Appropriate Synthesis of the Four-Bar Linkage

Joshua K. Pickard[a,*], Juan A. Carretero[b], Jean-Pierre Merlet[c]

[a] *Auctus project, INRIA Bordeaux Sud-Ouest, France*
[b] *Department of Mechanical Engineering, University of New Brunswick, Fredericton, New Brunswick, Canada*
[c] *Hephaistos project, Université Côte d'Azur, INRIA, France*

Abstract

The uncertainties arising from the fabrication and operation of mechanisms, specifically the four-bar linkage, significantly affect the expected performance of the mechanism. To accommodate uncertainties during the dimensional synthesis of the four-bar linkage, the desired coupler curve response is described by any number of precision point elements and trajectory elements, where each element is specified with an allowable error. A design description which accounts for bounded uncertainties is termed an appropriate design. An appropriate synthesis method is developed to synthesize the entire set of appropriate design solutions corresponding to the desired response. This method is able to completely explore the continuous design space and each synthesized appropriate design guarantees that the resulting four-bar linkage has a corresponding coupler point which lies inside each precision point response element, and a corresponding set of continuous coupler points which remain inside the trajectory response elements from start to finish.

Keywords: Synthesis, Uncertainties, Interval Analysis, Design, Linkage

1. Introduction to Linkage Synthesis

Exact synthesis in four-bar linkages involves obtaining geometric parameters which exactly produce a coupler curve. Generally a curve is approximated by a set of *precision points*. Analytical approaches are able to solve the problem and return exact solutions for a four-bar linkage which can guide the coupler point through no more than five crank-coordinated precision points, and no more than nine precision points [1]. Wampler *et al.* [2] obtain for the first time the complete solution for the nine precision point problem. Bai and Angeles [3] develop a new formulation of the synthesis problem to exactly determine the

*Corresponding author
Email addresses: Joshua.Pickard@inria.fr (Joshua K. Pickard), Juan.Carretero@unb.ca (Juan A. Carretero), Jean-Pierre.Merlet@inria.fr (Jean-Pierre Merlet)

geometric parameters corresponding to a given algebraic coupler curve equation. In addition, it is well known due to Roberts [4] that every four-bar linkage has two four-bar cognates, so three solutions are always expected.

In approximate synthesis, numerical methods are used to synthesize a linkage to perform within an acceptable deviation from a desired curve. The curve is generally approximated by a set of precision points. Global optimization methods such as genetic algorithms [5] and other stochastic or heuristic methods [6] have been used to find solutions to the synthesis problem by minimizing a cost function. These methods are unable to guarantee global optimality and often suffer from premature convergence. Recently, Goulet *et al.* [7] presented a synthesis method using a non-convex optimization routine which relies on a solver which applies constraint propagation and interval arithmetic. A more thorough exploration of the design parameter space may be achieved with this method. The solver requires the use of precision points approximating the desired curve, while a cost function minimizes the error between the actual and desired coupler curves. Other techniques such as machine learning and neural network approaches [8, 9] rely on a database of linkage solutions. Such techniques may suffer from the coverage of mechanisms in the database and sampling techniques.

The performance of a linkage cannot be guaranteed using the synthesis techniques currently available in the literature, as they are unable to account for the uncertainties inherent in mechanisms. The issue of uncertainties plays a significant role in the actual performance of a mechanism. In mechanism literature, their effects are generally studied using statistical and probabilistic approaches [10–13] or interval analysis [14–17] approaches. In the four-bar linkage, uncertainties make it no longer possible to obtain an exact solution for the coupler point and thus exact synthesis is not applicable. Instead, the coupler point can only be determined to lie within a region which is a function of the uncertainties. The border of this region cannot usually be obtained in closed-form or in parametric form so it has to be calculated numerically. It would be beneficial to account for the uncertainties during synthesis, such that the desired performance of a synthesized mechanism is guaranteed. Moreover, it would be useful to provide a set of allowable designs which satisfy the desired criteria. The end-user would then be free to select any of the designs which best suit their needs.

In the literature, the idea of synthesizing the optimal designs of mechanisms while accounting for uncertainties has been explored in several publications. Kalnas and Kota [18] consider the five-precision-point problem for synthesizing four-bar linkage motion generating mechanisms and present a method in which intermediate points are described as distributions. Monte Carlo simulations are utilized to identify valid design solutions which satisfy the considered distributions. In [19], mechanism synthesis with random and interval variables is considered in order to model the aleatory (due to inherent variation) and epistemic uncertainties (due to limited information) present in the physical system. A double loop Monte Carlo simulation is proposed to handle the interval and random variables and allows to identify more robust design solutions than a standard deterministic mechanism synthesis. Luo *et al.* [20] extend the previ-

ous work to use truncated distributions, as opposed to normal distributions, for the random variables. While these previously mentioned works provide more robust design solutions, a major drawback of these statistical and probabilistic approaches is the assumed nature of the uncertainty distribution and their reliance on sampling. There is no guarantee that the worst-case performance of the mechanism can be adequately described with such approaches. Furthermore, such approaches can only sample the design space and are therefore unable to provide the desired set of allowable designs.

Alternatively, interval analysis has become an attractive approach for managing uncertainties. Hao and Merlet [21] synthesized a set of optimal designs for the INRIA active wrist based on compulsory requirements for the workspace and accuracy of the mechanism. Routines based on interval analysis were developed to ensure that a set of design parameters satisfy the workspace and accuracy requirements. A feasible solution is found when either requirement is satisfied, while a valid solution is found when both requirements are satisfied. In [15], Merlet and Daney consider the *error synthesis* problem applied to a Gough platform. The authors give a prescribed workspace and use interval analysis techniques to synthesize the set of design parameters such that any pose in the workspace has a positioning error below some desired limit. The term *appropriate design* was introduced by Merlet and Daney [22] for the synthesis of parallel manipulators with uncertainties. The appropriate design methodology involves computing the *allowable regions* (the set of valid solutions) which take into account any uncertainties of the nominal values for the design parameters present in the mechanism (*e.g.*, manufacturing, assembly, sensors) and satisfy some desired characteristics of the mechanism. This guarantees that any physical instance of the mechanism built on these nominal values will satisfy the desired characteristics of the mechanism. The end-user is then presented with the allowable regions for the design parameters and is free to select any valid design to achieve the desired performance. This allows to choose the best compromise regarding any performance criteria that may have not be previously prescribed.

In this work, an appropriate design describes a mechanism whose description contains bounded uncertainties. This description allows any uncertainties due to the manufacturing, assembly, sensors, control, and also numerical round-off errors to be accounted for. The analysis or synthesis of an appropriate design are then fittingly termed *appropriate analysis* and *appropriate synthesis*, respectively. Appropriate analysis has been explored in the previous work [23]. As well, earlier works on appropriate synthesis have been explored in [24, 25]. The goal of this work is to provide a comprehensive overview of the appropriate synthesis method for obtaining the complete set of appropriate designs of a four-bar linkage which correspond to some desired coupler response (namely coupler positions given as a set of points or as whole continuous trajectories or segments of a continuous trajectory). This work builds on the existing literature by thoroughly exploring the formulation of the appropriate synthesis routine, the associated coupler curve response element verification tests, and linkage assembly and classification simplification routines. As well, the use of

conventional optimization techniques are combined with appropriate synthesis in order to identify regions of interest to quickly find appropriate design solutions for high-dimensional problems.

The contributions of this work are as follows. A description for precision points and trajectories, suitable for the appropriate synthesis problem, are introduced, and these are used to build desired coupler responses. Verification tests for the precision point and trajectory response elements are proposed. The coupler point of an appropriate design must be able to pass though each desired precision point, and must move from the start to finish of each desired trajectory. Linkage classification and linkage assembly simplification routines are proposed which aid the appropriate synthesis in ensuring that a synthesized design maintains a unique assembly throughout a desired coupler response. As well, only appropriate designs with a single classification (*e.g.*, crank-rocker) are considered as solutions. Finally, an appropriate synthesis routine is proposed which is able to obtain the complete set of appropriate designs (allowable regions) of a four-bar linkage which correspond to some desired coupler response. Several examples are presented to demonstrate the proposed appropriate synthesis method.

The outline of the paper is as follows. First, in Section 2, a brief overview of the relevant interval analysis concepts are presented. The description of the four-bar linkage and accompanying kinematic equations are provided in Section 3, and linkage classification and linkage assembly simplification routines are presented. The precision point and trajectory elements of a desired response are introduced in Section 4. The associated verification tests are also presented. In Section 5, an appropriate synthesis routine is presented for synthesizing the appropriate designs of a linkage which satisfy some desired coupler curve response. Several case studies are presented in Section 6 to illustrate the capabilities of the proposed methods. Methods for quickly identifying several appropriate design solutions are presented. Finally, the work is concluded in Section 7.

2. Overview of Interval Analysis

Interval arithmetic and *interval analysis* provide a means of performing reliable computations on computers. Allow an interval to be described by

$$[x] = [\underline{x}, \overline{x}] \tag{1}$$

where $\underline{x}$ and $\overline{x}$ are the lower and upper bounds of the interval. Thanks to the work on the formalization of the mathematical foundations of finite interval analysis, many programming languages now support interval data types. Outwardly rounded interval arithmetic provides rigorous enclosures for the ranges of operations and functions. For a desired value, $\underline{x}$ and $\overline{x}$ are chosen such that outward rounding ensures that the value is contained inside the interval and $\underline{x}$ and $\overline{x}$ are the nearest representable floating-point values. Outward rounding is available on all computers supporting IEEE 754 floating-point standard [26].

The width of an interval is given by

$$width([x]) = \overline{x} - \underline{x} \tag{2}$$

The midpoint of an interval is given by

$$\text{mid}([x]) = \underline{x} + width([x])/2 \tag{3}$$

Let $[\mathbf{x}]$ denote an interval vector. The interval evaluation of a function $f([\mathbf{x}])$ yields the inclusion function $[f]$, such that $f([\mathbf{x}])$ is contained inside of $[f]$. That is,

$$f([\mathbf{x}]) = \{f(\mathbf{x}) \mid \mathbf{x} \in [\mathbf{x}]\} \subseteq [f]. \tag{4}$$

The bounds of $[f]$ are inflated due to two well known properties of interval analysis: the wrapping effect, and the dependency problem (see [27]).

A general solving routine may be designed using interval analysis. Typically, solutions to problems may be found by incorporating three phases evaluated in a loop. These phases include: *simplification* (various simplification methods include 2B and 3B filtering [28], HC4 [29], ACID [30], and Newton [31]), *existence* (various existence methods include Interval Newton [32], Krawcyzk [31], and Newton-Kantorovitch [33]), and *bisection* (various bisection methods include Largest-first and Smear function [34]). For these phases, $[\mathbf{k}]$ denotes the known variables and $[\mathbf{u}]$ denotes the unknown variables. The system being solved is described by the set of equations $\mathbf{f}([\mathbf{u}], [\mathbf{k}]) = \mathbf{0}$.

Existence methods are guaranteed to converge to the unique solution $[\mathbf{u}^*]$. For the bisection phase, an arbitrary tolerance of β is used to limit the bisected width of the unknowns, such that β serves as the stopping criteria for the bisection. The bisection phase splits the unknowns $[\mathbf{u}]$ along a selected dimension into two sub-intervals $[\mathbf{u}_1]$ and $[\mathbf{u}_2]$. Since the union $[\mathbf{u}_1] \cup [\mathbf{u}_2] = [\mathbf{u}]$, bisection allows the entire unknown search space to be reliably explored and therefore a general solving routine is guaranteed to be able to find all solutions to the considered problem. A detailed overview of the three phases is provided in [23].

3. Four-bar Linkage Description

The four-bar linkage and its associated design parameters are provided in Figure 1. Link O_AA will always be assumed to be the input link with an input angle θ and length r. Link O_BB is the output link, which has an output angle ψ and length s. The coupler point is denoted $C = (C_x, C_y)$, where the coupler link is triangle ABC with edge lengths a, b, and c. Let g be the distance between points O_A and O_B. The parameters e and h are used to describe the location of C relative to segment AB. The signs of e and h are important to describe a unique configuration of the linkage. The equations describing the kinematics of the four-bar linkage, formulated using a coordinate representation, are given in Eq. (5) (see [23] for the formulation). This formulation is preferable for use with interval analysis methods due to the minimal occurrence of variables in the equations, which reduces the effects of the interval dependency problem.

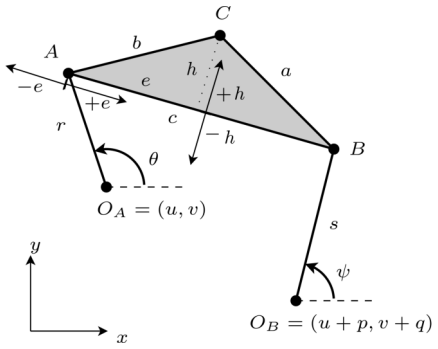


Figure 1: Linkage description

$$
\begin{aligned}
f_1 &:= ||O_A A||^2 = (u - A_x)^2 + (v - A_y)^2 = r^2 \\
f_2 &:= ||O_B B||^2 = ((p + u) - B_x)^2 + ((q + v) - B_y)^2 = s^2 \\
f_3 &:= ||AB||^2 = (A_x - B_x)^2 + (A_y - B_y)^2 = c^2 \\
f_4 &:= ||AC||^2 = (A_x - C_x)^2 + (A_y - C_y)^2 = b^2 \\
f_5 &:= ||BC||^2 = (B_x - C_x)^2 + (B_y - C_y)^2 = a^2 \\
f_6 &:= A_x = u + r\cos(\theta) \\
f_7 &:= A_y = v + r\sin(\theta) \\
f_8 &:= B_x = (p + u) + s\cos(\psi) \\
f_9 &:= B_y = (q + v) + s\sin(\psi) \\
f_{10} &:= b = \sqrt{e^2 + h^2} \\
f_{11} &:= a = \sqrt{(c - e)^2 + h^2} \\
f_{12} &:= g = \sqrt{p^2 + q^2} \\
f_{13} &:= C_x = A_x + 1/c\,((B_x - A_x)e + (A_y - B_y)h) \\
f_{14} &:= C_y = A_y + 1/c\,((B_y - A_y)e + (A_x - B_x)h)
\end{aligned}
\tag{5}
$$

3.1. The Design Parameters

The design parameters of the four-bar linkage given by system (5) are described by the interval vector $[\mathbf{d}]$. The interval domains of the parameters account for uncertainties and therefore can be represented as an interval vector:

$$[\mathbf{d}] = ([u], [v], [p], [q], [r], [s], [c], [e], [h])^T \tag{6}$$

The manufacturing uncertainties for the design parameters are described by the vector $\Delta\mathbf{d}$ as

$$\Delta\mathbf{d} = (\Delta u, \Delta v, \Delta p, \Delta q, \Delta r, \Delta s, \Delta c, \Delta e, \Delta h)^T \tag{7}$$

If an exact design $\mathbf{d}$ is considered, the appropriate design which accounts for manufacturing uncertainties is given as

$$[\mathbf{d}] = \mathbf{d} \pm \Delta\mathbf{d} \tag{8}$$

3.2. Classifications and Assemblies

Four-bar linkages can be sorted into two categories, *Grashof-type* and *non-Grashof-type* linkages, where each category of linkage has four associated classifications. Applying the description for circuits and branches adopted from Chase and Mirth [35], each linkage classification has a specific number of circuits and branches associated with the coupler curve of the linkage. The eight linkage classifications and associated circuits and branches (detailed in [23]) are described as follows:

1. *crank-rocker* – 2 circuits, each with 1 branch;
2. *rocker-crank* – 2 circuits, each with 2 branches;
3. *double-crank* – 2 circuits, each with 1 branch;
4. *double-rocker* – 2 circuits, each with 2 branches;
5. 00-*double-rocker* – 1 circuit with 2 branches;
6. 0π-*double-rocker* – 1 circuit with 2 branches;
7. $\pi 0$-*double-rocker* – 1 circuit with 2 branches;
8. $\pi\pi$-*double-rocker* – 1 circuit with 2 branches.

For appropriate synthesis, a four-bar linkage requires that an appropriate design solution has a unique linkage classification and that the desired response is accomplished by a single circuit and branch, although multiple branches may be desirable in certain cases. It is also possible to restrict the appropriate synthesis routine to find solutions corresponding to one or more specified linkage classifications. In order to assist the appropriate synthesis routine in finding valid solutions, filtering routines are proposed for linkage classifications and assemblies. This allows appropriate designs which do not satisfy the classification and assembly conditions to be quickly removed from the design search space.

3.2.1. Classification Filtering

A given appropriate design $[\mathbf{d}]$ may have multiple classifications. Let the list $\mathcal{L}_{class}$ contain the possible classifications associated with the design $[\mathbf{d}]$. In certain situations, it may be desirable to consider a particular classification of a linkage during synthesis. The list $\mathcal{L}_{allow\_class}$ contains all allowable classifications. A routine which considers the allowed and possible classifications is used as an additional simplification routine to remove a design $[\mathbf{d}]$ which results in only non-allowable classifications. It is also used to identify if a design $[\mathbf{d}]$ results in a single allowable classification. The function CLASSIFICATION$([\mathbf{d}], \mathcal{L}_{class}, \mathcal{L}_{allow_class})$ updates the possible classifications and returns:

-1 : the allowable classifications are not contained in $\mathcal{L}_{class}$;
1 : there is a unique allowable classification in $\mathcal{L}_{class}$;
0 : there is more than one classification in $\mathcal{L}_{class}$.

3.2.2. Assembly Filtering

A given appropriate design $[\mathbf{d}]$ and a set of associated coupler points may yield multiple assemblies (circuits and branches). It is necessary that certain restrictions be applied to the assemblies. A desired coupler curve response can only be achieved with a single circuit. In some cases, a change of branch may or may not be allowed. An assembly filtering routine is proposed and is used as an additional simplification method in order to remove a design $[\mathbf{d}]$ which cannot satisfy the assembly conditions. Let the list $\mathcal{L}_{branch}$ denote the list of associated branches and the list $\mathcal{L}_{circuit}$ denote the list of associated circuits for a given design $[\mathbf{d}]$ and coupler point. A branch flag identifies whether a desired coupler curve may be generated by one or two branches, whereas a single circuit is always enforced.

The function ASSEMBLY($\{[\mathbf{u}]\}, [\mathbf{k}], \{\mathcal{L}_{\mathbf{class}}\}, \{\mathcal{L}_{\mathbf{branch}}\}, \{\mathcal{L}_{\mathbf{circuit}}\}$) considers two or more coupler points and the assemblies associated with these points[1]. It utilizes CLASSIFICATION and finds the intersection of the classifications, branches, and circuits, and updates each list by the intersection. The following is returned:

-1 : the intersection of all assemblies is empty and the set of coupler points cannot be satisfied with the design $[\mathbf{d}]$;
1 : there is a unique assembly which satisfies every coupler point for the design $[\mathbf{d}]$;
0 : there is more than one possible assembly for the design $[\mathbf{d}]$.

3.3. Solving the Kinematics

To solve the kinematics of the four-bar linkage for the appropriate synthesis problem, a problem termed the *interior problem* is considered.

- **Interior problem**: *Determine a solution for angles* $[\theta]$ *and* $[\psi]$ *and coupler point* $([C_x], [C_y])$ *from the interior of the box* $([x], [y])$. For this problem, a box $([x], [y])$ is given and $[\theta]$ and $[\psi]$ are given as allowable ranges. The goal is to compute a solution for $[\theta]$, $[\psi]$ and $([C_x], [C_y])$ such that $([C_x], [C_y]) \subset ([x], [y])$.

The interior problem is useful for determining if an appropriate design satisfies certain coupler curve characteristics, as is required for synthesis. To solve the interior problem, an existence test can be formulated from Equations f_2–f_5,

[1]The notation $\{*\}$ denotes a set, such that each $[\mathbf{u}]$ has corresponding $\mathcal{L}_{class}$, $\mathcal{L}_{branch}$, and $\mathcal{L}_{circuit}$ lists.

which gives a square system in unknowns B_x, B_y, C_x, and C_y. The existence test for the interior problem is the same as the existence test for the forward problem (see [23]). To create various solving routines, a bisection phase may safely bisect the input and/or output angles $[\theta]$ and $[\psi]$.

4. Desired Coupler Curve Response: Description and Verification

The goal is to be able to synthesize the design parameters of four-bar linkages given the description of a desired coupler curve. The desired coupler curve will be described by a set of precision points and/or a set of trajectories. Each element, whether a precision point or a trajectory, is described with an allowable error. The descriptions of precision points and trajectories and the routines for verifying the satisfaction of the precision point and trajectory elements for a given appropriate design are presented in this section.

4.1. Precision Points

In classical linkage synthesis, precision points are described exactly and the synthesized linkage must pass through/near each point. In this work, due to the uncertainties in the design parameters, the coupler point will belong to some bounded region in space. To accommodate this, a precision point is described here as a set of intervals, such that each element may have an allowable error. A precision point will be denoted by $\mathcal{P}$ as

$$\mathcal{P} = ([C_x], [C_y], [\theta], [\psi]) \tag{9}$$

In order for a linkage to satisfy a precision point, there must exist a solution $\mathcal{P}^*$ for the linkage, such that the inclusion constraints $[C_x^*] \subset [C_x]$, $[C_y^*] \subset [C_y]$, $[\theta^*] \subset [\theta]$ and $[\psi^*] \subset [\psi]$ are guaranteed to be satisfied. That is, $\mathcal{P}^* \subset \mathcal{P}$. Multiple precision points may be considered.

4.2. Trajectories

A desired trajectory, having an allowable error, may also be considered. The desired trajectory, denoted $f(C_x([t]), C_y([t]))$, can be described by parametric equations as

$$f(C_x([t]), C_y([t])) = \{(C_x, C_y) \mid C_x = f_x(t),\ C_y = f_y(t),\ t \in [\underline{t}, \overline{t}]\} \tag{10}$$

An allowable error on the trajectory can be described by $\alpha\hat{\mathbf{n}}$ where $\hat{\mathbf{n}} = (\hat{n}_x, \hat{n}_y)$ is the unit normal along $f(C_x([t]), C_y([t]))$ and $\alpha \in [\underline{\alpha}, \overline{\alpha}]$ is the allowable error. The allowable trajectory can then be defined as

$$\begin{aligned} f(C_x([t]), C_y([t])) = \{(C_x, C_y) \mid C_x = f_x(t) + \alpha\hat{n}_x,\ C_y = f_y(t) + \alpha\hat{n}_y, \\ t \in [\underline{t}, \overline{t}],\ \alpha \in [\underline{\alpha}, \overline{\alpha}]\} \end{aligned} \tag{11}$$

Since it is difficult to account for the limits of the trajectory, the domain of $[t]$ can be inflated by some small amount δ to create start and finish end-points

for the trajectory. These end-points will have a width δ and are defined as $[t_{start}] = [\underline{t} - \delta, \underline{t}]$ and $[t_{finish}] = [\overline{t}, \overline{t} + \delta]$. The task element $\mathcal{T}$ corresponding to a trajectory is represented by the desired trajectory $f(C_x([t]), C_y([t]))$, input and output angles $[\theta]$ and $[\psi]$, allowable error $[\alpha]$, the parametric equation parameter $[t]$, and end-point width δ as

$$\mathcal{T} = \{f(C_x([t]), C_y([t])), [\theta], [\psi], [\alpha], [t], \delta\} \tag{12}$$

In order for a linkage to satisfy a trajectory, a solution must exist for each end-point, and the coupler curve must be continuous between the end-points and must remain within the boundaries of the trajectory. The steps to satisfy a trajectory are summarized as:

1. Ensure the start end-point is satisfied: There must exist a solution $\mathcal{T}^*$, such that
$$[C^*_{start}] \subset f(C_x([t]), C_y([t])),\ [\theta^*_{start}] \subset [\theta],\ [\psi^*_{start}] \subset [\psi], \\ [\alpha^*_{start}] \subset [\alpha], [t^*_{start}] \subset [t_{start}] \tag{13}$$
2. Ensure the finish end-point is satisfied: There must exist a solution $\mathcal{T}^*$, such that
$$[C^*_{finish}] \subset f(C_x([t]), C_y([t])),\ [\theta^*_{finish}] \subset [\theta],\ [\psi^*_{finish}] \subset [\psi], \\ [\alpha^*_{finish}] \subset [\alpha],\ [t^*_{finish}] \subset [t_{finish}] \tag{14}$$
3. Ensure the coupler curve is continuous between the start and finish end-points and remains within the boundaries of the trajectory: There must exist a solution $\mathcal{T}^*$ for each $[\theta] \in ([\theta^*_{start}] \square [\theta^*_{finish}])$, where $\square$ denotes the interval hull, such that
$$[C^*] \subset f(C_x([t]), C_y([t])),\ [\theta^*] = [\theta],\ [\psi *] \subset [\psi],\ [\alpha^*] \subset [\alpha], \\ [t^*] \subset ([t_{start}] \square [t_{finish}]) \tag{15}$$
Due to the periodicity of the angle θ, the interval hull $([\theta^*_{start}] \square [\theta^*_{finish}])$ has two associated domains. These domains correspond to clock-wise and counter-clock-wise rotations from $[\theta^*_{start}]$ to $[\theta^*_{finish}]$.

Multiple trajectories may be considered.

4.2.1. Example Trajectory

Consider the parametric curve for $C_y = C_x^2,\ C_x \in [-1, 1]$.

$$\begin{aligned} C_x &= t \\ C_y &= t^2 \end{aligned} \tag{16}$$

The normal $\mathbf{n}$ is $\mathbf{n} = [2C_x, -1]^T = [2t, -1]^T$ and the unit normal $\hat{\mathbf{n}}$ is $\mathbf{n}/||\mathbf{n}||$ with $||\mathbf{n}|| = \sqrt{4t^2 + 1}$. Thus the allowable parametric curve is

$$\begin{aligned} C_x &= t + \alpha \hat{n}_x = t + \alpha(2t/\sqrt{4t^2 + 1}) \\ C_y &= t^2 + \alpha \hat{n}_y = t^2 - \alpha(1/\sqrt{4t^2 + 1}) \\ t &\in [-1, 1] \end{aligned} \tag{17}$$

The allowable coupler curve is plotted in Figure 2 for $t \in [-0.9900, 0.9900]$ and $\alpha \in [-0.1000, 0.1000]$. The allowable end-points of the curve are given as $t_{start} \in [-1.0100, -0.9900]$ and $t_{finish} \in [0.9900, 1.0100]$, respectively. The trajectory must start and finish by satisfying the end-points. These end-points differ from precision points because they are not axis-aligned. Thus, the end-points better approximate the desired curve.

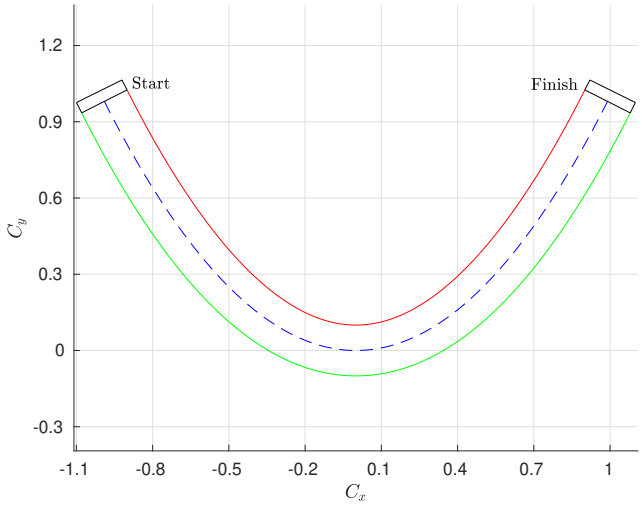


Figure 2: The desired trajectory corresponding to $C_y = C_x^2$.

4.3. Verification Test for Precision Points

Let a desired response $\mathcal{R}$ contain m precision points $\mathcal{P}_i$.

$$\mathcal{R} = (\mathcal{P}_1, \ldots, \mathcal{P}_m) \tag{18}$$

The goal is to determine if a linkage appropriate design $[\mathbf{d}]$ is able to achieve each response element in $\mathcal{R}$. A routine is proposed in order to verify the existence of a coupler point solution inside each of the precision points.

The known and unknown parameters will vary for the different phases: simplification, existence, and bisection. This ensures that the equations in the existence phase are as simple as possible, and that the coordinates are not bisected. The design parameters $[\mathbf{d}]$ will always be considered as knowns. In the simplification phase, the unknowns are selected as $[\mathbf{u}] = ([A_x], [A_y], [B_x], [B_y], [C_x], [C_y], [\theta], [\psi])$. In the existence phase, the unknowns are selected as $[\mathbf{u}] = ([B_x], [B_y], [C_x], [C_y])$, such that the interior problem may be used. In the bisection phase, the unknown parameters are selected as $[\mathbf{u}] = ([\theta], [\psi])$. Bisecting the angles, rather than the coordinates, ensures that the coordinates always contain the true joint positions. All remaining parameters may be considered as knowns.

A routine for verifying m precision points is presented in Algorithm 1. Since there are m precision points to be verified, all m associated unknown vectors will be considered simultaneously. If any precision point is able to quickly return no solution, then the routine exits and resources are not wasted on the other precision points. To accomplish this, the set of m unknown vectors, $\{[\mathbf{u}_i], \ldots, [\mathbf{u}_m]\}$, are added to the list $\mathcal{L}_u$. Each phase then accepts as input the set $\{[\mathbf{u}]\}$ and applies the corresponding operation to each element $[\mathbf{u}_i]$ from the set. The simplification phase simplifies all m unknowns and returns: 1 if one or more unknowns is simplified, -1 if any unknown becomes empty, 0 if the m unknowns cannot be simplified. The assembly routine verifies the assembly conditions previously described. The existence phase checks for the existence of a solution for all m unknowns and returns: 1 if all m precision points are verified, -1 if any unknown has no solution, 0 if a solution cannot be found for all m unknowns. The bisection phase bisects the unknowns which return 0 from the existence phase.

Algorithm 1: Verify m precision/trajectory points

```
function VERIFY_POINTS (R, [d], β);
Input  : desired response R, design space [d], and desired bisection resolution β
Output: verification - the desired response is: satisfied(1), unsatisfied(-1), partially satisfied(0)
L_u ← {[u_i], ..., [u_m]} ;                          /* Add set of unknowns to list */
L_class ← L_allow_class ;                  /* Add allowed classifications to list */
L_branch ← {1, 2} ;                          /* Add possible branches to list */
L_circuit ← {1, 2} ;                         /* Add possible circuits to list */
while L_u is not empty do
    Pop set {[u]} from L_u;
    simp = SIMPLIFICATION({[u]}, [k]) ;      /* Apply simplification techniques */
    if simp! = −1 then
        assembly = ASSEMBLY({[u]}, [k], {L_class}, {L_branch}, {L_circuit}) ;   /* Verify
          assembly conditions */
        if assembly == 1 then
            exist = EXISTENCE({[u]}, [k]) ;          /* Apply existence methods */
            if exist == 1 then
                return 1
            else if exist == 0 then
                L_u ← BISECTION({[u]}, β) ;          /* Apply bisection method */
            else
                return 0;
return −1;
```

4.3.1. Verification Test for Trajectories

Let a set of responses $\mathcal{R}$ contain n trajectories $\mathcal{T}_i$.

$$\mathcal{R} = (\mathcal{T}_1, \ldots, \mathcal{T}_n) \tag{19}$$

The goal is to determine if a linkage appropriate design $[\mathbf{d}]$ is able to achieve each response element in $\mathcal{R}$. A routine is proposed in order to verify the existence of a coupler point solution along the entire trajectory.

Like the precision point problem, the known and unknown parameters will vary for the different phases. The design parameters $[\mathbf{d}]$ will always be considered as knowns. In the simplification phase, the unknowns are selected as

$[\mathbf{u}] = ([A_x], [A_y], [B_x], [B_y], [C_x], [C_y], [\theta], [\psi], [t], [\alpha])$. In the existence phase, the unknowns are selected as $[\mathbf{u}] = ([B_x], [B_y], [C_x], [C_y], [t], [\alpha])$. Similar to the interior problem, a square system is created using equations f_2 through f_5 and the parametric equations of the trajectory curve. In the bisection phase, the unknown parameters are selected as $[\mathbf{u}] = ([\theta], [\psi])$. All remaining parameters may be considered as knowns.

In order to verify a trajectory, each $t \in [t]$ must have a solution which stays within the allowable error of the trajectory. Each solution must also satisfy the assembly conditions and the coupler curve must be continuous along $[t]$. The first consideration is the verification of the end-points of a trajectory. This is done in the same manner as the precision points verification, albeit with the addition of the parameters $[t]$ and $[\alpha]$ in the simplification and existence phases. If the end-points of each trajectory are verified, then the remainder of the trajectory can be considered. Conveniently, the verification test for an end-point is identical to the verification test for the remainder of the trajectory (internal-points). Therefore, a single routine can be used for the verification of trajectory end-points and internal-points. In addition, the structure of the trajectory point verification routine is identical to the precision point verification routine but with different simplification and existence tests, thus Algorithm 1 also describes the trajectory point verification procedure. The end-points or internal-points for all trajectories are considered simultaneously. For the end-point verification of n trajectories, let $m = 2n$. If any point returns no solution, the entire routine exits.

A second routine is required in order to organize the testing of the end-points and internal-points. This routine is presented in Algorithm 2. It makes repeated calls to Algorithm 1 in order to verify the end-points and the internal-points. The end-points are first verified. If the verification succeeds on all end-points then the interior points of each trajectory are considered in turn, starting from the lower bound of the input angle domain associated with the trajectory $(([\theta^*_{start}]\square[\theta^*_{finish}]))$ and incrementing by $\Delta\theta$ until the upper bound of the input angle domain is reached. If any interior point is unable to yield a solution, then the routine is unable to verify all of the trajectories and must exit.

A complication arises in the proposed trajectory verification algorithm resulting from the coupler curve of a linkage passing through the end-points of a trajectory more than once. The end-point verification in Algorithm 1 succeeds when a solution is found for each end-point which satisfies the assembly conditions; however, the input angle domain associated with the trajectory $(([\theta^*_{start}]\square[\theta^*_{finish}]))$ cannot be assumed to have minimal width. As an example, consider applying the trajectory verification to a crank-rocker linkage which yields input angles for the end-points as $[\theta^*_{start}] = [-\pi/2] \pm 0.0100$ and $[\theta^*_{finish}] = [\pi/3] \pm 0.0100$. The input angle domain associated with the trajectory may be either $([\theta^*_{start}]\square[\theta^*_{finish}]) = [-\pi/2 + 0.0100, \pi/3 - 0.0100]$ for minimal width or $([\theta^*_{start}]\square[\theta^*_{finish}]) = [\pi/3 + 0.0100, 3\pi/2 - 0.0100]$. Without sampling interior points, it is unknown which input angle domain actually connects the end-points. An *input angle rejection list* is proposed to ensure that the correct

Algorithm 2: Verify n trajectories

```
function VERIFY_TRAJECTORIES (R, [d], β, Δθ);
Input  : desired response R, design space [d], desired bisection resolution β, and angle resolution
         Δθ
Output: verification - the desired response is: satisfied(1), unsatisfied(-1), partially satisfied(0)
R_E = Generate end-point description from R;
verified = VERIFY_POINTS(R_E, [d], β) ;          /* Verify the trajectory end-points */
if verified == 1 then
    forall T_i ∈ R do
        [θ_i] = ([θ*_start]□[θ*_finish]) + [0, Δθ];
        while [θ_i] < ([θ*_start]□[θ*_finish]) do
            R_I = Generate interior-point description from T_i;
            verified = VERIFY_POINTS(R_I, [d], β) ;          /* Verify the trajectory
              interior-points */
            if verified! = 1 then
                return verified ;          /* An interior-point cannot be verified */
            [θ_i] = [θ_i] + Δθ;
    return 1 ;                             /* All interior-points have been verified */
else
    return verified
```

end-points are selected and the correct input angle domain is then tested. This creates a list $\mathcal{L}_\theta$ which contains domains of the input angle which are known to dissatisfy the trajectory. Each time the interior-point verification test in Algorithm 2 fails, the input angle associated with the interior-point leading to failure is added to the rejection list $\mathcal{L}_\theta$. The trajectory verification routine then returns back to the end-point verification test and now uses the list $\mathcal{L}_\theta$ as an additional filtering method. Only end-point solutions which yield an input angle domain that does not contain the input angles in the rejection list are subjected to the interior-point verification test. The trajectory can only be satisfied when both the end-point and interior-point verification tests are satisfied.

Several other improvements may be made to the proposed trajectory verification algorithm. The velocity $\mathbf{v}$ of the coupler point, with respect to θ, may be used to improve the trajectory verification. If $\mathbf{j}$ is a vector tangent to the trajectory (*i.e.*, $\mathbf{j} = [\hat{n}_y, -\hat{n}_x]$, the dot product of $\mathbf{j}$ and $\mathbf{v}$ should always be positive. This would ensure that the coupler point moves only in the desired direction of the trajectory. It may also be beneficial to sample a point inside the interior region of the trajectory to better predict the input angle domain. As well, the stopping criteria of the interior-point test may be changed so that the test is successful as soon as a solution is found inside the final end-point. It may not be necessary to test the entire input angle domain ($[\theta^*_{start}]\square[\theta^*_{finish}]$).

5. Appropriate Synthesis

A desired response, given by $\mathcal{R}$, contains a set of precision points and/or trajectories. Appropriate synthesis concerns finding all subsets of the initial design parameter space which generate a four-bar linkage that satisfies the desired response. This subset of designs is known as the *allowable region*. The

allowable region ensures that each desired response element is satisfied, via Algorithms 1 and 2, and that the design solutions satisfy classification restrictions and assembly conditions.

In order to compute the allowable region, a list of appropriate design solutions, denoted by $\mathcal{L}_{d_s}$, must be computed.

$$\mathcal{L}_{d_s} = \{[\mathbf{d}_1], \ldots, [\mathbf{d}_k]\} \tag{20}$$

Since $\Delta\mathbf{d}$ is the manufacturing uncertainties on the design parameters, each appropriate design solution $[\mathbf{d}]$ must satisfy

$$width([\mathbf{d}]) \geq 2\Delta\mathbf{d} \tag{21}$$

That is, the width of each design parameter in $[\mathbf{d}]$ must be greater than twice the corresponding manufacturing uncertainty. This is considered as a stopping criteria in the synthesis routine. An allowable design solution $[\mathbf{d}^*]$ is determined by deflating the bounds of $[\mathbf{d}]$ by $\Delta\mathbf{d}$

$$[\mathbf{d}^*] = [\underline{[\mathbf{d}]} + \Delta\mathbf{d}, \overline{[\mathbf{d}]} - \Delta\mathbf{d}] \tag{22}$$

This ensures that if an appropriate design solution is found, then $[\mathbf{d}^*]$ defines a reduced box whose any interior point may be chosen as a nominal design solution, while ensuring that the desired precision points and trajectories can be performed by the real mechanism. To compute the full allowable region, adjacent appropriate design solutions in $\mathcal{L}_{d_s}$ may be combined, where the boundaries of the region are deflated by $\Delta\mathbf{d}$.

In the previous sections, routines for the verification of precision points and trajectories have been presented. These routines may be combined into a single verification routine, described in Algorithm 3, which takes some description of $\mathcal{R}$ and determines if an appropriate design $[\mathbf{d}]$ is verified.

Algorithm 3: Verify an appropriate design for a desired response

```
function VERIFY_APPROPRIATE_DESIGN (R, [d], β, Δθ);
Input  : desired response R, design space [d], desired bisection resolution β, and angle resolution
         Δθ
Output: verification - the desired response is: satisfied(1), unsatisfied(-1), partially satisfied(0)
verified =VERIFY_POINTS(R, [d], β) ;                    /* Verify precision points */
if verified! = 1 then
    return verified;
verified =VERIFY_TRAJECTORIES(R, [d], β, Δθ) ;          /* Verify trajectories */
if verified! = 1 then
    return verified;
return 1;
```

For appropriate synthesis, each design parameter has an initial domain, such that the design parameters are able to be bisected, until a desired bisection resolution of β, in order to search for design solutions. An initial bisection resolution may be safely selected as $\beta = 2\Delta\mathbf{d}$, where each design parameter may have a different bisection resolution. The boundaries of the set of appropriate design solutions $\mathcal{L}_{d_s}$ are unable to be classified as solutions, but can also not be classified

as non-solutions; thus, there will always be a boundary layer between solutions and non-solutions. This provides three possible design parameter classifications:

1. Solution: every $\mathbf{d} \in [\mathbf{d}]$ contains a solution for each element in $\mathcal{R}$;
2. Non-solution: some $\mathbf{d} \in [\mathbf{d}]$ does not contain a solution for one or more elements in $\mathcal{R}$;
3. Unknown (boundary): it is unknown if every $\mathbf{d} \in [\mathbf{d}]$ contains a solution for each element in $\mathcal{R}$, given the current resolution.

It should be noted that unknown classifications may occur if $\Delta\mathbf{d}$ is relatively large, as the uncertainties in the system and the wrapping effect of interval analysis may prevent the detection of a solution or non-solution. The volume of the set of unknown boxes may be reduced with an incremental algorithm. If V_s is the volume of the set of solution boxes and V_u the volume of the set of unknown boxes then the real allowable region volume V is bounded by $V_s \leq V \leq V_s + V_u$, which allows to control the level of error in the solution. To refine the solution, only the unknown boxes from the previous run must be reconsidered. Such a process may be pursued until $(V_s + V_u)/V_s$ is arbitrarily low.

An appropriate synthesis routine is proposed in Algorithm 4, which returns three lists, the solutions $\mathcal{L}_{d_s}$, the boundaries $\mathcal{L}_{db}$, and the non-solutions $\mathcal{L}_{d_n}$ for the appropriate designs. A design list $\mathcal{L}_d$ contains the set of design parameters being considered. The verification routines are applied to each $[\mathbf{d}]$ in the list $\mathcal{L}_d$. If the routine returns 0, then bisection is applied to $[\mathbf{d}]$; bisection is only applied if $width([\mathbf{d}]) \geq 2\Delta\mathbf{d}$; otherwise, the design is saved to the boundary list. If the routine returns 1, then the design is verified and is saved to the solution list. If the routine returns -1, the design is not a solution and is discarded (or saved to the non-solution list). The boundary list may be incrementally refined by decreasing the bisection resolution β and considering only the boundary list ($\mathcal{L}_{db}$) obtained previously.

Algorithm 4: Appropriate synthesis routine

```
function APPROPRIATE_SYNTHESIS (R, [d], Δd, β, Δθ);
Input  : desired response R, design space [d], design uncertainties Δd, desired bisection
         resolution β, and angle resolution Δθ
Output: list of solutions L_ds, non-solutions L_dn, and boundaries L_db
L_d ← [d] ;                                          /* Add initial design domains to list */
while L_d is not empty do
    Pop [d] from L_d;
    appropriate = VERIFY_APPROPRIATE_DESIGN(R, [d], β, Δθ) ;        /* Apply the
      verification routines */
    if appropriate == 1 then
        L_ds ← [d] ;                                  /* Save design to solution list */
    else if appropriate == −1 then
        L_dn ← [d] ;                                  /* Save design to non-solution list */
    else
        (L_d, L_db) ← BISECTION([d], 2Δd) ;                        /* Apply bisection */
```

The appropriate synthesis routine is an ideal candidate for distributed computing since the treatment of a given box is not dependent on the treatment of another box. Hence the calculation may be distributed over several computers.

Furthermore, the communication overhead is low because a only a very limited number of floating points pairs have to be transmitted. Experimentally, if n computers are used with $n > 4$, then the computation time is reduced by more than $n - 1$.

6. Case Studies

In this section, the appropriate synthesis routine is applied to solve for the design solutions (allowable regions) of four-bar linkages for different task requirements. For visualization purposes, mostly 2D solutions are presented.

6.1. Obtaining Allowable Regions

Consider the desired coupler curve described in (23) with three precision points and two trajectories. The response elements are chosen, such that they correspond to the coupler curve obtained from the design parameters given in (24). The desired response elements are plotted in Figure 3.

$$\begin{aligned}
&\mathcal{P}_1 = \{[0.1400, 0.1600], [0.3337, 0.3537], [-\pi, \pi], [-\pi, \pi]\} \\
&\mathcal{P}_2 = \{[0.1900, 0.2100], [0.3737, 0.3937], [-\pi, \pi], [-\pi, \pi]\} \\
&\mathcal{P}_3 = \{[0.2400, 0.2600], [0.3237, 0.3437], [-\pi, \pi], [-\pi, \pi]\} \\
&f(C_x([t]), C_y([t])) = \{(C_x, C_y) \mid C_x = t,\ C_y = \alpha - 0.0650,\ t \in [\underline{t}, \overline{t}]\} \\
&\mathcal{T}_1 = \{f(C_x([t]), C_y([t])), [-\pi, \pi], [-\pi, \pi], [-0.0100, 0.0100], [0.1300, 0.1700], 0.0050\} \\
&\mathcal{T}_2 = \{f(C_x([t]), C_y([t])), [-\pi, \pi], [-\pi, \pi], [-0.0100, 0.0100], [0.1900, 0.2300], 0.0050\}
\end{aligned} \tag{23}$$

To demonstrate appropriate synthesis of the four-bar linkage, the design parameters are initialized as in (24). An uncertainty of $[\rho] = [-0.0001, 0.0001]$ is added to each design parameter. The coupler curve in Figure 3 corresponds to $[p] = [0.4000]$ and $[q] = [0.0000]$. The design parameter uncertainties are selected as $\Delta d_i = 0.0005$ and the bisection resolution is selected as $\beta = 0.0005$.

$$\begin{aligned}
&[\mathbf{d}] = ([u], [v], [p], [q], [r], [s], [c], [e], [h])^T \\
&[\mathbf{d}] = ([0.0000], [0.0000], [-1.0000, 1.0000], [-1.0000, 1.0000], [0.2400], [0.2400], \\
&\quad [0.2517], [0.1258], [0.1553])^T + [\boldsymbol{\rho}]^T
\end{aligned} \tag{24}$$

The allowable region considering the precision point task elements from (23) is shown in Figure 4. The design solutions do not consider restrictions on the branch configuration and any design which achieves the desired precision points, regardless of branch configuration is accepted as a solution (a single branch configuration may optionally be enforced). This generates several disconnected regions. Each allowable region is surrounded by unknown (boundary) design intervals. The design solutions all have a classification of 0π-double-rocker. Figure 5a zooms in on one of the regions. An appropriate design solution may be chosen from inside of the allowable regions while considering the manufacturing uncertainties. Selecting the values of $[p] = [0.5699, 0.5701]$ and $[q] = [0.4299, 0.4301]$

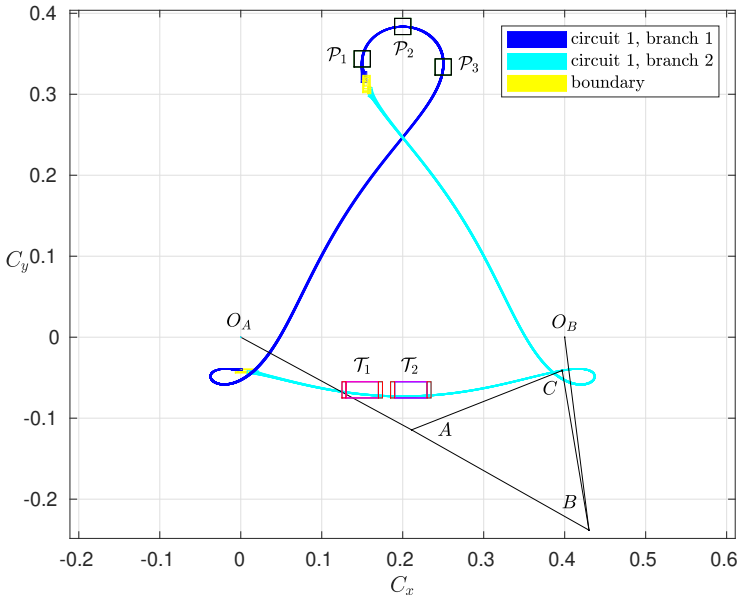


Figure 3: A plot of the desired response elements in (23).

from the allowable region of Figure 5a, the corresponding coupler curve and desired precision points are plotted in Figure 5b. Indeed, the coupler curve passes through each of the precision points.

Considering only the trajectory elements from (23), Figure 6 shows a subset of the allowable region considering the first trajectory. This image shows that there exists a division in the design solutions. Some designs in this region correspond to a folding linkage, and such a linkage, while theoretically possible, is impossible to manufacture as it can only be achieved using flexible links or introducing uncertainties into the joints. For this reason, folding linkages are not considered as design solutions in this work. Next, Figure 7 shows the allowable region considering both trajectories. Several disconnected regions are found. Selecting the values of $[p] = [0.2999, 0.3001]$ and $[q] = [0.0199, 0.0201]$ from the allowed region, the corresponding coupler curve and desired trajectories are plotted in Figure 8a. Selecting the values of $[p] = [0.2499, 0.2501]$ and $[q] = [-0.4401, -0.4399]$ from the allowed region, the corresponding coupler curve and desired trajectories are plotted in Figure 8b. These two coupler curves are quite different, yet are able to achieve the same objective. A linkage designer may consider additional criteria to further refine the solution list without having to again worry about the desired response, as the performance of all solutions have already been certified.

Using all task elements from (23), Figure 9 shows the complete set of appropriate design solutions to the problem. This is equal to the intersection of the allowed regions from the precision point solutions and trajectory solutions. This is an important point. If a set of requirements are selected for an appropriate

synthesis routine, and all of the requirements are simultaneously considered, then it may be possible that no appropriate design solutions are obtained because one (or several) of the requirements are too stringent. By considering all of the requirements simultaneously the user does not know which requirement needs to be relaxed. Another approach is to consider each requirement independently, then to find the intersection of the appropriate design solutions resulting from each requirement. This may also be approached in a consecutive manner, such that the appropriate design solutions from the first requirement are used as the starting point for the second requirement, as so on.

These examples help to demonstrate the potential of the appropriate design methodology. Additional constraints and response elements may always be added to further refine the allowed region. As well, the bisection resolutions may be reduced to improve the classification of unknown design intervals.

6.2. Finding Appropriate Design Solutions

In the previous examples only the 2D problem was considered, but the appropriate synthesis routine is capable of solving the full 9-dimensional problem to determine every possible linkage design that satisfies the desired response; however, the full problem is generally very time consuming (on the order of days

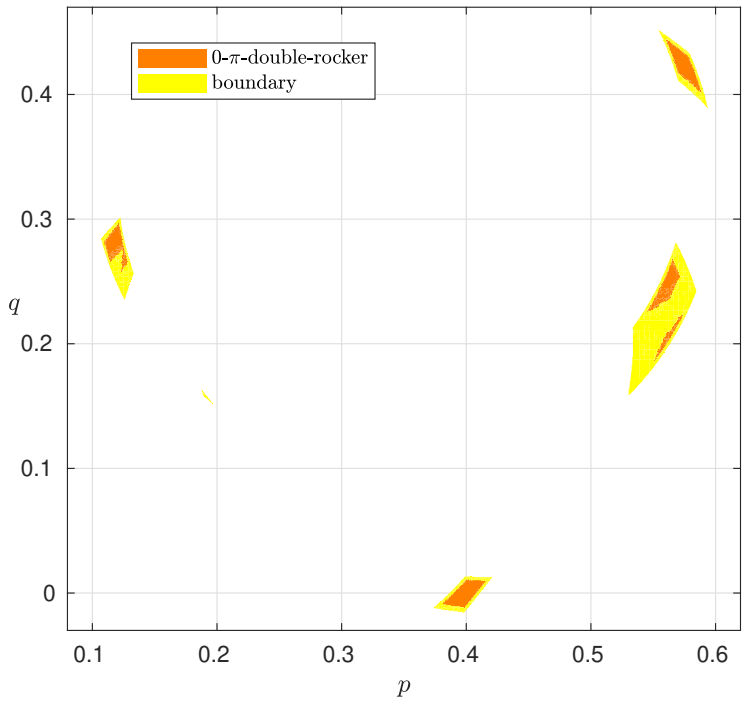
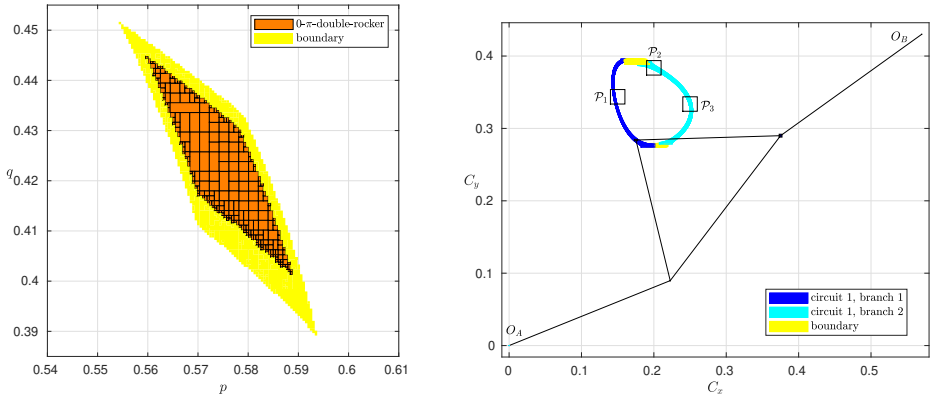


Figure 4: Appropriate design solutions for parameters p and q for precision points response elements in (23).



(a) (b)

Figure 5: a) Zoomed-in view of one of the disconnected allowed regions from Figure 4; b) the coupler curve corresponding to a solution from the allowed region.

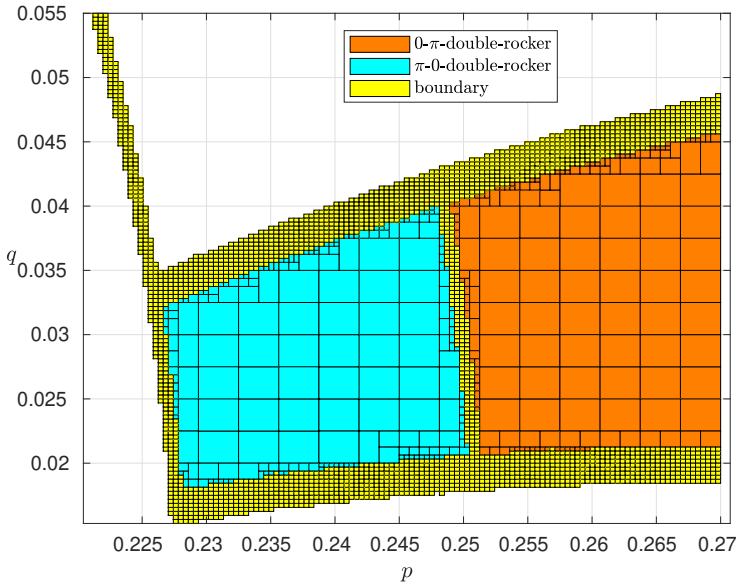


Figure 6: A subset of the design solutions for parameters p and q for the first trajectory response element in (23).

without parallel computing) and in most cases it is unnecessary to solve for every possible linkage design. The end-user may not require an infinite number of solutions. It may be desirable to determine several solutions to the problem, rather than obtaining the complete allowable region.

Two techniques are presented to assist in more quickly identifying appropriate design solutions. One technique considers reordering the design list in such

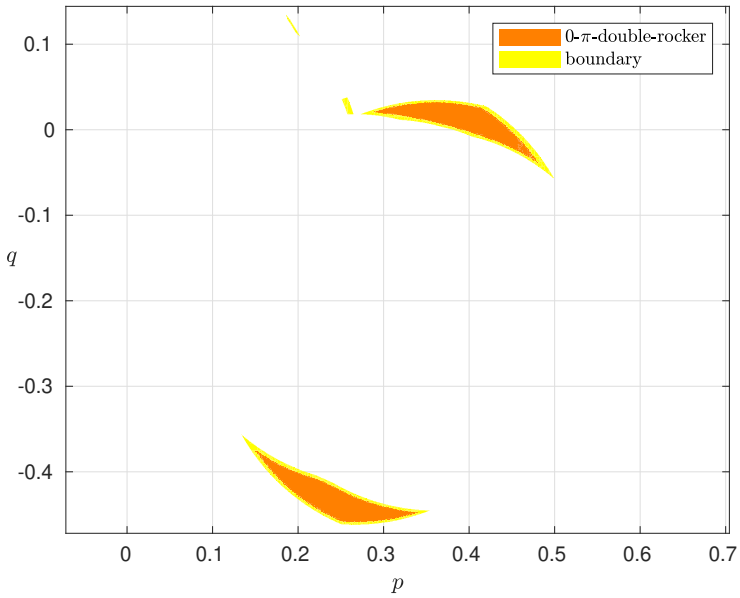


Figure 7: Design solutions for parameters p and q for all trajectory response elements in (23).

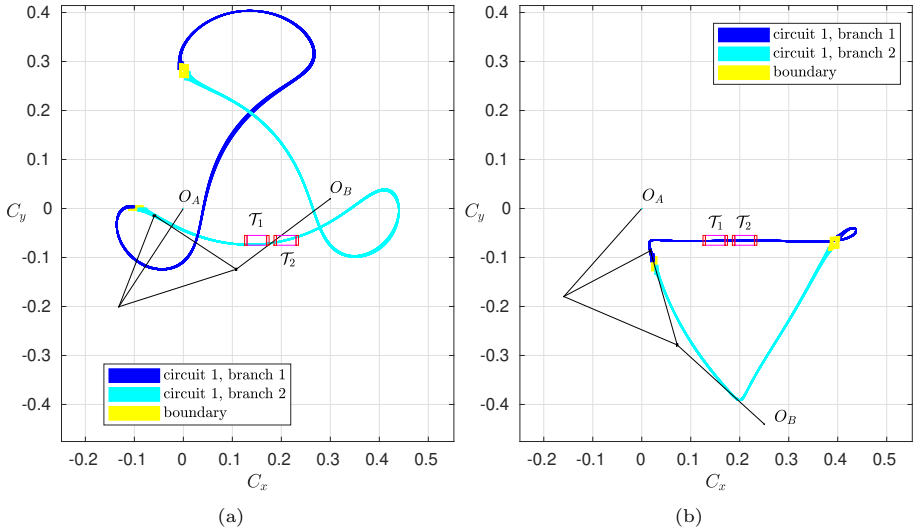


Figure 8: a-b) Coupler curves corresponding to a design from the allowed region.

a way that boxes more likely to contain a solution are tested first. In addition, it is possible to define a minimal distance between design solutions so that the neighbourhood of design solutions are not explored. Another technique considers applying conventional global optimization to find regions of interest, then inflating these regions and applying appropriate synthesis to determine appro-

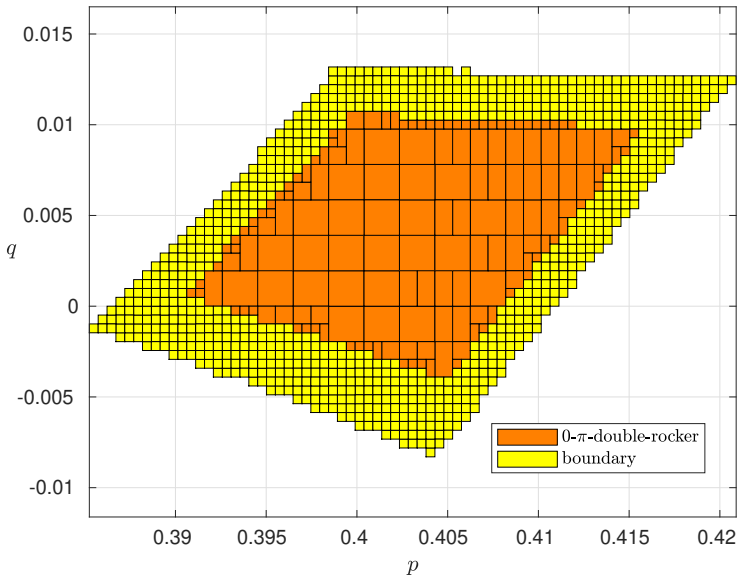


Figure 9: Design solutions for parameters p and q for all response elements in (23).

priate design solutions. Both of these techniques require a value which describes the offset of the coupler curve from the desired response, termed the *constraint violation*.

6.2.1. Constraint Violation

The total constraint violation of a design for a given task is equal to the sum of the constraint violations for each task element. The constraint violation associated with a precision point task element may be described by the generalized volume of the interval hull of $\mathcal{P}_{actual}$ of the linkage and $\mathcal{P}_{desired}$ of the precision point. These domains will always intersect since a non-intersection results in a non-solution and is removed from the search. The constraint value is normalized by the generalized volume of the precision point, such that a value of 1.0 implies that a solution is found for the precision point. The constraint violation for a precision point i (V_{p_i}) is therefore described as:

$$V_{p_i} = \frac{volume(\mathcal{P}_{actual} \square \mathcal{P}_{desired})}{volume(\mathcal{P}_{desired})} \tag{25}$$

and the constraint violation V_p associated with all m precision points is:

$$V_p = \sum_{i=1}^{m} V_{p_i} \tag{26}$$

For trajectories, the associated constraint violation may be described by the generalized volume of the interval hull of $\mathcal{T}_{actual}$ of the linkage and the desired

$\mathcal{T}_{desired}$ of the each trajectory end-point. Again, these domains will always intersect since a non-intersection results in a non-solution and is removed from the search. The constraint value is normalized by the generalized volume of the trajectory end-points. The constraint violation for an end-point j of trajectory i (V_{tij}) is therefore described as:

$$V_{tij} = \frac{volume(\mathcal{T}_{actual} \square \mathcal{T}_{desired})}{volume(\mathcal{T}_{desired})} \tag{27}$$

and the constraint violation V_t associated with all n trajectories is:

$$V_t = \sum_{i=1}^{n} V_{ti1} + V_{ti2} \tag{28}$$

The total constraint violation V for a given design is equal to the sum of V_p and V_t as

$$V = V_p + V_t \tag{29}$$

6.2.2. Reordering the Design List

The design list may be ordered by ascending values of constraint violation, such that the next design to be tested is the most likely to contain a solution. As well, global optimization routines may be used to minimize the constraint violation in order to find appropriate design solutions.

To reorder the design list, the constraint violation is divided by a factor which is proportional to the size of the design parameter box. This ensures that the search remains exploratory unless a very promising box is discovered. Here, this factor is set as the sum of the design parameter widths, such that a larger width decreases the value of constraint violation. Following the bisection, the design list is ordered by ascending values of constraint violation. This modification is tested on the three precision point elements in Eq. (23) using the same design parameter domains in Eq. (24) with the inflated parameters $v = p = q = [-1, 1]$. The solutions are restricted to have the same branch and circuit through each precision point. Table 1 presents the first 5 appropriate design solutions obtained for v, p and q. Considering the first solution, the values of of x, y, ψ and θ associated with each precision point are given in Table 2.

Table 1: Appropriate solutions for parameters v, p and q given three precision points.

	$[v]$	$[p]$	$[q]$
1	$[0.3184, 0.3193]$	$[0.1875, 0.1884]$	$[0.1573, 0.1582]$
2	$[0.3174, 0.3183]$	$[0.1875, 0.1884]$	$[0.1563, 0.1572]$
3	$[0.3184, 0.3193]$	$[0.1895, 0.1904]$	$[0.1563, 0.1572]$
4	$[0.3164, 0.3173]$	$[0.1895, 0.1904]$	$[0.1563, 0.1572]$
5	$[0.3174, 0.3183]$	$[0.1875, 0.1884]$	$[0.1573, 0.1582]$

Table 2: The corresponding values of x, y, ψ and θ for solution 1 of Table 1.

	$\mathcal{P}_1$	$\mathcal{P}_2$	$\mathcal{P}_3$
$[x]$	$[0.1527, 0.1576]$	$[0.1951, 0.1988]$	$[0.2462, 0.2596]$
$[y]$	$[0.3373, 0.34028]$	$[0.3877, 0.3906]$	$[0.3333, 0.3434]$
$[\psi]$	$[-0.8153, -0.8070]$	$[-0.5358, -0.5294]$	$[-2.0689, -2.0677]$
$[\theta]$	$[-0.8521, -0.8435]$	$[-0.5734, -0.5669]$	$[0.9094, 0.9104]$

6.2.3. Identifying Regions of Interest using Global Optimization

Global optimization routines can be applied to minimize the constraint violation in order to find appropriate design solutions. Here Differential Evolution [36] is applied to minimize the constraint violation. The variant DE/rand/1/bin is selected with a population size of 50. The full search space of the design parameters is considered and is initialized as $u = v = p = q = e = h = [-1, 1]$ and $r = s = c = [0.0100, 1]$. Uncertainties on the design parameters are neglected during the optimization in order to more quickly detect solutions. When a solution is found, the design parameters, which are exact, are saved. Since appropriate design solutions are of interest, the exact design parameters are inflated. This generates the regions of interest. These regions of interest can then be used as the starting domains of the appropriate synthesis routine. This initialization procedure quickly narrows in on a region which is known to contain a solution and likely has an appropriate design solution near. Table 3 presents several different solutions returned by the optimization routine. Each solution has the same branch and circuit at each precision point.

Table 3: Design solutions for all nine design parameters given three precision points.

	1	2	3
u	-0.9289	0.2212	-0.2683
v	-0.0932	0.9204	0.6631
p	0.2910	0.3546	0.6677
q	0.3199	-0.9982	-0.6887
r	0.0951	0.0706	0.0963
s	0.9363	0.2527	0.8318
c	0.4516	0.8053	0.7945
e	-0.8571	0.4755	0.3859
h	0.9240	-0.2638	-0.3640

Selecting the second design solution from Table 3, then inflating the design parameters and running the appropriate synthesis routine yields several solutions. Table 4 gives three adjacent appropriate design solutions. Figure 10 shows the coupler curve and desired response elements corresponding to the first appropriate design solution from Table 4.

Table 4: Appropriate design solutions for all nine design parameters given three precision points.

	1	2	3
$[u]$	$[0.2211, 0.2212]$	$[0.2211, 0.2212]$	$[0.2211, 0.2212]$
$[v]$	$[0.9204, 0.9205]$	$[0.9204, 0.9205]$	$[0.9204, 0.9205]$
$[p]$	$[0.3513, 0.3519]$	$[0.3507, 0.3512]$	$[0.3507, 0.3512]$
$[q]$	$[-1.0002, -1.0000]$	$[-1.0002, -1.0000]$	$[-0.9999, -0.9996]$
$[r]$	$[0.0705, 0.0707]$	$[0.0705, 0.0707]$	$[0.0705, 0.0707]$
$[s]$	$[0.2527, 0.2528]$	$[0.2527, 0.2528]$	$[0.2527, 0.2528]$
$[c]$	$[0.8053, 0.8054]$	$[0.8053, 0.8054]$	$[0.8053, 0.8054]$
$[e]$	$[0.4754, 0.4755]$	$[0.4754, 0.4755]$	$[0.4754, 0.4755]$
$[h]$	$[-0.2638, -0.2637]$	$[-0.2638, -0.2637]$	$[-0.2638, -0.2637]$

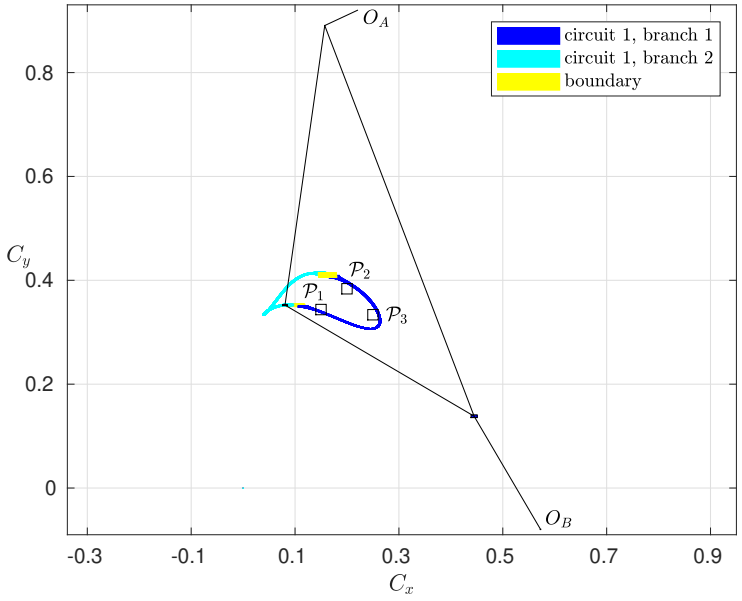


Figure 10: The coupler curve and desired response elements corresponding to the first appropriate design solution from Table 4.

7. Conclusion

The concept of appropriate design has been presented in this work and methods for the appropriate synthesis of the four-bar linkage have been developed. The proposed appropriate synthesis technique provides a new and robust way of accounting for uncertainties present in the geometric parameters of linkages during dimensional synthesis. These techniques rely on interval analysis and are able to provide reliable results, such that a physical instance of a design solution is guaranteed to achieve the desired performance. Furthermore, the approach developed is able to find every design solution to the synthesis problem, making it a useful design tool.

Exact descriptions of the coupler point are not possible with uncertainties on the design parameters, so routines are proposed for generating a new description of the coupler curve which properly represents the effects of the uncertainties in the coupler point. A desired task may be described by any combination of precision points and trajectories, each having a specified allowable error. Routines are proposed for the verification of precision points and trajectories for a given linkage design. These routines are utilized inside of an appropriate synthesis loop, such that all appropriate design solutions can be determined for a given task. Many examples are presented in the case studies section to demonstrate the capabilities of the proposed methods.

The appropriate synthesis routine performs well on low-dimensional design parameter spaces (2D-3D); however, on higher-dimensional problems, the time to find solutions increases exponentially. In order to make finding appropriate design solutions feasible for higher-dimensional problems, methods were introduced which rely on the constraint violation of a design for a given task. Conventional global optimization techniques are combined with appropriate synthesis in order to more quickly identify regions of interest and find appropriate design solutions within these regions. The performance of the proposed appropriate analysis and appropriate synthesis routines may be further improved by introducing more efficient heuristics. Distributed computing may also assist in significantly reducing computational time, as the presented methods are ideal for parallel computing.

Acknowledgement

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC). We also acknowledge the support of the Mitacs Globalink Research Award – Inria, as well as the Harrison McCain Foundation Visitorship Grant.

References

[1] K. Hunt, Kinematic Geometry of Mechanisms, Oxford engineering science series, Clarendon Press, 1978.

[2] C. Wampler, A. Morgan, A. Sommese, Complete solution of the nine-point path synthesis problem for four-bar linkages, Journal of Mechanical Design - Transactions of the ASME 114 (1) (1992) 153–159.

[3] S. Bai, J. Angeles, Coupler-curve synthesis of four-bar linkages via a novel formulation, Mechanism and Machine Theory 94 (2015) 177–187.

[4] S. Roberts, On three-bar motion in plane space, Proceedings of the London Mathematical Society s1-7 (1) (1875) 14–23.

[5] J. Cabrera, A. Simon, M. Prado, Optimal synthesis of mechanisms with genetic algorithms, Mechanism and Machine Theory 37 (10) (2002) 1165–1177.

[6] R. Bulatović, S. Stevan R. Dordević, On the optimum synthesis of a four-bar linkage using differential evolution and method of variable controlled deviations, Mechanism and Machine Theory 44 (1) (2009) 235–246.

[7] V. Goulet, W. Li, H. Cheong, F. Iorio, C.-G. Quimper, Four-Bar Linkage Synthesis Using Non-convex Optimization, Springer International Publishing, 2016, pp. 618–635.

[8] V. Unruh, P. Krishnaswami, A computer-aided design technique for semi-automated infinite point coupler curve synthesis of four-bar linkages, Journal of Mechanical Design 117 (1) (1995) 143–149.

[9] S. Coros, B. Thomaszewski, G. Noris, S. Sueda, M. Forberg, R. W. Sumner, W. Matusik, B. Bickel, Computational design of mechanical characters, ACM Trans. Graph. 32 (4) (2013) 83:1–83:12.

[10] A. Rugbani, K. Schreve, Modelling and analysis of the geometrical errors of a parallel manipulator micro-cmm, Vol. 371, 2012, pp. 105–117.

[11] J. Sovizi, A. Alamdari, S. Das, V. Krovi, Random matrix based uncertainty model for complex robotic systems, in: 2014 IEEE International Conference on Robotics and Automation (ICRA), 2014, pp. 4049–4054.

[12] G. Cui, H. Zhang, D. Zhang, F. Xu, Analysis of the Kinematic Accuracy Reliability of a 3-DOF Parallel Robot Manipulator, International Journal of Advanced Robotic Systems 12 (2) (2015) 15.

[13] Q. Zhao, J. Guo, J. Hong, Closed-form error space calculation for parallel/hybrid manipulators considering joint clearance, input uncertainty, and manufacturing imperfection, Mechanism and Machine Theory 142 (2019) 103608.

[14] S. Briot, I. A. Bonev, Accuracy analysis of 3-DOF planar parallel robots, Mechanism and Machine Theory 43 (4) (2008) 445–458.

[15] J.-P. Merlet, D. Daney, Dimensional synthesis of parallel robots with a guaranteed given accuracy over a specific workspace, in: Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005, pp. 942–947.

[16] Z. Zhan, X. Zhang, Z. Jian, H. Zhang, Error modelling and motion reliability analysis of a planar parallel manipulator with multiple uncertainties, Mechanism and Machine Theory 124 (2018) 55–72.

[17] S. E. Hraiech, A. Chebbi, Z. Affi, L. Romdhane, Error estimation and sensitivity analysis of the 3-UPU translational parallel robot due to design parameter uncertainties, Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 233 (8) (2019) 2713–2727.

[18] R. Kalnas, S. Kota, Incorporating uncertainty into mechanism synthesis, Mechanism and Machine Theory 36 (7) (2001) 843–851.

[19] X. Du, P. K. Venigella, D. Liu, Robust mechanism synthesis with random and interval variables, Mechanism and Machine Theory 44 (7) (2009) 1321–1337.

[20] K. Luo, J. Wang, X. Du, Robust mechanism synthesis with truncated dimension variables and interval clearance variables, Mechanism and Machine Theory 57 (2012) 71–83.

[21] F. Hao, J.-P. Merlet, Multi-criteria optimal design of parallel manipulators based on interval analysis, Mechanism and Machine Theory 40 (2) (2005) 157–171.

[22] J.-P. Merlet, D. Daney, Smart Devices and Machines for Advanced Manufacturing, Springer London, London, 2008, Ch. Appropriate Design of Parallel Manipulators, pp. 1–25.

[23] J. K. Pickard, J. A. Carretero, J.-P. Merlet, Appropriate analysis of the four-bar linkage, Mechanism and Machine Theory 139 (2019) 237–250.

[24] J. K. Pickard, J. A. Carretero, Appropriate synthesis of a crank rocker linkage, in: T. Uhl (Ed.), Advances in Mechanism and Machine Science, Springer International Publishing, 2019, pp. 1507–1516.

[25] J. K. Pickard, J. A. Carretero, J.-P. Merlet, Towards the appropriate synthesis of the four-bar linkage, Transactions of the Canadian Society for Mechanical Engineering 42 (1) (2018) 1–9.

[26] D. Stevenson, Floating-Point Working Group, Microprocessor Standards Subcommittee. IEEE standard for binary floating point arithmetic (IEEE/ANSI 754-1985), Tech. rep., IEEE (1985).

[27] R. Moore, R. Kearfott, M. Cloud, Introduction to Interval Analysis, Society for Industrial and Applied Mathematics, 3600 Market Street, 6th Floor, Philadelphia, PA, 19104-2688 USA, 2009.

[28] O. Lhomme, Consistency techniques for numeric csps, in: Proceedings of the 13th International Joint Conference on Artifical Intelligence - Volume 1, IJCAI'93, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993, pp. 232–238.

[29] F. Benhamou, F. Goualard, L. Granvilliers, J. Puget, Revising hull and box consistency, in: Int. Conf. ON Logic Programming, MIT press, 1999, pp. 230–244.

[30] B. Neveu, G. Trombettoni, Adaptive constructive interval disjunction, in: 2013 IEEE 25th International Conference on Tools with Artificial Intelligence, 2013, pp. 900–906.

[31] R. Moore, R. B. Kearfott, M. J. Cloud, Introduction to Interval Analysis, Society for Industrial and Applied Mathematics, 2009.

[32] R. Moore, Interval analysis, Prentice-Hall series in automatic computation, Prentice-Hall, 1966.

[33] J.-P. Merlet, Interval analysis for certified numerical solution of problems in robotics, International Journal of Applied Mathematics and Computer Science 19 (3) (2009) 399–412.

[34] R. Kearfott, N. I. Manuel, Intbis, a portable interval newton/bisection package, ACM Trans. on Mathematical Software 16 (2) (1990) 152–157.

[35] T. Chase, J. Mirth, Circuits and branches of single-degree-of-freedom planar linkages, Journal of Mechanical Design - Transactions of the ASME 115 (2) (1993) 223–230.

[36] R. Storn, K. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, Journal of Global Optimization 11 (4) (1997) 341–359.

Nomenclature

$0_A = (u, v)$ Coordinates of fixed end-point of input link

$0_B = (p, q)$ Coordinates of fixed end-point of output link (relative to 0_A)

$A = (A_x, A_y)$ Coordinates of moving end-point of input link

$B = (B_x, B_y)$ Coordinates of moving end-point of output link

$C = (C_x, C_y)$ Coordinates of coupler point

θ, ψ Input and output angles

a, b, c Edge lengths of triangle ABC

r, s, c, g Lengths of input, output, coupler, and fixed links

e, h Relative location of C for unique configuration

Δx Uncertainty of x

$\square$ Interval hull operation

$\mathbf{d}, [\mathbf{d}]$ Vectors of exact and appropriate design parameters

$[\mathbf{u}], [\mathbf{k}]$ Vectors of unknown and known appropriate parameters

β Bisection resolution

$\mathcal{L}_u$ List of vectors of unknown parameters

$\mathcal{L}_d, \mathcal{L}_{d_s}, \mathcal{L}_{d_n}, \mathcal{L}_{d_b}$ List of design parameters, solutions, non-solutions, and boundaries

$\mathcal{L}_{class}$ List of associated linkage classifications

$\mathcal{L}_{allow\_class}$ List of allowable linkage classifications

$\mathcal{L}_{branch}$ List of associated coupler curve branches

$\mathcal{L}_{circuit}$ List of associated coupler curve circuits

$\mathcal{R}, \mathcal{P}, \mathcal{T}$ Desired response, precision point response element, and trajectory response element

$f(C_x([t]), C_y([t]))$ Trajectory parametric curve

α, δ Trajectory allowable error and end-point width

$\mathcal{P}^*, \mathcal{T}^*$ Precision point and trajectory response element solutions

V_p, V_t, V Constraint violations (precision points, trajectories, combined)