

Article

Design, Experiments, and Path Planning for a Lightweight 3D Minimally Actuated Serial Robot with a Mobile Actuator

Or Bitton [†], Avi Cohen [†]  and David Zarrouk ^{*}

Department of Mechanical Engineering, Ben Gurion University of the Negev, Beer Sheva 84105, Israel; orbitt@post.bgu.ac.il (O.B.); avi5@post.bgu.ac.il (A.C.)

^{*} Correspondence: zadavid@bgu.ac.il

[†] These authors contributed equally to this work.

Abstract: This paper presents a novel three-dimensional (3D) minimally actuated serial robot (MASR) and its unique kinematic analysis. Unlike traditional robots, the 3D MASR features a passive arm devoid of wires or motors, comprising passive rotational and prismatic joints. A single mobile actuator (MA) traverses the arm, engages designated joints for operation, and locks them in place with a worm gear setup. A gripper is attached to the MA, enabling object transportation along the arm, reducing joint actuation, and optimizing task completion time. Our key contributions include the mechanical design, and in particular the robot's passive joints with their automated actuation mechanism, and a novel optimization algorithm leveraging neural networks (NNs) to minimize task completion time through advanced kinematic analysis. Experiments with a physical prototype of the 3D MASR demonstrate its major advantages: it is remarkably lightweight (2.3 kg for a 1 m long arm and a 1 kg payload) compared to similar robots; it is highly modular (five joints R and P actuated by a single MA); and part replacement is effortless due to the absence of wiring along the arm. These features are visually depicted in the accompanying video.

Keywords: serial robots; minimal actuation; mechanical design; mobile actuator; inverse kinematics; path planning



Citation: Bitton, O.; Cohen, A.; Zarrouk, D. Design, Experiments, and Path Planning for a Lightweight 3D Minimally Actuated Serial Robot with a Mobile Actuator. *Appl. Sci.* **2024**, *14*, 8204. <https://doi.org/10.3390/app14188204>

Academic Editor: Alessandro Lo Schiavo

Received: 7 July 2024

Revised: 25 August 2024

Accepted: 4 September 2024

Published: 12 September 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Conventional serial robots typically consist of multiple rigid links interconnected by actuated joints. Most commercially available models in three-dimensional space offer four to seven degrees of freedom. However, these traditional robots often struggle to perform tasks in narrow or confined areas, which limits their use in certain industries and impacts their commercial viability. Hyper-redundant robots, also known as snake robots, have attracted growing attention for their ability to maneuver around obstacles in confined spaces [1]. Over the past few decades, extensive research has produced numerous configurations and mechanisms for diverse applications, including search-and-rescue operations [2–6], maintenance tasks, and minimally invasive medical procedures [7–10]. Their lightweight design, associated with continuum robots, also makes them potential candidates for planetary exploration and space satellite maintenance [11–13]. Despite their advantages, the high dimensionality of highly redundant serial robots makes controlling and planning motion challenging.

In previous works, we introduced two innovative designs to address the complexities associated with the kinematics, actuation and dynamic modeling of these robots. The first was a minimally actuated reconfigurable track robot [14], which simplified kinematics and actuation. The second was the minimally actuated serial robot (MASR) [15,16], featuring a serial robot with a mobile actuator (MA) that traversed its links to rotate its joints. The MASR combined the advantages of minimally actuated and hyper-redundant robots by offering enhanced reliability, reduced weight, lower costs, increased modularity, and

simplified modeling due to its kinematic chain properties. These robots are particularly well-suited for applications where low weight is prioritized over high speed, such as in space exploration and agriculture.

Despite these advances, the design of the 3D MASR presented unresolved challenges, particularly concerning its kinematics. One key issue remained the determination of an optimal solution for the MASR's specific inverse kinematics (IK) problem, which involves finding an IK solution for the desired gripper position in Euclidean space by optimizing the MA's movement to activate the joints and sequentially determine the final position. IK methods are generally divided into analytical and numerical approaches. While analytical methods offer closed-form solutions, they are often impractical for complex and redundant robots like the 3D MASR [17]. Numerical methods, typically based on least squares and gradient descent, apply to a broader range of kinematic chains [18], but they suffer from inconsistent convergence rates and are highly dependent on the initial guess. Over time, various data-driven learning methods have been investigated to tackle complex IK challenges [19]. Notably, neural networks (NNs) can manage intricate kinematic systems and deliver real-time predictions [20–22]. However, these data-driven IK techniques are designed specifically for traditional manipulators. To respond to these issues, we present an upgraded 3D MASR robot with several mechanical enhancements. The new arm supports 3D movement through a rotating work-plane joint and features a unique prismatic joint, both of which utilize high-torque, precise worm gears. These improvements enhance the robot's strength, functionality, speed, reliability, and accuracy. In addition, the MA now includes a gripper, thus allowing for seamless object translation along the links without requiring arm rotation. The inclusion of an electronic controller and sensors enables automated, precise control over translation and rotation, as shown in Figure 1.

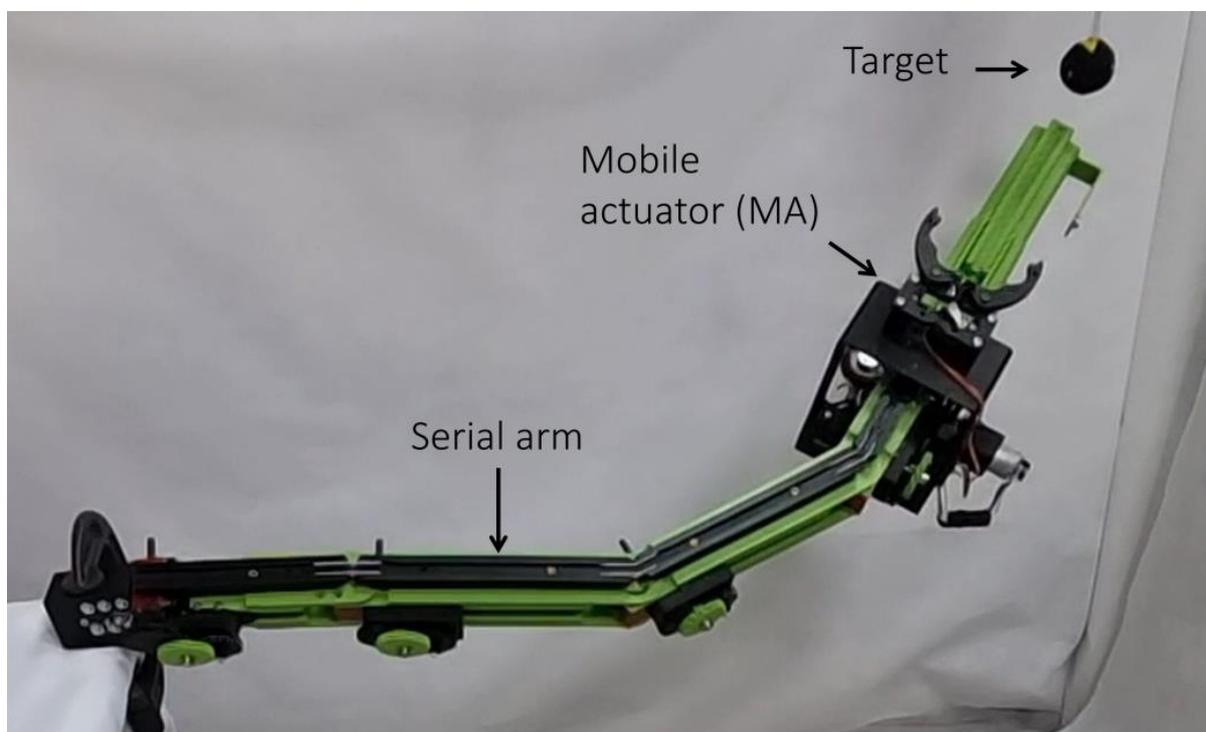


Figure 1. The newly developed 3D minimally actuated serial robot, 3D MASR, has a passive serial arm and a single mobile actuator (MA). It is configured with 4 rotational and 1 prismatic joint.

The path-planning algorithm described here incorporates an IK solver based on a NN, which provides a joint configuration solution that moves the MA to the desired pose given the current joint configuration, which is optimized for minimal traversal time between configurations for both the joints and the MA. These actions are minimized using a specially

designed regularization term tailored for every form of MASR. This approach leverages the unique advantages of the MA by enabling it to be positioned anywhere along the arm, thus requiring less joint actuation. Importantly, training the NN does not require extensive data collection from an actual or simulated robot since it relies on the computation of forward kinematics.

This paper is organized as follows: Section 2 presents the robot design, and Section 3 covers its control. Section 4 deals with the kinematic analysis, and Section 5 focuses on the development of a path-planning algorithm that reduces action time. In Section 6, we present the results of multiple experiments conducted with the robot.

2. Robot Design

The 3D MASR, illustrated in Figure 2, is an easily reconfigurable robot composed of a passive serial arm with multiple either rotational or prismatic joints without wiring or motors. A mobile actuator (MA) is incorporated into the robot's design, allowing it to move along the links and actuate the joints as necessary. In this specific design, the MASR has a total of five degrees of freedom (DOF) and is composed of four rotational joints and one prismatic joint. At the zero reference position (ZRP), the first joint's axis is parallel to the X-axis, whereas the next three joints' axes are parallel to the Z-axis. The last joint is prismatic. Note that other configurations are possible and that the design is easily reconfigurable, given that there is no wiring along the links. The total length of the links is 90 cm (105 cm when the prismatic link is extended). The first three links have a length of 20 cm, the length of the fourth link is 10 cm, and the last link can extend from 20 to 35 cm. A summary of the key specifics of the initial prototype appears in Table 1.

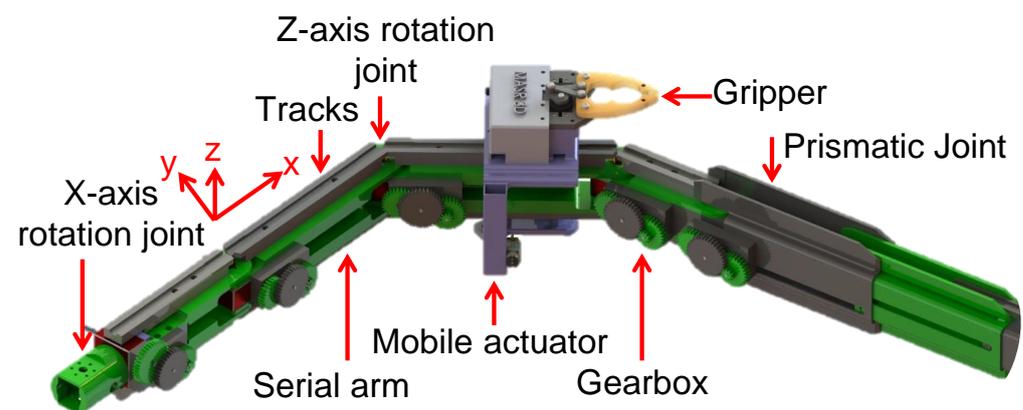


Figure 2. Mechanical design of the 3D MASR: a serial arm with five passive joints actuated by an MA equipped with a gripper.

Table 1. Main parameters of the robot.

Entries	Value
Serial arm length	90–105 cm
Serial arm weight	2 kg
MA weight	0.3 kg
z-axis joint rotation speed	14 degrees/s
x-axis joint rotation speed	14 degrees/s
Prismatic joint speed	8 cm/s
MA travel speed	10 cm/s
Maximum payload	1 kg
Repeatability	0.5 cm

2.1. The Serial Arm

In the current design, the serial arm consists of five links: two large links, each 20 cm long, two small links, each 10 cm long, and a prismatic link with a variable range of 20 to 40 cm, as illustrated in Figure 3. These links are interconnected through rotational and prismatic joints, thus making it possible to adjust the length of the arm from 90 to 105 cm. It has a total weight of 2 kg. The arm utilizes a high-quality worm wheel gearbox and a spur gear transmission to rotate the joints. The relative angle between two adjacent links can be adjusted within a range of $[-50^\circ, 50^\circ]$ except for the first link, which can rotate infinitely around the x -axis.

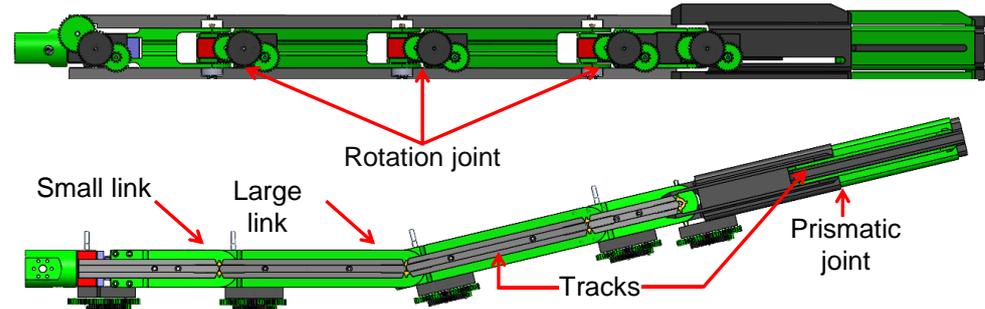


Figure 3. The passive arm is composed of five links that are connected by passive joints to a track on top of the links, which enables the MA to interface with it.

Tracks are installed on both the top and bottom of the arm. The MA interfaces with these tracks and is thus confined to moving along the arm's length. A toothed belt is fitted along the side of the arm to increase traction with the MA when traveling over the links, thus eliminating the possibility of sliding. A key feature of the arm is that it comprises passive components, including passive joints, and does not contain wiring or electronics like sensors and motors.

2.1.1. The Passive Rotation Joints

The passive rotational joint consists of a high-quality worm wheel gearbox with a backlash of less than 8 arc minutes and a spur gear transmission, as illustrated in Figure 4. The joint actuation motor engages the gear transmission to initiate the rotation of a link relative to the preceding one. This movement is transmitted to the gearbox's drive shaft and then transferred to the link.

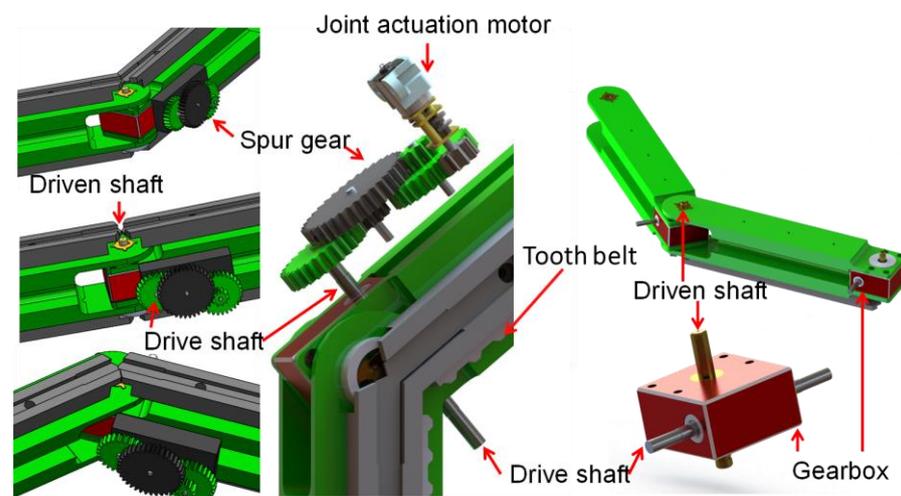


Figure 4. The passive rotational joint mechanism is composed of a worm wheel gearbox and a spur gear transmission. This transmission allows for sensitive and precise movement and the self-locking of the joints.

Overall, the transmission components have a total transmission ratio of 1:720, providing high sensitivity in link rotation (0.5 degrees per motor rotation). This high gear ratio allows the arm to be operated with a relatively small torque (see Section 3). The gearbox's self-locking capability ensures that the joint remains fixed at the desired angle at the end of the actuation.

2.1.2. The Passive Prismatic Joint

The passive prismatic joint, illustrated in Figure 5, is composed of two links, a tension wire, a pulley to wind the wire, and a spring that maintains the tension in the wire. The system operates by threading the wire through both links, so that pulling on the wire causes the springs to contract and the mechanism to close. The wire is attached to the pulley, allowing the direction of pulley rotation to either shorten or lengthen the wire and hence the mechanism itself. Similar to the other passive joints, the MA is responsible for rotating the pulley axis. In order to reduce the torque required for operation, the same spur gear transmission used in the rotary joint is attached to the pulley shaft. This setup serves to adjust the length of the prismatic link from 20 to 35 cm in this design setup.

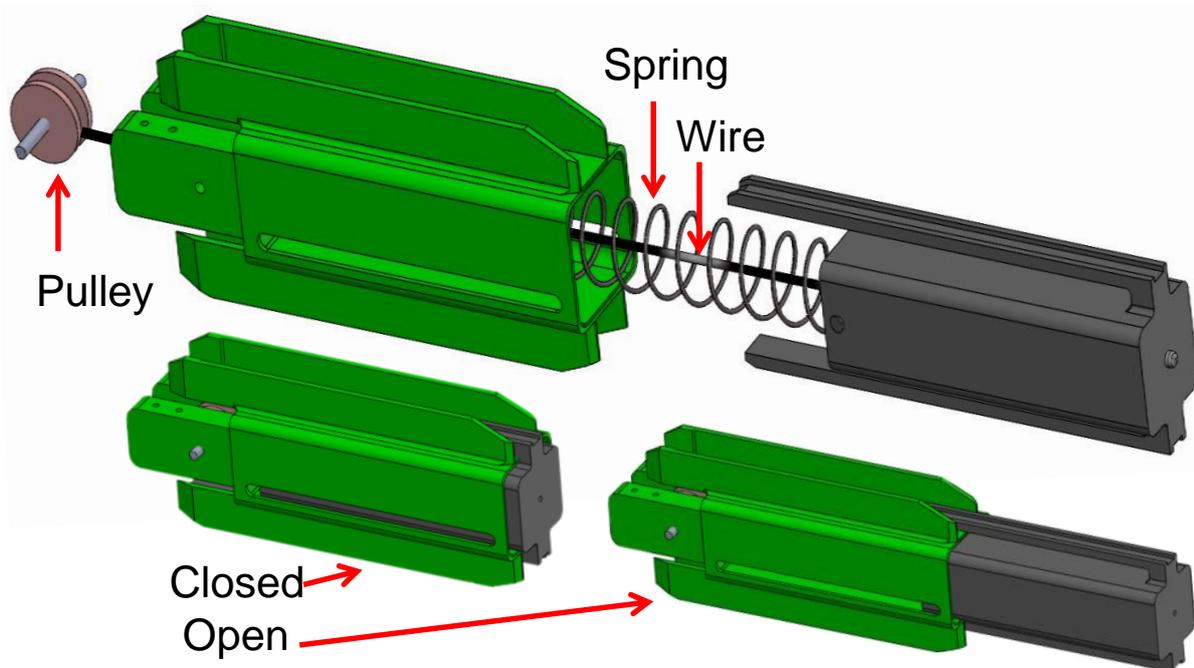


Figure 5. The passive prismatic joint. When the pulley is rotated, the mechanism changes the length of the prismatic joint. The spring keeps the thread taut.

2.2. The Mobile Actuator (MA)

The MA, illustrated in Figure 6, has three distinct mechanisms: a locomotive mechanism, a joint rotation mechanism, and a gripper. The MA travels along the arm's tracks and maintains stability using four bearings: two at the bottom and two at the top (see Figure 6—left), ensuring high grip and low friction between the MA and the links. The MA pauses at specified locations to activate the joints, which allows it to rotate the rotational joints and adjust the length of the prismatic joint. The MA grips objects using the gripper, thus facilitating object conveyance along the arm. These mechanisms enable the MA to maneuver along the arm effectively, actuate the joints, and handle objects as needed.

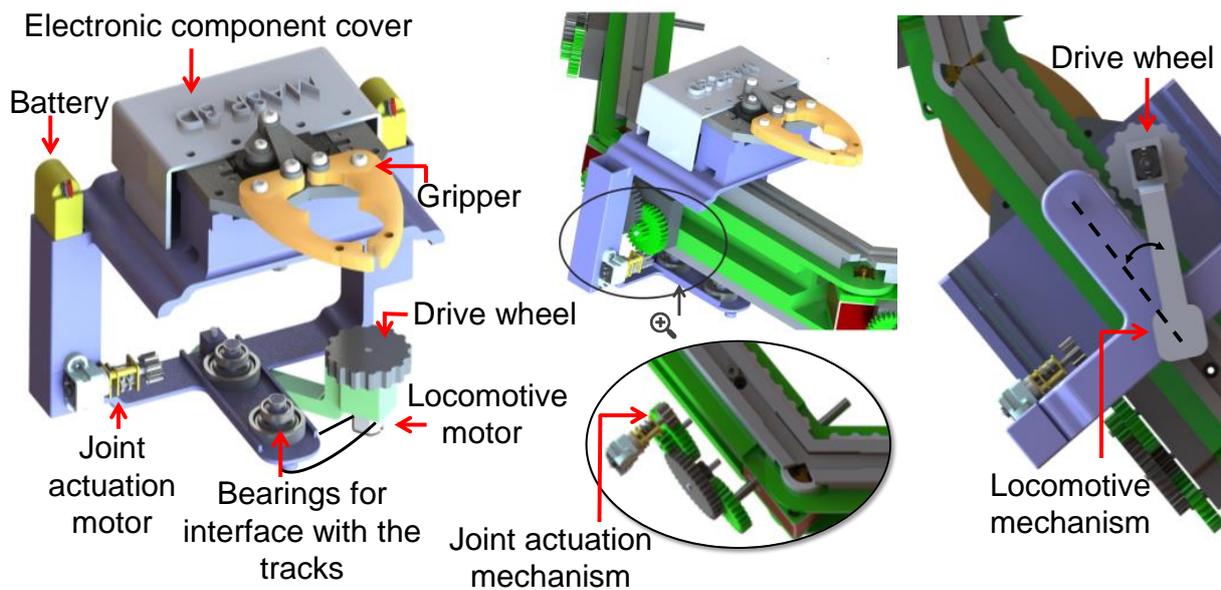


Figure 6. The MA comprises a locomotive mechanism, a joint rotation mechanism and a gripper. The controller, batteries and sensors are integrated onboard.

2.2.1. The Locomotive Mechanism

The locomotive mechanism uses a single drive wheel to transport the MA along the links. The wheel, mounted with a motor on a bracket with an angular degree of freedom, is attached to the arm's side with rubber clamps. This design allows the MA to navigate curved joints of up to 50 degrees without losing contact with the arm's side. The wheel's driving force and track reactions ensure smooth movement along the arm, even when on a curved path. At a curvature of 50 degrees or more, the mobile actuator cannot advance.

2.2.2. The Joint Rotation Mechanism

The joint rotation mechanism is fitted with a DC motor mounted on the MA's left side. The motor is equipped with a spur gear that engages the gear of a specific joint upon contact. This spur gear drives and adjusts the relative angle between the adjacent links or the length of the prismatic joint, as illustrated in Figure 6 and Video S1. The high transmission ratio of the joints (1:720) reduces the torque needed by the actuation mechanism, enabling relatively smooth motion.

When the MA passes a joint without activating it, the engagement between the motor's spur gear and the first gear transmission positioned on the link causes the transmission's first gear to rotate roughly one-tenth of a revolution. This rotation translates to a 1:7200 joint revolution, or 0.05 degrees. This slight rotation can be ignored since it is too small to impact the high transmission ratio and backlash.

2.2.3. The Gripper

The MA is equipped with a commercially available (unmodified) two-finger gripper that is actuated by a servo motor that can grasp objects measuring 2.5 cm to 10.5 cm in width, as illustrated in Figure 6. The gripper is attached to the MA, which can move along the arm and translate objects. Grippers with more fingers or other tools can be fitted for more sophisticated applications. Alternatively, the gripper can be attached to the end of the last link of the robot, as in conventional robots.

2.2.4. Manufacturing and Assembly

The robot is almost entirely 3D printed using FDM technology. Replacement is a straightforward task since no wires run through the links. Furthermore, the robot is fast

and easy to assemble, so that a variety of links can be adjusted to address a range of diverse tasks.

3. Actuation and Control

The primary advantage of the 3D MASR is its innovative use of a mobile actuator (MA), with only a few actuators that move along the robot's arm. This design reduces the robot's total weight while expanding its work space. For purposes of illustration, consider creating a serial arm with an MA but without the current joint rotation. This setup maintains the same workspace. However, the robot's size and weight would increase significantly if built without this innovation. For example, a 1-m-long serial arm and five joints would require a motor house containing one motor for each joint. Each motor weighs 82 g, and the motor house adds roughly 50 g. If the control system is positioned at the base, around 30 cables for power and signal transmission (2 for power and 4 for encoder power and signals for each joint) would be needed. These cables weigh 20 g per meter, adding up to 60 g in total for 30 cables. The design adjustments required to accommodate these cables would further increase the size and weight, with the cable management system adding about 50 g per link, based on a conservative estimate. Altogether, this would lead to a minimum weight increase of 970 g for the arm, representing a 48.5% increase in weight. Without the MA concept, the unique advantages of this robotic design would be significantly diminished.

3.1. Actuation

The robot, excluding its gripper, is actuated using two motors: a 20 mm motor to drive the MA along the links and a 25 mm motor to actuate the joints. Both motors operate within a voltage range of 6–9 volts, have different gear ratios, and come with a built-in encoder, thus enabling precise control of the actuators' movements. The 20 mm motor is equipped with a built-in gear with a ratio of 1:488, providing a torque of 3.5 kg-cm at 24 rpm under optimal working conditions. The 25 mm motor has a gear ratio of 1:4.4, which delivers a torque of 0.22 kg-cm at 1800 rpm. The motors and the gear transmissions were carefully selected to meet two primary requirements:

1. Ensure that the MA can travel against gravity while carrying a load of 1 kg. The system can carry a total load of 3.2 kg.
2. Ensure that the joints can rotate even when the robot carries a 1 kg load. In the current design, the motors can rotate the joints even when a load of 2.64 kg is applied at the proximity of the last link.

3.2. Control

The arm is entirely passive, so that the control system is located on the MA. A micro-controller, specifically an Arduino Nano, controls the MA's movement, joint rotation, and gripper. The angular displacement of the joint rotation mechanism is measured using a magnetic encoder that provides 24 counts per motor revolution. Given the large transmission ratio, a change of one degree between a pair of links requires counting approximately 400 encoder steps, enabling precise control.

To ensure that the MA stops precisely at the correct position to engage the gears and rotate the joints, small magnets are inserted into the centers of the joints, and a Hall effect sensor (A1302) is mounted on the MA. Inputted into the control algorithm is the initial configuration of the arm and a series of m actions, where each action specifies the i -th joint number and the required angle for that joint, θ_i . By using the encoder and Hall sensor, the MA can reliably bring the arm to the desired configuration.

The series of actions for the motors is derived from the motion planning algorithm, as presented in Figure 7. This setup allows for precise and reliable control of the robot's movements by ensuring the accurate positioning and actuation of the joints.

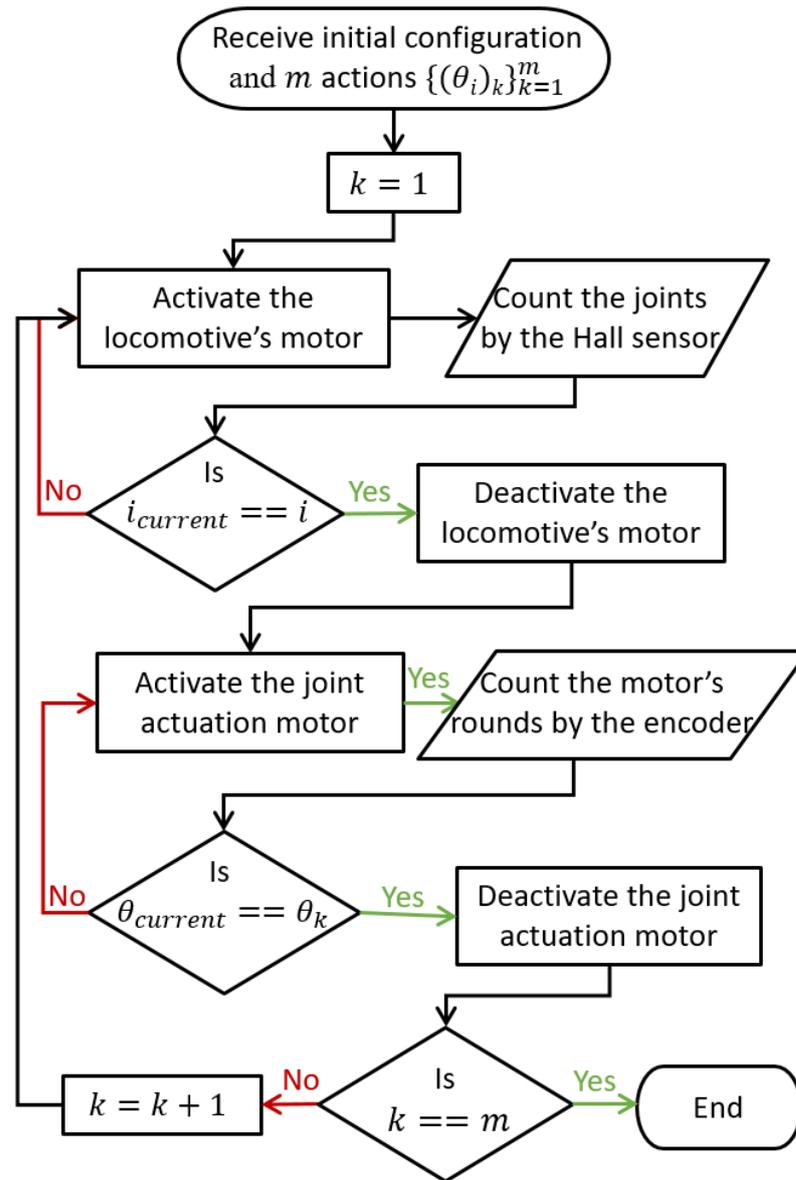


Figure 7. Flowchart of the robot's control system for a series of actions.

4. Kinematic Model

The robot's structure consists of a sequence of n links, each with a length of l_i , where $i = (0, 1, \dots, n - 1)$. These links are interconnected through n joints where $n - 1$ of them move in the same plane, and one rotates this plane. We denote the angle in the i -th joint as θ_i , as illustrated in Figure 8. In addition, since the gripper can move along the links, we need to include its position in the robot's coordinates. We define $d \in [0, \sum_{i=0}^{n-1} l_i]$ as the gripper's position along the arm. In the simple case where the gripper is attached to the tip of the last link (LL), we reduce it to one point: $d = \sum_{i=0}^{n-1} l_i$. In the second case, where the gripper is attached to the MA, it can be positioned anywhere along the links, defined as any link (AL). Finally, the configuration of the 3D MASR, \mathbf{q} , is defined as follows:

$$\mathbf{q} = (\theta_0, \theta_1, \dots, \theta_{n-1}, d)^T = (\boldsymbol{\theta}^T, d)^T \in \mathbb{R}^n. \tag{1}$$

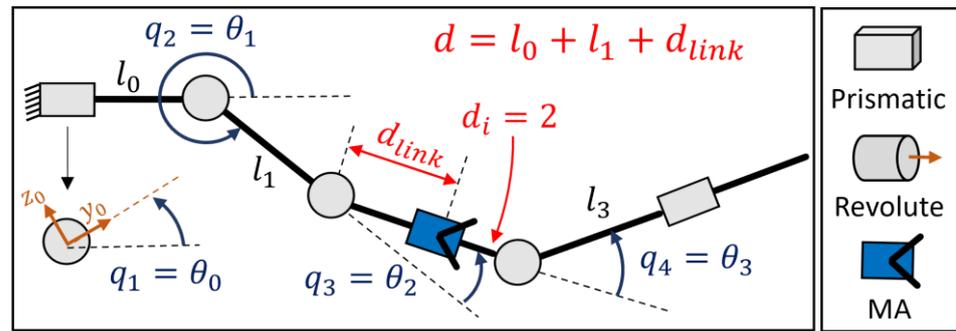


Figure 8. Schematic diagram of 3D MASR. A top/side view of the robot and definition of the geometric parameters.

4.1. Forward Kinematics (FK)

We define the gripper’s position relative to a fixed frame as \mathbf{x} , which represents the set of all the point (x, y, z) coordinates in space that the gripper can reach. The forward kinematics (FK) function maps from the configuration space \mathbf{q} to the workspace \mathbf{x} and is given by $\mathbf{x} = f(\mathbf{q})$. The \mathbf{q} space contains the joint rotation angles and the MA’s position along the arm, as presented in Equation (1).

To define the kinematics equation, d_i and d_{link} need to be extracted from d , where d_i represents the i -th link on which the gripper is located, and d_{link} represents the gripper position relative to the link it is on, as depicted in Figure 8. These parameters allow the FK function to map the gripper’s position within the workspace accurately.

The position of the gripper is given by

$$\mathbf{x} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = R_x(\theta_0) \cdot \mathbf{x}_p = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_0 & -\sin \theta_0 \\ 0 & \sin \theta_0 & \cos \theta_0 \end{bmatrix} \begin{pmatrix} x_p \\ y_p \\ z_p \end{pmatrix}. \tag{2}$$

where \mathbf{x}_p is the gripper position without rotation of the first joint, as follows:

$$\mathbf{x}_p = \begin{bmatrix} l_0 + \sum_{i=1}^{d_i-1} l_i \cos \left(\sum_{j=0}^i \theta_{j+1} \right) + d_{link} \cos \left(\sum_{j=0}^{d_i} \theta_{j+1} \right) \\ \sum_{i=1}^{d_i-1} l_i \sin \left(\sum_{j=0}^i \theta_{j+1} \right) + d_{link} \sin \left(\sum_{j=0}^{d_i} \theta_{j+1} \right) \\ 0 \end{bmatrix}. \tag{3}$$

4.2. Actuation Time

When using a single mobile actuator (MA) to control the joints of the serial arm, each joint can be actuated individually. The total time needed to reconfigure the joint angles and reach a specific target is composed of the time required to travel along the links, engage the joints, and rotate them. Assuming a constant linear velocity of the MA, denoted by \dot{d} , and a constant rotational velocity denoted by $\dot{\theta}$, the time required to reach a target can be calculated as follows:

1. The time, τ_d , required for the MA to move from its initial position, d_{init} , to the position of the target, d , which can be calculated as shown below:

$$\tau_d = \frac{|d_f - d_{init}|}{\dot{d}} \tag{4}$$

- The time, τ_θ , required to rotate the joints to from the initial arm angles, θ_{init} , to the desired angles, θ_f , which can be calculated as shown below:

$$\tau_\theta = \frac{|\theta_f^T \cdot \theta_{init}|}{\dot{\theta}} \tag{5}$$

Therefore, the approximation of the total time is dependent on the configuration of \mathbf{q}_{init} and \mathbf{q}_f , and is given by

$$\tau(\mathbf{q}_{init}, \mathbf{q}_f) = \tau_d + \tau_\theta \tag{6}$$

4.3. Workspace

Given that the joints in the current design are limited to 50 degrees of rotation per side, we analyzed the serial arm’s work volume in 3D space. The work volume was determined through an exhaustive search of the total area for possible solutions using the motion planning algorithm presented in Section 5.

Figure 9B illustrates the arm’s work volume when the gripper can be positioned anywhere on the arm (AL). Notably, when the gripper is located only at the edge (LL), the work volume is included within the volume of AL but is significantly smaller.

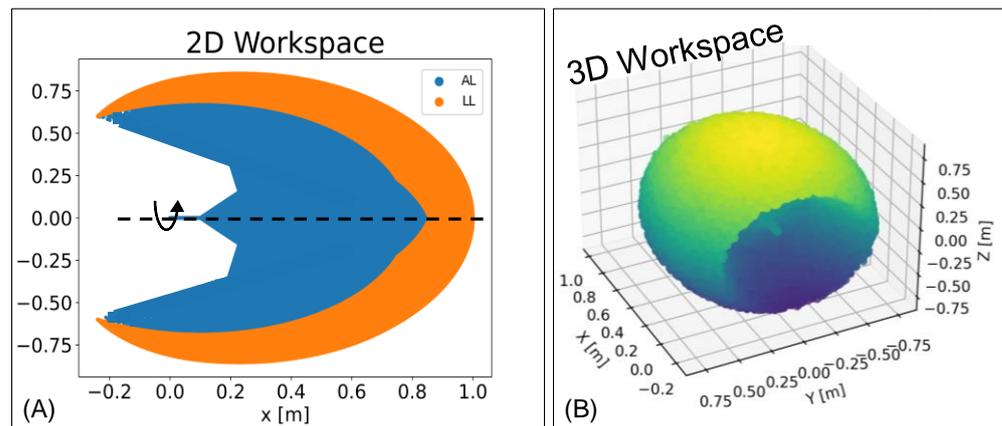


Figure 9. Work volume of the robot. (A) Work plane (before the rotation of the first joint). The LL work volume is included in the AL work volume. (B) AL work volume.

To make the work volume easier to understand, we first illustrate the work plane (before the rotation of the first joint) in Figure 9A. This plane rotates around the x-axis using the first joint (θ_0), thus creating a three-dimensional workspace as described in the kinematics equations.

5. Three-Dimensional Motion Planning Algorithm

The specific 3D MASR robot has five degrees of freedom (DOF), resulting in infinite solutions to reach a specific point in the 3D space, excluding orientation considerations. The inverse kinematics (IK) function maps from position space \mathbf{x} to the configuration space \mathbf{q} given by $\mathbf{q} = f^{-1}(\mathbf{x})$.

Here, the goal is to determine the final configuration \mathbf{q}_f that positions the gripper at the desired final position \mathbf{x}_{des} , given the initial joint configuration \mathbf{q}_{init} . Simultaneously, the aim is to minimize the positional error, e , and reduce the total time required to complete the given motion task, τ .

The analysis assumes that the robot consists of five links of length l_i , measuring [10, 20, 20, 10, 20–40] cm, respectively. The rotational joint angles range from -50 to 50 degrees, and the gripper position ranges from 0 to 100 cm, reflecting a specific configuration of the

experimental robot. The workspace is assumed to be obstacle-free, allowing the arm to move freely.

5.1. Method

The algorithm is based on minimizing a loss function \mathcal{L} , which combines the proximity of the robot to the target point and the total time it takes to reach it. This is typically achieved using numerical approaches. In this approach, the loss function is minimized by training a neural network (NN) using the Adam optimizer, an advanced version of stochastic gradient descent. This method can find an optimal solution for the IK problem.

The advantages of using NNs for solving IK include their ability to generalize from trained scenarios to efficiently handle new configurations and provide quick solutions once trained through the GPU, which makes them particularly well-suited for real-time applications. Further, unlike traditional numerical approaches, NNs are not sensitive to initial conditions. However, the training phase can take up to 20 min, and tuning the hyperparameters is based on expectation and trial and error.

During the training phase, we iteratively optimized the hyperparameters to balance the tradeoff between minimizing the positional error and the total time, but ultimately achieved sub-optimal results with relatively small errors (on the order of a few millimeters). To further increase the accuracy of the solution, we applied a linear approximation (using the Jacobian matrix) to reach sub-millimeter accuracy (see Section 5.2.2).

5.1.1. Architecture

The IK problem involves finding the optimal final configuration by minimizing the following loss function:

$$\mathcal{L}(\mathbf{x}_{des}, \hat{\mathbf{x}}, \tau) = \|\mathbf{x}_{des} - \hat{\mathbf{x}}\|_2 + \lambda \cdot \tau(\mathbf{q}_{init}, \hat{\mathbf{q}}), \tag{7}$$

τ is calculated from Equation (1), and λ is the regularization weight, which controls the importance of minimizing time in the training process.

The NN architecture takes the input data, consisting of the initial configuration, \mathbf{q}_{init} , and the desired position, \mathbf{x}_{des} , and outputs the estimated desired configuration $\hat{\mathbf{q}}$ that leads to the estimated desired position $\hat{\mathbf{x}}$, as illustrated in Figure 10. $\hat{\mathbf{x}}$ is calculated using the output of the forward kinematics, $\hat{\mathbf{x}} = f(\hat{\mathbf{q}})$, which is then inserted into the loss function with the total time, as shown in Equation (4). Note that this model only needs data for the MA positions, and the initial configurations can be randomized during training.

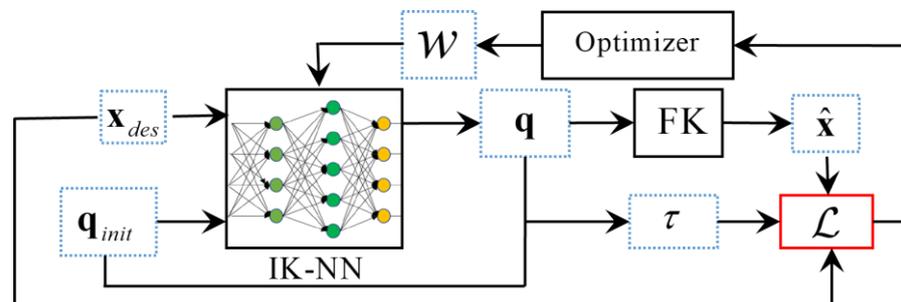


Figure 10. The architecture consists of four components: an IK block that uses a NN, an FK block to estimate position, a time calculation block, a loss function that defines the optimization problem, and an optimizer that updates the NN weights to minimize the loss function.

5.1.2. Data Generation

In our IK model, both training and testing require a dataset that comprehensively covers the robot’s workspace. We first calculated positions using forward kinematics equations and random joint values to determine the workspace. However, given the non-linearity of the forward kinematics, the initial dataset was not evenly distributed within the robot’s workspace, which resulted in a significantly lower probability of reaching

the edges of the workspace, as demonstrated in Figure 11. Consequently, the model trained disproportionately on certain workspace areas, which negatively impacted its generalization ability. To address this issue, we took the following steps:

1. Determined the outer shell of the workspace using Python's convex hull function.
2. Created an array of uniformly distributed points in a box format, with boundaries derived from the minimum and maximum values of the outer shell.
3. Filtered the points within the box, retaining only those inside the shell.

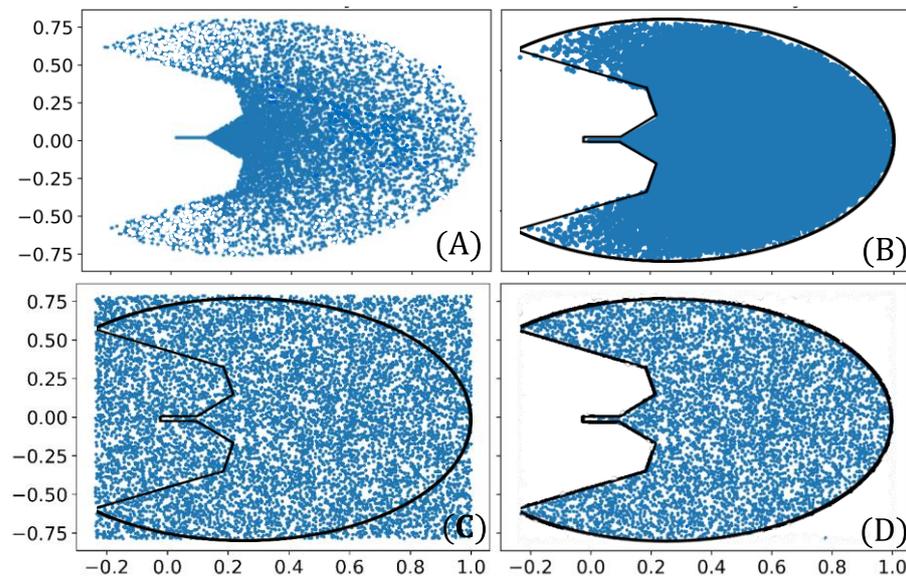


Figure 11. A section view of the data generation process. (A) The workspace points with a non-uniform distribution (fewer points were drawn at the edge). (B) Generating numerous points to ensure comprehensive coverage within the workspace and finding the workspace shell. (C) Creating a box with a uniform distribution within the boundaries of the shell. (D) Filtering the points within the box, retaining only those located inside the shell.

This method led to a balanced dataset featuring a uniform distribution, giving all points in the workspace an equal opportunity. This dataset was partitioned into 450,000 examples for training and validation and another 50,000 for evaluation. This strategy significantly boosted the model's ability to generalize across the entire workspace and helped prevent overfitting to specific areas.

5.1.3. Target Positional Error

The Euclidian distance, denoted as e , between the estimated position of the MA, \hat{x} , and the desired position, x_{des} , is expressed as

$$e = \sqrt{(x_{des} - \hat{x})^2 + (y_{des} - \hat{y})^2 + (z_{des} - \hat{z})^2}. \quad (8)$$

This distance served as the metric for evaluating the IK models, and is referred to as the positional error.

5.2. Results

This section describes the performance of two IK models using a uniform dataset generation approach. The first model, Model I, only minimizes the distance to the target ($\lambda = 0$). The second model, Model II, minimizes both distance and time ($\lambda \neq 0$). Our examination emphasizes the trade-off between precision and time efficiency, showcasing the effectiveness of the time-minimizing model.

5.2.1. Training

We trained two distinct IK models over 1300 epochs—one minimizing only distance and the other minimizing both accuracy and time actuation, termed Model I and Model II, respectively.

We employed an iterative approach to hyperparameter tuning to select the optimal hyperparameters for each model. Initial values were selected based on standard practice and prior experience. We then systematically varied these hyperparameters to identify combinations that improved the model's performance on our validation set. As expected, the accuracy-alone model achieved a more accurate result and decreased the error by 48% but increased time by 40% compared to the time and accuracy model (see Figure 12). The network architecture, hyperparameters, and main results for each model are detailed in Table 2.

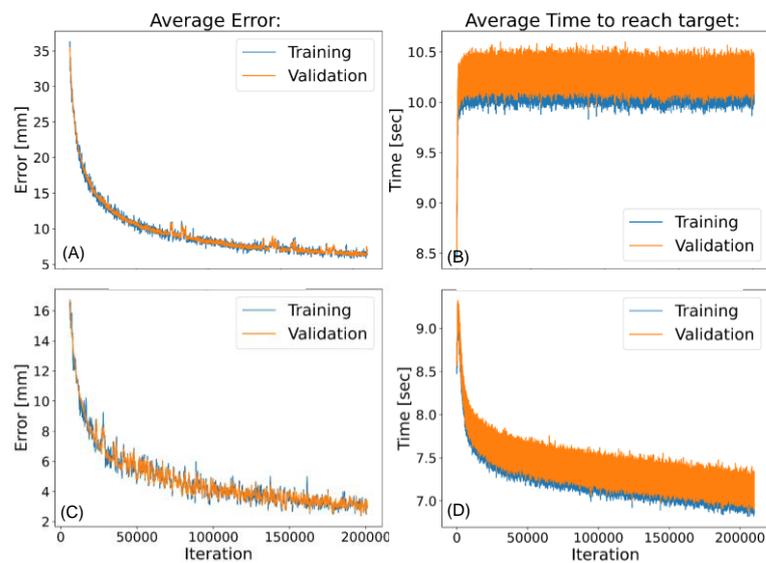


Figure 12. The learning curves for the two models: the error and time taken to execute a task (A,B) for Model I without a time constraint and (C,D) for Model II with a time constraint.

Table 2. Network architecture, hyperparameters, and main results.

	Model I	Model II
Network (#neurons)	[256, 128, 64, 32]	[256, 128, 64, 32]
Learning rate	0.001	0.002
Batch size	3000	3000
Regularization weight λ	0	0.01
Mean error \bar{e} (mm)	5.4	2.8
Mean time $\bar{\tau}$ (s)	7.3	10.2

This comparison highlights the tradeoff between distance, accuracy, and time efficiency, and demonstrates the importance of tailoring the model to specific task requirements and desired performance metrics.

5.2.2. Fine Tuning Using Jacobian Linear Approximation

Consistent with the training results, we observed a test error of 5.5 mm for the distance and time accuracy model (Model II) and 2.8 mm for the distance accuracy model (Model I), with average times of 7.3 s and 10.2 s, respectively, to achieve the goal.

Proximity to the target enabled a first-order approximation, and minor output adjustments significantly enhanced accuracy with minimal added time (less than 1%). The linear transformation (approximation) between the 3D cartesian coordinates to the robots' joints implemented a Jacobian matrix. To ensure the Jacobian matrix's invertibility, we performed

the transformation on 3 DOF (the minimum needed for a spatial solution). Changes were made to angle θ_0 , which rotates the robot’s working plane, d_i , the mobile actuator’s (MA’s) position along the arm, and θ_{d_i} , the joint moving the link where the mobile motor resides. The following equation illustrates these small output changes:

$$\begin{pmatrix} \Delta\theta_0 \\ \Delta\theta_{d_i} \\ \Delta d_i \end{pmatrix} = \begin{bmatrix} \partial x/\partial\theta_0 & \partial x/\partial\theta_{d_i} & \partial x/\partial d_i \\ \partial y/\partial\theta_0 & \partial y/\partial\theta_{d_i} & \partial y/\partial d_i \\ \partial z/\partial\theta_0 & \partial z/\partial\theta_{d_i} & \partial z/\partial d_i \end{bmatrix}^{-1} \cdot \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} = J^{-1} \cdot (\hat{\mathbf{x}} - \mathbf{x}_{des}). \tag{9}$$

This approach had a negligible influence on the loss function goal, since it was conducted near the target and only led to minor changes in the joint angles.

5.2.3. Testing and Examples

In this section, we present statistical results for 50,000 cases examining the model’s performance, accompanied by visual examples that illustrate the differences in accuracy and time efficiency between the two models. The results presented in Table 3 demonstrate the effectiveness of incorporating the Jacobian method as a complementary step subsequent to the network’s output. This method reduced the average error across 50,000 examples without significantly impacting task execution time. This approach has several notable advantages. The first is that by terminating the network training relatively early, we can prevent overfitting of the data while still achieving a low error rate.

Table 3. Testing the results of 50,000 unseen examples.

	NN		NN + Jacobian	
	Model I	Model II	Model I	Model II
Mean error \bar{e} (mm)	2.8	5.5	0.66	0.28
Mean time $\bar{\tau}$ (s)	10.2	7.3	10.6	7.91
$\bar{e} < 1$ mm (%)	80	73	98	95

Moreover, in scenarios where time constraints are crucial, the network brings us closer to the target points, and the subsequent application of the Jacobian method fine-tunes the angles to attain maximum accuracy with minimal impact on the overall task completion time. By using this approach, we reduced the average error to 0.28 mm (a reduction of 95%) for the model without time constraints, and to an average of 0.66 mm for the model with time constraints (a reduction of 77%). We observed that the modifications to the angles had a minimal impact (an increase of 0.82%) on the time required to perform the task, making this a robust solution. This combined strategy ensured improved accuracy without compromising efficiency.

Figure 13 depicts the algorithm’s successful attainment of an arbitrary target point from a non-zero initial configuration. In this example, the robot’s initial joint angles were $[-15.5, 7.5, 34.7, 46.8]$ degrees, and the gripper was positioned 37 cm from the first joint. The target was located at coordinates (55, 0, 50) cm. These parameters were entered into the algorithm. The resultant solution demonstrates the advantage of this type of optimization. For the model focusing solely on accuracy (Model I), the final joint angles were $[93.2, -27.2, 36.4, 11]$ degrees, and the gripper’s position was 72 cm along the arm (Figure 13A). The actuator made significant adjustments to three out of the four joints, reaching the target with an error of only 0.0052 mm in 6.9 s. While this error is by far smaller than the accuracy of the robot, it demonstrates the efficiency of the algorithm. On the other hand, for the model that optimizes for both precision and time (Model II), the final joint angles were $[93.2, 7.5, 34.7, 45.1]$ degrees, and the gripper’s position was 85 cm along the arm (Figure 13B). In this scenario, the MA operated efficiently, and only significantly made a change in the first joint, and it reached the target with an accuracy of 0.06 mm in a single second. This resulted in a notable time-save of approximately 5 s.

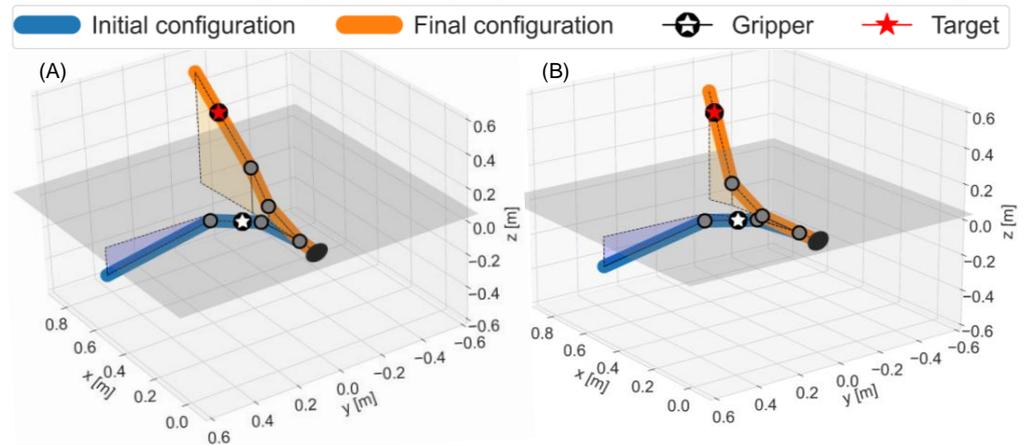


Figure 13. Starting with the non-zero initial configuration, the model with the time constraint only made a significant change in the first joint (B), whereas the model without time constraints made changes in all the joints (A). As a result, there was a substantial time-save of approximately 5 s.

5.2.4. Comparison to Inverse Kinematics (IK) Solver

We tested Model II against the Newton–Raphson (NR) solver from the Robotics-Toolbox [23] on 5000 trials, moving from random initial configurations q_{init} to desired MA positions x_{des} . These random configurations and positions were uniformly sampled from the 50,000 samples that were used to test the NN.

For each IK query, m solutions were generated using the NR. These m solutions were grouped into five sets, each solving for an arm with the first five k joints ($k = 1, \dots, 5$) and an additional prismatic joint simulating the MA at the final link, as illustrated in Figure 14. The solution with the minimum MA actuation time, τ , according to Equation (3), was selected from the m solutions. In addition to the positional error over the training and test data, Table 4 presents the mean runtime of the computer to generate the data presented in Table 3.

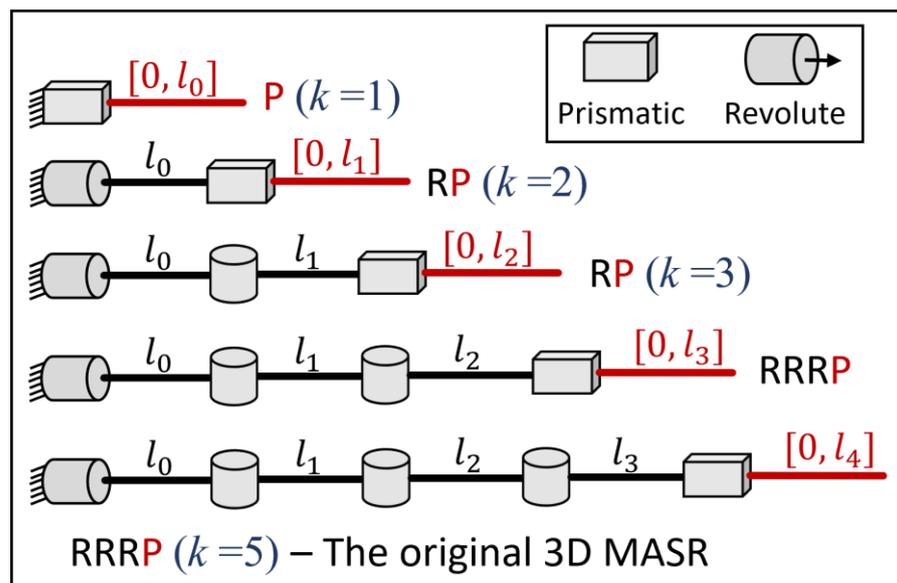


Figure 14. Illustration of the configurations for different arm setups ($k = 1, \dots, 5$). Each configuration shows the prismatic joint in red, simulating the MA at the last link for solving IK queries. These configurations were used to select the solution with the minimum actuation time from the NR solver’s generated solutions.

Table 4. Comparison of the NN IK model to the Newton–Raphson solver.

	Model II		Newton–Raphson	
	NN	NN + Jacobian	$m = 1000$	$m = 10,000$
Mean error \bar{e} (mm)	5.5	0.29	2.2	2.4
Mean time $\bar{\tau}$ (s)	7.3	7.91	8.5	8.2
$\bar{e} < 1$ mm	73	95	92	91
Computer runtime (s)	6×10^{-4}	1×10^{-3}	0.75	3.2

The NR-based IK was evaluated with $m = 1000$ and $m = 10,000$. The results for the numerical IK showed that the positional errors were higher than for the NN, but that the action time was not adequately minimized. In addition, the runtime for an IK query was at least two orders of magnitude longer than for the NN, leading to an excessively long planning time, which is not suitable for real-time applications.

6. Experiments and Results

This section presents some of the experiments conducted using a 3D-printed robotic prototype. The robot’s comprehensive functionality was tested in numerous experiments, encompassing activities such as reaching distinct points within a 3D space, utilizing the mobile actuator (MA) to grasp objects, transporting them along the links, and depositing them at designated target points. The following experiments exemplify the robot’s advantages for pick-and-place tasks, where their processes were pre-planned offline and executed automatically by the robot, as demonstrated in Video S1.

6.1. Joint Operation

The first experiment with this robot demonstrates the actuation of the joints using the MA. The experiment also measured the rotational and linear speeds of the joint actuation and the MA’s travel parameters, as recorded in Video S1 (see Figure 15). In step (A), the arm, starting from its initial position, was rotated by -90 degrees. In step (B), the robot is shown after the MA rotated the first joint around the x -axis by 90 degrees at 14 degrees per second. Note that the first joint can be rotated indefinitely since there is no wiring along the links. In step (C), the MA travelled along the arm’s length using the locomotive mechanism at 8 cm per second until it reached the third joint. In step (D), the MA rotated the third joint around the z -axis to an angle of 45 degrees at a speed of 14 degrees per second. In step (E), the third joint was returned to its original position, and the MA began advancing toward the prismatic joint, leading to the situation depicted in step (F). In step (G), the MA activated the prismatic joint, and shortened it by 10 cm at a speed of 8 cm per second. In step (H), the MA moved to the end of the arm and closed the gripper, as shown in Video S1.

This experiment highlights the robot’s ability to precisely control joint movements and the MA’s capability to traverse the arm and manipulate various joints effectively.

6.2. Simulation of Fruit Picking

This experiment mimicked picking fruit from a tree and placing it in a basket, as an example of one of the robot’s applications (see Figure 16). In (A), the arm is at the zero reference position in the initial condition. In (B), the MA rotates the first joint (x -axis rotation joint) by 35 degrees. The MA then moves toward the third and fourth joints (z -axis rotation joint) and rotates them to angles of 28 and 22 degrees, respectively, as seen in parts (C) through (E). In (F), the MA moves toward the end of the fifth link and closes the gripper on the hanging ball. In parts (G) and (H), while still gripping the ball, the MA moves backward to rotate the fourth and third joints to angles of -24 and -20 degrees, respectively. Finally, in (I), the MA reaches its target and releases the ball into the basket (see Video S1). The interest of this fruit-picking task lies in the fact that fruits usually grow in clusters, and thus require fewer joint actuations than conventional arms, and hence lower energy consumption.

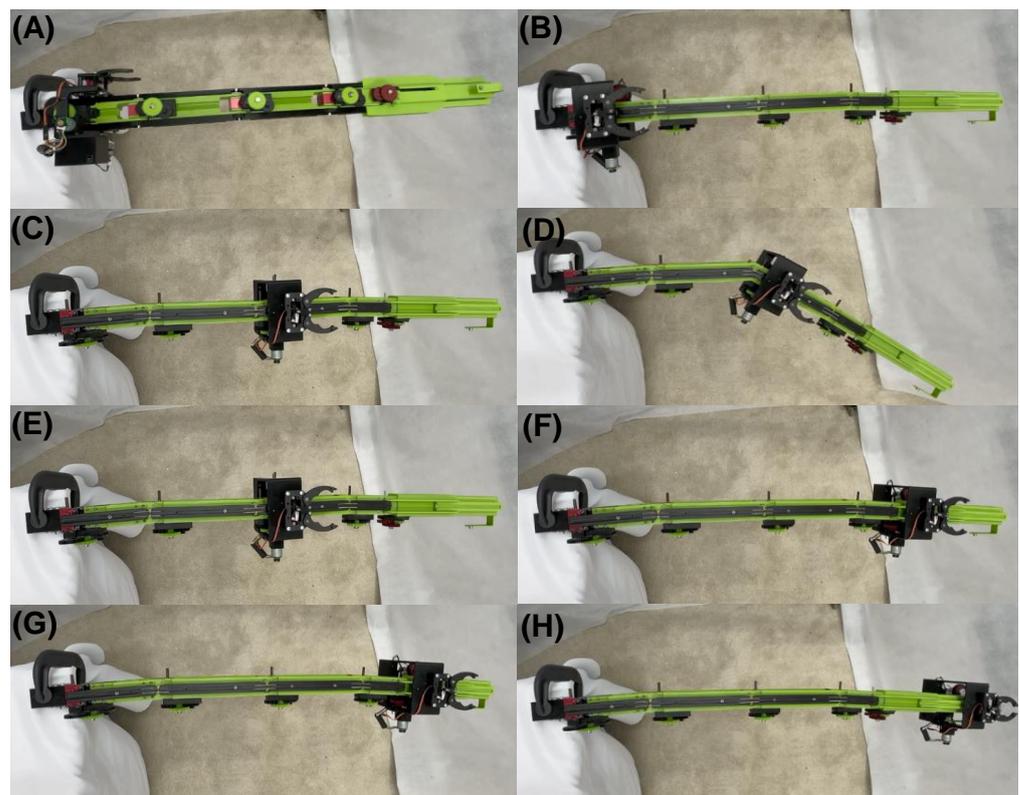


Figure 15. Stills of the joint operation of the 3D MASR. Starting in (A), the MA rotates the first joint (B), then advances to the center of the arm (C), rotates the joint (D), rotates it back (E), advances further (F), elongates the prismatic joint (G) and finally advances along the prismatic joint (H).

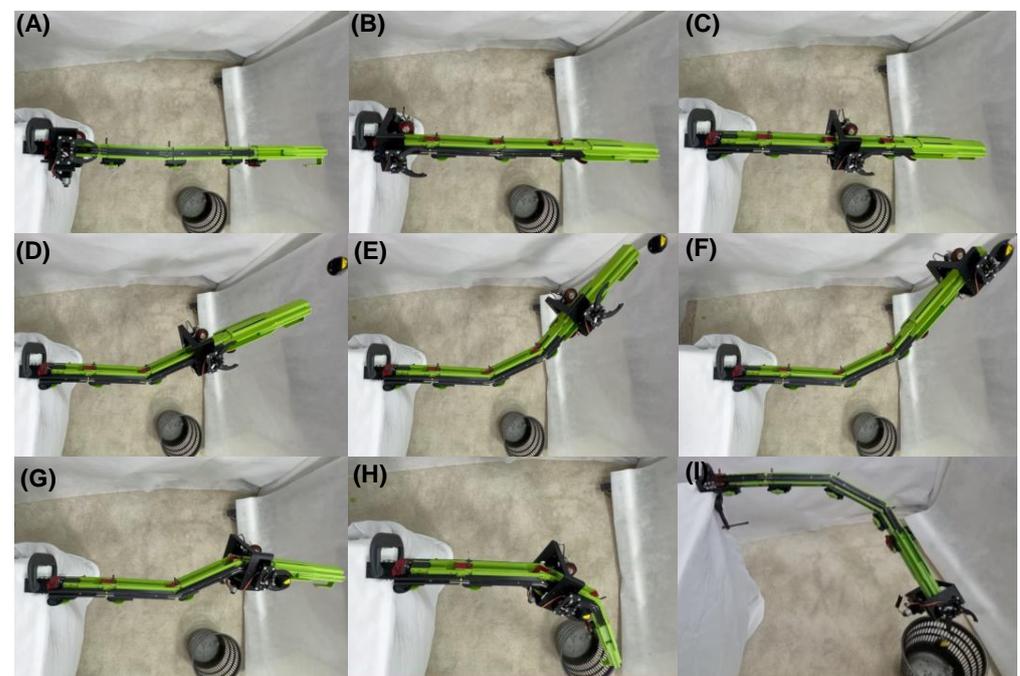


Figure 16. The 3D MASR picks a ball hanging above it, changes its configuration, and drops it into a basket. Starting in (A), the MA rotates the first joint (B), then advances along the arm (C), rotates the first joint (D), continues to advance along the arm (E), until it reaches its tip (F) and grabs a flexible ball, the MA returns (G), rotates the arm (H) and drops the ball in a bin (I).

6.3. Translating an Object along the Links

The next experiment (Figure 17) demonstrates the robot's ability to convey the travelling object along the arm from the arm's base to the last link without bending the arm, unlike in conventional arms. The movement begins at step (A). At this point, the mobile actuator closes its gripper on a ball suspended above it. Once the gripper grasps the ball, the mobile actuator moves to the third joint. It rotates this joint by 26 degrees, as shown in step (B). Next, in step (C), the mobile actuator moves to the last link to drop the ball into the target bowl. After the ball has been successfully dropped, the mobile actuator resets the arm to its initial position (D), as shown in Video S1.

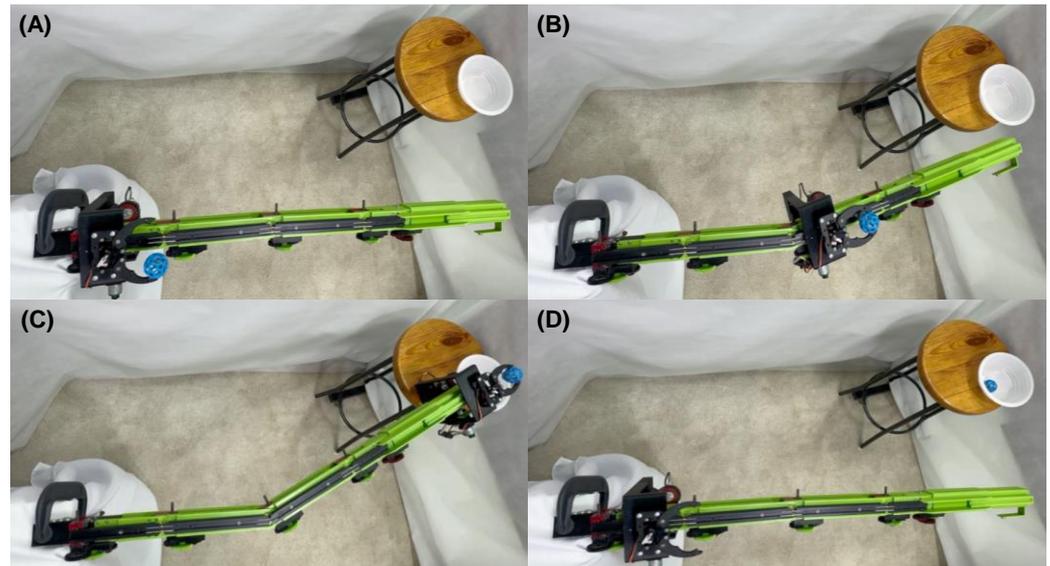


Figure 17. The 3D MASR picks a ball, translates it along the links, and drops it into a bowl. Starting in (A) while carrying a blue plastic ball, the MA advances to the center of the arm and rotates a joint (B), then continues towards a bowl (C), then it drops it and returns back to original position after rotating back the center joint (D).

7. Conclusions

This work introduced the 3D MASR, an innovative 3D minimally actuated serial robot designed for a variety of applications. This robot features a passive arm with passive joints and a mobile actuator (MA) with unique capabilities. The key highlights include the following:

- An MA that enables object manipulation and rapid translation along the arm.
- Enhanced reliability, reduced weight, and a simplified structure.
- Easy replacement and modification of links to cater to specific task requirements.
- Support for multiple MAs for increased flexibility.

To enable autonomous operation, we developed a smart control system that utilizes an encoder and magnetic sensor for automatic execution. Our design incorporates a direct kinematics model and a locomotion algorithm based on a time-based function optimized by a neural network. The primary goal is to minimize operation time and location error. Importantly, the use of the Jacobian matrix yielded a more accurate solution without significantly increasing operation time.

Our experimental prototype demonstrated proficiency in automatically performing pre-planned tasks, in particular when relocating objects with minimal operation time. While our 3D-printed version exhibited minor deformations due to flexibility and joint backlash, this can be mitigated by using composite materials.

However, it is essential to note that this robot operates more slowly than fully actuated serial arms, thus making it suitable for applications where high speed is not critical, such as space operations, agriculture, maintenance, painting, and search-and-rescue missions.

Our future work will focus on utilizing multiple MAs, enhancing the design, and developing advanced motion planning algorithms to handle obstacles and increase speed. We also intend to explore the collaborative utilization of multiple mobile robots to enhance overall performance and task efficiency.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/app14188204/s1>, Video S1: 3D MASR Minimally Actuated Serial Robot.

Author Contributions: Conceptualization, O.B. and D.Z.; methodology, O.B., A.C. and D.Z.; software, O.B. and A.C.; validation, O.B. and A.C.; formal analysis, O.B. and A.C.; investigation, O.B. and A.C.; writing—original draft preparation, O.B., A.C. and D.Z.; writing—review and editing, O.B., A.C. and D.Z.; supervision, D.Z.; funding acquisition, D.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported in part by the Israel Science Foundation (ISF, grant number 1843/23), the Pearlstone Center for Aeronautical Studies, and the Helmsley Charitable Trust through the Agricultural, Biological and Cognitive Robotics Initiative and the Marcus Endowment Fund, both at the Ben-Gurion University of the Negev.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the 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.

References

1. Mu, Z.; Zhang, L.; Yan, L.; Li, Z.; Dong, R.; Wang, C.; Ding, N. Hyper-Redundant Manipulators for Operations in Confined Space: Typical Applications, Key Technologies, and Grand Challenges. *IEEE Trans. Aerosp. Electron. Syst.* **2022**, *58*, 4928–4937. [[CrossRef](#)]
2. Arai, M.; Tanaka, Y.; Hirose, S.; Kuwahara, H.; Tsukui, S. Development of “Souryu-IV” and “Souryu-V”: Serially Connected Crawler Vehicles for in-Rubble Searching Operations. *J. Field Robot.* **2008**, *25*, 31–65. [[CrossRef](#)]
3. Ito, K.; Maruyama, H. Semi-Autonomous Serially Connected Multi-Crawler Robot for Search and Rescue. *Adv. Robot.* **2016**, *30*, 489–503. [[CrossRef](#)]
4. Alfalahi, H.; Renda, F.; Stefanini, C. Concentric Tube Robots for Minimally Invasive Surgery: Current Applications and Future Opportunities. *IEEE Trans. Med. Robot. Bionics* **2020**, *2*, 410–424. [[CrossRef](#)]
5. Han, S.; Chon, S.; Kim, J.; Seo, J.; Shin, D.G.; Park, S.; Kim, J.T.; Kim, J.; Jin, M.; Cho, J. Snake Robot Gripper Module for Search and Rescue in Narrow Spaces. *IEEE Robot. Autom. Lett.* **2022**, *7*, 1667–1673. [[CrossRef](#)]
6. Chitikena, H.; Sanfilippo, F.; Ma, S. Robotics in Search and Rescue (SAR) Operations: An Ethical and Design Perspective Framework for Response Phase. *Appl. Sci.* **2023**, *13*, 1800. [[CrossRef](#)]
7. Ota, T.; Degani, A.; Schwartzman, D.; Zubiate, B.; McGarvey, J.; Choset, H.; Zenati, M.A. A Highly Articulated Robotic Surgical System for Minimally Invasive Surgery. *Ann. Thorac. Surg.* **2009**, *87*, 1253–1256. [[CrossRef](#)] [[PubMed](#)]
8. Burgner-Kahrs, J.; Rucker, D.C.; Choset, H. Continuum Robots for Medical Applications: A Survey. *IEEE Trans. Robot.* **2015**, *31*, 1261–1280. [[CrossRef](#)]
9. Zhang, G.; Du, F.; Xue, S.; Cheng, H.; Zhang, X.; Song, R.; Li, Y. Design and Modeling of a Bio-Inspired Compound Continuum Robot for Minimally Invasive Surgery. *Machines* **2022**, *10*, 468. [[CrossRef](#)]
10. Seetohul, J.; Shafiee, M. Snake Robots for Surgical Applications: A Review. *Robotics* **2022**, *11*, 57. [[CrossRef](#)]
11. Fan, C.; Zhang, Y.; Xie, Z.; Liu, Y.; Li, Z.; Li, C.; Liu, H. Design of Space Robotic Arm-Hand System and Operation Research. In Proceedings of the 2018 IEEE International Conference on Mechatronics and Automation (ICMA), Changchun, China, 5–8 August 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 481–486.
12. Mu, Z.; Liu, T.; Xu, W.; Lou, Y.; Liang, B. Dynamic Feedforward Control of Spatial Cable-Driven Hyper-Redundant Manipulators for on-Orbit Servicing. *Robotica* **2019**, *37*, 18–38. [[CrossRef](#)]
13. Peng, J.; Xu, W.; Wang, F.; Han, Y.; Liang, B. A Hybrid Hand-Eye Calibration Method for Multilink Cable-Driven Hyper-Redundant Manipulators. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 5010413. [[CrossRef](#)]
14. Kislassi, T.; Zarrouk, D. A Minimally Actuated Reconfigurable Continuous Track Robot. *IEEE Robot. Autom. Lett.* **2020**, *5*, 652–659. [[CrossRef](#)]

15. Ayalon, Y.; Damti, L.; Zarrouk, D. Design and Modelling of a Minimally Actuated Serial Robot. *IEEE Robot. Autom. Lett.* **2020**, *5*, 4899–4906. [[CrossRef](#)]
16. Gal, Y.; Zarrouk, D. Task-Based Motion Planning Using Optimal Redundancy for a Minimally Actuated Robotic Arm. *Appl. Sci.* **2022**, *12*, 9526. [[CrossRef](#)]
17. Mueller, A. Modern Robotics: Mechanics, Planning, and Control [Bookshelf]. *IEEE Control Syst.* **2019**, *39*, 100–102. [[CrossRef](#)]
18. Wolovich, W.; Elliott, H. A Computational Technique for Inverse Kinematics. In Proceedings of the 23rd IEEE Conference on Decision and Control, Las Vegas, NV, USA, 12–14 December 1984; IEEE: Piscataway, NJ, USA, 1984; pp. 1359–1363.
19. Köker, R.; Öz, C.; Çakar, T.; Ekiz, H. A Study of Neural Network Based Inverse Kinematics Solution for a Three-Joint Robot. *Robot. Auton. Syst.* **2004**, *49*, 227–234. [[CrossRef](#)]
20. Gao, R. Inverse Kinematics Solution of Robotics Based on Neural Network Algorithms. *J. Ambient Intell. Humaniz. Comput.* **2020**, *11*, 6199–6209. [[CrossRef](#)]
21. Lu, J.; Zou, T.; Jiang, X. A Neural Network Based Approach to Inverse Kinematics Problem for General Six-Axis Robots. *Sensors* **2022**, *22*, 8909. [[CrossRef](#)]
22. Hassan, A.A.; El-Habrouk, M.; Deghedie, S. Inverse Kinematics of Redundant Manipulators Formulated as Quadratic Programming Optimization Problem Solved Using Recurrent Neural Networks: A Review. *Robotica* **2020**, *38*, 1495–1512. [[CrossRef](#)]
23. Corke, P.; Haviland, J. Not Your Grandmother’s Toolbox—The Robotics Toolbox Reinvented for Python. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi’an, China, 30 May–5 June 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 11357–11363.

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.