



Article An Innovative Optimization Design Procedure for Mechatronic Systems with a Multi-Criteria Formulation

Cuauhtémoc Morales-Cruz^{1,2}, Marco Ceccarelli² and Edgar Alfredo Portilla-Flores^{1,*}

- GIIM: Group of Research and Innovation in Mechatronics, Centro de Innovación y Desarrollo Tecnológico en Cómputo, Instituto Politécnico Nacional, Mexico City 07700, Mexico; cmoralesc0600@alumno.ipn.mx
- ² LARM2: Laboratory of Robot Mechatronics, University of Rome Tor Vergata, Via del Politecnico 1, 00133 Rome, Italy; marco.ceccarelli@uniroma2.it
- * Correspondence: aportilla@ipn.mx

Abstract: This paper presents an innovative Mechatronic Concurrent Design procedure to address multidisciplinary issues in Mechatronics systems that can concurrently include traditional and new aspects. This approach considers multiple criteria and design variables such as mechanical aspects, control issues, and task-oriented features to formulate a concurrent design optimization problem that is solved using but not limited to heuristic algorithms. Furthermore, as an innovation, this procedure address all considered aspects in one step instead of multiple sequential stages. Finally, this work discusses an example referring to Mechatronic Design to show the procedure performed and the results show its capability.

Keywords: mechatronics; design criteria; multi-objective optimization

1. Introduction

Mechatronics and Robotics are well-established engineering disciplines that were conceived as the integration of electronics and computing science with mechanical design. Currently, these disciplines are required to develop systems capable of dealing with complex tasks while improving system performance. Moreover, given current technological advances and the growth of system-level concepts, engineering design can consider new procedures to fulfill integration among all its components [1].

The integration of combined mechanical design and control optimization received attention from engineers and researchers to investigate and develop novel strategies. A concurrent design approach that worked on the improvement of the mechanical design toward reducing the complexity of a controller was proposed in [2], where the objective function was chosen to reduce the shaking force/moment. An extended version of this procedure is presented in [3], where the Design for Control (DFC) approach is introduced. DFC describes the importance of the simultaneous design of a controller and the mechanical subsystems. Robust design with robust control in [4] investigates the coupling between the mechanical system and control strategies for an electric DC motor using an optimization approach. A nested optimization scheme is described in [5] to solve the integrated design problem of mechanical and control of the motor driving system.

Two of the critical issues that guide the research in considering multiple aspects are reduction of energy consumption [6] and precision when developing a task [7]. In this context, a significant number of different approaches have addressed mainly these two problems by optimizing hardware, software, or a combination of these two. Hardware solutions involve a mechanical design to develop lightweight systems [8] and balanced multi-body systems with the aim of facilitating a control design algorithm. Software solutions deal with operation scheduling [9], operation control based on energy consumption [10] and accuracy [11]. Hardware and Software integration can be described as an initial step toward a use of dynamic models and suitable controls to include dynamics of a



Citation: Morales-Cruz, C.; Ceccarelli, M.; Portilla-Flores, E.A. An Innovative Optimization Design Procedure for Mechatronic Systems with a Multi-Criteria Formulation. *Appl. Sci.* **2021**, *11*, 8900. https:// doi.org/10.3390/app11198900

Academic Editors: Henrik Gordon Petersen and Kanstantsin Miatliuk

Received: 20 August 2021 Accepted: 20 September 2021 Published: 24 September 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 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/). system as for example in [12]. Moreover, an example of optimal concurrent formulation of mechanical characteristics and control strategies is presented in [13] by formulating a design problem as an optimization problem and solving it with both gradient-based and heuristic approaches. However, the concurrent consideration of multi-criteria aspects in the design of a synergistic combination of hardware and software can still be improved.

The research and inclusion of multiple criteria are also topics of interest in mechatronic design. For example, the Mechatronic Design Quotient (MDQ) is a multi-criteria index that presents a systematic methodology for mechatronic design [14]. This methodology considers seven main criteria: meeting the task requirements, component matching, efficiency, intelligence, reliability, controller friendliness, and cost. However, it performs the optimization process in two distinct stages to overcome numerical issues and require expert intervention when switching from these two stages. Concurrent optimization of the mechanical characteristics, control strategies, and selection of actuators is carried in [15] for industrial robots considering multiple criteria. Criteria considering workspace volume, manipulator dexterity, static efficiency, and stiffness are chosen to design the geometrical parameters of the mechanism for a robotic leg in [16].

As described above with different applications [2-16], there are several examples in the literature on integrating only traditional aspects in mechatronic design. However, they address the design problem across multiple stages that require user interaction to switch from one stage to another. Moreover, examples in the literature do not describe a procedure of formulating the problem from user requirements. Therefore, an intervention of a design expert is necessary in more than one stage. Some examples from the literature also show specific examples of merging those stages. However, they do not present how to extend that approach to other applications as a general formulation. This paper aims to introduce a Mechatronic Concurrent Design (MCD) procedure for mechatronic systems considering multiple criteria from traditional and recent aspects in a concurrent way to formulate a design problem as a constrained multi-objective optimization problem. This approach allows combining traditional and new aspects from different fields in one common representation that can be described as an upper-level interconnection of different subsystems. MCD is presented as a general formulation considering user requirements until the obtainment of a solution based on the abstraction of interconnected subsystems. The novelty of the proposed procedure is in an integration of multiple design stages in only one step. Furthermore, the main goal is to consider criteria from different areas of engineering in a concurrent way for a synergistic combination of hardware and software components. It also considers a heuristic-based approach to solve the formulated problem. However, different types of solvers can also be applied.

The following sections describe the steps to formulate a design problem using the MCD procedure. Section 2 describes the interpretation of user needs and expectations to characteristics and requirements of the procedure. Section 3 presents the traditional and new aspects that can be considered in the design problem and criteria evaluation. Section 4 presents the formulation of the procedure as a multi-objective optimization problem. Section 5 describes how to address the optimization problem using a heuristic approach. Finally, an example is discussed in Section 6 to show the procedure's capability with the synergistic integration of a mechatronic system.

2. Characteristics and Requirements

Design requirements are based on non-technical needs and expectations of a user, and they are often formulated in non-technical terms. Engineers and researchers identify those requirements in a design process and describe them using mathematical models. Figure 1 shows a scheme for interpreting and classifying user design requirements into specific topics that can be modeled mathematically. It also introduces the concept of strict and soft design requirements according to user description. This classification is not unique and must be adapted to specific characteristics of design problems by considering fundamental aspects of engineering, as pointed out in [17].



Figure 1. A scheme for modeling design requirements.

In an initial design stage, engineers will focus on solving strict design requirements related to user needs. Therefore, these requirements must be formulated properly so that a design procedure guarantees that they can be fulfilled. Strict design requirements will be selected to produce a feasible device that can be manufactured. Once it is feasible to be built, it also must be capable of doing the task that a user requires. For this reason, the first classification on the left side of Figure 1 is for obtaining a feasible and working device.

A first approach to formulate strict design requirements is by using inequality constraints. This approach can be applied directly to design parameters to bound them within practical upper and lower limits or indirectly by defining constraint functions in terms of the design parameters. In some cases, it is crucial to consider strong constraints to describe well-defined design prescriptions, and they can be formulated as equality constraints. For example, some mechanical aspects that are frequently described using equality or inequality constraints are related to stress, displacement, workspace, and mobility conditions.

On the other hand, soft design requirements in the right side of the scheme presented in Figure 1, are related to user expectations that are based on the system's desired behavior or performance. These requirements can be adjusted not only according to different applications but also they can change as the development of the design process evolves. In some cases, different user expectations are dependent of each other, and as a consequence, they can be described using a single expression. However, in general, these requirements can compromise with each other. This characteristic means that an improvement of one user expectation can negatively affect another user expectation. For this reason, a formulation for soft design requirements must be flexible enough to consider the trade-off among different user expectations.

Soft design requirements can be formulated using functions in terms of design parameters to evaluate desired criteria and constraints. It is possible to calculate a performance index for each selected criterion and to compare the results from all the computed solutions to select the best one. Some of the soft design requirements can be related to user expectations as for example reduction of energy consumption, increase of system performance, reduction of number of components, increase of safety, reduction of used materials, simplification of manufacturing process, and improvement of design cost, production, and operation.

From this context, the herein presented Mechatronic Concurrent Design procedure is based on a formulation of a design problem as a constrained multi-objective optimization problem. This approach can consider conveniently multidisciplinary aspects to provide a synergistic combination of solutions for the involved design requirements. User needs and expectations must guide a design process in Mechatronics. However, the influence of the user's expectations can be limited by a set of constraints coming from a specific design problem.

The next sections describe in detail the formulation of the herein proposed procedure in terms of mechatronic criteria and representation of the design problem as a constrained multi-objective optimization problem.

3. Design Problem and Criteria Evaluation

In the frame of Mechatronics Design, multiple criteria can be considered from different fields of engineering. It is essential to consider these criteria concurrently since all the hard-ware and software components interact in a successful mechatronic operation. Therefore, software and hardware components must work together in an integrated configuration that can be identified as a result of the following design procedure. Figure 2 shows a conceptual scheme that includes but is not limited to the presented topics from different fields for a general formulation of a design problem in Mechatronics. This figure includes the main traditional aspects of a mechatronic system and some emerging topics of the last years.

It is well recognized that mechanical aspects, control algorithms, and sensing are part of a mechatronic system. However, criteria from other fields can be included in the development of a final product such as finite element analysis (FEA), motion simulations aided by computers (CAD), ergonomics, manufacturing, energy consumption, and power supply considerations. Collaboration and communication of different mechatronic systems demand the inclusion of criteria coming from topics such as Artificial Intelligence (AI), Internet of Things (IoT), and cognitive sensing.



Figure 2. A scheme for conceptual design problem in Mechatronics.

All the aspects mentioned above in a conceptual design can be composed of several subtopics to express specific criteria. Table 1 lists some components of four traditional main aspects in Figure 2, where each one can be formulated using specific approaches. As an illustration of this situation, the mechanical topology aspect of a system, listed in Table 1, can be addressed by four different approaches as pointed out in [18], which are methods based on screw theory, methods based on the algebra structure of Lie group and the displacement sub-manifold, methods based on the theory of linear transformations and evolutionary morphology, and methods based on position-oriented-characteristic equations. These approaches focus on describing and on obtaining a solution of three aspects: number and type of pairs such as prismatic or revolute joints, number and type of links such as binary or ternary links, and connection relations between these two components. A specific research line has been addressing the development of topology applications. For example, in [19] with a topology synthesis procedure to provide a complete atlas, in [20] with a description of topology and kinematics of mechanisms to address topology synthesis and particularization of a solution automatically, and in [21] with simultaneous shape and topology optimization for planar mechanisms.

Control issues in Table 1 is another example of the variety of criteria formulation used in different methods for mechatronic systems control. Recent research in control theory focuses on the use of modern algorithms to compensate for the non-linearity of controlled system and robustness to deal with uncertainties as in [22], or the combination of different approaches to improve its performance such as in [23] with the use of passivity-based control formulation and PID controllers to strengthen the energy shaping function of mechanical systems. The main issue associated with the use of an independent PID

controller design is the tuning of proper parameters where it becomes a difficult task if the plant to be controlled exhibits nonlinear behavior as described in [24]. However, the implementation of PID controllers is still used extensively.

Sensing and task-oriented features in Table 1, as well as other topics, pointed out in Figure 2 can be included in a mechatronic design using a specific formulation for those criteria. However, these criteria are usually considered independently without taking into account the interaction with the other aspects. One way to consider all possible implications among different criteria of mechatronic design is to formulate a design problem as a constrained multi-objective optimization problem. Figure 3 shows a scheme of the elements that can be taken into account to formulate a design problem as an optimization problem. There are two main aspects to be considered in a problem modeling procedure, which are design requirements, as illustrated in Figure 1 and design criteria as showed in Figure 2. These two aspects are considered to formulate the constraints, design variables, and objective functions, which are the components of a constrained multi-objective optimization problem.

ation
ration
g
-

Table 1. Main topics for mechatronic systems in Figure 2.

Design variables in Figure 3 are the system parameters to be obtained as a result of the design procedure application. These parameters affect the behavior of a mechatronic system, so they are selected to ensure system response with the desired behavior. Most of them can only be modified in the design process, while some others can be adjusted when the system performs a desired task. For example, parameters for the sizing of a system vary while the design procedure takes place, but they are fixed when a solution is obtained. In contrast, design variables such as control algorithm parameters can be modified after the design procedure has finished making a fine-tune for the operation of a task.



Figure 3. A scheme of aspects for a multi-criteria design formulation.

It is essential to identify the strict design requirements presented in Figure 1 so that they are formulated in the problem modeling stage of Figure 3 mainly as constraints of the optimization problem to guarantee that they can be fulfilled. Therefore, strict design requirements are mainly part of variable bounds and constraint functions. On the other hand,

the problem modeling of objective functions is done by considering mainly the soft design requirements and design criteria. This classification of design requirements contributes to facilitating the modeling process to formulate a specific constrained multi-objective optimization problem that adequately represents a design problem in mechatronics. For example, two different approaches to classify and to formulate the energy consumption criterion listed in Table 1 are the consideration of an unlimited power supply or a limited one. In the first case, it is required to develop a repetitive task accurately. However, the user requires to reduce energy consumption since this task is carried out for long periods. In this example, energy consumption is considered a soft design requirement, while the error criterion is a strict design requirement. In the second case, a user may require that the system fulfill a minimum autonomous time of functionality. In contrast to the first example, the energy consumption criterion is formulated as a strict design requirement. Nevertheless, the Mechatronic Concurrent Design procedure guarantees to obtain solutions with different compromises, among other criteria. All of them are feasible solutions that satisfy the strict design requirements to develop a required task, and to select one solution that fulfills user expectations.

The computation strategy of the proposed Mechatronic Concurrent Design procedure is represented in the scheme of Figure 4. This procedure starts with the establishment of specifications of strict and soft design requirements, as shown in Figure 1. Then, a formulation of the design problem is carried out as described in Figure 3 to consider criteria from traditional and recent aspects presented in Figure 2.



Figure 4. A scheme for a Mechatronic Concurrent Design procedure using concepts in Figures 1–3.

The next stage in Figure 4 is the application of a solver to a constrained multi-objective optimization problem. The implemented numerical solver is independent of the problem formulation. Nevertheless, it is also an essential element of the Mechatronic Concurrent Design procedure since its functional operation is closely related to the formulated optimization problem's characteristics. The next step in Figure 4 is the application of a decision-maker to finally point out a specific solution that satisfies user expectations. This decision-maker can be configured before the execution of the solver to identify an optimal solution automatically. However, exploring different trade-offs among objective functions in a design problem leads to identify a proper configuration of the decision-maker. Once the solver finishes, the result is a set of optimal solutions that represent the best trade-off among all the formulated objective functions where all of them satisfy the problem constraints associated with the modeling of strict design requirements. After the execution of a solver, the decision-maker helps the user to identify a compromise among objective functions already obtained.

4. Formulation of Multi-Objective Design Problem

The conceptual scheme in Figure 4 represents a general structure for a formulation of a constrained multi-objective optimization problem within the framework of Mechatronic Concurrent Design that can be expressed in the form

$$\min_{\boldsymbol{p}\in\mathbb{R}^{q}} \boldsymbol{\Phi}(\boldsymbol{x},\boldsymbol{P},t) = [\Phi_{1},\Phi_{2},\ldots,\Phi_{m}]^{T}$$
(1)

with

$$\Phi_i = \int_{t_0}^{t_f} F_i(\mathbf{x}, \mathbf{P}, t) \, dt, \qquad i = 1, 2, \dots, m \tag{2}$$

$$\boldsymbol{x} = [x_1, x_2, \dots, x_n]^T \qquad \qquad \boldsymbol{x} \in \mathbb{R}^n$$
(3)

$$\boldsymbol{P} = [p_1, p_2, \dots, p_q]^T \qquad \boldsymbol{P} \in \mathbb{R}^q \qquad (4)$$

subject to

$$\boldsymbol{G}(\boldsymbol{x},\boldsymbol{P},t) \leq 0 \qquad \qquad \boldsymbol{G} \in \mathbb{R}^r \tag{5}$$

 $H(\mathbf{x}, \mathbf{P}, t) = 0 \qquad \qquad H \in \mathbb{R}^s \qquad (s < q) \tag{6}$

$$p_j^L \le p_j \le p_j^U \qquad \qquad j = 1, 2, \dots, q \qquad (7)$$

and by the system characteristics

$$\dot{\mathbf{x}} = f(\mathbf{x}(\mathbf{u}, \mathbf{P}, t), \mathbf{u}(\mathbf{P}, t), \mathbf{P}, t)$$
(8)

$$\boldsymbol{x}(t_0) = \boldsymbol{x}_0 \tag{9}$$

where Φ is the objective function vector and its components; Φ_m is the *m*th objective function of a selected criterion; *t* is the time variable; t_0 is the initial time and t_f is the final time of the operation range to be considered for a system evaluation; F_i is a function of a performance criterion in terms of equations of motion, parameters and time variable; *x* is the state-space vector of a dynamic model of a mechatronic system under design with *n* state variables; *P* is the design vector with *q* design parameters; *G* is the vector of inequality constraints with *r* elements; *H* is the vector of equality constraints with *s* elements; p_j^L and p_j^U are the lower and upper bounds of the *j*th design parameter; \dot{x} is the gradient state-space vector; x_0 is the initial condition vector of the dynamic model; and *u* is the signal control vector.

Figure 5 illustrates a representation of the mathematical formulation with Equations (1)–(9) for those cases that can be represented graphically. In general, there are two main spaces for the formulation of an optimization problem, the decision or parameter space,

and the search space, also called as the objective space. A design vector P contains all the design parameters such as dimensions, masses, control parameters, and operation measurements. The parameter space, labeled as A in Figure 5 is defined by the bounds of design variables in Equation (7).



Figure 5. A scheme for a graphical representation of the optimization problem in terms of: (**A**) parameters space in Equation (7); (**B**) feasible space in Equations (5)–(9); and (**C**) objective space in Equation (1).

The feasible space labeled as B in Figure 5, corresponds to all P candidate vectors that satisfy constraints described in Equations (5)–(9), which represent strict design requirements. Inequality constraints G in Equation (5) delimits larger regions in the feasible space, and some examples of this type of constraints in a mechatronic design are associated with kinematic characteristics, working operation regions, and limiting conditions. Equality constraints H in Equation (6) describe strict requirements and can represent for example mobility conditions in a design problem. The system's characteristics are described with a dynamic model in Equation (8) and its initial conditions in Equation (9) which also constrains the optimization problem. The state variables of this dynamic model represent, for example, position, velocity, acceleration, electrical current, or voltage characteristics. The dynamic model is affected by design parameters and inputs for actuators where the signal control vector u is a function in terms of design parameters. In this case, those parameters represent the gains of a selected control algorithm. Elements of the subset B are solutions that fulfill the strict design requirements formulated as in Figure 3.

The objective function space, labeled as C in Figure 5, corresponds to $\Phi_1 : \mathbf{P} \in \mathbb{R}^q \to \mathbb{R}^1, \ldots, \Phi_m : \mathbf{P} \in \mathbb{R}^q \to \mathbb{R}^1$ which maps a decision space \mathbf{P} in \mathbb{R}^m . It is convenient to formulate Φ_i as a convex function by using a square form of F_i before integrating this value within a numerical procedure. This step does not imply necessarily that the optimization problem will be convex after considering the constraints as pointed in [25]. The mapping evaluation from a decision space to an objective space only takes place when a candidate solution fulfills all the constraints in Equations (5)–(9).

There are several approaches to order the elements of the search space but the concept of Pareto dominance is of fundamental importance for multi-objective optimization, as it allows us to compare two solutions without adding any additional information as described in [26]. Then, solutions in the search space are ordered using the Pareto approach where given two vectors in the objective space, that is $\mathbf{\Phi}^{(1)} \in \mathbb{R}^m$ and $\mathbf{\Phi}^{(2)} \in \mathbb{R}^m$, then the point $\mathbf{\Phi}^{(1)}$ is said to Pareto dominate the point $\mathbf{\Phi}^{(2)}$ if and only if $\forall j \in \{1, \dots, m\} : \mathbf{\Phi}_i^{(1)} \leq \mathbf{\Phi}_i^{(2)}$

and $\exists k \in \{1, ..., m\}$: $\mathbf{\Phi}_k^{(1)} \leq \mathbf{\Phi}_k^{(2)}$. This means that one objective function value can not be improved without degrading the value of another objective function. Figure 5 shows a Pareto dominance region of the vector $\mathbf{\Phi}^{(1)}$ as represented in a 2D search space using a darker color. Therefore, the best trade-off among different criteria corresponds to solutions that are part of the Pareto front.

In summary, a design problem in the frame of Mechatronics can be formulated as a constrained multi-objective optimization problem using the proposed Mechatronic Concurrent Design procedure in terms of Equations (1)–(9). This approach provides a feasible way to consider strict and soft design requirements of a system under design as functions of optimality criteria, likewise referring to the general concepts in Figures 1–4, considering not only traditional mechatronic aspects.

5. Issues for Numerical Solutions

In general, numerical solving techniques are independent of the formulation of an optimization problem. However, a proper solver to address an optimization problem can be closely related to its characteristics in such a way that the structure of an optimization problem can suggest convenient solving algorithms.

Deterministic approaches that are based on gradient calculations impose rigorous conditions over a mathematical formulation structure, so the original problem might be reformulated for a convenient representation to apply a specific numerical solving technique as described in [27]. The main advantage of a deterministic solver is that once all algorithm requirements are fulfilled, it is possible to guarantee a locally optimal solution. Another option is the use of a heuristic technique that does not require any reformulation and can be applied directly to the formulated problem [28].

There are multiple options to select a solver algorithm that can fit an optimization problem structure. The development of new algorithms is still a topic of interest, as pointed out in [28]. This work uses the Differential Evolution (DE) algorithm proposed in [29] along with the feasibility rules by Deb as a constraint manager [30], and the use of Pareto-dominance approach for solving the presented optimization problem. This research selected DE as a first option because it is a popular heuristic algorithm extensively treated in the literature and is an excellent alternative to solve optimization problems formulated from real-world design problems described in [31]. However, other heuristic algorithms or any combination with gradient-based algorithms can be used to solve the optimization problem. Among other advantages, DE has a a few parameters compared to other heuristic algorithms and its implementation is fairly simple.

DE heuristic algorithm is a continuous stochastic search method that is based on evolution strategies [29]. The implementation of this heuristic method is represented with a scheme in Figure 6 and a detailed computation routine is shown in Algorithm 1. An initial population is generated using a uniform random distribution within the bounds that are described in Equation (7) or within the parameter space A as in Figure 5, where the elements of the generated candidates correspond to the design variables. The external memory data base (EMdb) corresponds to a variable that stores all the feasible non-dominated solutions, which this data base is initialized with an empty value. The next step is to evaluate all the initial population candidates in the objective functions, inequality, and equality constraints of Equations (1)–(7).

The crowding distance of an arbitrary point in the search space's domain corresponds to an estimation of the distance from the nearest neighbors along each of the objectives to measure the population's density. The higher the crowding distance, the higher is its isolation. Therefore, when the crowding criterion in Figure 6 is active, the algorithm prioritize the search around isolated solutions to raise the number of candidates, as pointed out in [28]. A crowding criterion selects the percentage of the total number of iterations computed using individuals from the current population or EMdb. Therefore, a crowding parameter (DE_{CRW}) tunes the algorithm's behavior to explore the search space or to exploit the search near to a found candidate that remains distant from other found solutions.

DE algorithm is based on the use of two operators, which are the mutation and crossover. These two operators are used to search in the parameter space, and they are applied for every design variable. The Crossover operator in Figure 6 determines the probability of selecting a value that can be computed using the mutation operator or the value from a previous iteration as in line 15 of the pseudo-code in Algorithm 1. In particular, a crossover value close to 1 indicates a higher probability of considering the value from a mutation process, and a value close to zero indicates a higher probability of using values from the previous iteration. A crossover parameter is compared against a value from a continuous random distribution between one and zero. Additionally, a discrete uniform distribution can be used to guarantee that at least one component is different from the previous vector. The mutation operator in Figure 6 is used after selecting three different candidates from a population in line 16 of Algorithm 1, which defines a search direction and a step size to move in that direction. DE operators are controlled by the crossover parameter as $(DE_{CR} \in [0,1])$ and the mutation parameter as $(DE_F \in [0,1])$. A crossover parameter can be selected at the beginning of execution and remains constant while the mutation factor is selected randomly for every design variable.



Figure 6. A flowchart of a DE implemented algorithm.

Algorithm 1: Pseudo code of a multi-objective DE implementation.
1 $It = 0$
² Initialize the external memory ${}^{lt}EMdb = \varnothing$
³ create a random initial population $I^{t}x_{i} \forall i = 1,NI$
4 evaluate objective functions ${}^{It} \mathbf{\Phi} = \mathbf{\Phi}({}^{It} \mathbf{x}_i)$
5 evaluate constraints ${}^{It}G = G({}^{It}x_i)$
6 evaluate non-dominated rules in search of feasible-non-dominated solutions to add to the
EMdb
7 for $It := 1$ to It_{max} do
s for $i := 1$ to NI do
9 if $DE_{CRW}It_{max} > It \& {}^{It}EMdb \ge 3$ then
10 select from ^{<i>It</i>} <i>EMdb</i> the three individuals with higher crowding value
$\{r_1 \neq r_2 \neq r_3\}$
11 else
select randomly $\{r_1 \neq r_2 \neq r_3\}$ from current population
13 end
14 for $j := 1$ to NV do
15 If $U(0,1) < DE_{CR} j = U\{0,NI\}$ then
16 $I^{II+1}v_{ij} = I^{I} x_{r_1j} + DE_F(I^{I} x_{r_2j} - I^{I} x_{r_3j})$
17 else
$18 \qquad \qquad It+1v_{ij} = It-1 x_{ij}$
19 end
20 end
evaluate objective functions ${}^{It} \mathbf{\Phi} = \mathbf{\Phi}({}^{It} v_i)$
evaluate constraints ${}^{It}G = G({}^{It}v_i)$
evaluate non-dominated rules in search of feasible-non-dominated solutions to add
to the <i>EMdb</i>
24 end
verify non-dominance among existing feasible-non-dominated solutions and recent
added candidates in the <i>EMdb</i>
26 end

Once the candidates of a population are generated according to the DE algorithm approach, all the candidates are evaluated using objective functions, inequality, and equality constraints of Equations (1)–(7). The last computations in Figure 6 are executed to verify the Pareto dominance of the candidates versus the current population in order to update the EMdb and to verify the same rules for all that stored in the EMdb. The stop criterion is a maximum number of iterations that can be fixed as an additional algorithm parameter, and this procedure is executed until that value is reached.

Algorithm 1 list the computations that are programmed in a routine to solve the formulated multi-objective optimization problem of Equations (1)–(9) using the above mentioned steps as summarized in the flowchart of Figure 6.

6. An Illustrative Example

The feasibility and characteristics of the proposed multi-criteria design procedure are discussed in this section. It is combined with a solving algorithm from heuristic techniques to illustrate the implementation and results of an example using a four-bar linkage design problem.

The dynamics of a closed-loop linkage is usually highly nonlinear due to the asymmetry of the geometrical structure. Thus, it represents a challenge to consider traditional design requirements and even more with the inclusion of other criteria. The four-bar mechanism has a significant number of applications in many different engineering solutions [32], and researchers keep studying it as, for example, in [33] with a review of some mechanical applications. However, currently user needs require that this mechanism accomplish more challenging tasks and fulfill user expectations in a better way. A design solution of a four-bar mechanism is reported as an example, considering design requirements as illustrated in Figure 1 to formulate a multi-objective optimization problem as in Figures 3 and 4. A scheme of a four-bar linkage with design parameters is shown in Figure 7. Table 2 summarizes the specific expressions for the design problem with Equations (1)–(9) that are formulated for the reported example. The parameters L_i , lc_i , θ_i , m_i , J_i represent dimension of links, position of mass center, angle respect the framework *xoy*, deviation angle for mass center, link masses, and inertia moments, respectively; t_r is the rise-time; t_p is the time at which the peak value occurs; θ_{2d} is the target velocity; u is the driving torque generated by a PID control algorithm; k_p , k_i , k_d are proportional, integral and derivative gains, respectively; and state variables of the dynamic model are $x_1 = \theta_2$ angular position and $x_2 = \theta_2$ angular velocity of the input link. From this context, the different nature of all the parameters included in the parameter vector can be seen. Therefore, it shows the capability of the proposed procedure to concurrently address in one stage the interconnection of multiple subsystems that are usually taken into account



in multiple stages using a sequential approach.

Figure 7. A scheme with parameters for a case of study in designing a four-bar linkage.

Strict design requirements are modeled as described in Figure 3 as constraints of the optimization problem. In this example those requirements are modeled as inequality constraints (*G*) and parameter bounds as listed in Table 2. As a design requirement, a kinematic configuration of the four-bar linkage must satisfy that the input link can rotate continuously, and the rocker must be located in the fourth link. These strict requirements are formulated as constraints g_1 to g_4 in Table 2, which represent the Grashof condition. Other user needs are a rise-time of 0.1 seconds and a maximum torque value of 300 *Nm*. Therefore, constraints g_5 and g_6 guarantee the fulfillment of those strict design requirements. The last three constraints, g_7 to g_9 , ensure that the position of the mass center stays within a position between the link-joints to prevent complex shapes for the links.

$\Phi_i \mid i = 1 \dots 3$	$x \in \mathbb{R}^2$	$P\in\mathbb{R}^{20}$	$oldsymbol{G} \in \mathbb{R}^{10}$	Parameter Bounds		
	$x_1 = \theta_2$	$p_{1,2,3,4} = L_{1,2,3,4}$	$g_1 = p_2 \le p_4$	$0.1 \le p_{1,2,3,4} \le 1 \text{ [m]}$		
$\Phi_1 = \int_{t_0}^{t_f} e^2 dt$	$x_2 = \dot{\theta_2}$	$p_{5,6,7} = l_{c_{2,3,4}}$	$g_2 = p_4 \le p_1$	$0 \le p_{5,6,7} \le 1 [m]$		
		$p_{8,9,10,11} = \theta_1, \varphi_{2,3,4}$	$g_3 = p_1 \le p_3$	$-\pi \le p_{8,9,10,11} \le \pi \text{ [rad]}$		
with: $e = \dot{\theta}_{2_d} - \dot{\theta}_2$		$p_{12,13,14} = m_{2,3,4}$	$g_4 = p_3 + p_2 \le p_4 + p_1$	$0.5 \le p_{12,13,14} \le 2$ [kg]		
		$p_{15,16,17} = J_{2,3,4}$	$g_5 = t_r - 0.1 \le 0$	$0.01 \le p_{15,16,17} \le 1 [\mathrm{kgm^2}]$		
4		$p_{18,19,20} = k_p, k_i, k_d$	$g_6 = u_{max} - 300 \le 0$	$0 \le p_{18,19,20} \le 15$ [–]		
$\Phi_2 = \int_{t_0}^{t_f} (u\dot{\theta}_2)^2 dt$			$g_7 = p_5 \le p_2$			
			$g_8 = p_6 \le p_3$	Control algorithm		
			$g_9 = p_7 \le p_4$			
$\Phi_3 = \dot{\theta_2}(t_p) - \dot{\theta_{2_d}}$				$u = k a \pm k \int a dt \pm k \dot{a}$		
				$u = \kappa_p c + \kappa_l \int_{t_0} cul + \kappa_d c$		
			$H\in \mathbb{R}^{0}$			

Table 2. Summary of data for the optimization problem formulation for the case of study in Figure 7.

Design variables, as illustrated in Figure 3 are those parameters obtained as a result of solving the formulated optimization problem. Table 2 lists design parameters (P) and its bounds referring to the design problem for the four-bar linkage in Figure 7. Table 3 lists the obtained values of a selected solution and classifies them. There are four parameters for a kinematic synthesis of the four-bar linkage, twelve parameters that tune its dynamic behavior, and three parameters that tune a control algorithm. These three synthesis processes are carried out concurrently in order to fulfill strict and soft design requirements.

Table 3. Numerical values of design parameters for an optimal solution for the case of study in Figure 7 and Table 2.

p_i	p_1	p_2	p_3	p_4	p_8	p_5	p_6	<i>p</i> ₇	p_{12}	p_{13}
Design variable	L_1	L_2	L_3	L_4	θ_1	l_{c_2}	l_{c_3}	l_{c_4}	m_2	m_3
Variable type	Kinematics					Dynamics				
Optimal	0.6497	0.1000	1.0000	0.6497	-2.8679	0.0010	0.8464	0.0018	2	0.5
Units	m	m	m	m	rad	m	m	m	kg	kg
p_i	p_{14}	p_{15}	p_{16}	<i>p</i> ₁₇	p_9	p_{10}	p_{11}	p_{18}	p_{19}	p_{20}
Design variable	m_4	J ₂	J ₃	J_4	φ_2	φ_3	$arphi_4$	k _p	k_i	k _d
Variable type	Dynamics					Control Gains				
Optimal	2	0.0100	0.0100	0.0100	1.8099	-0.9406	2.3051	15	15	10
Units	kg	kgm ²	kgm ²	kgm ²	rad	rad	rad	-	_	_

Soft design requirements and selected criteria, as illustrated in Figure 3 are formulated into three objective functions that are listed in Table 2. The first objective function (Φ_1) corresponds to a system performance criterion when doing a task. It measures the accumulated error generated by the system, and the target is a constant angular velocity of 30 rad/s. User expectation referring to a reduction of energy consumption is formulated using the second objective function (Φ_2) in Table 2. This criterion measures the accumulated power required by multiplying the driving torque and the angular velocity of the input link for the four-bar linkage in Figure 7. Another user expectation considered in this example is the reduction of overshoot when doing the selected task of reaching a constant angular velocity. The third objective function (Φ_3) formulates the reduction of overshoot as a soft design requirement in Table 2. The evaluation of these three criteria and constraints are computed after solving the dynamic model in Equation (10) as referring to Equations (8) and (9) for a four-bar mechanism. The motion equations are computed for an evaluation period of two seconds considering values in Table 2.

$$A(\theta_2)\dot{\theta}_2 + B(\theta_2, \dot{\theta}_2)\dot{\theta}_2 + C(\theta_2)g = u(P)$$
(10)

The solving process of the constrained multi-objective optimization process is carried out as illustrated in Figure 4 using the algorithm described in Figure 6 and Algorithm 1. The parameters of the implemented heuristic algorithm are a crossover range of $DE_{CR} \in [0.7, 0.9]$, a mutation range of $DE_F \in [0.1, 0.8]$, a crowding factor of $DE_{CRW} = 0.6$, a population of $DE_{NI} = 1000$ individuals, and a maximum number of iterations of $DE_{IT_{max}} = 1000$. The algorithm was executed 10 independent times. The results are illustrated in Figure 8, with the Pareto front after filtering the 10 executions. This figure illustrates the distribution of the obtained solutions and one solution is selected as the illustrative example to be analyzed in this section.



Figure 8. Pareto front in terms of the three objective functions for a feasible solution for the example in Figure 7.

In the initial stage of the algorithm execution, a random population is generated, as described in Algorithm 1 within the parameter space illustrated in Figure 5. At this point, the heuristic algorithm focuses on moving all the individuals in the population to the feasible space illustrated in Figure 5. This process is mainly related to finding feasible solutions that fulfill strict design requirements by using the constraint manager of the implemented heuristic algorithm. Figure 9 illustrates how the Grashof condition and the center of mass position (g_4 and g_7) are not fulfilled for the first iterations, and these values change from one unfeasible value to another every iteration. Once the heuristic algorithm finds feasible solutions that fulfill the constraints ($g_i \leq 0$), the frequency change of these values decreases dramatically, and they do not go back to non-feasible values.

Figure 10 shows the convergence of the solving process in terms of the number of candidates that fulfill all the strict design requirements formulated as inequality constraints in Equation (5) with data in Table 2. At the start of the heuristic algorithm execution, all individuals in the population are un-feasible. However, in about 100 iterations, all the candidate solutions fulfill the constraints. This characteristic means that all the individuals



in the population now satisfy the strict design requirements that ensure that the solutions can be built and develop the requested task.

Figure 9. Computed convergence of the solving process in terms of the Grashof constraint and mass center position (g_4 and g_7 in Table 2).



Figure 10. Computed convergence of the solving process in terms of the number of candidate solutions that fulfill strictdesign requirements for the first 200 algorithm iterations ($G \le 0$ in Table 2).

Once the heuristic algorithm has found feasible solutions, the solvers focus on finding solutions that improve the objective function values but at the same time ensures that the constraints are fulfilled. This task means that to find solutions that improve soft design requirements associated with user expectations. Figure 11 shows the computed convergence of the solving process for the criteria associated with the task performance and overshoot criterion in Table 2. At the beginning of the algorithm execution, the objective function's values are not registered since those values can be higher or lower for nonfeasible solutions. The algorithm function manager starts making decisions only after the individuals in the population are feasible. Therefore, the decreasing behavior of the objective functions corresponds to the feasible solutions presented in Figure 10. This figure illustrates that once feasible individuals in the population appear, it is possible to find configurations that improve overshoot criterion with fewer iterations than the first objective function. Figure 12 shows the convergence of the objective function associated with energy consumption. This figure shows that the search space is huge for this criterion since its evaluation can produce high values compared to the last registered values at the end of



the algorithm execution. It might represent an issue for the numerical solver that can be improved.

Figure 11. Computed convergence of the solving process in terms of Φ_1 (task performance) and Φ_3 (overshoot), similar to Figure 10.



Figure 12. Computed convergence of the solving process in terms of Φ_2 (energy consumption), similar to Figure 10.

Another essential aspect that can be analyzed is the convergence of the design parameters as the solving process takes place. Figure 13 shows the computed convergence in terms of the input link size parameter and its mass, similar to Figure 10. The value of this parameter changes rapidly from one iteration to another while there are no feasible individuals. However, when feasible individuals appear in the population, the frequency change of this value is reduced. This figure illustrates how the algorithm explores the parameter space in search of feasible configurations and the improvement of objective functions in those different stages.

Figure 14 shows the obtained behavior of the selected solution to be analyzed using the computed dynamic model. This figure shows the behavior of the input link angular velocity $x_2 = \dot{\theta}_2$ where it can be seen the performance of the system. There are still notable variations in comparison with the target velocity, and this characteristic also shows the high non-linearity of the study case. Nevertheless, the obtained response satisfy strict design requirements and an optimal trade-off with soft design requirements for the selected solution. Finally, Figure 15 shows the behavior of the driving control generated by the



PID control algorithm, where it can be seen that the amplitude of the input torque is small enough to keep the power consumption within low levels.

Figure 13. Computed convergence of the solving process in terms of the input link size parameter and its mass, L_2 and m_2 in Table 2 for example in Figure 7.



Figure 14. Computed dynamic model for the optimal solution for the case in Figure 7 in terms of angular velocity.



Figure 15. Computed driving torque generated by the tuned control algorithm for the case of Figure 7.

A direct comparison with results in the literature is not the aim for this paper. However, one comparison is discussed with the MCD obtained solution that improves the presented solution in [2] by 53% in terms of the objective function to evaluate performance of a solution (Φ_1) in Table 3. That solution in the literature corresponds to the application of mass redistribution according to complete shaking force balancing and partial shaking moment balancing. This result means that performance can be improved when considering more design aspects and concurrently solving the design problem.

7. Conclusions

This paper presents the Mechatronic Concurrent Design procedure to address traditional and new aspects of mechatronic design. This approach considers different criteria and design requirements concurrently by formulating a design problem as a constrained multi-objective optimization problem with multidisciplinary criteria more than from mechatronics. The innovation of the proposed procedure is the integration of multiple design stages in only one step. The MCD procedure is formulated to address high non-linear systems whose design solution requires a convenient use of heuristic techniques. It also contributes to facilitating the application of this procedure to non-continuous design parameters or the use of non-analytically modeling of a mechatronic system. An illustrative example shows the feasibility and capability of this approach's concepts and numerical characteristics to validate the presented procedure and clarify its application for mechatronic designs. The example shows how to include different criteria aspects such as performance task, energy consumption, and overshoot reduction in a synergistic combination to address soft design requirements. It also considers strict design requirements such as a possible kinematic configuration, rise time, maximum torque, and preferable mass location. Future work includes applying this procedure with different design examples that also consider criteria from other fields to make MCD a practical approach in formulating mechatronic design problems. Since the solution is related to both design problems and solvers, it is also essential to implement different solvers.

Author Contributions: Conceptualization, C.M.-C., M.C., and E.A.P.-F.; methodology, C.M.-C., M.C., and E.A.P.-F.; software, C.M.-C.; experiments, C.M.-C.; data curation, C.M.-C.; writing—original draft preparation, C.M.-C.; writing—review and editing, M.C. and E.A.P.-F.; supervision, M.C. and E.A.P.-F. 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.

Acknowledgments: The first author acknowledges Consejo Nacional de Ciencia y Tecnología and Instituto Politécnico Nacional for supporting his period of study at LARM2 of Rome Tor Vergata University in the A.Y. 2019–2021 within a double Ph.D. degree program.

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

References

- Bradley, D.; Russell, D.; Hehenberger, P.; Azorin-Lopez, J.; Watt, S.; Milne, C., From Mechatronics to the Cloud. In *Reinventing Mechatronics: Developing Future Directions for Mechatronics*; Yan, X.T., Bradley, D., Russell, D., Moore, P., Eds.; Springer: Cham, Switzerland, 2020; pp. 17–33. [CrossRef]
- Zhang, W.J.; Li, Q.; Guo, L.S. Integrated design of mechanical structure and control algorithm for a programmable four-bar linkage. *IEEE/ASME Trans. Mechatron.* 1999, 4, 354–362. [CrossRef]
- 3. Li, Q.; Zhang, W.J.; Chen, L. Design for control-a concurrent engineering approach for mechatronic systems design. *IEEE/ASME Trans. Mechatron.* **2001**, *6*, 161–169. [CrossRef]
- 4. Alyaqout, S.F.; Papalambros, P.Y.; Ulsoy, A.G. Combined Robust Design and Robust Control of an Electric DC Motor. *IEEE/ASME Trans. Mechatron.* **2011**, *16*, 574–582. [CrossRef]
- Zeng, T.; Ren, X.; Zhang, Y.; Li, G.; Na, J. An Integrated Optimal Design for Guaranteed Cost Control of Motor Driving System With Uncertainty. *IEEE/ASME Trans. Mechatron.* 2019, 24, 2606–2615. [CrossRef]

- Brossog, M.; others. Reducing the energy consumption of industrial robots in manufacturing systems. *Int. J. Adv. Manuf. Technol.* 2015, 78, 1315–1328.
- Kayacan, E.; Chowdhary, G. Tracking Error Learning Control for Precise Mobile Robot Path Tracking in Outdoor Environment. J. Intell. Robot. Syst. 2019, 95, 975–986. [CrossRef]
- 8. Loughlin, C.; Albu-Schäffer, A.; Haddadin, S.; Ott, C.; Stemmer, A.; Wimböck, T.; Hirzinger, G. The DLR lightweight robot: Design and control concepts for robots in human environments. *Ind. Robot. Int. J.* **2007**, *34*, 376–385.
- Fahmy, S.A. Optimal design and scheduling of cellular manufacturing systems: An experimental study. In Proceedings of the 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Budapest, Hungary, 9–12 October 2016; pp. 004532–004537.
- 10. Minchala-Avila, L.I.; Garza-Castañón, L.E.; Vargas-Martínez, A.; Zhang, Y. A Review of Optimal Control Techniques Applied to the Energy Management and Control of Microgrids. *Procedia Comput. Sci.* 2015, *52*, 780–787. [CrossRef]
- 11. Palli, G.; Melchiorri, C. Robust control of robots with variable joint stiffness. In Proceedings of the 2009 International Conference on Advanced Robotics, Munich, Germany, 22–26 June 2009; pp. 1–6.
- Zheng, C.; Bricogne, M.; Le Duigou, J.; Eynard, B. Survey on mechatronic engineering: A focus on design methods and product models. *Adv. Eng. Informa.* 2014, 28, 241–257. [CrossRef]
- Portilla-Flores, E.A.; Mezura-Montes, E.; Álvarez-Gallegos, J.; Coello-Coello, C.A.; Cruz-Villar, C.A. Integration of structure and control using an evolutionary approach: an application to the optimal concurrent design of a CVT. *Int. J. Numer. Methods Eng.* 2007, 71, 883–901. [CrossRef]
- Behbahani, S.; de Silva, C.W. System-Based and Concurrent Design of a Smart Mechatronic System Using the Concept of Mechatronic Design Quotient (MDQ). *IEEE/ASME Trans. Mechatron.* 2008, 13, 14–21. [CrossRef]
- 15. Padilla-Garcia, E.A.; Rodriguez-Angeles, A.; Reséndiz, J.R.; Cruz-Villar, C.A. Concurrent Optimization for Selection and Control of AC Servomotors on the Powertrain of Industrial Robots. *IEEE Access* **2018**, *6*, 27923–27938. [CrossRef]
- 16. Russo, M.; Herrero, S.; Altuzarra, O.; Ceccarelli, M. Kinematic analysis and multi-objective optimization of a 3-UPR parallel mechanism for a robotic leg. *Mech. Mach. Theory* **2018**, *120*, 192–202. [CrossRef]
- 17. Trevelyan, J. The Making of an Expert Engineer; CRC Press: London, UK, 2014. [CrossRef]
- Yang, T.L.; Liu, A.; Shen, H.; Hang, L.; Luo, Y.; Jin, Q. Topology Design of Robot Mechanisms; Springer: Cham, Switzerland, 2018. [CrossRef]
- 19. Li, T.; Ceccarelli, M. A design procedure for conceptual design of mechanisms. *Int. J. Mech. Robot. Syst.* 2013, *1*, 136–151. [CrossRef]
- Santiago-Valentín, E.; Portilla-Flores, E.A.; Mezura-Montes, E.; Vega-Alvarado, E.; Calva-Yáñez, M.B.; Pedroza-Villalba, M. A Graph-Theory-Based Method for Topological and Dimensional Representation of Planar Mechanisms as a Computational Tool for Engineering Design. *IEEE Access* 2019, 7, 587–596. [CrossRef]
- Yu, J.; Han, S.M.; Kim, Y.Y. Simultaneous Shape and Topology Optimization of Planar Linkage Mechanisms Based on the Spring-Connected Rigid Block Model. J. Mech. Des. 2020, 142, 011401. [CrossRef]
- 22. Szuster, M.; Hendzel, Z., Intelligent Control of Mechatronic Systems. In *Intelligent Optimal Adaptive Control for Mechatronic Systems*; Springer: Cham, Switzerland, 2018; pp. 51–59. [CrossRef]
- Ortega, R.; Donaire, A.; Romero, J.G., Passivity-Based Control of Mechanical Systems. In *Feedback Stabilization of Controlled Dynamical Systems: In Honor of Laurent Praly*; Petit, N., Ed.; Springer: Cham, Switzerland, 2017; pp. 167–199. [CrossRef]
- 24. Iplikci, S. A comparative study on a novel model-based PID tuning and control mechanism for nonlinear systems. *Int. J. Robust Nonlinear Control* 2010, 20, 1483–1501. [CrossRef]
- 25. Deb, K.; Deb, K., Multi-objective Optimization. In *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*; Burke, E.K.; Kendall, G., Eds.; Springer: Cham, Switzerland, 2014; pp. 403–449. [CrossRef]
- Emmerich, M.; Deutz, A. A tutorial on multiobjective optimization: Fundamentals and evolutionary methods. *Nat. Comput.* 2018, 17, 585–609. [CrossRef]
- 27. Yang, X.S. Introduction to Algorithms. In *Nature-Inspired Optimization Algorithms*; Yang, X.S., Ed.; Elsevier: Oxford, UK, 2014; pp. 1–21. [CrossRef]
- 28. Yang, X.S.; Karamanoglu, M. Nature-inspired computation and swarm intelligence: A state-of-the-art overview. In *Nature-Inspired Computation and Swarm Intelligence*; Yang, X.S., Ed.; Academic Press: Cambridge, MA, USA, 2020; pp. 3–18. [CrossRef]
- Storn, R.; Price, K. Differential Evolution—A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. J. Glob. Optim. 1997, 11, 341–359.:1008202821328. [CrossRef]
- Deb, K. An efficient constraint handling method for genetic algorithms. *Comput. Methods Appl. Mech. Eng.* 2000, 186, 311–338.
 [CrossRef]
- 31. Cabrera, J.; Simon, A.; Prado, M. Optimal synthesis of mechanisms with genetic algorithms. *Mech. Mach. Theory* 2002, 37, 1165–1177. [CrossRef]
- 32. Arakelian, V.H.; Smith, M.R. Shaking Force and Shaking Moment Balancing of Mechanisms: A Historical Review With New Examples. *J. Mech. Des.* **2005**, *127*, 334–339. [CrossRef]
- 33. Izmestiev, I. Four-bar linkages, elliptic functions, and flexible polyhedra. Comput. Aided Geom. Des. 2020, 79, 101870. [CrossRef]