Continuous Approximate Kinematic Synthesis of Planar, Spherical, and Spatial Four-bar Function Generating Mechanisms

by

Zachary A. COPELAND

A thesis submitted to
the Faulty of Graduate and Postdoctoral Affairs
in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy
in
Mechanical Engineering

Ottawa-Carleton Institute for Mechanical and Aerospace Engineering
Department of Mechanical and Aerospace Engineering
Carleton University
Ottawa, Ontario, Canada
April 26, 2024

Abstract

The focus of this work is on synthesising kinematic input-output (IO) function generators, specifically in the context of single degree of freedom four-bar linkages. The synthesis process typically involves minimising two key metrics: design error and structural error. *Design error* minimisation aims to reduce the arithmetic residual of the synthesis equation for the four-bar linkage, while *structural error* minimisation focuses on minimising the difference between prescribed and generated output parameters. The latter is crucial for real-world performance evaluation as it directly impacts the physical performance of the linkage, albeit requiring computationally intensive non-linear optimisation algorithms.

The objective of the research is to integrate the algebraic input-output equation across the input angle range to avoid explicit solution of the non-linear structural error optimisation problem. This approach, termed continuous approximate algebraic input-output synthesis, aims to expand the dataset used for kinematic synthesis to infinity by minimising the residual of the dot product of two arrays containing linkage geometry and desired input-output function information. Continuous approximate algebraic input-output synthesis effectively and simultaneously achieves both design and structural error minimisation by extending the dataset cardinality to infinitely many points, while still functioning as a linear least squares optimisation.

Comparisons with traditional non-linear structural error synthesis techniques demonstrates that continuous approximate algebraic input-output synthesis generates structural error minimised linkages with reduced errors between desired and synthesised functions. In addition to its computational efficiency, continuous approximate algebraic input-output synthesis is capable of producing improved results compared with classical problems within function generator synthesis, while also enabling combined type and dimensional synthesis, eliminating the need to define the linkage type beforehand.

This thesis also explores the concepts and limitations of multi-modal continuous approximate algebraic input-output synthesis, which allows designers to synthesise linkages that not only fulfill the desired input-output relationships but also maintain a specific secondary relationship within intermediate joint variables' motion.

Acknowledgements

I would like to acknowledge the research of Dr. Mirja Rotzoll and Dr. M. John D. Hayes, without which this thesis would not have been possible. Furthermore, I would like to thank Dr. Hayes for his continued patience in my production of the work contained herein.

Contents

Abstract iii

Acknowledgements v

List of Figures xi

List of Tables xv

List of Abbreviations xvii

List of Symbols xix

1 Introduction: Literature Review and Mathematical Background 1

1.1 Machines and Mechanisms 1

1.2 Planar Four-Bar Linkages 2

1.2.1 Thesis Problem Statement 2

1.2.2 Other Problems within Four-Bar Linkage Synthesis 4

1.3 Planar Four-Bar Function Generators 5

1.4 Literature Review 7

1.4.1 The Freudenstein Equation 9

1.4.2 The Design Error 10

Matrix Condition Number Minimisation 11

Singular Value Decomposition 13

1.4.3 Structural Error 14

1.5 Non-Planar Four-Bar Mechanisms 18

1.5.1 Spherical RRRR Linkage Trigonometric IO Equation 19

1.6 Extension of Current Theory . . . 21
1.6.1 Limitations of the Freudenstein Equation . . . 22
1.6.2 Integration of the Freudenstein Equation . . . 23
1.6.3 The Denavit-Hartenberg Convention and the Planar RRRR Input Output Equation . . . 26
Coordinate System Considerations . . . 29
1.6.4 Algebraic Input-Output Equations of the Planar RRRR Mechanism 30
1.6.5 Algebraic Input-Output Equations of the Planar RRRP Mechanism 33
Simplifications to the RRRP Input-Output Formulation . . . 33
1.6.6 Algebraic Input-Output Equations of the Planar PRRP Mechanism 34
DH Method Planar PRRP Input-Output Equations . . . 35
1.6.7 Algebraic Input-Output Equation of the Spherical RRRR Linkage . 36
1.6.8 Algebraic IO Equation of the Spatial RSSR Mechanism . . . 38
1.7 Problem Statement and Thesis Structure . . . 40
1.8 Statement of Originality . . . 42

2 Continuous Approximate Algebraic Function Generator Synthesis 45
2.1 Continuous Approximate Synthesis via the Algebraic Input-Output Equations . . . 45
2.2 Theory and Mathematical Modelling Concepts . . . 46
2.2.1 Mathematical Modelling Software . . . 48
2.3 Continuous Approximate Synthesis of Planar Four-Bar Function Generators . . . 48
2.3.1 Continuous Approximate Synthesis for the RRRR Function Generator . . . 48
2.3.2 Continuous Approximate Synthesis for the RRRP Linkage . . . 53
2.3.3 Continuous Approximate Synthesis for the PRRP Function Generator . . . 55
2.4 Non-Planar Function Generating Linkage Architectures . . . 58
2.4.1 Spherical RRRR CAAIOS Function Generator Synthesis . . . 58

2.4.2 Spatial RSSR Function Generating Linkage 62
2.5 Computational Considerations 67
2.5.1 Optimisation Formulation 67
2.5.2 Numerical Sensitivity, Floating Point Values, and the Initial Guess 69

3 Extensions to Continuous Approximate Algebraic IO Synthesis 71
3.1 Combined Type and Dimensional Synthesis for Planar Four-Bar Function Generators 71
3.2 Multi-Modal Continuous Approximate Synthesis 74
3.2.1 Mathematical Implementation of Multi-Modal Continuous Approximate Synthesis 75
3.2.2 Planar RRRR Multi-Modal Function Generation 76
Naïve Multi-Modal Synthesis Attempt 80
3.2.3 Constrained Multi-Modal Synthesis Example 83
Weighted Multi-Modal RRRR Function Generator Synthesis 85
3.2.4 Functionally Constrained Multi-Modal Function Generator Synthesis 87
3.2.5 Multi-Modal Synthesis in Practice 91
3.2.6 RRRP Multi-Modal Function Generator Synthesis 97
3.3 Discussion 101

4 Conclusions 103
4.1 Recommendations for Future Work 106

Bibliography 109

List of Figures

1.1 A general planar 4R function generator. 5
1.2 A general planar RRRP function generator. 6
1.3 A general planar PRRP function generator. 7
1.4 Kinematic geometry of the Denavit-Hartenburg parameters associated with the spherical RRRR function generating linkage architecture. 19
1.5 Demonstration figure for the labelling of the vectors required to define the spherical RRRR linkage vector loop closure equation [79]. 20
1.6 Enumeration of the DH coordinate systems and assignment rules. 27
1.7 DH parameters in a general serial 3R kinematic chain. 27
1.8 The stellated octahedron which occupies the centre of the design parameter space of a planar RRRR function generator. 32
1.9 The eight degenerate bi-cubic surfaces at the centre of the spherical RRRR design parameter space, with all 12 real unique lines highlighted to form the vertices of a stelated octahedron. 37
1.10 The kinematic geometry of the RSSR linkage, will relevant joint lengths and angles labelled for DH method parametrisation. 38

2.1 Comparison of RRRR desired and generated functions. 53
2.2 Comparison of RRRP desired and generated functions. 55
2.3 Comparison of PRRP desired and generated functions. 57
2.4 The results following the three point precision point solution for the spherical RRRR linkage architecture. 60
2.5 The results following the CAAIOS algorithm implementation for the spherical RRRR linkage architecture. 61

2.6 The kinematic geometry of the RSSR linkage, will relevant joint lengths and angles labelled for DH method parametrisation. 62
2.7 The results following the three point precision point solution for the spatial RSSR linkage architecture. 65
2.8 The results following the CAAIOS algorithm implementation for the spatial RSSR linkage architecture. 66

3.1 Comparison of 4R, RRRP, and PRRP desired and generated functions. . . 73
3.2 A general planar 4R function generator. 74
3.3 The prescribed, exact, and continuous synthesis approximation of Equation 3.3 in the v_1-v_4 plane. 80
3.4 The prescribed, continuous synthesis approximate, and the five functions generated by the identified link lengths in the v_i-v_j planes. 81
3.5 The desired competing v_1-v_3 function and the one generated by the linkage that approximates Equation 3.3. 81
3.6 The v_1-v_3 multi-modal results. 82
3.7 The v_1-v_4 multi-modal results. 83
3.8 The desired input-output relationship in $v_1 - v_3$ compared to the same relationship which is generated by the CAAIOS optimal linkage parameters. 84
3.9 The $v_1 - v_4$ equation resulting from the weighted multi-modal synthesis algorithm. 86
3.10 The $v_1 - v_4$ equation resulting from the weighted multi-modal synthesis algorithm. 87
3.11 The desired multi-modal function in $v_1 - v_3$ and the relationship resulting from the CAAIOS of the linkage with a defined IO relationship in $v_1 - v_4$. 89
3.12 The desired function in $v_1 - v_4$ and the relationship resulting from the CAAIOS in $v_1 - v_4$ coupled with the desired $v_1 - v_3$ relationship. 90
3.13 The desired multi-modal function in $v_1 - v_3$ and the relationship resulting from the CAAIOS in $v_1 - v_4$. 90

3.14 The polynomial interpolant, Equation (3.17), and the v_1-v_3 function, Equation (3.15), generated by the linkage that approximates Equation (3.3). . . 94
3.15 The v_1-v_3 multi-modal results. 96
3.16 The v_1-v_4 multi-modal results. 96
3.17 The v_1-v_3 RRRP multi-modal results. 100
3.18 The v_1-d_4 RRRP multi-modal results. 100

List of Tables

3.1 Variable substitutions used for continuous approximate type and dimensional planar function generator synthesis. 72
3.2 All identified parameters for the continuous approximate concurrent type and dimensional synthesis of a planar function generator. 72
3.3 Percentage error for all viable planar four-bar function generators. 73
3.4 Continuous approximate synthesis results generating Equation (3.3). . . . 79
3.5 Structural error generating Equation (3.3). 79
3.6 Required and $v_4 = f_1(v_1)$ generated values of θ_3 at required θ_1. 93
3.7 The $v_4 = f_1(v_1)$ and $v_3 = f_2(v_1)$ planar 4R multi-modal synthesis results. 95
3.8 The $v_4 = f_1(v_1)$ and $v_3 = f_2(v_1)$ planar 4R multi-modal synthesis structural errors. 95
3.9 Required and multi-modal generated values of θ_3 at required θ_1. 95
3.10 The $d_4 = f_1(v_1)$ and $v_3 = f_2(v_1)$ planar RRRP multi-modal synthesis results. 99
3.11 The $d_4 = f_1(v_1)$ and $v_3 = f_2(v_1)$ planar RRRP multi-modal synthesis structural errors. 99

List of Abbreviations

IO	**I**nput-**O**utput.
AIO	**A**lgebraic **I**nput-**O**utput,
CAAIOS	**C**ontinuous **A**pproximate **A**lgebraic **I**nput-**O**utput **S**ynthesis,
MMCAAIOS	**M**ulti-**M**odal **C**ontinuous **A**pproximate **A**lgebraic **I**nput-**O**utput **S**ynthesis.

List of Symbols

θ_i	i^{th} Joint Angle	rad
k_i	i^{th} Freudenstein Parameter	
a_i	i^{th} Link Length	
α	Input Link Dial Zero	rad
β	Output Link Dial Zero	rad
σ_i	i^{th} Singular Value	
ϵ	i^{th}Machine Precision	
v_i	i^{th} Tangent Half Angle	
d_i	i^{th} Slide Travel Distance	
τ_i	i^{th} Link Twist	rad
α_i	i^{th} Link Arclength	rad
$\mathbf{s_A}$	Synthesis Array	
$\mathbf{p_A}$	Parameter Array	

To my parents...
You encouraged me to start down this path, and I only wish that you were here to see me complete it.

1 Introduction: Literature Review and Mathematical Background

1.1 Machines and Mechanisms

Human beings have been using machines and mechanisms for millennia in order to assist in every variety of physical operation imaginable [1, 2]. The theory of machines and mechanisms, as a formal science, however, has its roots in more modern science as the result of the work of Franz Reuleaux [3] in the late 1800s. While Reuleaux may be remembered in modern times for his contributions to the science of curves of constant width, specifically the Reuleaux Triangle, his work on the kinematics of machinery laid the foundation for the formal analysis of machines and mechanisms. Specifically, he developed the abstraction of machines such that they are composed of pairs of components whose connections impose motion constraints, which when coupled with additional components result in kinematic chains which may be analysed depending on their constituent components. This definition was the first formal definition of a machine which allowed for their analysis to take place at a higher level than the realm of simple mechanics and instead focus on motion constraints, which developed the science of kinematics dramatically.

Machines and mechanisms take many forms and may involve any number of components, have any number of degrees of freedom, and they may be planar or spatial in nature. Typically, most machines and mechanisms will be comprised of pairs of components including, but not necessarily limited to: gears; pulleys; cables; links; cams; rotating shafts; and sliders [4]. While the modern science of machines and mechanisms uses

increasingly complex systems and components in order to accommodate the demands of increasingly complex machine design, some fundamental problems with regards to the design of simple mechanisms do remain as open problems in the science.

1.2 Planar Four-Bar Linkages

1.2.1 Thesis Problem Statement

Four-bar linkages are characterised as single degree of freedom simple closed kinematic chains [5]. These linkages can be planar, spherical, or spatial in design. For the purposes of the remaining discussion, attention will be limited to the state of the art of planar four-bar linkage design problems. However, for some special cases such as those associated with spherical four-bar linkages, identical processes may be used to elucidate the same information as its planar counterpart, because the plane may be considered a sphere of infinite radius.

The development of mathematical models for the design of the function generating linkages relies on the use of vector-closure equations [6], and was pioneered in 1954 by Ferdinand Freudenstein in his Ph.D thesis [7]. Freudenstein's work developed the methods that are now used for the design of four-bar function generating mechanisms; mechanisms which follow some prescribed input-output (IO) relationship. Given the time of the publication of this work, however, Freudenstein was ultimately in a position where his work was fundamentally limited by the computational capacities of technology in the 1950s. Specifically, in a discussion posted in response to Freudenstein's 1959 structural error minimisation paper, he stated that the computational cost of a single linkage design was one hour of computation on an IBM 650, with a cost of approximately $120 USD [8]. Adjusting strictly for inflation, the cost of design associated with this single mechanism would have been approximately $1261 USD in 2023. Clearly, given this prohibitive cost, a mechanism designer in the 1950s would have had to rely on designs that had already been developed and validated for their use. The validated design was contained within a so-called atlas of acceptable designs, such as "Analysis of

the four-bar linkage: its application to the synthesis of mechanisms" [9], which was initially published in 1951; this atlas of designs not only took into account the IO relationships of some linkages, but contained information about useful cases of coupler curves. As computational limits and capacities have expanded exponentially since the 1950s, this is no longer the case, and a mechanism may be readily designed by any sufficiently skilled designer on their desktop computer using software such as Maple, Matlab, or Geogebra. Due to these computational advancements, additional works such as [10] have developed an atlas of coupler point curves for the spherical RRRR function generator, and works such as those present within [11] make use of computational power to provide more readily available linkage designs.

It is towards the problem of input-output (IO) function generation that the work contained within this thesis can be readily applied. Kinematic synthesis of four-bar function generating mechanisms relies on the minimisation of certain performance indicators. Two of these metrics, referred to as the design error and structural error, are the focus of the work presented herein. The *design error* minimisation focuses on the minimisation of the arithmetic residual of the synthesis equation for the four-bar linkage at hand, while the *structural error* focuses on the minimisation of the difference between the *prescribed* and *generated* output parameters. When focused purely on the real-world performance of a linkage, one could argue that the *structural error* is the performance indicator of primary concern due to the fact that it has an obvious correlation to the physical performance of a linkage, although the minimisation of the *structural error* metric requires a non-linear minimisation algorithm which can be quite computationally intensive.

The objective of this work is the integration of the algebraic input-output (AIO) equation over the desired range of input angles to eliminate the need for the explicit solution of the non-linear structural error optimisation problem in order to implicitly drive the cardinality of the data set being used for the kinematic synthesis of the linkage to infinity. It will be demonstrated that the so-called continuous approximate algebraic input-output synthesis (CAAIOS) solution is accomplished solely through the minimisation of the residual of a dot product of two arrays containing information regarding

the geometry of the linkage and the desired input-output function.

1.2.2 Other Problems within Four-Bar Linkage Synthesis

Outside of IO function generation, kinematics problems in four-bar linkages tend to focus on the motion of the coupler link, labelled as link a_2 in Figure 1.1. This link is of particular interest due to its ability to guide any point attached to the link through general curvilinear planar motion [5]. This is to say that, if a point is fixed to the coupler link of a four-bar mechanism, that point will generate some path in the plane which depends on the geometry of the linkage at hand, as well as the assembly mode of the linkage [12, 13, 14]. Cognate linkages may also be useful when considering the available space allotted to a given linkage design for a specific path generation problem; according to the Roberts-Chebyshev Theorem [15], there exists three planar four-bar linkages which generate an identical coupler point curve [16], which have different base-fixed R-pair coordinates and link lengths, while maintaining the coordinates of a point in the coupler-fixed coordinate system. Thus, cognate linkages may allow a designer to maintain their originally intended coupler point curve in situations where the original orientation of the linkage may not be usable. More recently, in works such as [17, 18, 19, 20, 21, 22], additional effort has been placed on developing solutions to the path generation problem, using various types of numerical optimisation techniques alongside algebraic geometry. Notably, in [16], the authors develop a fully analytical method for the coupler point curve synthesis problem, alongside simultaneous identification of all cognate linkages, though it is based on constrained coupler point curves with no allowance for deviation as in [14].

Coupler point curves may be further extended to trajectory generation [12], the problem of developing some coupler point curve which not only controls the location of this point, but also the time difference between successive points. Coupler motion may yet be further extended to contain information relevant to rigid body guidance; the Burmester problem [23, 24, 25, 26, 27] is concerned with guiding the coupler through a specific set of positions with coupled orientations. While many analytical and graphical

solutions exist for the Burmester problem, the solutions rely on approximations based on a finite set of poses. While [28] laid the foundation for work towards the continuous approximate Burmester problem, the methods outlined rely heavily on developing a novel technique for defining n^{th} order surfaces in space [29], and as such the solution to the continuous approximate Burmester problem remains an open topic.

1.3 Planar Four-Bar Function Generators

The generalised function generation problem consists of identifying a mechanism which is able to approximate, in some sense, an IO function between a given pair of joint variables over some specified range of input variable for a given planar linkage architecture comprised of RR-, RP-, PR-, or PP-dyads[1]. Figure 1.1 illustrates a function generating four-bar RRRR linkage. If link a_1 is the input link and link a_3 is the output link, the IO function is specified as $\theta_4 = f(\theta_1)$.

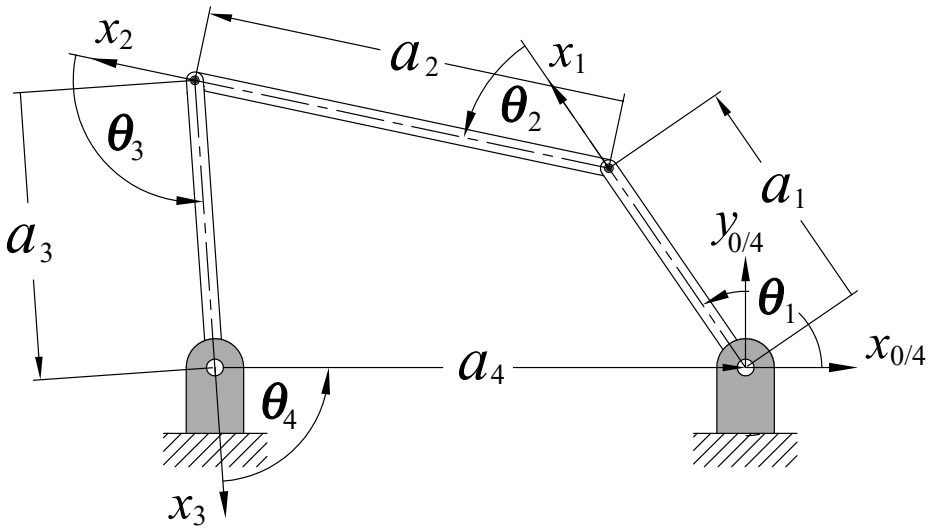


FIGURE 1.1: A general planar 4R function generator.

Figure 1.2 illustrates a function generating four-bar RRRP linkage, where the rightmost dyad of the RRRR linkage is replaced with an RP-dyad. If link d_1 remains the input link, and the output parameter is now d_4, the IO function is specified as $d_4 = f(\theta_1)$.

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

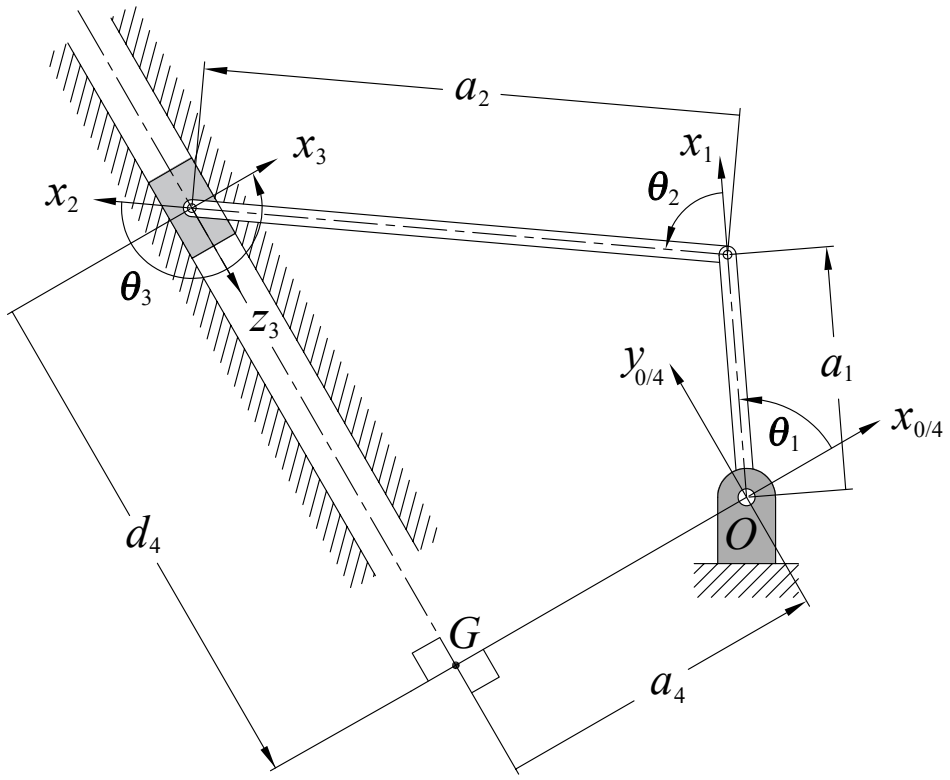


FIGURE 1.2: A general planar RRRP function generator.

Figure 1.3 illustrates a function generating four-bar PRRP linkage, where the leftmost dyad of the linkage is a PR-dyad, and the rightmost dyad of the linkage is an RP-dyad. In this case, the input parameter for the function generator is a_1, while the output parameter is d_4, meaning that the IO function is specified as $d_4 = f(d_1)$.

The function generation problem is often focused on either the *design*, or *structural error* minimisation, and is a dimensional synthesis problem - concerned strictly with developing the dimensions of the function generating linkage which minimises the vector norms of either of these performance indicators. 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 [30, 31]. The solutions to these problems were first published by Freudenstein in [32] and [30], respectively.

The design error minimisation leads to a linear least squares problem, while the structural error is a highly non-linear problem which requires an iterative optimisation

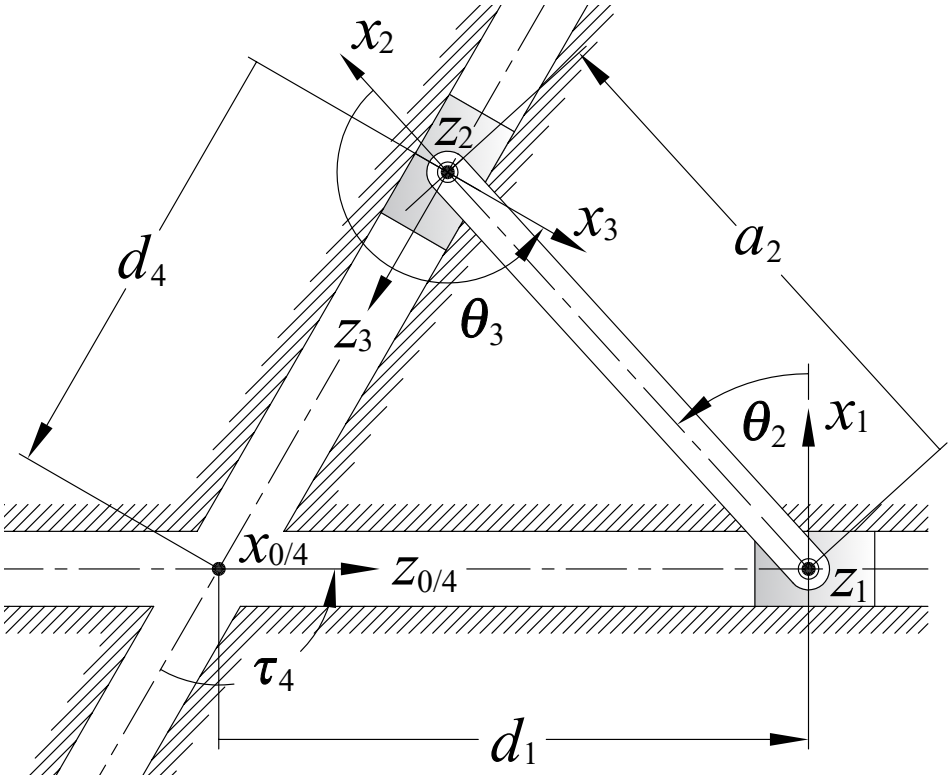


FIGURE 1.3: A general planar PRRP function generator.

approach in order to compare the generated function to the prescribed function [33, 30, 7]. From a design perspective, the structural error is viewed as the most important performance metric, since it concerns itself with the physical performance of the linkage, whereas the design error is solely concerned with the arithmetic performance of the linkage.

1.4 Literature Review

Considering that the problem of function generation was computationally pioneered in [7], first published in 1954, a number of novel modern techniques have been developed for planar four-bar function generator synthesis. These methods include, but are not limited to: multifactor optimisation of the function generator [34], function generator synthesis with consideration of the transmission angle from coupler to output link by various metrics [35, 36, 37, 38, 39], and methods which constrain the relative link lengths through mobility constraints [40]. Some recent works have also placed effort

towards incorporating machining tolerances into the kinematic synthesis of function generators [41] in order to incorporate more physically relevant metrics for the kinematic synthesis of linkages. Velocity and acceleration level dimensional synthesis of mechanisms has also been realised as an additional physical metric of relevance, and work such as that presented in [42] perform dimensional synthesis directly from these relationships.

Within four-bar function generation, order and branch defects indicate defects within the kinematic synthesis of mechanisms which occur when the generated linkage inverts the order of the points within a given function or path, or renders some parts of these functions or paths inaccessible within a given assembly mode of a linkage. Algorithms such as [43, 44] endeavour to eliminate these defects from the generation of linkages. Much like problems contained within the coupler pose curve world, function cognate linkages, or linkages which bear little physical resemblance to one another yet develop the same IO relationship [45] have also been investigated for their use in function generator and machine design.

In works such as [46, 47, 48, 49, 50, 51, 52, 53] novel computational algorithms or formulations of the dimensional synthesis problem are depicted, however, none of these methods depart, in any significant way, from the original design and structural error metrics pioneered by Freudenstein in his 1954 thesis in order to obtain an optimal linkage design. Typically, these problems all still rely on the same linear least-squares solutions of decades past [54], though some additional recent works have endeavoured towards improving the fidelity of these least-squares methods through optimal selection of precision points [55], or through the use of more advanced mathematical techniques such as the use of Fourier Coefficients [56]. Notably, efforts presented in [33] and [57] have lent credence to the idea that the structural error minimisation problem may be implicitly solved by driving the cardinality of the originating data set to infinity.

1.4.1 The Freudenstein Equation

The design of IO function generating linkages, specifically the case of RRRR planar function generating linkages, relies on the use of the synthesis equation which was developed by Ferdinand Freudenstein over the course of his Ph.D research in 1954 [7]. Specifically, the i^{th} configuration of a planar four-bar RRRR function generator is governed by,

$$k_1 + k_2 \cos(\theta_{1i}) - k_3 \cos(\theta_{4i}) = \cos(\theta_{1i} - \theta_{4i}), \tag{1.1}$$

where the k_i are the *Freudenstein Parameters* which denote ratios of the link lengths, illustrated in Figure 1.1, such that,

$$k_1 \equiv \frac{(a_1^2 - a_2^2 + a_3^2 + a_4^2)}{2a_1a_3}, \tag{1.2}$$

$$k_2 \equiv \frac{a_4}{a_1}, \tag{1.3}$$

$$k_3 \equiv \frac{a_4}{a_3}. \tag{1.4}$$

While the link lengths may be expressed as,

$$a_4 = 1, \tag{1.5}$$

$$a_1 = \frac{1}{k_2}, \tag{1.6}$$

$$a_3 = \frac{1}{k_3}, \tag{1.7}$$

$$a_2 = \sqrt{(a_4^2 + a_1^2 + a_3^2 - 2a_1a_3k_1)}, \tag{1.8}$$

where the ground-fixed link length, a_4 has its length set to unity, as it serves only to scale the overall size of the linkage, and the same function is generated regardless of the scale of the function generator. It is relatively simple to show that for $i = 3$ IO angle pairs, referred to in this case as *precision points*, the Freudenstein Equation may

be solved explicitly for the parameters which precisely generates the function, but only at the three precision points. For practical applications, it is often desired to have a set of n points such that $n > 3$ [58, 59]. However it is often desired that $n \gg 3$ in order to more precisely control the shape of the generated IO curve; solutions to such a system of overconstrained equations require numerical approximation, typically, the Moore-Penrose Generalised Inverse is used [60]. To reduce error in approximating the function over the prescribed range, the design and structural errors are the typical objective functions [7, 47, 48, 53].

1.4.2 The Design Error

The most intuitive approach to use for this approximation is the *design error*, which is functionally the minimisation of the residual of the Freudenstein Equation in an over-constrained system of n equations for $n \gg 3$. The first step is to develop a matrix-array representation of this problem which splits the right hand side, Freudenstein Parameters, and the trigonometric functions of the IO pair from Equation 1.1 into two arrays and a single matrix. The resulting equation has the form,

$$\mathbf{Sk} = \mathbf{b}. \tag{1.9}$$

The matrix $\mathbf{S}$, referred to as the synthesis matrix, takes the form of the $n \times 3$ matrix,

$$\mathbf{S} = \begin{bmatrix} 1 & \cos(\theta_{4_1}) & -\cos(\theta_{1_1}) \\ 1 & \cos(\theta_{4_2}) & -\cos(\theta_{1_2}) \\ \vdots & \vdots & \vdots \\ 1 & \cos(\theta_{4_n}) & -\cos(\theta_{1_n}) \end{bmatrix}, \tag{1.10}$$

while $\mathbf{b}$ is the $n \times 1$ array,

$$\mathbf{b} = \begin{bmatrix} \cos(\theta_{1_1} - \theta_{4_1}) \\ \cos(\theta_{1_2} - \theta_{4_2}) \\ \vdots \\ \cos(\theta_{1_n} - \theta_{4_n}) \end{bmatrix}, \tag{1.11}$$

and $\mathbf{k}$ is an array containing the three Freudenstein Parameters to be solved for by this approximation. When $n = 3$, the solution to this set of equations simply becomes,

$$\mathbf{k} = \mathbf{S}^{-1}\mathbf{b}, \tag{1.12}$$

but given that $n \gg 3$, we know that this can not be the case, and the design error equation actually becomes

$$\mathbf{d} = \mathbf{b} - \mathbf{S}\mathbf{k}, \tag{1.13}$$

where $\mathbf{d}$ is the design error array, or residuals corresponding to the m^{th} element in $\mathbf{S}$ and $\mathbf{b}$. Accordingly, an objective function must be derived in order for this optimisation to take place. In the case of the design error, the objective function for minimisation is,

$$z = \frac{1}{2}(\mathbf{d}^T\mathbf{W}\mathbf{d}), \tag{1.14}$$

where $\mathbf{W}$ is a diagonal matrix of positive valued weighting factors. Typically, $\mathbf{W}$ is used to define which points are more or less important to the minimisation, in order to target certain ranges or portions of the desired IO relationship more accurately than others. However, in practice, it is typical that $\mathbf{W} = \mathbf{I}$ is used as a starting point for the subsequent analysis, where $\mathbf{I}$ is the identity matrix.

Matrix Condition Number Minimisation

Now, in an ideal situation, one would employ the Moore-Penrose generalised inverse of the synthesis matrix, $\mathbf{S}^{-1}$ in order to compute the optimal $\mathbf{k}$ array. Conceptually, this is the simplest and most effective way to proceed. However, typically, matrix $\mathbf{S}$ is

either poorly conditioned, or sufficiently rank deficient (either in the analytical sense, or with singular values that are, numerically, so close to zero that they are dominated by roundoff error) so as to render the utility of this approach problematic at best. The invertibility of a matrix may be evaluated by examining the condition number of the matrix, κ, which is the ratio of the largest to the smallest singular value within the matrix. Thus, methods of reducing the condition number of the matrix are required. In order to minimise the condition number of the matrix, specific constants, typically referred to as dial zeroes, may be added to the equations in order to facilitate this [61]. Furthermore, minimising the condition number of a characteristic matrix within a system of linear equations tends towards improving the convergence of the system during an iterative solution procedure [62], where the characteristic matrix $\mathbf{C}$, of some matrix $\mathbf{A}$ is defined such that, $\mathbf{C} = \mathbf{A} - \lambda\mathbf{I}$.

The inclusion of dial zeroes means that Equation (1.1) may be altered by including values modifying θ_1 and θ_4. Typically, these values are defined as α and β, such that,

$$\theta_{1_i} = \alpha + \Delta\theta_{1_i}, \tag{1.15}$$

$$\theta_{4_i} = \beta + \Delta\theta_{4_i}, \tag{1.16}$$

which, upon substitution into 1.1 yields,

$$k_1 + k_2\cos(\alpha + \Delta\theta_{1_i}) - k_3\cos(\beta + \Delta\theta_{4_i}) = \cos(\alpha + \Delta\theta_{1_i} - (\beta + \Delta\theta_{4_i})). \tag{1.17}$$

This substitution allows the designer to modify matrix $\mathbf{S}$ (Equation (1.10)) to the form of,

$$\mathbf{S} = \begin{bmatrix} 1 & \cos(\beta + \Delta\theta_{4_1}) & -\cos(\alpha + \Delta\theta_{1_1}) \\ 1 & \cos(\beta + \Delta\theta_{4_2}) & -\cos(\alpha + \Delta\theta_{1_2}) \\ \vdots & \vdots & \vdots \\ 1 & \cos(\beta + \Delta\theta_{4_n}) & -\cos(\alpha + \Delta\theta_{1_n}) \end{bmatrix}, \tag{1.18}$$

where a method such as the Nelder-Mead downhill simplex algorithm may be used in order to minimise the condition number of matrix **S** through the tuning of α and β. Considering that this is an iterative procedure, some initial guess must be provided in order to develop dial zeroes which converge to static values; however, there exists little guidance as to the selection of the initial set of dial zeroes for the Nelder-Mead minimisation, and this decision relies on the experience of the designer. Once this operation has been completed, the remaining portion of the design error synthesis may be completed as per the methods previously outlined.

Singular Value Decomposition

Singular Value Decomposition (SVD) is one method which is useful for inverting over-constrained systems of linear equations that may contain singular, or numerically close to singular values [63, 64]. SVD not only allows a designer to invert a non-square and poorly-conditioned matrix, **S**, but it also allows the designer to observe which values of **b** are driving the solution away from a minimum condition number, thus allowing for the elimination of those elements from the equation in order to increase the fidelity of the least-squares solution-based results. In cases where the classical techniques of Gaussian elimination [65, 66], Householder reflections [67, 68], or the Moore-Penrose Generalised Inverse [69, 70, 71] fail to yield acceptable or useful results for the inversion of **S** in order to develop the design error minimising values of k_i, one may use SVD in order to decompose the matrix **S** into the following three matrices:

$$\mathbf{S} = \mathbf{U}_{m\times n}\Sigma_{n\times n}\mathbf{V}^T_{n\times n}, \tag{1.19}$$

where $\mathbf{U}_{m\times n}$ is a column-orthonormal matrix, $\Sigma_{n\times n}$ is a diagonal matrix containing the singular values of matrix **S**, and $\mathbf{V}^T_{n\times n}$ is a square orthogonal matrix whose columns and rows are orthonormal. The advantages of SVD begin with these matrices; given that $\mathbf{U}_{m\times n}$ and $\mathbf{V}^T_{n\times n}$ are both orthonormal matrices, their inverses are equal to their transpose, while matrix $\Sigma_{n\times n}$ may be inverted simply by taking the reciprocal of each diagonal element. This implies that, in general,

$$\mathbf{S}^{-1} = \mathbf{V}_{n\times n}\Sigma^{-1}_{n\times n}\mathbf{U}^{T}_{m\times n}, \tag{1.20}$$

which holds true unless any value in $\Sigma_{n\times n}$ is sufficiently close to zero, thus forcing its inverse to tend towards infinity. In such a case, one may simply remove that IO pair, which is close to zero, from the problem by allowing the corresponding entry in $\Sigma^{-1}_{n\times n}$ to be identically zero. Once this step has occurred, the optimal Freudenstein parameter set which minimises the 2-norm of Equation (1.12) is,

$$\mathbf{k}_{opt} = \mathbf{V}\Sigma^{-1}\mathbf{U}^{T}\mathbf{b}. \tag{1.21}$$

While the aforementioned solution is fundamentally complete, one aspect yet remains to be addressed; how close to zero is "sufficiently close to zero"? Clearly, it is desired to have some metric for the elimination of the problematic and poorly conditioned values in $\Sigma_{n\times n}$ which would allow a designer to make a consistent decision from one linkage design to the next. As is common in computational applications, the primary source for the metric which is used to zero these problematic values comes from the computational accuracy of the machine, ϵ, upon which the design is being developed. Accordingly, a magnitude of σ_i which constitutes a poorly conditioned entry may be taken as,

$$\frac{\sigma_i}{\sigma_{max}} < Rank(\mathbf{S})\epsilon. \tag{1.22}$$

Thus, SVD allows the designer to observe precisely which σ_i fail this simple inequality, and allows for their elimination from the set of points being used for the discrete approximate synthesis problem.

1.4.3 Structural Error

If the design error computes the linkage parameters which serve to minimise the residual of the Freudenstein equation over a given IO range, the structural error serves to compute the linkage which most effectively matches the desired and the generated IO

pairs for a given linkage. Mathematically, the structural error problem aims to minimise the value of the stuctural error array, **s**, where **s** is defined as,

$$\mathbf{s} \equiv \left[\theta_{4_i,Pres} - \theta_{4_i,Gen}\right], i\epsilon 1, 2.., n, \tag{1.23}$$

over n elements. Given the nature of this quantity, it stands to reason that the structural error problem requires a designer to compute a given set of errors, **s**, based upon some initial guess of Freudenstein's k_i values, at which point some change is levelled to the k_i parameters, defined as Δk_i. Subsequently, **s** is computed again and the problem returns to the top of the loop. Thus, this structural error minimisation procedure is a nonlinear, iterative, least-squares problem.

As with the design error procedure, an objective function, in this case ζ may be defined as,

$$\zeta = \frac{1}{2}\mathbf{s}^T\mathbf{W}\mathbf{s}, \tag{1.24}$$

where **W** is, again, a matrix of weighting factors. If **W** is defined as **I**, the objective function simplifies to,

$$\zeta = \frac{1}{2}\mathbf{s}^T\mathbf{s} = \frac{1}{2}\|\mathbf{s}\|^2. \tag{1.25}$$

This objective function will be minimised over the Freudenstein parameters, k_i, by the normality condition, or the gradient of ζ with respect to **k**,

$$\frac{\delta\zeta}{\delta\mathbf{k}} = \left(\frac{\delta\mathbf{s}}{\delta\mathbf{k}}\right)^T\frac{\delta\zeta}{\delta\mathbf{s}} = 0. \tag{1.26}$$

In order for the remaining operations to take place, a majority of the effort must be placed in developing a minimisation condition which will allow the objective function to be minimised. Typically, for an ideal function generator, the synthesis equation,

$$\mathbf{S}\mathbf{k} - \mathbf{b} = 0, \tag{1.27}$$

would be used. However, given the formulation of Equation (1.23), what is desired is a minimisation routine that is in terms of the *generated* output angle, alongside its static input angle partner, and the Freudenstein parameters. In essence, we require a form of the Freudenstein equation (Equation (1.1)) f, such that,

$$f(\theta_{4,Gen}, k_i, \theta_1) = 0. \tag{1.28}$$

For n input angles, θ_1, f will be an $n \times 1$ array of values. Now, for any given optimal set of k_i, f may be rewritten as the sum of its partial derivatives such that,

$$\frac{df}{d\mathbf{k}} = 0_{n\times 3} = \left(\frac{\delta f}{\delta \mathbf{k}}\right)_{n\times 3} + \left(\frac{\delta f}{\delta \theta_{4,Gen}}\right)_{n\times n} \left(\frac{\delta \theta_{4,Gen}}{\delta \mathbf{k}}\right)_{n\times 3}, \tag{1.29}$$

where the final partial derivative can be show to simplify to,

$$\frac{\delta \theta_{4,Gen}}{\delta \mathbf{k}} = -\left(\frac{\delta \mathbf{s}}{\delta \mathbf{k}}\right), \tag{1.30}$$

due to the definition of $\mathbf{s}$ from Equation (1.23); $\theta_{4,Pres}$ is invariant in k_i, as it is an ideal and prescribed value. The second partial derivative is an $n \times n$ diagonal matrix, $\mathbf{D}$, such that,

$$\mathbf{D}_{n\times n} = \frac{\delta f_n}{\delta \theta_{4,Gen}}. \tag{1.31}$$

Given that it is known that f is linear in k_i, and Equation (1.29) is equal to zero, it can be rearranged such that,

$$\frac{\delta \theta_{4,Gen}}{\delta \mathbf{k}} = -\mathbf{D}^{-1} \frac{\delta f}{\delta \mathbf{k}}, \tag{1.32}$$

and that,

$$\frac{\delta f}{\delta \mathbf{k}} \equiv \mathbf{S}. \tag{1.33}$$

Upon substitution of these identities into Equation (1.26) it can be shown that,

$$\frac{\delta \zeta}{\delta \mathbf{k}} = \mathbf{S}^T \mathbf{D}^{-1} \mathbf{s} = \mathbf{0}. \tag{1.34}$$

In order to minimise this condition, a $\Delta \mathbf{k}$ must be constructed from these values, alongside some initial guess for the structural error minimising linkage parameters, k_i^ν. However, given the sensitivity of this minimisation procedure, the designer must select an initial set of k_i^ν that is quite close to the structural error minimising k_i. In order to accomplish this in a fully general and universal manner, one must first optimise the linkage for the design error minimising k_i, which will then be used as the initial guess, k_i^ν, for the structural error minimisation routine.

In the ideal situation, we know that,

$$\theta_{4,Gen}(k_i) = \theta_{4,Pres}, \tag{1.35}$$

but, considering that this will never be the case in an overconstrained system of equations, we endeavour to compute,

$$\theta_{4,Gen}(k_i^\nu + \Delta k_i) = \theta_{4,Pres}, \tag{1.36}$$

such that the quantity modifying Δk_i, serving to perturb the estimated optimal k_i^ν, decreases in magnitude below the computational accuracy of the computer being used, and thus leaves the optimal k_i equal to k_i^ν at that iteration. In order to do this, the left hand side of the equation may be expanded in a Taylor series, such that,

$$\theta_{4,Gen}(k_i^\nu + \Delta k_i) = \theta_{4,Gen}(k_i^\nu) + \frac{\delta \theta_{4,Gen}}{\delta k_i}|_{k_i^\nu} \Delta \mathbf{k}, \tag{1.37}$$

and ignoring the higher-order terms. Using Equations (1.32) and (1.33), Equation (1.37) may be rewritten as,

$$\mathbf{D}^{-1} \mathbf{S} \Delta \mathbf{k} = \theta_{4,Gen}(k_i^\nu) - \theta_{4,Pres} = -\mathbf{s}^\nu, \tag{1.38}$$

therefore allowing for the computation of Δk as either the Moore-Penrose Generalised

Inverse of this equation at the ν^{th} step, or through the use of Householder transformations. This procedure exits when,

$$\|\Delta \mathbf{k}\| < \epsilon, \tag{1.39}$$

while,

$$\epsilon > 0, \tag{1.40}$$

where ϵ is the machine precision of the computer being used to produce the minimisation, and the k_i^ν at this step represents the structural error minimising linkage parameters for the function and four-bar linkage at hand.

1.5 Non-Planar Four-Bar Mechanisms

Planar four-bar function generating mechanisms tend to be the default mechanism construction for dimensional synthesis of function generating linkages, though spatial linkages are an obvious extension of this concept. Spherical linkages also exist, and can be shown to be a special case wherein the plane upon which the planar function generating linkage is constructed is a sphere with infinite radius [72]. Thus, spherical linkages can be shown to be the most general planar four-bar RRRR function generating linkage, where the equation for the non-spherical linkages may be extracted when the radius of the sphere upon which the linkage is constructed is set to infinity.

While planar four-bar function-generating mechanisms may be designed through the use of the Freudenstein equation and supporting metrics in the form of the design and structural errors, the design of non-planar four-bar function generating mechanisms is somewhat more complex. In the cases of the spatial function generators, there does not exist a general form of the IO function, and thus the methods rely on comparison to optimal solutions obtained via planar RRRR function generating mechanisms [73], the use of CAD packages [74, 75], or on the use of complicated numerical minmisation schema [76] in order to accomplish this task. While [77] produces results

which do synthesise planar, spherical, and spatial linkages for function-generator synthesis, the work relies on discontinuous spline curves and multiple simultaneous objective function minimisations in order to perform the synthesis. Furthermore, this method is only appropriate for dimensional synthesis up to ten precision points (as dictated by the dimensional synthesis points as well as the points required for the satisfaction of the additional objective functions), and as such is unsuitable for the extension to overconstrained systems of equations.

In the case of the spherical RRRR linkage, however, solutions based on vector-loop closure methods, much the same as the derivation of the original Freudenstein equation for planar RRRR linkages, do exist [5, 78].

1.5.1 Spherical RRRR Linkage Trigonometric IO Equation

The spherical four-bar function generator is constructed on the surface of a sphere of finite radius, such that all revolute joint axes intersect at the center of the sphere, as shown in Figure 1.4.

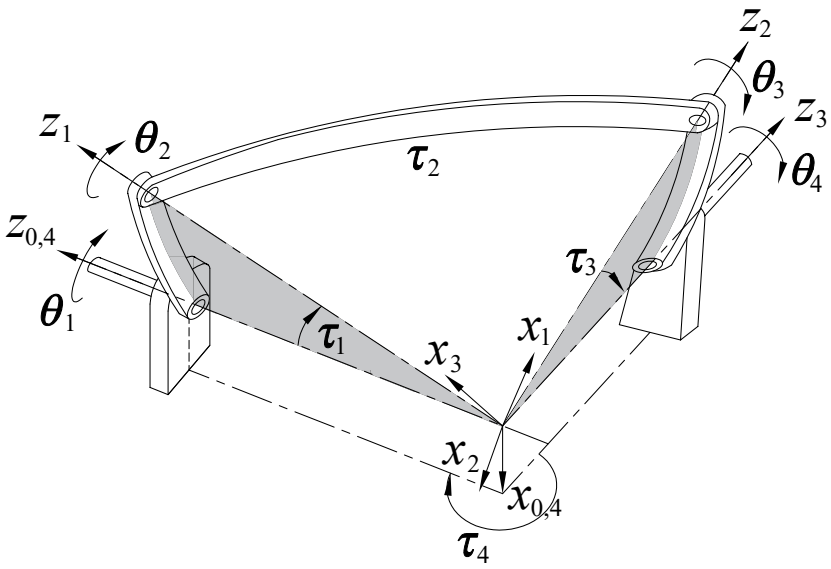


FIGURE 1.4: Kinematic geometry of the Denavit-Hartenburg parameters associated with the spherical RRRR function generating linkage architecture.

Using vector-loop closure equations which are similar to the methods of Freudenstein, the IO relationship of the spherical RRRR linkage may be determined based on the arclengths of the links which represent the distances between revolute joint centers of the mechanisms [5]. Is is assumed that the linkage derivation is being based on a quadrilateral projected onto the surface of a sphere such that the vertices are joined in order of $OABC$ and reconnecting to O, shown in Figure 1.5.

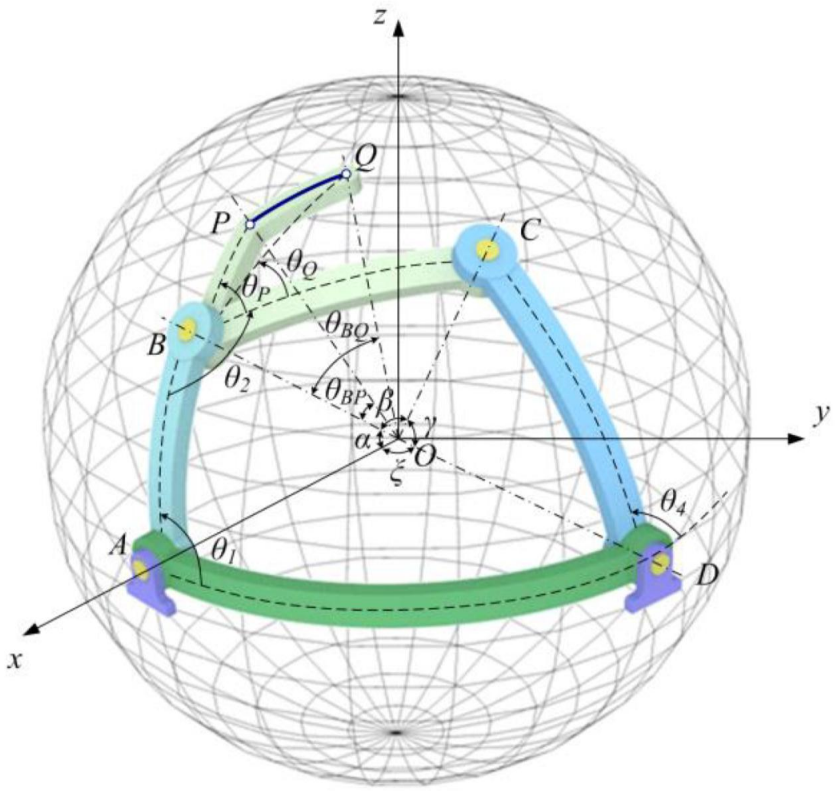


FIGURE 1.5: Demonstration figure for the labelling of the vectors required to define the spherical RRRR linkage vector loop closure equation [79].

The following arclengths, τ may then be defined, based on Figure 1.5, as the inverse cosines of the vectors connecting each revolute joint on the surface of the sphere upon which the mechanism is constructed. Explicitly, these arclengths are defined as,

$$\tau_1 = cos^{-1}(OA), \tag{1.41}$$
$$\tau_2 = cos^{-1}(AB), \tag{1.42}$$
$$\tau_3 = cos^{-1}(BC), \tag{1.43}$$
$$\tau_4 = cos^{-1}(CO), \tag{1.44}$$

where the input-output relationship may be expressed as,

$$A(\theta_1)cos(\theta_4) + B(\theta_1)sin(\theta_4) = C(\theta_1), \tag{1.45}$$

while the coefficients $A(\theta_1), B(\theta_1)$ and $C(\theta_1)$ are defined as,

$$A(\theta_1) = cos(\theta_1)sin(\tau_1)cos(\tau_4)sin(\tau_3) - cos(\tau_1)sin(\tau_4)sin(\tau_3), \tag{1.46}$$
$$B(\theta_1) = sin(\theta_1)sin(\tau_1)sin(\tau_3), \tag{1.47}$$
$$C(\theta_1) = cos(\tau_2) - cos(\theta_1)sin(\tau_1)sin(\tau_4)cos(\tau_3) - cos(\tau_1)cos(\tau_4)cos(\tau_3). \tag{1.48}$$

Equation (1.45) contains two solutions for the output angle which satisfies the equation for every single input angle parameter, indicating that for spherical RRRR function generators, there is always an elbow-up and elbow-down configuration if the linkage can be assembled. Considering its similarities to the Freudenstein equation, many of the same techniques regarding design and structural error minimisation are used in order to synthesise spherical RRRR function generators. However, the construction of the objective functions required to solve for the design and structural errors will be fundamentally different, and bear no resemblance to the planar RRRR solutions [40].

1.6 Extension of Current Theory

Considering the nature of the *design and structural error* metrics, it would stand to reason that a designer would prefer to synthesise a linkage with as many precision points as possible in order to develop a function generating linkage which approximates the desired function as closely as possible over the desired IO range of the linkage. While this goal is reasonable, limitations exist in the form of singular values within the inversion

of the matrices required to solve for the values of the linkage parameters, k_i. Accommodating this desire can take several forms. For example, the design error may have some minimum desired value set, above which point an iterative method is applied in order to develop a linkage which most closely approximates the function over the desired range by increasing the cardinality of the generating dataset incrementally until the design error lower bound is reached. From this point, the designer may use the typical structural error synthesis method to complete the design of the linkage.

Although it would stand to reason that if the ideal data set for the kinematic synthesis of a mechanism has as many points as possible within it, that integration is an obvious design step. Considering that on a data set of sufficiently large cardinality, the design and structural error minimising linkages tend to converge to the same link lengths [33], integration would also eliminate the need for the explicit computation of the iterative non-linear structural error. A data set of infinite cardinality would also necessarily eliminate the need for algorithms such as SVD, as in a data set of infinitely many points the impact of any subset of points on the error is, by definition, zero. While it was demonstrated in [57] that this is possible, it will be shown that the applicability of this method to the Freudenstein equation is limited due to the mathematical nature of the equation itself.

1.6.1 Limitations of the Freudenstein Equation

While the Freudenstein equation has, historically, been used to synthesise four-bar function generating linkages, there are some distinct and notable limitations within its formulation. First and foremost, while the Freudenstein equation may be used for the kinematic synthesis of linkages, due to the nature of the development of the equation, analysis of the suitability of any linkage is necessarily a secondary step within the kinematic synthesis problem; the Freudenstein equation itself is only formulated to showcase the IO relationship, and is completely insensitive to secondary characteristics of a linkage, such as the transmission angle.

Freudenstein's method for the development of IO equations may be followed identically in order to develop expressions for some of the intermediate joint angle parameters, though the resulting equations make use of fundamentally different link length parameters, k_i, within their formulation, see [80, 72] for example. However, this is typically not performed and instead, designers tend to rely on geometric constructions of the linkages or on secondary mathematical tools to check these secondary characteristics. The lack of intermediate joint angle relationships also means that the functional relationship between the input of the input dyad and the other output parameters is completely unknown. This thesis will show that, through the development of AIO equations, all of these limitations may be completely circumvented in order to provide a clear and concise method for the optimal kinematic IO design of four-bar function generators.

1.6.2 Integration of the Freudenstein Equation

Within [57], a method for the integration of the Freudenstein equation was developed through the use of a functional metric space which depends solely on the continuity of the functional IO relationship. The explicit definition of the procedure for accomplishing this continuous approximation with the Freudenstein equation is replicated herein for emphasis, while the theory and proofs required to generate the following relations can be found in [57]. At its most fundamental, this method relies on the minimisation of the residual of the integrated Freudenstein equation over the input angular range for which the IO function is to be developed,

$$\| \mathbf{d}(\alpha,\beta) \|_2 = \Big(\int_{\Delta\theta_{1,i}}^{\Delta\theta_{1,f}} (k_1 + k_2\cos(\alpha+\Delta\theta_1) - k_3\cos(\beta+\Delta\theta_4) - \cos(\alpha+\Delta\theta_1-(\beta+\Delta\theta_4))^2)\Big)^{\frac{1}{2}}, \quad (1.49)$$

which simplifies, following some algebraic manipulation, to,

$$\| \mathbf{d}(\alpha,\beta) \|_2^2 = \mathbf{k}^T\mathbf{A}(\alpha,\beta)\mathbf{k} - 2\mathbf{e}(\alpha,\beta)^T + c(\alpha,\beta). \quad (1.50)$$

Where the matrix, $\mathbf{A}(\alpha, \beta)$ is a positive 3×3 semidefinite matrix with six distinct elements, a_{ij},

$$a_{11} = \int_{\Delta\theta_{1,0}}^{\Delta\theta_{1,f}} d\Delta\theta_1 = \Delta\theta_{1,f} - \Delta\theta_{1,0}, \tag{1.51}$$

$$a_{12} = \int_{\Delta\theta_{1,0}}^{\Delta\theta_{1,f}} cos(\beta + \Delta\theta_4) d\Delta\theta_1, \tag{1.52}$$

$$a_{13} = -\int_{\Delta\theta_{1,0}}^{\Delta\theta_{1,f}} cos(\alpha + \Delta\theta_1) d\Delta\theta_1, \tag{1.53}$$

$$a_{22} = \int_{\Delta\theta_{1,0}}^{\Delta\theta_{1,f}} cos^2(\beta + \Delta\theta_4) d\Delta\theta_1, \tag{1.54}$$

$$a_{23} = -\int_{\Delta\theta_{1,0}}^{\Delta\theta_{1,f}} cos(\beta + \Delta\theta_4) cos(\alpha + \Delta\theta_1) d\Delta\theta_1; \tag{1.55}$$

$$a_{33} = \int_{\Delta\theta_{1,0}}^{\Delta\theta_{1,f}} cos(\alpha + \Delta\theta_1) d\Delta\theta_1. \tag{1.56}$$

The vector $\mathbf{e}(\alpha, \beta)$ is a 3-dimensional vector,

$$e_1 = \int_{\Delta\theta_{1,0}}^{\Delta\theta_{1,f}} cos(\alpha + \Delta\theta_1 - \beta - \Delta\theta_4) d\Delta\theta_1, \tag{1.57}$$

$$e_2 = \int_{\Delta\theta_{1,0}}^{\Delta\theta_{1,f}} cos(\beta + \Delta\theta_4) cos(\alpha + \Delta\theta_1 - \beta - \Delta\theta_4) d\Delta\theta_1, \tag{1.58}$$

$$e_3 = \int_{\Delta\theta_{1,0}}^{\Delta\theta_{1,f}} cos(\alpha + \Delta\theta_1) cos(\alpha + \Delta\theta_1 - \beta - \Delta\theta_4) d\Delta\theta_1. \tag{1.59}$$

Finally, the scalar coefficient, $c(\alpha, \beta)$ is expressed as,

$$c(\alpha, \beta) = \int_{\Delta\theta_{1,0}}^{\Delta\theta_{1,f}} cos^2(\alpha + \Delta\theta_1 - \beta - \Delta\theta_4) d\Delta\theta_1, \tag{1.60}$$

where all terms containing $\Delta\theta_4$ are computed as the desired functional relationship between $\Delta\theta_4$ and, $\Delta\theta_1$, such that $\Delta\theta_4 = f(\Delta\theta_1)$. Each of these factors requires numerical integration over the desired input-output range, and thus iteration in the event that the input-output range is changed, provided that an undesirable result is developed from the initial attempt. If $\mathbf{A}(\alpha, \beta)$ is *positive definite*, the optimal Freudenstein parameters, $\mathbf{k}^*(\alpha, \beta)$, which minimise the design error of the function generating linkage, $\| \mathbf{d}(\alpha, \beta) \|_2$ are:

$$\mathbf{k}^*(\alpha,\beta) = \mathbf{A}^{-1}(\alpha,\beta)\mathbf{e}(\alpha,\beta), \tag{1.61}$$

while the square of the design error, being minimised by the above parameter set may be expressed as,

$$min \parallel \mathbf{d}(\alpha,\beta) \parallel_2^2 = c(\alpha,\beta) - \mathbf{e}(\alpha,\beta)^t\mathbf{A}^{-1}(\alpha,\beta)\mathbf{e}(\alpha,\beta). \tag{1.62}$$

The positive definiteness of $\mathbf{A}(\alpha,\beta)$ is dependent on the fact that the dial zero set, (α,β), minimises the condition number, κ of $\mathbf{A}(\alpha,\beta)$, and is only valid if the (α,β) set is a unique global minimising set. The Nelder-Mead downhill simplex algorithm [81] may be used to iteratively minimize the condition number of $\mathbf{A}(\alpha,\beta)$ for a given IO range. From this point, the optimal Freudenstein parameters may be computed using these optimal dial zeroes. While this method can be shown to, in the general sense, compute the structural error minimising linkage parameters without explicitly evaluating the structural error of the function generating linkage by continuously minimising the design error of the linkage, it is a large computational undertaking. Using Matlab and its adaptive Lobatto quadrature [82] minimisation schema, the algorithm requires more than four hours to run to completion on an Intel 32-Bit Dual core CPU @ 3.10 GHz.

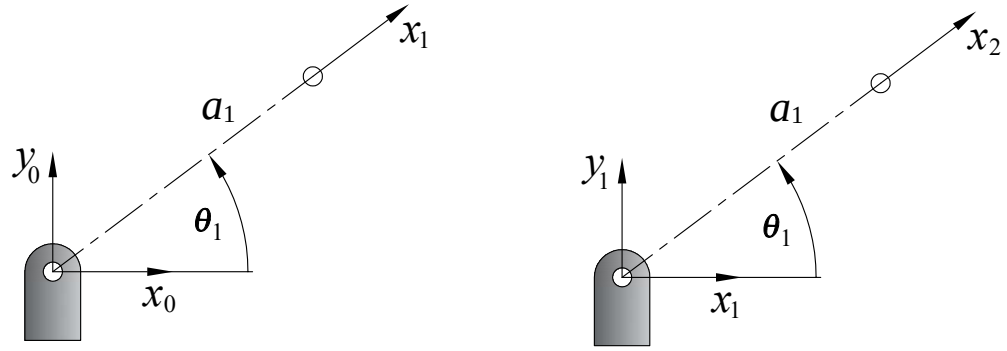
While it is entirely conceivable that modern computational capacities could reduce this time by an order of magnitude, a more expeditious method is clearly required. Further to this desire, the derivation of the aforementioned quantities was a non-trivial task, and is thus not applicable in general for a designer who is not extremely skilled in algebraic manipulations. For example, see [83] for the versions of Equation (1.51) through Equation (1.60) in their original form, before simplification. Thus, a more compact representation of the IO relationship contained within the Freudenstein formulation is required in order to expedite the continuous approximation methods. In order to generalise the development of these equations, the original Denavit-Hartenberg convention must be revisited.

1.6.3 The Denavit-Hartenberg Convention and the Planar RRRR Input Output Equation

Before discussing the continuous approximate synthesis approach, it will be useful to recall the matrix method for kinematic analysis and synthesis of linkages, which we call the Denavit-Hartenberg (DH) method, developed by Jacques Denavit and Richard Hartenberg. The method was first published in 1955 [84], and subsequently in their textbook on kinematic synthesis [2] in 1964. These methods will be vital in deriving the IO equations of all four-bar function generating linkages in a form that is more readily useful for optimisation techniques. The first step in the DH method applied to an arbitrary kinematic chain requires the identification and numbering of all joint axes.

In order to align with our previous publications on the development of the AIOs required for the work presented within this thesis, the same slightly modified, DH parameter coordinate system and naming convention will be used herein. The slight modification compared to the typical four-bar function generating architecture is that we have, as the relatively stationary coordinate system at the start of a kinematic chain, $x_0,\ y_0,\ z_0$. Hence, at any joint in the kinematic chain we measure, for example, the relative angle θ_i of link a_i about joint axis z_{i-1} from x_{i-1} to x_i, see Figure 1.6a. Whereas, in the original paper [85] the relatively stationary coordinate system at the start of a kinematic chain is the $x_1,\ y_1,\ z_1$ coordinate system, and the relative angle θ_i of link a_i is measured about z_i from x_i to x_{i+1}, see Figure 1.6b. Because the modification is so subtle, and because the mechanical engineering world has moved away from the original assignment rules found in [2, 84], we will henceforth refer to our version of the parametrisation simply as the DH method.

The DH parametrisation involves the allocation of coordinate systems to each link in the chain that move with the link, using a set of rules to locate the origin of the coordinate system and the orientation of the basis vectors. The position and orientation of consecutive links is defined by a homogeneous transformation matrix which maps coordinates of points in the coordinate system attached to link i to those of the same points described in a coordinate system attached to link $i-1$.



(A) Slightly modified definition of θ_1. (B) Original definition of θ_1.

FIGURE 1.6: Enumeration of the DH coordinate systems and assignment rules.

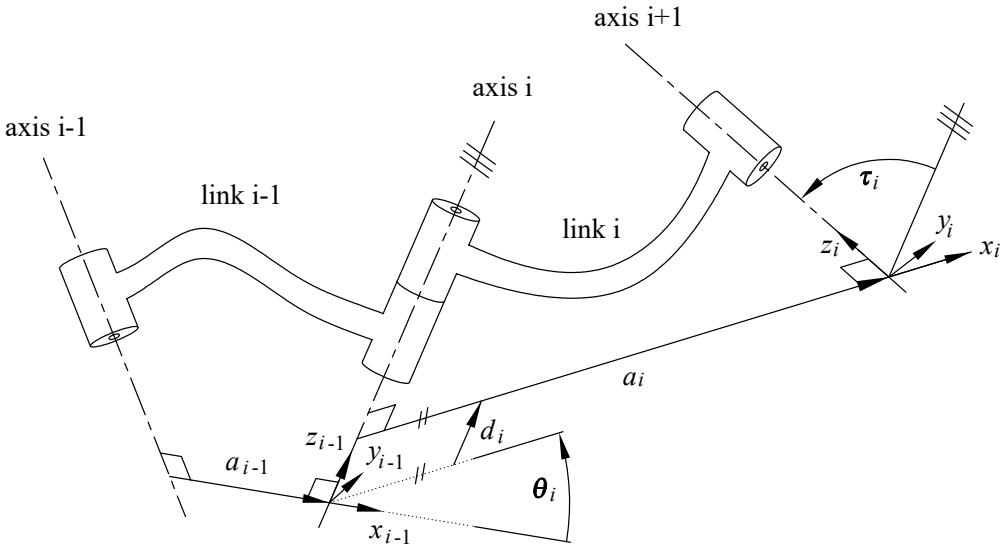


FIGURE 1.7: DH parameters in a general serial 3R kinematic chain.

To visualise the four DH parameters, consider two sequential arbitrary neighbouring links, $i-1$ and i. Two such links are illustrated, together with their DH parameters, in Figure 1.7. The DH parameters [84] are defined in the following way with our subtle modification.

θ_i, joint angle: the angle from x_{i-1} to x_i measured about z_{i-1}.

d_i, link offset: the distance from x_{i-1} to x_i measured along z_{i-1}.

τ_i, link twist: the angle from z_{i-1} to z_i measured about x_i.

a_i, link length: the directed distance[2] from z_{i-1} to z_i measured along x_i.

The DH coordinate transformation matrix, using the European convention for homogeneous coordinate arrays $[w,\ x,\ y,\ z]^T$, where w is the homogenising coordinate, is

$$
{}^{i-1}\mathbf{T}_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ a_i\cos\theta_i & \cos\theta_i & -\sin\theta_i\cos\tau_i & \sin\theta_i\sin\tau_i \\ a_i\sin\theta_i & \sin\theta_i & \cos\theta_i\cos\tau_i & -\cos\theta_i\sin\tau_i \\ d_i & 0 & \sin\tau_i & \cos\tau_i \end{bmatrix}. \tag{1.63}
$$

We then algebraise Equation 1.63 using the tangent half-angle substitutions for the joint and twist angles, where,

$$
\begin{aligned}
v_i &= \tan\left(\frac{\theta_i}{2}\right), \Rightarrow \cos\theta_i = \frac{1-v_i^2}{1+v_i^2}, \quad \sin\theta_i = \frac{2v_i}{1+v_i^2}, \\
\alpha_i &= \tan\left(\frac{\tau_i}{2}\right), \Rightarrow \cos\tau_i = \frac{1-\alpha_i^2}{1+\alpha_i^2}, \quad \sin\tau_i = \frac{2\alpha_i}{1+\alpha_i^2}.
\end{aligned}
$$

The detailed computations leading to the results presented herein use the Maple library MyKinematics [86] which requires the European homogeneous coordinate convention. The forward and inverse kinematics of serial chains are the concatenations of the individual transformation matrices in the appropriate order [13]. For example the forward kinematics problem of determining the position and orientation of the n^{th} link in a serial kinematic chain described in a relatively fixed non-moving base coordinate system 0, given the relevant DH parameters and values for the n joint variables, becomes conceptually simple matrix multiplication.

[2]Two points define a line in the plane, and the directed distance along that line is the location of the second point relative to the first. Within function generator synthesis, if a negative distance arises from the computation of any a_i value, the second point defining the centre of the revolute joint at the distal end of that link is on the opposite side of the first revolute joint as would be expected from the typical joint circulation direction.

The DH method was largely intended for planar, spherical, and spatial four-bar simple closed kinematic chains, but has since become nearly universally applied and synonymous with the kinematics of mechanical systems in general, and robot mechanical systems in particular, see [87, 88, 89, 90] for instance, but there are many other modern examples. The serial nR chain is conceptually closed by equating the forward kinematics transformation matrix to the identity.

$$ {}^{0}\mathbf{T}_{n} \;=\; {}^{0}\mathbf{T}_{1}\,{}^{1}\mathbf{T}_{2}\,{}^{2}\mathbf{T}_{3}\cdots{}^{n-2}\mathbf{T}_{n-1}\,{}^{n-1}\mathbf{T}_{n} \;=\; \mathbf{I}. \tag{1.64} $$

The resulting matrix represents a set of implicit equations in terms of the link constants and all n joint angles. If we restrict ourselves to the planar RRRR simple closed kinematic chain, and the IO equation that relates θ_4 to θ_1, then the intermediate angles θ_2 and θ_3 must be eliminated using the available equations. What remains is a single implicit equation in θ_4 and θ_1, or in the cases of the algebraic equations, v_1 and v_4 respectively.

Coordinate System Considerations

While all of the algebraic input-output equations have been derived using the DH method coordinate systems, in order to compare results contained within this thesis to those which are common within the literature, it will be required to present the algebraic equations in terms of their coordinate systems as defined by Freudenstein and the bulk of the literature up to this date. However, due to the complexities of the generation of the spherical RRRR function generator and moreover the spatial RSSR function generator, these equations will be presented explicitly in terms of the DH method parameter assignments, thus indicating a counter-clockwise joint parameter circulation.

This change of convention may seem needlessly confusing at first, but there exists a simple mathematical reason for the switch; the derivation of the Freudenstein equation itself is reliant on the use of the absolute coordinate systems and angles. If the derivation is attempted using counter-clockwise positive joint angle circulation, it is not possible to close the vector loop in such a way so as to obtain these absolute joint angle parameters. Thus, the Freudenstein equation was derived using a clockwise circulation. Of note, the

Freudenstein equation was developed in 1954, while the DH method and convention would not emerge for another year, until Denavit and Hartenberg first published their works in 1955. It may be of use for the reader to consider this reference coordinate system change to be a modernisation in the development of the Freudenstein equation in order to ensure that it is expressed with the most mathematically consistent, and modern, modelling standards which are available.

Fortunately, the equations displayed herein within the DH coordinate system result in the same optimal function generator linkage dimensional synthesis results as those developed through the classical vector loop closure methods. This similarity enables an easy comparison between the linkages and their relevant comparatives in the literature.

1.6.4 Algebraic Input-Output Equations of the Planar RRRR Mechanism

Classically, the solution to a planar four-bar function generation problem is conducted using the Freudenstein equation [32], and it has been demonstrated in [57] that this form of the function generator problem can be integrated in order to synthesise the design and structural error minimising linkage for a given IO function over some desired range. However, the expressions and functions required to perform this minimisation are quite complicated, and a more compact form of the problem is greatly desirable. It will herein be shown that a computational simplification of the trigonometric Freudenstein IO equation is achieved with the algebraic IO equation.

The algebraic IO equations were derived in [91, 92, 93] through the use of displacement constraints projected into a planar subset of Study's soma space for each of the three planar four-bar function generator architectures. We will commence with the algebraic IO equation for the planar RRRR mechanism, seen in Equation (1.65). 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 Figure 1.1:

$$Av_1^2v_4^2 + Bv_1^2 + Cv_4^2 + D - 8a_1a_3v_1v_4 = 0; \tag{1.65}$$

where,

$$A = A_1A_2 = (a_1 - a_2 + a_3 - a_4)(a_1 + a_2 + a_3 - a_4), \tag{1.66}$$
$$B = B_1B_2 = (a_1 + a_2 - a_3 - a_4)(a_1 - a_2 - a_3 - a_4), \tag{1.67}$$
$$C = C_1C_2 = (a_1 - a_2 - a_3 + a_4)(a_1 + a_2 - a_3 + a_4), \tag{1.68}$$
$$D = D_1D_2 = (a_1 + a_2 + a_3 + a_4)(a_1 - a_2 + a_3 + a_4). \tag{1.69}$$

Using the same derivation techniques as in [92], all $v_i - v_j$ algebraic IO equations, corresponding to the IO equation with each independent set of angular parameters for the RRRR linkage are,

$$A_1B_2v_1^2v_2^2 + A_2B_1v_1^2 + C_1D_2v_2^2 - 8a_2a_4v_1v_2 + C_2D_1 = 0, \tag{1.70}$$
$$A_1B_1v_1^2v_3^2 + A_2B_2v_1^2 + C_2D_2v_3^2 + C_1D_1 = 0, \tag{1.71}$$
$$A_1D_2v_2^2v_3^2 + B_2C_1v_2^2 + B_1C_2v_3^2 - 8a_1a_3v_2v_3 + A_2D_1 = 0, \tag{1.72}$$
$$A_1C_1v_2^2v_4^2 + B_2D_2v_2^2 + A_2C_2v_4^2 + B_1D_1 = 0, \tag{1.73}$$
$$A_1C_2v_3^2v_4^2 + B_1D_2v_3^2 + A_2C_1v_4^2 + 8a_2a_4v_3v_4 + B_2D_2 = 0, \tag{1.74}$$

where each $v_i - v_j$ equation is named by the first $v_i - v_j$ present in the equation.

Now, in contrast to the Freudenstein methods and their derivations using absolute angular parameters, each of the six algebraic input-output equations depends on the same set of bilinear link length coefficients. The impact of this fact is difficult to overstate; in the original vector loop closure method, if a function generation problem is desired in the $v_1 - v_3$ IO pair, all of the linkage parameter coefficients, k_i, will have a completely different form from those constructed for the $v_1 - v_4$ IO equation. Furthermore, this fact holds true for each and every $v_i - v_j$ pair comprising the linkage. Not only does the fact that the linear link length coefficients remain the same for each $v_i - v_j$ pair make the equations themselves more easily used, but it implies something decidedly powerful about the linkage design space itself; each set of link lengths which satisfy any of the above equations must, by definition, satisfy all of the equations, and thus, any single set of link lengths simultaneously defines six separate functions.

Upon investigation found in [94], it was shown that the design parameter space corresponding to the RRRR function generating linkage actually contains stellated octahedron, shown in Figure 1.8. The space which this octahedron occupies may be considered

a projective space, where a_4, the ground fixed link length, is arbitrarily treated as the homogenising coordinate. Following this, each axis of the space corresponds to link length a_1, a_2, or, a_3. The eight planes containing the equilateral traingular faces of the octahedron are the eight bilinear factors of the coefficients in Equations (1.66) through (1.69). A linkage which satisfies the requirements for assembly occupies exactly one unique point within this design parameter space, and this point is either outside, or directly on the surface of, this stellated octahedron.

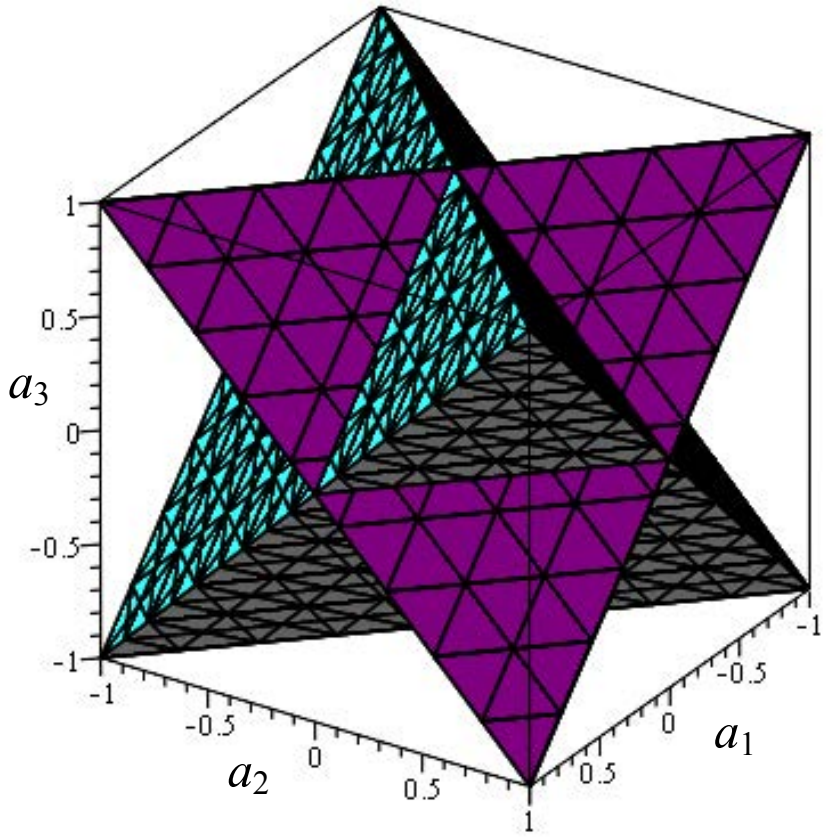


FIGURE 1.8: The stellated octahedron which occupies the centre of the design parameter space of a planar RRRR function generator.

While the shape contained within the design parameter space is of little consequence to the designer, as linkages which are present on the surface of the stellated octahedron will only be conditionally assemblable, it is interesting to note that the stellated octahedron is the only regular compound of two tetrahedra, while also being the simplest of the polyhedral compounds [95].

1.6.5 Algebraic Input-Output Equations of the Planar RRRP Mechanism

For an RRRP mechanism, the IO parameters are the input angle parameter v_1 and the output P-pair linear excursion d_4, while the design parameters are link lengths a_1, a_2, a_4, and the slider inclination angle parameter v_4, illustrated in Figure 1.2. Following an identical derivation methodology to the 4R mechanism, the RRRP algebraic IO equation is obtained:

$$Ad_4{}^2v_1{}^2 + Cd_4v_1{}^2 - 8a_1d_4v_1v_4 + Bd_4{}^2 + Ev_1{}^2 + Dd_4 + F = 0, \tag{1.75}$$

such that,

$$A = v_4^2 + 1, \tag{1.76}$$
$$B = v_4^2 + 1, \tag{1.77}$$
$$C = -2(v_4 - 1)(v_4 + 1)(a_1 + a_4), \tag{1.78}$$
$$D = 2(v_4 - 1)(v_4 + 1)(a_1 - a_4), \tag{1.79}$$
$$E = (v_4^2 + 1)(a_1 + a_2 + a_4)(a_1 - a_2 + a_4), \tag{1.80}$$
$$F = (v_4^2 + 1)(a_1 + a_2 - a_4)(a_1 - a_2 - a_4). \tag{1.81}$$

Simplifications to the RRRP Input-Output Formulation

For all other linkage architectures contained herein, only one set of algebraic IO (AIO) equations will be presented, however, in the case of the RRRP mechanism, a single ubiquitous simplification may be realised, which fundamentally alters the formulation of the IO relationships. In order to compare the methods contained herein with literature, however, the formulation of the IO relationships from the previous section must be used. Thus, for completeness, both representations of the RRRP IO equations must be discussed.

The simplification for the AIOs for the RRRP function generating linkage takes a the form of a single rotation, so as to ensure that the vector which contains the first revolute joint and the normal to the output track of the mechanism is horizontal. This necessarily sets the value of the track angle, θ_4, equal to 90 degrees. Thus when algebraising this

value, the tangent half angle parameter, v_4, is set equal to 1. This single rotation changes all of the IO equations to be far more compact, and with no loss in generality of the solution. The RRRP $(\nu_1 - d_4)$ IO equation then becomes,

$$v_1^2 d_4^2 + R v_1^2 + d_4^2 - 4 a_1 v_1 d_4 + S = 0, \tag{1.82}$$

where

$$R = R_1 R_2 = (a_1 + a_2 - a_4)(a_1 - a_2 - a_4), \tag{1.83}$$

$$S = S_1 S_2 = (a_1 + a_2 + a_4)(a_1 - a_2 + a_4). \tag{1.84}$$

The remaining five IO equations in terms of these bilinear link length coefficients then become:

$$\begin{aligned} R_2 v_1^2 v_2^2 + R_1 v_1^2 - S_2 v_2^2 + 4 a_2 v_1 v_2 - S_1 &= 0; & (1.85)\\ R_1 v_1^2 v_3^2 + R_2 v_1^2 - S_2 v_3^2 - S_1 &= 0; & (1.86)\\ S_2 v_2^2 v_3^2 - R_2 v_2^2 - R_1 v_3^2 - 4 a_1 v_2 v_3 &= 0; & (1.87)\\ v_2^2 d_4^2 - R_2 S_2 v_2^2 + d_4^2 - R_1 S_1 &= 0; & (1.88)\\ v_3^2 d_4^2 + R_1 S_2 v_3^2 + d_4^2 + 4 a_2 v_3 d_4 - R_2 S_1 &= 0. & (1.89) \end{aligned}$$

For comparisons to literature, the equations listed in Section 1.6.5 will be used, but for extensions to the classical theory contained herein, the formulations contained within this subsection will be used due to their compact nature.

1.6.6 Algebraic Input-Output Equations of the Planar PRRP Mechanism

Following from these two cases, the PRRP IO equation is presented in terms of the clockwise vector loop closure methods typical of classical methods. The algebraic IO

equation for a PRRP mechanism, with IO variables d_1 and d_4 and constant design parameters v_1, a_2, and v_4 (see Figure 1.3) can be obtained in the same way, yielding:

$$A{d_1}^2 + B{d_4}^2 + Cd_1d_4 + Dd_1 + Ed_4 + F = 0, \tag{1.90}$$

where,

$$A = (v_4^2 + 1)(v_1^2 + 1), \tag{1.91}$$
$$B = (v_4^2 + 1)(v_1^2 + 1), \tag{1.92}$$
$$C = -2(v_1v_4 - v_1 + v_4 + 1)(v_1v_4 + v_1 - v_4 + 1), \tag{1.93}$$
$$D = 2a_4(v_4^2 + 1)(v_1 - 1)(v_1 + 1), \tag{1.94}$$
$$E = -2a_4(v_4 - 1)(v_4 + 1)(v_1^2 + 1), \tag{1.95}$$
$$F = -(v_4^2 + 1)(v_1^2 + 1)(a_2 - a_4)(a_2 + a_4). \tag{1.96}$$

In the case of the PRRP function generator, the DH parameter equations differ significantly from those which result from the coordinate system associated with classical vector-loop closure methods, and for comparison to literature, both must be discussed herein.

DH Method Planar PRRP Input-Output Equations

When following the DH method for parameterising the PRRP function generator as contained within [93], three constant factors are revealed following the Groebner basis elimination and are defined as,

$$\begin{aligned} T &= a_2^2(\alpha_4^2 + 1); \\ U &= a_2(\alpha_4^2 - 1); \\ V &= a_2(\alpha_4^2 + 1). \end{aligned}$$

Using these coefficients the six algebraic IO equations are:

$$(\alpha_4^2+1)(d_1^2+d_4^2)-2(\alpha_4^2-1)d_1d_4-T=0; \tag{1.97}$$

$$2\alpha_4 d_1 v_2^2+Uv_2^2+2\alpha_4 d_1-4a_2\alpha_4 v_2-U=0; \tag{1.98}$$

$$2\alpha_4 d_1 v_3^2-Vv_3^2+2\alpha_4 d_1+V=0; \tag{1.99}$$

$$\alpha_4 v_2 v_3-v_2-v_3-\alpha_4=0; \tag{1.100}$$

$$2\alpha_4 v_2^2 d_4+Vv_2^2+2\alpha_4 d_4-V=0; \tag{1.101}$$

$$2\alpha_4 v_3^2 d_4-Uv_3^2+4a_2\alpha_4 v_3+2\alpha_4 d_4+U=0. \tag{1.102}$$

The PRRP linkage concludes the gamut of planar linkages for which AIOs have been developed at the time of this publication, and thus the discussion moves towards the description of the AIO functions associated with the spherical RRRR linkage.

1.6.7 Algebraic Input-Output Equation of the Spherical RRRR Linkage

In the case of the spherical RRRR linkage, the derivation of the AIO equations which are used for this thesis is similar to that of the planar four-bar function generating linkages. Following the use of the Groebner basis elimination algorithm, the following six algebraic equations are derived:

$$A_1A_2v_1^2v_4^2+B_1B_2v_1^2+C_1C_2v_4^2+8\alpha_1\alpha_3(\alpha_4^2+1)(\alpha_2^2+1)v_1v_4+D_1D_2=0; \tag{1.103}$$

$$A_1B_2v_1^2v_2^2+A_2B_1v_1^2+C_1D_2v_2^2+8\alpha_2\alpha_4(\alpha_1^2+1)(\alpha_3^2+1)v_1v_2+C_2D_1=0; \tag{1.104}$$

$$A_1B_1v_1^2v_3^2+A_2B_2v_1^2+C_2D_2v_3^2+C_1D_1=0; \tag{1.105}$$

$$A_1D_2v_2^2v_3^2+B_2C_1v_2^2+B_1C_2v_3^2-8\alpha_1\alpha_3(\alpha_4^2+1)(\alpha_2^2+1)v_2v_3+A_2D_2=0; \tag{1.106}$$

$$A_1C_1v_2^2v_4^2+B_2D_2v_2^2+A_2C_2v_4^2+B_1D_1=0; \tag{1.107}$$

$$A_1C_2v_3^2v_4^2+B_1D_2v_3^2+A_2C_1v_4^2+8\alpha_2\alpha_4(\alpha_1^2+1)(\alpha_3^2+1)v_1v_2+B_2D_1=0. \tag{1.108}$$

In keeping with previous linkage architectures, the linkage design parameter sets A_i, B_i, C_i, and D_i are identical from one (v_i, v_j) equation set to another, containing only

the linkage arclength, α_i, terms. However, these linkage parameter sets themselves are more complicated, taking the form of functions which are cubic in, α_i, such that,

$$A_1 = \alpha_1\alpha_2\alpha_3 - \alpha_1\alpha_2\alpha_4 + \alpha_1\alpha_3\alpha_4 - \alpha_2\alpha_3\alpha_4 + \alpha_1 - \alpha_2 + \alpha_3 - \alpha_4, \tag{1.109}$$
$$A_2 = \alpha_1\alpha_2\alpha_3 - \alpha_1\alpha_2\alpha_4 - \alpha_1\alpha_3\alpha_4 - \alpha_2\alpha_3\alpha_4 - \alpha_1 - \alpha_2 - \alpha_3 + \alpha_4, \tag{1.110}$$
$$B_1 = \alpha_1\alpha_2\alpha_3 + \alpha_1\alpha_2\alpha_4 - \alpha_1\alpha_3\alpha_4 - \alpha_2\alpha_3\alpha_4 + \alpha_1 + \alpha_2 - \alpha_3 - \alpha_4, \tag{1.111}$$
$$B_2 = \alpha_1\alpha_2\alpha_3 + \alpha_1\alpha_2\alpha_4 + \alpha_1\alpha_3\alpha_4 - \alpha_2\alpha_3\alpha_4 - \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4, \tag{1.112}$$
$$C_1 = \alpha_1\alpha_2\alpha_3 - \alpha_1\alpha_2\alpha_4 - \alpha_1\alpha_3\alpha_4 + \alpha_2\alpha_3\alpha_4 - \alpha_1 + \alpha_2 + \alpha_3 - \alpha_4, \tag{1.113}$$
$$C_2 = \alpha_1\alpha_2\alpha_3 - \alpha_1\alpha_2\alpha_4 + \alpha_1\alpha_3\alpha_4 + \alpha_2\alpha_3\alpha_4 + \alpha_1 + \alpha_2 - \alpha_3 + \alpha_4, \tag{1.114}$$
$$D_1 = \alpha_1\alpha_2\alpha_3 + \alpha_1\alpha_2\alpha_4 + \alpha_1\alpha_3\alpha_4 + \alpha_2\alpha_3\alpha_4 - \alpha_1 - \alpha_2 - \alpha_3 - \alpha_4, \tag{1.115}$$
$$D_2 = \alpha_1\alpha_2\alpha_3 + \alpha_1\alpha_2\alpha_4 - \alpha_1\alpha_3\alpha_4 + \alpha_2\alpha_3\alpha_4 + \alpha_1 - \alpha_2 + \alpha_3 + \alpha_4. \tag{1.116}$$

Due to the nature of these bicubic functions, it is implied that, when viewing the design parameter space of the spherical RRRR linkage, the sides of the figure which encapsulate the center of the design parameter space will not be planar. Indeed, this is the case, as when the design parameter space of the spherical RRRR linkage was examined, contains the edges of a stellated octahedron, but not the faces. This figure is shown in Figure 1.9.

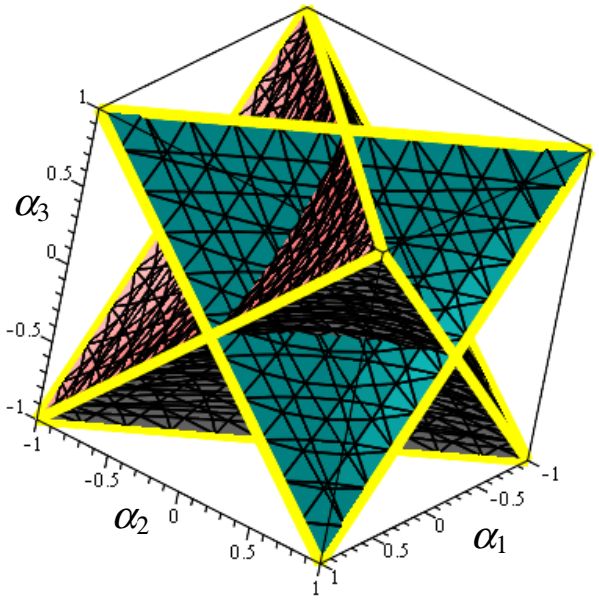


FIGURE 1.9: The eight degenerate bi-cubic surfaces at the centre of the spherical RRRR design parameter space, with all 12 real unique lines highlighted to form the vertices of a stelated octahedron.

The eight degenerate bi-cubic surfaces contain three real finite lines each, for a total of 12 distinct lines. These lines are the intersections of different pairs of surfaces, and all 12 of them intersect to form the edges of a stellated octahedron.

1.6.8 Algebraic IO Equation of the Spatial RSSR Mechanism

The derivation of the algebraic IO equation of the spatial RSSR function generating mechanism requires a fundamentally different set of tools than the previous planar and spherical four-bar linkages. The geometry of this function generator is shown in Figure 1.10 in order to inform the remainder of this discussion. Some additional modelling steps must be discussed to inform the reader of some subtle differences in what will be presented herein. The modelling of the RSSR linkage requires the modelling of two RS-dyads, where the spherical joint is a higher order joint with multiple degrees of freedom. In order to account for these additional degrees of freedom, the modelling must examine the spherical joint by breaking it down into its constituent lower order joints: in this case, a triplet of mutually orthogonal revolute joints whose axes intersect in the joint centre [5].

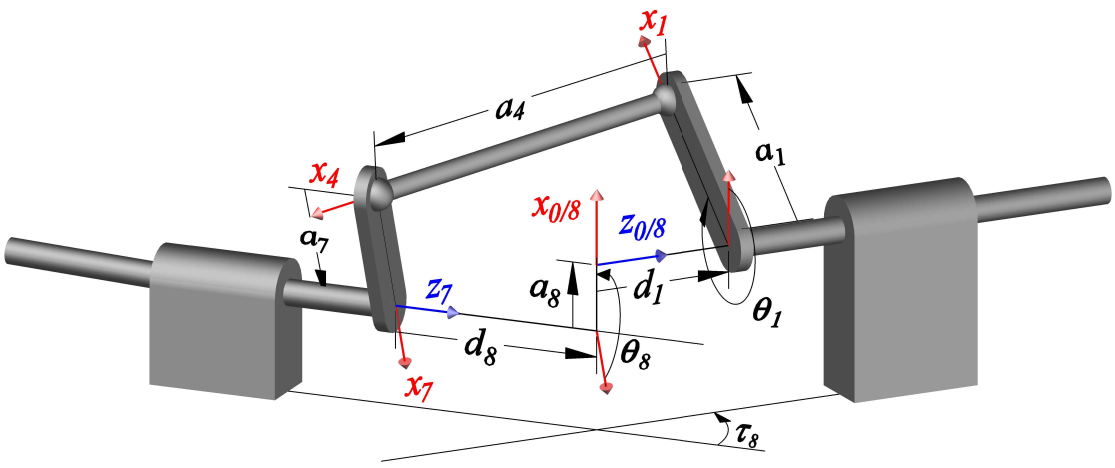


FIGURE 1.10: The kinematic geometry of the RSSR linkage, will relevant joint lengths and angles labelled for DH method parametrisation.

Due to the representation of the RS-pair within this modelling step, it is necessarily implied that while the $v_i - v_j$ input-output equations may exist, they will be largely meaningless due to the coupled nature of each of the six modelled revolute joints within

the pair of spherical joints attached to the coupler of this mechanism. What would instead be desired is the vector corresponding to the orientation of the coupler link in space for the analysis of linkage properties such as the transmission angle. The algebraic equations which would represent this property have not, as of the time of writing this thesis, been formulated, and will not be discussed further. Nevertheless, the same DH method based kinematic closure technique was exercised for the RSSR linkage in order to obtain the $v_1 - v_4$ input-output equation. However, due to the complexity of the intermediate equations, the Groebner basis algorithm was unable to eliminate the image space coordinates and intermediate joint angle parameters in order to develop the algebraic input-output equation, a fundamental divergence from the results of previous linkage architectures. Instead, the linear implicitisation algorithm was used to perform the derivation for this linkage architecture [96].

Following the linear implicitisation algorithm and the application of some clever algebraic manipulations, the algebraic input-output equation of the RSSR linkage can be shown to be,

$$Av_1^2v_8^2 + 8d_1\alpha_8a_7v_1^2v_8 + 8d_8\alpha_8a_1v_1v_8^2 + Bv_1^2 + 8a_1a_7(\alpha_8 - 1)(\alpha_8 + 1)v_1v_8 + Cv_8^2 + 8d_8\alpha_8a_1v_1 + 8d_1\alpha_8a_7v_8 + D = 0, \quad (1.117)$$

where each of the coefficients, A, B, C, and D are comprised of a product of bilinear link length terms, as well as a quadratic term containing the twist angle between the joint rotation axes of each ground-fixed revolute joint of each dyad, and a term common to all four quantities. These linkage design parameter coefficients are defined as,

$$A = (\alpha_8^2 + 1)A_1A_2 + R, \quad (1.118)$$
$$B = (\alpha_8^2 + 1)B_1B_2 + R, \quad (1.119)$$
$$C = (\alpha_8^2 + 1)C_1C_2 + R, \quad (1.120)$$
$$D = (\alpha_8^2 + 1)D_1D_2 + R, \quad (1.121)$$

where each A_i, B_i, C_i, and D_i term may be expressed as linear functions of the link length parameters, a_i,

$$A_1 = (a_1 - a_4 + a_7 - a_8), \tag{1.122}$$
$$A_2 = (a_1 + a_4 + a_7 - a_8), \tag{1.123}$$
$$B_1 = (a_1 + a_4 - a_7 - a_8), \tag{1.124}$$
$$B_2 = (a_1 - a_4 - a_7 + a_8), \tag{1.125}$$
$$C_1 = (a_1 - a_4 - a_7 + a_8), \tag{1.126}$$
$$C_2 = (a_1 + a_4 - a_7 + a_8), \tag{1.127}$$
$$D_1 = (a_1 + a_4 + a_7 + a_8), \tag{1.128}$$
$$D_2 = (a_1 - a_4 + a_7 + a_8). \tag{1.129}$$

The single common term, R, is then expressed as,

$$R = (d_1 - d_8)^2 \alpha_8^2 + (d_1 + d_8)^2, \tag{1.130}$$

which is quadratic in the revolute joint center offsets, and the twist angle between their axes of rotation. As with the non-homogeneous planar four-bar function generators, the design parameter space for the spatial RSSR function generator would make little sense to the reader, and furthermore, it would only be visible in a sixth dimensional space in its most general form.

It may also be of interest to the reader to note that the signs and link length parameters present within each coefficient contained in Equations (1.122) through (1.129) are precisely the same as those contained within the exact same coefficients of the planar RRRR IO equations. Indeed, when $d_1 = d_8 = \alpha_8 = 0$, the exact planar RRRR bilinear link length coefficients are replicated in the RSSR link length coefficients.

1.7 Problem Statement and Thesis Structure

Planar four-bar function generating linkage synthesis problems are typically solved by discrete exact or approximate synthesis methods as outlined previously in this chapter. However, the structural error minimisation problem solution requires the designer to

initially solve the design error problem to provide an initial guess of sufficient fidelity to the numerical solver.

The continuous approximate function generator synthesis approach integrates the Freudentstein synthesis equations over a range of input values [57]. The cardinality of the IO data set becomes infinity, thereby forcing the design and structural errors of the function generator to converge, converting a non-linear least squares optimisation problem to one that is a linear least squares problem. However, the integration of the trigonometric Freudenstein Equation for IO function generation is quite onerous, and whatever gains in computational efficiency associated with no longer having to minimise the structural error are easily lost due to the computational cost of the integration.

In several other recent advances [91, 92, 97, 98, 99] it has been shown that there is a consistent and general approach to deriving an algebraic form of the Freudenstein IO equation. It stands to reason that an algebraic form of the IO function significantly simplifies the integration approach, requiring the user to only minimise one single equation with a given set of initial assumptions that may be readily computed from the AIO equation and desired IO function. Integrating the square of the AIO equation of any planar four-bar linkage and subsequently minimising the residual of this equation provides the structural error minimising linkage parameters. Given the set of linkage parameters that are generated from this minimisation, the AIO function may be solved explicitly to describe the function that is generated by the linkage, facilitating its comparison to the desired function through use of a simple integral and a percentage error calculation. Different planar function generating kinematic architectures may be compared to each other by way of the same percentage error computation, leading to efficient concurrent type and dimensional synthesis for any planar function generator problem.

It is often desired to obtain information regarding the displacement analysis between different links in the identified kinematic chain. Additionally, the designer must typically understand the angular velocities and accelerations associated with the identified kinematic geometry. There are six angle pairings in the quadrangle defined by a planar

RRRR mechanism; the first two derivatives of these six AIO equations lead to significantly simplified techniques to determine the extreme angular velocities and accelerations of the four links in the mechanism [100].

It is towards the solution to the problem of CAAIOS that the remainder of this thesis will be focused. First, an algorithm for the continuous approximate minimisation of the residual of these function generating AIO equations will be presented alongside results that allow for validation through the comparison of these link lengths to those resulting from the classical approaches to structural error minimisation. Following this, an algorithm for the concurrent type and dimensional synthesis problem will be presented with a demonstrative case for validation of the results which state that, not only can continuous approximate synthesis be easily employed, but that the computational simplicity of its implementation allows for the concurrent identification of the best error minimising four-bar linkage type as well as its corresponding dimensions [101]. Furthermore, given the myriad of constraints and applications that exist in the design of planar four-bar linkages, a method for multi-modal continuous approximate algebraic input-output synthesis (MMCAAIOS) will be developed, wherein which an extension of the CAAIOS algorithm may be used to generate a linkage which can be made to approximate modestly competing functions between different joint pairs [102, 103].

1.8 Statement of Originality

Certain aspects of planar, spherical, and spatial four-bar mechanism dimensional synthesis for function generation are presented herein for the first time. The following original contributions are of particular interest.

1. The continuous approximate IO dimensional synthesis technique is presented for planar, spherical, and spatial four-bar mechanisms for the first time.

2. That the continuous approximate IO dimensional synthesis technique demonstrates, in compact form, that the solution to the non-linear structural error minimisation problem is implied through the solution to the design error minimising

linkage when the cardinality of the prescribed dataset is infinite.

3. The algebraic IO equations allow for the comparison of generated and desired functions through the use of an integral and ratio of areas.

4. The CAAIOS algorithm performs its optimisation several orders of magnitude faster than equivalent methods which rely on the Freudenstein equation.

5. It is demonstrated that the CAAIOS technique presented herein can be applied to all planar, spherical, and spatial four-bar algebraic IO equations which exist within the literature without modification.

6. Combined type and dimensional synthesis is presented for the first time, where due to the ease of computing the design error minimising linkage via the CAAIOS method, the designer need not explicitly specify the linkage architecture used to generate the function.

7. Multimodal continuous approximate dimensional synthesis for competing function generation is described for the first time, and its limitations are explored.

Portions of these original results have appeared in five refereed publications: [28, 93, 101, 102, 103].

2 Continuous Approximate Algebraic Function Generator Synthesis

2.1 Continuous Approximate Synthesis via the Algebraic Input-Output Equations

The following sections will describe the novel algorithm for continuous approximate IO dimensional synthesis which can be applied to each of the planar RRRR, RRRP, and PRRP, as well as the spherical RRRR and spacial RSSR kinematic architectures. However, for the general case, the algorithm steps may be simply stated as:

1. Square the desired algebraic IO relationship for the planar four bar linkage architecture.

2. Express this squared function in an array containing the linkage parameters, $\mathbf{p}_A$, and synthesis matrix, $\mathbf{S}$, defined in Section 2.5.1.

3. Express the output parameter as the desired function of the input parameter and substitute the results into the synthesis matrix, $\mathbf{S}$.

4. Integrate the matrix obtained in step three, numerically, over the desired bounds for the approximation, generating the integrated synthesis matrix.

5. Expand the elements within the integrated $\mathbf{S}$ through multiplication with $\mathbf{p}_A$.

6. Generate initial guesses for the optimal linkage parameters by solving the linear AIO equation for three precision points.

7. Minimise the residual of this expanded synthesis equation, $\mathbf{p}_A^T \mathbf{S}_i \mathbf{p}_A$, over the field of real numbers with the precision point method linkage parameter solutions as an initial guess.

Ideally, the numerical integration required for step four of this process would be an analytical integral, however, in general, analytical integrals for these functions do not exist due to the complexity of the elements. 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, thereby identifying the design parameters that generate the desired function with the smallest possible structural error. Application of this methodology to the three planar architectures, and two non-planar architectures will be described in detail along with examples and comparisons with results of the discrete approximation methods based on classical structural error minimisation techniques.

2.2 Theory and Mathematical Modelling Concepts

The design and structural error minimisations, as stated in Sections 1.4.2 and 1.4.3, both aim to serve the same purpose of developing an error minimising function generator; the former operates through the minimisation of the residual of an equation which describes the system, while the latter minimises the difference between the desired output and the generated output parameter. While the structural error itself is a far better measure of the ability of a linkage in question to generate the desired input-output relationship, it is significantly more complicated from a computational perspective, and requires an initial guess which is typically developed through the minimisation of the design error before starting the structural error minimisation problem.

One of the most fundamental conceptual links that is being drawn within this thesis is that the minimisation of the structural error is implied by the solution which results

from minimising the design error over an infinite closed interval, and thus eliminates the need the solve the structural error problem. A claim such as this requires more than simple observational evidence, and should provide a methodologically and phenomenologically consistent explanation for how exactly this occurs. For the purposes of the analysis to follow herein, the problem formation is decidedly simple; considering the form of the AIOs as presented within Chapter 1, the design error takes a slightly more obvious and physically meaningful form.

If a set of input-output pairs is used to develop a linkage via conventional approximate dimensional synthesis techniques, in the algebraic sense, the minimisation algorithm will identify the linkage parameters which reduce the length of the line at each input-output pair to its minimum values depending on the set of points used for the minimisation. If this is extended to infinitely many points, as in the case of the integral, the minimisation of the resulting value is simply minimising the difference between the areas of the desired curve and what is generated by the linkage. This concept also provides a more physically meaningful metric which will be leveraged herein to develop comparisons between mechanisms and link length parameter sets: the difference between the areas under the curve of the desired function versus what is generated by the algebraic input-output equation.

It will be shown that not only does the integration and minimisation of the squared algebraic input-output equation yield exceptional results which improve upon the results developed by conventional structural error minimisation techniques found in the literature, but that the method is ubiquitous to every algebraic IO function generation relationship which exists, regardless of their architectural configuration.

It should also be noted that because the Freudenstein equation may be transformed to be identical to the AIO equations through the use of the tangent-half angle substitutions, that this property of the AIOs implies, by definition, that the same property is true for the Freudenstein equation. This implies that the same property is true of the classical design error problems, but that the nature of the proof is simply more mathematically complicated, and is obfuscated due to the trigonometric relationships used within the

Freudenstein derivation.

2.2.1 Mathematical Modelling Software

In order to accomplish the work involved in this thesis, Maple 2021 has been used for all of the computations. Maple was chosen due to its combined proficiency of algebraic manipulation and its ability to leverage an in-built library of minimisation algorithms. For the cases of the minimisation algorithms used in this thesis, Maple uses its host of Sequential Quadratic Programming (SQP) minimisation algorithms [104]. Constraints for all optimisations required in order to execute the algorithms are placed in terms of a simple initial guess on a continuous function, Maple will use its SQP solver to select either the modified Newton simplex [105] or Nelder-Mead downhill simplex algorithms [81]. While Nelder-Mead is often used for highly sensitive functions, the modified Newton downhill simplex algorithm is easily computed for all functions within this thesis, as their partial derivatives may be easily computed.

While it would be possible to produce a convergence time comparison between different optimisation algorithms, such a task lies well outside of the scope of this thesis. Additionally, while it is possible that the computational efficiency of these algorithms may be further enhanced through a more appropriate numerical engine, such as leveraging Python (and its wealth of C-suite optimisation libraries) to perform the minimisations, the algebraic manipulations available in Maple coupled with its in-built optimisation algorithms make it the most obvious single program to perform the entirely of the work presented herein.

2.3 Continuous Approximate Synthesis of Planar Four-Bar Function Generators

2.3.1 Continuous Approximate Synthesis for the RRRR Function Generator

For the purposes of the derivation of the CAAIOS for the planar RRRR function generator, the $v_1 - v_4$ AIO equation will be used for demonstration. 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. In order to accomplish this, the equation is partitioned into a matrix and array representation, where the synthesis matrix contains all of the IO variables, while the array which pre- and post-multiplies this matrix contains the four bilinear factors of the link lengths along with the $-8a_1a_4$ term, which scales with the bilinear v_1v_4 IO variable product in Equation (1.65). Hence, the squared $v_1 - v_4$ IO equation for all RRRR linkages is,

$$\mathbf{p_A}^T\mathbf{S}(v_1, v_4)\mathbf{p_A} = 0, \tag{2.1}$$

where,

$$\mathbf{p_A} = \begin{bmatrix} A \\ B \\ C \\ D \\ -8a_1a_4 \end{bmatrix} \tag{2.2}$$

and,

$$\mathbf{S}(v_1, v_4) = \begin{bmatrix} v_1^4v_4^4 & 2v_1^4v_4^2 & 2v_1^2v_4^4 & 2v_1^2v_4^2 & 2v_1^3v_4^3 \\ 0 & v_1^4 & 2v_1^2v_4^2 & 2v_1^2 & 2v_1^3v_4 \\ 0 & 0 & v_4^4 & 2v_4^2 & 2v_1v_4^3 \\ 0 & 0 & 0 & 1 & 2v_1v_4 \\ 0 & 0 & 0 & 0 & v_1^2v_4^2 \end{bmatrix}. \tag{2.3}$$

Next, the desired function, specified as $v_4 = f(v_1)$, is substituted into Equation (2.3) yielding

$$\mathbf{S}(v_1, f(v_1)) = \begin{bmatrix} v_1^4f(v_1)^4 & 2v_1^4f(v_1)^2 & 2v_1^2f(v_1)^4 & 2v_1^2f(v_1)^2 & 2v_1^3f(v_1)^3 \\ 0 & v_1^4 & 2v_1^2f(v_1)^2 & 2v_1^2 & 2v_1^3f(v_1) \\ 0 & 0 & f(v_1)^4 & 2f(v_1)^2 & 2v_1f(v_1)^3 \\ 0 & 0 & 0 & 1 & 2v_1f(v_1) \\ 0 & 0 & 0 & 0 & v_1^2f(v_1)^2 \end{bmatrix}. \tag{2.4}$$

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

$$\min_{(a_1,a_2,a_3,a_4)\in\mathbb{R}} \left(\begin{bmatrix} A & B & C & D & -8a_1a_4 \end{bmatrix} \int_{v_{1min}}^{v_{1max}} \mathbf{S}(v_1, f(v_1)) \begin{bmatrix} A \\ B \\ C \\ D \\ -8a_1a_4 \end{bmatrix} \right). \tag{2.5}$$

In Equation (2.46), the elements A, B, C, and D correspond to the four bilinear factors defined in Equations (1.66) through (1.69). However, these procedures are immensely sensitive to the initial guess.

In order to determine a useful initial guess, the solver must first be given a set of link lengths which are, in some sense, relatively close to the optimal link lengths. To accomplish this, one may solve the exact synthesis problem for this linkage. The three precision points are determined with the desired function $v_4 = f(v_1)$, and normalising the link lengths with a_4, leading to $a_4 = 1$. Three points may be chosen within the range for which the four-bar linkage is being synthesised, typically at random. However, the designer must be careful so as to not choose points which would generate a linear dependence between the AIO equations being used for the precision point solution. For example, if a function which is being synthesised over the range of $v_1 = [-3, 3]$ and is symmetric, the user must not select symmetric points within this range, as the precision point solution will be indeterminate in this case.

With the initial guess for the link lengths, as well as a set of constraints for the optimiser, the linkage parameters which minimise the residual of Equation (2.46) may be computed. The constraints require the linkage to be a real linkage, $a_i^2 \geq 0$, forcing the link lengths to be real numbers. While zero-value link lengths do not represent a real four-bar linkage, this inclusive inequality constraint is required as a limitation of the solver. Negative link lengths must not be discarded as, given the range of input angle and the desired IO function, any of the link lengths may in fact be identified as negative in value. While the concept of a negative length may appear to be fundamentally flawed, as a length can not be negative, in the context of a function generator a link length is a directed distance on a vector which connects two revolute joint centres A and B. Given this fact, if the minimisation of the residual of the AIO equation requires a

negative link length, the negative sign simply indicates that the directed distance along the line which contains the revolute joint centres A and B is directed opposite to the direction of circulation.

Given the novelty of this approach, the following examples for the three planar function generating linkage architectures, where the coupler can have general plane motion, will be presented and compared to existing solutions within the literature. The following demonstration for the RRRR linkage architecture will be computed with the same function published in [91],

$$v_4 = 2 + \tan\left(\frac{v_1}{v_1^2 + 1}\right), \tag{2.6}$$

where (v_1, v_4) represent the tangent half angle parameters associated with the input and output link joint angles θ_1 and θ_4, respectively. The function $v_4 = f(v_1)$ is substituted into Equation (2.3). Once this substitution has been made, the resulting matrix is integrated between the bounds desired for the approximation, in this case $v_1 = 0 \ldots 2$. 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, which implicitly minimise the structural error.

Reported in [91] are the link lengths optimised using a discrete IO set whose cardinality is 10, which are,

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} -0.23 \\ 1.20 \\ 1.43 \\ 1 \end{bmatrix}. \tag{2.7}$$

Whereas the link lengths derived from the continuous approximation method are

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} -0.22 \\ 1.18 \\ 1.43 \\ 1 \end{bmatrix}, \tag{2.8}$$

showing an extremely high degree of agreement between the two results. The precision point and optimised linkage functions are both solved directly from the algebraic IO equation, Equation (1.65), as polynomials in the form of $v_4 = f(v_1)$. The precision

point method yields the following AIO equation with linkage parameters a_i from Equation (2.7),

$$-1.0153\, {v_1}^2 {v_4}^2 + 2.6075\, v_1 v_4 + 3.4640\, {v_1}^2 - 1.4173\, {v_4}^2 + 5.6695, \tag{2.9}$$

which may be solved for v_4 to plot the generated AIO function for each assembly mode of the linkage. The optimal AIO equation from the CAAIOS method after substitution of the values from Equation (2.8) into Equation (1.65) is,

$$-0.9944\, {v_1}^2 {v_4}^2 + 2.4522\, v_1 v_4 + 3.4394\, {v_1}^2 - 1.3540\, {v_4}^2 + 5.5321, \tag{2.10}$$

which may also be solved for v_4 to plot each AIO function for the two assembly modes of the optimal linkage. Both AIO functions were simplified for inclusion in this thesis proposal as the full algorithm uses rational representations of numbers to avoid concatenating computational errors associated with computations using floating point representations. Figure 2.1 shows both the precision point and CAAIOS optimised IO relationships.

Given the proximity of these curves to one another, a simple visual inspection is insufficient to determine the magnitude of the areas between the curves, thus a different method using the integrals of the generated functions to compute the percentage difference between the discrete and continuous algebraic approximations is used. For the RRRR function generator case, the following errors are computed relative to the desired function,

$$\text{Discrete Approximation Method} = 0.421\%, \tag{2.11}$$

$$\text{Continuous Approximate Optimised Function} = 0.012\%, \tag{2.12}$$

indicating a significant reduction in the percentage error associated with the continuous approximation method of an order of magnitude.

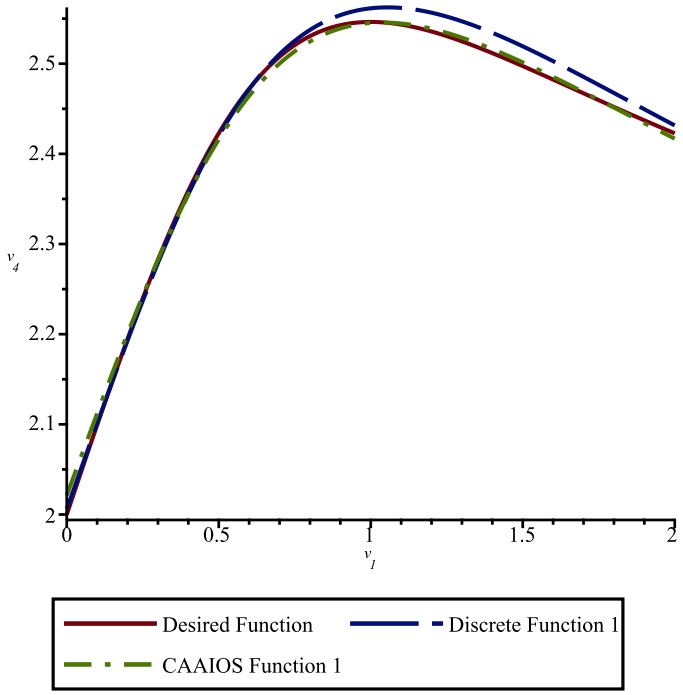


FIGURE 2.1: Comparison of RRRR desired and generated functions.

2.3.2 Continuous Approximate Synthesis for the RRRP Linkage

Given the nature of the continuous approximate design error minimisation, the approach is easily modified for any desired planar four-bar function generator topology. Consider the RRRP function generator illustrated in Figure 1.2. For comparisons to literature, the equations and coefficients developed in Section 1.6.5 will be used. After the AIO equation associated with this kinematic architecture, Equation (1.75), Equation (2.13) may be pre- and post-multiplied by an array containing the linkage parameter coefficients from Equation (1.76) to obtain the full squared AIO equation for the RRRP linkage architecture:

$$\mathbf{S}(v_1, a_3) = \begin{bmatrix} v_1^4 d_4^4 & 2v_1^2 d_4^4 & 2v_1^4 d_4^3 & 2v_1^3 d_4^3 & 2v_1^2 d_4^3 & 2v_1^4 d_4^2 & 2v_1^2 d_4^2 \\ 0 & d_4^4 & 2v_1^2 d_4^3 & 2v_1 d_4^3 & 2d_4^3 & 2v_1^2 d_4^2 & 2d_4^2 \\ 0 & 0 & v_1^4 d_4^2 & 2v_1^3 d_4^2 & 2v_1^2 d_4^2 & 2v_1^4 d_4 & 2v_1^2 d_4 \\ 0 & 0 & 0 & v_1^2 d_4^2 & 2v_1 d_4^2 & 2v_1^3 d_4 & 2v_1 d_4 \\ 0 & 0 & 0 & 0 & d_4^2 & 2v_1^2 d_4 & 2d_4 \\ 0 & 0 & 0 & 0 & 0 & v_1^4 & 2v_1^2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2.13}$$

Substituting the desired IO function, $d_4 = f(v_1)$, into Equation (2.13) and integrating between the specified input range, the expression used in the minimisation algorithm becomes,

$$\min_{(a_1,a_2,a_4,v_4)\in\mathbb{R}} \left(\begin{bmatrix} A & B & C & -8a_1v_4 & D & E & F \end{bmatrix} \int_{v_{1min}}^{v_{1max}} \mathbf{S}(v_1, f(v_1)) \begin{bmatrix} A \\ B \\ C \\ -8a_1v_4 \\ D \\ E \\ F \end{bmatrix} \right). \quad (2.14)$$

To compare the continuous approximation results with previously published results, a function from [92] will be used,

$$d_4 = \frac{-{v_1}^2 + 1}{{v_1}^2 + 1}, \quad (2.15)$$

between the bounds of $v_1 = -3 \cdots 3$. Upon solving the precision point problem and integrating Equation (2.13) with the desired function substituted for d_4, the squared AIO function may be minimised using the same constraints for the RRRR case, constraining the minimisation procedure strictly to real valued link lengths. From [92], the following linkage parameters were identified following a Newton-Gauss iterative minimisation routine over fifty precision points within the design space,

$$\begin{bmatrix} a_1 \\ a_2 \\ a_4 \\ v_4 \end{bmatrix} = \begin{bmatrix} 0.9426 \\ 1.1587 \\ 1 \\ 1.5 \cdot 10^{-5} \end{bmatrix}, \quad (2.16)$$

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

$$\begin{bmatrix} a_1 \\ a_2 \\ a_4 \\ v_4 \end{bmatrix} = \begin{bmatrix} 0.9554 \\ 1.1894 \\ 1 \\ 1.17 \cdot 10^{-10} \end{bmatrix}, \quad (2.17)$$

again showing a strong agreement between the two methods. Figure 2.2 shows three functions for this RRRP function generator example.

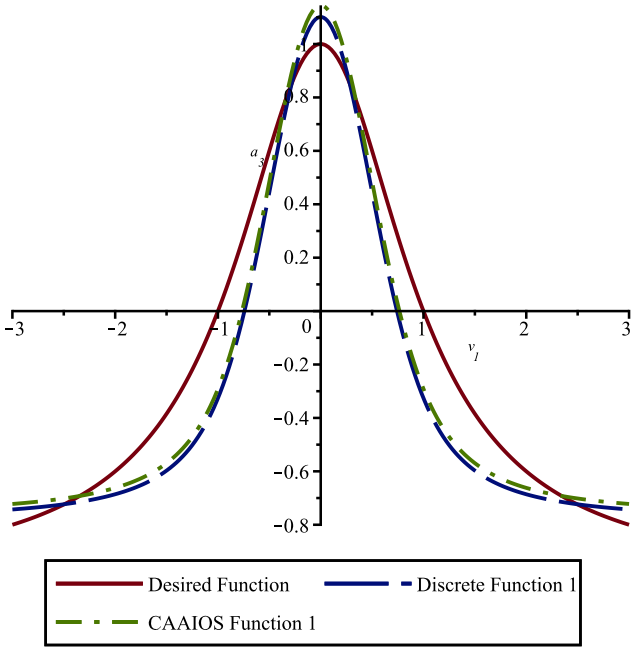


FIGURE 2.2: Comparison of RRRP desired and generated functions.

The percentage error between the desired and generated IO curves are,

$$\text{Discrete Optimisation Method} = 10.504\%, \tag{2.18}$$

$$\text{Continuous Approximate Optimised Function} = 8.766\%, \tag{2.19}$$

showing a reduction in the error by a factor of approximately 19.8% relative to the discrete structural error minimisation routine.

2.3.3 Continuous Approximate Synthesis for the PRRP Function Generator

The PRRP planar function generator illustrated in Figure 1.3 is now considered. Equation (1.90) is squared, yielding Equation (2.20) which is pre- and post-multiplied by an array containing the linkage parameter functions contained in Equation (1.91) to obtain the full squared AIO synthesis equation for the PRRP architecture,

$$\mathbf{S}(a_1, d_4) = \begin{bmatrix} a_1^4 & 2a_1^2 d_4^2 & 2a_1^3 d_4 & 2a_1^3 & 2a_1^2 d_4 & 2a_1^2 \\ 0 & d_4^4 & 2a_1 d_4^3 & 2a_1 d_4^2 & 2d_4^3 & 2d_4^2 \\ 0 & 0 & a_1^2 d_4^2 & 2a_1^2 d_4 & 2a_1 d_4^2 & 2a_1 d_4 \\ 0 & 0 & 0 & a_1^2 & 2a_1 d_4 & 2a_1 \\ 0 & 0 & 0 & 0 & d_4^2 & 2d_4 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2.20}$$

A desired function, which is specified as $d_4 = f(d_1)$ can be substituted into Equation (2.20). After integrating this expression between the desired input range of this function generator, the expression for the minimisation then becomes,

$$\min_{(a_2, a_4, v_1, v_4) \in \mathbb{R}} \left(\begin{bmatrix} A & B & C & D & E & F \end{bmatrix} \int_{a_{1min}}^{a_{1max}} \mathbf{S}(a_1, f(a_1)) \begin{bmatrix} A \\ B \\ C \\ D \\ E \\ F \end{bmatrix} \right). \tag{2.21}$$

Comparing the standard methodologies to the continuous approximate methodology will once again be completed through the use of a test case included in [92]. The desired $d_4 = f(d_1)$ IO function in this case is,

$$d_4 = \cos(d_1), \tag{2.22}$$

over the range of $d_1 = 0 \cdots 2$. 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 synthesis matrix in Equation (2.20) is integrated between the bounds of this function generator problem, and subsequently minimised. The structural error minimising linkage parameters resulting from the non-linear Newton-Gauss optimisation procedure from [92] are,

$$\begin{bmatrix} a_2 \\ a_4 \\ v_4 \\ v_1 \end{bmatrix} = \begin{bmatrix} 2.0313 \\ 1 \\ -1.1868 \\ 0.1353 \end{bmatrix}, \tag{2.23}$$

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

$$\begin{bmatrix} a_2 \\ a_4 \\ v_4 \\ v_1 \end{bmatrix} = \begin{bmatrix} 2.0364 \\ 1 \\ -1.1986 \\ 0.1291 \end{bmatrix}, \tag{2.24}$$

showing again, a large degree of agreement between the two. The results of the CAAIOS are illustrated in Figure 2.3.

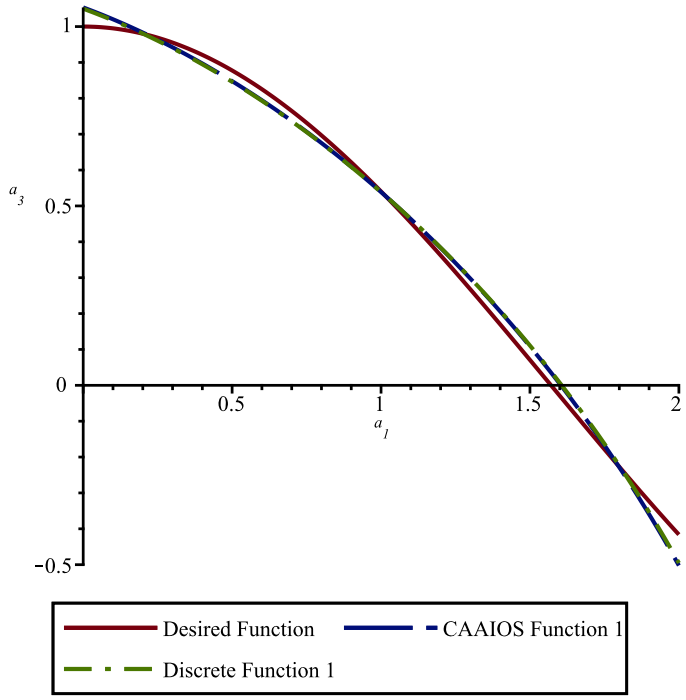


FIGURE 2.3: Comparison of PRRP desired and generated functions.

Once again, these generated 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,

$$\text{Precision Point Method} = 0.146\%, \tag{2.25}$$

$$\textit{Continuous Approximate Method} = 0.054\%, \tag{2.26}$$

showing a reduction in the error of approximately 168.31% relative to the precision point solution error.

2.4 Non-Planar Function Generating Linkage Architectures

While the CAAIOS algorithm has been shown to be operational without loss in generality for all planar four-bar function-generating linkage architectures, its applicability and any modifications to the cases of non-planar four bar function-generating linkages will be examined in this section. For the purposes of this discussion, first, the spherical RRRR linkage architecture will be discussed, followed by the spatial RSSR linkage architecture.

2.4.1 Spherical RRRR CAAIOS Function Generator Synthesis

The spherical RRRR function generator is now considered. Due to the projective similarities between the geometries of the linkages, the spherical RRRR linkage CAAIOS methods and the equations which comprise them are nearly identical to the planar RRRR function generating linkage, with the synthesis matrix, **S**, being identical. The only variation between these two formulations is found within the linear term inside of the parameter array, and thus,

$$\begin{bmatrix} A & B & C & D & E \end{bmatrix} \mathbf{S}(v_1, v_4) \begin{bmatrix} A \\ B \\ C \\ D \\ E \end{bmatrix} = 0, \tag{2.27}$$

where E is a factor which is bi-quadratic in link arclengths α_4 and α_2, recall Equation (1.103), which is also bilinear in the remaining link arclengths, α_1 and α_3,

$$E = 8\alpha_1\alpha_3(\alpha_4^2 + 1)(\alpha_2^2 + 1). \tag{2.28}$$

The synthesis matrix, **S**, may then be defined as,

$$\mathbf{S}(v_1, v_4) = \begin{bmatrix} v_1^4 v_4^4 & 2v_1^4 v_4^2 & 2v_1^2 v_4^4 & 2v_1^2 v_4^2 & 2v_1^3 v_4^3 \\ 0 & v_1^4 & 2v_1^2 v_4^2 & 2v_1^2 & 2v_1^3 v_4 \\ 0 & 0 & v_4^4 & 2v_4^2 & 2v_1 v_4^3 \\ 0 & 0 & 0 & 1 & 2v_1 v_4 \\ 0 & 0 & 0 & 0 & v_1^2 v_4^2 \end{bmatrix}. \tag{2.29}$$

The $v_1 - v_4$ synthesis matrix for the spherical RRRR function generator is identical to the synthesis matrix for the $v_1 - v_4$ planar RRRR function generator shown in Equation (2.3). For this example, we will also be approximating the same function as in the planar RRRR function generator example, in order to reveal some notable differences in the way results are presented in the case of the spherical mechanism. That is to say, the function to approximate is,

$$v_4 = f(v_1) = 2 + tan\left(\frac{v_1}{v_1^2 + 1}\right), \tag{2.30}$$

where, once again, $v_1 = 0 \cdots 2$. First, the precision point problem will be solved. Just as with the planar RRRR function generating linkage, we will arbitrarily set $\alpha_4 = 1$. The three precision points we will choose for the approximation are, $v_1 = 0, 1$, and 2. The resulting precision point synthesis arclengths will be used as an initial guess;

$$\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix} = \begin{bmatrix} -0.048 \\ -0.498 \\ 0.788 \\ 1 \end{bmatrix}. \tag{2.31}$$

Now, when the solution to the precision point problem with the spherical RRRR function generator is computed, the designer will note that multiple sets of precision point solutions will exist. Within this specific example, Maple identified ten sets of solutions, eight of which produce viable linkages, two of which are the trivial solution of $\alpha_1 = \alpha_3 = 0$ and $\alpha_2 = \pm\alpha_4$. This increase in the number of solutions is due to the fact that the mechanism itself lies on the surface of a sphere. This implies that, provided the revolute centers are located in the same locations on the surface of the sphere, it does not matter if the link wraps around the entire circumference of the sphere. Thus, for the case of the spherical RRRR function generator, it is typically recommended that the designer select the shortest set of link arclengths. The resulting IO function is shown in

Figure 2.4.

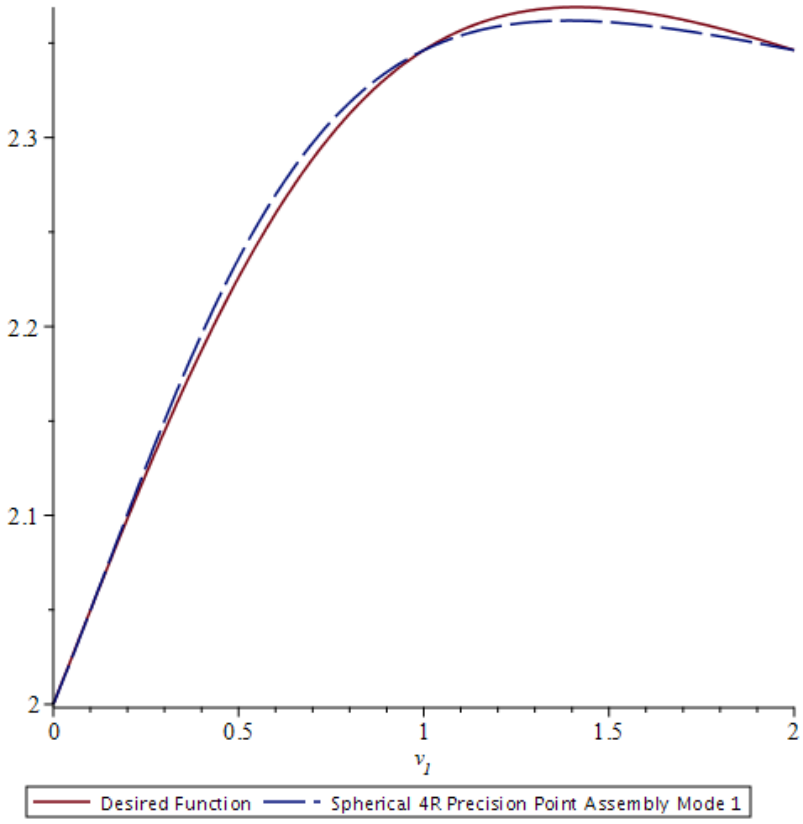


FIGURE 2.4: The results following the three point precision point solution for the spherical RRRR linkage architecture.

Upon selection of the initial guess for the arclengths of this spherical RRRR function generator, one can substitute Equation (2.30) into matrix **S** and minimise the resulting expression such that,

$$\min_{(\alpha_1,\alpha_2,\alpha_3,\alpha_4)\in\mathbb{R}} \left(\begin{bmatrix} A & B & C & D & E \end{bmatrix} \int_{v_{1min}}^{v_{1max}} \mathbf{S}(v_1, f(v_1)) \begin{bmatrix} A \\ B \\ C \\ D \\ E \end{bmatrix} \right). \tag{2.32}$$

Following this minimisation, the following set of optimal linkage arclengths are determined to be,

$$\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix} = \begin{bmatrix} -0.049 \\ -0.500 \\ 0.776 \\ 1 \end{bmatrix}. \tag{2.33}$$

The comparison between the functions associated with the precision point solutions contained in Equation (2.31) and the CAAIOS solution in Equation (2.33) are plotted in Figure 2.5.

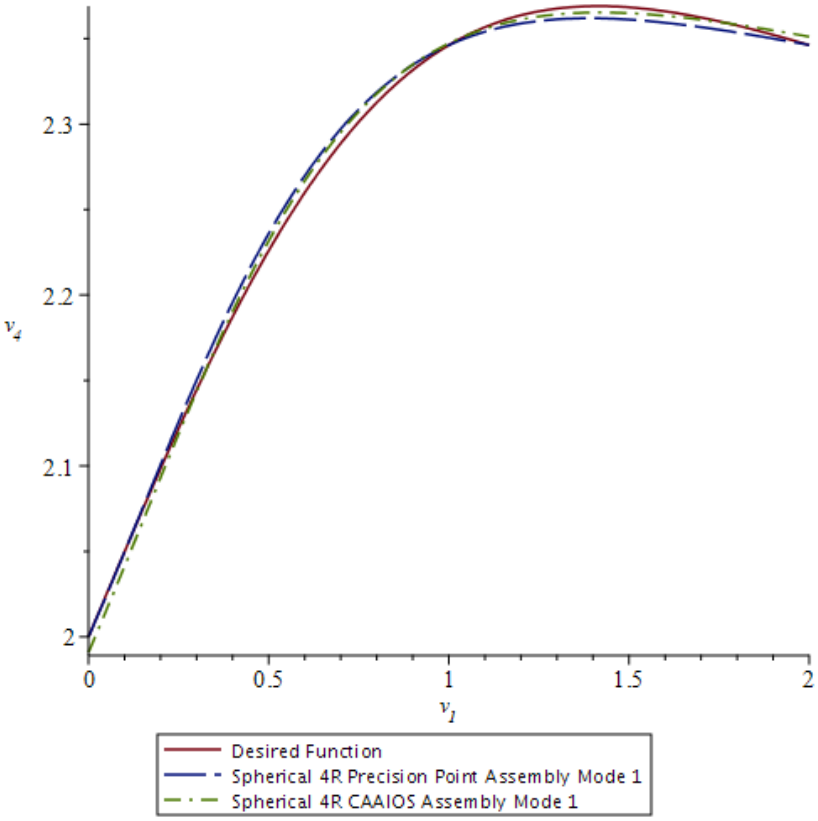


FIGURE 2.5: The results following the CAAIOS algorithm implementation for the spherical RRRR linkage architecture.

While this function is, itself, quite well behaved with the precision point solution, the CAAIOS solution does clearly generate a function which lies in closer proximity to the desired function, which is particularly notable as v_1 increases past 1. Upon integration of these three functions between the bounds used for the optimisation, the error percentages may be computed,

$$\text{Precision Point Method} = 0.000145\%, \tag{2.34}$$

$$\text{Continuous Approximate Method} = 0.000000789\%. \tag{2.35}$$

While only a modest improvement in error, as expressed as the difference in area underlying the curves, is realised in the absolute sense, the relative error is several orders of magnitude lower in the CAAIOS method.

2.4.2 Spatial RSSR Function Generating Linkage

Finally, we will conclude the demonstrations of the CAAIOS algorithm with the only spatial mechanism for which an AIO equation has been derived: the spatial RSSR mechanism. For review, the algebraic input-output equation from Chapter 1 is reproduced in Equation (2.36), while its geometry is shown again in Figure 2.6. Please note that, due to the algebraic representation via tangent half angle parameters, the angular parameters (θ_1, θ_8) represent the IO pair and are expressed in Equation (2.36) as (v_1, v_8), while the parameter τ_8 is represented as α_8.

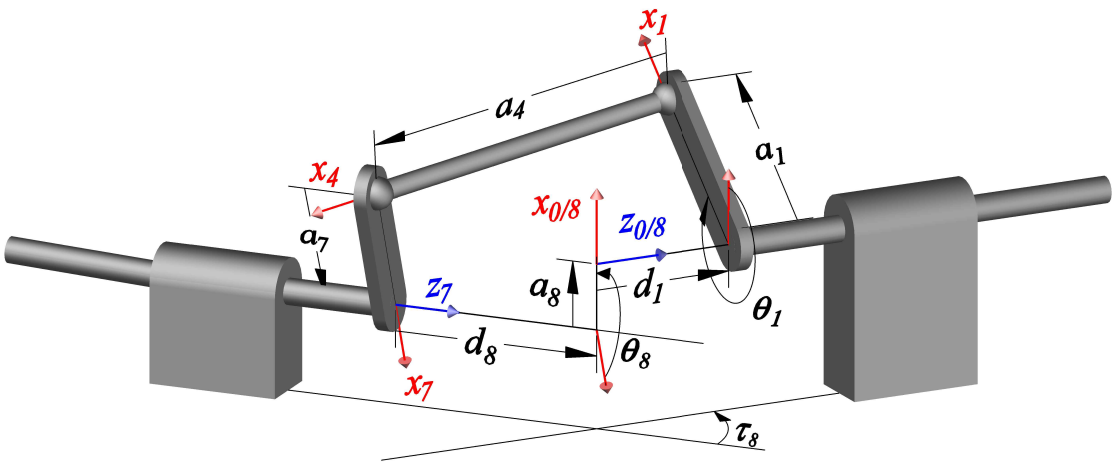


FIGURE 2.6: The kinematic geometry of the RSSR linkage, will relevant joint lengths and angles labelled for DH method parametrisation.

$$Av_1^2v_8^2 + 8d_1\alpha_8a_7v_1^2v_8 + 8d_8\alpha_8a_1v_1v_8^2 + Bv_1^2 + 8a_1a_7(\alpha_8 - 1)(\alpha_8 + 1)v_1v_8 + Cv_8^2 + 8d_8\alpha_8a_1v_1 + 8d_1\alpha_8a_7v_8 + D = 0, \quad (2.36)$$

Despite the spatial nature of the linkage in question, the set-up of the CAAIOS algorithm remains identical to all previously mentioned cases. First, square the AIO, second, separate all linkage parameters and input-output parameters into an array and matrix, respectively. The parameter vector associated with the RSSR linkage is,

$$\mathbf{p}_{RSSR} = \begin{bmatrix} A \\ 8d_1\alpha_8a_7 \\ 8d_8\alpha_8a_1 \\ B \\ 8a_1a_7(\alpha_8 - 1)(\alpha_8 + 1) \\ C \\ 8d_8\alpha_8a_1 \\ 8d_1\alpha_8a_7 \\ D \end{bmatrix}, \quad (2.37)$$

while the matrix associated with this function generating architecture is,

$$\mathbf{S} = \begin{bmatrix} v_1^4 v_8^4 & 2v_1^4 v_8^3 & 2v_1^3 v_8^4 & 2v_1^4 v_8^2 & 2v_1^2 v_8^4 & 2v_1^3 v_8^3 & 2v_1^3 v_8^2 & 2v_1^2 v_8^3 & 2v_1^2 v_8^2 \\ 0 & v_1^4 v_8^2 & 2v_1^3 v_8^3 & 2v_1^4 v_8 & 2v_1^2 v_8^3 & 2v_1^3 v_8^2 & 2v_1^3 v_8 & 2v_1^2 v_8^2 & 2v_1^2 v_8^1 \\ 0 & 0 & v_1^2 v_8^4 & 2v_1^3 v_8^2 & 2v_1 v_8^4 & 2v_1^2 v_8^3 & 2v_1^2 v_8^2 & 2v_1 v_8^3 & 2v_1 v_8^2 \\ 0 & 0 & 0 & v_1^4 & 2v_1^2 v_8^2 & 2v_1^3 v_8 & 2v_1^3 & 2v_1^2 v_8 & 2v_1^2 \\ 0 & 0 & 0 & 0 & v_1^4 v_8^4 & 2v_1^4 v_8^4 & 2v_1^4 v_8^4 & 2v_1^4 v_8^4 & 2v_1^4 v_8^4 \\ 0 & 0 & 0 & 0 & v_8^4 & 2v_1 v_8^3 & 2v_1 v_8^2 & 2v_8^3 & 2v_8^2 \\ 0 & 0 & 0 & 0 & 0 & v_1^2 v_8^2 & 2v_1^2 v_8 & 2v_1 v_8^2 & 2v_1 v_8 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & v_8^2 & 2v_8 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2.38}$$

For ease of implementation, we will also be generating the same function mechanism as the planar and spherical RRRR function generating mechanisms, such that the desired input-output relationship may be expressed as,

$$v_8 = 2 + \tan\left(\frac{v_1}{v_1^2 + 1}\right). \tag{2.39}$$

Due to the number of the free parameters associated with the RSSR function generating mechanism, however, we will elect to choose six precision points, while arbitrarily setting the input-output revolute joint offset parameter, a_8, equal to 1. However, unlike with the planar and spherical RRRR the designer must exercise some caution in the selection of the arbitrary scaling factor for the RSSR linkage; if a naive designer were to select α_8 as their fixed parameter, they would necessarily be fixing the geometry of the linkage itself, *not simply the scale of the linkage*.

In addition to this fact, the number of parameters associated with the RSSR function generator allows it to, in general, produce far more accurate representation of any given IO relationship. For the sake of this analysis, this implies that the input angular range over which this function must be synthesised in order to induce some relatively large error within even the precision point solution to the problem. Thus, the range over which this function will be synthesised is larger than within the planar or spherical

RRRR synthesis examples, $v_1 = -1 \ldots 3$. Arbitrarily, we choose six precision points in v_1 over this range. The resulting precision point solution set is,

$$\begin{bmatrix} a_1 \\ a_4 \\ a_7 \\ a_8 \\ d_1 \\ d_8 \\ \alpha_8 \end{bmatrix} = \begin{bmatrix} -0.2663 \\ -2.2645 \\ 1.3538 \\ 1 \\ -4.2038 \\ 2.6429 \\ 0.4953 \end{bmatrix}, \tag{2.40}$$

where $a_8 = 1$ was chosen as the fixed link length for this demonstration. Figure 2.7 shows the resulting synthesised function on the interval of $v_1 = -1 \ldots 3$.

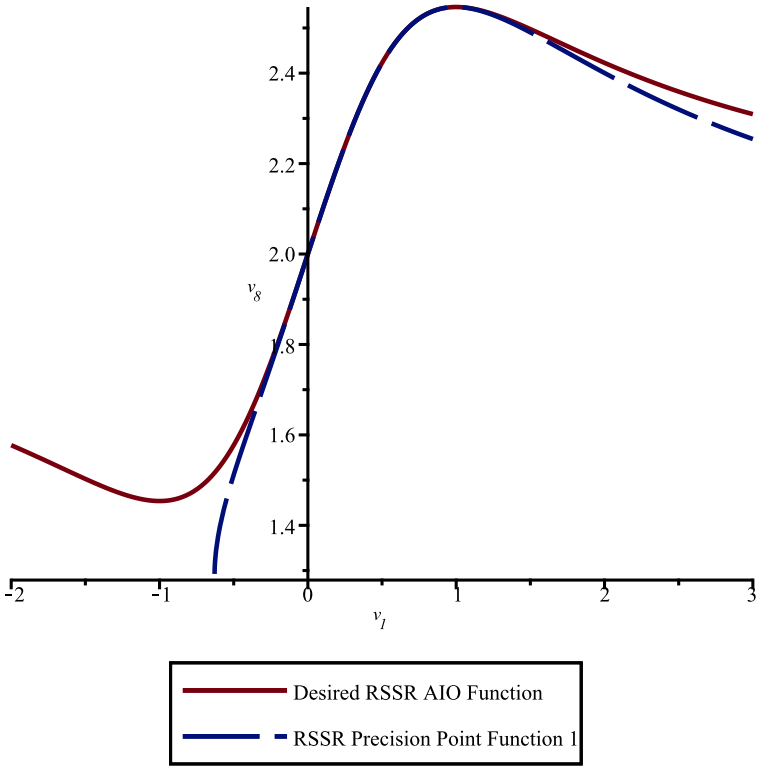


FIGURE 2.7: The results following the three point precision point solution for the spatial RSSR linkage architecture.

Now the RSSR CAAIOS algorithm may be initialised with the initial guesses presented in Equation (2.40) such that,

$$\min_{(a_1,a_4,a_7,d_1,d_8,\alpha_8)\in\mathbb{R}} \left(\mathbf{p}_{RSSR}^T \int_{v_{1min}}^{v_{1max}} \mathbf{S}_{RSSR}(v_1, f(v_1))\Delta v_1 \mathbf{p}_{RSSR} \right). \tag{2.41}$$

Following the minimisation routine, the CAAIOS optimal linkage parameters are identified as,

$$\begin{bmatrix} a_1 \\ a_4 \\ a_7 \\ a_8 \\ d_1 \\ d_8 \\ \alpha_8 \end{bmatrix} = \begin{bmatrix} -0.1989 \\ -2.2683 \\ 1.4113 \\ 1 \\ -3.7761 \\ 2.5701 \\ 0.6534 \end{bmatrix}, \tag{2.42}$$

Once the optimal linkage parameters are identified, the function generated by the CAAIOS linkage dimensions is plotted alongside the solutions from the precision point method, and the original target function in Figure 2.8.

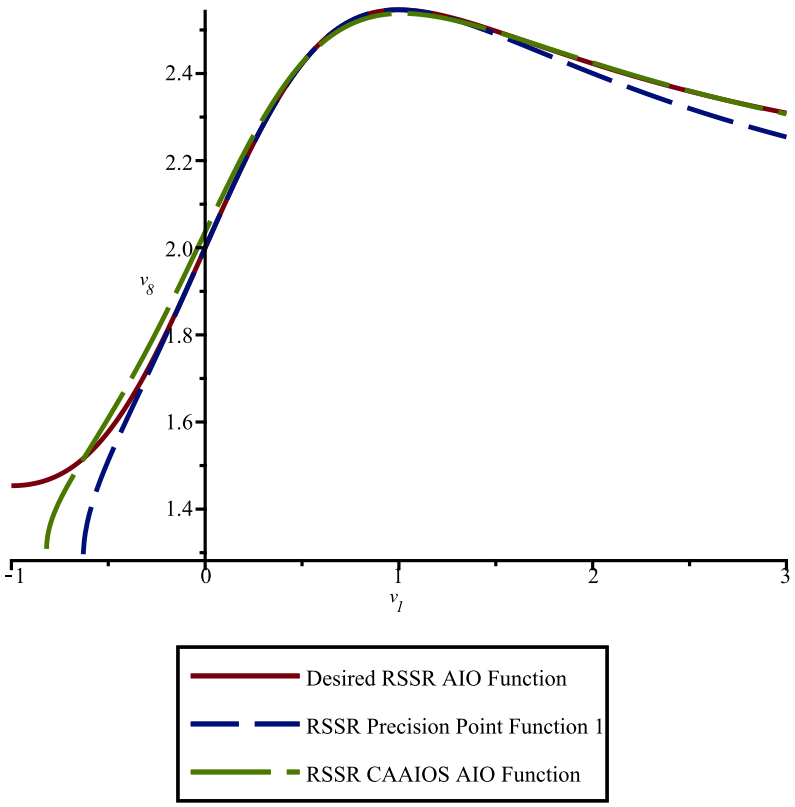


FIGURE 2.8: The results following the CAAIOS algorithm implementation for the spatial RSSR linkage architecture.

Clearly, the CAAIOS derived function lies in far closer proximity to the target function than the precision point method function over the range for which the linkage was optimised.

2.5 Computational Considerations

2.5.1 Optimisation Formulation

While the matrix-array formulation of this problem is convenient, as it offers a very concise and easily understood demonstration of the form of the algorithm and the squared IO function, the implementation of a matrix for numerical integration necessarily requires a nested *for* loop in order to be properly implemented within Maple. However, this matrix-array formulation is not the only representation which is useful for the function generator dimensional synthesis optimisation; a representation that does not require the user to pre-define the matrix of squared IO variables may also be realised by simply splitting the squared algebraic IO equation into variables and coefficients in two arrays. In the interest of brevity, only one such case will be presented, the planar RRRR function generator. The RRRR function generator will be presented in terms of the linkage parameter array, $\mathbf{p}_A$, and the linkage synthesis array, $\mathbf{s}_A$. In full, these vectors take the form of,

$$\mathbf{p}_A = \begin{bmatrix} A^2 \\ 2\,AB \\ B^2 \\ -16\,Aa_1\,d_4 \\ -16\,Ba_1\,d_4 \\ 2\,AC \\ 64\,{a_1}^2{d_4}^2 + 2\,AD + 2\,BC \\ 2\,BD \\ -16\,Ca_1\,d_4 \\ -16\,Da_1\,d_4 \\ C^2 \\ 2\,CD \\ D^2 \end{bmatrix}, \tag{2.43}$$

and,

$$\mathbf{s}_A(v_1, v_4) = \begin{bmatrix} {v_1}^4{v_4}^4 \\ {v_1}^4{v_4}^2 \\ {v_1}^4 \\ {v_1}^3{v_4}^3 \\ {v_1}^3 v_4 \\ {v_1}^2{v_4}^4 \\ {v_1}^2{v_4}^2 \\ {v_1}^2 \\ v_1\,{v_4}^3 \\ v_1\,v_4 \\ {v_4}^4 \\ {v_4}^2 \\ 1 \end{bmatrix}, \tag{2.44}$$

such that,

$$\begin{bmatrix} A = A_1 A_2 \\ B = B_1 B_2 \\ C = C_1 C_2 \\ D = D_1 D_2 \\ -8a_1 d_4 \end{bmatrix} = \begin{bmatrix} (a_1 - a_2 + a_3 - a_4)(a_1 + a_2 + a_3 - a_4) \\ (a_1 + a_2 - a_3 - a_4)(a_1 - a_2 - a_3 - a_4) \\ (a_1 - a_2 - a_3 + a_4)(a_1 + a_2 - a_3 + a_4) \\ (a_1 + a_2 + a_3 + a_4)(a_1 - a_2 + a_3 + a_4) \\ -8a_1 a_3 \end{bmatrix}. \tag{2.45}$$

Subsequently, the integration of the array containing the linkage angular parameters proceeds identically to the previous version with the matrix-array representation. However, considering Equations (2.44) and (2.45) it simplifies to,

$$\min_{(a_1,a_2,a_3,a_4)\in\mathbb{R}} \left(\mathbf{p}_A \cdot \int_{v_{1min}}^{v_{1max}} \mathbf{s}_A(v_1, f(v_1)) \right), \tag{2.46}$$

while all other planar four-bar function generators have a near identical representation, differing only by the input-output pair contained in the minimisation of $\mathbf{s}_A$. This representation, referred to as the *CAAIOS* equation, not only allows for the elimination of the nested *for* loop that was required for integrating and writing values into an empty synthesis matrix, but allows automation in the separation of the linkage parameter vector, $\mathbf{p}_A$, and the synthesis vector, $\mathbf{s}_A(v_i, v_j)$. Furthermore, while the simplification of this algorithm by hand may be decidedly time consuming, especially for the spherical RRRR and spatial RSSR linkages, this process is expedited dramatically via software such as Maple 2023; if one squares the AIO equation (without expanding the coefficients), Maple may be used to automatically collect the coefficients and the IO relationship in separate vectors.

2.5.2 Numerical Sensitivity, Floating Point Values, and the Initial Guess

During the course of the research presented herein, it was seen that the AIO equations are decidedly sensitive to rounding errors and numerical resolution. Results of the continuous approximate algebraic input-output synthesis and all preceding precision point solutions are presented herein with floating point values. However, the computations within Maple *must* be completed using rational representations of numbers. If this is not explicitly the case, the simultaneous solutions required for the precision point solution

may not be obtainable, and moreover, the CAAIOS algorithm may yield only the trivial solutions as viable sets of link lengths.

Optimisation problems are, in general, sensitive to the solution set used as an initialisation point for the solver. While it would be convenient to be able to use a single default solution set for the CAAIOS algorithm, attempting to do so in the general case leads to the only solution to the optimisation algorithm being the trivial solution, or to the minimisation algorithm being unable to converge before its maximum iterations are reached. This fact gives rise to the methodology presented within the previous sections of this thesis, wherein which the precision point method must be used in order to generate an initial guess which is suitably close to the guess which will result from the CAAIOS algorithm. If the minimisation is initialised with an initial guess which simply contains general placeholder values for a given linkage architecture, the minimisation algorithm will, in general, *not proceed* to a local minimum which is suitable for the application at hand.

Additionally, during the testing of this algorithm it was found that the only time where the precision point method fails to produce a solution is when the desired function to be synthesised in $v_j = f(v_i)$ is symmetric about the v_j axis, *and* the designer chooses precision points which are symmetric about the same axis. This symmetry causes a linear dependency between the equations, but this linear dependency evaporates if either condition is altered or not realised in the first place. Thus, if this ever occurs during the solution to the precision point problem, the designer must simply choose a slightly different precision point set that is not symmetric about the v_j axis.

For example, if the RSSR function generating linkage example stated previously is executed with a set of symmetric precision points such as $v_1 = \pm 1$, the precision point method will not be capable of identifying a valid linkage parameter set for initialising the CAAIOS minimisation routine. The designer could use a method such as Chebyshev point distribution algorithms [106] in order to avoid this in general, though it should always remain a point to check in the event that the precision point method fails to generate a real solution, or fails to resolve at all.

3 Extensions to Continuous Approximate Algebraic IO Synthesis

3.1 Combined Type and Dimensional Synthesis for Planar Four-Bar Function Generators

Typically, when a function generator is to be optimised, the first step is to choose the linkage architecture that will be used to generate it; this step can either be informed through the subject matter expertise of the designer, or through mechanical considerations associated with the application. However, it is possible that the linkage architecture which was chosen is not necessarily the globally optimal planar four-bar linkage for the generation of this function over the desired range. For academic considerations, it will be supposed that the designer has the increasingly rare privilege of complete design control for the purposes of this discussion. 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. Given a desired function, 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.

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

$$h = \frac{1 - t^2}{1 + t^2}, \tag{3.1}$$

such that $t = 0 \cdots 1$, with variable substitutions shown in Table 3.1.

Variable	**Linkage Architecture**		
General Case	*4R*	*RRRP*	*PRRP*
t	v_1	v_1	d_1
h	v_4	d_4	d_4

TABLE 3.1: Variable substitutions used for continuous approximate type and dimensional planar function generator synthesis.

Following the procedure as previously described, each mechanism type may be synthesised and compared to each other. For the sake of brevity, Table 3.2 lists the three mechanisms with the optimal linkage parameters identified via the CAAIOS algorithm.

Linkage Parameter	**Linkage Architecture**		
	RRRR	*RRRP*	*PRRP*
v_1	v_1	v_1	1.995
v_4	v_4	0.0903	0.5114
a_1 *or* d_1	2.6×10^{-5}	0.7288	d_1
a_2	0.9999	1.247	1.707
a_3 *or* d_4	2.3×10^{-5}	d_4	d_4
a_4	1	1	1

TABLE 3.2: All identified parameters for the continuous approximate concurrent type and dimensional synthesis of a planar function generator.

Once each set of error-minimising linkage parameters has been computed, the explicit IO functions that each of these mechanisms generate are computed. These three functions are plotted in Figure 3.1.

At first glance, it would appear that the RRRR function generating architecture is, by far, the best suited for this approximation. However, a_1 and a_3, the input and output link lengths, are five orders of magnitude smaller than the computed a_2 and a_4, the coupler and base link length. The parameters for the RRRP and PRRP linkages are all reasonable values, but the RRRR linkage values are peculiar. This strange outcome is a result of the fact that the solver is constrained so that $a_i^2 \geq 0$, meaning that these values can be identically equal to zero, or as close to zero as the floating point accuracy of the computer being used for the minimisation algorithm allows. Indeed, $a_1 = a_3 = 0$ and $a_2 = a_4 = 1$ is a consistent trivial solution to any planar function generation problem

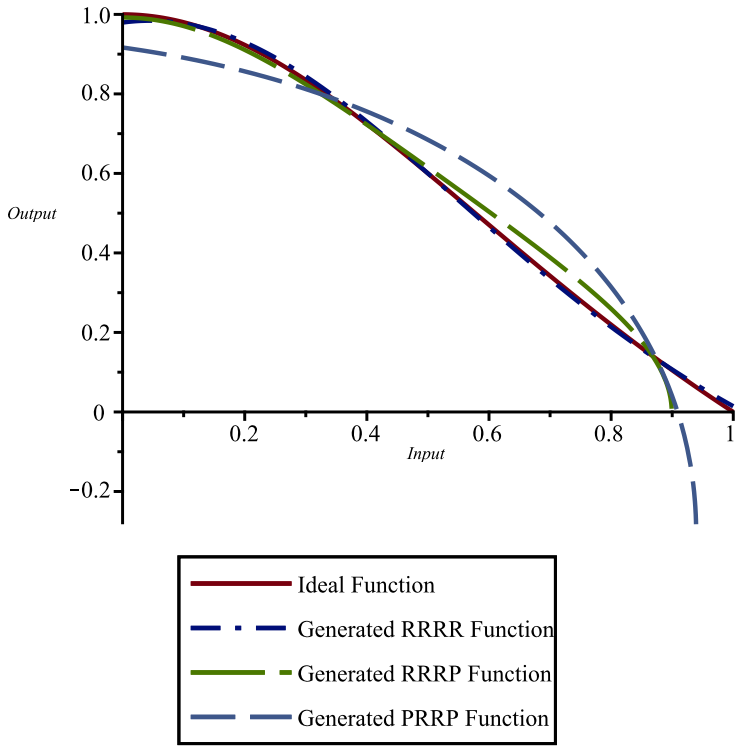


FIGURE 3.1: Comparison of 4R, RRRP, and PRRP desired and generated functions.

with the RRRR kinematic architecture and must be discarded as a useful result. Given this fact, the comparison of the RRRR function generator will be omitted in the subsequent analysis. Table 3.3 shows the percentage error relative to the desired function of each of the polynomials generated by the RRRP and PRRP linkages, respectively.

	Generated RRRP	**Generated PRRP**
Percentage Difference	0.0143%	0.3685%

TABLE 3.3: Percentage error for all viable planar four-bar function generators.

From both Figure 3.1 and Table 3.3, it is clear that the RRRP function generator is, indeed, the most well suited of the planar four-bar architectures to generate the desired function. Now, this answer may seem relatively obvious, as the equation that was being generated is identical to the equation presented in Equation (2.15), 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.

3.2 Multi-Modal Continuous Approximate Synthesis

In the previous sections of this thesis, the algorithms for continuous approximate type and dimensional synthesis have been presented. Now these concepts will be combined and used to extend the theory already described to what will be referred to as *multi-modal function generator synthesis*. Throughout the derivation of the various AIO equations for planar four-bar function generators, some modifications to the typical perspective on IO theory were examined in order to determine whether or not the additional coupled IO pairs (the aforementioned v_i, v_j) equations have equivalent algebraic representations to the classically examined IO function generators. Typically, within four-bar IO function generator synthesis, the designer is most concerned with the $\theta_1 - \theta_4$ IO pair relationship; however, from Figure 3.2 one may consider that the $\theta_1 - \theta_3$ IO pair is of interest, or any other pair of angles. This extension and specific example results from subsequent analyses for the kinetic and dynamic properties of a linkage once the kinematic synthesis process has been completed, most notably the analysis of the transmission angle.

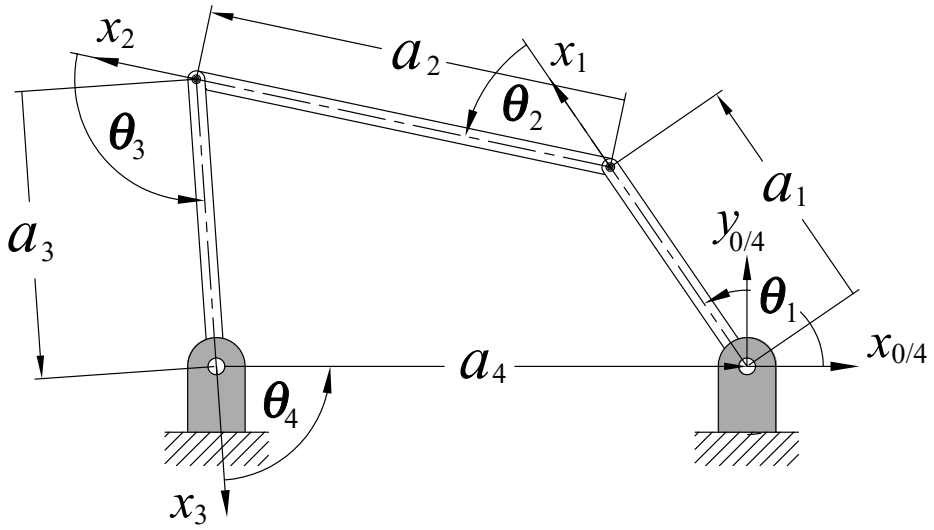


FIGURE 3.2: A general planar 4R function generator.

Multi-modal IO synthesis is the problem of optimising a given linkage architecture to simultaneously satisfy two functions between different IO joint pairs of the same linkage. These IO function generators may be designed so as to approximate their functions over different angular ranges, depending on the needs or choices of the designer. Extension of the current planar four-bar function generation algorithm to generating multiple functions between different pairs of joint angles relies heavily on the principles outlined in the CAAIOS algorithm. The proposed steps are as follows:

1. Identify the two pairs of joint parameter IO functions that will be the subject of the following design loop.
2. Identify the ranges over which the desired IO functions will be approximated.
3. Use the previously defined CAAIOS algorithm to develop the IO function generator which best approximates **one** of the IO functions.
4. Integrate both synthesis arrays from the respective CAAIOS equations over the prescribed ranges of the approximation for each IO function in order to define their CAAIOS equations.
5. Sum these two CAAIOS equations.
6. Using the optimal linkage parameters identified in the third step of this algorithm as an initial guess, minimise the sum of these IO equations over the field of real numbers.

3.2.1 Mathematical Implementation of Multi-Modal Continuous Approximate Synthesis

This section will establish the mathematical basis for this operation, while following sections will be used to develop examples, and establish limitations to the implementation of the multi-modal continuous approximate algebraic input-output synthesis (MMCAAIOS) methods. First, it is proposed that the MMCAAIOS process may be accomplished with the following equation:

$$\min_{(p_1 \cdots p_n) \in \mathbb{R}} \left(\mathbf{c_1} \cdot \int_{v_{i_{1_{\min}}}}^{v_{i_{1_{\max}}}} \mathbf{s_1}(v_{i_1}, f_1(v_{i_1}))dv_{i_1} + \mathbf{c_2} \cdot \int_{v_{i_{2_{\min}}}}^{v_{i_{2_{\max}}}} \mathbf{s_2}(v_{i_2}, f_2(v_{i_2}))dv_{i_2} \right) = 0; \tag{3.2}$$

where $\mathbf{p_i}$ is used to denote the parameters of any given four-bar function generating mechanism architecture; $\mathbf{c_1}$ is the array of coefficient terms associated with the primary function being synthesised; $\mathbf{s_1}$ is the synthesis array associated with the primary function being synthesised; $v_{i_{1_{min}}}$ and $v_{i_{1_{max}}}$ represent the lower and upper bounds of the synthesis range associated with the input angular parameter of the first function, respectively; $\mathbf{c_2}$ is the constant terms associated with the secondary function being synthesised; $\mathbf{s_2}$ is the synthesis array associated with the secondary function being synthesised; and $v_{i_{2_{min}}}$ and $v_{i_{2_{max}}}$ represent the lower and upper bounds of the synthesis range associated with the input angular parameter of the secondary function, respectively.

The application of the MMCAAIOS methodology is identical to that of the typical CAAIOS from this point. Now that the mathematical relationship for the MMCAAIOS algorithm has been defined, the limitations of its applicability must be elucidated.

3.2.2 Planar RRRR Multi-Modal Function Generation

The typical function generation problem concerns $\theta_4 = f(\theta_1)$ and the corresponding v_1-v_4 IO equation; however, considering Figure 3.2, one may wish to also consider the v_1-v_3 pair of angles, or any other of the remaining four pairs. For this *proof-of-concept* of the multi-modal continuous approximate synthesis method we shall begin with the synthesis of two arbitrarily competing functions, $v_4 = f_1(v_1)$ and $v_3 = f_2(v_1)$. The reason for this choice is that the v_3 angle parameter is a measure of the transmission angle, which can useful as a metric to discriminate between four bar mechanisms that have practical use from those that do not [35, 36, 37, 38, 39].

This idea has a philosophical existential question associated with it. Namely, when a mechanism is identified to generate, for example, $v_4 = f_1(v_1)$, the five other $v_j = f_2(v_i)$ functions are explicitly defined. Suppose a $v_3 = f_2(v_1)$ function was needed that was different from the one imposed by the initially generated $v_4 = f_1(v_1)$ function. The question now becomes, "does a linkage exist that is the best compromise between the

competing prescribed functions?" The answer is, in general, no. However, it will be shown that polynomial interpolants [107] or heavily constrained target functions can be used to perturb one of the functions and the multi-modal synthesis algorithm may succeed. The design parameter space of planar RRRR function generator linkages is defined [37, 108, 109] as the four-dimensional homogeneous space spanned by the mutually orthogonal basis vectors $\mathbf{a}_1,\ \mathbf{a}_2,\ \mathbf{a}_3$, normalised with respect to frame length $a_4 = 1$. Distinct points in this homogeneous space, $(a_1 :\ a_2 :\ a_3 :\ 1)$, where the delimiter : has been used to indicate the use of homogeneous coordinate ratios, represent distinct planar RRRR linkages. Each point is a linkage that generates six distinct functions between the six distinct angle parings between different links. The linkages identified to generate the prescribed $v_4 = f_1(v_1)$ and $v_3 = f_2(v_1)$ functions represent two distinct points, and therefore two distinct linkages. It will be illustrated in Section 3.2.2 that, in general, the synthesis of competing functions is not possible in any useful way. However, in Section 3.2.5 it will be shown that it is possible to subtly perturb one of the functions used for the mechanism synthesis, leading to useful results.

First, the CAAIOS will be completed for a function in the v_1-v_4 parameter set in order to identify all generated functions from the ideal linkage as a point of comparison for the subsequent MMCAAIOS implementation demonstration.

Now, let the prescribed $v_4 = f_1(v_1)$ function be,

$$v_4 = f_1(v_1) = 2 + \tan\left(\frac{v_1}{{v_1}^2 + 1}\right), \quad -\frac{1}{2} \leq v_1 \leq 2. \tag{3.3}$$

Next, identify the linkage that will approximately generate this function using continuous approximate synthesis. The first step is to square Equation (1.65), then separate the link length coefficients into arrays $\mathbf{c}_1$ and $\mathbf{s}_1$, yielding,

$$\mathbf{c}_1 = \begin{bmatrix} A^2 \\ 2AB \\ B^2 \\ -16Aa_1a_3 \\ -16Ba_1a_3 \\ 2AC \\ 64a_1^2a_3^2 + 2AD + 2BC \\ 2BD \\ -16Ca_1a_3 \\ -16Da_1a_3 \\ C^2 \\ 2CD \\ D^2 \end{bmatrix}, \quad \mathbf{s}_1 = \begin{bmatrix} v_1^4v_4^4 \\ v_1^4v_4^2 \\ v_1^4 \\ v_1^3v_4^3 \\ v_1^3v_4 \\ v_1^2v_4^4 \\ v_1^2v_4^2 \\ v_1^2 \\ v_1v_4^3 \\ v_1v_4 \\ v_4^4 \\ v_4^2 \\ 1 \end{bmatrix} = \begin{bmatrix} v_1^4f_1(v_1)^4 \\ v_1^4f_1(v_1)^2 \\ v_1^4 \\ v_1^3f_1(v_1)^3 \\ v_1^3f_1(v_1) \\ v_1^2f_1(v_1)^4 \\ v_1^2f_1(v_1)^2 \\ v_1^2 \\ v_1f_1(v_1)^3 \\ v_1f_1(v_1) \\ f_1(v_1)^4 \\ f_1(v_1)^2 \\ 1 \end{bmatrix}. \tag{3.4}$$

The exact synthesis problem is then solved in order to obtain an initial guess for the optimisation, using the prescribed function pairs that satisfy Equation (3.3):

$$(v_1,\ v_4) = \left(-\frac{1}{2},\ \frac{32287}{20471}\right);\ \left(\frac{3}{4},\ \frac{49597}{20471}\right);\ \left(2,\ \frac{48857}{19383}\right).$$

Note that to obtain these three precision IO pairs, the lower and upper bounding values of the synthesis range for v_1 have been selected, alongside an arbitrary value in between, while the corresponding value of v_4 satisfies the prescribed function, Equation (3.3).

In this classic RRRR exact synthesis problem a unique solution which contains the link length a_4 as a free parameter may be obtained as:

$$a_1 = -\frac{21111}{109000}a_4,\ a_2 = \frac{21021}{18196}a_4,\ a_3 = \frac{21518}{15263}a_4,\ a_4 = a_4. \tag{3.5}$$

Arbitrarily, one sets $a_4 = 1$ and evaluates the integral, then minimises the design error residual using the normalised link lengths in Equation (3.5) as the initial guess:

$$\min_{(a_1,a_2,a_3,a_4)\in\mathbb{R}} \left(\mathbf{c}_1 \cdot \int_{v_1=-\frac{1}{2}}^{v_1=2} \mathbf{s}_1(v_1, f_1(v_1))\right). \tag{3.6}$$

The minimisation is accomplished using the `Optimization` solvers in Maple 2023, as with all previously noted examples, which converges to the link lengths listed in

Table 3.4.

TABLE 3.4: Continuous approximate synthesis results generating Equation (3.3).

Link length	a_1	a_2	a_3	a_4
Rational	$-\frac{13077}{78259}$	$\frac{45079}{42170}$	$\frac{27203}{20556}$	$\frac{101727}{110482}$
Floating point	-0.167098992	1.068982689	1.323360576	0.920756322
Normalised	-0.1814801460	1.160983273	1.437253857	1

TABLE 3.5: Structural error generating Equation (3.3).

Structural error	Exact synthesis	Continuous approximate synthesis
	0.024159094	-0.002471306

Comparisons of the structural error, defined as the area between the prescribed and generated functions, in the v_1-v_4 plane are enumerated in Table 3.5. It is observed that the structural error for the function generated by the continuous approximate synthesis linkage is an order of magnitude smaller than that of the function generated by the exact synthesis linkage, as can be observed by casual visual inspection of the graphs plotted in Figure 3.3.

This approximately generated $v_4 = f_1(v_1)$ function exactly generates five additional $v_j = f_2(v_i)$ functions given the link lengths identified to approximately generate the prescribed function. These functions are exact in the sense that they have not been explicitly prescribed. These five functions between the angle parameters v_1-v_2, v_1-v_3, v_2-v_3, v_2-v_4, and v_3-v_4 are generated by the identified link lengths, and are illustrated in Figure 3.4, along with the prescribed and continuous approximate $v_4 = f_1(v_1)$ functions.

Now that it has been shown that the function in question is possible to generate, with high fidelity, from a planar RRRR function generator, the process of MMCAAIOS will be applied in a naïve manner so as to demonstrate one of the limitations of the MMCAAIOS algorithm in the following section.

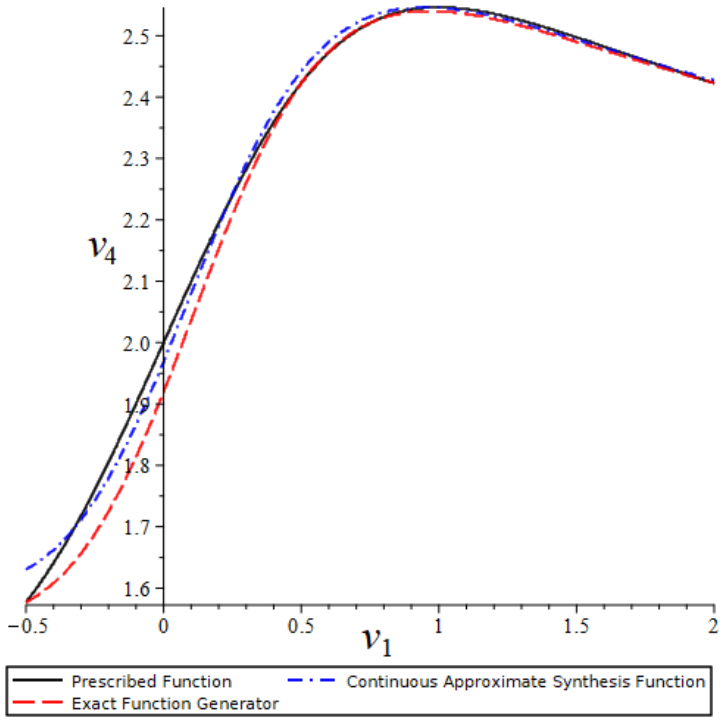


FIGURE 3.3: The prescribed, exact, and continuous synthesis approximation of Equation 3.3 in the v_1-v_4 plane.

Naïve Multi-Modal Synthesis Attempt

Suppose, now, that it is desired to identify a linkage that can approximately generate the v_1-v_4 function in Equation (3.3) and approximately generate a competing v_1-v_3 function that is very different from the v_1-v_3 function generated by the link lengths listed in Table 3.4, and shown in Figure 3.4. The pragmatic mechanical engineer response to such a wish is simply that it is not possible with a planar RRRR linkage. But, should it not be possible to identify a compromise linkage that will generate both desired functions with tolerable structural error? The naïve answer is that it must surely be possible to some extent.

Let us examine this problem from the pragmatic mechanical engineering perspective and select the additional v_1-v_3 function to be,

$$v_3 = f_2(v_1) = 2 + \tan\left(\frac{v_1^2}{{v_1}^2 + 1}\right), \quad -2 \leq v_1 \leq 2. \tag{3.7}$$

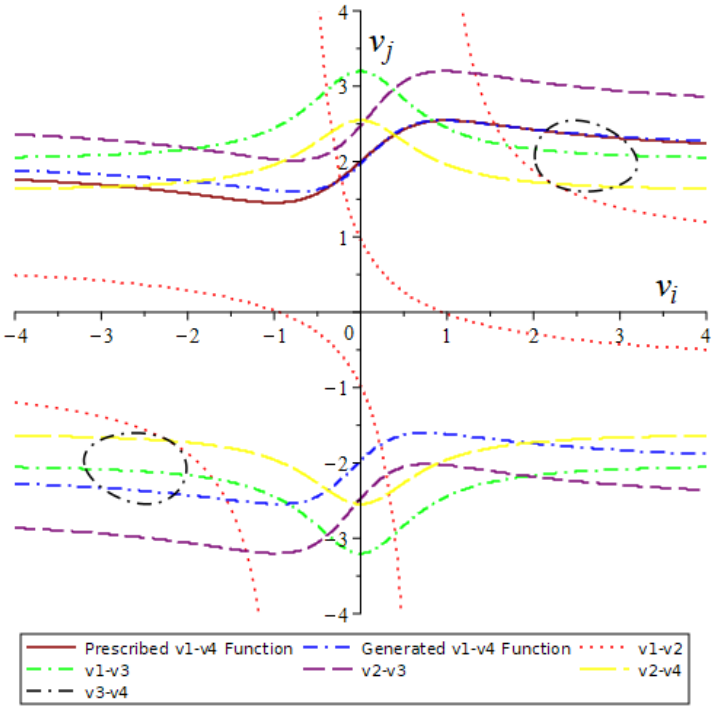


FIGURE 3.4: The prescribed, continuous synthesis approximate, and the five functions generated by the identified link lengths in the v_i-v_j planes.

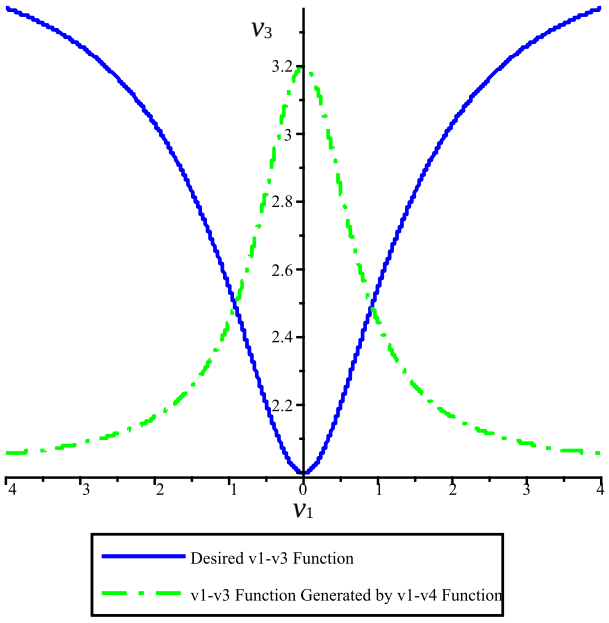


FIGURE 3.5: The desired competing v_1-v_3 function and the one generated by the linkage that approximates Equation 3.3.

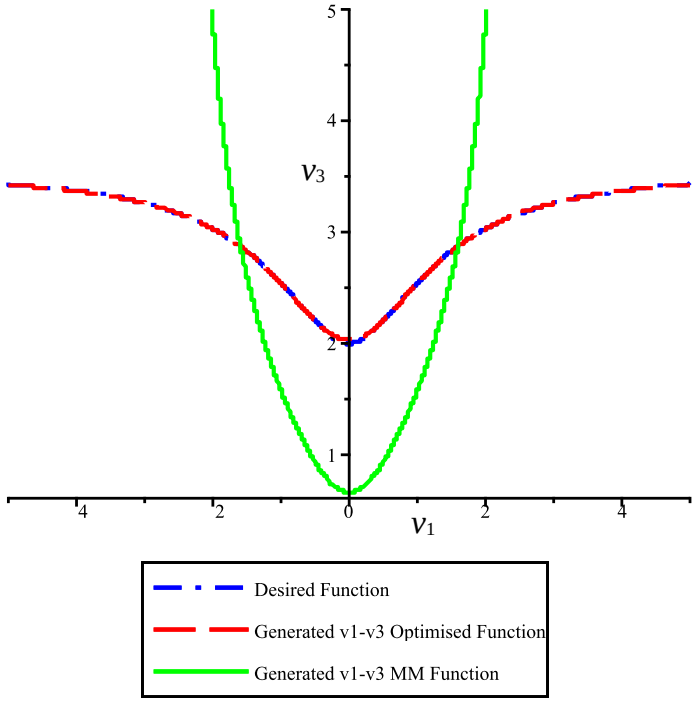


FIGURE 3.6: The v_1-v_3 multi-modal results.

The v_1-v_3 function generated by the linkage that approximately generated the prescribed v_1-v_4 function can be seen in Figure 3.4, and is reproduced for comparison with the very different desired v_1-v_3 function in Figure 3.5. The range $-2 \leq v_1 \leq 2$ is then selected for the prescribed v_1-v_3 function, and $0 \leq v_1 \leq 2$ for the prescribed v_1-v_4 function. Initial guesses for the four link lengths must now be selected in order to Equation (3.2). Arbitrarily, and naïvely, $(a_1,\ a_2,\ a_3,\ a_4) = (1,\ 1,\ 1,\ 1)$, is selected for link length initial guesses, following which the minimise command is allowed to minimise the residual of Equation (3.2). This yields in the remarkably poor results illustrated in Figures 3.6 and 3.7.

Clearly, the naïve approach to the multi-modal function generator synthesis problem yields fundamentally unacceptable results, and more must be controlled before it is to be of use.

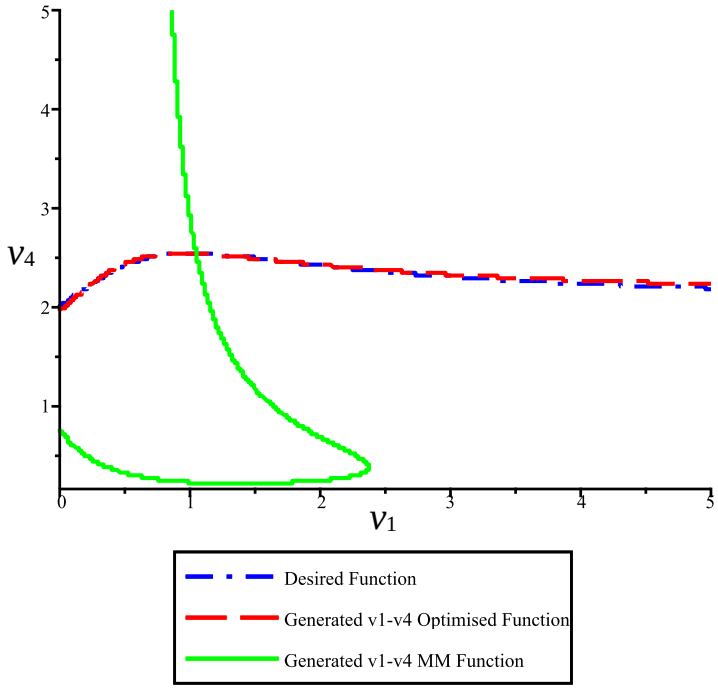


FIGURE 3.7: The v_1-v_4 multi-modal results.

3.2.3 Constrained Multi-Modal Synthesis Example

While the previous example failed to produce a linkage which was even remotely capable of producing the desired functional relationship within both the $v_1 - v_4$ and $v_1 - v_3$ angular parameters, the initial guess used for the optimisation is one which typically is not capable of producing a suitable function generator even for the standard CAAIOS case. Thus, an example should instead be derived through a MMCAAIOS example using a more constrained initial guess in order to ensure that the algorithm itself is not being rendered useless by a simple numerical initialisation problem. The RRRR function generating mechanisms presented in the previous section will be expanded upon within this section. For reference, the target function is,

$$v_4 = 2 + \tan\left(\frac{v_1}{v_1^2 + 1}\right), \tag{3.8}$$

and the resulting link lengths from the CAAIOS method are,

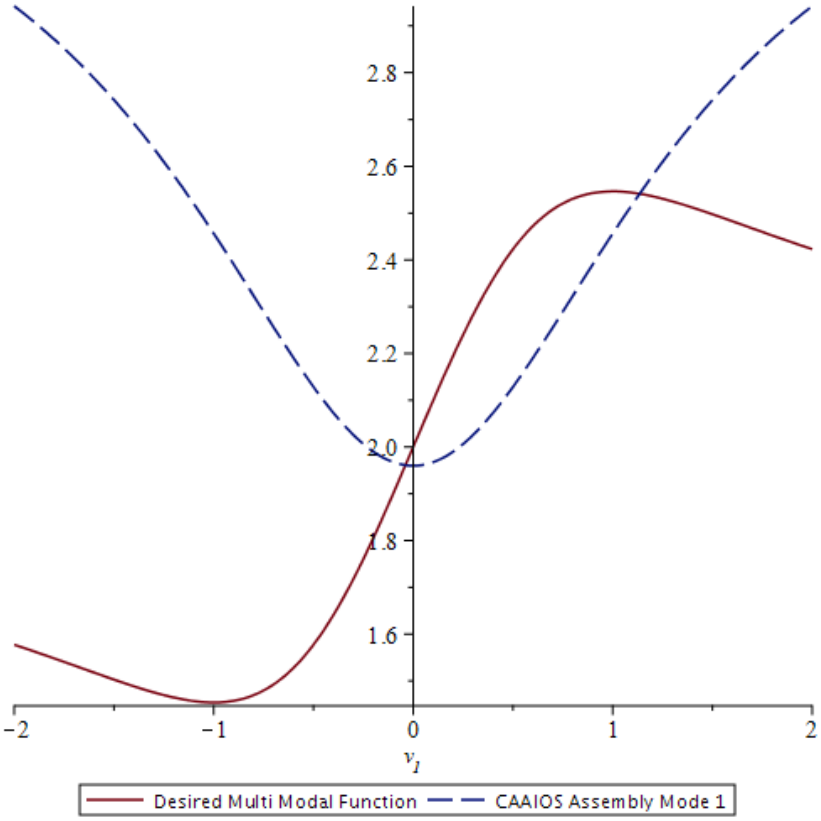


FIGURE 3.8: The desired input-output relationship in $v_1 - v_3$ compared to the same relationship which is generated by the CAAIOS optimal linkage parameters.

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} -0.217 \\ 1.180 \\ 1.415 \\ 1 \end{bmatrix}. \tag{3.9}$$

Now, we define the desired $v_1 - v_3$ functional relationship to be identical to that which is expressed in the $v_1 - v_4$ parameter set. First, we compare the desired function shown to what is generated by the optimal parameters in Equation (3.9) by solving the planar RRRR algebraic input-output relationship in $v_1 - v_3$ for v_3. The comparison between this and the desired relationship is show in Figure 3.8.

Clearly, the desired multi-modal relationship requires a fundamental alteration in the function expressed by the function generator in $v_1 - v_3$. Now, Equation (3.2) will be used to complete the minimisation technique. Unfortunately, in general, this procedure fails; the solver within Maple 2023 is not able to find a point which reduces the error,

and the minimisation algorithm fails for the above relationship. While it was possible to compute an improved point with the previous example, it should be noted that the function generator which results from using $a_i = 1$ as an initial guess for the minimisation algorithm generated a large error in both functions being synthesised, while using the initial guess which was computed from the CAAIOS through the approximation of the $v_1 - v_4$ relationship yields a solution which is simply the lowest possible total error solution available.

Evidently, additional techniques must be employed in order to define the limitations of the MMCAAIOS algorithm, and to adjust for the numerical sensitivity of the optimisation process.

Weighted Multi-Modal RRRR Function Generator Synthesis

The failure of the MMCAAIOS method appears to result from a lack of sensitivity within the problem statement to the errors within the secondary $v_1 - v_3$ relationship. Considering this lack of numerical sensitivity, the following modification to Equation (3.2) is proposed,

$$\min_{(a_1,a_2,a_3,a_4)\in\mathbb{R}} \left(W_1 \mathbf{c}_1 \cdot \int_{v_{i_{1_{\min}}}}^{v_{i_{1_{\max}}}} \mathbf{s}_1(v_{i_1}, f_1(v_{i_1}))dv_{i_1} + W_2 \mathbf{c}_2 \cdot \int_{v_{i_{2_{\min}}}}^{v_{i_{2_{\max}}}} \mathbf{s}_2(v_{i_2}, f_2(v_{i_2}))dv_{i_2} \right), \tag{3.10}$$

where the factors W_1, and W_2, in Equation (3.10) are some non-zero positive weighting factors. These factors, within the problem which follows, are some positive value that serve to amplify the impact of the errors within either the $v_1 - v_4$ or $v_1 - v_3$ input-output relationship, in order to drive more sensitivity within the optimisation routine. For brevity, several weighting factors were used for W_2 in order to discover the point at which the multi-modal equation would converge. The factor which was seen to produce a convergence within the multi modal relationship was $W_2 = 400$, indicating that in order for the multimodal algorithm to converge, the error induced within the original $v_1 - v_4$ relationship is 400 times larger than the error reduced within the $v_1 - v_3$ relationship. Figure 3.9 shows the resulting $v_1 - v_4$ function.

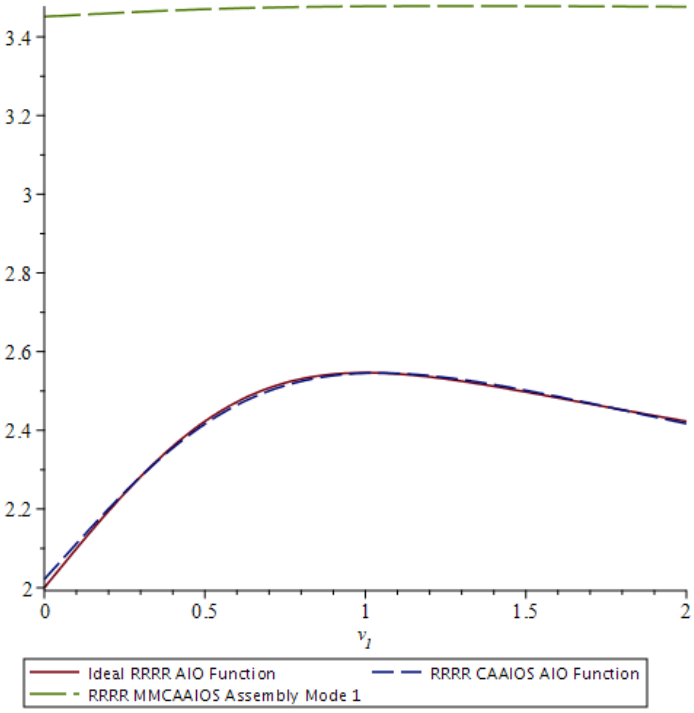


FIGURE 3.9: The $v_1 - v_4$ equation resulting from the weighted multi-modal synthesis algorithm.

Clearly, the relationship in $v_1 - v_4$ is untenable if the goal was to preserve as much of the original relationship as possible and to simply fine-tune the relationship within the $v_1 - v_3$ input-output function. However, considering the damage wrought to the original relationship, the $v_1 - v_3$ function must be significantly improved in order to justify the reduction in accuracy within the $v_1 - v_4$ relationship. Figure 3.10 shows the relationship resulting from the weighted multi-modal synthesis algorithm, which clearly has not improved sufficiently to justify the loss in accuracy within the $v_1 - v_4$ IO relationship.

It is obvious from this demonstration that, while a weighting factor can be used in order to force convergence to the error minimising linkage within some secondary set of IO parameters, the effect of the weighting factor destroys the relationship which was originally derived. Thus, if the multi-modal synthesis algorithm is to be successful in a meaningful way, additional constraints must be levelled on the secondary function to be generated.

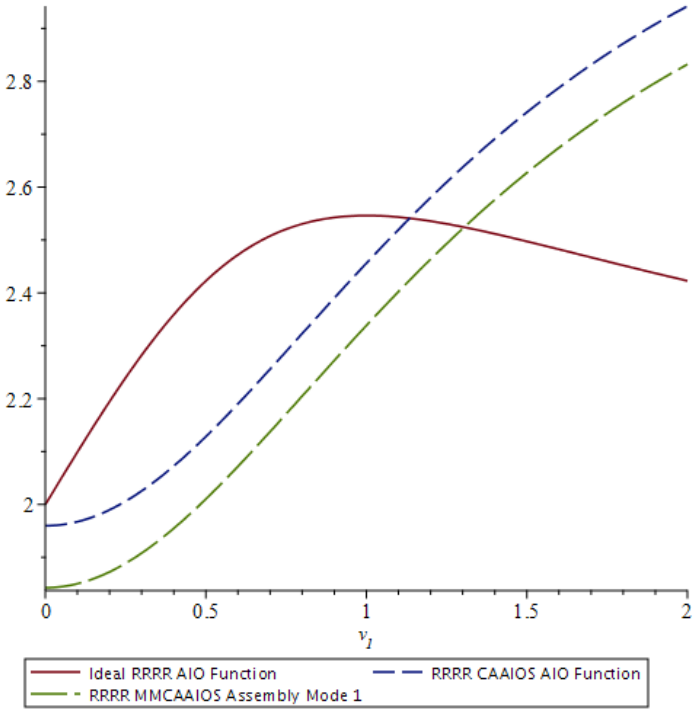


FIGURE 3.10: The $v_1 - v_4$ equation resulting from the weighted multi-modal synthesis algorithm.

3.2.4 Functionally Constrained Multi-Modal Function Generator Synthesis

Based on the above it is now evident that, in general, a single function generator cannot be made to produce arbitrarily competing functions, and furthermore that using a weighting strategy in order to force the linkage to generate the secondary function degenerates the relationship for which the original linkage was intended. However, is the MMCAAIOS algorithm even so much as tangentially useful, or does the nature of the motion constraints levelled by four-bar function generators render this approach without applicability? The following section will investigate the implications of a functional constraint on the input-output relationship in the $v_1 - v_3$ pair and its impacts on the applicability and limits of the MMCAAIOS algorithm.

The $v_1 - v_4$ equation to be synthesised is the same target from the previous section of this thesis, Equation (3.8), resulting in the following set of link lengths following the standard continuous approximate synthesis approach,

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} -0.217 \\ 1.180 \\ 1.415 \\ 1 \end{bmatrix}, \tag{3.11}$$

which are generated when the continuous approximate synthesis algorithm is used on the interval of $v_1 = 0 \ldots 2$. For the purposes of this demonstration, the same interval will be maintained for the $v_1 - v_3$ function generator, implying that the designer is attempting to modify the parameter pair which would represent the transmission angle of the previously obtained function generator over the same range as the function generator was synthesised. In order to attempt to define some of the boundaries of the MMCAAIOS method, a heavily constrained target function in $v_1 - v_3$ will be used, which is similar to what is produced when the $v_1 - v_3$ relationship is solved from the parameters in Equation (3.9). Upon investigation of the $v_1 - v_3$ function generated from the original CAAIOS solution shown in Section (3.2.3), the target function,

$$v_3 = \frac{9}{5} + \tan\left(\frac{v_1^2}{v_1^2 + 1}\right), \tag{3.12}$$

is defined as a close target function. Figure 3.11 shows the differences between the $v_1 - v_3$ CAAIOS IO function and the desired relationship expressed by Equation (3.12).

The desired $v_1 - v_3$ function is, in this case, primarily a vertical translation, and therefore simply a shift of the $v_1 - v_3$ input-output function generated by the previously completed CAAIOS problem. Using a uniformly weighted multimodal function generator architecture where $W_1 = W_2 = 1$, and integrating both input-output relationships in terms of the input parameter, v_1, the following function is minimised,

$$\min_{(a_1,a_2,a_3,a_4)\in\mathbb{R}} \left(\mathbf{c}_1 \cdot \int_{v_{1_{1_{\min}}}}^{v_{1_{1_{\max}}}} \mathbf{s}_1(v_{1_1}, f_1(v_{1_1}))dv_{1_1} + \mathbf{c}_2 \cdot \int_{v_{1_{2_{\min}}}}^{v_{1_{2_{\max}}}} \mathbf{s}_2(v_{1_2}, f_2(v_{1_2}))dv_{1_2} \right), \tag{3.13}$$

where the initial guesses used for the link length parameters are taken from Equation (3.11), $f_1(v_1)$ is defined in Equation (3.8), and $f_2(v_2)$ is defined in Equation (3.12). Following minimisation of the residual of this equation, the following optimal link lengths are obtained,

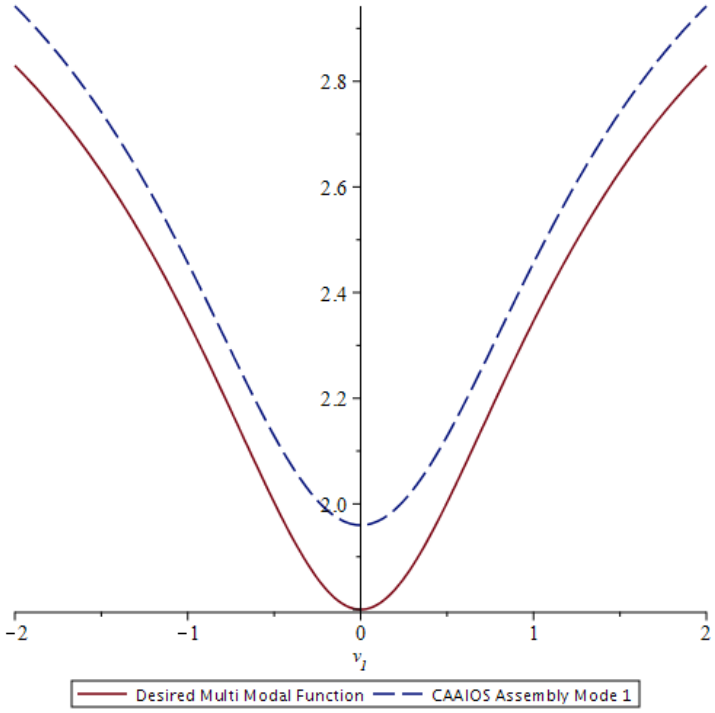


FIGURE 3.11: The desired multi-modal function in $v_1 - v_3$ and the relationship resulting from the CAAIOS of the linkage with a defined IO relationship in $v_1 - v_4$.

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} -0.239 \\ 1.182 \\ 1.388 \\ 1 \end{bmatrix}, \tag{3.14}$$

which deviate only slightly from the link lengths presented in Equation (3.11). However, as seen in previous sections, a small deviation may have massive implications in the ability of a mechanism to satisfy the desired motion constraints. However, as can be seen in Figure 3.12, the $v_1 - v_4$ function generated following the MMCAAIOS algorithm only deviates slightly from the previously defined optimal function generator.

While this is an encouraging result, was the original goal of synthesising Equation (3.12) realised in this iteration of the multi-modal synthesis problem? Figure 3.13 shows that, in fact, the original goal was realised; the desired relationship is far more closely approximated.

This result would seem to indicate that the multi-modal synthesis approach is viable for modest changes into some $v_i - v_j$ input-output pair in a linkage which has already

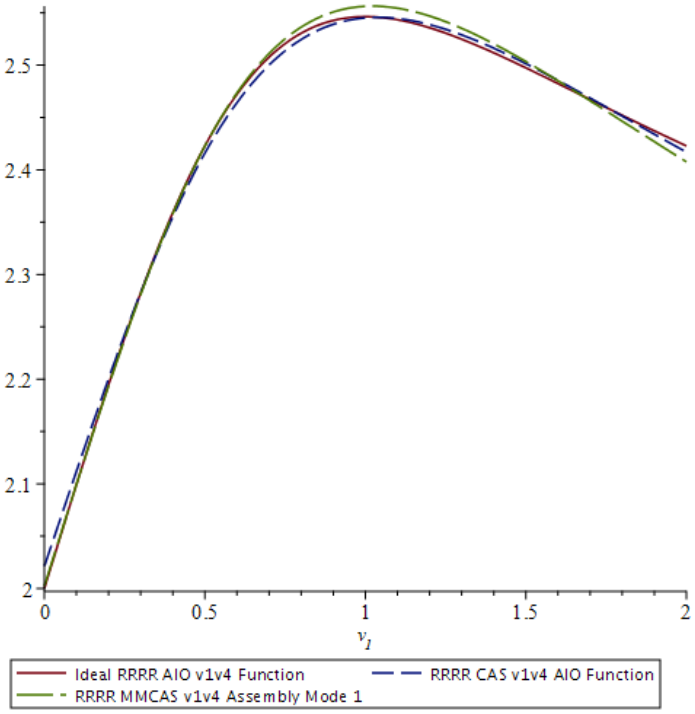


FIGURE 3.12: The desired function in $v_1 - v_4$ and the relationship resulting from the CAAIOS in $v_1 - v_4$ coupled with the desired $v_1 - v_3$ relationship.

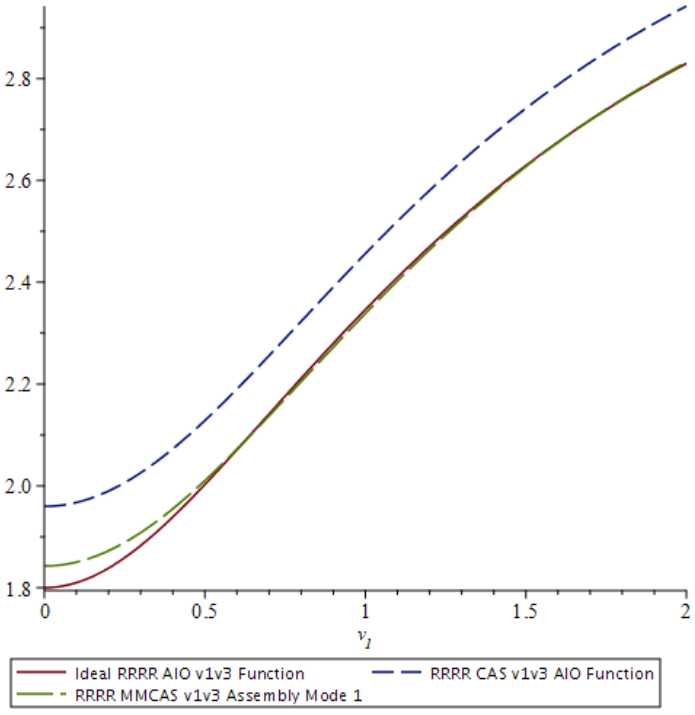


FIGURE 3.13: The desired multi-modal function in $v_1 - v_3$ and the relationship resulting from the CAAIOS in $v_1 - v_4$.

been designed so as to optimally generate an IO relationship in some other $v_i - v_j$ pair. While the original goal of the MMCAAIOS problem was to be able to develop a single function generator which generates two separate competing functions, the results contained within this section indicate that it may be possible to fine-tune an optimal function generator in order to either modify the transmission angle of the linkage through its functional range, or to ensure that some motion envelope is satisfied in another angular pair. Admittedly, this example is somewhat contrived, but it is meant to serve as a proof of concept.

3.2.5 Multi-Modal Synthesis in Practice

As previously described, the second, third, fourth, et c., prescribed functions need to be constrained with respect to the five functions generated by the link lengths that approximately generate the first prescribed function in the absence of a useful initial guess. If it is desired to generate a different, though heavily constrained, $v_3 = f_2(v_1)$ function for example, a generatable function may be specified as an interpolant of the one determined by the specified primary $v_4 = f_1(v_1)$ function. To do this, Lagrange polynomial interpolation has proven to be an acceptable method.

The first step is to solve the v_1-v_3 IO equation imposed by the generated $v_4 = f_1(v_1)$ function. This yields the exact $v_3 = f_2(v_1)$ function generated by the identified a_i link lengths that approximately satisfy the specified $v_4 = f_1(v_1)$ function. Select n (v_1, v_3) IO pairs from the exact $v_3 = f_2(v_1)$ function generated by the identified a_i to use as inputs for the Lagrange polynomial interpolation formula. In general, this method takes the n points in an arbitrary x-y plane, with no two x_i the same and returns a polynomial of degree at most $d \leq n - 1$.

The Lagrange polynomial interpolant is a linear combination

$$L(x) \quad = \quad \sum_{i=1}^{n} y_i l_i(x),$$

of Lagrange basis polynomials,

$$l_i(x) = \prod_{\substack{1\leq m\leq n \\ m\neq i}} \frac{x-x_m}{x_i-x_m} = \left(\frac{x-x_1}{x_i-x_1}\right)\left(\frac{x-x_2}{x_i-x_2}\right)\cdots\left(\frac{x-x_n}{x_i-x_n}\right).$$

For the computational *proof-of-concept* example we will use a system of primary and secondary prescribed functions. The primary function is arbitrary. But the secondary is some Lagrange polynomial interpolant of the function imposed by the link lengths identified that approximately generate the primary function. The link lengths that approximately generate the primary function will be used as initial guesses for the multi-modal synthesis with the secondary polynomial interpolant function. The primary function we wish to generate with a planar RRRR closed kinematic chain is Equation (3.3). The corresponding v_1-v_3 function exactly generated by the identified link lengths is obtained from the v_1-v_3 IO equation, Equation (1.71), using the a_i from the $v_4 = f_1(v_1)$ continuous approximate synthesis step listed in Table 3.4 is,

$$v_3 = \pm\frac{11268158900\sqrt{\left(v_1^2+\frac{28145}{62561}\right)\left(v_1^2+\frac{43467}{38278}\right)}}{5593605380v_1^2+2516456313}. \tag{3.15}$$

Suppose that this crank-rocker four-bar linkage was required to precisely time two punch presses. Four holes created by the presses are required to be precisely located on a single automotive quarter panel which is advanced in a jig under the action of the input link of the mechanism. One quarter panel completely advances per 360° rotation of the input crank link. The first punch press is actuated by a trigger that is activated under the action of θ_4, while the second is actuated by θ_3. The $v_4 = f_1(v_1)$ trigger function is that of Equation (3.3). However, after the linkage is synthesised, the resulting $v_3 = f_2(v_1)$ function, Equation (3.15), does not satisfy the angle requirement. The trigger for this punch press must be actuated when the input angle locating the quarter panel has the precise values $\theta_1 = 0.00° \pm 0.05°$ and $\theta_1 = 90.00° \pm 0.05°$. At these input angles the corresponding values of θ_3 must be precisely $\theta_3 = 145.25° \pm 0.05°$ and $\theta_3 = 135.25° \pm 0.05°$. Unfortunately, while the values of θ_4 generated by the linkage

obtained by continuous approximate synthesis as listed in Table 3.4 are within tolerance for the required input angles those for θ_3 are not. The required angle generated by this linkage at $\theta_1 = 0.00^\circ \pm 0.05^\circ$ is $\theta_3 = 145.50^\circ \pm 0.05^\circ$ and at $\theta_1 = 90.00^\circ \pm 0.05^\circ$ is $\theta_3 = 135.10^\circ \pm 0.05^\circ$, both out of tolerance, though only marginally, see Table 3.6. Relaxing the tolerances is deemed to not be an acceptable design course of action. In this case, subtly perturbing the $v_3 = f_2(v_1)$ function generated by the required $v_4 = f_1(v_1)$ function, Equation (3.3), may yield the required θ_4 and θ_3 output angles.

TABLE 3.6: Required and $v_4 = f_1(v_1)$ generated values of θ_3 at required θ_1.

Required θ_1	$0.00^\circ \pm 0.05^\circ$	$90.00^\circ \pm 0.05^\circ$
Required θ_3	$145.25^\circ \pm 0.05^\circ$	$135.25^\circ \pm 0.05^\circ$
Generated θ_3	145.50°	135.10°

To achieve this, we will attempt to use Lagrange polynomial interpolation to obtain a different, but constrained function using $n = 4$ points on the (upper signed) $v_3 = f_2(v_1)$ curve, Equation (3.15):

$$(v_1, v_3) = \left(-\frac{1}{2}, \frac{62167}{21933}\right), \left(\frac{1}{4}, \frac{80364}{26089}\right), \left(\frac{3}{5}, \frac{64227}{23462}\right), \left(\frac{11}{10}, \frac{39821}{16629}\right).$$

The resulting degree 3 Lagrange polynomial function $v_3 = f_2(v_1)$ is

$$\begin{aligned} v_3 = {} & \frac{140152452564627675650}{146115499161206849967} v_1^3 - \frac{148500638129317309265}{97410332774137899978} v_1^2 \\ & - \frac{136182081139230857387}{584461996644827399868} v_1 + \frac{57010242995943671417}{17710969595297799996}, \end{aligned} \tag{3.16}$$

$$\text{for} \quad -\frac{1}{10} \leq v_1 \leq \frac{5}{4}.$$

Let this be the specified secondary function. Both the interpolant, Equation (3.17), and the precise v_1-v_3 function, Equation (3.15), generated by the link lengths that were identified to approximately generate Equation (3.3) are illustrated in Figure 3.14.

Careful examination of Figure 3.14 reveals that both Equation (3.15) and (3.17) are

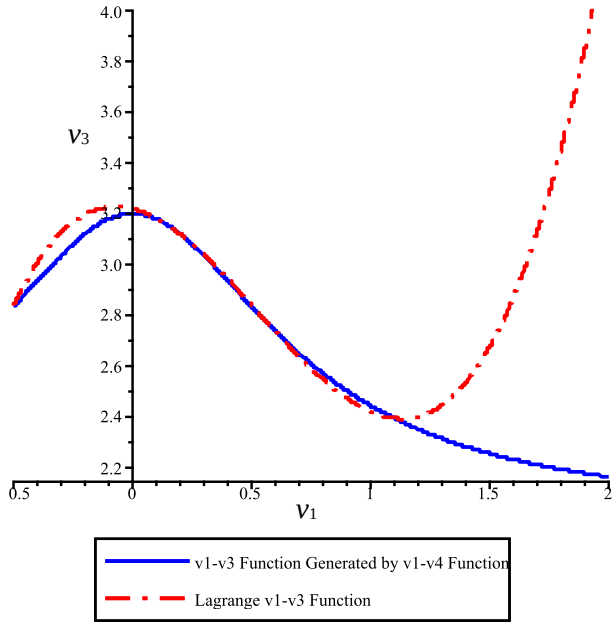


FIGURE 3.14: The polynomial interpolant, Equation (3.17), and the v_1-v_3 function, Equation (3.15), generated by the linkage that approximates Equation (3.3).

very close to each other in the range $-\frac{1}{10} \leq v_1 \leq \frac{5}{4}$. To demonstrate that our kinematic model of the geometry of multi-modal synthesis will lead to a computationally useful result, we will use these as the integration limits for the v_1-v_3 secondary function. Hence, the primary $v_4 = f_1(v_1)$, Equation (3.3), and secondary $v_3 = f_2(v_1)$, Equation (3.17), are used to generate the respective synthesis equations with variable angle parameters expressed as v_1 and $f_1(v_1)$ in the primary, and v_1 and $f_2(v_1)$ in the secondary. The two synthesis equations are squared, then the coefficients and variables are separated into the arrays $\mathbf{c}_1$, $\mathbf{s}_1(v_1, f_1(v_1))$, $\mathbf{c}_2$, and $\mathbf{s}_2(v_1, f_2(v_1))$, from which the target function is constructed,

$$\min_{(a_1,a_2,a_3,a_4)\in\mathbb{R}} \left(\mathbf{c}_1 \cdot \int_{v_1=-\frac{1}{2}}^{v_1=2} \mathbf{s}_1(v_1, f_1(v_1)) dv_1 + \mathbf{c}_2 \cdot \int_{v_1=-\frac{1}{10}}^{v_1=\frac{5}{4}} \mathbf{s}_2(v_1, f_2(v_1)) dv_1 \right). \tag{3.17}$$

The multi-modal computations for Equation (3.17) converge to the link lengths listed in Table 3.7. The results are graphically illustrated in Figure 3.16 and the structural

errors, defined as the areas between the prescribed and generated functions are listed in Table 3.8.

TABLE 3.7: The $v_4 = f_1(v_1)$ and $v_3 = f_2(v_1)$ planar 4R multi-modal synthesis results.

Link length	a_1	a_2	a_3	a_4
Floating point	-0.1478064777	0.9299394483	1.148016662	0.8023065449
Normalised	-0.1842269375	1.159082466	1.430895297	1

TABLE 3.8: The $v_4 = f_1(v_1)$ and $v_3 = f_2(v_1)$ planar 4R multi-modal synthesis structural errors.

	Structural error
$v_4 = f_1(v_1)$ only	-0.002471306
$v_4 = f_1(v_1)$ multi-modal	0.009542948
$v_3 = f_2(v_1)$ only	0.005358289
$v_3 = f_2(v_1)$ multi-modal	0.004161159

It is to be seen that the structural error for the $v_4 = f_1(v_1)$ results increases by a factor of nearly four, but is still tolerably small. While the structural error for the $v_3 = f_2(v_1)$ multi-modal results decreases modestly. However, the important outcome in this case is that at the required input angles the corresponding required values of θ_4 are still within tolerance, and those of θ_3 are as well. When $\theta_1 = 0.00^\circ \pm 0.05^\circ$ the multi-modal linkage generates $\theta_3 = 145.25^\circ \pm 0.05^\circ$ and at $\theta_1 = 90.00^\circ \pm 0.05^\circ$ we obtain $\theta_3 = 135.28^\circ \pm 0.05^\circ$, both within tolerance. The relevant values of this outcome are listed in Table 3.9.

TABLE 3.9: Required and multi-modal generated values of θ_3 at required θ_1.

Required θ_1	$0.00^\circ \pm 0.05^\circ$	$90.00^\circ \pm 0.05^\circ$
Required θ_3	$145.25^\circ \pm 0.05^\circ$	$135.25^\circ \pm 0.05^\circ$
Generated θ_3	145.25°	135.28°

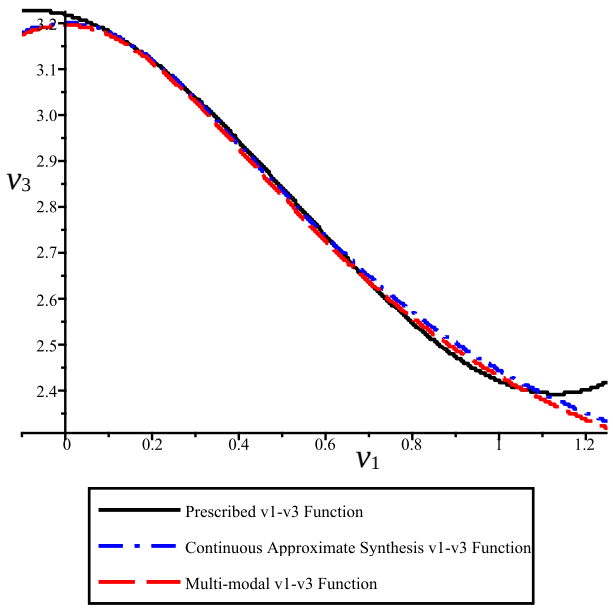


FIGURE 3.15: The v_1-v_3 multi-modal results.

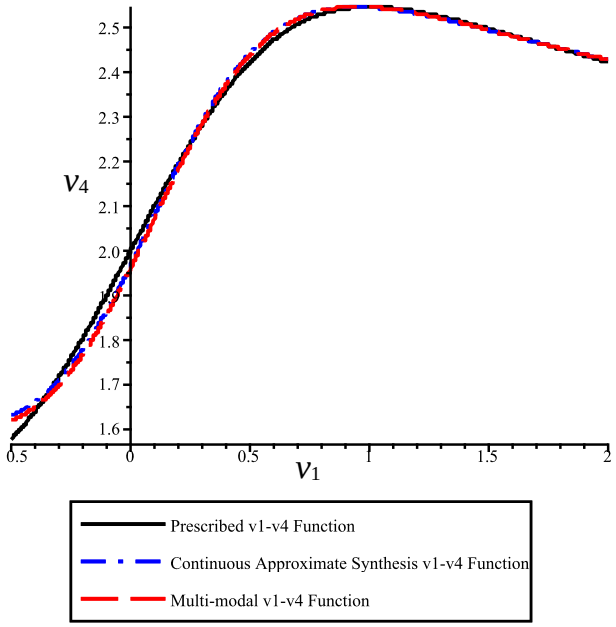


FIGURE 3.16: The v_1-v_4 multi-modal results.

While it can be seen from Table 3.7 that the structural error associated with the $v_1 - v_4$ IO equation increased in magnitude following the MMCAAIOS algorithm, the structural error associated with the $v_1 - v_3$ IO pair decreased, and the physical constraints (shown in Table 3.9) associated with the required operational characteristics of the linkage were realised.

3.2.6 RRRP Multi-Modal Function Generator Synthesis

Considering the results from the previous sections of this thesis, what remains to be seen is whether or not the MMCAAIOS is an abberation of the planar RRRR linkage geometry, or if this technique is equally as applicable to other function generator architectures. As with all fundamentally novel design techniques, difficulties emerge in identifying methods whereby the new technique may be validated against previous accepted techniques within the literature. Problematically, multi-modal synthesis has never, to the best of the author's knowledge, been attempted before, and thus comparisons are not possible for these examples. In the interest of developing a more unified theorem, the DH parameter equations, as shown in Section 1.6.5 will be used for the MMCAAIOS example to follow. For ease of reference, the RRRP IO corresponding to the $v_1 - d_4$ IO pairing is replicated here as,

$$v_1^2 d_4^2 + R v_1^2 + d_4^2 - 4a_1 v_1 d_4 + S = 0, \tag{3.18}$$

where

$$R = R_1 R_2 = (a_1 + a_2 - a_4)(a_1 - a_2 - a_4),$$

$$S = S_1 S_2 = (a_1 + a_2 + a_4)(a_1 - a_2 + a_4),$$

$$v_1 = \tan \frac{\theta_1}{2}.$$

The remaining joint variable parameter pairings lead to the following five RRRP algebraic IO equations, and are replicated here for ease of reference:

$$R_2v_1^2v_2^2 + R_1v_1^2 - S_2v_2^2 + 4a_2v_1v_2 - S_1 = 0; \tag{3.19}$$

$$R_1v_1^2v_3^2 + R_2v_1^2 - S_2v_3^2 - S_1 = 0; \tag{3.20}$$

$$S_2v_2^2v_3^2 - R_2v_2^2 - R_1v_3^2 - 4a_1v_2v_3 + S_1 = 0; \tag{3.21}$$

$$v_2^2d_4^2 - R_2S_2v_2^2 + d_4^2 - R_1S_1 = 0; \tag{3.22}$$

$$v_3^2d_4^2 + R_1S_2v_3^2 + d_4^2 + 4a_2v_3d_4 - R_2S_1 = 0. \tag{3.23}$$

The primary $d_4 = f_1(v_1)$ function is arbitrarily chosen to be

$$d_4 = 2 - \ln\left(\frac{v_1^2}{v_1^2+1}\right), \quad \frac{1}{10} \leq v_1 \leq 6. \tag{3.24}$$

To generate an initial guess for the multi-modal synthesis, the CAAIOS process is completed, identifying the following link lengths:

$$a_1 = -\frac{21527}{19453}, \; a_2 = \frac{62456}{9833}, \; a_4 = \frac{66527}{13759}. \tag{3.25}$$

After following similar computation steps as for the planar RRRR multi-modal synthesis example in Section 3.2.5, we determine the secondary $v_3 = f_2(v_1)$ function again as a degree 3 Lagrange interpolant:

$$\begin{aligned} v_3 = {} & \frac{8575459781525718313}{2128203922635547524924}v_1^3 - \frac{926446934929263804951}{709401307545182508 3080}v_1^2 \\ & + \frac{145850030457909132287}{123732786199741135170}v_1 + \frac{3255237430904027623667}{1773503268862956270770}. \end{aligned} \tag{3.26}$$

Then arbitrarily, but without loss of generality, assign the integration limits for this perturbed secondary function to be the same as those of the primary function. The multi-modal synthesis is then performed by evaluating

$$\min_{(a_1,a_2,a_4)\in\mathbb{R}} \left(\mathbf{c}_1 \cdot \int_{v_1=\frac{1}{10}}^{v_1=6} \mathbf{s}_1(v_1, f_1(v_1))dv_1 + \mathbf{c}_2 \cdot \int_{v_1=\frac{1}{10}}^{v_1=6} \mathbf{s}_2(v_1, f_2(v_1))dv_1 \right). \tag{3.27}$$

The numerical optimiser in Maple 2023 converges to the link lengths listed in Table 3.10, while the structural errors for each of the two generated functions are listed in Table 3.11. To help visualise the areas between the prescribed and generated functions the results are illustrated in Figures 3.17 and 3.18. It can be seen that the structural error in $v_1 - v_3$ decreases for the multi-modal synthesis results.

TABLE 3.10: The $d_4 = f_1(v_1)$ and $v_3 = f_2(v_1)$ planar RRRP multi-modal synthesis results.

Link length	a_1	a_2	a_4
Rational	$-\frac{26513}{23888}$	$\frac{85324}{13461}$	$\frac{127711}{26510}$
Floating point	-1.10988780904891	6.33860782950366	4.81746510770270

TABLE 3.11: The $d_4 = f_1(v_1)$ and $v_3 = f_2(v_1)$ planar RRRP multi-modal synthesis structural errors.

	Structural error
$d_4 = f_1(v_1)$ only	-0.23104280
$d_4 = f_1(v_1)$ multi-modal	-0.24046271
$v_3 = f_2(v_1)$ only	0.23488469
$v_3 = f_2(v_1)$ multi-modal	0.15360825

These examples are intended to serve as a conceptual demonstration of the MMCAAIOS procedure, however, it is not intended as a prediction of how multi-modal function generator synthesis will behave in the general case. While the limitations of the MMCAAIOS are still somewhat unclear from these contrived examples, they include, but are not necessarily limited to: which types of functions may be generated by each angle parameter pair for a given planar function generator architecture; how these possibilities change based on the initial linkage parameters used for the optimisation; the effective limits on precisely how much the error associated with the designers initial function generator will increase given how far the secondary function is made to

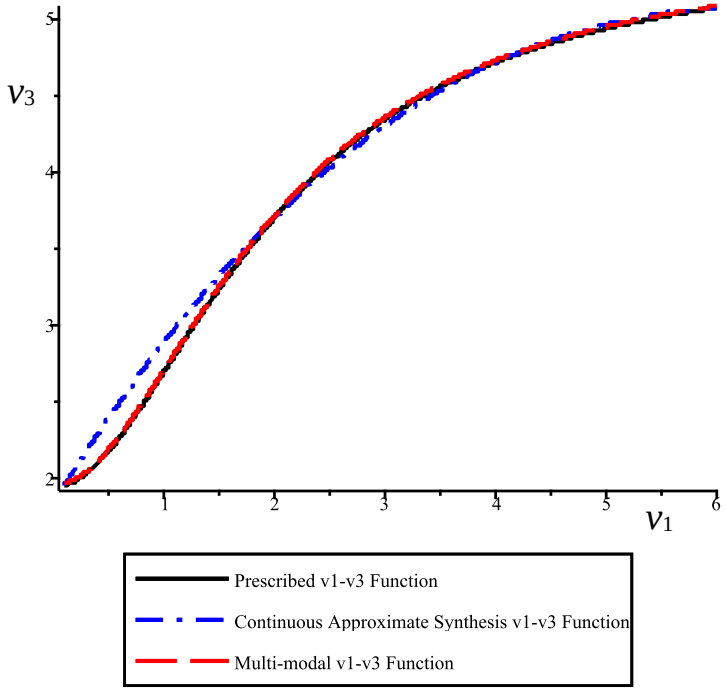


FIGURE 3.17: The v_1-v_3 RRRP multi-modal results.

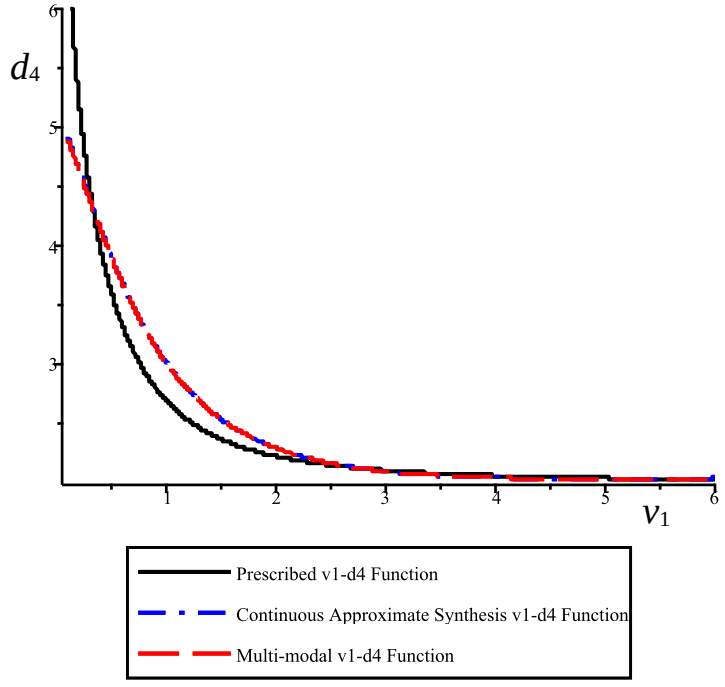


FIGURE 3.18: The v_1-d_4 RRRP multi-modal results.

deviate from its initial form.

3.3 Discussion

The main goal of this chapter was to describe the limitations, and to explore possible applications of, the novel four-bar planar mechanism algorithm that implicitly drives the cardinality of the IO data set used to generate the over constrained set of synthesis equations to infinity. The computational efficiency of the algorithm itself allows it to be used in ways that classical approaches may not have been used in the past. These extensions include the ability to simultaneously identify the type and dimensions associated with some optimal function generator through a simple comparison of the integrals of the functions generated by each linkage architecture following optimal synthesis. Additionally, the MMCAAIOS problem was attempted; the act of using the same function generating linkage to approximate two separate functions in different IO pairs, simultaneously. While the original intent of the multi-modal function appears to be impossible, to generate two arbitrarily competing functions with the same linkage in two different IO pairs, the use of the MMCAAIOS algorithm was demonstrated through two constrained examples.

The algorithm was demonstrated with these multi-modal synthesis examples, in a proof-of-concept fashion, to simultaneously generate primary and perturbed secondary functions in each of an RRRR and an RRRP planar linkage, and to demonstrate that generation of competing functions with a planar four-bar linkage is, in general, not possible. Further, the use of a weighting function on the secondary relationship within the MMCAAIOS algorithm does precisely nothing to improve the results and is, in fact, more deleterious to the maintenance of the original optimally generated function than it improves the performance within the synthesis of the secondary function. When the MMCAAIOS is used to approximate only modestly competing functions, however, the results lead to reductions in the structural errors within the secondary function to a large degree, and a modest increase to the structural errors associated with the primary function. Certainly, any planar four-bar mechanism generates an output joint parameter that

is a distinct function of the input joint variable parameter. The linkage that generates this distinct function also exactly determines five additional functions between the remaining pairs of variable joint parameters. The synthesis examples in this chapter have demonstrated that it is possible to approximately generate two distinct, though heavily constrained, IO functions between different variable joint pairs that have not been already determined by the linkage geometry. This simple result illustrates the tremendous value represented by the algebraic input-output equations as design and analysis tools.

The algebraic IO equations described herein, together with the MMCAAIOS algorithm, stand to enable designers of industrial automated production and assembly systems to approach optimisation in a new way: different linkages in the mechanical system that are capable of generating multiple different prescribed functions so that each link in the chain can simultaneously perform different tasks, or perhaps to assist in preparing linkages which are suitable for optimal force transfer through the linkage while simultaneously maintaining some desired input-output relationship. While the practicality of this is, of course, conjecture, it does suggest the continued generalisation and development of MMCAAIOS is justified and worth the investigative effort.

4 Conclusions

The practice of the kinematic synthesis of linkages has evolved over the centuries, from strictly graphical techniques to those which evolved from the Freudenstein equation first published in 1954. Classical techniques use a discrete synthesis method which relies on the minimisation of either the design or the structural error associated with the function being generated by the linkage. While the design error is a linear least squares minimisation, it is a technique which relies solely on the minimisation of the residual of the Freudenstein equation, which bears no resemblance to a physical property of the linkage. In contrast to this, the structural error minimises the difference between the prescribed and generated function at each precision point, and works towards directly minimising the IO error of the resulting linkage. However useful the structural error minimisation technique may be, it is often difficult to implement due to its highly non-linear nature. In addition to these difficulties, the formulation for the design and structural errors relies on the use of absolute angular measurements, meaning the that examination of the intermediate joint angles, and any input-output equations associated with these intermediate joint angles, relies on fundamentally different factors in terms of joint length or angle parameters. Further to these concerns, it has been noted in the literature that as the cardinality of the data set used to develop the minimisation algorithm for a four-bar linkage becomes large, the design error minimising linkage approaches the structural error minimising linkage, implying that the design error minimisation may be used in place of the structural error minimising linkage given a sufficient number of precision points.

The Freudenstein equation was successfully integrated and used to develop a continuous approximation algorithm in previous works [57, 83] in order to conceptually

drive the cardinality of the target data set to infinity. However, given the complexity of the integration procedure and the numerical sensitivity of the problem, the method was often less practical than a classical structural error minimisation algorithm. Further to these computational difficulties, the formulation of intermediate joint angle input-output equations was still difficult, and the form of their coefficients was still impossible to generalise. Thus, a more elegant representation of the planar four-bar function generating mechanism input-output equation was sought. Upon the development of the algebraic input-output equations, however, these concerns evaporated. This technique allowed for every single combination of input-output equations to be generated, with uniform and consistent coefficients, for all of the planar (RRRR, RRRP, PRRP) linkages, in addition to the spherical RRRR and the spatial RSSR linkages.

Given the advancements of the algebraic input-output equations, and all of the advantages they convey in addition to their ease of use, the problem of the continuous approximate dimensional synthesis of four-bar function generating linkages was revisited. This thesis contains an exhaustive list of the work required in order to completely generalise the minimisation of the design error, over a continuous set of infinitely many points, for every single linkage architecture for which the algebraic form of the input-output equation exists. The methods presented herein are fundamentally general, simplifying the minimisation of the design and subsequently the structural errors of the linkage to a single equation,

$$\min_{(p_1,p_2,\ldots,p_n)\in\mathbb{R}} \left(\mathbf{p}_A \int_{v_{1min}}^{v_{1max}} \mathbf{s}_A(v_1, f(v_1)) \right), \tag{4.1}$$

where the parameter array, $\mathbf{p}_A$, containing linkage geometric properties, p_i, is concatenated against the integrated synthesis array, $\mathbf{s}_A$, which contains the desired input-output function, and subsequently minimised over the field of real numbers, using the simple solution to the three point precision problem as an initial guess. The application of this method requires precisely no exceptions for all linkage architectures examined within this thesis, and is thus completely generalised. Given the form of the algebraic input-output equations, the design error can be considered to be the distance between

the curve generated by the linkage and the desired curve, while the structural error is the area between those two curves. This metric allows for an efficient comparison of the performance of linkages in generating the desired curve. This continuous approximate input-output synthesis method was validated against examples found in literature, and shown to produce results of higher fidelity and lower structural error than that which was produced through a discrete structural error minimisation algorithm. Furthermore, the continuous approximate algebraic input-output synthesis algorithm is able to achieve these results orders of magnitude faster than conventional methods. The continuous approximate synthesis method was then used to synthesise a linkage for all five linkage architectures in order to demonstrate the ubiquitous nature of its formulation and computational efficiency.

Given the computational efficiency presented by the continuous approximate synthesis method, its application was extended to questions that are not typically easily answered by conventional methods. Two additional design considerations are levelled within this thesis. Firstly, the problem of combined type and dimensional synthesis; for a given function, what is the best possible linkage architecture to use to synthesise it over a given range. Secondly, multi-modal function generator synthesis, or the practice of synthesising two distinctly different functions within different input-output pairs of a single linkage. Combined type and dimensional synthesis was demonstrated to be easily accessible and simply implemented with the continuous approximate synthesis method. Multi-modal function generator synthesis was determined to have a fundamentally limited and constrained application space. The original goal of developing the multi-modal synthesis algorithm was to develop a single function generating linkage which could replicate two or more competing functions to some suitable level of fidelity within different input-output pairs of the linkage. The goal of this technique was primarily to develop a linkage which could be optimised for a trade off of transmission angle and input-output function generation to reduce the analysis required after a function generating linkage is created, in order to ascertain how useful the linkage may be in practice. Unfortunately, through analysis, it was found that the only cases in which

this is possible involved a simple translation of the input-output curve of the linkage within one of the input-output pairs of the multi-modal function generation problem. The problem was further investigated in order to ascertain if the multi-modal algorithm could function effectively through the use of scaling factors in order to increase the sensitivity of the overall system to the perturbing function, but ultimately the limitations stand. Finally, multi-modal function generator synthesis was demonstrated within an example which may occur in the development of a machine in order to show its narrow, but effective, use case.

While additional research questions are levelled by the work presented herein, such as the kinetic and dynamic dimensional synthesis of a four-bar linkage, this thesis describes a completely unified and generalised method for the dimensional synthesis of four-bar function generating linkages.

4.1 Recommendations for Future Work

The work presented herein is applicable to the kinematic synthesis of linkages for function generating synthesis, though the methods presented herein are not necessarily useful exclusively for this problem. It may also be possible to develop more complete constraints for the minimisation alogrithm, in order to target linkages which have special properties, such as Grashof linkages, or in the case of the planar RRRR linkages, crank-rocker, or double-crank mechanisms. It is likely that the initial guess passed into the minimisation algorithm would be required to satisfy these constraints in order to allow for convergence, although this is a supposition which is presented without substantiation and one which will require additional investigation.

Future work expanding upon the methods previously discussed should also investigate the applicability for the CAAIOS methods to the kinetic and dynamic synthesis of linkages. Further to this, it remains to be seen whether or not the MMCAAIOS would be useful for the concurrent kinematic and kinetic (or dynamic) synthesis of function generating mechanisms. While the thesis focused entirely on the minimisation of kinematic error in the development of function generators, the kinetic and dynamic properties are

often points of concern for designers; using the MMCAAIOS algorithm, a designer may be able to target subtle modifications to the kinetic and dynamic profiles of a linkage in order to allow the linkage to satisfy design requirements which are driven by the availability of components, or the tolerances of the remainder of the machine in which the linkage is being placed.

Bibliography

[1] M. Ceccarelli, *Distinguished Figures in Mechanism and Machine Science: Their Contributions and Legacies Part 1*. Springer, 2007.

[2] R. Hartenberg and J. Danavit, *Kinematic Synthesis of Linkages*. New York: McGraw-Hill, 1964.

[3] F. Reuleaux, *The Kinematics of Machinery: Outlines of a Theory of Machines*. Courier Corporation, 1876.

[4] J. J. Uicker, G. R. Pennock, and J. E. Shigley, *Theory of Machines and Mechanisms*, 5th ed. Oxford University Press New York, 2017.

[5] J. M. McCarthy and G. S. Soh, *Geometric Design of Linkages*, 2nd ed. Springer Science & Business Media, 2010.

[6] O. Bottema and B. Roth, *Theoretical Kinematics*. Dover Publications, Inc., New York, NY, U.S.A., 1990.

[7] F. Freudenstein, "Design of Four-link Mechanisms," Ph.D. dissertation, Columbia University, New York, N.Y., USA, 1954.

[8] T. Goodman, "Discussion: Structural Error Analysis in Plane Kinematic Synthesis," *Journal of Engineering for Industry*, vol. 81, no. 1, pp. 21–22, 1959.

[9] J. A. Hrones and G. L. Nelson, *Analysis of the Four-bar Linkage: its Application to the Synthesis of Mechanisms*. Technology Press of the Massachusetts Institute of Technology, 1978.

[10] G. Mullineux, "Atlas of Spherical Four-bar Mechanisms," *Mechanism and Machine Theory*, vol. 46, no. 11, pp. 1811–1823, 2011.

[11] S. B. Torgal, K Tripathi, and N. Nagar, "Simulation of Software for Four-bar Function Generator Mechanism," in *11th National Conference on Machines and Mechanisms, Indian Institute of Technology, Delhi*, 2003.

[12] J. Angeles, A. Alivizatoss, and R. Akhras, "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.

[13] H.-P. Schröcker, M. L. Husty, and J. M. McCarthy, "Kinematic Mapping Based Assembly Mode Evaluation of Planar Four-bar Mechanisms," *Journal of Mechanical Design*, vol. 129, no. 9, pp. 942–929, 2007.

[14] R. Fox and K. Willmert, "Optimum Design of Curve-generating Linkages with Inequality Constraints," *Journal of Manufacturing Science and Engineering*, vol. 89, no. 1, pp. 144–151, 1967.

[15] E. Verstraten, "Cognate Linkages the Roberts–Chebyshev Theorem," in *Explorations in the History of Machines and Mechanisms*, Springer, 2012, pp. 505–519.

[16] R. Wu, R. Li, and S. Bai, "A Fully Analytical Method for Coupler-curve Synthesis of Planar Four-bar Linkages," *Mechanism and Machine Theory*, vol. 155, 2021.

[17] S. Bai and J. Angeles, "Coupler-curve Synthesis of Four-bar Linkages via a Novel Formulation," *Mechanism and Machine Theory*, vol. 94, pp. 177–187, 2015.

[18] R. Akhras and J. Angeles, "Unconstrained Nonlinear Least-square Optimization of Planar Linkages for Rigid-body Guidance," *Mechanism and Machine Theory*, vol. 25, no. 1, pp. 97–118, 1990.

[19] N Nariman-Zadeh, M Felezi, A Jamali, and M Ganji, "Pareto Optimal Synthesis of Four-bar Mechanisms for Path Generation," *Mechanism and Machine Theory*, vol. 44, no. 1, pp. 180–191, 2009.

[20] H. Martínez-Alfaro, "Four-bar Mechanism Synthesis for n Desired Path Points Using Simulated Annealing," in *Advances in Metaheuristics for Hard Optimization*, Springer, 2007, pp. 23–37.

[21] J. Chu and J. Sun, "Numerical Atlas Method for Path Generation of Spherical Four-bar Mechanism," *Mechanism and Machine Theory*, vol. 45, no. 6, pp. 867–879, 2010.

[22] S. Ebrahimi and P. Payvandy, "Efficient Constrained Synthesis of Path Generating Four-bar Mechanisms Based on the Heuristic Optimization Algorithms," *Mechanism and Machine Theory*, vol. 85, pp. 189–204, 2015.

[23] L. E. H. Burmester, *Lehrbuch der Kinematik: Für Studirende der Maschinen-Technik, Mathematik und Physik Geometrisch Dargestellt*. A. Felix, 1888, Leipzig, Germany.

[24] S. Deshpande and A. Purwar, "A Task-driven Approach to Optimal Synthesis of Planar Four-bar Linkages for Extended Burmester Problem," *Journal of Mechanisms and Robotics*, vol. 9, no. 6, 2017.

[25] C. Chen, S. Bai, and J. Angeles, "A Comprehensive Solution of the Classic Burmester Problem," *Transactions of the Canadian Society for Mechanical Engineering*, vol. 32, no. 2, pp. 137–154, 2008.

[26] J. Angeles and S. Bai, "Some Special Cases of the Burmester Problem for Four and Five Poses," in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2005, pp. 307–314.

[27] J. K. Pickard, J. A. Carretero, and J.-P. Merlet, "Appropriate Synthesis of the Four-bar Linkage," *Mechanism and Machine Theory*, vol. 153, p. 103 965, 2020.

[28] Z. A. Copeland and M. J. D. Hayes, "Coupler Pose Curves for Planar 4R Mechanisms," The 4^{th} International Workshop on Fundamental Issues, Applications, and Future Research Directions for Parallel Mechanisms, Manipulators, and Machines, 2020.

[29] D. M. Y. Sommerville, *Analytical Geometry of Three Dimensions*. Cambridge University Press, 1934.

[30] F. Freudenstein, "Structural Error Analysis in Plane Kinematic Synthesis," *Journal of Engineering for Industry*, vol. 81, no. 1, pp. 15–21, 1959.

[31] S. Tinubu and K. Gupta, "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.

[32] F. Freudenstein, "Approximate Synthesis of Four-bar Linkages," *Transactions of the ASME*, vol. 77, no. 8, pp. 853–861, 1955.

[33] M.J.D. Hayes, 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*, 1999, pp. 437–442.

[34] G. Sutherland and J. Siddall, "Dimensional Synthesis of Linkages by Multifactor Optimization," *Mechanism and Machine Theory*, vol. 9, no. 1, pp. 81–95, 1974.

[35] K. Gupta, "Design of Four-bar Function Generators with Mini-Max Transmission Angle," *Journal of Engineering for Industry*, vol. 99, no. 2, 1977.

[36] C. Barker and G.-H. Shu, "Three-position Function Generation of Planar Four-bar Mechanisms with Equal Deviation Transmission Angle Control," *Journal of Mechanisms Transmissions and Automation*, vol. 110, no. 4, pp. 435–439, 1988.

[37] C. Gosselin and J. Angeles, "Optimization of Planar and Spherical Function Generators as Minimum-defect Linkages," *Mechanism and Machine Theory*, vol. 24, no. 4, pp. 293–307, 1989.

[38] A. Rao, "Synthesis of 4-bar Function-generators Using Geometric Programming," *Mechanism and Machine theory*, vol. 14, no. 2, pp. 141–149, 1979.

[39] R. Alizade, F Freudenstein, and P. Pamidi, "Optimum Path Generation by Means of the Skew Four-bar Linkage," *Mechanism and Machine Theory*, vol. 11, no. 4, pp. 295–302, 1976.

[40] Z. Liu and J. Angeles, "Least-square Optimization of Planar and Spherical Four-bar Function Generator Under Mobility Constraints," *Journal of Mechanical Design*, vol. 140, no. 4, pp. 569–573, 1992.

[41] F.-C. Chen and H.-H. Huang, "Application of Taguchi Method on the Ttolerance Design of a Four-bar Function Generation Mechanism," in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 47446, 2005, pp. 727–733.

[42] G. Guj, Z. Dong, and M. Di Giacinto, "Dimensional Synthesis of Four Bar Linkage for Function Generation with Velocity and Acceleration Constraints," *Meccanica*, vol. 16, pp. 210–219, 1981.

[43] K. Waldron and E. Stevensen, "Elimination of Branch, Grashof, and Order Defects in Path-angle Generation and Function Generation Synthesis," *Journal of Mechanical Design*, vol. 101, no. 3, 1979.

[44] K. Russell and Q. Shen, "Expanded Spatial Four-link Motion and Path Generation With Order and Branch Defect Elimination," *Inverse Problems in Science and Engineering*, vol. 21, no. 1, pp. 129–140, 2013.

[45] K. Lakshminarayana and K. C. B. Raju, "Function-cognate Mechanisms: General Theory and Application," *Mechanism and Machine Theory*, vol. 20, no. 5, pp. 389–397, 1985.

[46] T. S. Todorov, "Synthesis of Four Bar Mechanisms as Function Generators by Freud–enstein-Chebyshev," *Journal of Robotics and Mechanical Engineering Research*, vol. 1, no. 1, pp. 1–6, 2015.

[47] K. Mehar, S. Singh, and R. Mehar, "Optimal Synthesis of Four-bar Mechanism for Function Generation with Five Accuracy Points," *Inverse Problems in Science and Engineering*, vol. 23, no. 7, pp. 1222–1236, 2015.

[48] R. S. Rose and G. N. Sandor, "Direct Analytic Synthesis of Four-bar Function Generators with Pptimal Structural Error," *Journal of Engineering for Industry*, 1973.

[49] N. R. Patel and V. B. Shah, "Synthesis of 4-bar Function Generator With Minimum Structural Error by Using Computational Approach," *Internation Journal of Advanced Engineering and Research Development*, vol. 2, no. 5, 2015.

[50] A. Jaiswal and H. Jawale, "Comparative Study of Four-bar Hyperbolic Function Generation Mechanism With Four and Five Accuracy Points," *Archive of Applied Mechanics*, vol. 87, pp. 2037–2054, 2017.

[51] K. Watanabe, "Approximate Synthesis of Plane Four-bar Mechanism for Function Generation," *Bulletin of JSME*, vol. 17, no. 109, pp. 951–958, 1974.

[52] A. F. Al-Dwairi, F. T. Dweiri, and O. M. Ashour, "A Novice-centered Decision-support System for Type Synthesis of Function-generation Mechanisms," *Mechanism and Machine Theory*, vol. 45, no. 9, pp. 1252–1268, 2010.

[53] S. Rao and A. Ambekar, "Optimum Design of Spherical 4-R Function Generating Mechanisms," *Mechanism and Machine Theory*, vol. 9, no. 3-4, pp. 405–410, 1974.

[54] W.-T. Lee and K. Russell, "Developments in Quantitative Dimensional Synthesis (1970–present): Four-bar Path and Function Generation," *Inverse Problems in Science and Engineering*, vol. 26, no. 9, pp. 1280–1304, 2018.

[55] M. Shariati and M. Norouzi, "Optimal Synthesis of Function Generator of Four-bar Linkages Based on Distribution of Precision Points," *Meccanica*, vol. 46, no. 5, pp. 1007–1021, 2011.

[56] X. Li, S. Wei, Q. Liao, and Y. Zhang, "A Novel Analytical Method for Function Generation Synthesis of Planar Four-bar Linkages," *Mechanism and Machine Theory*, vol. 101, pp. 222–235, 2016.

[57] A. Guigue and M. J. D. Hayes, "Continuous Approximate Synthesis of Planar Function-generators Minimising the Design Error," *Mechanism and Machine Theory*, vol. 101, pp. 158–167, 2016.

[58] B. Belzile and J. Angeles, "Reflections Over the Dual Ring—applications to Kinematic Synthesis," *Journal of Mechanical Design*, vol. 141, no. 7, 2019.

[59] J. Han and W. Liu, "On the Solution of Eight-precision-point Path Synthesis of Planar Four-bar Mechanisms Based on the Solution Region Methodology," *Journal of Mechanisms and Robotics*, vol. 11, no. 6, 2019.

[60] C. F. Van Loan and G Golub, *Matrix Computations*. The Johns Hopkins University Press, 1996.

[61] Z. Liu and J. Angeles, "Data-conditioning in the Optimization of Function-generating Linkages," in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, American Society of Mechanical Engineers, 1993, pp. 419–426.

[62] A. K. Cline, C. B. Moler, G. W. Stewart, and J. H. Wilkinson, "An Estimate for the Condition Number of a Matrix," *SIAM Journal on Numerical Analysis*, vol. 16, no. 2, pp. 368–375, 1979.

[63] G. H. Golub and C. Reinsch, "Singular Value Decomposition and Least Squares Solutions," in *Linear Algebra*, Springer, 1971, pp. 134–151.

[64] S. Nakamura, *Applied Numerical Methods in C*. Prentice-Hall, Inc., 1993.

[65] N. J. Higham, "Gaussian Elimination," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 3, no. 3, pp. 230–238, 2011.

[66] J. F. Grcar, "Mathematicians of Gaussian Elimination," *Notices of the AMS*, vol. 58, no. 6, pp. 782–792, 2011.

[67] A. S. Householder, *The Theory of Matrices in Numerical Analysis*. Dover Publications, Inc., 2013.

[68] A. S. Householder, "On the Convergence of Matrix Iterations," *Journal of the ACM (JACM)*, vol. 3, no. 4, pp. 314–324, 1956.

[69] R. Penrose, "On Best Approximate Solutions of Linear Matrix Equations," in *Mathematical Proceedings of the Cambridge Philosophical Society*, Cambridge University Press, vol. 52, 1956, pp. 17–19.

[70] E. H. Moore, "On the Reciprocal of the General Algebraic Matrix," *Bulletin of the American Mathematical Society*, vol. 26, pp. 294–295, 1920.

[71] R. Penrose, "A Generalized Inverse for Matrices," in *Mathematical Proceedings of the Cambridge Philosophical Society*, Cambridge University Press, vol. 51, 1955, pp. 406–413.

[72] M. J. D. Hayes, M. Rotzoll, A. E. Iraei, A. Nichol, and Q. Bucciol, "Algebraic Differential Kinematics of Planar 4R Linkages," in *2021 20th International Conference on Advanced Robotics (ICAR)*, IEEE, 2021, pp. 1060–1065.

[73] A. M. Rao, G. N. Sandor, D Kohli, and A. Soni, "Closed Form Synthesis of Spatial Function Generating Mechanism for the Maximum Number of Precision Points," *Journal of Engineering for Industry*, vol. 95, 1973.

[74] J. Klein, "A cad Add-in for Synthesis of RSSR Function Generators," Ph.D. dissertation, University of California, Irvine, 2005.

[75] S. Dhall and S. Kramer, "Computer-aided Design of the RSSR Function Generating Spatial Mechanism Using the Selective Precision Synthesis Method," in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, American Society of Mechanical Engineers, vol. 110, 1987, pp. 203–208.

[76] V. Cossalter, M. Da Lio, and A. Doria, "Optimum Synthesis of Spatial Function Generator Mechanisms," *Meccanica*, vol. 28, pp. 263–268, 1993.

[77] Z. Liu and J. Angeles, "Optimization of Planar, Spherical and Spatial Function Generators Using Input-Output Curve Planning," *Journal of Mechanical Design*, vol. 140, no. 116, pp. 915–919, 1994.

[78] J. J. Cervantes-Sánchez, H. I. Medellín-Castillo, J. M. Rico-Martínez, and E. J. González-Galván, "Some Improvements on the Exact Kinematic Synthesis of Spherical 4R Function Generators," *Mechanism and Machine Theory*, vol. 44, no. 1, pp. 103–121, 2009.

[79] W. Liu, H. Si, C. Wang, J. Sun, and T. Qin, "Dimensional Synthesis of Motion Generation in a Spherical Four-bar Mechanism," *Transactions of the Canadian Society for Mechanical Engineering*, 2023.

[80] R. L. Norton, *Design of Machinery*, 3rd ed. McGraw-Hill Higher Education, 2003.

[81] J. A. Nelder and R. Mead, "A Simplex Method for Function Minimization," *The Computer Journal*, vol. 7, no. 4, pp. 308–313, 1965.

[82] L. F. Shampine, "Vectorized Adaptive Quadrature in MATLAB," *Journal of Computational and Applied Mathematics*, vol. 211, no. 2, pp. 131–140, 2008.

[83] A. Guigue and M. J. D. Hayes, "Continous Approximate Synthesis of Planar Function-Generators Minimising the Design Error," in *Proceedings of the 8th CCTOMM Symposium on Machanisms, Machines, and Mechatronics, Carleton University, Ottawa, ON, Canada*, 2015.

[84] F. Freudenstein, "Approximate Synthesis of Four-bar Linkages," *Transactions of the American Society of Mechanical Engineers*, vol. 77, no. 6, pp. 853–859, 1955.

[85] J. Denavit and R. S. Hartenberg, "A Kinematic Notation for Lower-pair Mechanisms Based on Matrices," *Journal of Applied Mechanics*, vol. 22, 1955.

[86] M. Husty and M. Pfurner, *Mykinematics: Maple Library for Kinematics Using Study's Soma Coordinates*, Private communication during the 2019 IFToMM World Congress, Krakow, Poland, June 2019.

[87] J. Craig, *Introduction to Robotics: Mechanics and Control, Edition Three*. Pearson Education, 2005.

[88] M. Husty, A. Karger, H. Sachs, and W. Steinhilper, *Kinematik und Robotik*. Springer-Verlag, 2013.

[89] L.-W. Tsai, *Robot Analysis: the Mechanics of Serial and Parallel Manipulators*. John Wiley & Sons, 1999.

[90] L. Sciavicco and B. Siciliano, *Modelling and Control of Robot Manipulators*. Springer Science & Business Media, 2001.

[91] M. J. D. Hayes, M. L. Husty, and M. Pfurner, "Input-output Equation for Planar Four-bar Linkages," in *International Symposium on Advances in Robot Kinematics*, Springer, 2018, pp. 12–19.

[92] M. Rotzoll, M. J. D. Hayes, and M. L. Husty, "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.

[93] M. J. D. Hayes, M. Rotzoll, Q. Bucciol, and Z. A. Copeland, "Planar and Spherical Four-bar Linkage vi- vj Algebraic Input–Output Equations," *Mechanism and Machine Theory*, vol. 182, pp. 205–222, 2023.

[94] M. J. D. Hayes, M. Rotzoll, and M. L. Husty, "Design Parameter Space of Planar Four-bar Linkages," in *Advances in Mechanism and Machine Science: Proceedings of the 15th IFToMM World Congress on Mechanism and Machine Science 15*, Springer, 2019, pp. 229–238.

[95] H. S. M. Coxeter, *Regular Polytopes*. Dover Publication Inc., New York, NY, USA, 1973.

[96] M. Rotzoll, M. H. Regan, M. L. Husty, and M. J. D. Hayes, "Kinematic Geometry of Spatial RSSR Mechanisms," *Mechanism and Machine Theory*, vol. 185, 2023.

[97] M. Rotzoll, Q. Bucciol, and M. J. D. Hayes, "Algebraic Input-output Angle Equation Derivation Algorithm for the Six Distinct Angle Pairings in Arbitrary Planar 4R Linkages," in *20th Internations Conference on Advanced Robotics (ICAR)*, 2021.

[98] M. Rotzoll, M. J. D. Hayes, M. L. Husty, and M. Pfurner, "A General Method for Determining Algebraic Input-Output Equations for Planar and Spherical 4R Linkages," in *International Symposium on Advances in Robot Kinematics*, Springer, 2020, pp. 90–97.

[99] M. Rotzoll and M. J. D. Hayes, "A General Method for Determining Algebraic Input-Output Equations for the Slider-Crank and the Bennett Linkage," in *11th CCToMM Symposium on Mechanisms, Machines, and Mechatronics*, Ontario Tech University, Oshawa, ON, Canada, 2021.

[100] M. J. D. Hayes, M. Rotzoll, A. E. Iraei, A. Nichol, and Q. Bucciol, "Algebraic Differential Kinematics of Planar 4R Linkages," in *20th International Conference on Advanced Robotics (ICAR)*, pp. 6–10.

[101] Z. Copeland, M. Rotzoll, and M. J. D. Hayes, "Concurrent Type and Dimensional Continuous Approximate Function Generator Synthesis for All Planar Four-bar

Mechanisms," in *11th CCToMM Symposium on Mechanisms, Machines, and Mechatronics, eds. S. Nokleby, and P. Cardou, Ontario Tech University, Oshawa, ON, Canada*, 2021.

[102] Z. A. Copeland and M. J. D. Hayes, "Multi-modal Continuous Approximate Algebraic Input-Output Synthesis of Planar Four-Bar Function Generators," in *USCToMM Symposium on Mechanical Systems and Robotics*, Springer, 2022, pp. 10–19.

[103] Z. A. Copeland and M. J. D. Hayes, "Multi-modal Continuous Approximate Synthesis of Planar Four-bar Function Generators," *International Journal of Mechanisms and Robotic Systems*, vol. 5, no. 3, pp. 246–269, 2023.

[104] Maplesoft. "Methods Used by the Optimization Package." (2023), [Online]. Available: https://www.maplesoft.com/support/help/maple/view.aspx?path=Optimization (visited on 09/22/2023).

[105] T. J. Ypma, "Historical Development of the Newton–Raphson Method," *SIAM review*, vol. 37, no. 4, pp. 531–551, 1995.

[106] R. Alizade, F. C. Can, and Ö. Kilit, "Least Square Approximate Motion Generation Synthesis of Spherical Linkages by Using Chebyshev and Equal Spacing," *Mechanism and Machine Theory*, vol. 61, pp. 123–135, 2013.

[107] E. Waring, "Vii. Problems Concerning Interpolations," *Philosophical Transactions of the Royal Society of London*, no. 69, pp. 59–67, 1779.

[108] C Gosselin and J Angeles, "Mobility Analysis of Planar and Spherical Linkages," in *CIME.*, vol. 7, 1988, pp. 56–60.

[109] M. J. D. Hayes and M. Rotzoll, "Mobility Classification in the Design Parameter Space of Spherical 4R Llinkages," in *11th CCToMM Symposium on Mechanisms, Machines, and Mechatronics, Ontario Tech University, Oshawa, ON, Canada*, 2021.