CONCURRENT TYPE AND DIMENSIONAL CONTINUOUS APPROXIMATE FUNCTION GENERATOR SYNTHESIS FOR ALL PLANAR FOUR-BAR MECHANISMS

Zachary Copeland[1], Mirja Rotzoll[1], M. John D. Hayes[1]
[1]*Department of Mechanical and Aerospace Engineering, Carleton University, Ottawa, ON, Canada*
Email: ZackCopeland@cmail.carleton.ca; mirja.rotzoll@carleton.ca; john.hayes@carleton.ca

ABSTRACT

This paper presents a novel algorithm for the continuous approximate synthesis of planar four-bar function generators using an algebraic form of the input-output equation for any given architecture minimising the structural error. Each identified mechanism architecture is evaluated relative to a structural error minimising mechanism generated with a Newton-Gauss non-linear optimisation procedure to estimate the performance of the new algorithm. Finally, an algorithm is presented wherein both type and dimensional synthesis of the function generating mechanism may be accomplished simultaneously without reliance on heuristics.

Keywords: continuous approximate kinematic synthesis; structural error minimisation; planar four-bar function generators.

SYNTHÈSES SIMULTANÉES DE TYPE ET DIMENSIONNELLE APPROXIMATIVES CONTINUES DE GÉNÉRATEURS DE FONCTION POUR TOUS LES MÉCANISMES PLAN À QUATRE BARRES

RÉSUMÉ

Cet article présente un nouvel algorithme pour la synthèse approximative continue de générateurs de fonctions plan à quatre barres en utilisant une forme algébrique de l'équation d'entrée-sortie pour toute architecture donnée minimisant l'erreur structurelle. Chaque architecture de mécanisme identifiée est évaluée par rapport à un mécanisme de minimisation d'erreur structurelle généré avec une procédure d'optimisation non linéaire de Newton-Gauss afin d'estimer les performances du nouvel algorithme. Enfin, un algorithme est présenté dans lequel à la fois les synthèses de type et dimensionelle du mécanisme de génération de fonction peuvent être accomplis simultanément sans nécessiter d'heuristique.

Mots-clés : synthèse approximate continue ; minimisation d'erreur structurelle ; liaison plane à quatre barres.

1. INTRODUCTION

The study of planar four-bar linkages involves a large variety of problems, ranging from those concerned with guiding a point along a specific curve (the coupler curve), to guiding a rigid body (the coupler) through a series of positions and orientations (the Burmester problem), to further refining these guidance problems to generate specific trajectories [1] (poses with an instant in time associated with each discrete pose), while also containing problems which are concerned with the transmission of forces and torques [2] through the links, or designing an optimally balanced linkage [3]. An additional subset of this gamut of problems is the function generation problem; it consists, primarily, of developing a mechanism which is able to approximate, in some sense, an input-output (IO) relationship for a given planar linkage architecture comprising RR-, RP-, PR-, or PP-dyads[1]. That is to say, given some desired functional relationship between the input and output links of a planar four-bar mechanism, develop the linkage which best approximates this function over some desired range. This IO function generation problem is the focus of this paper. Fig. 1 illustrates a function generating four-bar RRRR linkage. If link a_1 is the input link and link a_3 is the output the function is specified as $\theta_4 = f(\theta_1)$.

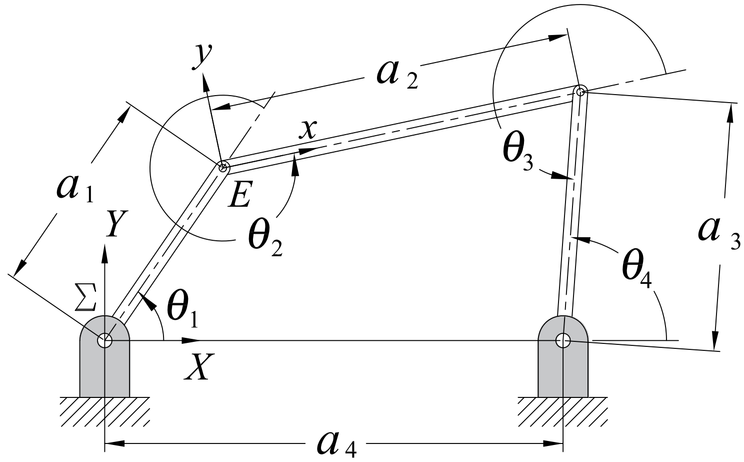


Fig. 1. A general planar 4R function generator.

The function generation problem is often focused on either the *design*, or *structural error* minimisation. The design error indicates the residual incurred by a specific linkage in satisfying its synthesis equations, whereas the structural error is the difference between the prescribed and generated linkage output values for a given input value [4]. The design error minimisation can be expressed as a linear least squares problem, while the more relevant problem of the structural error is a highly non-linear problem which requires an iterative optimisation approach in order to compare the generated function to the prescribed function [5].

2. ALGEBRAIC IO FUNCTIONS

It has been shown in [5, 6] that as the cardinality of the data set which is used for the design error optimisation of any given planar four-bar linkage tends to infinity, the solution for the design error minimising linkage parameters converges to that of the structural error; this indicates that the solution to the non-linear structural error problem is implied by the solution of the linear design error problem over an infinite number of IO pairs. From this idea, the concept of continuous approximate IO function generator synthesis arose.

Classically, the solution to these problems is conducted using the Freudenstein equation [7], and it has

[1] R and P indicate revolute and prismatic joints connecting a pair of rigid links, also known as R- and P-pairs.

been demonstrated in [6] that this form of the function generator problem can be integrated in order to derive the design and structural error minimising linkage for a given IO function over some arbitrary range; however, the expressions and functions required to perform this minimisation are exceedingly complicated, and a more compact form of the problem is desired. It will herein be shown that a simplification of the trigonometric Freudenstein IO relationship is achieved with the algebraic IO equation.

The algebraic IO equations were derived in [8, 9] through the use of displacement constraints projected into a planar subset of Study's soma space for each of the planar four-bar function generator architectures: four-bar linkages comprised of two RR-dyads; one RR-dyad and one PR-dyad; and two PR-dyads. We will commence with the algebraic IO equation for the planar RRRR mechanism, which will henceforth be referred to as a 4R mechanism, see Eq. (1). In this equation the IO variables v_1 and v_4 are the tangent of the half angle parameters of the input and output angles θ_1 and θ_4, illustrated in Fig. 1:

$$Av_1^2v_4^2+Bv_1^2+Cv_4^2+D-8a_1a_3v_1v_4=0; \tag{1}$$

where

$$\begin{bmatrix} A \\ B \\ C \\ D \\ -8a_1a_3 \end{bmatrix} = \begin{bmatrix} (a_1-a_2-a_3+a_4)(a_1+a_2-a_3+a_4) \\ (a_1-a_2+a_3+a_4)(a_1+a_2+a_3+a_4) \\ (a_1-a_2+a_3-a_4)(a_1+a_2+a_3-a_4) \\ (a_1+a_2-a_3-a_4)(a_1-a_2-a_3-a_4) \\ -8a_1a_3 \end{bmatrix}. \tag{2}$$

For an RRRP mechanism, the IO parameters are the input angle parameter v_1 and the output P-pair linear excursion a_3 while the constant design parameters are link lengths a_1, a_2, a_4, and the slider inclination angle parameter v_4, illustrated in Fig. 2a. Following an identical derivation methodology the RRRP algebraic IO equation is obtained:

$$A{a_3}^2{v_1}^2+Ca_3{v_1}^2-8a_1a_3v_1v_4+B{a_3}^2+E{v_1}^2+Da_3+F=0; \tag{3}$$

such that

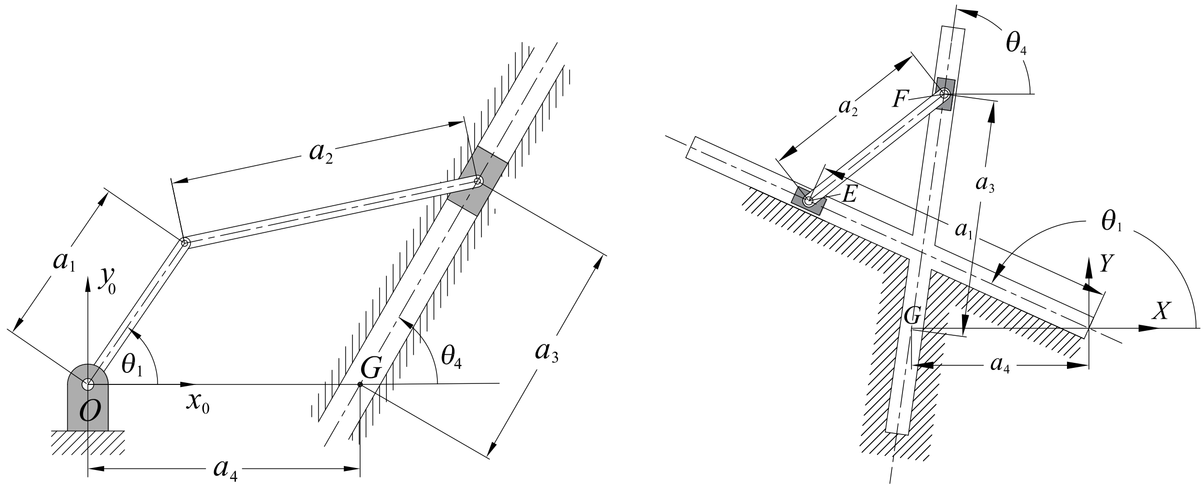
$$\begin{bmatrix} A \\ B \\ C \\ -8a_1v_4 \\ D \\ E \\ F \end{bmatrix} = \begin{bmatrix} v_4^2+1 \\ v_4^2+1 \\ -2(v_4-1)(v_4+1)(a_1+a_4) \\ -8a_1v_4 \\ 2(v_4-1)(v_4+1)(a_1-a_4) \\ (v_4^2+1)(a_1+a_2+a_4)(a_1-a_2+a_4) \\ (v_4^2+1)(a_1+a_2-a_4)(a_1-a_2-a_4) \end{bmatrix}. \tag{4}$$

Following from these two cases, the algebraic IO equation for a PRRP mechanism, with IO variables a_1 and a_4 and constant design parameters v_1, a_2, and v_4, see Fig. 2b, can be obtained in the same way, yielding:

$$A{a_1}^2+B{a_3}^2+Ca_1a_3+Da_1+Ea_3+F=0; \tag{5}$$

where

$$\begin{bmatrix} A \\ B \\ C \\ D \\ E \\ F \end{bmatrix} = \begin{bmatrix} (v_4^2+1)(v_1^2+1) \\ (v_4^2+1)(v_1^2+1) \\ -2(v_1v_4-v_1+v_4+1)(v_1v_4+v_1-v_4+1) \\ 2a_4(v_4^2+1)(v_1-1)(v_1+1) \\ -2a_4(v_4-1)(v_4+1)(v_1^2+1) \\ -(v_4^2+1)(v_1^2+1)(a_2-a_4)(a_2+a_4) \end{bmatrix}. \tag{6}$$



(a) A general planar RRRP mechanism.

(b) A general planar PRRP mechanism.

Fig. 2. Planar RRRP and PRRP mechanisms.

Typically, the design error minimisation for the function generation problem may be undertaken from this juncture; after taking some set of $n > 3$ IO pairs over the desired input range using the desired IO function and appropriate variables for the desired architecture, the error residual of this algebraic IO equation is minimised in order to obtain the design error minimising linkage parameters over the desired input range. However, given that the endeavour of the work presented herein is towards the continuous approximation method, and not a demonstration of the discrete method (see [9] for these discussions), some alterations to the algebraic IO equation must be made so as to accommodate the integration of the resulting equation in order to obtain an effective cardinality of infinity for the set of IO pairs for the minimisation step.

3. CONTINUOUS APPROXIMATE SYNTHESIS VIA ALGEBRAIC IO CURVE

3.1. Continuous Approximate Synthesis Algorithm Overview

While the following sections of this paper will cover, in detail, the algorithm used for each specific planar function generator architecture case. Each of these three algorithms follow an identical set of steps which differ only in the algebraic input-output equation used to initiate the process. These steps are as follows:

1. Square the desired algebraic input-output relationship for the planar four bar linkage architecture in question.
2. Separate this squared function into a vector containing the linkage parameters and a synthesis matrix, S.
3. Substitute the desired functional relationship between the input and output variables into the output parameter in the synthesis matrix, S.
4. Integrate this matrix, numerically, over the desired bounds for the approximation.
5. Expand the matrix-vector formulation with the integrated matrix.
6. Minimise the residual of this expanded synthesis equation over the field of real numbers.

After the minimisation procedure outlined in the final step, the resulting linkage parameters are the ones which will minimise the residual of the synthesis equation for the approximation over the desired range, thus best approximating the desired input-output relationship. After this step, the actual generated input-output function may be solved directly from the algebraic input-output equation with the optimised linkage parameters substituted. From this point, each case will be described in detail with specific examples of the completed synthesis problem and a comparison with the discrete approximation problem.

3.2. Continuous Approximate Synthesis for the 4R Function Generator

First, we begin by squaring the algebraic IO equation in order to eliminate the residual error values that are equal in magnitude yet opposite in sense, thus effectively annihilating the effects of these errors from the approximation. In order to accomplish this, the equation is broken into a matrix and vector representation, where the matrix contains all of the IO variables (in the 4R case, (v_1, v_4) respectively), while the vector which pre- and post-multiplies this matrix contains the linear functions of the linkage parameters in Eq. (2). Hence, the squared IO equation for all 4R linkages can be expressed as,

$$\begin{bmatrix} A & B & C & D & -8a_1a_4 \end{bmatrix} S(v_1, v_4) \begin{bmatrix} A \\ B \\ C \\ D \\ -8a_1a_4 \end{bmatrix} = 0, \tag{7}$$

where,

$$S(v_1, v_4) = \begin{bmatrix} v_1^4v_4^4 & 2v_1^4v_4^2 & 2v_1^2v_4^4 & 2v_1^2v_4^2 & 2v_1^3v_4^3 \\ 0 & v_1^4 & 2v_1^2v_4^2 & 2v_1^2 & 2v_1^3v_4 \\ 0 & 0 & v_4^4 & 2v_4^2 & 2v_1v_4^3 \\ 0 & 0 & 0 & 1 & 2v_1v_4 \\ 0 & 0 & 0 & 0 & v_1^2v_4^2 \end{bmatrix}. \tag{8}$$

In a second step, some desired function is specified for planar 4R linkages as $v_4 = f(v_1)$, is then substituted into Eq. (8) yielding

$$S(v_1, f(v_1)) = \begin{bmatrix} v_1^4f(v_1)^4 & 2v_1^4f(v_1)^2 & 2v_1^2f(v_1)^4 & 2v_1^2f(v_1)^2 & 2v_1^3f(v_1)^3 \\ 0 & v_1^4 & 2v_1^2f(v_1)^2 & 2v_1^2 & 2v_1^3f(v_1) \\ 0 & 0 & f(v_1)^4 & 2f(v_1)^2 & 2v_1f(v_1)^3 \\ 0 & 0 & 0 & 1 & 2v_1f(v_1) \\ 0 & 0 & 0 & 0 & v_1^2f(v_1)^2 \end{bmatrix}. \tag{9}$$

Once this substitution has been made, the resulting matrix is integrated between the bounds desired for the approximation, leading to the following expression, required for the minimisation algorithm,

$$\min_{(a_1,a_2,a_3,a_4)\in\mathbb{R}} \left(\begin{bmatrix} A & B & C & D & -8a_1a_4 \end{bmatrix} \int_{v_{1min}}^{v_{1max}} S(v_1, f(v_1)) \begin{bmatrix} A \\ B \\ C \\ D \\ -8a_1a_4 \end{bmatrix} \right). \tag{10}$$

In Eq. (10), the elements $[A, B, C, D]$ correspond to the linear factors defined in Eq. (2). However, in general, no arbitrary set of link lengths may be assumed for planar function generators as a starting point for the minimisation algorithm, as these procedures are immensely sensitive to the initial guess.

In order to acquire an effective initial guess to start the solver the problem must first have a starting set of link length values which are, in some sense, relatively close to the desired optimal linkage. In order to accomplish this, one will solve the exact synthesis problem for this linkage; given three points within the desired IO range of the mechanism, using the same $v_4 = f(v_1)$ relationship, and normalising the linkage leading to $a_4 = 1$. Given the initial set of link lengths, and a choice set of constraints for the optimiser, the linkage of best fit may be obtained. Defining the constraints that are required for these linkages to represent a real planar linkage is somewhat straight forward; it is simply required that $a_i^2 \geq 0$, forcing the link lengths to be real numbers. Negative link lengths can not be discarded, as given the range and function, any of the link lengths may in fact be negative in value, thus this constraint is designed solely to discard the imaginary solutions to this problem. While the concept of a negative length may appear to be fundamentally flawed insofar as the fact that a length can not be negative, it is more accurate to think of a link length as a directed distance on a vector which connects two revolute joint centres A and B. While a positive link length parameter may indicate that the directed distance connects the revolute joint centres in order of AB, a negative link length parameter may be thought of as connecting the same two revolute joint centers in order of BA, opposite in sense to the positive ordering.

Given the novelty of this approach, the following examples for all planar function generating linkage architectures, where the coupler can have general plane motion, will be presented and compared to existing solutions within the literature. For example, the following demonstration for the 4R linkage architecture, will be computed with the same function published in [8]. The desired function is

$$v_4 = 2 + \tan\left(\frac{v_1}{v_1^2+1}\right), \tag{11}$$

where (v_1, v_4) represent the tangent half angle parameters associated with the input and output link orientation, respectively. From this point, the functional relationship $v_4 = f(v_1)$ is substituted into Eq. (8). Once this substitution has been made, the entire matrix is integrated between the bounds desired for the approximation; in this case $v_1 = 0..2$. Upon completion of this integration, the squared IO equation can be used to identify design parameters a_1, a_2, a_3, and a_4 that minimise the structural error.

Reported in [8] are the link lengths optimised with a Newton-Gauss approach using a discrete IO set whose cardinality is 10, which are

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} -0.23 \\ 1.20 \\ 1.43 \\ 1 \end{bmatrix}. \tag{12}$$

Whereas the link lengths derived from the continuous approximation method are,

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} -0.22 \\ 1.18 \\ 1.43 \\ 1 \end{bmatrix}, \tag{13}$$

showing an extremely high degree of agreement between the two results. Fig. 3 shows both the precision point and optimised linkages in broken lines, with the solid line representing the desired function from Eq. (11). The precision point and optimised linkage functions are both solved directly from the algebraic IO equation as polynomials in the form of $v_4 = f(v_1)$.

Given the proximity of these curves to one another, a simple visual inspection is insufficient to determine the magnitude of the differences between the curves, thus a different method using the integrals of

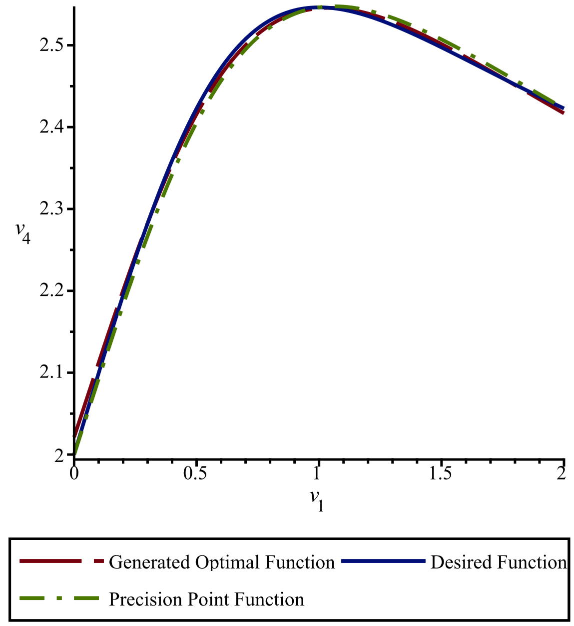


Fig. 3. Comparison of RRRR desired and generated functions.

these functions to compute the percentage error from one method to the other is used. For the 4R function generator case, the following errors are observed relative to the desired function,

$$\text{Precision Point Method} = 0.0953\%, \tag{14}$$

$$\text{Continuous Approximate Optimised Function} = -0.0126\%, \tag{15}$$

indicating a reduction in the percentage error associated with the continuous approximation method by a factor of approximately 7.556.

3.3. Continuous Approximate Synthesis for the RRRP Function Generator

Given the nature of the continuous approximate design error minimisation, the approach is easily modified for any desired planar four-bar planar function generator topology. First, a single revolute joint at the distal end of Fig. 1 will be replaced by a slider, for a linkage configuration shown in Fig. 2a. After Eq. (3) is squared, Eq. (16) may be pre- and post-multiplied by Eq. (4) to obtain the full squared IO function for the RRRP linkage architecture:

$$S(v_1, a_3) = \begin{bmatrix} v_1^4a_3^4 & 2v_1^2a_3^4 & 2v_1^4a_3^3 & 2v_1^3a_3^3 & 2v_1^2a_3^3 & 2v_1^4a_3^2 & 2v_1^2a_3^2 \\ 0 & a_3^4 & 2v_1^2a_3^3 & 2v_1a_3^3 & 2a_3^3 & 2v_1^2a_3^2 & 2a_3^2 \\ 0 & 0 & v_1^4a_3^2 & 2v_1^3a_3^2 & 2v_1^2a_3^2 & 2v_1^4a_3 & 2v_1^2a_3 \\ 0 & 0 & 0 & v_1^2a_3^2 & 2v_1a_3^2 & 2v_1^3a_3 & 2v_1a_3 \\ 0 & 0 & 0 & 0 & a_3^2 & 2v_1^2a_3 & 2a_3 \\ 0 & 0 & 0 & 0 & 0 & v_1^4 & 2v_1^2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{16}$$

After substituting some desired functional IO relationship, which for the planar RRRP function generating linkages is specified as $a_3 = f(v_1)$, into Eq. (16) and integrating between the bounds desired for the approximation at hand, the expression used in the minimisation algorithm becomes,

$$\min_{(a_1,a_2,a_4,v_4)\in\mathbb{R}} \left(\begin{bmatrix} A & B & C & -8a_1v_4 & D & E & F \end{bmatrix} \int_{v_{1min}}^{v_{1max}} S(v_1, f(v_1)) \begin{bmatrix} A \\ B \\ C \\ -8a_1v_4 \\ D \\ E \\ F \end{bmatrix} \right), \tag{17}$$

where $[A,B,C,D,E,F]$ are the parameter terms for the algebraic input-output equation from Eq. (4). Once again, in order to compare the continuous approximation methods with previously accepted values, a function from [9] will be used. Specifically, the $a_3 = f(v_1)$ IO relationship for this example is,

$$a_3 = \frac{-{v_1}^2 + 1}{{v_1}^2 + 1}, \tag{18}$$

between the bounds of $v_1 = -3..3$. Upon solving the precision point problem and integrating Eq. (16) with the desired function in place of a_3, the squared IO function may be minimised using the same constraints as were implemented in the 4R case. From [9], the following linkage parameters were identified following a Newton-Gauss iterative minimisation routine over fifty points within the design space,

$$\begin{bmatrix} a_1 \\ a_2 \\ a_4 \\ v_4 \end{bmatrix} = \begin{bmatrix} 0.9426 \\ 1.1587 \\ 1 \\ 1.5 \cdot 10^{-5} \end{bmatrix}, \tag{19}$$

while the linkage parameters identified through the continuous approximate synthesis methods are,

$$\begin{bmatrix} a_1 \\ a_2 \\ a_4 \\ v_4 \end{bmatrix} = \begin{bmatrix} 0.9554 \\ 1.1894 \\ 1 \\ 1.17 \cdot 10^{-10} \end{bmatrix}, \tag{20}$$

again showing a very high level of agreement between the solution to the non-linear Newton-Gauss solution, and the linear continuous approximate synthesis methods. Fig. 4 shows three functions for this RRRP function generator example; in broken lines, it shows the precision point method alongside the optimal linkage, while the solid line represents the desired IO relationship.

Once again, the integral of the equations resulting from this optimisation and precision point method, extracted directly from the algebraic IO function generator equation as a polynomial in the form of $a_3 = f(v_1)$, is compared to the ideal function,

$$\text{Precision Point Method} = -117.89\%, \tag{21}$$

$$\text{Continuous Approximate Optimised Function} = -52.17\%, \tag{22}$$

showing a reduction in the error relative to the precision point method of a factor of approximately 2.26.

3.4. Continuous Approximate Synthesis for the PRRP Function Generator

Continuing from the previous section, the first revolute joint in the chain will now be replaced with a prismatic joint, creating a PRRP planar function generator, illustrated in Fig. 2b. Once again, Eq. (5) is

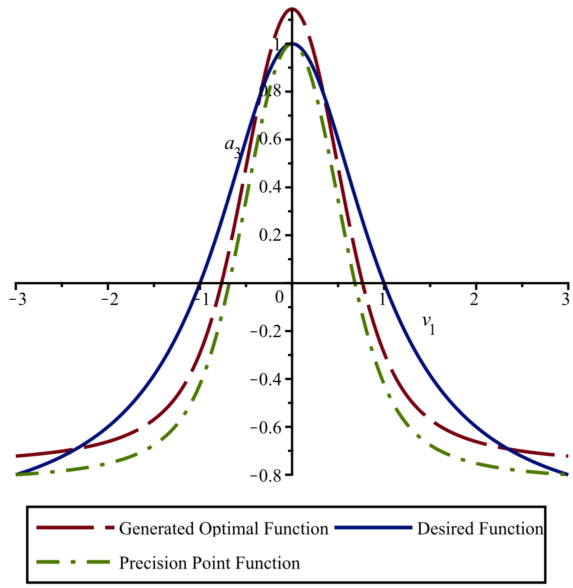


Fig. 4. Comparison of RRRP desired and generated functions.

squared, following which it may be written as Eq. (23) which is pre and post-multiplied by Eq. (6) to obtain the full squared algebraic IO function generator equation for the PRRP architecture.

$$S(a_1,a_3) = \begin{bmatrix} a_1^4 & 2a_1^2a_3^2 & 2a_1^3a_3 & 2a_1^3 & 2a_1^2a_3 & 2a_1^2 \\ 0 & a_3^4 & 2a_1a_3^3 & 2a_1a_3^2 & 2a_3^3 & 2a_3^2 \\ 0 & 0 & a_1^2a_3^2 & 2a_1^2a_3 & 2a_1a_3^2 & 2a_1a_3 \\ 0 & 0 & 0 & a_1^2 & 2a_1a_3 & 2a_1 \\ 0 & 0 & 0 & 0 & a_3^2 & 2a_3 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{23}$$

Again, a desired functional relationship, which is specified for planar PRRP linkages as $a_3 = f(a_1)$ can be substituted into Eq. (23). After integrating this expression between the desired bounds of this function generator, the expression for the minimisation then becomes,

$$\min_{(a_2,a_4,v_1,v_4)\in\mathbb{R}} \left(\begin{bmatrix} A & B & C & D & E & F \end{bmatrix} \int_{a_{1min}}^{a_{1max}} S(a_1, f(a_1)) \begin{bmatrix} A \\ B \\ C \\ D \\ E \\ F \end{bmatrix} \right), \tag{24}$$

where $[A,B,C,D,E,F]$ are the linkage parameter vectors for the PRRP architecture listed in Eq. (6). Comparing the standard methodologies to the continuous approximate methodology will once again be completed through the use of a test case included in [9]. The $a_3 = f(a_1)$ IO function in this case is,

$$a_3 = \cos(a_1), \tag{25}$$

over the range of $a = 0..2$. The solution will proceed in identical fashion to the previous cases, with the precision point method being used to generate initial guesses, at which point the squared algebraic IO matrix in Eq. (23) is integrated between the bounds of this function generator problem, and subsequently minimised assuming that the linkage parameters are real numbers. From [9], it is known that the structural error minimising linkage parameters resulting from the nonlinear Newton-Gauss optimisation procedure are,

$$\begin{bmatrix} a_2 \\ a_4 \\ v_4 \\ v_1 \end{bmatrix} = \begin{bmatrix} 2.0313 \\ 1 \\ -1.1868 \\ 0.1353 \end{bmatrix}, \tag{26}$$

while the optimal link lengths from the continuous approximate algebraic IO method are,

$$\begin{bmatrix} a_2 \\ a_4 \\ v_4 \\ v_1 \end{bmatrix} = \begin{bmatrix} 2.0364 \\ 1 \\ -1.1986 \\ 0.1291 \end{bmatrix}, \tag{27}$$

showing again, a large degree of agreement between the discrete approximate structural error minimising linkage, and the continuous approximate design error minimising linkage. Following this procedure, Fig. 5 shows the results, with the desired function presented with a solid line, and the broken lines representing the precision point method and continuous approximate method.

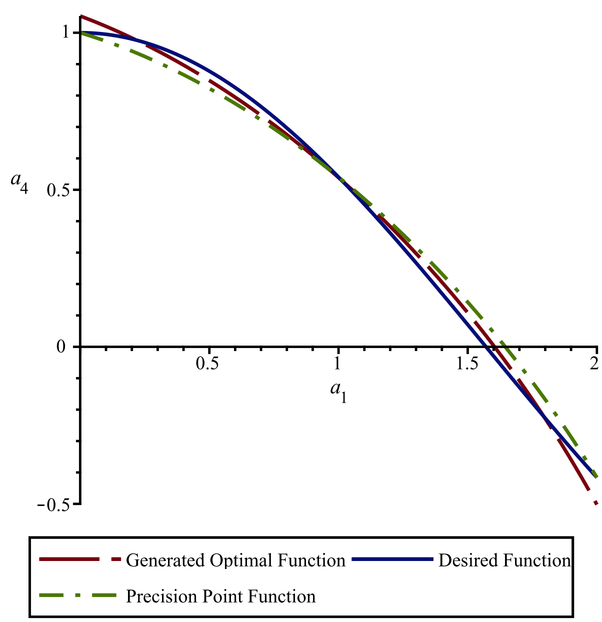


Fig. 5. Comparison of PRRP desired and generated functions.

Once again, these functions are compared via integration over the bounds of the approximation range in order to compare the precision point method to the continuous approximation method. The percentage errors relative to the desired IO function are,

$$\text{Precision Point Method} = -1.300\%, \tag{28}$$
$$Continuous\,Approximate\,Method = -0.054\%, \tag{29}$$

showing a reduction in the error between these methods of a factor of approximately 23.84.

4. COMBINED TYPE AND DIMENSIONAL SYNTHESIS FOR PLANAR FOUR-BAR FUNCTION GENERATORS

Typically, when a function generator is being optimised, the first step that the designer takes upon finalising the function which is to be approximated over some given range is to choose the linkage architecture that will be used to generate it. However, it is possible that the linkage architecture which was chosen by the designer is not necessarily the globally optimal planar four-bar linkage for the generation of this function over the desired range. This may be remedied by designers who are intimately familiar with linkage design, but this is a process that relies much on the skill of the individual and very little on the efficiency of the optimisation process itself. Therefore, it is proposed that the type and dimensional synthesis for planar four-bar function generating linkages is combined so as to remove this decision based portion of the design process in general. In essence; given a function to be generated by a planar four-bar linkage, which linkage architecture best approximates it over the given range, and how does the function that is generated by the linkage compare to the desired function? Given some function, $h = f(t)$, the following substitutions are proposed for each planar four-bar linkage architecture.

Variable	**Linkage Architecture**		
General Case	*4R*	*RRRP*	*PRRP*
t	v_1	v_1	a_1
h	v_4	a_3	a_3

Table 1. Variable substitutions used for continuous approximate type and dimensional planar function generator synthesis.

For the purposes of the following concurrent type and dimensional synthesis example, the function to be approximated will be,

$$h = \frac{-t^2+1}{t^2+1}, \tag{30}$$

such that $t = 0..1$. Following the exact procedure as depicted in the aforementioned demonstrative cases, each mechanism may be synthesised and compared to each other. For the sake of brevity, Table 2 shows all the mechanisms with the optimal linkage parameters identified, with the appropriate IO variables for the architecture at hand being represented by the variable name itself.

Linkage Parameter	**Linkage Architecture**		
	4R	*RRRP*	*PRRP*
v_1	v_1	v_1	1.995
v_4	v_4	0.0903	0.5114
a_1	$2.6x10^{-5}$	0.7288	a_1
a_2	0.9999	1.247	1.707
a_3	$2.3x10^{-5}$	a_3	a_3
a_4	1	1	1

Table 2. All identified parameters for the continuous approximate concurrent type and dimensional synthesis of a planar function generator.

Once all of the error-minimising linkage parameters have been computed, the explicit IO functions that each of these mechanisms define are computed. These explicit functions are plotted in Fig. 6. At first glance, it would appear that the 4R function generating architecture is, by far, the best suited for this approximation. However; one must pay close attention to the link lengths and what these values imply about the linkage

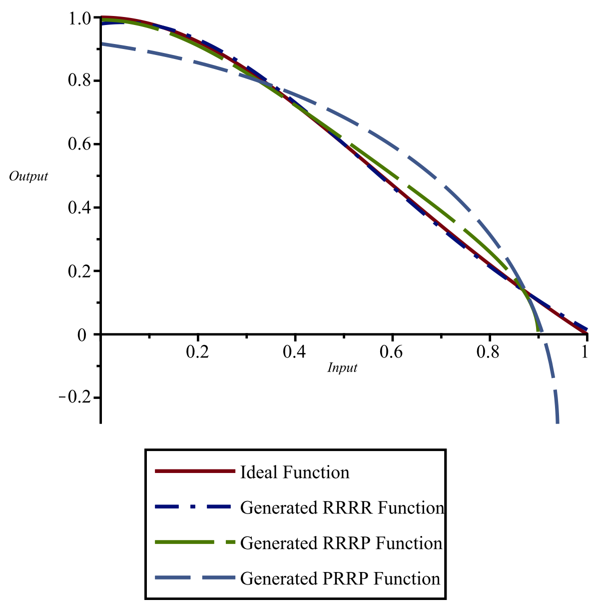


Fig. 6. Comparison of 4R, RRRP, and PRRP desired and generated functions.

itself. The parameters for the RRRP and PRRP linkages are all reasonable values, but the 4R linkage values are peculiar; the relative magnitude of (a_1, a_3) compared to a_2 imply that the coupler and base link of this four-bar linkage are over ten thousand times as long as the input and output links. This strange occurrence is a result of two facts; the solver is constrained explicitly so that $a_i^2 \geq 0$, meaning that these values can not ever be identically equal to zero, and secondly that the orientation of a point does not exist - given a vanishingly small length, it is impossible to define its orientation, and it can either be thought of as not having an orientation, or possessing every possible orientation simultaneously. Were the link lengths during this optimisation allowed to converge to zero, it would be observed that links a_1 and a_2 both converge to be identically zero; indeed, $a_1 = a_3 = 0$ and $a_2 = a_4 = 1$ is a consistent trivial solution to any planar function generation problem with the 4R architecture and must be discarded as a result. Given this fact, the comparison of the 4R function generator will be omitted in the subsequent analysis. Table 3 shows the percentage error relative to the desired function of each of the polynomials generated by the RRRP and PRRP linkages respectively.

	Desired Function	**Generated RRRP**	**Generated PRRP**
Percentage Error	0%	0.0143%	0.3685%

Table 3. Percentage error for all viable planar four-bar function generators.

From both Fig. 6 and Table 3, it is clear that the RRRP function generator is, indeed, the most well suited of the planar four-bar architectures to generate Eq. (30). Now, this answer may seem relatively obvious, as the equation that was being generated is identical to the equation presented in Eq. (18), but that contrivance was by design so as to ensure that the results of the algorithm could be verified; it was expected that the RRRP function generator would outperform its planar compatriots given this fact.

5. CONCLUSIONS

While a large family of planar four-bar linkage problems exist, most are typically solved by discrete exact or approximate methods. Function generation has already been solved with a continuous approximation method [6], driving the cardinality of the originating data set towards infinity, thereby making the design and structural errors of the function generator to converge, effectively converting a non-linear optimisation problem to one that is linear. However the integration of the trigonometric Freudenstein equation for IO generation is quite challenging, and whatever gains in computational efficiency associated with no longer having to minimise the structural error problem are easily lost due to the computational cost of the integration.

Given the algebraic form of the IO function, however, this integral approach is simplified significantly, requiring the user to only minimise one single equation with a given set of initial assumptions that may be readily computed from the algebraic IO function. Integrating the square of the algebraic IO function of any planar four-bar linkage and subsequently minimising this function provides linkage parameters which are nearly identical to those developed through the solution to the discrete approximate structural error problem. Not only are these linkage parameters nearly identical to those generated through a classical non-linear minimisation approach to determine the structural error of a given linkage, the set of optimal parameters significantly reduced the error when compared with the precision point method used to generate the solutions themselves. Given the set of linkage parameters that are generated from this minimisation, the algebraic IO function may be solved explicitly to define the function that is generated by the linkage, facilitating its comparison to the desired function through use of a simple integral and percentage error calculation. This statement may be further extended to comparing every single planar function generator to each other by way of these same comparative metrics in order to facilitate concurrent type and dimensional synthesis for any planar function generator problem.

In the future, the methodology presented herein will be extended to any four-bar function generating linkage. Provided algebraic IO functions for any function generator, including spherical or spatial linkages, the continuous approximate design error minimisation problem should be able to explicitly compute the total error minimising linkage parameters without the need to compute the structural error of these linkages, thus eliminating the need to solve the overtly complex non-linear structural error problem.

REFERENCES

1. Angeles, J., Alivizatoss, A. and Akhras, R. "An unconstrained nonlinear least-square method of optimization of RRRR planar path generators." *Mechanism and Machine Theory*, Vol. 23, No. 5, pp. 343–353, 1988.
2. Gosselin, C. and Angeles, J. "Optimization of planar and spherical function generators as minimum-defect linkages." *Mechanism and Machine Theory*, Vol. 24, No. 4, pp. 293–307, 1989.
3. Gosselin, C.M., Moore, B. and Schicho, J. "Dynamic balancing of planar mechanisms using toric geometry." *Journal of Symbolic Computation*, Vol. 44, No. 9, pp. 1346–1358, 2009.
4. Tinubu, S. and Gupta, K. "Optimal synthesis of function generators without the branch defect." *Journal of Mechanisms, Transmissions, and Automation in Design*, Vol. 106, No. 3, pp. 348–354, 1984.
5. Hayes, M., Parsa, K. and Angeles, J. "The effect of data-set cardinality on the design and structural errors of four-bar function-generators." In "Proceedings of the Tenth World Congress on the Theory of Machines and Mechanisms, Oulu, Finland," pp. 437–442, 1999.
6. Guigue, A. and Hayes, M.J.D. "Continuous approximate synthesis of planar function-generators minimising the design error." *Mechanism and Machine Theory*, Vol. 101, pp. 158–167, 2016.
7. Freudenstein, F. "Approximate synthesis of four-bar linkages." *Resonance*, Vol. 15, No. 8, pp. 740–767, 2010.
8. Hayes, M.J.D., Husty, M.L. and Pfurner, M. "Input-output equation for planar four-bar linkages." In "International Symposium on Advances in Robot Kinematics," pp. 12–19. Springer, 2018.
9. Rotzoll, M., Hayes, M.J.D. and Husty, M.L. "An algebraic input–output equation for planar RRRP and PRRP linkages." *Transactions of the Canadian Society for Mechanical Engineering*, Vol. 44, No. 4, pp. 520–529, 2020.