be greater than $\boldsymbol{W_{01}}$ because $\boldsymbol{W_2}$ requires a greater approximation capacity as it is located in the inverse compensation term in (66), avoiding any singularity in the control law. The gains $\boldsymbol{\Lambda_{\zeta_1}}$ and $\boldsymbol{\Lambda_{\zeta_2}}$ are included in the tracking control, and their values are selected by trial and error in order to guarantee fast and stable dynamics for the tracking sliding surface (62). A more detailed description of the structure of the neural network used is presented in Appendix A.

Table 2. Gains of the robust neural network identifier.

Gain	Value
$\boldsymbol{k_{\chi_1}}, \boldsymbol{k_{\chi_2}}$	$\mathrm{diag}(2.25, 2.25),\ \mathrm{diag}(2.25, 2.25)$
$\tilde{\alpha}_x,\ \tilde{\alpha}_y$	$0.1,\ 0.1$
$\alpha_{\tilde{\chi}_1},\ \alpha_{\tilde{\chi}_2},\ \alpha_{\chi}$	$0.9,\ 1.7,\ 0.9$
$\boldsymbol{\Upsilon}$	$\mathrm{diag}(1.5, 1.5)$
$\boldsymbol{\Lambda_{\chi_1}}, \boldsymbol{\Lambda_{\chi_2}}, \boldsymbol{\Lambda_{\chi_3}}$	$\mathrm{diag}(15, 15),\ \mathrm{diag}(5, 5),\ \ \mathrm{diag}(0.1, 0.1)$
$\boldsymbol{\Gamma_1},\ \boldsymbol{\Gamma_2},$	$\mathrm{diag}(0.5, 0.5),\ \mathrm{diag}(0.2, 0.2)$
$\boldsymbol{\Sigma_1},\ \boldsymbol{\Sigma_2},$	$\mathrm{diag}(0.1, 0.1),\ \mathrm{diag}(0.1, 0.1)$
$\boldsymbol{\Omega_1},\ \boldsymbol{\Omega_2},$	$\mathrm{diag}(1.2, 1.2),\ \mathrm{diag}(0.75, 0.75)$
$\boldsymbol{W_{01_1}},\ \boldsymbol{W_{01_2}}$	$[1.2,\ 2,\ 1.6,\ 2.6]^\top,\ [3.6,\ 2.7,\ 4.9,\ 4]^\top$
$\boldsymbol{W_{02_1}}$	$[0.019,\ 0.049,\ 0.061,\ 0.033]^\top$
$\boldsymbol{W_{02_2}}$	$[0.09,\ 0.68,\ 0.48,\ 0.78]^\top$

Table 3. Gains of the sliding-mode control strategy.

Gain	Value
$\boldsymbol{k_{\zeta_1}},\ \boldsymbol{k_{\zeta_2}}$	$\mathrm{diag}(1.2, 1.2),\ \mathrm{diag}(1.2, 1.2)$
$\boldsymbol{\Lambda_{\zeta_1}},\ \boldsymbol{\Lambda_{\zeta_2}}$	$\mathrm{diag}(1, 1),\ \mathrm{diag}(1, 1)$
$\alpha_{\zeta_1},\ \alpha_{\zeta_2},\ \sigma_{\zeta}$	$0.9,\ 1.7,\ 0.9$
$\eta_x,\ \eta_y$	$0.1,\ 0.1$

In the simulation study, explicit actuator saturation constraints were imposed to reflect realistic operating conditions, with the surge force limited to $-200\ \mathrm{N} < \tau_u < 200\ \mathrm{N}$ and the yaw moment limited to $-100\ \mathrm{Nm} < \tau_r < 100\ \mathrm{Nm}$. These saturation bounds highlight the practical applicability of the proposed control strategy, as the commanded inputs do not exceed the physical capabilities of the propulsion system. For example, the saturation

bounds can be fulfilled using a propulsion system composed of two T500 thrusters from BlueRobotics [36], whose main specifications are summarized in Table 4. In particular, this propulsion configuration is capable of generating a maximum thrust of 315.6 N in the forward (FWD) direction and 205.8 N in reverse (REV) when operating at a nominal voltage of 24 V. Considering that the thrusters are located on the physical platform at a distance of 0.5 m from the CoG, the maximum torque that can be generated is 157.8 Nm in the forward (FWD) direction and 102.9 Nm in reverse (REV).

Table 4. T500 thruster main specifications.

Parameter	Value
Full Throttle FWD/REV Thrust @ 24 V	157.8/102.9 N
Depth Rating	300 m
Full Throttle Current and Power @ 24 V	43.5 A
Full Throttle RPM @ 24 V	3100 RPM
Propeller Diameter	114.5 mm
Propeller Pitch	25.5°

For comparison purposes, the performance of the control strategy proposed in this manuscript is compared with a super-twisting (ST)-based control law proposed in [17], which is defined as

$$\begin{aligned} \boldsymbol{u} =& \frac{1}{\boldsymbol{R}(\psi)\boldsymbol{g}(\boldsymbol{\chi_1},\boldsymbol{\chi_2})}\Big(-\boldsymbol{k_{ST}}\tilde{\boldsymbol{\zeta}}_{\boldsymbol{2}} - \boldsymbol{R}(\psi)\boldsymbol{f}(\boldsymbol{\chi_1},\boldsymbol{\chi_2}) + \ddot{\boldsymbol{\chi}}_{\boldsymbol{1_d}} \\ & -\boldsymbol{2L}|\boldsymbol{s_{ST}}|^{1/2}\mathbf{sgn}(\boldsymbol{s_{ST}}) - \frac{\boldsymbol{L^2}}{\boldsymbol{2}}\int \mathbf{sgn}(\boldsymbol{s_{ST}})\Big) \end{aligned} \tag{73}$$

where $\boldsymbol{k_{ST}}$ and $\boldsymbol{L}$ are positive diagonal matrices of dimension 2×2. The corresponding sliding surface is defined as

$$\boldsymbol{s_{ST}} = \tilde{\boldsymbol{\zeta}}_{\boldsymbol{2}} + \boldsymbol{k_{ST}}\tilde{\boldsymbol{\zeta}}_{\boldsymbol{1}} \tag{74}$$

with gains selected as $\boldsymbol{k_{ST}} = \mathrm{diag}(10, 10)$ and $\boldsymbol{L} = \mathrm{diag}(1, 1)$.

It is important to emphasize that the super-twisting-based control law is designed under the assumption that both the dynamic model structure and its parameters are known. In contrast, the purpose of the comparison is to demonstrate that a comparable level of tracking performance can be achieved using the proposed SM-DNNI strategy, even in the absence of prior knowledge of the system parameters.

The desired reference signal used in the simulations is given by

$$\begin{aligned} \chi_{1_{dx}} =& 12.5\sin\Big(\frac{\pi}{30}t\Big) \\ \chi_{1_{dy}} =& 12\cos\Big(\frac{\pi}{60}t\Big) - 12 \end{aligned} \tag{75}$$

This trajectory consists of single sinusoidal terms with large amplitudes and fixed frequencies; it is smooth and varies slowly. Additionally, the presence of a bias along the Y axis allows the evaluation of offset compensation.

Three simulation scenarios are presented. First, the SM-DNNI-based sliding-mode control is evaluated in the absence of external disturbances. Then, periodic external disturbances are introduced. Finally, wave-induced disturbances that affect the performance of the USV are considered.

https://doi.org/10.3390/act15050273

5.1. Simulation Free-Disturbances

Figure 2 illustrates the trajectory tracking in the X–Y plane for the reference signal defined in Equation (75), obtained using the SM-DNNI and ST controllers under disturbance-free conditions and with saturation applied to the control signal.

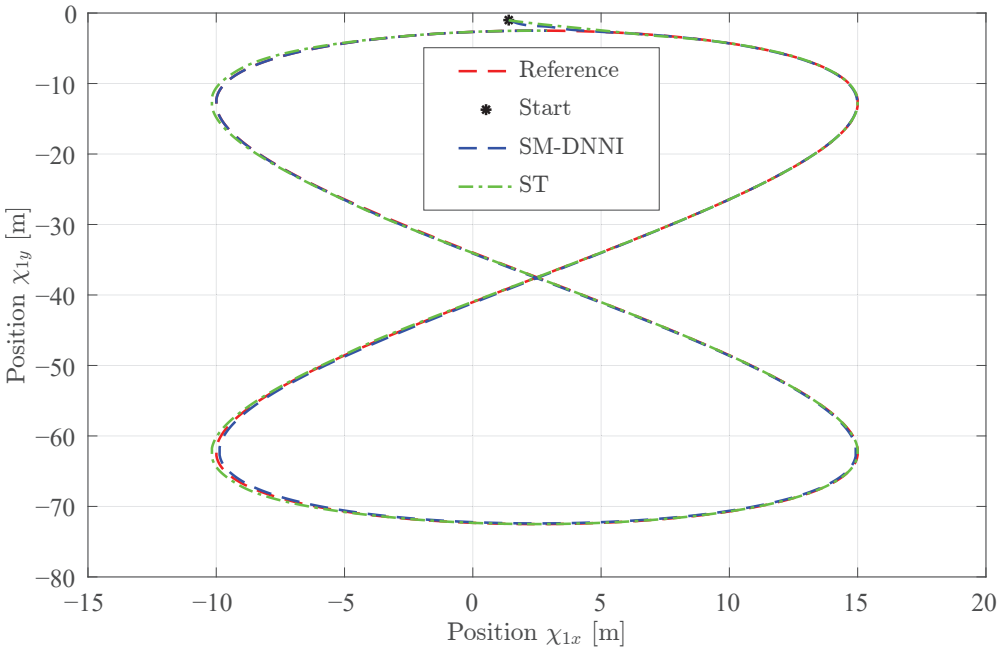
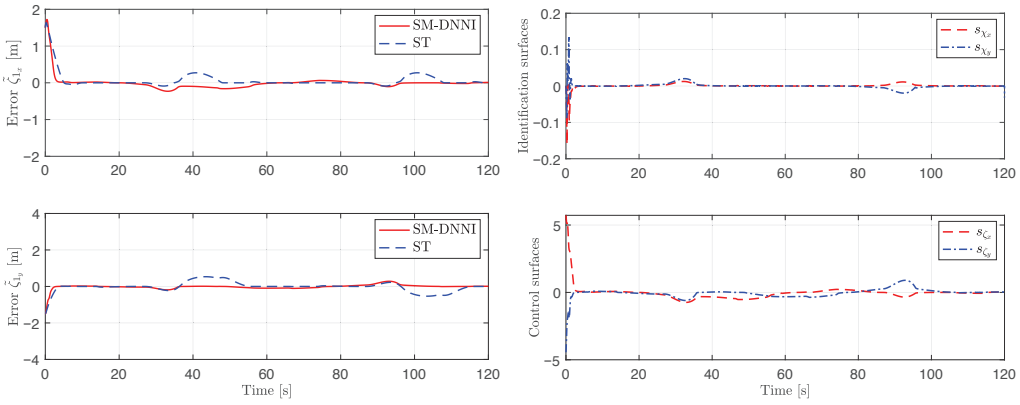


Figure 2. Trajectory tracking in the X-Y plane.

The corresponding tracking errors along the X and Y axes are presented in Figure 3a. Both the SM-DNNI and ST controllers track the reference signal with comparable settling times in the absence of disturbances. The ST controller was tuned under disturbance-free conditions and under saturation-free conditions to achieve performance comparable to, or exceeding, that of the controller defined in Equation (66), which is based on the SM-DNNI. Figure 3b presents the sliding surface of the neural network identification errors and the sliding surfaces of the neural controller along the X and Y axes, respectively. It can be seen that both sliding surfaces practically converge in finite time to a bounded region near zero.

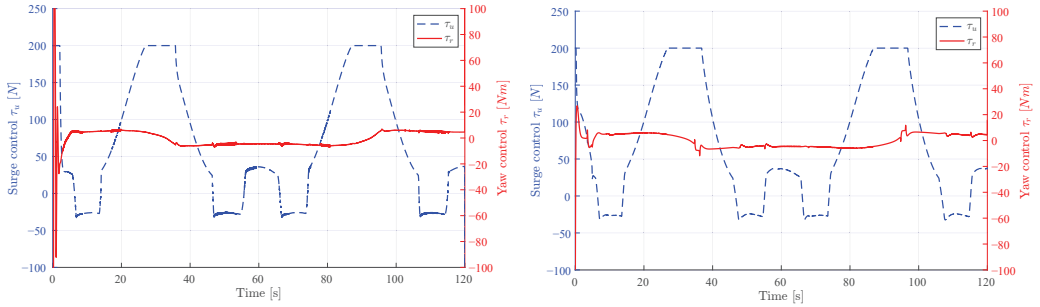


(**a**) (**b**)

Figure 3. (**a**) Trajectory tracking errors in the X (**Top**), and Y (**Bottom**) axes for the SM-DNNI and the ST controllers. (**b**) Sliding surface of the neural network identification errors (**Top**), and sliding surfaces of the neural controller under external disturbances, along the X and Y axes (**Bottom**).

https://doi.org/10.3390/act15050273

Figure 4 presents the control inputs for the surge and yaw dynamics, as defined in Equation (66), under actuator saturation conditions. The results indicate that both inputs remain within acceptable limits for the actuators of the unmanned surface vehicle.



(**a**) (**b**)

Figure 4. Control signals with actuator saturation for the surge and yaw dynamics. (**a**) SM-DNNI controller, (**b**) ST controller.

5.2. Simulations Under Periodic Disturbances

The periodic disturbances used to analyze the robustness of the proposed control strategy are given as follows

$$\begin{aligned}\Delta_u &= a_1 + b_1 \sin\left(\frac{3\pi}{10}t - \frac{\pi}{3}\right) + c_1 \cos\left(\frac{1\pi}{10}t + \frac{\pi}{4}\right)\\ \Delta_v &= a_2 + b_2 \sin\left(\frac{3\pi}{5}t + \frac{\pi}{4}\right) + c_2 \cos\left(\frac{3\pi}{10}t - \frac{\pi}{6}\right)\\ \Delta_r &= a_3 + b_3 \sin\left(\frac{9\pi}{10}t + \frac{\pi}{6}\right) + c_3 \cos\left(\frac{3\pi}{5}t - \frac{\pi}{3}\right)\end{aligned} \tag{76}$$

where amplitudes a_i, b_i, c_i, $i = 1, \ldots, 3$ are defined in Table 5 for two different cases.

Figure 5 illustrates the trajectory tracking performance in the X–Y plane for the reference signal defined in Equation (75), using the SM-DNNI and ST controllers under disturbance conditions. Both controllers achieve satisfactory tracking; however, the SM-DNNI controller demonstrates improved accuracy, as evidenced by its reduced deviation from the desired trajectory compared to the ST controller.

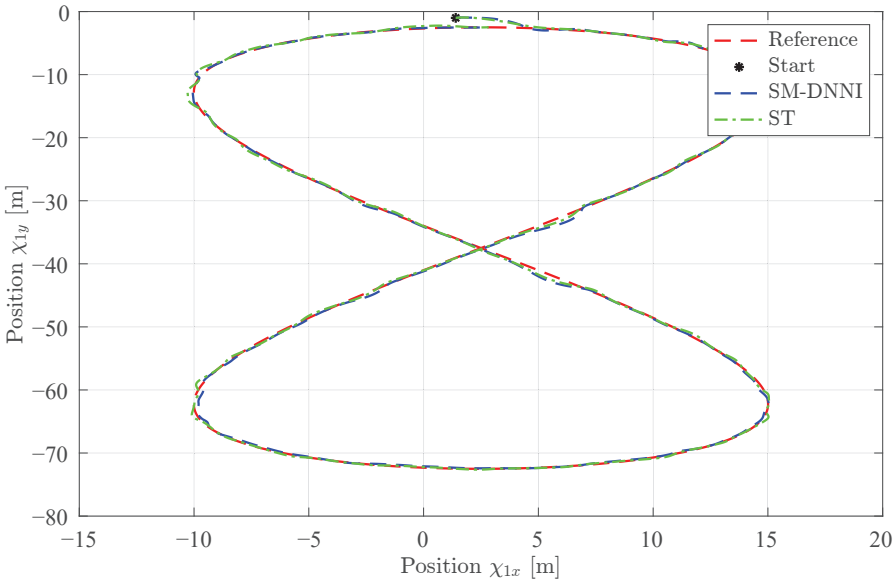
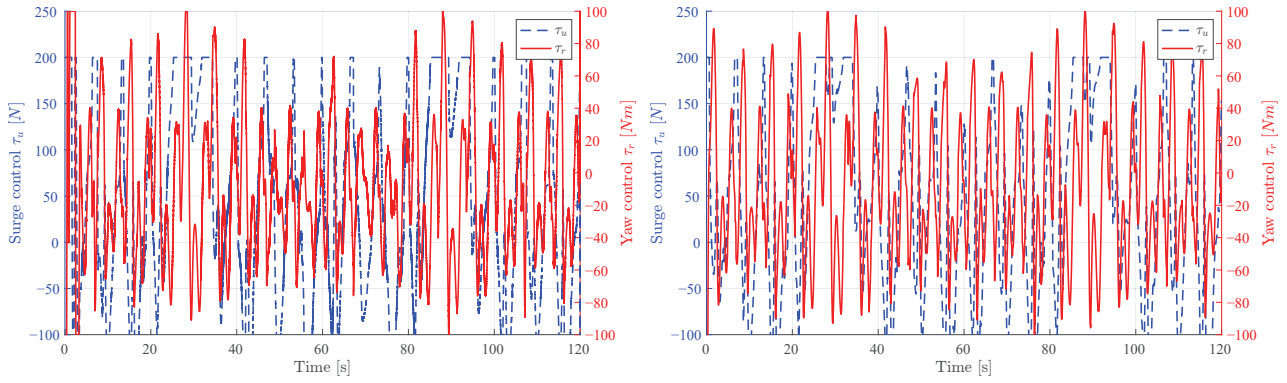


Figure 5. Trajectory tracking in the X-Y plane under periodic disturbances.

https://doi.org/10.3390/act15050273

Figure 6 presents the control inputs for the surge and yaw dynamics, as defined in Equation (66) under external disturbances. The results indicate that both inputs remain within acceptable limits for the actuators of the unmanned surface vehicle. Note that the oscillations observed in the control inputs are primarily caused by the external disturbances introduced in the simulation. Nevertheless, the controller effectively compensates for these disturbances, maintaining satisfactory tracking performance.



(**a**) (**b**)

Figure 6. Control signals under actuator saturation and periodic disturbances for the surge and yaw dynamics. (**a**) SM-DNNI controller, (**b**) ST controller.

Table 5 summarizes the results obtained by applying the ISE, IAE, and ITAE performance indices to the simulation data.

Case 1 corresponds to the scenario without control signal saturation and in the absence of external disturbances. In this scenario, the super-twisting controller outperforms the proposed approach in two of the indices evaluated. However, when saturation is introduced in the control signal, the proposed method demonstrates improved handling, as reflected in Case 2, where all performance indices are lower than those of the super-twisting controller.

When control signal saturation is present, and periodic external disturbances are included in the simulation, the proposed approach achieves slightly better performance than the super-twisting controller, as shown in Cases 3 and 4. Overall, the identifier–controller strategy presented in this manuscript provides superior robustness against control signal saturation and periodic external disturbances in the USV, compared to a state-of-the-art model-based robust control approach.

Table 5. Performance indices for periodic external disturbances. The values in bold represent the smallest tracking error in each case.

Case	Control Signal Saturation	Periodic Disturbances	Controller	ISE	IAE	ITAE
1	NO	No disturbances	SM-DNNI	**6.70**	15.22	590.12
			ST	7.05	**10.08**	**75.29**
2	YES	No disturbances	SM-DNNI	**9.19**	**21.86**	**850.86**
			ST	15.63	36.04	1894.1
3	YES	$a_1 = 0.35, b_1 = 1.75, c_1 = 0.7$ $a_2 = 0.315, b_2 = 1.05, c_2 = 1.75$ $a_3 = 0.297, b_3 = 2.45, c_3 = 2.8$	SM-DNNI	**10.67**	**29.50**	**1295.7**
			ST	11.565	35.13	1621.5
4	YES	$a_1 = 0.7, b_1 = 3.5, c_1 = 1.4$ $a_1 = 0.63, b_1 = 2.1, c_1 = 3.5$ $a_1 = 0.595, b_1 = 4.9, c_1 = 5.6$	SM-DNNI	**22.74**	**54**	**2610.5**
			ST	24.99	69.94	3771.5

https://doi.org/10.3390/act15050273

5.3. Simulations Under Wave-Induced Disturbances

Given that the wave-induced interference forces acting on unmanned surface vehicles depend on both the wave characteristics and the vehicle's motion state, the corresponding disturbance model is expressed as [37,38]

$$\begin{aligned}\Delta_u &= \sum_{i=1}^{N} \rho g B_h L T \cos(\psi_a) s_i(t) \\ \Delta_v &= \sum_{i=1}^{N} -\rho g B_h L T \sin(\psi_a) s_i(t) \\ \Delta_r &= \sum_{i=1}^{N} \frac{1}{24} \rho g B_h L \left(L^2 - B_h^2\right) \sin(2\psi_a) s_i^2(t)\end{aligned} \tag{77}$$

where L, T, and B_h are the length, width, and draught of the USV and are defined in Table 1, ϱ is the seawater density, and ψ_a is called the encounter angle, defined as the angle between the direction of the wave and the course of the USV. The wave energy is denoted by $s_i(t)$, defined as

$$s_i(t) = A_i \frac{2\pi}{\lambda_i} \sin(\omega_{ei} t + \phi_i) \tag{78}$$

where ω_{ei} is the encounter frequency, A_i is the wave amplitude, λ_i is the wavelength, and ϕ_i is the randomly distributed phase on $[0, 2\pi)$.

Figure 7 presents the trajectory tracking in the X–Y plane for the reference signal defined in Equation (75), using the SM-DNNI and ST controllers under wave-induced disturbances. The effects of the external disturbances can be observed in some regions of the X-Y plane, where slight oscillations are present in the upper-left region. Moreover, when the ST controller is used, the vehicle's trajectory exhibits greater deviations from the desired trajectory than those observed with the SM-DNNI approach, particularly in the upper-left and lower-left regions.

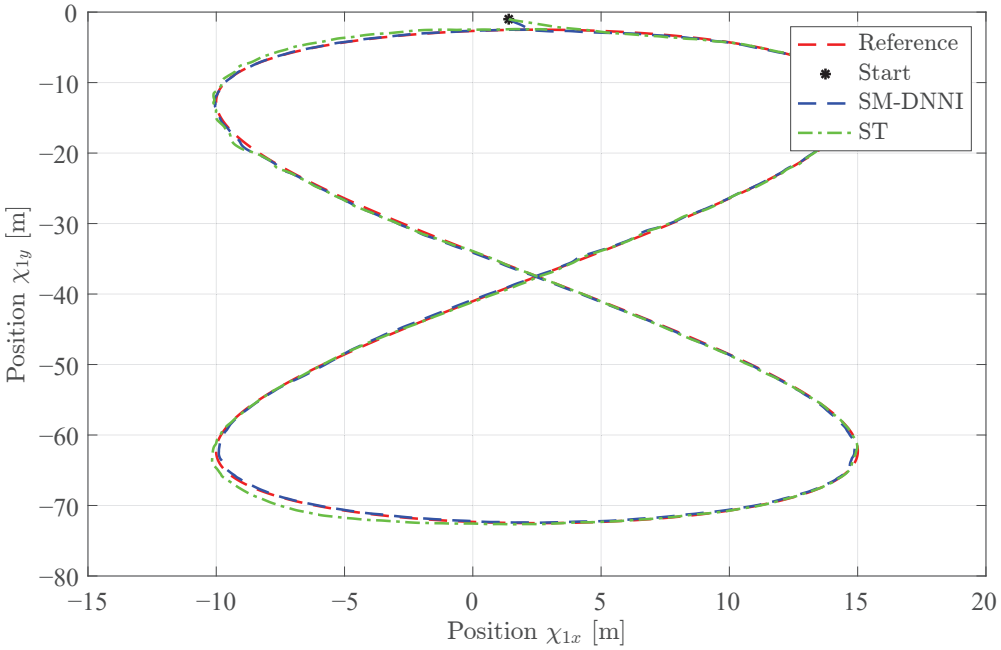


Figure 7. Trajectory tracking in the X-Y plane under wave-induced disturbances.

Figure 8 illustrates the control inputs for the surge and yaw dynamics, as defined in Equation (66), under conditions of control signal saturation and wave-induced disturbances. The results show that both inputs remain within the maximum limits imposed by the USV

https://doi.org/10.3390/act15050273

actuators. Compared with Figure 6, the oscillations observed in the control signals exhibit higher-frequency variations, which are necessary to effectively compensate for the effects of wave-induced disturbances, preserving satisfactory tracking performance.

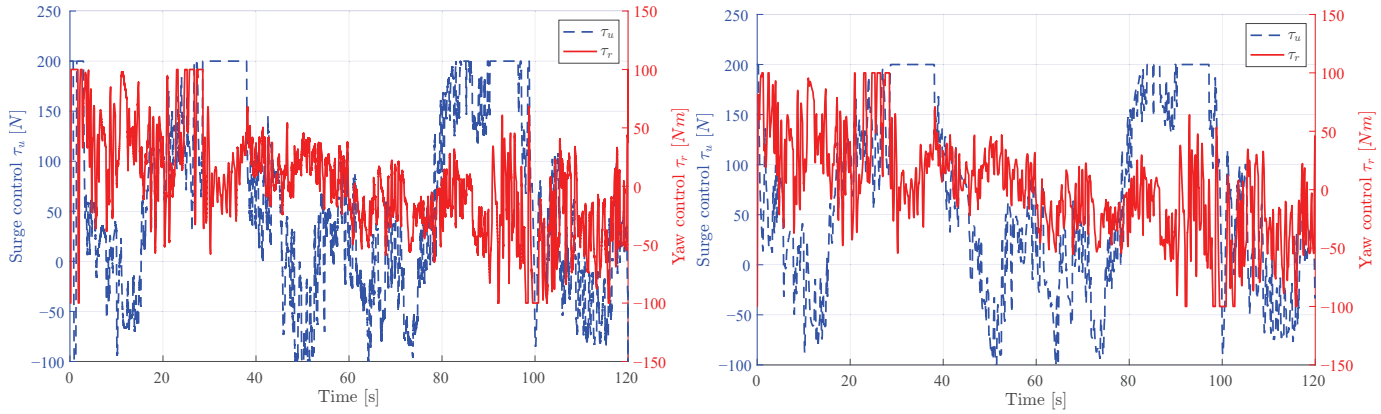


Figure 8. Control signals under actuator saturation and wave-induced disturbances for the surge and yaw dynamics. (**a**) SM-DNNI controller, (**b**) ST controller.

Table 6 presents the evaluation of the ISE, IAE, and ITAE performance indices in the presence of wave-induced disturbances affecting the USV. The simulations consider three wave amplitudes: 0.2 m, 0.3 m, and 0.4 m. For each amplitude, ten simulations were performed, and the corresponding average performance indices are reported for each controller.

Table 6. Performance indices for wave-induced disturbances. The values in bold represent the smallest tracking error in each case.

Wave Amplitude A_i	Controller	ISE	IAE	ITAE
0.2 m	SM-DNNI	**9.64**	**23.86**	**894.01**
	ST	13.43	31.54	1192.22
0.3 m	SM-DNNI	**13.13**	**30.86**	**1243.12**
	ST	17.56	43.05	1914.90
0.4 m	SM-DNNI	**18.46**	**41.64**	**1786.14**
	ST	28.11	62.58	3187.32

Across all wave conditions, the proposed identification–control strategy consistently demonstrates superior performance compared to the super-twisting controller.

5.4. Control Effort Analysis

Tables 7 and 8 present the control effort comparison obtained from the numerical simulations. The control effort in surge and yaw is quantified as the sum of the absolute values of the corresponding control inputs.

Table 7 summarizes the results for both disturbance-free simulations and simulations under periodic disturbances. In the disturbance-free scenario (Case 1), the super-twisting controller yields a higher control effort than the SM-DNNI controller, which is consistent with the reduced tracking error observed when the control input is not subject to saturation.

When control saturation is present (Cases 2–4), the SM-DNNI controller exhibits a higher control effort, in agreement with its improved tracking performance compared to the super-twisting controller. Furthermore, in Cases 3 and 4, the presence of disturbances

https://doi.org/10.3390/act15050273

leads to a significant increase in yaw control effort relative to Cases 1 and 2, due to the additional compensation required to maintain accurate tracking of the reference trajectory.

Table 7. Control effort for periodic external disturbances. The values in bold represent the larges control effort in each case.

Case	Controller	Surge Control τ_u (values $\times 10^6$)	Yaw Control τ_r (values $\times 10^6$)
1	SM-DNNI	1.0160	0.0677
	ST	**1.0278**	**0.0718**
2	SM-DNNI	**1.0164**	**0.0683**
	ST	1.0091	0.0600
3	SM-DNNI	**0.9226**	**0.2830**
	ST	0.8935	0.2871
4	SM-DNNI	**1.1755**	**0.4488**
	ST	1.0871	0.4829

Table 8 summarizes the control effort under wave-induced disturbances. For most wave amplitudes, the SM-DNNI controller shows slightly higher control effort than the ST controller, consistent with its lower tracking error (Table 6). While the surge control effort increases moderately with wave amplitude, the yaw control effort approximately doubles with each increment, indicating the increased compensation needed to mitigate wave effects.

Table 8. Control effort for wave-induced forces. The values in bold represent the larges control effort in each case.

Wave Amplitude A_i	Controller	Surge Control τ_u (values $\times 10^6$)	Yaw Control τ_r (values $\times 10^6$)
0.2 m	SM-DNNI	0.9880	**0.1365**
	ST	**0.9967**	0.1330
0.3 m	SM-DNNI	**1.0038**	**0.2304**
	ST	1.0015	0.2215
0.4 m	SM-DNNI	**1.0643**	**0.4146**
	ST	1.0553	0.4015

As shown in Tables 7 and 8, the SM-DNNI controller exhibits a higher control effort than the super-twisting controller. This additional effort results in improved performance, particularly in terms of reduced tracking error.

6. Conclusions

This study investigated the trajectory tracking problem of USVs in the presence of model uncertainties, unknown parameters, external disturbances, and actuator saturation. A novel dynamic neural network identifier was developed using a nonsingular sliding variable constructed from identification errors. Rigorous stability analysis demonstrated finite-time convergence of both the identification errors and neural network weights, yielding improved stability properties compared to existing dynamic neural network identification approaches. The information generated by the proposed identifier was further integrated into a sliding-mode control framework to achieve nonlinear feedback compensation of

https://doi.org/10.3390/act15050273

unknown USV dynamics and environmental disturbances. The resulting nonsingular sliding-mode controller guarantees finite-time convergence of the tracking errors to zero. Consequently, the closed-loop system that combines the dynamic neural network identifier and the sliding-mode controller effectively addresses the trajectory tracking problem for USVs operating under partially unknown and highly nonlinear conditions.

Based on the results obtained, the effectiveness of the proposed identifier–controller framework has been validated through simulation studies in which actuator saturation constraints and external disturbances representative of realistic operating conditions were explicitly incorporated. The practical applicability of the control strategy has been confirmed, as the control inputs were consistently maintained within the physical limits of the propulsion system. Furthermore, comparative analysis with a state-of-the-art model-based super-twisting controller has demonstrated that the proposed framework achieves comparable tracking performance while eliminating the requirement for prior knowledge of the system's dynamic parameters.

Future work will focus on the experimental implementation of the proposed identifier–controller scheme to further assess its practical performance and robustness under real-world operating conditions.

Author Contributions: Conceptualization, F.M.P., E.S.E., and J.S.C.-R.; methodology, E.S.E.; software, J.P.O.O. and L.R.G.C.; validation, F.M.P. and J.S.C.-R.; formal analysis, F.M.P. and E.S.E.; investigation, F.M.P., E.S.E., and J.S.C.-R.; resources, O.G.-S. and L.R.G.C.; writing—original draft preparation, F.M.P., J.S.C.-R., and E.S.E.; writing—review and editing, F.M.P., E.S.E., J.P.O.O., O.G.-S., and L.R.G.C.; visualization, J.S.C.-R., J.P.O.O., and O.G.-S.; supervision, O.G.-S. and L.R.G.C.; funding acquisition, F.M.P. and L.R.G.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors upon request.

Acknowledgments: This work has been partially supported by the Mexican Secretariat of Science, Humanities, Technology, and Innovation.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

USV	Unmanned Surface Vehicle
RBFNN	Radial Basis Function Neural Networks
HPP	Hand Position Point
ANN	Artificial Neural Network
DNN	Dynamic Neural Network
SM-DNNI	Sliding-Mode–based Dynamic Neural Network Identifier
CoG	Center of Gravity
DoF	Degree of Freedom
ST	Super-Twisting
REV	Reverse Direction
FWD	Forward Direction
ISE	Integral Square Error
IAE	Integral Absolute Error
ITAE	Integral Time-weighted Absolute Error

Appendix A. Neural Network Structure

Appendix A.1. Architecture

The neural network architecture of the proposed SM-DNNI strategy is given in Equation (23):

$$\begin{aligned}\dot{\hat{\chi}}_1 =& \hat{\chi}_2 + \Upsilon\tilde{\chi}_1 \\ \dot{\hat{\chi}}_2 =& R(\psi)\underline{\left[\hat{W}_1^\top \Phi(\chi_1,\chi_2) + \hat{W}_2^\top \Theta(\chi_1,\chi_2)u\right]} \\ &+ \kappa_{\chi 1}(\tilde{\chi}_2 - \Upsilon\tilde{\chi}_1) + \kappa_{\chi 2}\dot{\mu}(\tilde{\chi}_1) + \Lambda_{\chi 1}s_\chi \\ &+ \Lambda_{\chi 2}\mathbf{sig}(s_\chi)^{\sigma_\chi} + \Lambda_{\chi 3}\mathbf{sgn}(s_\chi)\end{aligned} \tag{A1}$$

The underlined expression in Equation (A1) represents the neural network structure within the proposed identifier. A detailed description of the neural network architecture is provided in Figure A1. The network consists of an input layer that receives the state vector components (χ_1, χ_2), a hidden layer composed of sixteen neurons with sigmoidal activation functions, and an output layer formed by a weighted sum of these activation functions through sixteen neural weights. Additionally, the input signal components τ_u and τ_r are included as external inputs, completing the series-parallel identification structure of the SM-DNNI.

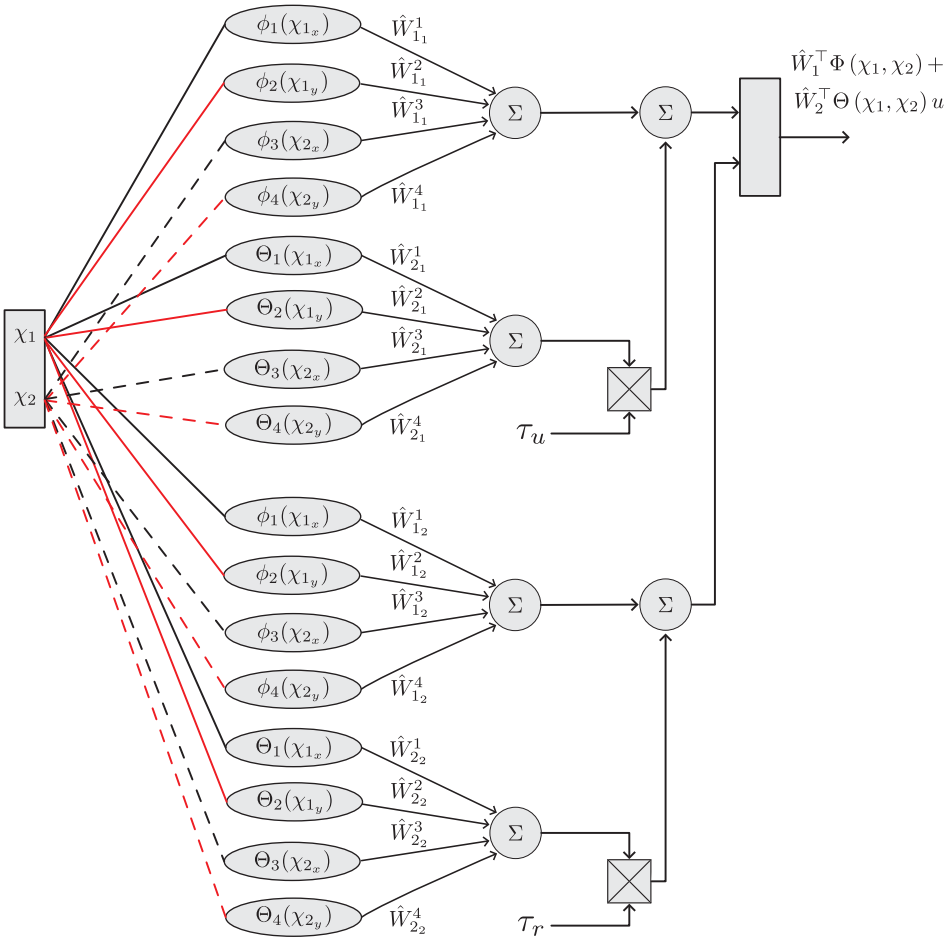


Figure A1. Neural network architecture of the SM-DNNI strategy. The input layer has four inputs: χ_{1_x} (black solid line), χ_{1_y} (red solid line), χ_{2_x} (black dotted line), χ_{2_y} (red dotted line).

Appendix A.2. Training Procedure

The training procedure is conducted in two steps: offline training and online adaptation.

https://doi.org/10.3390/act15050273

- Offline training. The offline training procedure is performed to estimate the initial neural weights $\boldsymbol{W_{01}}$ and $\boldsymbol{W_{01}}$. To obtain these weights, the nominal dynamics are given by

$$\begin{aligned}\dot{\boldsymbol{\chi}}_{\mathbf{1}} &= \boldsymbol{\chi}_{\mathbf{2}} \\ \dot{\boldsymbol{\chi}}_{\mathbf{2}} &= \boldsymbol{R}(\psi)\boldsymbol{W_{01}}^{\top}\boldsymbol{\Phi}(\boldsymbol{\chi}_{\mathbf{1}},\boldsymbol{\chi}_{\mathbf{2}}) + \boldsymbol{R}(\psi)\boldsymbol{W_{02}}^{\top}\boldsymbol{\Theta}(\boldsymbol{\chi}_{\mathbf{1}},\boldsymbol{\chi}_{\mathbf{2}})\boldsymbol{u}\end{aligned}$$

  Note that this equation does not include the term $\tilde{\boldsymbol{f}}_{\boldsymbol{\chi}}$. Using a training dataset containing the state variables $\boldsymbol{\chi}_{\mathbf{1}}$ and $\boldsymbol{\chi}_{\mathbf{2}}$, together with the control input $\boldsymbol{u}$, a least-squares algorithm is used to estimate the initial neural weights.
- Online adaptation. The online adaptation stage is performed during the simulation of the closed-loop dynamics, which include the USV dynamics, the SM-DNNI, and the control law. In this step, the identifier proposed in Equation (23) is used, while the neural weights are updated according to Equations (26) and (27). Through this online adaptation process, the neural weights are adjusted, allowing the SM-DNNI to identify the dynamics of the transformed system represented by Equation (12).

Appendix A.3. Dataset Characteristics

The dataset used for the *offline training* stage is generated using a low-performance controller that does not provide disturbance rejection capabilities. This controller is used exclusively for dataset generation. To ensure that the system evolves over a representative region of the state space, sinusoidal reference signals with different frequencies are defined along the X and Y axes. In particular, the following reference signals are used to generate the training dataset:

$$\begin{aligned}\chi_{1_{dx}} &= 60\sin(0.21t)\cos(0.1t) \\ \chi_{1_{dy}} &= 80\cos(0.1t)\cos(0.15t)\end{aligned} \tag{A2}$$

Appendix A.4. Hyperparameters

- Batch size. In the proposed approach, the offline training stage uses all available data grouped into a regressor matrix to compute the initial neural weights. The system is simulated for 120 s with a sampling period of 0.001 s, resulting in 120,001 samples for each input variable and for the control signal. Consequently, the regression matrix consists of five columns, each containing 120,001 samples. During the online adaptation stage, the neural weights are updated at every sampling instant using the information available at that time.
- Number of neurons. A total of 16 activation functions are employed in the neural network terms included in the SM-DNNI dynamics.
- Number of layers. The neural network consists of an input layer, a hidden layer, and an output layer, following a structure analogous to that of a radial basis function neural network.
- Learning rates. The learning rates used for the neural network weight updates are given by the matrices $\boldsymbol{\Gamma_1}$, $\boldsymbol{\Gamma_2}$, $\boldsymbol{\Sigma_1}$, $\boldsymbol{\Sigma_2}$, $\boldsymbol{\Omega_1}$, and $\boldsymbol{\Omega_2}$, which are used in Equations (26) and (27). Their numerical values are reported in Table 2.

References

1. Xiao, Z.; Zhao, J.; Liu, Z.; Yang, G. Path Planning and Tracking Control for Unmanned Surface Vehicle Based on Adaptive Differential Evolution Algorithm. *Actuators* **2026**, *15, 13* .
2. Li, Z.; Li, M.; Jing, X.; Yuan, C.; Wang, K. Adaptive Sliding Mode Control for Unmanned Surface Vehicle Trajectory Tracking Based on Event-Driven and Control Input Quantization. *Actuators* **2025**, *14*, 457. [CrossRef]

3. Ahmed, F.; Xiang, X.; Jiang, C.; Xiang, G.; Yang, S. Survey on traditional and AI based estimation techniques for hydrodynamic coefficients of autonomous underwater vehicle. *Ocean Eng.* **2023**, *268*, 113300. [CrossRef]
4. Dong, Z.; Yang, X.; Zheng, M.; Song, L.; Mao, Y. Parameter identification of unmanned marine vehicle manoeuvring model based on extended Kalman filter and support vector machine. *Int. J. Adv. Robot. Syst.* **2019**, *16*, 1729881418825095. [CrossRef]
5. Gibson, S.B.; Stilwell, D.J. Hydrodynamic Parameter Estimation for Autonomous Underwater Vehicles. *IEEE J. Ocean. Eng.* **2020**, *45*, 385–394. [CrossRef]
6. Ouyang, Z.L.; Zou, Z.J. Nonparametric modeling of ship maneuvering motion based on Gaussian process regression optimized by genetic algorithm. *Ocean Eng.* **2021**, *238*, 109699. [CrossRef]
7. Ren, Z.; Xia, T. Fixed-Time Output Feedback Distributed Cooperative Event-Triggered Control for Multiple Surface Vessels With Prescribed Performance Constraints. *IEEE Access* **2023**, *11*, 15198–15210. [CrossRef]
8. Sun, M. Underactuated Surface Vessels Trajectory Tracking Control Based on Concise Finite-Time Disturbance Observer Under False Data Injection Attacks. *IEEE Access* **2024**, *12*, 33605–33612. [CrossRef]
9. Neto, A.F.D.S.; Honório, L.D.M.; Da Silva, M.F.; Junior, I.C.D.S.; Westin, L.G.F. Development of optimal parameter estimation methodologies applied to a 3DoF autonomous surface vessel. *IEEE Access* **2021**, *9*, 50035–50049. [CrossRef]
10. Liu, J.; Wang, B. Adaptive Fuzzy Control of USV Based on Improved Finite-Time Command-Filtered Backstepping Method. *IEEE Access* **2025**, *13*, 70313–70323. [CrossRef]
11. Wang, Q.; Jiang, C.; Ning, J.; Hao, L.; Yin, Y. Fuzzy Course Tracking Control of Unmanned Surface Vehicle with Actuator Input Quantization and Event-Triggered Mechanism. *Actuators* **2025**, *14*, 130. [CrossRef]
12. Pomet, J.B.; Thuilot, B.; Bastin, G.; Campion, G. A hybrid strategy for the feedback stabilization of nonholonomic mobile robots. In *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*; IEEE Computer Society: Nice, France, 1992; pp. 129–134.
13. Li, X.; Wen, C.; Chen, C. Adaptive formation control of networked robotic systems with bearing-only measurements. *IEEE Trans. Cybern.* **2020**, *51*, 199–209. [CrossRef]
14. Matouš, J.; Paliotta, C.; Pettersen, K.Y.; Varagnolo, D. The hand position concept for control of underactuated underwater vehicles. *IEEE Trans. Control Syst. Technol.* **2024**, *32*, 2223–2239. [CrossRef]
15. Paliotta, C.; Lefeber, E.; Pettersen, K.Y.; Pinto, J.; Costa, M.; de Figueiredo Borges de Sousa, J.T. Trajectory Tracking and Path Following for Underactuated Marine Vehicles. *IEEE Trans. Control Syst. Technol.* **2019**, *27*, 1423–1437.
16. Alvaro-Mendoza, E.; Gonzalez-Garcia, A.; Castañeda, H.; León-Morales, J.D. Novel adaptive law for super-twisting controller: USV tracking control under disturbances. *ISA Trans.* **2023**, *139*, 561–573. [CrossRef]
17. Gonzalez-Garcia, A.; Castañeda, H.; De León-Morales, J. Unmanned surface vehicle robust tracking control using an adaptive super-twisting controller. *Control Eng. Pract.* **2024**, *149*, 105985. [CrossRef]
18. Matouš, J.; Pettersen, K.Y.; Varagnolo, D.; Paliotta, C.; Ruud, E.L. Adaptive Hand Position for Underactuated Underwater Vehicles. *IEEE J. Ocean. Eng.* **2025**, *50*, 1647–1656. [CrossRef]
19. Yu, W.; Poznyak, A.S. Indirect adaptive control via parallel dynamic neural networks. *IEE Proc.-Control Theory Appl.* **1999**, *146*, 25–30.
20. Poznyak, A.S.; Sanchez, E.N.; Yu, W. *Differential Neural Networks for Robust Nonlinear Control: Identification, State Estimation and Trajectory Tracking*; World Scientific: Singapore, 2001.
21. Cervantes-Rojas, J.S.; Mu noz, F.; Chairez, I.; González-Hernández, I.; Salazar, S. Adaptive tracking control of an unmanned aerial system based on a dynamic neural-fuzzy disturbance estimator. *ISA Trans.* **2020**, *101*, 309–326. [CrossRef]
22. Mu noz, F.; Cervantes-Rojas, J.S.; Valdovinos, J.M.; Sandre-Hernández, O.; Salazar, S.; Romero, H. Dynamic neural network-based adaptive tracking control for an autonomous underwater vehicle subject to modeling and parametric uncertainties. *Appl. Sci.* **2021**, *11*, 2797. [CrossRef]
23. Mu noz, F.; Valdovinos, J.M.; Cervantes-Rojas, J.S.; Cruz, S.S.; Santana, A.M. Leader–follower consensus control for a class of nonlinear multi-agent systems using dynamical neural networks. *Neurocomputing* **2023**, *561*, 126888. [CrossRef]
24. Gomez-Correa, M.; Guarneros-Sandoval, A.; Chairez, I.; Menon, P.P.; Cruz-Ortiz, D.; Ballesteros, M.; Edwards, C. Robotic systems identifier based on exponential barrier Lyapunov functions. *Appl. Math. Model.* **2025**, *150*, 116490. [CrossRef]
25. Guarneros-Sandoval, A.; Ballesteros, M.; Chairez, I. Neuroidentifier for a class of nonlinear systems: A Sliding Modes approach. In *Proceedings of the 2025 11th International Conference on Control, Decision and Information Technologies (CoDIT)*; IEEE: Piscataway, NJ, USA, 2025; Volume 1, pp. 2463–2468.
26. Llorente-Vidrio, D.; Chairez, I.; Fuentes-Aguilar, R.Q. Non-parametric identifier of systems with uncertain model using a Lyapunov-based continuous form of back-propagation method. *Neurocomputing* **2025**, *631*, 129670. [CrossRef]
27. Ballesteros, M.; Fuentes-Aguilar, R.Q.; Chairez, I. Exponential Continuous Non-Parametric Neural Identifier With Predefined Convergence Velocity. *IEEE/CAA J. Autom. Sin.* **2022**, *9*, 1049–1060. [CrossRef]
28. Llorente-Vidrio, D.; Chavez-Galaviz, J.; Fuentes-Aguilar, R.Q.; Chairez, I.; Mahmoudian, N. Robust sliding-mode control of an underwater ROV via neural differential identification of model uncertainties. *Ocean Eng.* **2025**, *341*, 122728. [CrossRef]

29. Zuo, Z. Nonsingular fixed-time consensus tracking for second-order multi-agent networks. *Automatica* **2015**, *54*, 305–309. [CrossRef]
30. Sai, H.; Xu, Z.; He, S.; Zhang, E.; Zhu, L. Adaptive nonsingular fixed-time sliding mode control for uncertain robotic manipulators under actuator saturation. *ISA Trans.* **2022**, *123*, 46–60. [CrossRef]
31. Liu, Y.; Li, H.; Lu, R.; Zuo, Z.; Li, X. An Overview of Finite/Fixed-Time Control and Its Application in Engineering Systems. *IEEE/CAA J. Autom. Sin.* **2022**, *9*, 2106–2120. [CrossRef]
32. Gonzalez-Garcia, A.; Castañeda, H. Guidance and Control Based on Adaptive Sliding Mode Strategy for a USV Subject to Uncertainties. *IEEE J. Ocean. Eng.* **2021**, *46*, 1144–1154. [CrossRef]
33. Fossen, T.I. Marine Control Systems: Guidance, navigation and control of ships. In *Rigs and Underwater Vehicles*; Marine Cybernetics: Trondheim, Norway, 2002.
34. Do, K.D.; Pan, J. *Control of Ships and Underwater Vehicles: Design for Underactuated and Nonlinear Marine Systems*; Springer: London, UK, 2009.
35. Wang, Y.; Zhu, K.; Chen, B.; Jin, M. Model-free continuous nonsingular fast terminal sliding mode control for cable-driven manipulators. *ISA Trans.* **2020**, *98*, 483–495. [CrossRef]
36. BlueRobotics. T500 Thruster. 2026. Available online: https://bluerobotics.com/store/thrusters/t100-t200-thrusters/t500-thruster/ (accessed on 23 March 2026)
37. Mu, D.; Lang, Z.; Fan, Y.; Zhao, Y. Time-varying encounter angle trajectory tracking control of unmanned surface vehicle based on wave modeling. *ISA Trans.* **2023**, *142*, 409–419. [CrossRef] [PubMed]
38. Mu, D.; Li, J.; Wang, G.; Fan, Y. Research on path following control of unmanned ship based on fast wave inversion disturbance compensation and preset performance. *Ocean Eng.* **2024**, *304*, 117864. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

actuators

MDPI

Article

Design and Modeling of a Robot for Rehabilitation of the Sit-to-Stand Movement and Walking

Isela G. Carrera [1], Hector A. Moreno [1,*] and Jose Luis Ordoñez-Avila [2,3]

[1] Facultad de Ingeniería Mecánica y Eléctrica, Universidad Autónoma de Coahuila, Monclova 25750, Coah, Mexico; iselacarrera@uadec.edu.mx
[2] Facultad de Ingeniería, Universidad Evangelica de El Salvador (UEES), San Salvador 1101, El Salvador; jlordonez@unitec.edu
[3] Facultad de Ingeniería, Universidad Tecnológica Centroamericana (UNITEC), San Pedro Sula 21112, Honduras
* Correspondence: h_moreno@uadec.edu.mx

Abstract

Disabilities of the lower extremities significantly affect a person's ability to perform activities of daily living. Many people have been affected by this type of disability due to birth disease or injury from accidents, strokes or even old age. The technical aids used in this type of disability are very basic, and rehabilitation is mainly performed by therapists. Rehabilitation consists of repetitions of exercises with normal movements that must be performed for prolonged periods of time. On the other hand, therapists, having to support the weight of the patient, tend to get injured. This paper introduces the design and modeling a robotic device intended to assist the therapist in the rehabilitation of sit-to-stand (STS) and walking movements, focusing primarily on the technical aspects of the system. The robot is designed to safely support the user's weight and guide the user with appropriate movements according to the usual biomechanics of STS. This paper presents the solution of the inverse kinematic modeling of both the position and velocity of the robot mechanism, as well as the dynamic analysis. A series of simulations is conducted to evaluate the performance of the proposed mechanical architecture during the STS task, providing quantitative information on the system dynamics and the interaction forces between the user and the robot. The mathematical model was employed in the design of a prototype intended for children aged 8–12 years, capable of supporting up to 50 kg and providing a vertical motion range of 20–90 cm. The main structural elements of the robot, its control architecture, and its operation during the execution of the STS task are described. Finally, the conclusions of this work are discussed, and future work derived from this research is outlined.

Keywords: dynamic modeling; lower limbs; rehabilitation; robotics; sit-to-stand; walking

1. Introduction

There are 1.3 billion people with disabilities in the world, according to the World Health Organization [1]. In the case of Mexico, 64.1% of people with disabilities have lower limb disabilities, for which activities such as walking, climbing, and descending are partially or totally limited. The technical aids for people with this type of disability remain very limited and focus on well-known devices such as canes (24.2%), walkers (18.1%), crutches (3%), wheelchairs (37%) and the help of other people (17.7%). In 2014, the main causes of disability among the population of Mexico were diseases (41.3%) and advanced

age (33.1%). The other most reported causes of disability by the population are problems originated during birth (10.7%) and accidents (8.8%). People suffering from lower-limb disabilities can range from children to the elderly. Lower-limb disabilities account for 36.2% of cases in children, and among older adults age 60 and older, the proportion is 81.3% [2]. With these statistics presented by Instituto Nacional de Estadística y Geografía (INEGI), it is conclusive that lower-limb disabilities are the predominant ones, and it is justified to look for technical tools to provide a better quality of life by possible rehabilitation.

Rehabilitation robotics has been demonstrated to be an effective tool for providing therapies that promote motor and functional recovery [3]. This paper introduces a robotic design aimed at assisting patients during sit-to-stand and gait rehabilitation, focusing on the technical aspects of the system and providing a detailed mathematical analysis of its mechanics. One of the main contributions of this work is the formulation of the inverse dynamics model for the STS task, which accounts for the interaction forces between the user and the robot. The procedure for solving the inverse dynamics problem, its implementation, and its numerical validation are presented. In addition, the development of a prototype of the robot is described, covering the main aspects considered during the mechanical design, the definition of the control architecture, and the integration of the system components, with the aim of ensuring safe and effective operation.

An empirical test of the proposed robotic system was conducted with a representative user, an 8-year-old child, to preliminarily assess their functional performance and the overall technical feasibility of the platform. This evaluation was intended solely as a proof of concept and did not constitute a formal clinical trial or a systematic therapeutic assessment. Consequently, no clinical conclusions regarding rehabilitation outcomes are drawn from this preliminary experiment. Comprehensive clinical validation, including controlled trials with a larger patient population and standardized assessment protocols, is planned as a subsequent phase of this research.

In this section, we investigate important aspects in the realm of lower-limb disabilities, like the diseases that lead to such impairments, the biomechanics of the STS movement and the rehabilitation techniques aimed at facilitating the process of standing up from a seated position. In the next section, we conducted a state-of-the-art survey of robotic devices for rehabilitation. The third section presents the concept design for our approach. After that, the kinematics and dynamics of the robot, alongside simulations conducted in accordance with the proposed dimensions and ranges of movement, are presented. Finally, the fabrication of the prototype is introduced, and its performance is discussed.

1.1. Diseases That Cause Motor Disability

Mobility is a fundamental ability for personal development, as it enables autonomy and independence in performing activities of daily living. Human mobility involves the musculoskeletal system, several dissociable neural systems, central nervous system activation, muscle activation, and the integration of sensory information [4]. Impairments in the functioning of these systems may lead to mobility disorders or limitations in individuals [5]. Some of these conditions include:

- Cerebrovascular Accident or Stroke. Stroke is primarily caused by an interruption of blood flow to the brain due to either a blocked artery (ischemic stroke) or the rupture of a blood vessel (hemorrhagic stroke). In older adults, this condition can lead to significant impairments in balance, coordination, muscle spasticity, and proprioception [6,7].
- Spinal Cord Injury. This condition typically results from a sudden, traumatic impact to the spine that causes vertebral fractures. An incomplete spinal cord injury is characterized by the preservation of some motor or sensory function below the level of the injury [8,9].

https://doi.org/10.3390/act15060323

- Cerebral Palsy. Cerebral palsy is the result of abnormal brain development or damage to the developing brain [10]. It occurs before birth, at birth or during the first years of life [11]. There are four types of cerebral palsy: spastic, dyskinetic, ataxic and mixed [10]. This disease causes many problems in maintaining posture, standing upright and walking, but patients can learn to manage the condition with rehabilitation.

Mobility problems caused by these injuries or diseases may or may not be permanent. In cases where the condition is permanent, assistive technologies can be developed to support individuals in performing activities of daily living, as well as to assist caregivers in tasks involving patient mobility and weight support. On the other hand, for people who can still be rehabilitated, the development should focus on techniques or devices that can help in rehabilitation therapy, such as appropriate and repetitive movement, as well as exercise the neuronal brain activity to detect intentions of movement and motivation [12].

1.2. Biomechanics of the STS Movement

One of the most overlooked aspects of lower-limb mobility is the ability to transition from sitting to standing and subsequently initiate walking, as reviewed in [13]. Individuals affected by mobility impairments (ranging from children to older adults) often require rehabilitation to perform the sit-to-stand movement correctly, as demonstrated in [14]. Important aspects of the biomechanics of the STS movement are summarized next:

- The standing up movement is divided into four phases by several authors: flexion momentum, transfer momentum, extension, and stabilization. Figure 1 shows a simulation made with OpenSim [15].
- Initially, the major mass of the body is resting on the buttocks in a stable sitting posture.
- During the flexion momentum phase, the upper body (head and trunk) moves forward, and the hip and forefoot joints rotate in the same manner around the ankle joint (dorsiflexion).
- Once the head and trunk rotate forward, causing the mass of the hip to move horizontally (this part is also called the momentum transfer phase), the extension phase begins with the knees extending, closely followed by extension of the hips and ankles.
- The buttocks are lifted off the seat during the extension phase; the greatest muscular force occurs when lifting the mass of the body off the seat. During the rest of the extension phase, the hips, knees and ankles continue their extension to bring the body to a stable upright posture, which is also called the stabilization phase.

References [16,17] provide data on the trajectories of a person's joints during the standing-up movement. From the description of the STS movement presented above, it can be seen that the design of a robot to rehabilitate this movement requires actuators to control hip position and torso orientation.

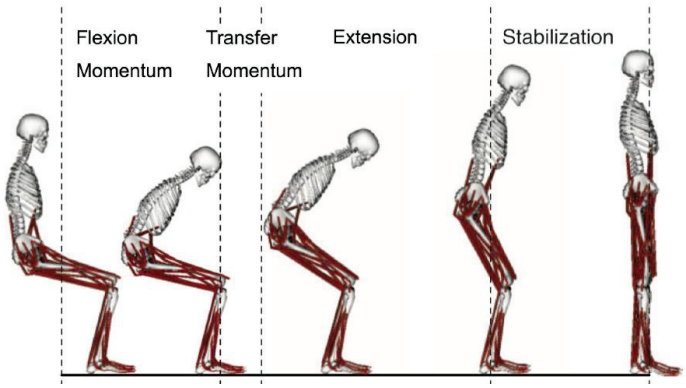


Figure 1. Phases of the STS biomechanical movement.

1.3. Solutions to Assist and Aid in the Rehabilitation of Lower-Limb Disabilities

Assistive devices for people with lower-limb problems that have been frequently observed for decades are canes, crutches, walkers, and wheelchairs. Other devices encountered have a specific design, such as powered armchairs or cushions for standing up, devices for getting up from the toilet, standing frames, walkers with a movable seat for sitting, and wheelchairs that can change to an extended version to keep a person standing. These devices make it possible to move from sitting to standing and vice versa.

On the other hand, for rehabilitation, physiotherapists have traditionally applied physical techniques to restore movement and function in people affected by disability. In the case of the STS movement, the therapist provides guidance to the patient in order to facilitate the momentum transfer. The therapist applies a pulling force on the patient's arms. This approach allows the therapist to assist with a portion of the patient's weight, thereby reducing the amount of force required to maintain extension of the upper trunk on the hips. Therapists can use simple mechanical devices (for example, a table equipped with wheels) during sessions to help with the rehabilitation movements [18]. In the case of gait rehabilitation, therapy consist in exercises applied to of all lower-limb components, including the toes, foot, leg, thigh, and hip. For example, in cases of stroke, common toe impairments include claw toe and hammer toe [19]. Exercises for treating claw toe include assisted toe extension and flexion, typically guided by a physiotherapist. Basic weight-shifting exercises are also recommended to improve balance and strengthen the lower limbs. Technology based interventions for stroke patients consist of body-weight–supported treadmill training [20] and functional electrical stimulation [21]. For sarcopenia, the recommended therapies include progressive resistance exercises, high-intensity resistance training, aerobic workouts, and plyometric exercises [22].

It can be observed that several movements, such as those provided by assistive devices, and the movements performed during rehabilitation therapies, are similar, which shows an opportunity in the development of devices that can be used to provide both assistance and rehabilitation, or both activities simultaneously.

2. Robotic Technologies for the Assistance and Rehabilitation of Lower-Limb Disabilities

Within the field of healthcare robotics, two main categories of robotic systems can be identified: assistive robots and rehabilitation robots. Assistive robots support users in daily activities, enhancing autonomy, safety, and quality of life, while rehabilitation robots aim to recover or improve motor and cognitive functions through controlled therapeutic exercises. This section first presents a state-of-the-art review of assistive robots, followed by a discussion of rehabilitation robots.

Assistive robots are designed to support individuals with mobility impairments by enhancing physical function and promoting independence in daily activities. These systems integrate advanced sensors, actuators, and control algorithms to adapt to user needs and environmental contexts. Assistive robots are usually mobile platforms. Different walker designs supporting the STS task have been described in the literature [23–26]. The wheels of the mobile platforms can be actuated [23] or passive [24], and therefore, these devices can provide active or passive assistance in walking. The device presented in [23], which can be used for assistance or rehabilitation, features a support pad mounted on a hybrid mechanism with three degrees of freedom, combining serial and parallel kinematic chains. The lifting mechanism is actuated by servomotors coupled with worm gears, allowing the device to maintain its position even in the event of a power failure. The mobile platform is driven by two motors, each attached to a front wheel. During the STS task, the user is supported by their chest against the pad. Other similar devices are presented in [25,26].

https://doi.org/10.3390/act15060323

Another walker design with STS assistance is presented in [24]. This design features four caster wheels, allowing the user to walk once in an upright position. A harness is used to assist in lifting the user, and the standing motion is facilitated by a four-bar mechanism powered by a pneumatic actuator. In [27], the design of an assistive device for elderly people is described. According to the authors, the device is designed to replicate the motion of a caregiver during support tasks. The user is supported at the arms and back. The active walker system consists of a differential-drive mobile platform equipped with two actuated wheels and two passive caster wheels.

Complete assistive robots are designed for individuals with disabilities who have little or no potential for recovery. These devices typically enable users to change postures and carry out daily activities with their hands free, such as washing, cooking, or retrieving items from countertops and cabinets. These devices often include a harness system for lifting the user and are equipped with wheels for mobility. Among them are the Tek Robotic Mobilization Device (TEK RMD) and Quality of Life Through Locomotion (QOLO). The TEK RMD is a commercially available four-wheel mobile robot designed to assist disabled individuals with standing, sitting, and mobility in the upright position [28] Users control the robot using a joystick. QOLO [29] is a mobile device; in this case, it consists of a deployable chair, and the user is attached by their lower limbs. Indeed, because the device lacks a frontal component, the user is able to carry out tasks using their hands, without obstruction.

One of the most innovative approaches in rehabilitation is the use of robotic systems that assist users or actively perform therapeutic movements in coordination with them. Robots have been proposed for rehabilitation treatments because they fulfill key requirements, such as enabling the repetition of specific exercises and providing at least partial weight support for the patient. Robots can also be reprogrammed, depending on each person, giving support to patients and therapists and saving and recording the patient's progress in some interface. Some prototypes and commercial products have been developed to support the STS task, both for rehabilitation and assistive purposes. Rehabilitation devices aim to replicate the natural STS movement observed in healthy individuals by applying controlled forces to specific parts of the patient's body, enabling repetitive practice of the motion.

In [30], the STS trainer is introduced. This device is designed to provide adjustable levels of mechanical support and variable STS transfer speeds. It consists of a deployable chair to which the user is secured at the seat and backrest. This chair is a one-degree-of-freedom mechanism. The concept of STS-Care device, described in [31], is designed for the study and rehabilitation of STS movements in individuals recovering from stroke. The STS-Care device consists of a three-degree-of-freedom parallel kinematic mechanism, composed of six limbs that connect a fixed base to a mobile platform. The patient is supported on this mobile platform. The mechanism is actuated by a set of linear actuators housed in the base, which are powered by motors driving pulley and belt systems.

There is a design in [32] for a rehabilitation robot for the sit-to-stand (STS) transition based on a dual-rope system (front rope and rear rope). It features innovative impedance control combined with interaction through an interactive game. It recognizes the user's intended movement through force sensors (load cells). If the patient has strength, the rope rises quickly; if they struggle, it moves slowly to assist them; if the patient falls or collapses, the robot brakes or descends at a safe speed.

There are some previous proposals of robotic systems designed to assist and rehabilitate STS and walking. In [33] the concept design of a mobile structure with an arm of two degrees of freedom for the rehabilitation of lower limbs is presented. The robot is equipped with a wheeled main support frame that enables movement synchronized

https://doi.org/10.3390/act15060323

with the patient's walking. Reference [13] presents the concept of a device designed to assist users in STS transitions, sitting down, and performing locomotion-related tasks. The robot comprises three links connected via distinct joints: a ceiling-mounted rail traction system, a vertical-axis rotational joint, and an actuated prismatic joint responsible for the vertical displacement of the arm support. Reference [34] presents the use of the Ballbot to assist individuals in standing up. The Ballbot is a mobile robotic platform that maintains dynamic balance on a single spherical wheel. In the reported experiments, user interaction was achieved by applying pulling forces to the user's hands.

Another more recent design [35] is a lower-limb rehabilitation robot, which is innovative in that it focuses on providing real assistance tailored to the needs of hemiplegic patients. The mechanism can be configured or coupled to perform continuous passive motion in bed, preventing muscle atrophy and joint stiffness when the patient lacks voluntary motor control. It also assists with strength training and sit-to-stand transition trajectories. Furthermore, the robot guides the limbs during bipedal standing using a body-weight support system over a treadmill or a structured walkway. The mechanical structure of the lower-limb features active motorized joints at the hip and knee within the sagittal plane to replicate flexion and extension movements. It includes a passive or semi-active mechanism (utilizing springs or mechanical limits) to correct drop foot. Finally, the frame is equipped with sliding links and quick-fastening systems to adapt the thigh and shank lengths to the height percentile of different patients.

Finally, a special category of devices for lower-limb rehabilitation is exoskeletons. Exoskeletons are robotic devices consisting of serial kinematic chains whose joints are actuated by servomotors, providing the necessary torque to move each of the segments of the lower limbs. They are classified mainly according to whether the structure is fixed [36] or mobile [37] (allowing the user to walk). Although exoskeletons provide superior mobility compared to other rehabilitation mechanisms, they require the use of multiple actuators, and, in the case of mobile models, balance control is a significant challenge.

In a recent study [38], the mechanical redesign of a lower-limb exoskeleton assembly was presented. A concave and convex roller mechanism was introduced at the knee joint to replicate the actual rolling and sliding behavior of the human joint. Simulations demonstrated that this redesign achieves trajectory tracking within an approximate 5% error margin compared to reference data derived from motion capture (mocap) and clinical gait analysis.

3. Papalotl: A Robot for Assisting in the Rehabilitation of the Sit-to-Stand Movement and Walking

Figure 2 shows the concept of the Papalotl robot. The robotic system comprises a mobile platform and a backrest equipped with a body-weight support system (BWS). The mobile platform includes a rear structural frame and a pair of front limbs, which can be flexed or extended relative to the rear structure via two linear actuators. Flexion of the front limbs elevates the BWS. The patient is secured to the BWS using a harness, ensuring stable support during movement. A separate linear actuator adjusts the orientation of the BWS relative to the rear structure. Horizontal translation of the mobile platform is achieved through actuated front wheels. Coordinated control of the limb actuators, BWS orientation, and wheel motion enables the execution of STS transitions. The mechanism helps the user follow the correct standing up movement in the sagittal plane and rehabilitate the transfer momentum phase. Once the user is in the standing position, the robot can be used to support the user's weight and perform walking rehabilitation exercises.

https://doi.org/10.3390/act15060323

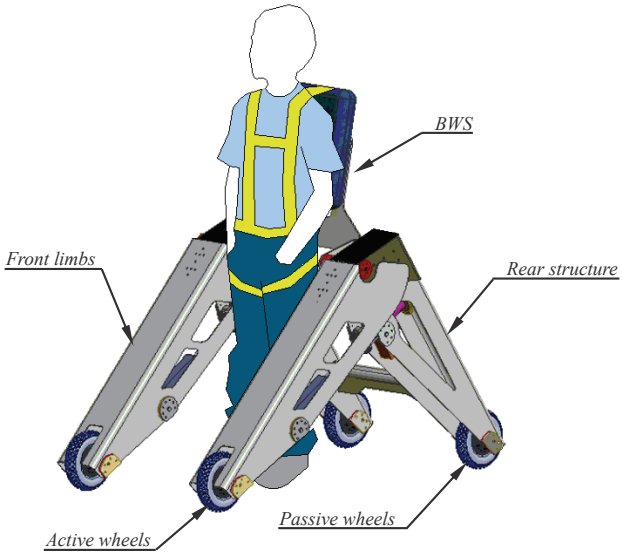


Figure 2. Concept of the Papalotl robot.

The differential drive system, powered by the two active wheels, enables the robot to follow paths with diverse geometries or execute turns during walking therapy, providing greater flexibility in the types of exercises that can be performed. To achieve efficient control of the differential drive system, it is necessary to adapt omnidirectional wheels to the rear structure.

The concept of the device proposed in this work is characterized and distinguished from the previously cited solutions as follows:

1. This device integrates two functionalities that are typically provided separately in other systems: assistance in the STS task, as offered by the devices described in [30,31,39], and assistance during gait, as provided by the systems presented in [40].
2. The robot is capable of moving in different directions and performing both translational and rotational motions in the plane, unlike other concepts in which the motion of the robot is constrained to predetermined trajectories [13].
3. The structure of the robot inherently provides statically stable equilibrium, as it has four points of support. In contrast, exoskeletons require robust control strategies to maintain balance dynamically [37].
4. The points of support of Papalotl are distributed around the user's position, which provides safety in supporting and balancing the patient during therapy. In other works, the device is positioned in front of the user [26], and the support polygon does not include the location of the user's feet. This configuration may lead to stability issues when the robot is required to fully support the user.
5. The control of the hip height and the orientation of the BWS is achieved using lead screws, which prevents the robot from falling in the event of a power loss, since lead screws are self-locking, unlike robots that use gear trains [36].
6. The user is supported by the torso, rather than by the arms as in [25,26,34]. In such devices, the execution of the therapy may depend, to some extent, on the strength of the patient's arms. On the other hand, performing rehabilitation exercises while leaning on the arms may generate a reflex to use them even when assistance is not available.
7. The mechanism requires a small number of actuators, five servomotors, which could be reduced to four by means of a power transmission mechanism for the extension of the front limbs with respect to the rear structure.

https://doi.org/10.3390/act15060323

The proposed robot is designed to provide body weight support, stabilize the user's trunk, and assist postural control during gait tasks. Consequently, the system is intended for patients who are able to initiate and maintain gait patterns but still require physical assistance. According to the Functional Ambulation Categories (FAC) scale [41], the target population corresponds to individuals classified as FAC level 2.

In the context of gait tasks, a direct comparison between the proposed robot and fully actuated lower-limb exoskeletons is not appropriate. Exoskeletons are generally more suitable for individuals with low ambulation capacity (FAC levels 0–1). Both systems present technical and biomechanical advantages and limitations and are designed to address different clinical needs and functional levels. Therefore, these technologies should be regarded as complementary tools within the rehabilitation spectrum, each optimized for different stages of functional recovery.

3.1. Robot Analysis

This section presents the inverse kinematics and dynamics of the robot; after that, a series of simulations is performed to assess the performance of the mechanism during the STS task.

3.1.1. Position Analysis

In the position analysis, the position of the user's hip, **h**, and the orientation of the torso, θ_m, are considered task coordinates; the extension of the linear actuators and the movement of the active wheels are considered the joint variables. Figure 3 shows a kinematic scheme of the robot with the notation of the geometric parameters and kinematic variables.

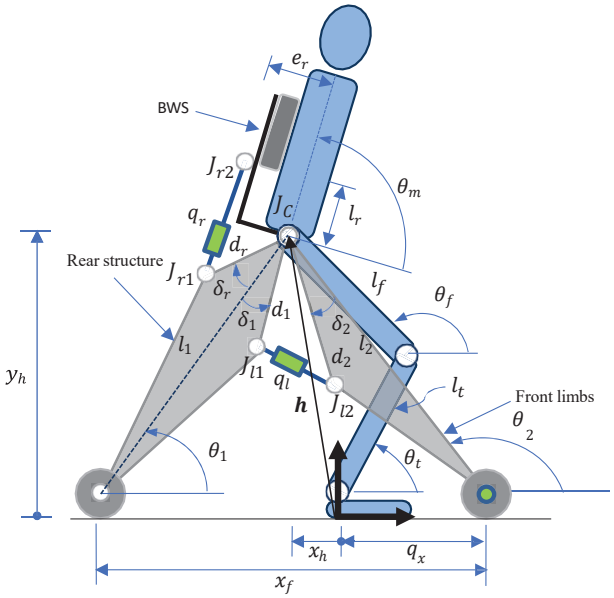


Figure 3. Kinematic scheme of the Papalotl robot.

The position of the hip is given by the following vector:

$$\mathbf{h} = \begin{bmatrix} x_h \\ y_h \end{bmatrix} = \begin{bmatrix} l_t \cos\theta_t + l_f \cos\theta_f \\ l_t \sin\theta_t + l_f \sin\theta_f + h_a \end{bmatrix} \tag{1}$$

where l_t and θ_t are the length and orientation of the tibia, and l_f and θ_f are the length and orientation of the femur, respectively, while h_a is the height of the person's ankle joint with respect to the ground.

https://doi.org/10.3390/act15060323

In this analysis, it is assumed that, through adjustments in the BWS, the hip joint of the user is aligned with the central joint of the robot, J_C, at which the main links of the robot are connected. Given the position of the hip of the user, specifically the height denoted by the variable y_h, the angle θ_1 (orientation of the rear structure) is obtained by the following expression:

$$\theta_1 = \arcsin\left(\frac{y_h - R_w}{l_1}\right) \tag{2}$$

where l_1 is the length of the rear structure (measured from joint J_C to the axis of the rear wheels) and R_w is the radius of the wheels. From the vector loop presented in Figure 4a, the following vector sum is obtained:

$$\mathbf{l}_1 - \mathbf{l}_2 - \mathbf{x}_f = \mathbf{0} \tag{3}$$

where $\mathbf{l}_1 = \langle l_1, \theta_1 \rangle$, $\mathbf{l}_2 = \langle l_2, \theta_2 \rangle$ and $\mathbf{x}_f = \langle x_f, 0 \rangle$. Parameter l_2 is the length of the front limbs, measured from joint J_C to the axis of the front wheels. In the previous equation, the unknowns are the distance between the rear and front wheels x_f and the orientation of the front limbs given by the angle θ_2. Equation (3) can be expressed in terms of the following two equations:

$$l_2 \cos\theta_2 = l_1 \cos\theta_1 - x_f \tag{4}$$

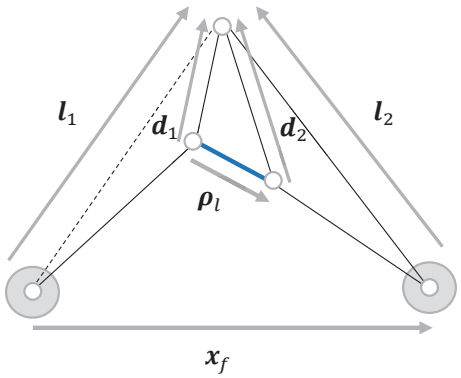
$$l_2 \sin\theta_2 = l_1 \sin\theta_1 \tag{5}$$

By squaring both equations and adding the resulting expressions, θ_2 is eliminated, yielding a quadratic equation in terms of x_f:

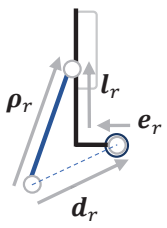
$$l_2^2 = l_1^2 - 2x_f l_1 \cos\theta_1 + x_f^2 \tag{6}$$

and its solution is:

$$x_f = l_1 \cos\theta_1 \pm \sqrt{l_1^2(\cos^2\theta_1 - 1) + l_2^2} \tag{7}$$



(a)



(b)

Figure 4. Closed vector loops of (**a**) the moving platform and (**b**) the BWS considered for the analysis.

https://doi.org/10.3390/act15060323

On the other hand, the angle θ_2 is determined by substituting (7) into (4) and performing the corresponding algebraic manipulation:

$$\theta_2 = \arccos\left(\frac{\pm\sqrt{l_1^2(\cos^2\theta_1 - 1) + l_2^2}}{l_2}\right) \tag{8}$$

From Figure 3, it can also be observed that:

$$x_f = l_1\cos\theta_1 - x_h + q_x \tag{9}$$

By substituting x_f from (7) into (9), the position q_x of the front wheels, which is controlled by the servomotors, can be determined:

$$q_x = x_h \pm \sqrt{l_1^2(\cos^2\theta_1 - 1) + l_2^2} \tag{10}$$

To determine the length of the linear actuators of the limbs, from Figure 4a, the following loop closure equation can be written:

$$\boldsymbol{\rho}_l = \mathbf{d}_1 - \mathbf{d}_2 \tag{11}$$

where $\boldsymbol{\rho}_l = \langle q_l, \theta_l\rangle$, $\mathbf{d}_1 = \langle d_1, \theta_1 + \delta_1\rangle$ and $\mathbf{d}_2 = \langle d_2, \theta_2 - \delta_2\rangle$. Parameters l_1 and l_2 are the distances from the central joint, J_C, to the rotational joints, J_{l1} and J_{l2}, at the ends of the linear actuator, respectively. Angles δ_1 and δ_2 are used to complete the definition of the position of the rotational joints, as shown in Figure 3. From these equations, the length of the linear actuator, q_l, and its orientation, θ_l, can be obtained. The solutions of these variables are:

$$q_l = \pm\sqrt{X_l^2 + Y_l^2} \tag{12}$$

and

$$\theta_l = \arctan(Y_l/X_l) \tag{13}$$

where $X_l = d_1\cos(\theta_1 + \delta_1) - d_2\cos(\theta_2 - \delta_2)$ and $Y_l = d_1\sin(\theta_1 + \delta_1) - d_2\sin(\theta_2 - \delta_2)$.

The extension of the linear actuator that controls the orientation of the BWS can be obtained by analyzing the following loop closure equation (see Figure 4b):

$$\boldsymbol{\rho}_r = \mathbf{d}_r + \mathbf{e}_r + \mathbf{l}_r \tag{14}$$

where the vectors of the previous expression are $\boldsymbol{\rho}_r = \langle q_r, \theta_r\rangle$, $\mathbf{d}_r = \langle d_r, \theta_1 - \delta_r\rangle$, $\mathbf{e}_r = \langle e_r, \theta_m + \pi/2\rangle$ and $\mathbf{l}_r = \langle l_r, \theta_m\rangle$. Parameter d_r denotes the distance between the central joint J_C and the rotational joint J_{r1}. Angle δ_r determines the position of J_{r1} with respect to the central line of the rear structure. Lengths e_r and l_r specify the position of the rotational joint J_{r2} relative to the central joint J_C. As the previous equations show, the length q_r, and orientation θ_r, of the linear actuator are given by:

$$q_r = \pm\sqrt{X_r^2 + Y_r^2} \tag{15}$$

and

$$\theta_r = \arctan(Y_r/X_r) \tag{16}$$

where $X_r = l_r\cos\theta_m + e_r\cos(\theta_m + \pi/2) + d_r\cos(\theta_1 - \delta_r)$ and $Y_r = l_r\sin\theta_m + e_r\sin(\theta_m + \pi/2) + d_r\sin(\theta_1 - \delta_r)$.

https://doi.org/10.3390/act15060323

3.1.2. Velocity Analysis

In this section, the kinematic analysis of velocity is presented. Given the hip linear velocity $\dot{\mathbf{h}}$ and the BWS angular velocity ω_m, the velocities of the joint variables are determined. The angular velocity of the rear structure is given by the following expression:

$$\omega_1 = \frac{\dot{y}_h}{l_1 \cos\theta_1} \tag{17}$$

The angular velocity of the front limbs is obtained, taking the derivative of (3) with respect to time:

$$\omega_2 = \frac{l_1 \cos\theta_1}{l_2 \cos\theta_2}\omega_1 \tag{18}$$

On the other hand, the linear velocity $\dot{x}_f$ can be computed with:

$$\dot{x}_f = (l_2 \cos\theta_1 \tan\theta_2 - l_1 \sin\theta_1)\omega_1 \tag{19}$$

The velocity of the joint variable q_x is given by the following expression:

$$\dot{q}_x = \dot{x}_h - l_1 \sin\theta_1 \omega_1 + \dot{x}_f \tag{20}$$

The next part of the analysis is solved by taking the derivative of (11). The linear velocity of the front limbs' linear actuators is given by:

$$\dot{q}_l = d_1 \sin(\theta_l - \theta_1 - \delta_1)\omega_1 - d_2 \sin(\theta_l - \theta_2 - \delta_2)\omega_2 \tag{21}$$

Following the same procedure above, the derivative of the vector sum in (14) consists of a pair of equations whose solution provides the linear velocity of the BWS linear actuator:

$$\dot{q}_r = d_r \sin(\theta_r - \theta_1 + \delta_r)\omega_1 + [l_r \sin(\theta_r - \theta_m) + e_r \sin(\theta_r - \theta_m - \pi/2)]\omega_m \tag{22}$$

where ω_m is the time derivative of θ_m.

Finally, the relationships between the kinematic variables of the system by means of Jacobian matrices can be obtained. The vector of joint velocities $\dot{\mathbf{q}} = \begin{bmatrix} \dot{q}_x & \dot{q}_l & \dot{q}_r \end{bmatrix}^T$, and the vector of operational velocities $\dot{\mathbf{x}} = \begin{bmatrix} \dot{x}_h & \dot{y}_h & \omega_m \end{bmatrix}^T$ are related by the following expression:

$$\dot{\mathbf{q}} = \mathbf{K}\dot{\mathbf{x}} \tag{23}$$

where the Jacobian matrix $\mathbf{K}$ has the following elements: $k_{11} = 1$, $k_{12} = \tan\theta_2$, $k_{13} = 0$, $k_{21} = 0$, $k_{22} = d_1 \sin(\theta_l - \theta_1 - \delta_1)/l_1 \cos\theta_1 - d_2 \sin(\theta_l - \theta_2 - \delta_2)/l_2 \cos\theta_2$, $k_{23} = 0$, $k_{31} = 0$, $k_{32} = d_r \sin(\theta_r - \theta_1 + \delta_r)/l_1 \cos\theta_1$ and $k_{33} = l_r \sin(\theta_r - \theta_m) + e_r \sin(\theta_r - \theta_m - \pi/2)$.

3.1.3. Dynamics of the Robot

In this section, the inverse dynamics problem associated with the STS task is formulated. The objective is to determine the forces required at the robot actuators, f_l, f_r, and τ_w, given the trajectories of the user's hip position, $\mathbf{h}$, and torso angle, θ_m, together with their corresponding time derivatives, during the execution of the STS movement.

Figure 5 shows the free-body diagrams of the robot considered for this analysis. For the rear structure, we have the following pair of equations of motion:

$$m_s \mathbf{a}_s = m_s \mathbf{g} + \mathbf{f}_{ws} + 2\hat{\boldsymbol{\rho}}_l f_l - 2\mathbf{f}_{sl} - \mathbf{f}_{sb} + \hat{\boldsymbol{\rho}}_r f_r \tag{24}$$

https://doi.org/10.3390/act15060323

and

$$I_s\alpha_s = \mathbf{r}_a \times \mathbf{f}_{ws} + 2\mathbf{r}_b \times \hat{\boldsymbol{\rho}}_l f_l - \mathbf{r}_c \times (2\mathbf{f}_{sl} + \mathbf{f}_{sb}) + \mathbf{r}_d \times \hat{\boldsymbol{\rho}}_r f_r \tag{25}$$

where m_s is the mass of the structure, $\mathbf{a}_s$ is the acceleration of the center of mass, $\mathbf{g}$ is the acceleration of gravity, $\hat{\boldsymbol{\rho}}_l$ and $\hat{\boldsymbol{\rho}}_r$ are unit vectors of $\boldsymbol{\rho}_l$ and $\boldsymbol{\rho}_r$, respectively (see Figure 4), f_l is the force of the linear actuators between the rear structure and the front limbs, f_r is the force of the linear actuator between the rear structure and the BWS, $\mathbf{f}_{sl}$ is the force that the rear structure apply to the front limbs at the point C (F in the front limbs) where both elements are connected by joint J_C, and in the same way, $\mathbf{f}_{sb}$ is the force that the rear structure applies to the BWS at point C (H in the BWS), while $\mathbf{f}_{ws}$ is the force that the rear wheels apply to the structure. On the other hand, I_s is the moment of inertia of the structure in the z-axis, and α_s is the angular acceleration of the structure. Vectors $\mathbf{r}_a$, $\mathbf{r}_b$, $\mathbf{r}_c$ and $\mathbf{r}_d$ correspond to the positions where forces $\mathbf{f}_{ws}$, f_l, $\mathbf{f}_{sl}$, $\mathbf{f}_{sb}$ and f_r are applied to the structure, respectively, with respect to the center of mass of the structure. Given that $\mathbf{f}_{ws}$ only has a vertical component, we have $\mathbf{r}_a \times \mathbf{f}_{ws} = r_{ax} f_{wsy}$.

For the front limbs, the following dynamic equations can be stated:

$$m_l\mathbf{a}_l = m_l\mathbf{g} + \mathbf{f}_{wl} - \hat{\boldsymbol{\rho}}_l f_l + \mathbf{f}_{sl} \tag{26}$$

and

$$I_l\alpha_l = \mathbf{r}_e \times \mathbf{f}_{wl} + \mathbf{r}_f \times \mathbf{f}_{sl} - \mathbf{r}_g \times \hat{\boldsymbol{\rho}}_l f_l - \tau_w \tag{27}$$

where m_l is the mass of the limbs, and $\mathbf{a}_l$ is the acceleration of the center of mass. Vector $\mathbf{f}_{wl}$ is the force that the wheels apply to the limbs, and τ_w is the torque that the rotational motor of the wheels generate, and it is equal to $\tau_w = -f_{wlx}R_w$. On the other hand, I_l is the moment of inertia of the limbs in the z axis, and α_l is the angular acceleration of the limbs. Vectors $\mathbf{r}_e$, $\mathbf{r}_f$ and $\mathbf{r}_g$ correspond to the positions where forces $\mathbf{f}_{wl}$, $\mathbf{f}_{sl}$ and f_l are applied to the limbs, respectively, with respect to the center of mass of the limbs.

For the BWS, we have the following dynamic equations:

$$m_b\mathbf{a}_b = m_b\mathbf{g} + \mathbf{f}_{sb} - \hat{\boldsymbol{\rho}}_r f_r - \mathbf{f}_U \tag{28}$$

and

$$I_b\alpha_b = \mathbf{r}_h \times (\mathbf{f}_{sb} - \mathbf{f}_U) - \mathbf{r}_i \times \hat{\boldsymbol{\rho}}_r f_r - \tau_U \tag{29}$$

where m_b is the mass of the BWS, and $\mathbf{a}_b$ is the acceleration of the center of mass. Vectors $\mathbf{f}_U$ and τ_U are the linear force and torque that the robot applies to the user to perform the STS task. Vectors $\mathbf{r}_h$ and $\mathbf{r}_i$ correspond to the positions where forces $\mathbf{f}_{sb}$, $\mathbf{f}_U$ and f_r are applied to the BWS with respect to its center of mass.

Equations from (24) to (29) form a linear system that can be written in matrix form in the following way:

$$\begin{bmatrix} 2\hat{\boldsymbol{\rho}}_l & \hat{\boldsymbol{\rho}}_r & \begin{bmatrix}0 & 1\end{bmatrix}^T & \mathbf{0}_{2\times2} & -\mathbf{I}_{2\times2} & -2\mathbf{I}_{2\times2} \\ 2\mathbf{r}_b \times \hat{\boldsymbol{\rho}}_l & \mathbf{r}_d \times \hat{\boldsymbol{\rho}}_r & r_{ax} & \mathbf{0}_{1\times2} & -[\mathbf{r}_c]_\times & -2[\mathbf{r}_c]_\times \\ -\hat{\boldsymbol{\rho}}_l & \mathbf{0}_{2\times1} & \mathbf{0}_{2\times1} & \mathbf{I}_{2\times2} & \mathbf{0}_{2\times2} & \mathbf{I}_{2\times2} \\ -\mathbf{r}_g \times \hat{\boldsymbol{\rho}}_l & 0 & 0 & [\mathbf{r}_e]_\times + \begin{bmatrix}R_w & 0\end{bmatrix} & \mathbf{0}_{1\times2} & [\mathbf{r}_f]_\times \\ \mathbf{0}_{2\times1} & -\hat{\boldsymbol{\rho}}_r & \mathbf{0}_{2\times1} & \mathbf{0}_{2\times2} & \mathbf{I}_{2\times2} & \mathbf{0}_{2\times2} \\ 0 & -\mathbf{r}_i \times \hat{\boldsymbol{\rho}}_r & 0 & \mathbf{0}_{1\times2} & [\mathbf{r}_h]_\times & \mathbf{0}_{1\times2} \end{bmatrix} \begin{bmatrix} f_l \\ f_r \\ f_{wsy} \\ \mathbf{f}_{wl} \\ \mathbf{f}_{sb} \\ \mathbf{f}_{sl} \end{bmatrix} = \begin{bmatrix} m_s(\mathbf{a}_s - \mathbf{g}) \\ I_s\alpha_s \\ m_l(\mathbf{a}_l - \mathbf{g}) \\ I_l\alpha_l \\ m_b(\mathbf{a}_b - \mathbf{g}) + \mathbf{f}_U \\ I_b\alpha_b + \tau_U + \mathbf{r}_h \times \mathbf{f}_U \end{bmatrix} \tag{30}$$

where $[\cdot]_\times$ is a cross product operator. Solving the previous equation gives the solution of the inverse dynamics of the robot.

In order to determine the values of $\mathbf{f}_U$ and τ_U required during the lifting of the user, it is necessary to perform a dynamic analysis of the person. To perform the dynamic analysis

https://doi.org/10.3390/act15060323

of the user during the STS task, the body is considered as a multibody system composed of three elements, the lower leg, the thigh and the torso. For the lower leg, we have the following dynamic equations:

$$m_t \mathbf{a}_t = m_t \mathbf{g} + \mathbf{f}_a + \mathbf{f}_k \tag{31}$$

and

$$l_t \alpha_t = \mathbf{r}_{at} \times \mathbf{f}_a + \mathbf{r}_{kt} \times \mathbf{f}_k \tag{32}$$

where m_t and I_t are the mass and the moment of inertia of the lower leg, and $\mathbf{a}_t$ and α_t are the linear and angular acceleration of the same segment, respectively. Vectors $\mathbf{f}_a$ and $\mathbf{f}_k$ are the forces at the ankle and knee joints. Vectors $\mathbf{r}_{at}$ and $\mathbf{r}_{kt}$ represent the positions of the ankle and knee joints with respect to the center of mass of the lower leg. Similarly, the following equations of motion for the thigh can be written:

$$m_f \mathbf{a}_f = m_f \mathbf{g} - \mathbf{f}_k + \mathbf{f}_h \tag{33}$$

and

$$l_f \alpha_f = -\mathbf{r}_{kf} \times \mathbf{f}_k + \mathbf{r}_{hf} \times \mathbf{f}_h \tag{34}$$

where m_f and I_f are the mass and the moment of inertia thigh, and $\mathbf{a}_f$ and α_f are the linear and angular acceleration of the same segment, respectively. Vector $\mathbf{f}_h$ is the force at the hip joint. Vectors $\mathbf{r}_{at}$ and $\mathbf{r}_{at}$ represent the positions of the knee and hip joints with respect to the center of mass of the tibia.

Finally, for the torso, we have the following dynamic equations:

$$m_c \mathbf{a}_c = m_b \mathbf{g} - 2\mathbf{f}_h + \mathbf{f}_U \tag{35}$$

and

$$l_c \alpha_c = r_{hc} \times (\mathbf{f}_U - 2\mathbf{f}_h) + \tau_U \tag{36}$$

where m_c and I_c are the mass and the moment of inertia of the torso, and $\mathbf{a}_c$ and α_c are the linear and angular acceleration of the same segment, respectively. Vector $\mathbf{r}_{hc}$ represent the position of the hip joint with respect to the center of mass of the torso. Equations from (31) to (36) can be written in matrix form in the following way:

$$\begin{bmatrix} \mathbf{I}_{2\times2} & \mathbf{I}_{2\times2} & \mathbf{0}_{2\times2} & \mathbf{0}_{2\times2} & \mathbf{0}_{2\times1} \\ [\mathbf{r}_{at}]_\times & [\mathbf{r}_{kt}]_\times & \mathbf{0}_{1\times2} & \mathbf{0}_{1\times2} & 0 \\ \mathbf{0}_{2\times2} & -\mathbf{I}_{2\times2} & \mathbf{I}_{2\times2} & \mathbf{0}_{2\times2} & \mathbf{0}_{2\times1} \\ \mathbf{0}_{1\times2} & -[\mathbf{r}_{kf}]_\times & [\mathbf{r}_{hf}]_\times & \mathbf{0}_{1\times2} & 0 \\ \mathbf{0}_{2\times2} & \mathbf{0}_{2\times2} & -2\mathbf{I}_{2\times2} & \mathbf{I}_{2\times2} & \mathbf{0}_{2\times1} \\ \mathbf{0}_{1\times2} & \mathbf{0}_{1\times2} & -2[\mathbf{r}_{hc}]_\times & [\mathbf{r}_{hc}]_\times & 1 \end{bmatrix} \begin{bmatrix} \mathbf{f}_a \\ \mathbf{f}_k \\ \mathbf{f}_h \\ \mathbf{f}_U \\ \tau_U \end{bmatrix} = \begin{bmatrix} m_t(\mathbf{a}_t - \mathbf{g}) \\ I_t \alpha_t \\ m_f(\mathbf{a}_f - \mathbf{g}) \\ I_f \alpha_f \\ m_c(\mathbf{a}_c - \mathbf{g}) \\ I_c \alpha_c \end{bmatrix} \tag{37}$$

Figure 6 illustrates the computational flow used for the inverse dynamics calculation of the Papalotl robot. The process begins with the definition of the target points that the robot must follow, based on the biomechanics of the STS task. Trajectory planning is then performed to generate continuous time profiles of position, velocity, and acceleration. These trajectories are provided as inputs to the inverse kinematics module, where the corresponding joint variables and the positions of the centers of mass (CoM) of the robot links are obtained. Using this information, the linear and angular accelerations of the robot links are computed. Simultaneously, the inverse dynamics of the user is calculated to obtain the force $\mathbf{f}_U$ and torque τ_U that the robot must apply to the user. Finally, all of these results are integrated within the inverse dynamics formulation to compute the actuator

https://doi.org/10.3390/act15060323

forces required for the robot to track the desired trajectory, considering both the robot and user dynamics.

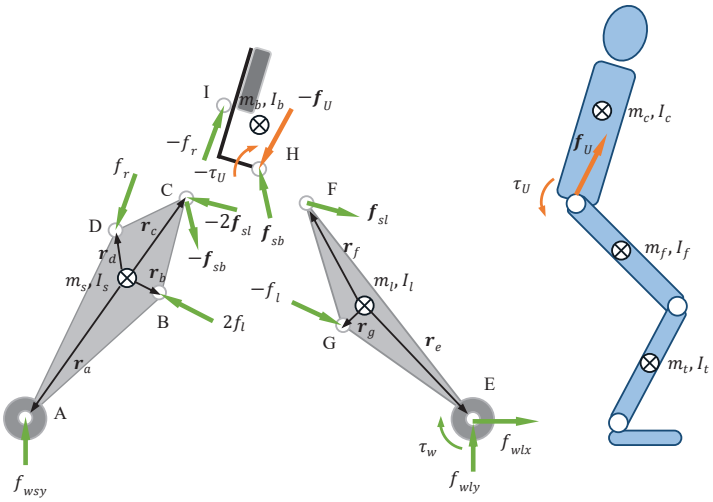


Figure 5. Free-body diagrams of the robot mechanism.

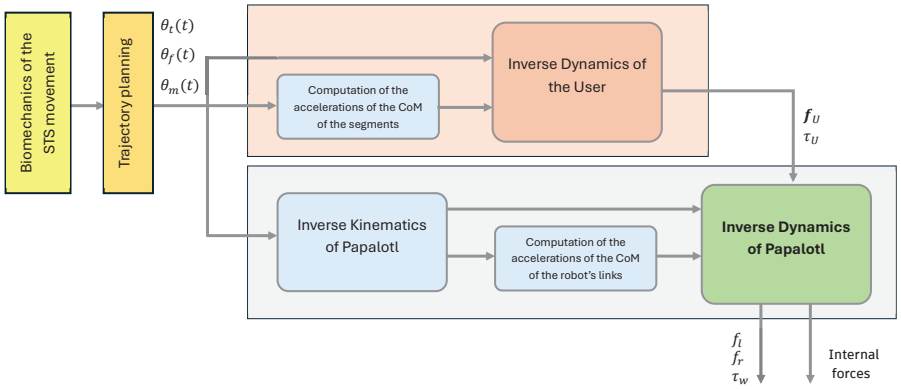


Figure 6. Computation of the inverse dynamics.

The inverse kinematics model was graphically validated using SolidWorks 2023. On the other hand, a numerical validation procedure was implemented to assess the inverse dynamics model by comparing its results with the forward dynamics computed individually for each link of the mechanism in Simulink. In this approach, the instantaneous forces generated in the joints during the execution of a test trajectory were obtained using the inverse dynamics model. The test trajectory corresponded to the execution of the STS task for a 12-year-old child, which is described in detail in the following section. The instantaneous accelerations of the links were computed using their respective equations of motion and the force history previously obtained from the inverse dynamics calculation. These accelerations were subsequently integrated to obtain the velocities and position trajectories. The resulting motion was analyzed to verify the accurate reproduction of the reference trajectory. The test trajectory was discretized into 2000 points. Due to the numerical stiffness associated with the dynamics of the mechanism from which the applied forces are derived, the implicit fixed-step integrator `ode14x` is employed, as it provides a stable and consistent integration for this type of numerical input. Model consistency was assessed using the Root Mean Square Error (RMSE) and the coefficient of determination (R^2) for the link coordinates. Table 1 presents the values of the RMSE and R^2 for the coordinates of each of the links. The maximum positional RMSE was 0.0174 m in the Y coordinate of the BWS center of mass (2.03% normalized error), while the highest angular

RMSE was 0.0059 rad in the front limbs (0.22% normalized error). R^2 values were close to or greater than 0.9, indicating that over 90% of the trajectory variability obtained through forward dynamics is explained by the inverse dynamics solution and thus demonstrating a high level of consistency between both models.

Table 1. RMSE and R^2 values for each link and coordinate.

Link	Coordinate	RMSE	R^2
Rear structure	X-Coordinate	0.0096	0.9925
	Y-Coordinate	0.0145	0.9076
	Angle	0.0004	0.9999
Front limbs	X-Coordinate	0.0080	0.9810
	Y-Coordinate	0.0190	0.8792
	Angle	0.0059	0.9977
BWS	X-Coordinate	0.0007	0.9999
	Y-Coordinate	0.0174	0.9596
	Angle	0.0040	0.9996

3.2. Simulation of the STS Task

This section presents the results of a series of simulations using the model of the robot that was previously introduced. The purpose of these simulations is to analyze the performance of the robot during a STS task. In these simulations we considered the dimensions of children of 8, 10 and 12 years old according to reference [42]. Table 2 shows the size and weight of the leg, thigh, and trunk segments corresponding to each age. The body weights were calculated based in [18].

Table 2. Heights and weights of 8-, 10- and 12-year-old boys [18,42].

Age (Years)	Weight (kg)	Height (mm)	L. Leg (mm)	L. Thigh (mm)	L. Torso (mm)	W. Leg (kg)	W. Thigh (kg)	W. Torso (kg)
8	29.3	1279	354	250	404	1.44	3.21	12.69
10	36.3	1381	393	267	444	1.79	3.97	16.22
12	42.7	1480	427	288	478	2.11	4.68	20.37

In [17], a descriptive study of the STS movement pattern presents the sequence of values of the orientation of different segments of the human body during the movement. We use data of five different postures, taken at 25% intervals of the movement, to generate the input trajectories in the simulations. Table 3 shows the values of the angles of the body lower leg θ_t the ankle, the thigh θ_f and the torso θ_m measured with respect to the horizontal line, as can be seen in Figure 3. Some adjustments were made to the data given by [17]. To obtain θ_t, an angle of 21.74° was added to obtain the measurement with respect to the horizontal line. Additionally, the last values of the variables were adjusted to obtain an almost vertical posture (θ_t is not equal to 90° because the system reach a singular configuration).

Table 3. Angles (°) of the lower leg, the thigh and the torso in five postures of the STS task.

Angle	0%	25%	50%	75%	100%
θ_t	84.01	81.52	77.70	85.45	87.00
θ_f	169.40	166.46	145.23	109.70	90.00
θ_m	79.78	64.77	49.40	75.90	90.00

https://doi.org/10.3390/act15060323

Figure 7 shows the trajectories of θ_t, θ_f and θ_m using a cubic spline, with constraints at the velocities, with these being equal to zero at the beginning and end of the trajectory. The movement time of the tests performed in [17] ranged from 1.3 to 2.5 s; for the simulations presented, here we choose a period of time of 2.0 s to perform the STS task.

Figure 8 shows a sequence of five postures of the robot during the STS task. The green lines show the trajectories of the centers of mass of the robot links. In this graph, we used the dimensions of the prototype that was built, which is described in the following section. The polygons that represent the links in the graph have their vertices at the points where the joints are placed, and therefore, their shape is, in appearance, different from that of the prototype parts. The yellow lines show the trajectories of the centers of mass of the user segments. In the figure, the graphic elements representing the user only show the segments of the tibia, femur and trunk, and the head is not included. The geometry of the user considers the parameters of a 12-year-old child, according to Table 2. An interesting aspect of the biomechanics of the STS movement, which can be observed in this figure, is that the trajectory of the trunk's center of mass has the shape of a flipped L. In the first part of the movement, the center of mass moves horizontally until it reaches the point where its projection on the ground coincides with the position of the feet, and then it follows an almost vertical path.

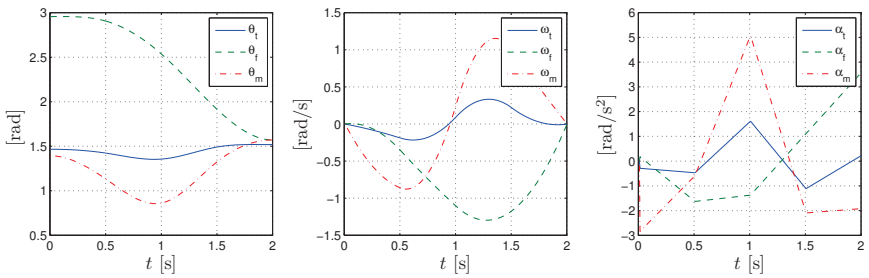


Figure 7. Trajectories of θ_t, θ_f and θ_m using a cubic spline during the STS task.

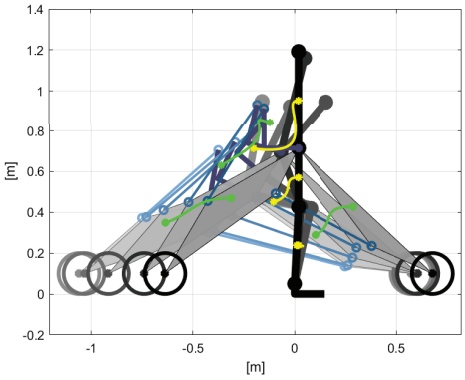


Figure 8. Sequence of robot postures during the STS task. The green lines show the trajectories of the centers of mass of the robot links, and the yellow lines show the trajectories of the centers of mass of the user segments.

Figure 9 shows the solution of the inverse kinematics of the robot during STS task. In this figure, $q_w = (q_x + q_{x\text{ini}})/R_w$, where $q_{x\text{ini}}$ is the initial value of q_x for the executed trajectory. On the other hand, $\dot{q}_w = \dot{q}_x/R_w$. The geometric parameters of the fabricated robot used in the simulations are $l_1 = 0.9$ m, $l_2 = 0.9$ m, $d_r = 0.518$ m $e_r = 0.17$ m, $l_r = 0.2$ m, $R_w = 0.1$ m, $\delta_1 = 20°$, $\delta_2 = 10°$ and $\delta_r = 13°$. From the results obtained in the simulations, it it is observed that the maximum length of q_l is 0.729 m, and its minimum value is 0.539 m, resulting in a range of motion of 0.19 m for this movement in children from 8 to 12 years of age. In the case of the BWS linear actuator, the maximum length of q_r is 0.665 m and the minimum is 0.477 m and, resulting in a range of motion of 0.188 m. The

https://doi.org/10.3390/act15060323

maximum angular rotation of active wheels is 1.267 rad (72.6°) for the case of the size of an 8-year-old child. The maximum required velocities at the actuators are $\dot{q}_{l\,\max} = 0.1837$ m/s, $\dot{q}_{r\,\max} = 0.2365$ m/s and $\dot{q}_{w\,\max} = 1.919$ rad/s.

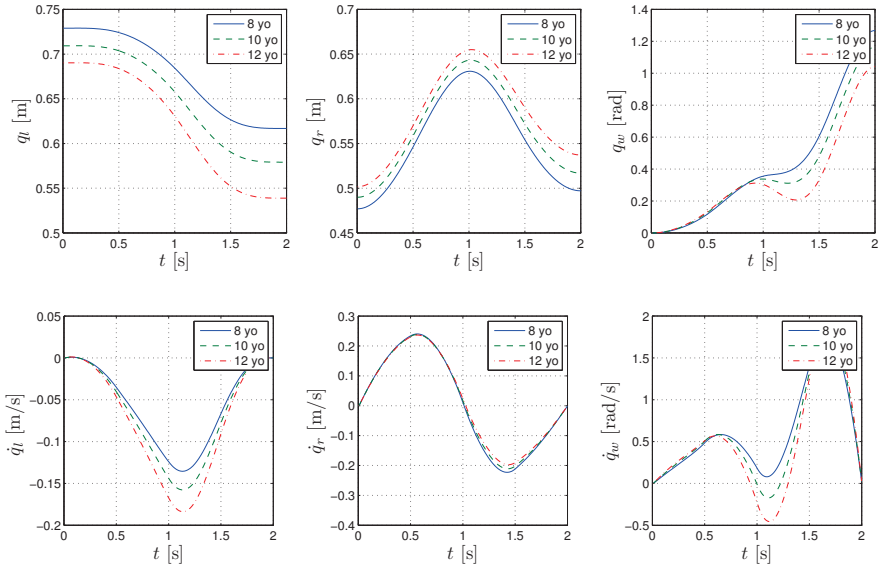


Figure 9. Solution of the inverse kinematics of the robot during the STS task.

Figure 10 shows the required force and torque applied to lift the user from the sitting position. Figure 11 presents the solution of the inverse dynamics of the robot during the STS task. The inertial parameters of the robot are $m_s = 20.048$ kg, $m_l = 11.678$ kg, $m_b = 5.609$ kg, $I_s = 1.468$ kg-m^2, $I_l = 0.976$ kg-m^2 and $I_b = 0.106$ kg-m^2. The position of the joints in the rear structure given with respect to a reference frame in its center of mass are: ${}^1\mathbf{r}_a = \begin{bmatrix} 0.048 & -0.490 \end{bmatrix}^T$ m, ${}^1\mathbf{r}_b = \begin{bmatrix} 0.133 & 0.175 \end{bmatrix}^T$ m, ${}^1\mathbf{r}_c = \begin{bmatrix} 0.048 & 0.410 \end{bmatrix}^T$ m and ${}^1\mathbf{r}_d = \begin{bmatrix} -0.068 & -0.095 \end{bmatrix}^T$ m. On the other hand, the position of the joints in the front limbs given with respect to a reference frame in its center of mass are: ${}^2\mathbf{r}_e = \begin{bmatrix} 0.025 & -0.507 \end{bmatrix}^T$ m, ${}^2\mathbf{r}_f = \begin{bmatrix} 0.025 & 0.393 \end{bmatrix}^T$ m and ${}^2\mathbf{r}_g = \begin{bmatrix} -0.079 & -0.198 \end{bmatrix}^T$. The position of the joints of the BWS given with respect to a reference frame in its center of mass are: ${}^b\mathbf{r}_h = \begin{bmatrix} 0.141 & -0.130 \end{bmatrix}^T$ m and ${}^b\mathbf{r}_i = \begin{bmatrix} -0.029 & 0.070 \end{bmatrix}^T$ m. The maximum force required at the linear actuator of the structure is 540.58 N. On the other hand, for the BWS actuator, the maximum force is 206.83 N. The maximum required torque in the active wheels is 3.50 N-m at the end of the trajectory. Figure 12 presents the required power in the actuators during the STS task, the maximum power is required in the linear actuator of the mobile platform $P_{l\,\max} = 81.47$ W.

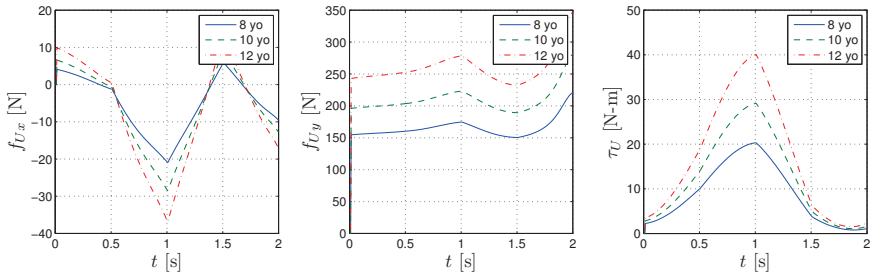


Figure 10. Required force and torque applied to the user to perform the STS task.

The results show that, as the user's age increases, the requirements in terms of speed, force, and power also increase for all actuators. Inverse dynamics additionally provides

https://doi.org/10.3390/act15060323

the internal forces acting at the joints of the robot. The maximum forces generated during the STS task, corresponding to the case of the 12-year-old child, were used to verify the mechanical strength of the robot components during the design process.

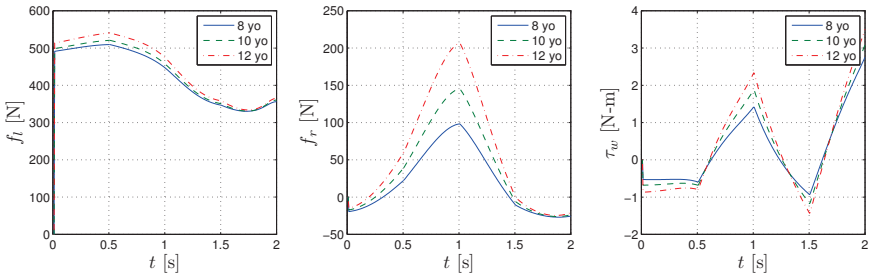


Figure 11. Solution of the inverse dynamics of the robot during the STS task.

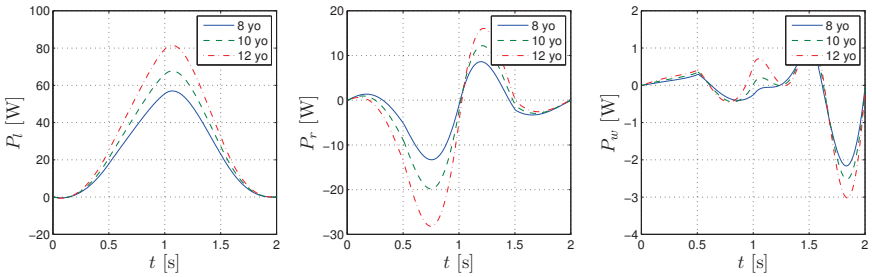


Figure 12. Required power in the actuators during the STS task.

On the other hand, although trajectory planning using cubic splines allows for an adequate reproduction of the movement during the STS task, discontinuities are observed in the applied forces $\mathbf{f}_U$ and $\boldsymbol{\tau}_U$ acting on the user. This effect is more noticeable at the beginning of the trajectory, where forces must be applied almost instantaneously, without starting from zero values. This situation may cause discomfort to the user; therefore, it is considered appropriate to investigate possible adaptations to the trajectory planning strategy in order to mitigate this effect.

4. Development of a Physical Prototype

Figure 13 shows the construction of the Papalotl robot prototype, and Figure 14 shows the robot being used by an 8-year-old-child. The design of the robot mechanical structure was guided by the following main requirements:

- (R1) Load capacity: The robot shall be capable of supporting a user mass of up to 50 kg. This value corresponds to the upper range of body mass for a 12-year-old child according to standard growth charts [43].
- (R2) Vertical range of motion: The robot shall provide a vertical range of motion from 20 cm above the floor to a maximum height of 90 cm, approximately corresponding to the height of a 12-year-old child.
- (R3) Motion speed: The STS movement shall be executable in approximately 2 s, with the capability to operate at lower speeds, depending on the therapeutic activities to be performed.

The main structure of the robot is made of stainless steel, so that it can be tested in hospital areas and give high structural rigidity. The dimensioning of the robot geometry was done through a trial-and-error iterative process, taking into account the following considerations:

- Results of the inverse dynamics.
- Link collision avoidance.

https://doi.org/10.3390/act15060323

- Space for actuator integration.
- Dimensions of available mechanical components.
- Manufacturing feasibility.



Figure 13. Construction of the prototype of the Papalotl rehabilitation robot.



Figure 14. Current version of the prototype with a user.

Each component of the mechanism was evaluated using Finite Element Analysis in SolidWorks, calculating Von Mises equivalent stresses and considering the elastic limits of the materials. The maximum forces at each joint, generated by applying a 50 kg load at the central joint over the robot's entire range of motion, were used as a reference for the analysis to ensure compliance with the load capacity requirement (R1). All components exhibit a safety factor greater than 2, ensuring the structural integrity of the mechanism. Similarly, the maximum joint forces were obtained from the inverse dynamics of the robot during the STS movement and were used to evaluate the resistance of the mechanical elements, in order to verify compliance with the motion execution requirement (R3). Inverse kinematics was employed to determine the motion ranges of the actuated joints and to verify compliance with requirement (R2). Moreover, collision avoidance between links was

https://doi.org/10.3390/act15060323

evaluated throughout the entire vertical range of motion to prevent mechanical interference and also ensure compliance with requirement (R2). Another aspect taken into account was the availability of space for actuator integration, transmission elements, maintenance access, and ease of assembly/disassembly. Additionally, the dimensions of commercially available mechanical components were considered to ensure proper integration. Finally, the manufacturability of the design was evaluated to guarantee that the mechanical parts could be fabricated and assembled using standard manufacturing processes and tolerances.

The selection of the control equipment was based on a set of functional and integration requirements essential for the system's intended therapeutic application. Primarily, the motor controller was required to support the synchronized operation of no fewer than five motors, ensuring precise and coordinated multi-axis motion. Furthermore, the system needed to provide robust data acquisition capabilities to enable the efficient integration of various sensors used in the real-time control of the robotic platform. Programming flexibility across multiple languages was also considered essential to facilitate the development of a comprehensive and adaptable user interface, capable of managing interactions among the user, the therapist, and the robotic system. In addition, the control platform was expected to support the integration of external devices, thereby expanding the system's functionality to accommodate a wide range of therapeutic tasks and rehabilitation scenarios.

According to these requirements, the Galil motion controller DMC 4080 was selected as the main controller of the robot. The Galil motion controller works as a master-slave control system. The Galil motion controller acts as the master, while the slave devices execute the commanded motion through power amplification and local current regulation. This controller was selected for its multi-axis capability, allowing the coordination of multiple motors simultaneously. Some of its features are:

- Proportional Integral and Derivative (PID) compensation with speed and acceleration prediction.
- Multitasking program memory to simultaneously run eight application programs.
- Unassigned inputs and outputs to synchronize movement with external events.

The control architecture of the system is illustrated in Figure 15. Each motor is equipped with an external control driver specifically configured to match its individual characteristics. These drivers serve as the primary controllers, receiving command inputs—either direct current (DC) voltage or pulse-width modulation (PWM) signals—from the multi-axis controller.

Position feedback for trajectory generation is obtained through encoders integrated into each actuator, enabling the creation of both position and velocity profiles. The Galil controller operates in PVT (position–velocity–time) mode, which allows the definition of time-synchronized motion profiles for up to eight axes, facilitating coordinated multi-axis control.

The robot is controlled by the operator through a dedicated computer that communicates with the motion controller via an Ethernet cable or through a router. The robot can be operated either through the commands available in a graphical user interface or by means of a joystick. The user interface and the system's control logic were programmed in LabVIEW Version 14.0.

The dedicated computer is also intended to perform the acquisition and processing of the user's biosignals. These include physiological signals such as electroencephalography (EEG), electromyography (EMG), and other relevant measurements that allow the analysis of the patient's progress during the rehabilitation process. On this platform, movement intention detection algorithms based on the measurement of the user's biosignals [44] are intended to be implemented, with the aim of preliminarily evaluating their performance and analyzing their potential usefulness as support for the therapeutic process.

https://doi.org/10.3390/act15060323

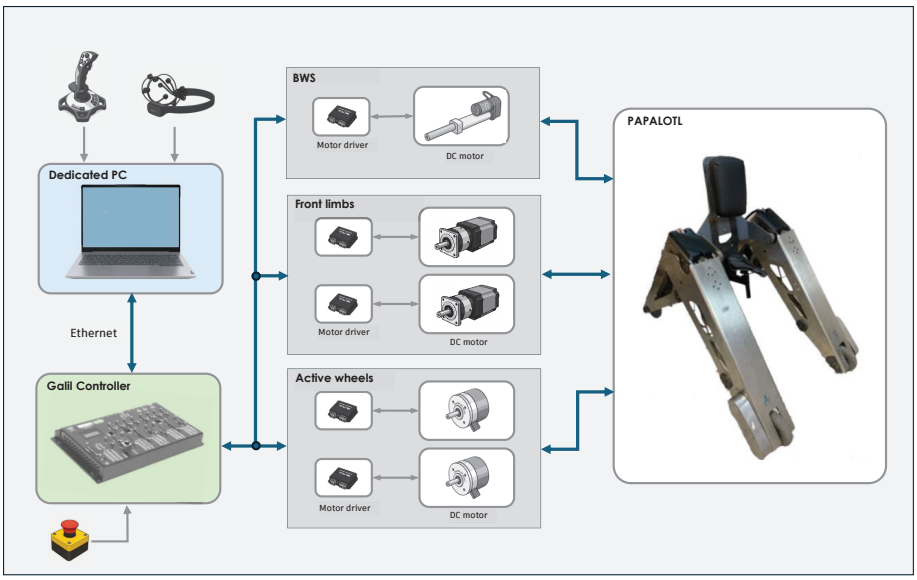


Figure 15. Control architecture of the robot, the arrows indicate the data flow between the interface and the controller, as well as the control signals from the controller to the drivers. The drivers send current to the motors, and feedback on position and speed is received from the motor sensors.

To regulate the position of the BWS mechanism, a Glideforce LACT8P-12V-20 (Pololu, Las Vegas, NV, USA) linear actuator was employed. The lifting mechanism consists of two linear actuators, each powered by a 337 W DC motor with a maximum speed of 5310 RPM and a peak torque of 343.4 in-oz. Each motor is equipped with a 1400 counts-per-turn quadrature encoder featuring dual channels and RS-422 line drivers (AndyMark Inc., Kokomo, IN, USA) To achieve the required torque output, each motor is coupled to a planetary gearbox with a reduction ratio of 5.95:1. The gearbox output shaft is connected to a pair of leadscrews with a pitch of 1/6 inch. Motor current is regulated by two Pololu Dual G2 High-Power Motor Drivers (24 V, 18 A).

For the active wheels, two MAXON EC 90 brushless motors (Maxon International Ltd., Brünigstrasse, Sachseln, Switzerland) were utilized. These motors have a nominal power of 90 W, a rated voltage of 24 V, a no-load speed of 3190 RPM, and a nominal (continuous) torque of 444 mNm. Each motor is coupled to a high-ratio gearbox providing a 936:1 reduction, resulting in a maximum continuous output torque of 30 Nm. The brushless motors are controlled using ESCON 50/5 servo drivers (Maxon International Ltd., Brünigstrasse, Sachseln, Switzerland).

The platform incorporates safety measures at both the mechanical and control levels. The mechanical limits of the prismatic joints only allow configurations within biomechanically safe ranges. Additionally, an emergency stop button is directly connected to the controller to halt the robot if necessary. When activated, the command disables the actuators and stops all robot motion. Furthermore, since self-locking lead screws are used in the transmission, the mechanism does not collapse when the motors are not energized.

To perform the STS movement, the robot's inverse kinematics is first solved, taking into account the user's dimensions. Subsequently, the history of position-velocity values of the joint variables is processed and converted into counts (equivalent to the encoder counts per revolution) and velocity expressed in counts per unit time. These profiles, along with the time steps, are executed by the Galil controller, which runs a feedback control loop using data from the position sensors.

Using the PVT mode of the controller, motion profiles are programmed, specifying the relative distance to move and the speed within a time frame. The controller interpolates the motion profiles using a third-degree polynomial equation. Figure 16 shows the position and

https://doi.org/10.3390/act15060323

velocity encoder readings of the linear actuator of one of the front limbs, when the robot is performing a test of the STS task. In this test, the robot provides assistance to a healthy 8-year-old child. The trajectory was executed over a period of 10 s, with a discretization of 200 samples.

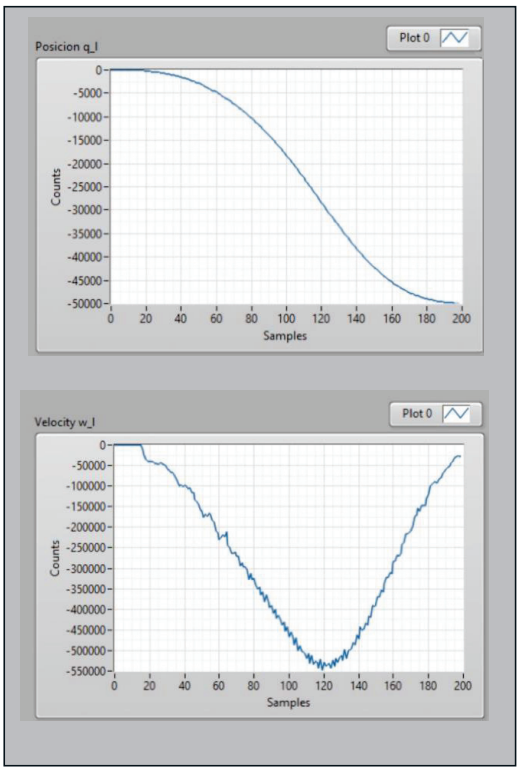


Figure 16. Displacement and velocity of the linear actuator of one of the front limbs during for STS task, in units of counts vs. samples and counts/s vs. samples, respectively.

Figure 17 shows four images of the robot performing the STS movement. It can be observed in the sequence of images that the synchronized movement of the actuators allows the structure of the robot to move symmetrically, while simultaneously lifting the user in the correct manner according to the biomechanics of the STS movement and the inverse kinematics of the robot. A video of this test can be seen at: https://www.youtube.com/watch?v=m5GgzLYEieY (accessed on 3 April 2024).

The experimental evaluation of the robot shows that the proposed architecture allows the integration of assistance for the STS task and support during gait within a single platform. During the experiments, it was possible to validate the distinctive characteristics of the proposed concept. By coordinating the actuators of the front limbs and the active wheels, it is possible to generate arbitrary trajectories for the user's hip in the sagittal plane. In this way, the mobile platform does not require a two degree of freedom linkage specifically intended to generate motion in this plane at the hip level. Instead, the motion of the active wheels is used to control the displacement along the x-axis during the STS movement. During gait assistance, the user is supported by the BWS, while the active wheels control the forward motion of the platform. The robot ensures safe support for the patient due to its inherent static stability and by keeping them within the support polygon. The use of self-locking lead screws provides an additional level of safety when the motors are not active.

https://doi.org/10.3390/act15060323



Figure 17. Test of the robot lifting an 8-year-old child with movement profiles calculated in synchrony (www.youtube.com/watch?v=m5GgzLYEieY) (accessed on 3 April 2024).

One observed limitation of the current geometry of the prototype is the large space required when the front limbs are fully extended. A more thorough dimensional synthesis is required to address this issue and improve the overall design of the robot. In this work, a set of functions was identified that led to the development of a robot concept whose characteristics are presented in Section 3. These functions and characteristics can be used to formulate the objectives and requirements of a design problem, from which new mechanical architectures could be developed that more adequately satisfy the requirements of the problem addressed.

5. Robot-Assisted Sit-to-Stand Rehabilitation Exercises

This section explains several sit-to-stand rehabilitation exercises recommended by [18,45] that can be supported by the robot during execution. The mechanism was designed to be capable of lifting a person even from the floor; consequently, it can perform various exercises due to the following functionalities:

1. Supporting the patient's body and weight during ascending and descending movements.
2. Guiding the patient's hips upward, downward, forward, backward, or through a combination of these trajectories.
3. Guidance of the patient's trunk in both forward and backward inclinations.

5.1. Momentum Strategies Exercises

There are two basic rehabilitation strategies described in [18] to transfer from sitting to standing utilized: a momentum-transfer strategy and zero-momentum strategy.

The momentum strategy: Involves initially generating forward momentum by flexing forward at the hips quickly as the trunk and head translate in a horizontal direction, causing the center of mass (COM) to shift toward and over the feet. Here, in these exercises, an important element of practice is moving the upper body forward over the feet (i.e., hip flexion with upper trunk extended). Having the patient cross the arms in front while guided to flex forward and the hips helps to keep the upper trunk extended and minimizes upper trunk flexion (Figure 18a). Alternatively, the patient's arms can be supported on a rolling tray table that can be guided or a large exercise ball. (Figure 18a,b): This strategy can be guided by the robot by maintaining the backrest vertical while simultaneously translating the hips upward and forward through the seat and the harness attachment, as observed in the functional testing results (Figure 17).

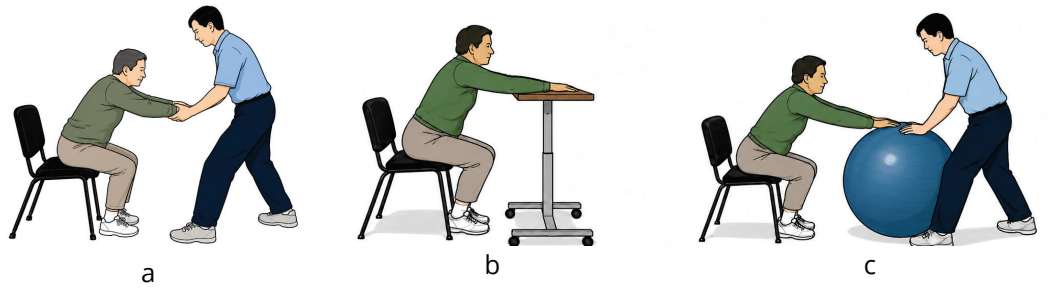


Figure 18. Momentum transfer strategy, sit-to-stand rehabilitation. (**a**) The therapist provides support for implementing the strategy. (**b**) Supported by a rolling tray table (**c**) Supported by a large exercise ball.

The zero-momentum strategy: It is forward flexion of the trunk until the COM is within the base of support (BOS) of the feet, as can be seen in the Figure 19. Then, there is a vertical lift of the body mass to the standing position. The zero-momentum strategy is more stable than the momentum-transfer strategy but requires grater muscle force to perform. People with lower extremity weakness who utilize the zero-momentum require arm rest to push off with their upper extremities. This type of strategy can be executed in a robot-assisted manner, as the device can guide the trunk's trajectory at a very low speed to ensure patient safety. Using the same kinematic model, it would also be possible to design a different trajectory to program into the robot, thus practicing the opposite movement—from standing to sitting. In fact, performing this movement with a robotic assistive device provides the patient with much greater safety since, without this assistance, the patient would always have to turn around and look at the surface; in contrast, the robot's assistance can give the patient a great sense of security and peace of mind.



Figure 19. Forward flexion of the trunk.

5.2. Partial-Range and Strength (Isometric and Eccentric)

On the other hand, there are the exercises of Partial-Range and Strength (Isomentric and Eccentric). These are ideal for the current prototype since they demand a high load capacity (torque) but low or zero velocity.

- *Thigh-Off Training/Hovering:* This is the most demanding exercise. It consists of lifting the hips just 2 to 5 cm off the seat and holding that position isometrically (statically) for 3 to 5 s before sitting back down. This is where the 50 kg load capacity of the motor is crucial, while the low speed serves as an advantage to avoid destabilizing the patient.

https://doi.org/10.3390/act15060323

- *Assisted Partial Squats*: The patient starts from a standing position and descends slowly in a controlled manner (eccentric phase) until stopping halfway and then moves back up to a full stand.
- *Asymmetric Feedback Exercises (for Hemiparesis)*: Commonly used in rehabilitation for patients who have suffered a stroke and tend to shift all of their weight to their unaffected side. Asymmetric Weight-Bearing Exercises: The affected foot is placed slightly behind the unaffected foot (which mechanically forces it to bear more weight) during an attempt to stand. The robotic device can compensate for the force that the affected leg fails to generate.
- *Eccentric Control Exercises (Controlled Lowering):* Controlled Lowering (Stand-to-Sit): The exercise focuses exclusively on the lowering phase. The patient starts in an upright position and must take between 4 and 6 s to sit down, resisting gravity. The low-speed constraint of the motors can act as a safety brake, assisting the patient if they lose control and preventing a sudden drop onto the chair.

5.3. Rehabilitation Exercises for Children with Spasticity

In the case of children with spasticity, certain preparatory exercises must be mastered before training them to stand up. First and foremost, sitting posture (sedestation) must be corrected. The robot's capability to support the child's body weight, combined with its adjustable height settings, can significantly aid in properly positioning the knees and ankles. On the other hand, there are forward trunk flexion exercises, as exemplified in [46]. Children with spasticity frequently exhibit an "extensor thrust" (a rigid backward pushing motion). To counteract this reflex, forward leaning is trained. In this scenario, the robot offers a distinct advantage: it can slowly move the actuated backrest forward to guide the rehabilitation path and then return just as slowly to the initial position.

Weight-Shifting and Elevation (Standing Up/Bipedalism): Once the child masters the forward leaning phase, they can perform the following exercises with assisted support from the robot:

- Adapted Supports at Varying Heights: This exercise is traditionally practiced using chairs or benches with adjustable heights. The training begins from a higher surface (which reduces mechanical difficulty), and the height is gradually decreased as the strength of the quadriceps and glutees increases. This entire progression can be dynamically adjusted and automated using the robot.
- Use of Stable Traction Surfaces: This involves placing a bar, a firm table, or a walker directly in front of the patient. The child can hold onto it to support the transition, progressively reducing manual assistance to foster autonomy. Since the robot is an automated device equipped with front traction wheels—or designed to function as a motorized traction walker—it could be effectively utilized to experiment with this type of motion.
- Figure 20 illustrates the concept of how the robot can be utilized. Starting from an initial seated state Figure 20a, the robot's legs and front wheels can be actuated to adjust the height and guide the hips, leading to a final standing state Figure 20b. In this position, the system can subsequently be used as an automated walker to support gait rehabilitation.

https://doi.org/10.3390/act15060323



(a) (b)

Figure 20. (**a**) Initial position and (**b**) final robot position for rehabilitation exercises.

5.4. Gait Training

Another advantage is that the robot functions as an automated walker: it supports the patient's body weight and can gradually guide them forward and through turns. Consequently, this device serves as a valuable tool for therapists; although it does not directly exercise the legs, it assists in correcting the gait of patients who still require physical support while walking. This therapeutic approach was reviewed years ago at the Infanta Sofía Rehabilitation Center, and it is also being implemented at the CRIT Teletón Saltillo, with which an agreement has been signed to collaborate on the next phase of the project.

6. Discussion (Future Work)

The core concept of this fabricated robot has been established as a testbed for the development of further research regarding human–robot interaction and user intention recognition: Currently, a human–robot interface is being developed to measure electroencephalographic (EEG) brain signals using a headband equipped with an Inertial Measurement Unit (IMU). This setup aims to identify and measure the user's intention to perform the sit-to-stand motion and actuate the robot accordingly; this approach has already been utilized to drive the robot's actuators. However, implementing this requires identifying specific signal patterns, and one of the primary techniques employed is based on Deep Learning. Successfully identifying these patterns will enable a more comprehensive rehabilitation process, as it establishes a direct connection between cortical motor intention and robotic actuation. Furthermore, future work aims to implement a control strategy based on measuring the forces exerted by the user or patient via the robot itself—a type of control framework that can be reviewed in works such as [32]. Measuring user forces through the robot will be achieved using sensors such as strain gauges integrated into the seat, alongside a force plate to measure ground reaction forces (GRF) under the patient's feet.

To directly measure the movement intention of the muscles, electromyographic (EMG) sensors can be utilized. A recent study in [47] analyzing the biomechanics of sit-to-stand and stand-to-sit transfers provides critical insights by incorporating extensive data from EMG sensors. This work analyzes muscle activation patterns alongside trunk and knee biomechanics in the sagittal plane during both sit-to-stand (STS) and stand-to-sit (STSi) tasks. This study is highly relevant for future work because it identifies physical and physiological variables that can serve as natural triggers for robot control and actuation,

https://doi.org/10.3390/act15060323

leading to full-scale rehabilitation. In fact, that study lays the scientific foundation for developing hybrid, multimodal control schemes (fusing sEMG + IMU sensors). This framework allows assistive robots or exoskeletons to harmoniously and dynamically tune their robotic compliance as the patient's physical recovery progresses. Therefore, the subsequent phase of our project will integrate this type of muscular sensing with the headband's IMU and the EEG signals currently under development, creating a holistic feedback loop that maps movement intention directly to robotic actuation. Additionally, an inverse dynamics model is available to achieve superior motion control, which will be implemented by measuring the torque of the robotic actuators. To fully utilize this torque-measurement-based inverse dynamics control, specific hardware modifications must be made to the low-level motor drivers.

7. Conclusions

Rehabilitation robotics is one of the most promising trends in contemporary therapeutic medicine, providing solutions that enhance monitoring, precision and consistency in treatment. This paper has introduced the design of the Papalotl robot, a device intended to assist in the rehabilitation of the STS movement and walking. The robot has five degrees of freedom, and it is capable of controlling the position of the user's hip and torso orientation through the coordinated movement of the motors. On the other hand, by means of the active wheels it can follow and support the user in gait rehabilitation tasks. The structure provides an inherent static equilibrium through four ground contact points via the robot's wheels, positioned around the user to enhance safety during therapy. The integration of differential drive enables the robot to navigate a variety of path geometries and perform turning maneuvers during gait therapy, thereby increasing the flexibility and range of therapeutic exercises.

This work has presented the development of a comprehensive analytical model of the robot addressing crucial aspects such as robot position, velocity, and dynamics, providing necessary tools for the control of the robot, and the design the mechanical architecture of the prototype. This mathematical model provides quantitative information on the system's dynamic behavior and the interaction between the user and the robot, which enables the evaluation of trajectory planning algorithms for the STS task. The presented mathematical modeling constitutes a useful tool to guide the design of future clinical evaluations prior to the application of therapies in patients.

The design of the physical prototype and its control architecture was described in detail. An empirical test of the robot, conducted with a representative patient, an eight-year-old child, validated the functionality of the proposed system and the presented inverse kinematics model. This work opens multiple lines of research, including optimal dimensional synthesis using the presented mathematical modeling, as well as the development of trajectory planning algorithms to support effective rehabilitation therapy. On the other hand, ongoing work is focused on the development of a user interface designed to assist the patient in controlling the robot through the interpretation of electrical brain signals. The purpose is to develop a comprehensive rehabilitation therapy, with the hypothesis that both thought and movement supported by the robot would help the patient to rebuild neuronal connections in an efficient way.

Although preliminary empirical testing supports the feasibility of the platform, comprehensive clinical validation remains a subject of future work. Further research will focus on experimental trials with multiple patient groups and the assessment of long-term therapeutic outcomes in clinical settings.

https://doi.org/10.3390/act15060323

Author Contributions: Conceptualization, I.G.C. and H.A.M.; Methodology, I.G.C. and H.A.M.; Validation, H.A.M., I.G.C. and J.L.O.-A.; Formal analysis, H.A.M.; Investigation, I.G.C., H.A.M. and J.L.O.-A.; Writing—original draft, I.G.C. and H.A.M.; Writing—review & editing, H.A.M., I.G.C. and J.L.O.-A.; Visualization, H.A.M. and J.L.O.-A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Programa para el Desarrollo Profesional Docente (PRODEP) of the Mexican Ministry of Public Education (Secretaría de Educación Pública, SEP), grant number DSA/103.5/16/10616 and by the Programa de Fortalecimiento de Cuerpos Académicos IDCA 31581, Academic Group Code UACOAH-CA-142.

Institutional Review Board Statement: Ethical review and approval were waived for this study, as it involved the evaluation of a research prototype under controlled laboratory conditions and did not constitute a clinical study. Informed consent was obtained from the participant's legal guardian prior to participation and data acquisition.

Informed Consent Statement: Informed consent was obtained from the participant's legal guardian for participation in the study and for the publication of images.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: The authors express their gratitude to Juan Carrera, Isidro Hernández, and Daniel Parra for their valuable support in the development of this project. The authors also acknowledge Universidad Evangélica de El Salvador (UEES) and Universidad Tecnológica Centroamericana (UNITEC) for their institutional support.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. World Health Organization. *Global Report on Health Equity for Persons with Disabilities*; World Health Organization: Geneva, Switzerland, 2022.
2. INEGI. *La Discapacidad en México, Datos al 2014*; Technical Report; Instituto Nacional de Estadística y Geografía: Aguascalientes, Mexico, 2014.
3. Kang, J.; Martelli, D.; Vashista, V.; Martinez-Hernandez, I.; Kim, H.; Agrawal, S.K. Robot-driven downward pelvic pull to improve crouch gait in children with cerebral palsy. *Sci. Robot.* **2017**, *2*, eaan2634. [CrossRef] [PubMed]
4. Vaughan, C.L. Theories of bipedal walking: An odyssey. *J. Biomech.* **2003**, *36*, 513–523. [CrossRef]
5. Cignetti, F.; Decker, M.; Stergiou, N. Complexity and Human Gait. *Rev. Andal. Med. Deporte* **2010**, *3*, 2–12.
6. Lindgren, A. Risk Factors. In *Oxford Textbook of Stroke and Cerebrovascular Disorders*, 1st ed.; Oxford University Press: New York, NY, USA, 2014; Chapter 2, pp. 9–10.
7. Cramer, S.C.; Wolf, S.L.; Adams, H.P.; Chen, D.; Dromerick, A.W.; Dunning, K.; Ellerbe, C.; Grande, A.; Janis, S.; Lansberg, M.G.; et al. Stroke Recovery and Rehabilitation Research. *Stroke* **2017**, *48*, 813–819. [CrossRef]
8. American Association of Neurological Surgeons. *Spinal Cord Injury: Causes, Symptoms, and Treatments*; AANS Patient Information; American Association of Neurological Surgeons: Rolling Meadows, IL, USA, 2026.
9. Eunice Kennedy Shriver National Institute of Child Health and Human Development. *Spinal Cord Injury: Fact Sheet*; National Institutes of Health (NIH): Bethesda, MD, USA, 2026.
10. Healthline Editorial Team. *Everything You Need to Know About Cerebral Palsy*; Healthline Medical Network: New York, NY, USA, 2026.
11. Centers for Disease Control and Prevention. *Cerebral Palsy: What Is Cerebral Palsy?* U.S. Department of Health and Human Services: Washington, DC, USA, 2026.
12. Carrera, I.; Moreno, H.A.; Sierra, S.; Campos, A.; Munera, M.; Cifuentes, C.A. Technologies for Therapy and Assistance of Lower Limb Disabilities: Sit to Stand and Walking. In *Exoskeleton Robots for Rehabilitation and Healthcare Devices*; SpringerBriefs in Applied Sciences and Technology; Springer: Singapore, 2020. [CrossRef]
13. Carrera, I.; Moreno, H.A.; Saltarén, R.; Pérez, C.; Puglisi, L.; Garcia, C. ROAD: Domestic assistant and rehabilitation robot. *Med. Biol. Eng. Comput.* **2011**, *49*, 1201–1211. [CrossRef] [PubMed]
14. Esquenazi, A.; Talaty, M.; Packel, A.; Saulino, M. The ReWalk powered exoskeleton to restore ambulatory function to individuals with thoracic level motor complete spinal cord injury. *Am. J. Phys. Med. Rehabil.* **2012**, *91*, 911–921. [CrossRef]

15. Delp, S.; Anderson, F.; Arnold, A.; Loan, P.; Habib, A.; John, C.; Guendelman, E.; Thelen, D. OpenSim: Open-Source Software to Create and Analyze Dynamic Simulations of Movement. *IEEE Trans. Biomed. Eng.* **2007**, *54*, 1940–1950. [CrossRef] [PubMed]
16. Schenkman, M.; Berger, R.A.; Riley, R.W.; Mann, P.O.; Hodge, W.A. *Whole-Body Movements During Rising to Standing from Sitting*; Technical Report 2217543; MGM Institute of Health Professions: Navi Mumbai, India, 1990.
17. Nuzik, S.; Lamb, R.; VanSant, A.; Hirt, S. Sit-to-stand movement pattern. *J. Am. Phys. Ther. Assoc.* **1985**, *66*, 1708–1713. [CrossRef]
18. O'Sullivan, S.B.; Schmitz, T.J. *Improving Functional Outcomes in Physical Rehabilitation*, 2nd ed.; F. A. Davis Company: Philadelphia, PA, USA, 2016.
19. Laurent, G.; Valentini, F.; Loiseau, K.; Hennebelle, D.; Robain, G. Clawtoes in hemiplegic patients after stroke. *Ann. Phys. Rehabil. Med.* **2010**, *53*, 77–85. [CrossRef]
20. da Cunha, I.T.; Lim, P.A.; Qureshy, H.; Henson, H.; Monga, T.; Protas, E.J. Gait outcomes after acute stroke rehabilitation with supported treadmill ambulation training: A randomized controlled pilot study. *Arch. Phys. Med. Rehabil.* **2022**, *83*, 1258–1265. [CrossRef]
21. Tanovic, E. Effects of functional electrical stimulation in rehabilitation with hemiparesis patients. *Bosn. J. Basic Med. Sci.* **2009**, *9*, 49–53. [CrossRef]
22. del Peral, J.R.; Josa, S.J. Ejercicios de resistencia en el tratamiento y prevención de la sarcopenia en ancianos. Revisión sistemática. *Gerokomos* **2018**, *29*, 133–137.
23. Chugo, D.; Asawa, T.; Kitamura, T.; Jia, S.; Takase, K. A rehabilitation walker with standing and walking assistance. In *Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France, 22–26 September 2008*; IEEE: Piscataway, NJ, USA, 2008; pp. 260–265. [CrossRef]
24. Chang, Y.; Deep Singh, A.; Girard, J.; Wu, Z. *Sit-to-Stand and Mobility Assistance Device*; Technical Report; Department of Mechanical Engineering, University of Michigan: Ann Arbor, MI, USA, 2010.
25. Xiong, G.; Gong, J.; Zhuang, T.; Zhao, T.; Liu, D.; Chen, X. Development of Assistant Robot with Standing-up Devices for Paraplegic Patients and Elderly People. In *Proceedings of the 2007 IEEE/ICME International Conference on Complex Medical Engineering, Beijing, China, 23–27 May 2007*; IEEE: Piscataway, NJ, USA, 2007; pp. 62–67. [CrossRef]
26. Salah, O.; Sessa, S.; El-Bab, A.M.R.F.; Kobayashi, Y.; Takanishi, A.; Fujie, M. Modeling and simulation for support robot tracking a human sit to stand motion. In *Proceedings of the 2016 28th International Conference on Microelectronics (ICM), Giza, Egypt, 17–20 December 2016*; IEEE: Piscataway, NJ, USA, 2016; pp. 81–84. [CrossRef]
27. Salah, O.; Ramadan, A.A.; Sessa, S.; Ismail, A.A.; Fujie, M.; Takanishi, A. ANFIS-based Sensor Fusion System of Sit-to-stand for Elderly People Assistive Device Protocols. *Int. J. Autom. Comput.* **2013**, *10*, 405–413. [CrossRef]
28. Matia Mobility. Tek RMD: Robotic Mobility Device. Product Specifications and Operational Overview. 2026. Available online: https://www.matiamobility.com/tek-rmd/ (accessed on 23 May 2026).
29. Kim, J.; Gu, G.M.; Heo, P. Robotics for Healthcare. In *Biomedical Engineering: Frontier Research and Converging Technologies*; Biosystems and Biorobotics; Springer: Cham, Switzerland, 2016; Volume 9. [CrossRef]
30. Matjačić, Z.; Zadravec, M.; Oblak, J. Sit-to-Stand Trainer: An Apparatus for Training "Normal-Like" Sit to Stand Movement. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2016**, *24*, 639–649. [CrossRef] [PubMed]
31. Allouche, B.; Saade, A.; Dequidt, A.; Vermeiren, L.; Remy-Neris, O. Design and Control of an Assistive Device for the Study of the Post-stroke Sit-to-Stand Movement. *J. Bionic Eng.* **2018**, *15*, 647–660. [CrossRef]
32. Cao, E.; Inoue, Y.; Liu, T.; Shibata, K. A Sit-to-Stand Training Robot and Its Performance Evaluation: Dynamic Analysis in Lower Limb Rehabilitation Activities. *J. Syst. Des. Dyn.* **2012**, *6*, 466–481. [CrossRef]
33. Liu, K.; Ye, M.; Yan, P.; Li, C. Design and control of mechanical structures for sit-to-rehabilitation training. In *Proceedings of the 2016 IEEE International Conference on Mechatronics and Automation, Harbin, China, 7–10 August 2016*; IEEE: Piscataway, NJ, USA, 2016; pp. 931–936. [CrossRef]
34. Shomin, M.; Forlizzi, J.; Hollis, R. Sit-to-stand assistance with a balancing mobile robot. In *Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015*; IEEE: Piscataway, NJ, USA, 2015; pp. 3795–3800. [CrossRef]
35. Li, J.; Zhang, F.; Wang, S.; Zhou, Y.; Cheng, X. A Multistage Hemiplegic Lower-Limb Rehabilitation Robot: Design and Gait Trajectory Planning. *Sensors* **2024**, *24*, 2310. [CrossRef]
36. Menga, G.; Ghirardi, M. Control of the Sit-to-Stand Transfer of a Biped Robotic Device for Postural Rehabilitation. *Robotics* **2019**, *8*, 91. [CrossRef]
37. Zhou, J.; Yang, S.; Xue, Q. Lower limb rehabilitation exoskeleton robot: A review. *Adv. Mech. Eng.* **2021**, *13*, 16878140211011862. [CrossRef]
38. Goh, C.H.; Anthony, J.; Ballard, C. Variable-Gravity RoboREHAB for Gait and Sit-to-Stand Rehabilitation: A System-Level Integration Study with Simulation and Benchtop Prototype Evidence. *Appl. Sci.* **2026**, *16*, 4336. [CrossRef]
39. Fattah, A.; Agrawal, S.; Catlin, G.; Hamnett, J. Design of a passive gravity balanced assistive device for sit-to-stand tasks. *J. Mech. Des.* **2006**, *128*, 1122. [CrossRef]

40. Hwang, D.; Choi, E.; Cho, K. Usability Test for an Over-Ground Walking Assistance Robotic Device Based on the Mecanum Wheel. *Appl. Sci.* **2025**, *15*, 5294. [CrossRef]
41. Holden, M.K.; Gill, K.M.; Magliozzi, M.R.; Nathan, J.; Piehl-Baker, L. Clinical gait assessment in the neurologically impaired: Reliability and meaningfulness. *Phys. Ther.* **1984**, *64*, 35–40. [CrossRef] [PubMed]
42. Ávila Chaurad, R.; León, R.L.P.; Muñoz, E.L.G. *Dimensiones Antropométricas de Población Latinoamericana*, 2nd ed.; Universidad de Guadalajara: Guadalajara, Mexico, 2007.
43. Centers for Disease Control and Prevention. *Growth Charts: CDC Data Files*; National Center for Health Statistics (NCHS): Hyattsville, MD, USA, 2026.
44. Rodríguez-Obregón, D.E.; Moreno, H.A.; Carrera, I.G.; Lopez-Lopez, L. A Decade-Long Systematic Review of EEG-Driven Gait-Intention Interfaces for Lower-Limb Rehabilitation and Prosthetics. *Appl. Mech. Mater.* **2025**, *929*, 107–118. [CrossRef]
45. French, B.; Thomas, C.H.; Coupe, J.; McMahon, N.E.; Connell, L.; Harrison, J.; Robinson, L.; Ackerley, S.J.; Watkins, C.L. Repetitive task training for improving functional ability after stroke. *Cochrane Database Syst. Rev.* **2016**, *2016*, CD006073. [CrossRef] [PubMed]
46. Pediatric Physical Therapy Exercises. Aprendiendo a Inclinarse Hacia Adelante al Sentarse: Ejercicios para Niño con Parálisis Cerebral #10. YouTube Video. 2020. Available online: https://youtu.be/3PffsYz-rA4 (accessed on 21 April 2026).
47. Bhardwaj, S.; Khan, A.A.; Muzammil, M. Lower limb rehabilitation using multimodal measurement of sit-to-stand and stand-to-sit task. *Disabil. Rehabil. Assist. Technol.* **2019**, *14*, 835–842. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

actuators

MDPI

Article

Validation of Pneumatic Actuation for Fast Fatigue Testing of Additive-Manufactured Polymers

Davide D'Andrea, Giacomo Risitano and Dario Santonocito *

Engineering Department, University of Messina, 98166 Messina, Italy; davide.dandrea@studenti.unime.it (D.D.); grisitano@unime.it (G.R.)
* Correspondence: dsantonocito@unime.it

Abstract

In the modern industrial context, many manufacturers design universal testing machines (UTMs) equipped with servo-hydraulic or electromechanical linear actuators, which offer excellent control capabilities and high-quality force signal measurement, at the expense of high costs due to the need for hydraulic power units or dedicated electrical networks. The complexity of these systems discourages manufacturers of mechanical components, especially the ones produced through additive manufacturing (AM), from investing in machines for the determination of mechanical properties according to international standards, settling instead for information derived from technical datasheets of the base material (filament or powders), which rarely include information about fatigue life. Within this context, the Fast Fatigue Machine (FFM), designed by KnoWow srl and ItalSigma srl, makes mechanical characterization of materials a process accessible to any organization that may require it. This was made possible by designing a pneumatic benchtop testing machine with a built-in setup for Thermographic Methods (TMs) usage. The aim of this work is to validate pneumatic actuators as a viable alternative to servo-hydraulic systems, demonstrating their effectiveness and reliability. Frequency analysis on both sinusoidal waveforms, root mean square error (RMSE) evaluation, and percentage total harmonic distortion (THD%) calculations showed that, while the servo-hydraulic system closely follows the load signal with a THD of around 5%, regardless of the applied load intensity, the pneumatic system exhibits higher distortion (THD of approximately 9%, strongly dependent on the load levels) and a high-frequency harmonic component, which, however, does not affect the overall results. Life cycle assessment (LCA) analysis confirmed the convenience of the pneumatic system and TMs in material testing and fatigue characterization.

Keywords: thermographic methods; tensile; fatigue; pneumatic; servo-hydraulic; additive manufacturing; LCA; sustainability

1. Introduction

Fast Fatigue Machine (FFM) is a pneumatic benchtop universal testing machine (UTM) developed in collaboration between KnoWow Srl, a University of Messina's academic spin-off, and ItalSigma srl, an Italian manufacturer of testing systems. The FFM can perform both quasi-static tensile tests and fatigue tests with a maximum force of 2.5 kN and a maximum frequency of 15 Hz.

The concept behind the development of the FFM is to make the material characterization a process accessible to companies and research institutions. The servo-hydraulic system's requirement for a hydraulic power unit, consisting of a pump, oil filter, and

control unit [1], can be expensive and unaffordable for many manufacturers. This is particularly critical as the need for material characterization is increasing due to the rise in additive manufacturing (AM) technologies. Whereas in the past, designers could rely on trustworthy technical datasheets reporting both static and, where available, fatigue design parameters, the presence of numerous AM materials, whose mechanical properties are highly dependent on the manufacturing process parameters [2], poses a challenge in rapidly obtaining such information. While static material characterization is generally straightforward, fatigue characterization represents an open design challenge since it is an energy-, material-, and time-consuming process [3,4]. According to ASTM and ISO standards, stress-number of cycles to failure (SN) curve in high-cycle fatigue (HCF) regime should be determined by examining a large set of specimens tested under several loading conditions, due to the probabilistic nature of the fatigue problem.

Many researchers proposed their own approach to rapidly assess the fatigue life of metals, polymers and composites as alternatives to standard methodologies. For thin aluminum films adopted in the fabrication of MEMS devices, Bae et al. [5] developed a very compact HCF testing system employing electromagnetic shakers with a high-frequency response and 10 N load capacity capable of retrieving the SN curve on aluminum films in a short period of time. For the same application, Kim et al. [6] developed an electrodynamic actuator capable of performing tests with high loading frequencies, with a real-time measurement of the deformation. This kind of study also can be conducted by adopting image analysis, for example, during low-cycle fatigue tests, allowing for the study of the strain field in mechanical components, like butt-welded joints, by adopting a digital camera [7] instead of strain gauge. Under uniaxial and multiaxial random loading conditions on metals, several pieces of testing equipment can be adopted, like servo-hydraulic and shaker tables, considering several specimens' geometries [8]. Ultrasonic testing systems can be adopted to rapidly assess the fatigue life of metals under several loading conditions [9]; however, some issues may arise, like excessive heating during the fatigue test, the design of specimens and different fracture surfaces compared to conventional servo-hydraulic testing systems [10]. To improve the fatigue testing of AM materials, Nicoletto [11] proposed the adoption of the Mini FP2 machine, a compact testing machine which allows for the testing of small, notched specimens under bending conditions.

An approach for the rapid assessment of the fatigue properties of materials is represented by Thermographic Methods (TMs) introduced by Risitano and his School. Since fatigue is a dissipative phenomenon, it is possible to measure with an infrared camera (IR) the temperature evolution of the material subjected to fatigue test conditions.

Risitano's Thermographic Method (RTM) [12] is a well-established methodology to obtain insight into the fatigue life of materials from a temperature's trend measured on a specimen loaded with a stepwise increase in the applied stress level [13]. A noticeable increase in temperature occurs just for stress levels above the fatigue limit of the material under study. The RTM allows for the assessment of the fatigue limit and SN curve of the material testing a reduced number of specimens and in a short time.

In 2013, Risitano and Risitano introduced the Static Thermographic Method (STM) to derive, with a quasi-static tensile test, the first damage within the material. This macroscopic stress value can be related, if applied in a cyclic way, with the fatigue limit of the material. More information and comparison with RTM and a conventional fatigue test can be found in [14,15].

In 2022, the FFM was patented and designed to facilitate the testing setup to apply TMs by inserting an infrared (IR) camera and a video camera for digital image correlation (DIC) analysis. It was chosen to use a pneumatic actuator to simplify the working fluid

circuit, since it is enough to use a compressor or a compressed air distribution system that is easy to find in most companies.

Pneumatic actuation represents a simplified and cleaner alternative to hydraulic systems, employing gases, typically air, instead of liquids as the working fluid. Because of the lower density, and consequently higher compressibility of air, pneumatic systems exhibit reduced stiffness and greater compliance. Although pneumatic cylinders are extensively used in industrial automation, such systems have traditionally been challenging to control. However, the growing interest in compliant and soft robotic systems has recently prompted the research community to reconsider the potential of pneumatic actuation for more advanced and complex applications. While the pneumatic cylinder remains the most common industrial actuator, a wide range of alternative devices now exploit the power of compressed air for actuation [16]. The simplicity of linear pneumatic actuators allows for their employment in most applications, and to manufacture them by AM process as it was performed in [17]. Although servo-hydraulic actuators are widely employed in material testing applications [18], the use of pneumatic actuators has remained limited due to the inherent compressibility of air, which can influence the accuracy and repeatability of experimental results.

The present study aims to assess and validate pneumatic actuators as a viable alternative to conventional servo-hydraulic systems by comparing the performance of both actuation technologies under quasi-static and fatigue loading conditions. The findings are supported by a gate-to-gate life cycle assessment (LCA), conducted to demonstrate the relative advantages of the two systems in terms of cost-effectiveness and environmental sustainability.

LCA is an analytical tool that captures the overall environmental impacts of a product, process or human activity from raw material acquisition, through the production and use phases, to waste management. This comprehensive view makes LCA unique in the suite of environmental management tools available to decision makers [19]. The LCA analysis highlighted the environmental advantages of employing a pneumatic actuation system, which become even more significant when TMs are applied.

2. Materials and Methods

Two experimental setups have been used for comparisons. The first one consists of a FPF25, a traditional servo-hydraulic UTM produced by ItalSigma srl (Forlì, Italy) which can perform both quasi-static and fatigue tests, exploiting a maximum force of 25 kN in tension-compression, with a maximum frequency reached of 20 Hz. The second one consists of an FFM, the patented prototype which has been described in the introduction.

Due to machines' different dimensions, two geometries of specimens, both realized in polyammide-12 (PA12) manufactured by HP MultiJet Fusion Process [20] have been used to verify the independence of results on load waveform. The ASTM D638 type I specimens were used to evaluate performance on FPF25 (Figure 1a), while specimens designed according to Avanzini et al. [21] were used to test FFM's performance (Figure 1b). In fact, producing smaller test specimens, especially when testing materials obtained through additive manufacturing, represents an advantage for companies. Specimens' geometries are reported in Figure 1c,d, in which units are expressed in millimeters. Both specimens' sets were manufactured in XY direction, which was demonstrated to be the one characterized by better fatigue behavior [22].

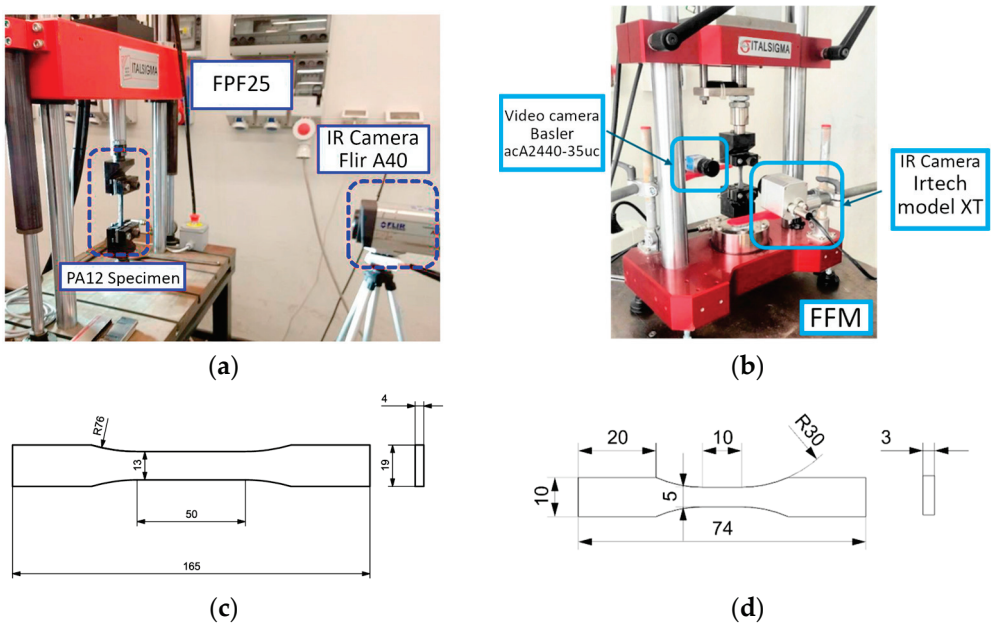


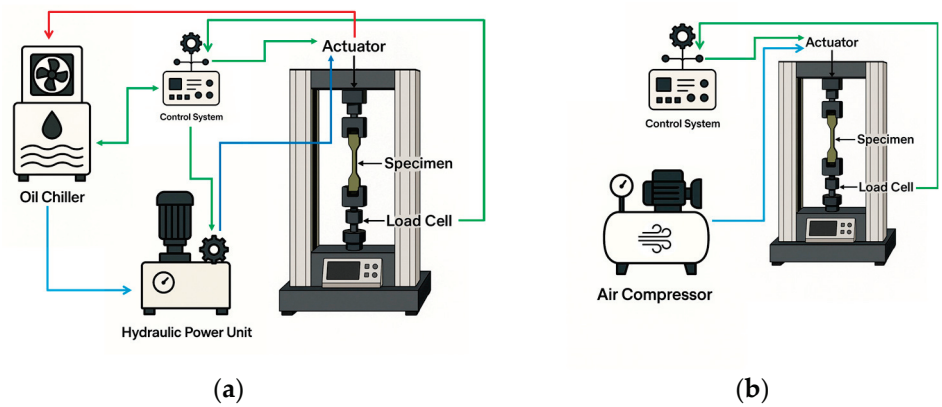
Figure 1. (**a**) FPF25 setup; (**b**) FFM setup; (**c**) ASTM D638—Type I PA12 MJF specimen; (**d**) Smooth [21] PA12 MJF Specimen; Units expressed in millimeters.

Surface temperature measurements were performed using a FLIR A40 (Wilsonville, OR, USA) thermal camera for the tests carried out on the FPF25, and an IRtech (Milan, Italy) XT model thermal imaging camera for the tests performed on the FFM. The FLIR A40 is equipped with an FPA sensor offering a resolution of 320 × 240 pixels and a thermal sensitivity of 80 mK. The IRtech XT features an FPA detector with an optical resolution of 382 × 288 pixels, a thermal sensitivity of 80 mK, and operates at a frame rate of 27 Hz. Both cameras exhibit similar technical specifications, ensuring consistency and reliability in the thermal measurements. Temperature fields associated with quasi-static mechanical tests were acquired using a 10 fps framerate, while those related to fatigue tests were recorded by capturing one frame every 15 s.

Using oil as the working fluid requires a dedicated apparatus designed to reach and maintain the desired pressure, ensuring that its properties remain stable under operating conditions. For this reason, in addition to the servo-hydraulic actuator, a hydraulic power unit and a cooling system are also necessary, as it is shown in representative diagram in Figure 2a. In Figure 2b, the FFM's system representative diagram consisting of the tensile machine mounting the pneumatic actuator, an air compressor and the control system, is shown. In both diagrams, the analog-digital connections and the information required by the control system (such as transducer measurements, oil temperature, and pressure settings) are represented by green arrows, which also represent the feedback signals used in the closed-loop calibration process. These signals are continuously transmitted to the control system, which compares them with reference values and adjusts the actuator command accordingly.

The pressurized oil flow is indicated in blue. The hydraulic fluid used by the actuator is directed to the chiller (red arrow), where it is cooled before being recirculated into the pump (cyan arrow) to maintain the oil within an optimal temperature range, preventing viscosity changes that could affect dynamic response, control accuracy, and overall system efficiency.

Pneumatic actuator just requires pressured air to work, indicated by cyan arrow, and the air is discharged into the atmosphere without causing any safety or environmental problems.



(**a**) (**b**)

Figure 2. (**a**) Representative diagram of the FPF25 (servo-hydraulic actuator); (**b**) Representative diagram of the FFM (pneumatic actuator).

After both systems have been described, LCA analysis can be carried out. A gate-to-gate approach was adopted [23], as the present study focuses solely on the usage phase of the testing systems within the material characterization process. In this context, the gates considered correspond to the entry to and exit from the laboratory. Consequently, this phase does not include analyses related to raw materials, transportation, manufacturing of the machines and devices, or end-of-life processes. The discussion is centered on four parameters, selected as the environmental impact indicator, namely: Climate Change, expressed in kilograms of CO_2 equivalent (kg CO_2 eq.) and representing the potential contribution of greenhouse gas emissions to global warming; Ozone Depletion, expressed in kilograms of CFC-11 equivalent (kg CFC-11 eq.) and describing the potential of emitted substances to degrade the stratospheric ozone layer; Particulate Matter Formation, quantified in terms of disease incidence and reflecting the health impacts associated with the formation of fine airborne particles; and Water Use, reported as cubic meters of world-equivalent water deprived (m^3 world-eq deprived) and indicating the potential freshwater deprivation caused by water consumption [24–26]. These indicators allow for a comprehensive assessment of the potential environmental burdens associated with the system under study, capturing impacts on atmospheric emissions, human health, and resource consumption. The calculation of these parameters was carried out using OpenLCA software (version 2.3.0).

For the FPF25, the total absorbed power amounts to 14.7 kW, resulting from the combined consumption of the cooling system (7.2 kW) and the hydraulic power unit (7.5 kW).

For the FFM, a total power demand of 1.47 kW was considered, entirely attributable to the air compressor.

Specimens' manufacturing process, based on the Multijet Fusion technology developed by HP [20], requires an energy consumption of 0.05 kWh for the powder-grinding stage and 27.4 kWh for the part-printing stage. The resulting values were calculated for 1 kg of material and subsequently normalized to the mass and number of specimens used in both configurations.

Quasi-static monotonic tensile tests have been performed in displacement control with a crosshead velocity of 2.4 mm/min, chosen after preliminary tests carried out to determine the best velocity to clearly observe the cooling phase in the surface temperature's trend [22] during quasi-static tensile tests. By doing so, a displacement ramp was used as a reference. Percentage differences between the measured and the reference displacement (disp) has been evaluated as reported in Equation (1).

$$\mathrm{diff}_{\%} = \frac{\mathrm{disp}_{\mathrm{reference}} - \mathrm{disp}_{\mathrm{measured}}}{\mathrm{disp}_{\mathrm{reference}}} \quad (1)$$

Fatigue tests were carried out by applying an increasing load to a single specimen, acquiring five complete periods characterized by a frequency of 1 Hz, to avoid excessive self-heating effect in polymers. Regarding the damping properties of PA12 under fatigue loading condition, it exhibits a linear behavior in the first loading cycles; then, when the material's temperature reaches the glass transition value, the energy loss, as highlighted by the hysteresis loop, is more marked. This will lead to a decrease in the material stiffness. On the other hand, when the glass transition temperature is not reached, no stiffness reduction occurs [27]. Since with the FFM, we can monitor the material's temperature during the fatigue tests, the frequency has been chosen to avoid the glass transition temperature of the material, maintaining a constant stiffness during the cycles.

Seven increasing stress levels have been applied to each specimen type on both testing machines according to what is reported in Table 1 to apply RTM [22]. Fatigue loads were characterized by a load ratio equal to 0.1 (R = 0.1). PID parameters were preliminarily tuned by manually changing the proportional, integral and derivative factors in order to ensure that the measured force respects maximum and minimum reference values.

Table 1. Maximum stress and corresponding maximum force for FPF25 and FFM's setups.

Testing Machine	**FPF25**	**FFM**
Maximum Stress [MPa]	**Maximum Force [N]**	
21	1092	315
23	1196	345
25	1300	375
27	1404	405
29	1508	435
31	1612	465
33	1716	495

The signal used for harmonic analysis was obtained from the load cell of the testing machines and sampled at 100 Hz for 5 s at the end of each load step.

To increase the frequency resolution (Δf) and enable the discrimination of closely spaced harmonics, the acquired signal was repeated ten times prior to the spectral analysis as indicated in Equation (2), in which f_s is the sampling frequency and N is the total samples number. At a fixed sampling frequency, the frequency resolution Δf improves as the number of available samples increases. Repeating the signal effectively increases the total number of samples, thereby reducing Δf and allowing for a more precise representation of the spectral components.

$$\Delta f = \frac{f_s}{N} \tag{2}$$

The transitions between consecutive 5 s segments were smoothed by applying a linear tapering at the segment boundaries. Specifically, the amplitude of the last few samples of each segment was gradually reduced to match the beginning of the next segment, creating a continuous signal without interruptions [28].

Fast Fourier transform (FFT) analysis was performed to evaluate the harmonic content of the fatigue load applied by the testing machine.

The amplitude of each harmonic component was estimated by applying the FFT algorithm to the time-domain signal, extracting the magnitude of the corresponding spectral coefficient and normalizing it by the total number of samples to obtain its true amplitude [29,30] as reported in Equation (3):

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi \frac{kn}{N}} \tag{3}$$

where x[n] is the signal value in time domain at sample index n and X[K] is the complex-valued spectral coefficient at frequency index k; n is the index in discrete time and is between 0 and N−1, as well as k, which is the index of discrete frequency.

The FFT of a real-valued signal returns both positive and negative frequency components. Therefore, the amplitude spectrum is computed as a single-sided spectrum by taking the absolute value of the coefficients and doubling the amplitudes of all positive frequency components, except for the zero-frequency component and, if present, the Nyquist frequency. This procedure ensures that the amplitudes of the harmonic components correspond to their actual peak values in the time-domain signal.

Each harmonic amplitude was then compared to that of the fundamental harmonic, thus quantifying the distortion and the relative contribution of higher-order harmonics to the overall load waveform. The performance of the actuators was evaluated by calculating the root mean square error (RMSE) and the total harmonic distortion (THD%) [31,32], respectively, using Equations (4) and (5).

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2} \tag{4}$$

where x_i is the measured value, $\hat{x}_i$ is the reference value and n is the number of observations.

$$\mathrm{THD\%} = \frac{\sqrt{\sum_{h=2}^{\infty} A_h^2}}{A_1} \cdot 100 \tag{5}$$

where A_1 is the fundamental harmonic amplitude and A_h is the h-th harmonic amplitude.

Both metrics provide insight into the quality of the actuator's output waveform. RMSE quantifies the deviation of the measured signal from its reference signal, with values approaching zero indicating a close correspondence between the two. THD%, on the other hand, characterizes waveform distortion by expressing the ratio between the root of the sum of the squared amplitudes of all higher-order harmonics (starting from the second) and the amplitude of the fundamental harmonic. Highly distorted signals exhibit large THD% values, whereas an ideal sinusoidal waveform yields a THD% equal to zero [31]. It is worth noting that RMSE alone provides limited information on waveform quality, as signals exhibiting significant harmonic distortion may still produce low RMSE values with respect to the reference signal.

3. Results and Discussion

The performance of the actuators during the quasi-static tests was evaluated by calculating the percentage differences between the measured displacement and the reference ramp. In Figure 3a, it can be observed that FPF25 servo-hydraulic actuator's crosshead displacement closely follows the reference ramp, as evidenced by the small percentage differences between the measured and reference displacements [33].

In the first graph, on the left, the force–displacement relationship is presented. The central graph shows that the measured displacement, represented by the continuous blue line, accurately tracks the reference ramp, indicated by the orange dashed line. In the last graph, it can be observed that the instantaneous measured value deviates by approximately 4% from the reference.

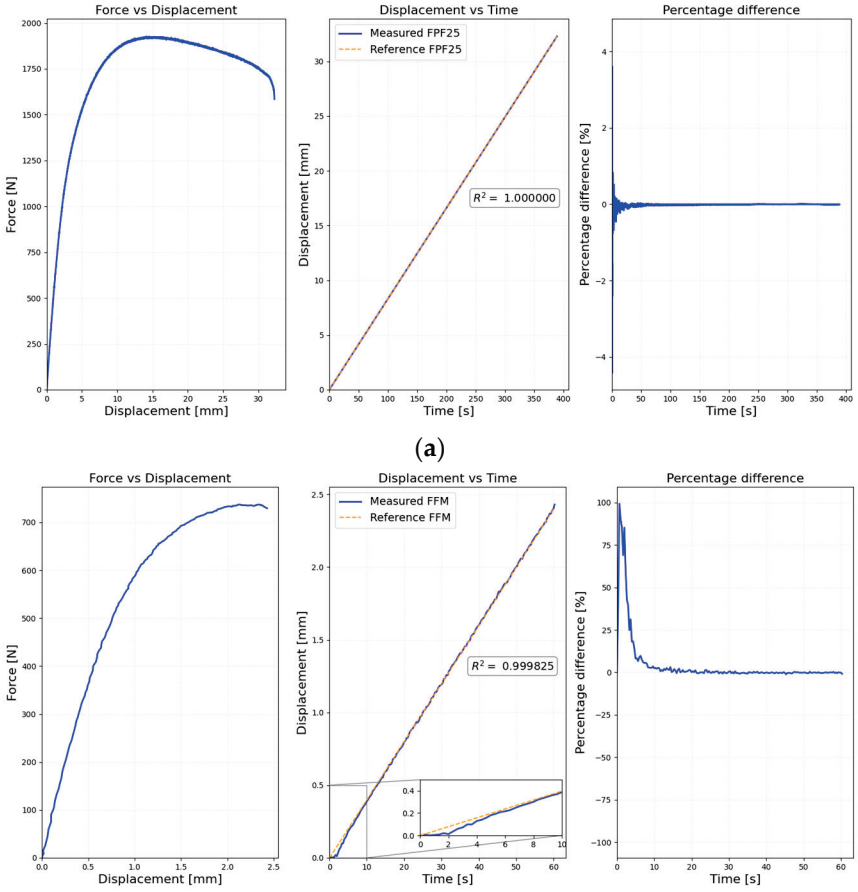


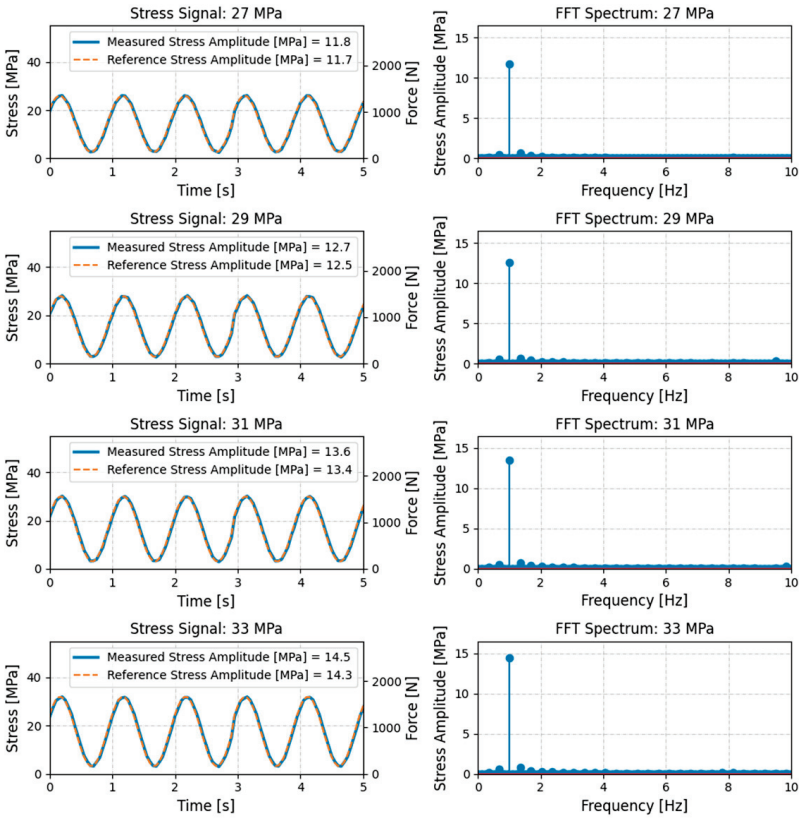
Figure 3. (**a**) FPF25 performance during monotonic quasi-static tensile tests; (**b**) FFM performance during monotonic quasi-static tensile tests.

In Figure 3b, the curves obtained from the pneumatic actuator mounted on the FFM are presented in the same order as those of the FPF25. In the first graph on the left, the force–displacement trend exhibits a behavior similar to that of the servo-hydraulic actuator, except that no load reduction phase is observed, and failure occurs directly. In the central graph, it can be observed that the pneumatic actuator requires a longer time to reach equilibrium conditions, as indicated by the deviation between the measured displacement and the reference ramp at the beginning of the test [17] evidenced by the inset, which shows a magnification of 60% in width and 15% in height over the initial 10 s. This behavior is also evident in the last graph, where percentage differences of approximately 100% are calculated in this phase and last longer if compared to servo-hydraulics measured displacement. Such deviations are attributed to the compressibility of air [34,35], which also influences the resulting stress–strain curve. The ASTM D638 standard [36], which provides guidelines for quasi-static tensile testing of polymers, states that an initial settling phase may occur and suggests resolving it by simply extending the actual linear portion of the curve until it intersects the abscissa axis. It is also important to note that, due to the different specimen dimensions, the measured displacement at failure is higher for FPF25 specimens, as it can be observed in y-axis of central graph (Figure 3a), while the strain values obtained with both actuators remain consistent. Finally, the coefficient of determination, reported in both displacement vs. time graphs, approaching unity indicates that the measured displacement closely follows the reference ramp profile, despite the initial response delay.

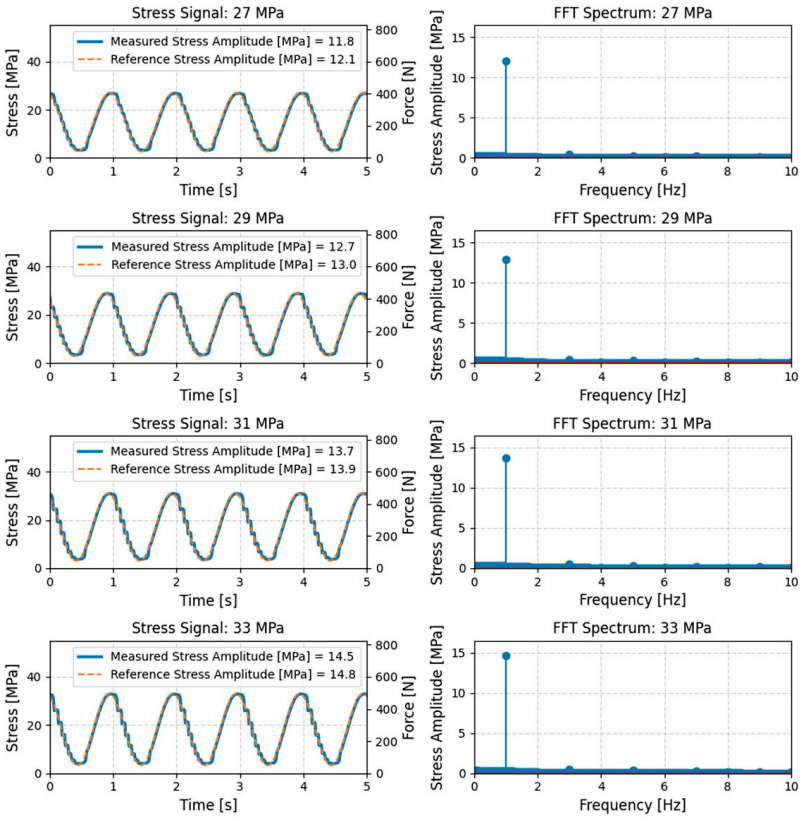
The analysis began with the evaluation of the cyclic stress waveforms corresponding to each load level, applied to both specimen types using the FPF25 and FFM testing machines, according to the procedures detailed in Section 2.

FFT analysis was performed to assess whether the servo-hydraulic and pneumatic actuators could reproduce stress levels close to the imposed values [33,34]. During the PID control parameter setup, a preliminary visual inspection of the measured load trends highlighted clear differences between the two actuation systems [35]: while the servo-hydraulic actuator was able to deliver an approximately sinusoidal waveform within the prescribed upper and lower stress limits, the pneumatic actuator inevitably generated a distorted waveform. Nevertheless, priority was given to ensuring that the imposed maximum and minimum force values were effectively reached. The optimal P, I, D parameters combination resulted to be P = 0.23, I = 0.02 and D = 0. On the left side of Figure 4a, the last four sinusoidal load signals corresponding to the stress amplitude levels defined in the stepwise fatigue test conducted on a PA12 type I specimen with the FPF25 servo-hydraulic actuator are reported. The measured stress amplitude, calculated as half of the difference between the maximum and minimum stress values, was found to be consistent with the reference value, as no significant deviations were detected. On the right side of Figure 4a, the results of the FFT analysis for the same load levels are presented. In agreement with the high quality of the load waveform, as qualitatively observed from the time-domain graphs, no significant harmonic content was detected in the frequency domain. Accordingly, the calculated first harmonic amplitude was approximately equal to the stress amplitude previously determined. On the other hand, the pneumatic actuator produced a poorer cyclic load waveform, as previously discussed. Figure 4b presents the distorted load signals measured for the last four maximum stress levels. While no distortion was observed during the tensile phase of each cycle, the distortion was concentrated in the relaxation phase, where the load approached its minimum value. The distortion observed in the relaxation phase is attributable to the air properties and to the enforcement of minimum and maximum force constraints. Without PID tuning, the force signal during this phase would remain undistorted, but it would fail to track the reference values. Therefore, the distortion results from a controlled relaxation phase, which effectively manifests as a step-like relaxation. This phenomenon was confirmed by the FFT analysis, which revealed the presence of harmonic content in the frequency domain. Although the corresponding amplitudes were relatively small, they nonetheless affected the signal quality in the time domain. On the right side of Figure 4b, the harmonic amplitudes associated with frequencies up to 10 Hz are reported, as higher frequencies did not exhibit significant amplitudes. However, this does not imply that higher frequencies can be neglected, as they still contribute to the THD%. For this reason, the next discussions will focus on just the fundamental, the third and the fifth harmonics characteristics.

Figure 5 reports the trends of the first (a), third (b), and fifth (c) harmonics as a function of the applied maximum stress. With regard to the first harmonic amplitude, it can be observed that the FPF25 and FFM provide similar results. Particular attention, however, should be paid to the lower load levels: despite the correct selection of control parameters, the pneumatic actuator applied a load higher than the reference value. This behavior can be addressed to the smaller force increment required to reach the target stress and to the pneumatic actuator's highly nonlinear behavior due to factors like air compressibility and friction force [37]. Specifically, the force increment corresponding to a stress step for the FFM and its specimen is 30 N, whereas for the FPF25 it is 104 N.



(**a**)



(**b**)

Figure 4. FFT analysis results for FPF25 (**a**) and FFM (**b**).

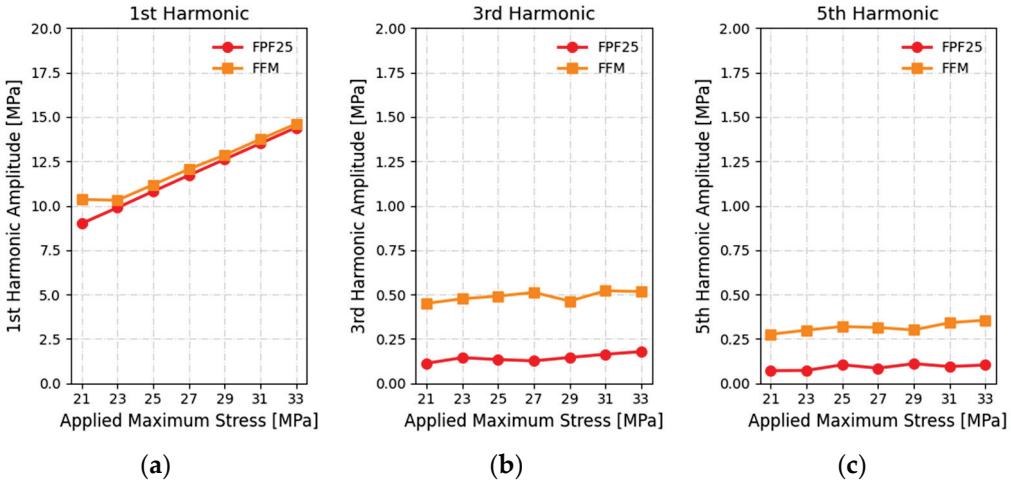


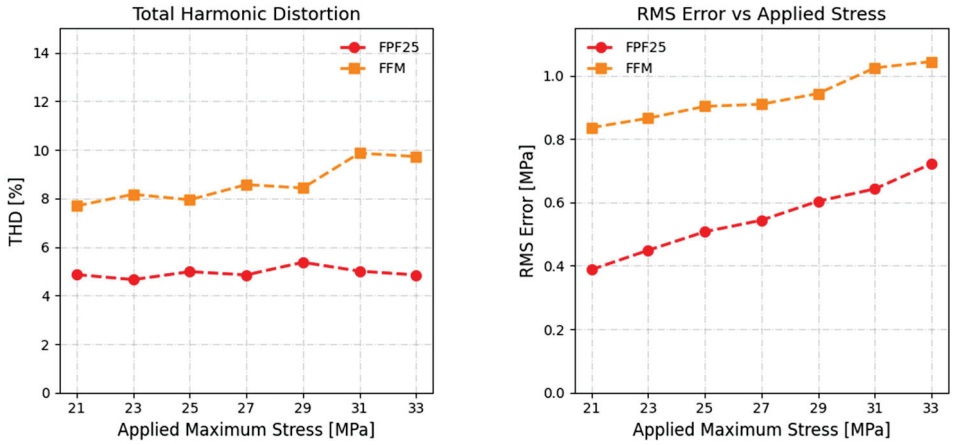
Figure 5. (**a**) First; (**b**) Third; (**c**) Fifth harmonic amplitude versus applied maximum stress for FPF25 and FFM.

Regarding the third harmonic amplitude associated with each stress level, it can be observed that the stress signals measured with the FFM are characterized by higher third harmonic amplitudes, with a mean value of 0.49 ± 0.03 MPa, which is significantly greater than the mean value obtained with the FPF25 (0.14 ± 0.02 MPa). Nevertheless, the third harmonic amplitudes remain well below the fatigue limit of PA12, estimated to be equal to 26.2 MPa [22], and therefore do not represent a concern for the fatigue characterization. The same consideration applies to the fifth harmonic amplitude, as reported in the final graph, where a mean value of 0.32 ± 0.03 MPa was measured for the FFM and 0.09 ± 0.02 MPa for the FPF25. Detailed results are reported in Table 2, together with their average values and standard deviations.

Table 2. First, third and fifth harmonics amplitude's values.

	First Harmonic Amplitude [MPa]		Third Harmonic Amplitude		Fifth Harmonic Amplitude	
Maximum Stress [MPa]	**FPF25**	**FFM**	**FPF25**	**FFM**	**FPF25**	**FFM**
21	9.00	10.35	0.11	0.45	0.07	0.28
23	9.89	10.30	0.14	0.48	0.07	0.3
25	10.81	11.18	0.13	0.49	0.10	0.32
27	11.71	12.06	0.13	0.51	0.09	0.31
29	12.61	12.85	0.15	0.46	0.11	0.30
31	13.51	13.75	0.16	0.52	0.09	0.34
33	14.40	14.61	0.18	0.52	0.10	0.36
Average value	11.70	12.16	0.14	0.49	0.09	0.32
Standard deviation	1.95	1.67	0.02	0.03	0.02	0.03

Focusing on the THD% (Figure 6a), which accounts for the entire harmonic content except for the fundamental harmonic, it can be observed that the FPF25 exhibits an average THD% of 4.94 ± 0.22% across all load steps, while the FFM shows a higher value of 8.63 ± 0.85%. This difference is attributed to the characteristics of the working fluid. It is also noteworthy that the THD% of the FFM increases consistently with the applied maximum stress. In the RMSE graph (Figure 6b), it can be observed how increases with applied stress in both cases, highlighting how RMSE alone cannot entirely represent the actuator's performance, since it encloses both amplitude and phase differences between measured and reference signals. An RMSE of 0.551 ± 0.115 MPa was obtained for FPF25, and 0.932 ± 0.078 MPa for FFM.



(**a**) (**b**)

Figure 6. (**a**) Percentage of total harmonic distortion for FPF25 (red circles) and FFM (orange squares); (**b**) Root mean square error for FPF25 (red circles) and FFM (orange squares).

Detailed results are enlisted in Table 3, together with their mean values and standard deviation for both testing machines.

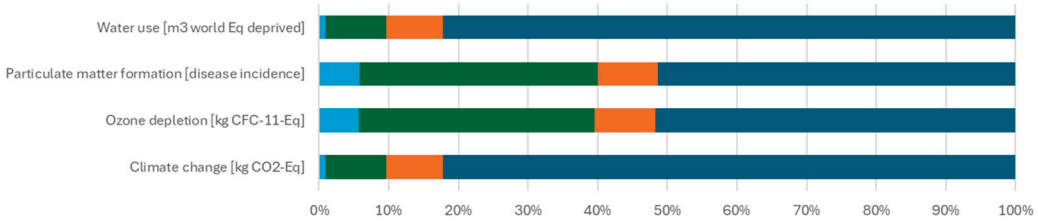
Table 3. THD% and RMSE results for FPF25 and FFM.

	Total Harmonic Distortion (THD%) [%]		**Root Mean Square Error (RMSE) [MPa]**	
Maximum Stress [MPa]	**FPF25**	**FFM**	**FPF25**	**FFM**
21	4.86	7.69	0.388	0.836
23	4.65	8.17	0.449	0.865
25	4.98	7.94	0.507	0.903
27	4.85	8.57	0.544	0.910
29	5.37	8.43	0.604	0.944
31	5.00	9.87	0.643	1.024
33	4.85	9.72	0.722	1.044
Average value	4.94	8.63	0.551	0.932
Standard deviation	0.22	0.85	0.115	0.078

The LCA analysis, carried out using Italian low-voltage electricity mix, yielded significant results, as the simplicity of the pneumatic system had a clear effect on the environmental impact indicator, Climate Change, expressed in kilograms of equivalent CO_2 gas emissions (kg CO_2 eq.). The comparison showed that, for an equal operating time of 795 h, corresponding to the conventional fatigue testing campaign PA12 MJF specimens carried out by D'Andrea et al. [22] investigating nine stress levels, each represented by a dedicated specimen, even though additional constant-amplitude tests may be required, the servo-hydraulic actuation system resulted in a climate change impact of 4160 kg CO_2 eq, whereas the FFM system, equipped with a pneumatic actuator, was characterized by an impact of only 416 kg CO_2 eq. Regarding Ozone Depletion, the analysis confirmed the same trend: the servo-hydraulic system exhibited a substantially higher impact compared to the pneumatic solution, respectively, equal to 0.00441 and 0.00073 kg CFC-11 eq. Similar behavior was observed for Water Use. The FFM system, owing to its intrinsically lower power consumption, required significantly fewer water-related resources throughout the testing cycle, whereas the servo-hydraulic system showed a markedly higher water deprivation footprint, respectively, 2540 and 254 m^3 world-eq deprived.

Finally, the indicator associated with Particulate Matter Formation, expressed in terms of disease incidence, further highlighted the environmental benefits of the pneumatic actuator leading to a considerably smaller impact on human health (0.31 disease incidence) if compared with the servo-hydraulic configuration (1.88 disease incidence). These results are primarily associated with the higher energy demand of the servo-hydraulic actuator, which amplifies the contribution of electricity-related emissions and impacts on environment and human health.

The sustainability advantage of the pneumatic actuation system, already evident under standard fatigue testing conditions, becomes even more pronounced when Thermographic Methods are employed, as these significantly reduce the total testing time to 84 h and the number of tested specimens to 6. In Figure 7 results discussed above are reported expressed as the percentage of the highest LCA parameters, corresponding to the ones associated with conventional fatigue characterization carried out by FPF25. The FFM approach, combined with TMs, demonstrated to be the most sustainable solution for both quasi-static and fatigue characterization of polymeric materials. Detailed results are reported in the table below.



	Climate change [kg CO2-Eq]	Ozone depletion [kg CFC-11-Eq]	Particulate matter formation [disease incidence]	Water use [m3 world Eq deprived]
FFM - TM	44.2	4.85×10^{-4}	0.21	26.94
FPF25 - TM	441.4	2.89×10^{-3}	1.25	269.42
FFM - Conventional	415.7	7.36×10^{-4}	0.32	254.22
FPF25 - Conventional	4155.2	4.41×10^{-3}	1.88	2542.22

Figure 7. Gate-to-gate LCA comparison between FPF25 and FFM testing systems on PA12 fatigue campaign.

4. Conclusions

Results demonstrated how servo-hydraulic and pneumatic actuators perform when used in material testing. Quasi-static mechanical tests, carried out to determine the technical properties of the materials, exhibited similar behavior. The test performed using a servo-hydraulic system under displacement control produced a typical force–displacement relationship for polymer materials. The system was characterized by a fast and accurate response and was able to follow the reference displacement ramp with a maximum deviation of 4% in the initial stage, in which both the specimen and the testing machine reached equilibrium related to the material load distribution, the control system stabilization, and the achievement of the desired oil pressure. Optimal results were obtained in terms of dynamic performance, as the measured fatigue load waveform was approximately sinusoidal, and the FFT analysis did not reveal any significant harmonic distortion and amplitudes. This is confirmed by the THD% analysis, which yielded a value of $4.94 \pm 0.22\%$, remaining constant with increasing reference maximum stress. Conversely, the RMSE increased with the applied load, indicating linearly growing deviations between the measured and reference forces, because of combined distortion and negligible time delays. The complexity of FPF25's testing system, which consists of the testing machine itself, the controller, the pressure station, and the cooling unit ensures high performance and allows for the performance of tensile and compression tests in every condition. On the other hand, the pneumatic actuator mounted in the FFM proved to be a valid alternative to hydraulic actuation. Despite the simplicity of its system, it provided performance levels adequate

for the intended application. From the quasi-static point of view, the only issue observed was a delay in response to the reference displacement ramp, which resulted in a percentage difference of approximately 100% between the measured and reference signals. However, this deviation can be corrected in accordance with the ASTM D638 standard.

Fatigue load waveforms exhibited a low harmonic content dependent on the applied stress, as indicated by the FFT analysis, which provided information on the harmonic composition, described by the higher THD% values and the amplitudes of harmonics of orders greater than the fundamental. While the THD% increased with the applied load, reaching values of approximately 9% at the highest stress levels, the amplitudes of harmonics higher than the first remained much smaller than both the fundamental component and the material's fatigue limit, confirming that preferring to reach maximum and minimum reference values maintaining the desired load ratio is more important than avoiding distortion in signal. RMSE calculation resulted in higher values than those calculated on FPF25, since it takes into account higher distortion and phase's delay between measured and reference signals.

Both actuation systems produced fundamental amplitudes consistent with the imposed sinusoidal reference load signal, demonstrating that a pneumatic actuation system is suitable for the testing of polymer materials. With an appropriate air treatment setup, namely, a compressor with a higher volumetric flow rate, it would be possible to reach significantly higher frequencies while maintaining the same performance characteristics described in the present work.

The testing of polymer materials is particularly critical for this type of actuation, due to the high and prolonged elastic recovery typical of plastics.

While an applicable force of 2.5 kN is sufficient to test most polymers and to use specimens compliant with standard geometries, this is not feasible for metallic materials, whose yield and ultimate strengths are significantly higher. Consequently, performing static and fatigue characterization on metals would require specimens with very small cross-sectional areas, falling outside the standard requirements. For this reason, future design developments will focus on the use of a pneumatic actuator capable of applying higher forces.

The FFM proved to be a valid alternative to conventional testing machines equipped with servo-hydraulic actuation systems, whose purchase and operation still represent a considerable burden for companies and research institutions requiring both static and fatigue characterization of their materials. While operators must accept some compromises in terms of a response delay during static tests and signal distortion under dynamic loading, the advantages offered by the FFM system in terms of sustainability and operational simplicity are substantial. The advantages in terms of sustainability become even more significant when considering the use of Thermographic Methods, which can reduce the total testing time required for complete material characterization from 795 to 84 h, and consequently lower the energy consumption associated with the characterization process.

Author Contributions: Conceptualization, D.S. and G.R.; methodology, D.D. and D.S.; software, D.D.; validation, D.S. and G.R.; formal analysis, D.D.; investigation, D.D. and D.S.; resources, D.S. and G.R.; data curation, D.D. and D.S.; writing—original draft preparation, D.D.; writing—review and editing, D.S.; visualization, D.D.; supervision, D.S.; project administration, D.S. and G.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Dataset available on request from the authors.

Acknowledgments: The authors would thank Italsigma Srl for its support.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Hossein Sharghi, A.; Karami Mohammadi, R.; Farrokh, M.; Zolfagharysaravi, S. Feed-Forward Controlling of Servo-Hydraulic Actuators Utilizing a Least-Squares Support-Vector Machine. *Actuators* **2020**, *9*, 11. [CrossRef]
2. Nandhakumar, R.; Venkatesan, K. A Process Parameters Review on Selective Laser Melting-Based Additive Manufacturing of Single and Multi-Material: Microstructure, Physical Properties, Tribological, and Surface Roughness. *Mater. Today Commun.* **2023**, *35*, 105538. [CrossRef]
3. Moghanlou, M.R.; Azizian-Farsani, E.; Mahmoudi, A.; Khonsari, M.M. Accelerated Fatigue Characterization of Additively Manufactured Continuous Carbon Fiber Reinforced Thermoplastic: A Thermodynamic Approach. *Compos. Part A Appl. Sci. Manuf.* **2025**, *192*, 108805. [CrossRef]
4. Zhang, L.; Jiang, B.; Zhang, P.; Yan, H.; Xu, X.; Liu, R.; Tang, J.; Ren, C. Methods for Fatigue-Life Estimation: A Review of the Current Status and Future Trends. *Nanotechnol. Precis. Eng.* **2023**, *6*, 025001. [CrossRef]
5. Bae, J.S.; Oh, C.S.; Park, K.S.; Kim, S.K.; Lee, H.J. Development of a High Cycle Fatigue Testing System and Its Application to Thin Aluminum Film. *Eng. Fract. Mech.* **2008**, *75*, 4958–4964. [CrossRef]
6. Kim, C.Y.; Song, J.H.; Lee, D.Y. Development of a Fatigue Testing System for Thin Films. *Int. J. Fatigue* **2009**, *31*, 736–742. [CrossRef]
7. Tateishi, K.; Hanji, T. Low Cycle Fatigue Strength of Butt-Welded Steel Joint by Means of New Testing System with Image Technique. *Int. J. Fatigue* **2004**, *26*, 1349–1356. [CrossRef]
8. Marques, J.M.E.; Benasciutti, D.; Niesłony, A.; Slavič, J. An Overview of Fatigue Testing Systems for Metals under Uniaxial and Multiaxial Random Loadings. *Metals* **2021**, *11*, 447. [CrossRef]
9. Nikitin, A.; Bathias, C.; Palin-Luc, T. A New Piezoelectric Fatigue Testing Machine in Pure Torsion for Ultrasonic Gigacycle Fatigue Tests: Application to Forged and Extruded Titanium Alloys. *Fatigue Fract. Eng. Mater. Struct.* **2015**, *38*, 1294–1304. [CrossRef]
10. Peng, W.; Zhang, Y.; Qiu, B.; Xue, H. A Brief Review of the Application and Problems in Ultrasonic Fatigue Testing. *AASRI Procedia* **2012**, *2*, 127–133. [CrossRef]
11. Nicoletto, G. A Novel Test Method for the Fatigue Characterization of Metal Powder Bed Fused Alloys. *Procedia Struct. Integr.* **2017**, *7*, 67–74. [CrossRef]
12. La Rosa, G.; Risitano, A. Thermographic Methodology for Rapid Determination of the Fatigue Limit of Materials and Mechanical Components. *Int. J. Fatigue* **2000**, *22*, 65–73. [CrossRef]
13. Fargione, G.; Geraci, A.; La Rosa, G.; Risitano, A. Rapid Determination of the Fatigue Curve by the Thermographic Method. *Int. J. Fatigue* **2002**, *24*, 11–19. [CrossRef]
14. Risitano, A.; Risitano, G. Determining Fatigue Limits with Thermal Analysis of Static Traction Tests. *Fatigue Fract. Eng. Mater. Struct.* **2013**, *36*, 631–639. [CrossRef]
15. Corigliano, P.; Cucinotta, F.; Guglielmino, E.; Risitano, G.; Santonocito, D. Fatigue Assessment of a Marine Structural Steel and Comparison with Thermographic Method and Static Thermographic Method. *Fatigue Fract. Eng. Mater. Struct.* **2020**, *43*, 734–743. [CrossRef]
16. Davis, S. Pneumatic Actuators. *Actuators* **2018**, *7*, 62. [CrossRef]
17. Krause, J.; Bhounsule, P. A 3D Printed Linear Pneumatic Actuator for Position, Force and Impedance Control. *Actuators* **2018**, *7*, 24. [CrossRef]
18. Lee, Y.B.; Park, J.W.; Lee, G.C. A Study on Failure Analysis and High Performance of Hydraulic Servo Actuator. *Appl. Sci.* **2020**, *10*, 7451. [CrossRef]
19. Curran, M.A. Life Cycle Assessment: A Review of the Methodology and Its Application to Sustainability. *Curr. Opin. Chem. Eng.* **2013**, *2*, 273–277. [CrossRef]
20. Khorasani, M.; MacDonald, E.; Downing, D.; Ghasemi, A.; Leary, M.; Dash, J.; Sharabian, E.; Almalki, A.; Brandt, M.; Bateman, S. Multi Jet Fusion (MJF) of Polymeric Components: A Review of Process, Properties and Opportunities. *Addit. Manuf.* **2024**, *91*, 104331. [CrossRef]
21. Avanzini, A.; Battini, D.; Pandini, S. Static and Fatigue Behavior in Presence of Notches for Polyamide 12 (PA12) Additively Manufactured via Multi Jet Fusion^{TM} Process. *Int. J. Fatigue* **2022**, *161*, 106912. [CrossRef]
22. D'Andrea, D.; Crisafulli, D.; Risitano, G.; Santonocito, D. Rapid Fatigue Life Evaluation over Time of PA12 Obtained by Multijet Fusion Printing Process. *Int. J. Fatigue* **2025**, *200*, 109124. [CrossRef]

23. Hampo, C.C.; Akmar, A.B.; Majid, M.A.A. Life Cycle Assessment of an Electric Chiller Integrated with a Large District Cooling Plant. *Sustainability* **2021**, *13*, 389. [CrossRef]
24. Halkes, R.T.; Hughes, A.; Wall, F.; Petavratzi, E.; Pell, R.; Lindsay, J.J. Life Cycle Assessment and Water Use Impacts of Lithium Production from Salar Deposits: Challenges and Opportunities. *Resour. Conserv. Recycl.* **2024**, *207*, 107554. [CrossRef]
25. Notter, D.A. Life Cycle Impact Assessment Modeling for Particulate Matter: A New Approach Based on Physico-Chemical Particle Properties. *Environ. Int.* **2015**, *82*, 10–20. [CrossRef]
26. van den Oever, A.E.M.; Puricelli, S.; Costa, D.; Thonemann, N.; Lavigne Philippot, M.; Messagie, M. Revisiting the Challenges of Ozone Depletion in Life Cycle Assessment. *Clean. Environ. Syst.* **2024**, *13*, 100196. [CrossRef]
27. Van Hooreweder, B.; De Coninck, F.; Moens, D.; Boonen, R.; Sas, P. Microstructural Characterization of SLS-PA12 Specimens under Dynamic Tension/Compression Excitation. *Polym. Test.* **2010**, *29*, 319–326. [CrossRef]
28. McKechan, D.J.A.; Robinson, C.; Sathyaprakash, B.S. A Tapering Window for Time-Domain Templates and Simulated Signals in the Detection of Gravitational Waves from Coalescing Compact Binaries. *Class. Quantum Gravity* **2010**, *27*, 084020. [CrossRef]
29. Kallel, A.Y.; Hu, Z.; Kanoun, O. Comparative Study of AC Signal Analysis Methods for Impedance Spectroscopy Implementation in Embedded Systems. *Appl. Sci.* **2022**, *12*, 591. [CrossRef]
30. Belega, D.; Petri, D.; Dallet, D. Accurate Amplitude and Phase Estimation of Noisy Sine-Waves via Two-Point Interpolated DTFT Algorithms. *Measurement* **2018**, *127*, 89–97. [CrossRef]
31. Arranz-Gimon, A.; Zorita-Lamadrid, A.; Morinigo-Sotelo, D.; Duque-Perez, O. A Review of Total Harmonic Distortion Factors for the Measurement of Harmonic and Interharmonic Pollution in Modern Power Systems. *Energies* **2021**, *14*, 6467. [CrossRef]
32. Yuce, E.; Minaei, S.; Tokat, S. Root-Mean-Square Measurement of Distinct Voltage Signals. *IEEE Trans. Instrum. Meas.* **2007**, *56*, 2782–2787. [CrossRef]
33. Sahu, G.N.; Singh, S.; Singh, A.; Law, M. Static and Dynamic Characterization and Control of a High-Performance Electro-Hydraulic Actuator. *Actuators* **2020**, *9*, 46. [CrossRef]
34. Sorli, M.; Gastaldi, L.; Codina, E.; De Las Heras, S. Dynamic Analysis of Pneumatic Actuators. *Simul. Pract. Theory* **1999**, *7*, 589–602. [CrossRef]
35. Pustavrh, J.; Hočevar, M.; Podržaj, P.; Trajkovski, A.; Majdič, F. Comparison of Hydraulic, Pneumatic and Electric Linear Actuation Systems. *Sci. Rep.* **2023**, *13*, 20938. [CrossRef]
36. *ASTM D-638*; Standard Test Method for Tensile Properties of Plastics. ASTM International: West Conshohocken, PA, USA, 1998.
37. Rad, C.R.; Hancu, O. An Improved Nonlinear Modelling and Identification Methodology of a Servo-Pneumatic Actuating System with Complex Internal Design for High-Accuracy Motion Control Applications. *Simul. Model Pract. Theory* **2017**, *75*, 29–47. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

MDPI AG
Grosspeteranlage 5
4052 Basel
Switzerland
Tel.: +41 61 683 77 34

Actuators Editorial Office
E-mail: actuators@mdpi.com
www.mdpi.com/journal/actuators



Disclaimer/Publisher's Note: The title and front matter of this reprint are at the discretion of the Guest Editors. The publisher is not responsible for their content or any associated concerns. The statements, opinions and data contained in all individual articles are solely those of the individual Editors and contributors and not of MDPI. MDPI disclaims responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.





Academic Open
Access Publishing

mdpi.com

ISBN 978-3-7258-8858-0