An OOP based Four-Bar Mechanism Design and Simulation (4MDS) Software

A Thesis Presented by

Abhijit Toravi

to

The Graduate School

in partial fulfillment of the Requirements

for the degree of

Master of Science

In

Mechanical Engineering

Stony Brook University,

December 2013

Stony Brook University

The Graduate School

Abhijit Toravi

We, the thesis committee for the above candidate for the Master of Science degree,

hereby recommend

Acceptance of this thesis.

Dr. Anurag Purwar, Advisor,

Research Associate Professor, Mechanical Engineering Department

Dr. Q. Jeffrey Ge, Chairman of Thesis Committee

Professor, Mechanical Engineering Department

Dr. Shikui Chen, Inside Member

Assistant Professor, Mechanical Engineering Department

Dr. Sotirios Mamalis, Inside Member

Assistant Professor, Mechanical Engineering Department

This thesis is accepted by the Graduate School.

Charles Taber

Interim Dean of the Graduate School

Abstract of the Thesis

An OOP based Four-Bar Mechanism Design and Simulation (4MDS) Software

By

Abhijit Toravi

Master of Science

In

Mechanical Engineering

Stony Brook University

2013

Despite having been a research topic alive for a long time, various proposed solutions to planar mechanism design for the approximate motion synthesis have been non-linear in nature. In general, the algorithms proposed are computationally expensive and require dealing with the type and dimension synthesis separately.

The thesis presents an interactive software tool for unified type and dimension synthesis of planar four bar mechanism and its simulation.

It implements an efficient and linear algorithm that naturally extracts the geometric constraints of a motion and leads directly to the type and dimensions of a mechanism for motion generation.

The software has been developed using C++, OpenGL and Qt. The software uses an Object Oriented Programming (OOP) architecture which can be extended to other types of mechanism.

Table of Contents

List of Figures

ACKNOWLEDGEMENTS

Small things contribute to an achievement but an achievement is not a small thing.

It is a privilege to work under the guidance of Prof. Anurag Purwar, who has always motivated me to think with audacity and solve the problems with perseverance. He has been extremely enthusiastic and patient while attending to my work. With sincerity and honesty, I would like to express my profound sense of gratitude and indebtedness to my adviser and mentor.

I am obliged to to Prof. Jeff Ge, Prof. Shikui Chen and Prof. Sotiris Mamalis for their presence on the dissertation committee and their valuable suggestions in developing software for the present work.

I am grateful to Prof. Lin-Shu Wang: Graduate Program Director and Mrs. Mayra Santiago: Graduate Program Secretary of Mechanical Engineering Department for providing me infrastructure and facilities while working on this project.

I wish to acknowledge the loving care and attention of family; my mother Mrs. Arati, father Dr. Jitendra Toravi, sister Radhika and Minu; Uncle Dr. Avinash Lonkar and his family for their moral support. It is a privilege to be under the warm cover of these loved ones and it will be an understatement to say thanks to them.

My special thanks to lab mates- Xiangyun, Xin and Kartik, fellow batch mates- Bikash, Manisha and Yogesh as well as friends- Amit, Onkar and Nilesh for their excellent company and support during the thesis work.

Many thanks are due to all those help in many ways was inestimable value during the work.

I am thankful to Research foundation for SUNY, for the financial help provided by them during the project work.

1. Introduction

Planar Four Bar Linkages are simplest type of mechanisms found in machines. They are constructed from four links connected by joints with each other. One of the links is grounded known as ground link. Joints are either revolute or prismatic. There are six types of planar four bar linkages: RRRR, RRPR, RRRP, PRRP, PRPR, and RPPR.

1.1 Synthesis for Motion Generation

Linkage synthesis problems are generally grouped into three categories: motion generation for guiding a rigid body through a specified set of poses or task positions, path generation for guiding a point along a specified curve, and function generation for correlating the angular positions of two links. The Synthesis mode of this software focusses on kinematic synthesis of four bar linkages for motion generation.

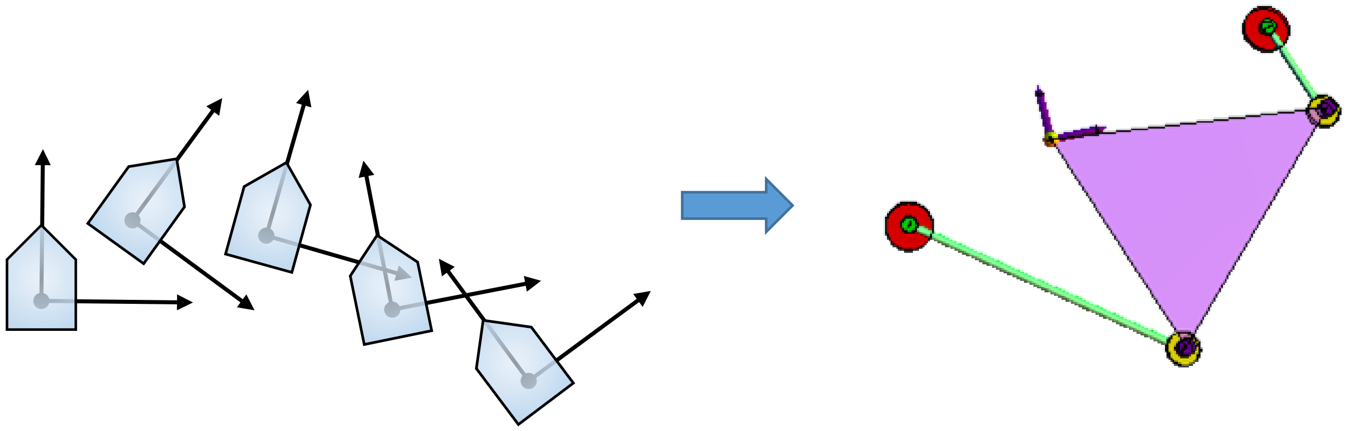


Figure 1: Motion Generation

Traditional approach to linkage synthesis is a two-step process: type synthesis/ number synthesis followed by dimensional synthesis. Number synthesis, i.e., the determination of the number of links and joints for a given value of F, completely independent of dimensional synthesis. The number synthesis gives rise to classification and enumeration of planar mechanisms. Dimensional synthesis aims to determine significant dimensions and starting position of Mechanism of preconceived type for specified task and prescribed performance.

1.2 Unified Type and Dimensional Synthesis

The software uses a task driven approach to unified type and dimensional synthesis of four bar mechanism described in [1]. Using the Image Space of planar displacements a class of quadrics, called Generalized- or G-manifolds are obtained, with eight linear and homogeneous coefficients as a unified representation for constraint manifolds of all four types of planar dyads, RR, PR, and PR, and PP. Given a set of image points that represent planar displacements, the problem of synthesizing a planar four-bar linkage is reduced to finding a pencil of G-manifolds that best fit the image points in the least squares sense. This least squares problem is solved using Singular Value Decomposition. The linear coefficients associated with the smallest singular values are used to define a pencil of quadrics. Additional constraints on the linear coefficients are then imposed to obtain a planar four-bar linkage that best guides the coupler through the given displacements. The result is an efficient and linear algorithm that naturally extracts the geometric constraints of a motion and leads directly to the type and dimensions of a mechanism for motion generation.

1.3 Simulation and Analysis

There are a lot of Software available in the market which perform simulation and analysis of the computed Mechanism. Kinematic Analysis involves position, velocity, and acceleration analysis as well as coupler curve animation and found in much of the major CAD software. However, there are very few tools available in the market which has ability to perform kinematic synthesis as well as analysis/ simulation of computed mechanism. Many of such analysis software has very limited or no synthesis capabilities. Software presented in the thesis focusses on both the aspects of kinematics i.e. synthesis and simulation of computed four bar mechanism.

The rest of the thesis is organized as follows. Chapter 2 involves a literature survey on planar four-bar design methods and tools. Chapter 3 discusses the fundamental concepts and theories related to four bar mechanism simulation and synthesis. It also presents the algorithm used to solve planar four bar Mechanism Synthesis problem. Chapter 4 presents the Object Oriented Architecture of the Software. Chapter 5 states design capabilities and guide to use the software tool developed. Few examples of simulation and synthesis are presented in Chapter 6. Conclusions and future work are part of Chapter 7.

2. Background

2.1 Introduction

This chapter involves a literature survey on planar four-bar design methods and tools. The focus of commercial CAD software has been on shape representation and manipulation, assembly and detail drawing creation, and integration with computer-aided manufacturing (CAM) and analysis packages. They do not help in automating the mechanism design process. However, there has been a great deal of academic research in the development of software systems for the synthesis of mechanisms. In the following section we discuss their approach to the problem.

2.2 Other software and their brief description

Modern implementation of design software for rigid body linkages include KINSYN III [2], LINCAGES [3, 4], Kihonge et al. [5], Spades [6], Sphinx [7], Sphinxpc [8], and Osiris [9]. The first comprehensive synthesis software for spatial linkages is Synthetica [10]. All of these mechanism design software focus on the dimensional synthesis with specialized mechanism topologies.

Work by Thomas R. Chase, Gary L. Kinzel, and Arthur G. Erdman [11] provides an historical overview of the developments in computer aided planar linkage synthesis in the time window of 1955 to 2012. Much of the following literature has been discussed in their work.

First software based approach to mechanism was taken by Freudenstein and Sandor [12]. Their work demonstrated the existence of Burmester point trios at which the rotation of the moving plane between precision positions is a constant multiple of the rotation of the circle-point center-point vector. They also developed equations for finite-motion generation and for both finite and infinitesimal path generation.

“KINSYN” [2] program developed by Kaufman was the first to utilize interactive approach, to enable the user to interact with the program while it was running. It featured a list of synthesis capabilities, which includes motion generation for two, three, four and five precision positions. The program was capable of designing linkages with slider joints in addition to revolute joints. It could also analyze tentative solutions to determine their Grashof type, circuit, branch and order of traveling through the prescribed positions, transmission angle, and acceleration. It could animate solutions on the display device, including multi-loop extensions to the basic four-bar solution. [11].

The early linkage synthesis programs were typically written in FORTRAN. However C and C++ is mostly used today.

SyMech [23] and WATT [24] used to be two well-known commercial grade software systems for planar mechanism synthesis, but are not available any more. Although, there have been some recent developments on providing mechanism design software to a broad class of users MechGen [26] provides a plug-in to Solidworks [27] for five position planar four-bar and six-bar linkage synthesis. WorkingModel 2D [28] and Ch Mechanism Toolkit [52]) are primarily a mechanism analysis and simulation tools.

A recent trend is the emergence of web based 3D CAD applications, such as TinkerCAD [30], 3DTin [31], PadCAD [32], and ShapeSmith [33].

SPADES is an interactive graphics based software package for Spatial Mechanism Design. It provides a platform for the synthesis of a mechanism that guides a body through either three or four prescribed positions in space. GUI of SPADES is shown in figure [3].

SPHINX (Software for Synthesizing Spherical 4R Mechanisms): displayed in figure [4] is a computer workstation-based software package developed by Professor Pierre Larochelle and Professor J. Michael McCarthy. It includes design of 4R linkage that guides a body either through four planar positions or through four spatial orientations. Although it is not much interactive and uses dialog boxes to input data.

OSIRIS can synthesize spherical and spatial mechanisms for 2, 3, 4 locations (orientations) and has filters to eliminate most unusable mechanism designs. There is also a module to help design spherical mechanisms for general spatial tasks. Figure [2] shows OSIRIS GUI.

Synthetica is a Java program that is used to synthesize, analyze and simulate spatial linkages. Its open architecture allows advanced users to add Java code to the system. Currently it runs on Windows, MacOSX, and Linux and is based on OpenGL. Its features include dimensional synthesis of spatial mechanisms, Kinematic analysis for serial and parallel linkages and Trajectory planning. It's GUI is shown in figure [5].

Ch Mechanism Toolkit developed by Softintegration Inc. is a software toolkit for analysis and design of mechanisms. The toolkit was developed using Ch, an embeddable C/C++ interpreter. It has synthesis capability limited to three positions.

Mechanism Generator 3.0 software reads a SolidWorks sketch of a planar 3R chain that reaches five task positions and computes a set of Watt 1 six-bar linkages that approximate movement through the task positions within specified tolerances.

SAM (Synthesis and Analysis of Mechanisms): shown in figure [6] from Artas Engineering [25] can perform at most three position synthesis for planar linkages. SAM offers a set of design wizards which helps to synthesize mechanisms for specific tasks, such as:

- Angle function generation (a minimum of 3 pairs of input/output angles are to be satisfied).
- 3-Position/angle synthesis of the coupler plane.
- Approximated straight line motion.
- Exact straight line motion.

GIM [13], shown in figure [7] – developed by COMPMECH research group in the University of the Basque Country is able of perform-

- Function generation.
- Precision point synthesis- 3,4,5 points
- Motion generation- 3 positions and additional constraint (on fixed or moving pivot), 4 Position synthesis.

It also animates the computed mechanism.

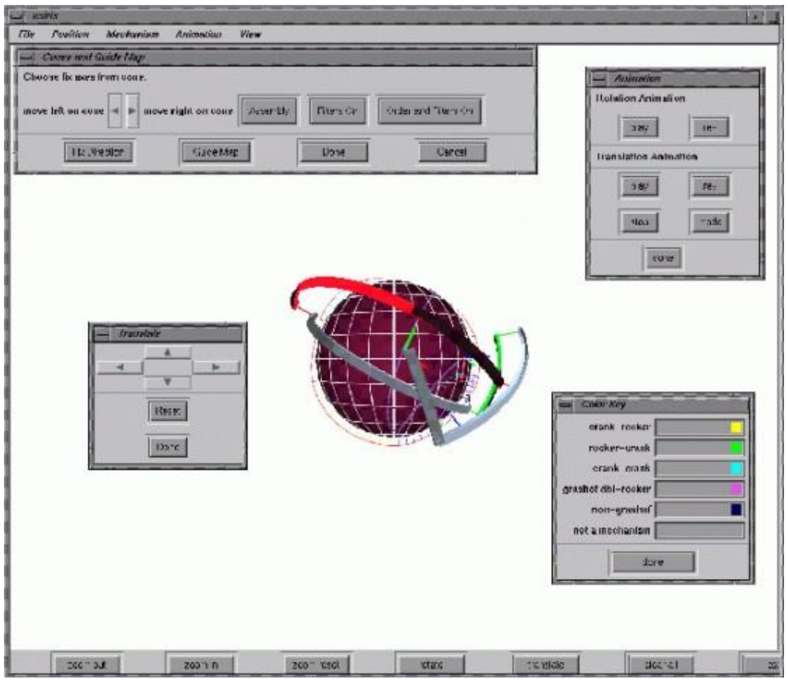
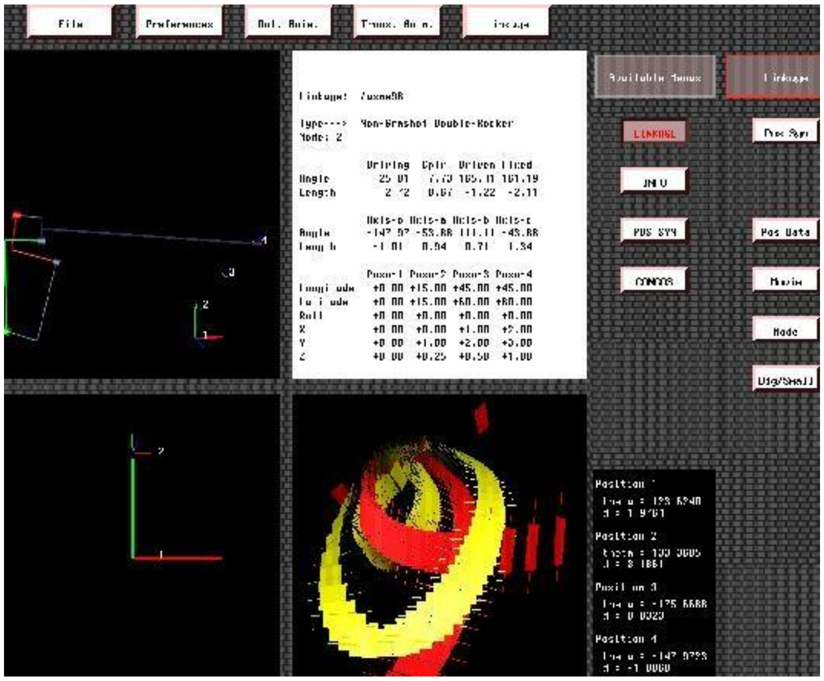


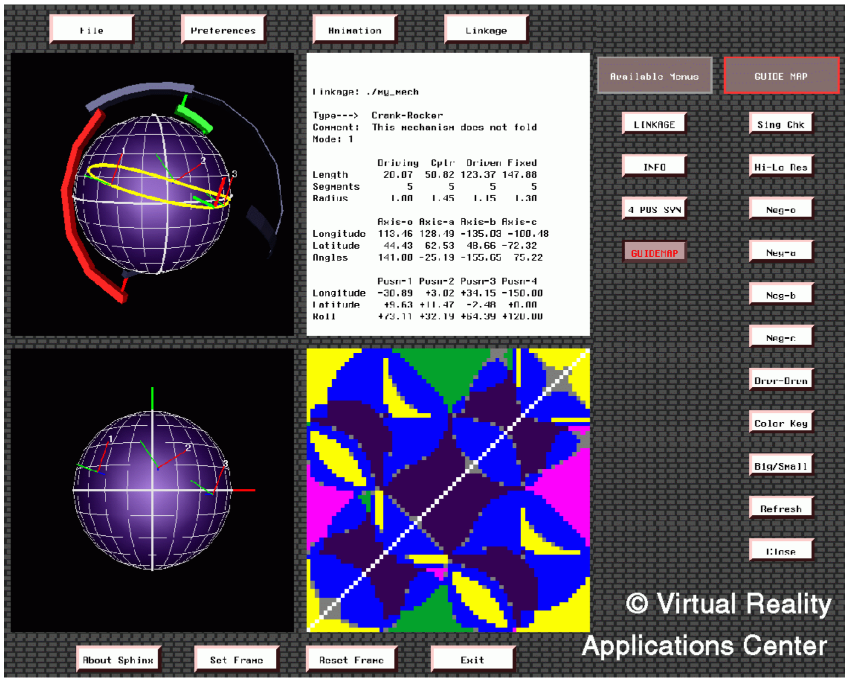
Figure 2: OSIRIS GUI



```
Linkage:  /uxmmDR

Type--->  Non-Grashof Double-Rocker
Mode: 2

           Driving  Cplr  Driven Fixed
Angle        25 01   7.73 165.11 161.19
Length        2 72   0.67  -1.22  -2.11

           Axis-o Axis-a Axis-b Axis-c
Angle      -147 97 -53.88 111.11 -43.88
Length       -1 01   0.94   0.71   1.34

           Posn-1 Posn-2 Posn-3 Posn-4
Longitude   +0 00 +15.00 +45.00 +45.00
Latitude    +0 00 +15.00 +60.00 +80.00
Roll        +0 00  +0.00  +0.00  +0.00
X           +0 00  +0.00  +1.00  +2.00
Y           +0 00  +1.00  +2.00  +3.00
Z           +0 00  +0.25  +0.50  +1.00
```

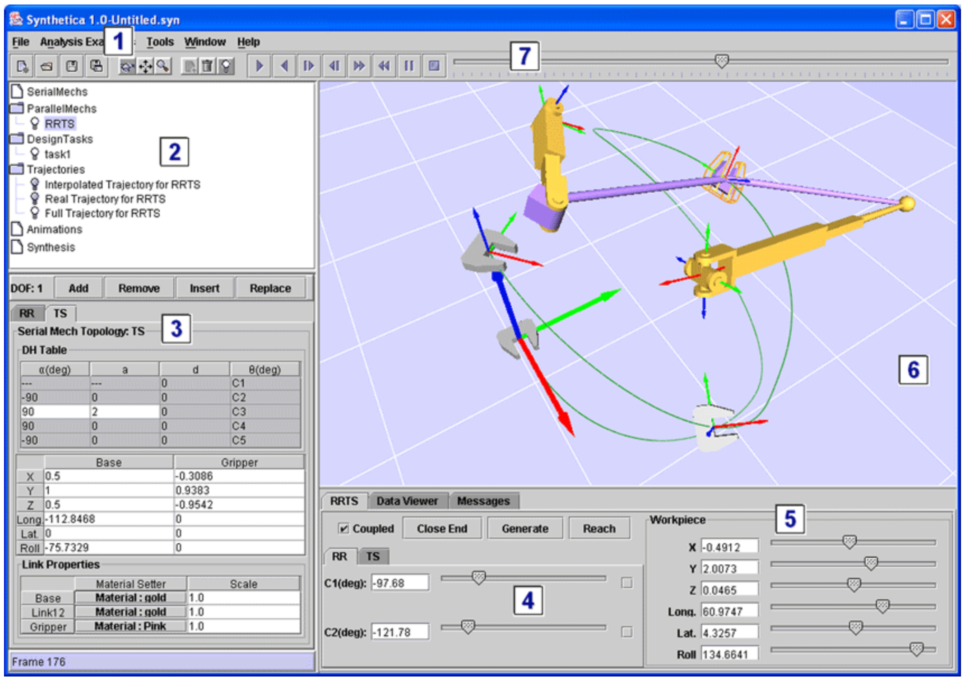
```
Position 1
 theta = 123 6240
 d = 1 9761
Position 2
 theta = 133 3605
 d = 8 1861
Position 3
 theta = -175 6888
 d = 0 0323
Position 4
 theta = -147 9723
 d = -1 0000
```

Figure 3: SPADES GUI



```
Linkage: ./my_mech

Type--->  Crank-Rocker
Comment:  This mechanism does not fold
Mode: 1

           Driving  Cplr  Driven Fixed
Length       20.07  50.82 123.37 147.88
Segments         5      5      5      5
Radius        1.00   1.45   1.15   1.30

           Axis-o Axis-a Axis-b Axis-c
Longitude  113.46 128.49 -135.03 -100.48
Latitude    44.43  62.53  48.66 -72.32
Angles     141.00 -25.19 -155.65  75.22

           Posn-1 Posn-2 Posn-3 Posn-4
Longitude  -30.89  +3.02 +34.15 -150.00
Latitude    +9.63 +11.47  -2.48  +0.00
Roll       +73.11 +32.19 +64.39 +120.00
```

Figure 4: SPHINX GUI



Serial Mech Topology: TS

DH Table

α(deg)	a	d	θ(deg)
---	---	0	C1
-90	0	0	C2
90	2	0	C3
90	0	0	C4
-90	0	0	C5

	Base	Gripper
X	0.5	-0.3086
Y	1	0.9383
Z	0.5	-0.9542
Long	-112.8468	0
Lat	0	0
Roll	-75.7329	0

Link Properties

	Material Setter	Scale
Base	Material : gold	1.0
Link12	Material : gold	1.0
Gripper	Material : Pink	1.0

[✓] Coupled

C1(deg): -97.68

C2(deg): -121.78

Workpiece

X	-0.4912
Y	2.0073
Z	0.0465
Long.	60.9747
Lat.	4.3257
Roll	134.6641

Figure 5: SYNTHETICA GUI

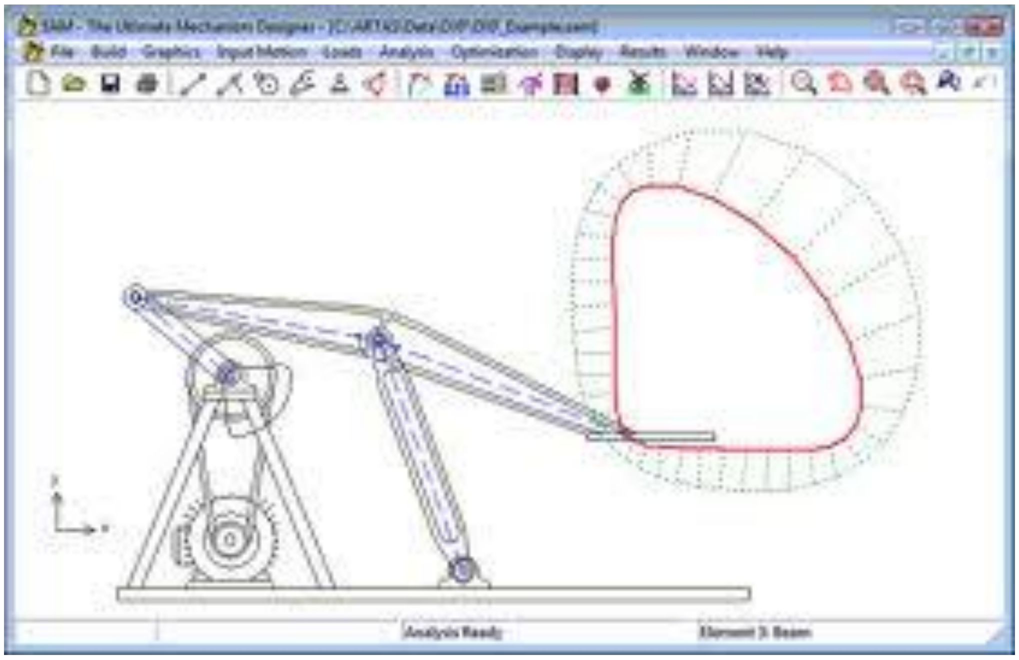
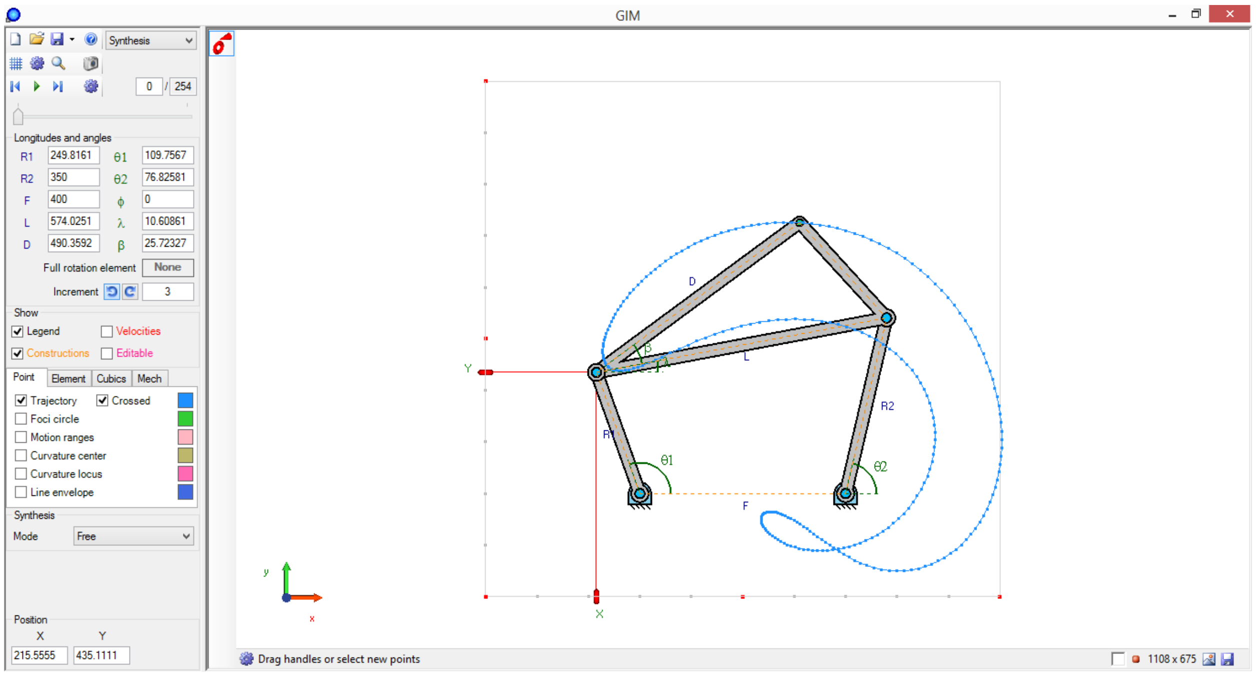


Figure 6: SAM GUI



0 / 254

Longitudes and angles

R1	249.8161	θ1	109.7567
R2	350	θ2	76.82581
F	400	ϕ	0
L	574.0251	λ	10.60861
D	490.3592	β	25.72327

Full rotation element: None

Increment: 3

Show

- [x] Legend
- [ ] Velocities
- [x] Constructions
- [ ] Editable

Point | Element | Cubics | Mech

- [x] Trajectory
- [x] Crossed
- [ ] Foci circle
- [ ] Motion ranges
- [ ] Curvature center
- [ ] Curvature locus
- [ ] Line envelope

Synthesis

Mode: Free

Position

X	Y
215.5555	435.1111

Drag handles or select new points

1108 x 675

Figure 7: GIM GUI

2.3 Technical approaches used in CAD software

As explained in [11] Motion generation has been approached using two fundamentally different approaches.

In the first, a large number of positions of the moving plane (coupler) are specified, and the best linkage which moves the coupler through the positions in an approximate sense is determined through a mathematical optimization process. In this approach, the coupler is unlikely to pass through any of the positions exactly. This approach is based more on optimization concepts than on kinematic concepts.

The second approach, is based on precision position synthesis. In this approach, the linkage is designed such that the coupler passes through a modest number of prescribed positions exactly. This approach usually results in multiple solutions. Optimization may be used ultimately to select the best linkage from the domain of possible linkages; however, optimization constitutes a secondary process.

In the precision position synthesis approach to motion generation, 2-5 positions of the coupler relative to the reference link can be specified. The two position problem yields three infinities of solutions. Even with five positions, multiple solutions can result.

2.4 Other Techniques used for synthesis and comparison with our approach:

This section refers to content in [1]The earliest approach to the motion synthesis problem was dealt with by Burmester [14], who posited that a given four-bar linkage can go through at most five positions exactly. Some key texts that describe state of the art as well as established methods and theory in kinematic synthesis of machines are by McCarthy and Soh [15], Sandor and Erdman [16], Hunt [17], Hartenberg and Denavit [18], Suh and Radcliffe [19]. Despite having been a research topic alive for a long time, various proposed solutions to planar mechanism design for the approximate motion synthesis have been non-linear in nature. In general, the algorithms proposed are computationally expensive and require dealing with the type and dimension synthesis separately. Using a kinematic mapping of planar kinematics, a general algebraic method for unified type and dimensional synthesis of planar four-bar linkages is used in the software, which reveals the geometric constraints implicit in the given motion via a linear, two-step process. The method is fast, efficient and provides type and dimensions of the mechanisms which can execute that motion.

3. Fundamentals

3.1 Introduction

This chapter introduces the fundamental concepts and theories related to four bar mechanism simulation and synthesis. It also presents the algorithm used to solve planar four bar Mechanism Synthesis problem. The concepts and the algorithm are discussed in detail in the work of Purwar et al. [1]. Circuit, Branch and Order defects are described by [20]. Grashof criteria is well defined in [21].

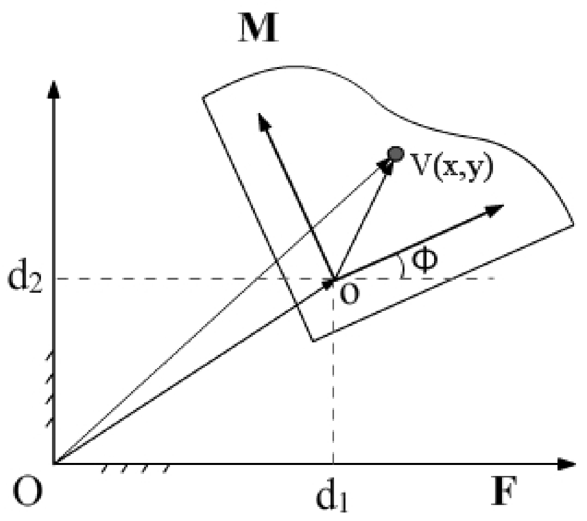
3.2 Quaternions:

Quaternions is a well-known concept in 3D computer graphics, computer vision etc. They were first described by Irish mathematician William Rowan Hamilton in 1843. The quaternions are a non-commutative extension of the complex numbers. Quaternions combine both translation and rotation in one format of numbers.

Quaternion is represented in the form of $ai+bj+ck+d$.
(a, b, c, d are real numbers; $i^2 = j^2 = k^2 = -1$, $ij = k$, $ji = -k$, $jk = i$, $kj = -i$, $ki = j$, $ik = -j$).
$a^2 + b^2 + c^2 + d^2$ is called the modulus of a quaternion.

3.3 Kinematic Mapping

Kinematic mapping approach was proposed by Ravani and Roth [22]. As described in [1], planar displacements in Cartesian Space are represented by points in a three dimensional projective space (called as image space in Planar Kinematics). Workspace constraints map into algebraic manifolds (constraint manifolds) in image space. In this way, a single degree of freedom motion of a planar mechanism is represented by the intersection curve of two Constraint Manifolds. The problem of motion approximation is transformed into an algebraic curve fitting problem in the image space, where various methods in approximation theory may be applied.



$$Z_1 = \frac{1}{2}\left(d_1 \sin\frac{\phi}{2} - d_2 \cos\frac{\phi}{2}\right),$$
$$Z_2 = \frac{1}{2}\left(d_1 \cos\frac{\phi}{2} + d_2 \sin\frac{\phi}{2}\right),$$
$$Z_3 = \sin\frac{\phi}{2},$$
$$Z_4 = \cos\frac{\phi}{2},$$

Figure 8: Quaternions

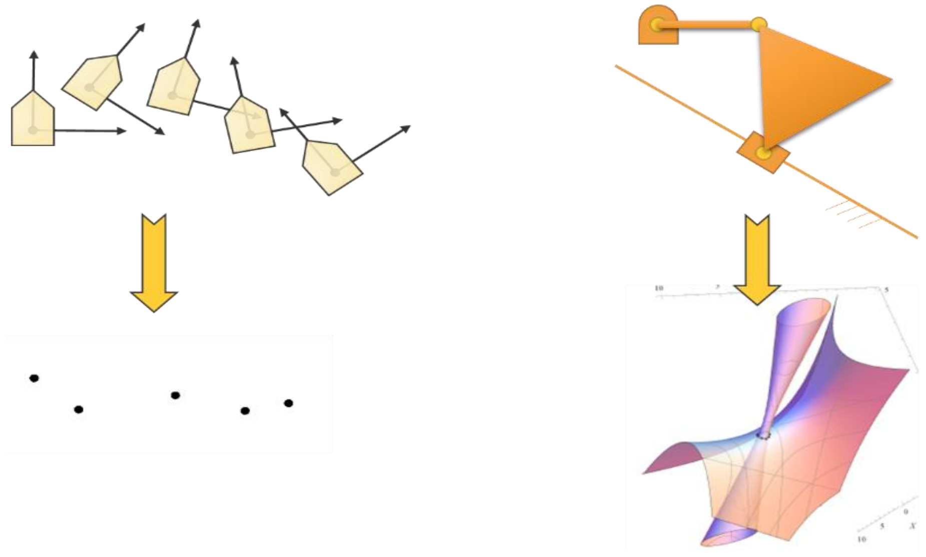


Figure 9: Kinematic Mapping

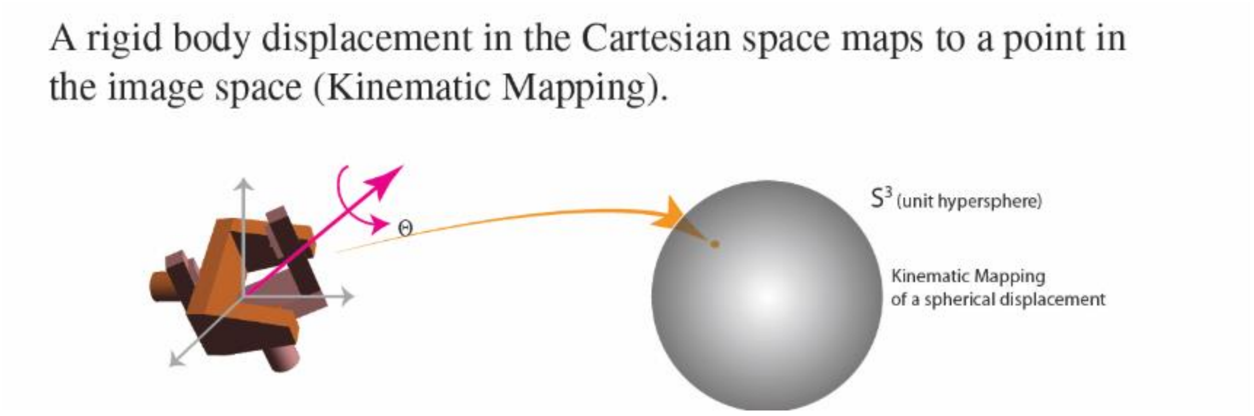

A rigid body displacement in the Cartesian space maps to a point in the image space (Kinematic Mapping).

Figure 10: Kinematic Mapping, displacement to point

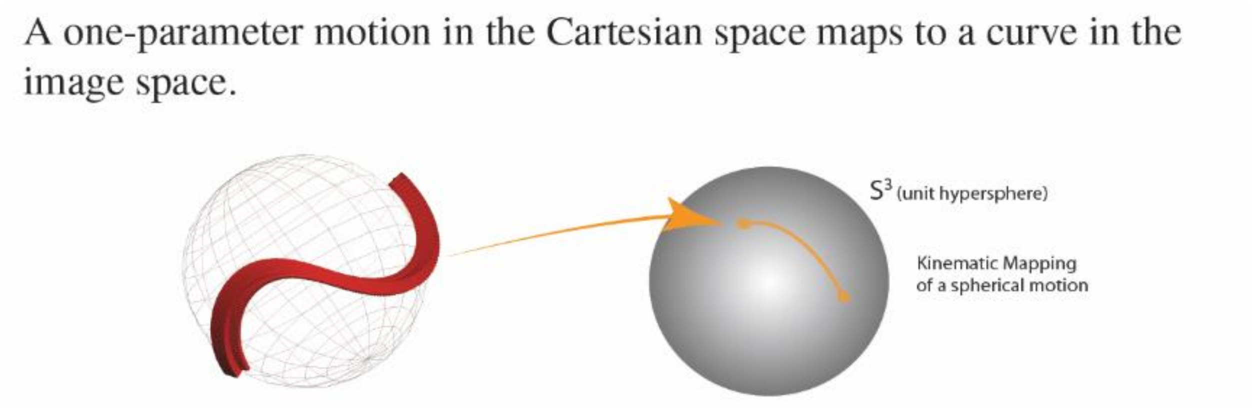

A one-parameter motion in the Cartesian space maps to a curve in the image space.

Figure 11: Kinematic Mapping, Motion to Curve

3.4 Transformation of points and lines

A planar displacement can be decomposed as the translation of a point in the moving body and rotation by an angle phi. M denotes co-ordinate frame attached to moving body and F be the fixed reference frame. Planar displacement can be represented as transformation of point or line co-ordinates from M to F.

Point co-ordinate transformation can be represented as X= [H]x

$$[H] = \begin{bmatrix} Z_4^2 - Z_3^2 & -2Z_3Z_4 & 2(Z_1Z_3 + Z_2Z_4) \\ 2Z_3Z_4 & Z_4^2 - Z_3^2 & 2(Z_2Z_3 - Z_1Z_4) \\ 0 & 0 & Z_3^2 + Z_4^2 \end{bmatrix}$$

Line transformation for the same transformation is given by

$$[\overline{H}] = \begin{bmatrix} Z_4^2 - Z_3^2 & -2Z_3Z_4 & 0 \\ 2Z_3Z_4 & Z_4^2 - Z_3^2 & 0 \\ 2(Z_1Z_3 - Z_2Z_4) & 2(Z_2Z_3 + Z_1Z_4) & Z_3^2 + Z_4^2 \end{bmatrix}$$

For details on kinematic mapping and transformation of point and lines, Please refer Ravani and Roth [22] and Bottema and Roth [34].

3.5 Constraining planar displacement

These are the geometric constraints associated with the mechanism. A mechanism is decomposed in two dyads. Therefore planar motions of rigid body (coupler) is subjected to one of the four types of geometric constraints.

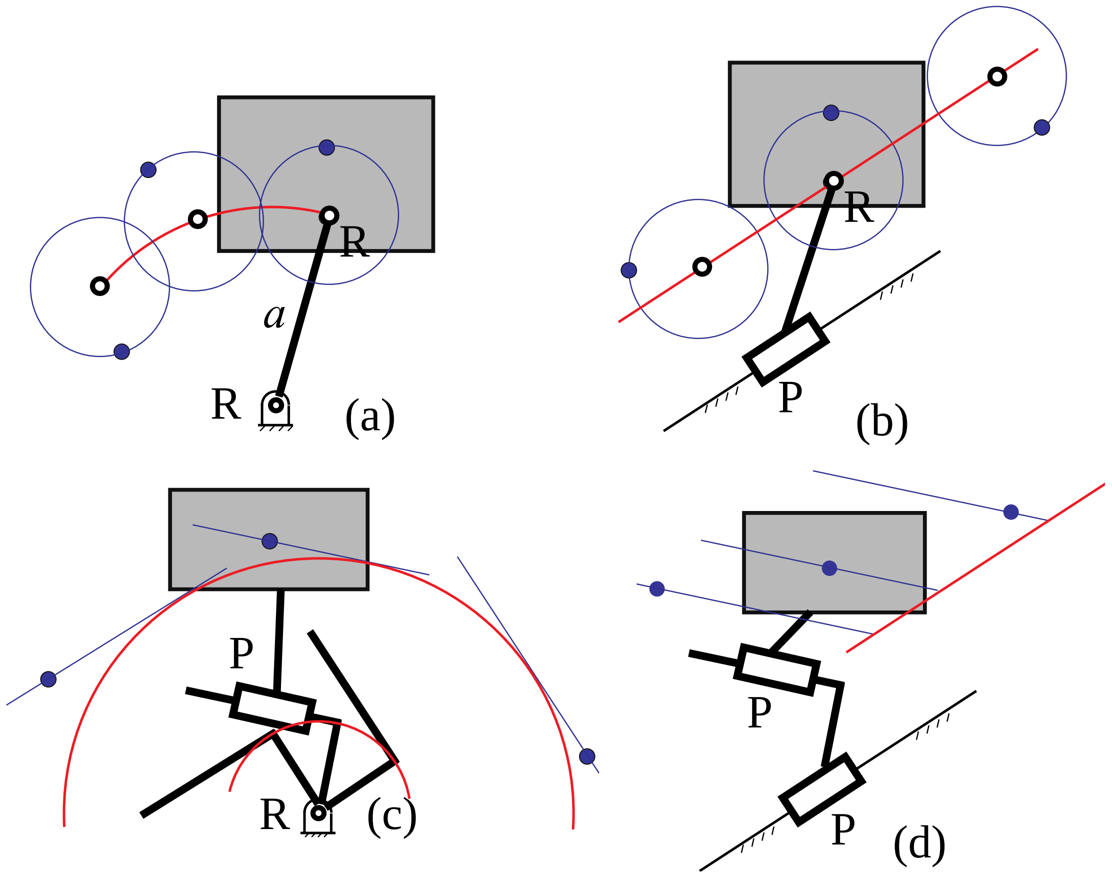


Figure 12: Constraints, various dyads

1. One of its points stays on a circle: - planar RR dyad.
2. One of its points stays on a line: - planar PR dyad.
3. One of its lines stays tangent to a circle: - planar RP dyad.

Planar Motion is subjected to any two constraints resulting in 1 DOF motion.

Following is a uniform representation of Line and a circle. Where center of a circle is given by *a= (a1, a2, a0)*. When *a0*= 0. It represents a line.

$$2a_1X_1 + 2a_2X_2 + a_3X_3 = a_0(\frac{X_1^2 + X_2^2}{X_3})$$

Following yields two lines which are *r*-distance away

$$a_1L_1 + a_2L_2 + a_0L_3 = \pm a_0 r\sqrt{L_1^2 + L_2^2}.$$

3.6 Algebraic fitting of pencil of G- manifolds

Consider the problem of fitting a pencil of G-Manifolds to a set of N Image space points, arranged in such a way that they define an image curve rather than a surface. The problem can be formulated as an over constrained linear problem [A]q= 0 [1].

$$[A] = \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} & A_{15} & A_{16} & A_{17} & A_{18} \\ \vdots & & & & & & & \vdots \\ \vdots & & & & \ddots & & & \vdots \\ \vdots & & & & & & & \vdots \\ A_{N1} & A_{N2} & A_{N3} & A_{N4} & A_{N5} & A_{N6} & A_{N7} & A_{N8} \end{bmatrix}$$

$$\begin{aligned} A_{i1} &= Z_{i1}^2 + Z_{i2}^2, \quad A_{i2} = Z_{i1}Z_{i3} - Z_{i2}Z_{i4}, \\ A_{i3} &= Z_{i2}Z_{i3} + Z_{i1}Z_{i4}, \quad A_{i4} = Z_{i1}Z_{i3} + Z_{i2}Z_{i4}, \\ A_{i5} &= Z_{i2}Z_{i3} - Z_{i1}Z_{i4}, \quad A_{i6} = Z_{i3}Z_{i4}, \\ A_{i7} &= Z_{i3}^2 - Z_{i4}^2, \quad A_{i8} = Z_{i3}^2 + Z_{i4}^2. \end{aligned}$$

3.7 Singular Value Decomposition

In linear algebra, the Singular Value Decomposition of an *N X 8* matrix *[A]* is a factorization of the form

[A]= [U] [S] [V]^T

Where *[U]* is *N x N* is orthonormal matrix, whose N columns, called left singular vectors of [A] ; [S] is an N x 8 rectangular diagonal matrix, whose values are square roots of eigenvalues of [A][A]^T ; [V]^T is an 8 x8 orthonormal matrix, whose 8 columns, called right singular vectors are eigenvectors of [A]^T[A]. (Golub [35]).

For a two-dimensional set of points (or surface data), if there is a perfect fit, there will be only one singular value that is equal to zero. This leads to a unique singular vector that defines the coefficients of a G-manifold that best fits the data. In this case, the matrix [A] is rank deficient by one. For a one-dimensional set of points (or curve data), if there is a perfect fit, there will be at least two singular values that are close to zero. In general, however, there may not be a perfect fit in which case, we will select three smallest singular values which correspond to the least square solution of [A]q = 0. In this case, they lead to a singular plane, which may be defined by orthonormal singular vectors v1, v2 and v3. Corresponding to three smallest singular values. Thus, we may represent a general unit singular vector in the singular plane by

$$q = v_1*a+v_2*b+ v_3.$$

The singular vector q given by this equation defines a pencil of G-manifolds as q varies. It needs emphasizing here that the three singular vectors corresponding to the smallest singular values may not necessarily represent the valid C-manifolds of a dyad because they may not satisfy the two conditions

Among the G-manifolds given by this equation, only those defined by the values of q that satisfy following equation are C-Manifolds

$$q_1q_6+q_2q_5-q_3q_4 = 0$$

$$2q_1q_7-q_2q_4-q_3q_5 = 0$$

To find the appropriate values of q, we formulate the least squares problem: as

$$e = sqrt (E)$$

$$E = [q_1q_6+q_2q_5-q_3q_4]^2+ [2q_1q_7-q_2q_4-q_3q_5]^2$$

3.8 Algorithm for synthesis

1. Convert given positions to quaternions.
   User inputs position in terms of Cartesian co-ordinates d_1, d_2, and phi

2. Compute Matrix A- using given quaternions.

3. Singular Value Decomposition- v_1, v_2, v_3 are singular vectors corresponding to singular values close to zero.

4. Compute Constraint Fitting Error before optimization (indication that G-Manifold does not define C-Manifold a valid dyad type even though they do intersect in the given image curve defined by the given positions).

5. Pencil of G-Manifolds given by $q= a*v_1+b*v_2+1*v_3$.

6. Impose additional constraints-, constraint fitting error must be zero and get values of a, b.

7. Zero, two or four real pairs are obtained. - C manifolds are obtained using those real roots.

8. Identify type of dyads according to C-Manifolds.

9. Find Dyad parameters and create dyads.

10. Choose two dyads at a time to form mechanism and obtain coupler curve.

3.9 Loop closure equations:

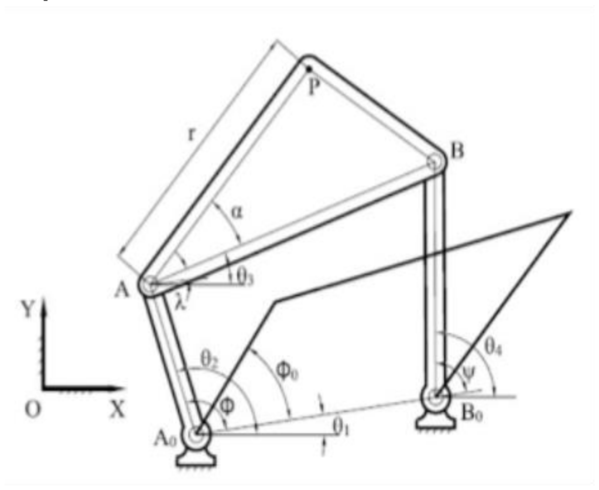


Figure 13: Planar four bar Mechanism- RRRR

Planar four bar mechanism has degree of freedom 1. Actuator is connected to the driving dyad and unknowns such as coupler position and orientation, output dyad can be represented as a function of input angle or Moving pivot location of driving dyad.

Input-Output relationship can be found out using loop closure equations:

$r_1e^{i*theta1}+r_4e^{i*theta4} = r_2e^{i*theta2}+r_3e^{i*theta3}$

Where r_1, r_2, r_3 and r_4 denote link lengths of ground link, input link, floating link and output link. 'Theta's' are their corresponding angles.

3.10 Grashof criteria:

The sum of the shortest (s) and longest (l) link of a planar four-bar linkage cannot be greater than the sum of remaining two links (p & q) if there is to be continuous relative motion between the links.

Using these definitions for s, l, p, and q, it is possible to specify the following relationships:

If l+s > p+q then the linkage is a Non-Grashof type

If l+s =< p+q then the linkage is a Grashof type

Non-Grashof type linkages have only one circuit and two branches. Since there are two branches, the linkage must pass through a stationary configuration in order to pass between those branches. A stationary configuration occurs when the transmission angle is equal to 0^0 or 180^0.

Grashof type linkages are characterized by having two circuits. If the shortest link, s, is the ground link, then this type of linkage is referred to as a crank-crank since both sides of the linkage can complete full rotations (often useful for function generation mechanisms). Also, possible is the crank-rocker in which one side of the linkage is capable of full rotation while the other side is not.

A circuit is defined as all possible orientations of the links which can be realized without disconnecting any of the joints. If a circuit contains stationary configurations, a branch is defined as a continuous series of positions on the circuit between two stationary configurations. Using these definitions, four-bar linkages satisfying the Grashof criteria have two circuits while those that do not have one. The single circuit of a non-Grashof four-bar linkage has two branches. A crank-rocker or double-crank has two circuits, but since they do not contain stationary configurations, branching is irrelevant. The two circuits of a rocker-crank or Grashof double-rocker both contain two branches.

3.11 Circuit defect

It is well known that the circuit of a mechanism as the set of all possible orientations of links that can be realized without disconnecting any of the joints. Also known as assembly configuration.

If a mechanism must be disassembled to move from one position to another, the two positions reside on separate circuits. The circuits of a mechanism are independent of the choice of driving link(s).

3.12 Branch defect

If a mechanism must be disassembled to move from one position to another, the two positions reside on separate circuits. The circuits of a mechanism are independent of the choice of driving link(s).

3.13 Order defect:

The order defect occurs when the coupler moves through the task positions in the wrong order as the input link is driven in a uniform direction. While an individual dyad cannot suffer from branch and circuit defects, it can have an order defect, which is relevant only if it is the driving dyad since the driven dyad that completes the four-bar linkage need not exhibit order. Of the three defects, only order can be assessed by examining the potential driving dyad, and thus, only order can be addressed during, and even prior to, synthesizing a single dyad.

4. OOP Architecture

4.1 Introduction

This chapter gives an overview of the architecture and describes the top level software components and their interaction. It also enlists some of data members and member functions.

It is well known that, four major features of object oriented language are Encapsulation, Data Abstraction, Polymorphism and Inheritance. Encapsulation is also known as data hiding. Non-essential details (algorithm, working principle) are hidden from user. Abstraction is implemented by creating classes and storing data in the form of attributes of objects of that class. Restricted access is provided to the data (attributes can be accessed by calling functions of that class). Inheritance refers to extending the functionality of the class. Polymorphism means taking more than one form at a time. This is achieved as compile time polymorphism or runtime polymorphism.

The software presented in the work follows all the object oriented principles. It is extend-able to spherical and spatial cases and other types of linkages. It is developed using C++ on Visual Studio IDE. The interface is developed using Qt 4.8.4. Rendering is done using OpenGL. It uses 'ARMADILLO' for Linear Algebra Calculations. It consists of several sections in terms of functionality. The code written is extend-able and re-usable in a way that one just needs to import the definitions of a certain class and can use all the functionality or methods provided in the class. In the following section we will discuss some important classes and the functionality they provide.

4.2 About Qt

Qt: Qt is a cross-platform application and UI framework for developers using C++ or QML, a CSS & JavaScript like language. Qt Creator is the supporting Qt IDE.
Qt, Qt Quick and the supporting tools are developed as an open source project governed by an inclusive meritocratic model. Qt can be used under open source (LGPL v2.1) or commercial terms.

4.3 About Armadillo

Armadillo is a C++ linear algebra library (matrix maths). The Armadillo library can be distributed and/or modified under the terms of the Mozilla Public License 2.0 (MPL)

4.4 User Interface (Front End)

This section introduces the components of user interface and briefly describes their functionality.

1. Main Window (AlgebraicFitting):
   It is a top level class. It inherits QMainWindow class from Qt Architecture. It also acts as link between other independent classes.
   Data Members: objects or pointers to objects of major classes.
   Member Functions: signal and slot connections.

2. Cartesian Space Window (Widget_Mech):
   This window is used to display the given positions, the animation of the mechanism. This widget acts as an active widget and used for interaction with user. It accepts data in the form of mouse movements for inputting positions, editing the plotted positions, inputting dyads and editing dyads. It inherits QWidget class.
   Data Members: Objects of elements related to image space.
   Member Functions: draw functions and mouse events.

3. The Image Space Window (Widget_Image):
   This window displays image space points corresponding to the positions in image space, the constraint manifold as well as the image curve projected on the hyper-plane. It is child of QWidget class.
   Data Members: Objects of elements related to image space.
   Member Functions: draw functions.

4. Synthesis Frame(frame_syntehsis):
   The frame contains checkboxes allowing the user to view all the synthesized dyads. Choosing two dyads out of synthesized dyads form a mechanism. It is connected with class Branch, class Motion, Widget_Image and Widget_Mech using signal and slot architecture of Qt. It inherits QFrame.

5. Analysis Frame(frame_analysis):
   The frame contains checkboxes allowing the user to choose type of object to input on the screen. It contains checkboxes corresponding to Dyads- RR, RP and PR and Moving Frame. Its parent class is QFrame.

6. Saving, Reloading and Output:
   This functionality allows the user to save, reload or output the text file or screenshot of the widgets. These options can be found under drop down list of

File. A text file is generated which specifies the details of the mechanism obtained as a result of synthesis. They use QFileDialog.

7. Menu Bar and other Tool Bars:
   Contains objects of type QAction connected to functions through signals and slots.

4.5 Back End

1. Mechanism Synthesis Class:
   It Inherits QObject class.
   It contains the algorithm for synthesis (equal to or more than 5 positions).
   Data Members: vectors and matrices (as defined in armadillo).
   Member Functions: Algorithm for synthesis.

2. Class Dyad : (Abstract)
   Works as abstract base class for dyad classes RR, RP and RP. Contains object of class ‘Link’, Fixed Pivot and Moving Pivot.
   Data Members: Object of type link.
   Member Functions: pure virtual draw functions, functions common to all dyads.

3. Class Branch:
   It works as a link between UI, mechanism class and dyad classes. It is derived from type QObject and connected to other classes with Qt signal and slot architecture.
   Data Members: Pointer to Objects of type mechanism.
   Member Functions: Algorithm for synthesis, slots connecting other classes.

4. Planar Four Bar Mechanism Class:
   It contains object corresponding to floating link, coupler link and coupler point and functions related to simulation of Mechanism. As it is a class, functionality can be extended to displaying more than one mechanism on the screen by creating multiple objects of this class.
   Data Members: pointer to dyads, coupler etc.
   Member Functions: Functions related to editing & simulation of mechanism.

5. Class Motion:
   This class is contains objects of type positions, corresponding image space points and functions related to them.

Data Members: Objects of type positions (Cartesian and image space).
Member Functions: Functions related to positions- add, delete, edit.

6. Classes corresponding to dyads:
   There are classes for individual dyads: RR, RP and PR. They are derived from abstract class Dyad. Each class has its own implementation, constraint manifold object and draw functions.
   Data Members: dyad parameters (which are unique) i.e. constraints and joints.
   Member Functions: set and get parameters, type and joint types.

7. Constraint Manifold Classes:
   There are two classes that fall into this category, Hyperboloid and Hyperbolic corresponding to dyads. RR dyad manifold corresponds to Hyperboloid while RP and PR dyads correspond to Hyperbolic Paraboloid.
   Data Members: related to constraint manifolds.
   Member Functions: construct manifolds.

There are other classes (helper classes) that assist these main classes to achieve certain functionality.

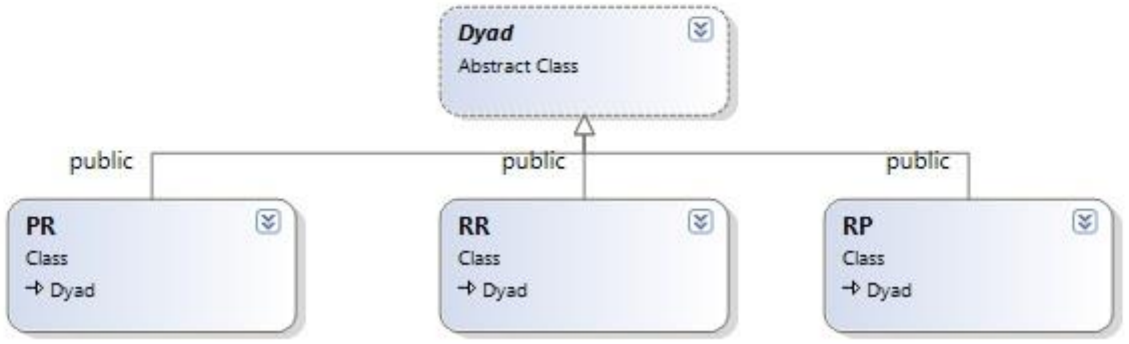


Figure 14: Dyad- Class Diagram

5. Design Capabilities

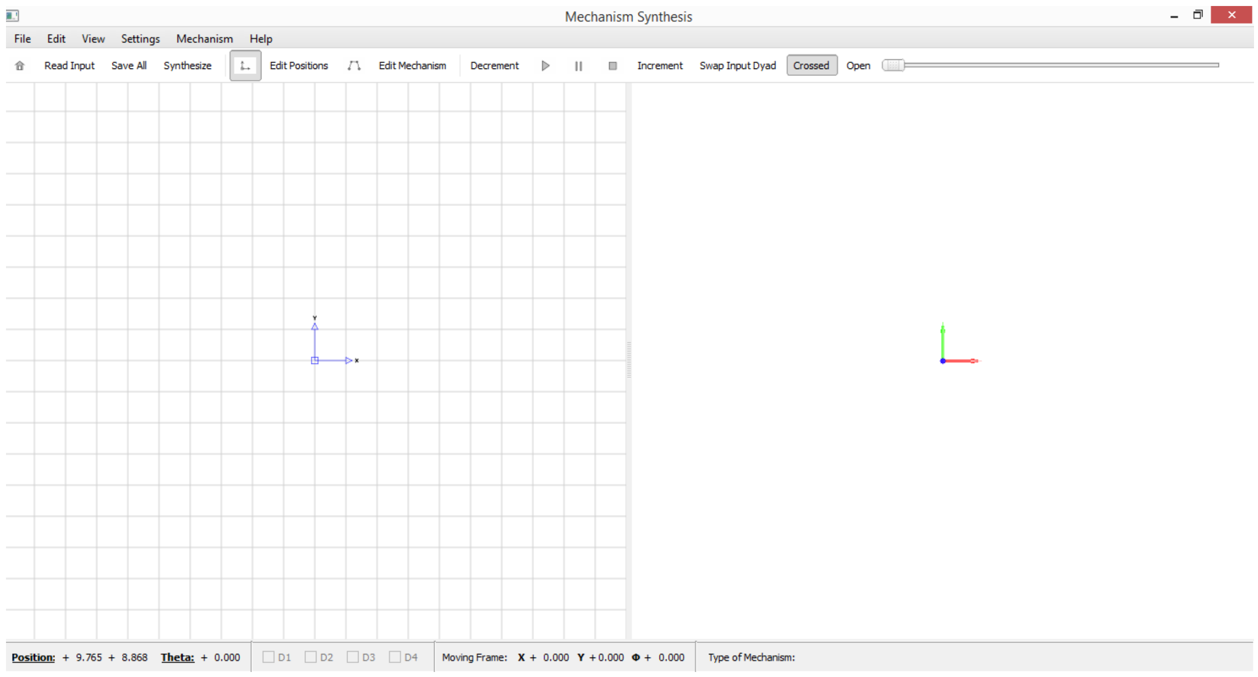
5.1 Introduction

This chapter discusses the functionality of the software. The main intention behind the software was to make it more interactive. It is not intended to be just a research tool but also for academic purposes and non-expert users. So, emphasis is given on mouse movements and animation rather than displaying only the numerical data.

Software has four modes: Input Positions, Edit Positions, Input Mechanism and Edit Mechanism. Its capabilities include simulation and synthesis. It is able to display all six types of four bar mechanisms by combining RR, RP and PR dyads. Driven link is automatically chosen. It is possible to interactively edit and delete the positions, dyads and location of moving frame.

Algorithm used for synthesis is described later in the chapter and works for five or more than five positions.

5.2 Main Window



Position: + 9.765 + 8.868 Theta: + 0.000

Moving Frame: X + 0.000 Y + 0.000 Φ + 0.000

Type of Mechanism:

Figure 15: Main Window of Software

5.3 Simulation-

1. Input Dyads as constraints- Click and drag for constrained input
2. Mechanism is assembled as soon as two dyads are inputted.
3. Adding the moving frame- Moving frame may be added in order to show coupler path.
4. Pre-Plots the coupler curve- Coupler path is pre-computed.
5. Play, pause and stop the animation.
6. Edit Mechanism interactively- Mechanism parameters can be changed.

5.4 Synthesis

1. Input Positions: interactively or through text file
2. As soon as fifth position is entered, Mechanism is synthesized. Coupler curve is plotted.
3. Interactively edit input positions.
4. New Mechanism is computed as soon as positions are changed.
5. Coupler curve changes as mechanism is changed.
6. Synthesized Mechanism can also be edited interactively.

5.5 Other Functions

1. Turn grid on and off.
2. Selective display (Cartesian and Image Space) - Using the handle to drag display widgets.

5.6 Validation of Synthesis Results:

As soon as coupler curve is plotted, the program samples out 5 positions.

The sampled positions can be read as an input and the original mechanism is recreated (also up to 5 more mechanisms are obtained which pass though those positions).

6.Examples

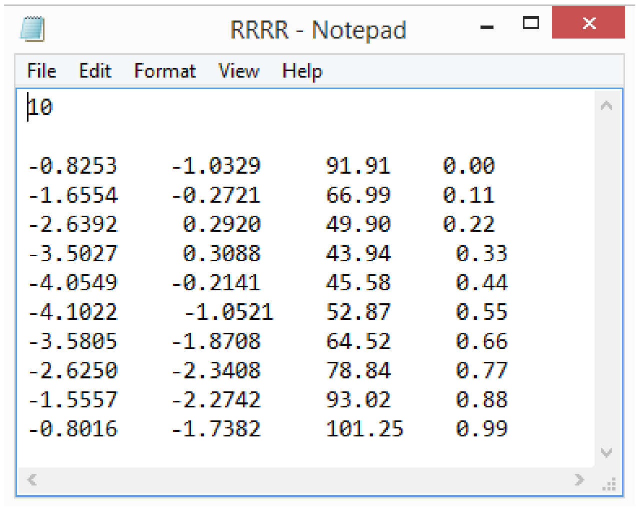
6.1 Introduction

This chapter presents a number of examples of simulation and synthesis, generated by the software.

6.2 Synthesis

1. 10 Positions (reading from input file).

In this example, user reads 10 positions from an input file. We represent a step by step process to obtain the solution. Figure [16] shows the output file through which input positions are read. The first, second and third column represents x, y co-ordinates and orientation of moving frame for the positions, the same positions in the Cartesian space are realized in [17]. Figure [18] shows matrix 'A' as described in the algorithm for synthesis. It is of dimension 10 x 8. SVD results are calculated in displayed in [19] showing right singular vectors obtained before optimization.



```
10

-0.8253    -1.0329     91.91    0.00
-1.6554    -0.2721     66.99    0.11
-2.6392     0.2920     49.90    0.22
-3.5027     0.3088     43.94     0.33
-4.0549    -0.2141     45.58     0.44
-4.1022      -1.0521   52.87     0.55
-3.5805    -1.8708     64.52     0.66
-2.6250    -2.3408     78.84     0.77
-1.5557    -2.2742     93.02     0.88
-0.8016    -1.7382     101.25    0.99
```

Figure 16: Example 1, 10 Positions-input file

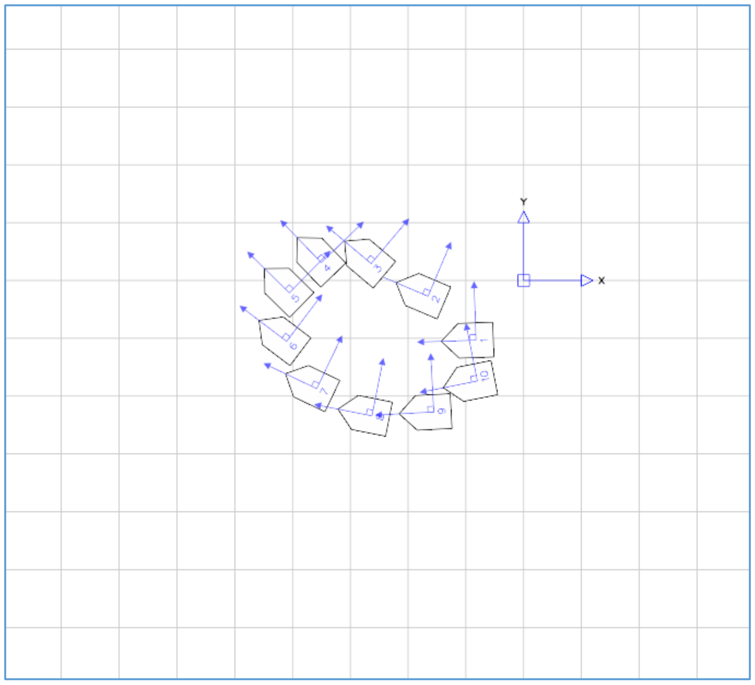
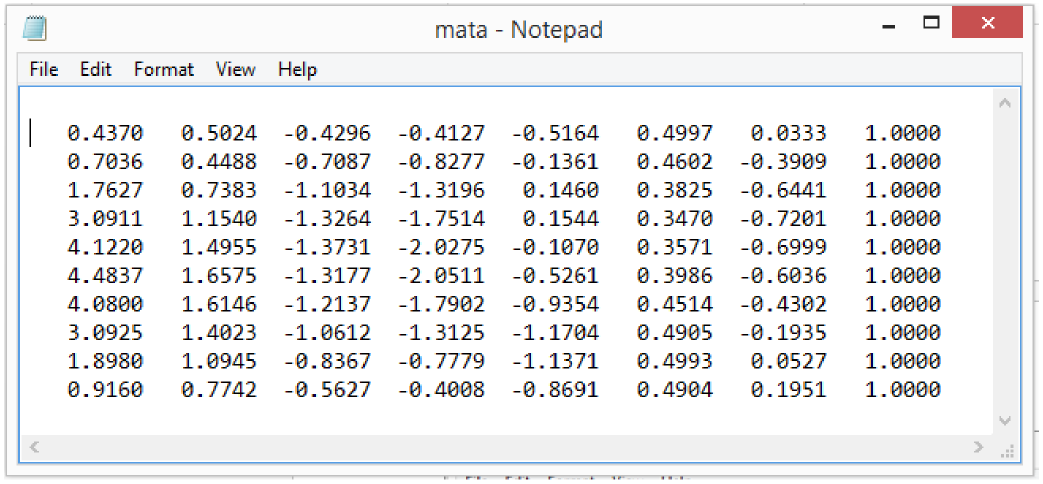
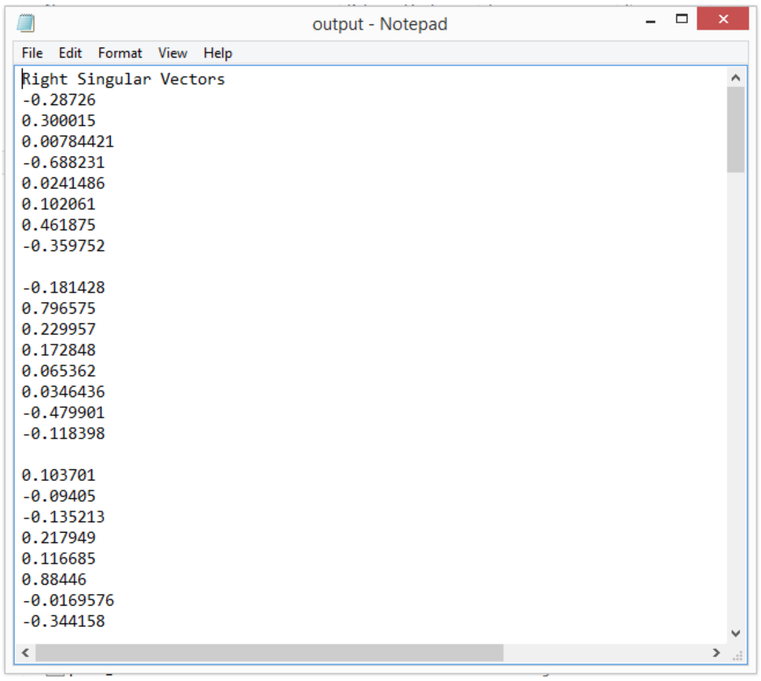


Figure 17: Example 1, 10 positions- Cartesian



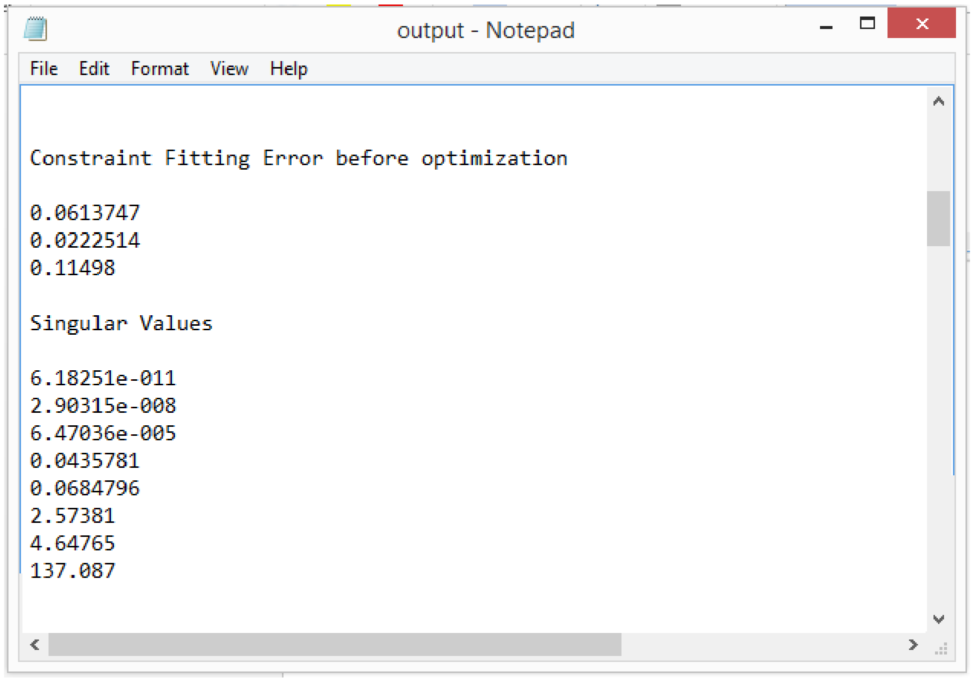
```
  0.4370   0.5024  -0.4296  -0.4127  -0.5164   0.4997   0.0333   1.0000
  0.7036   0.4488  -0.7087  -0.8277  -0.1361   0.4602  -0.3909   1.0000
  1.7627   0.7383  -1.1034  -1.3196   0.1460   0.3825  -0.6441   1.0000
  3.0911   1.1540  -1.3264  -1.7514   0.1544   0.3470  -0.7201   1.0000
  4.1220   1.4955  -1.3731  -2.0275  -0.1070   0.3571  -0.6999   1.0000
  4.4837   1.6575  -1.3177  -2.0511  -0.5261   0.3986  -0.6036   1.0000
  4.0800   1.6146  -1.2137  -1.7902  -0.9354   0.4514  -0.4302   1.0000
  3.0925   1.4023  -1.0612  -1.3125  -1.1704   0.4905  -0.1935   1.0000
  1.8980   1.0945  -0.8367  -0.7779  -1.1371   0.4993   0.0527   1.0000
  0.9160   0.7742  -0.5627  -0.4008  -0.8691   0.4904   0.1951   1.0000
```

Figure 18: Example 1, 10 positions matrix A



```
Right Singular Vectors
-0.28726
0.300015
0.00784421
-0.688231
0.0241486
0.102061
0.461875
-0.359752

-0.181428
0.796575
0.229957
0.172848
0.065362
0.0346436
-0.479901
-0.118398

0.103701
-0.09405
-0.135213
0.217949
0.116685
0.88446
-0.0169576
-0.344158
```

Figure 19: Example 1, 10 positions, right singular vectors



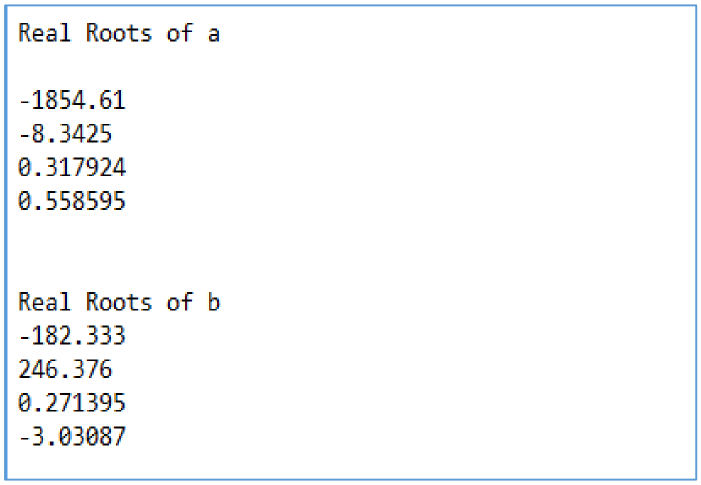
```
Constraint Fitting Error before optimization

0.0613747
0.0222514
0.11498

Singular Values

6.18251e-011
2.90315e-008
6.47036e-005
0.0435781
0.0684796
2.57381
4.64765
137.087
```

Figure 20: Example 1, 10 positions: CFE and singular values

Three singular values with values close to zero are obtained. Corresponding singular vectors are shown in the image above. It can be seen that the constraint fitting error is not equal to zero for the singular vectors obtained.

Pencil of G-Manifolds is represented by equation

$$q= (av_1+bv_2+v_3) / (a^2+b^2)^{0.5}$$

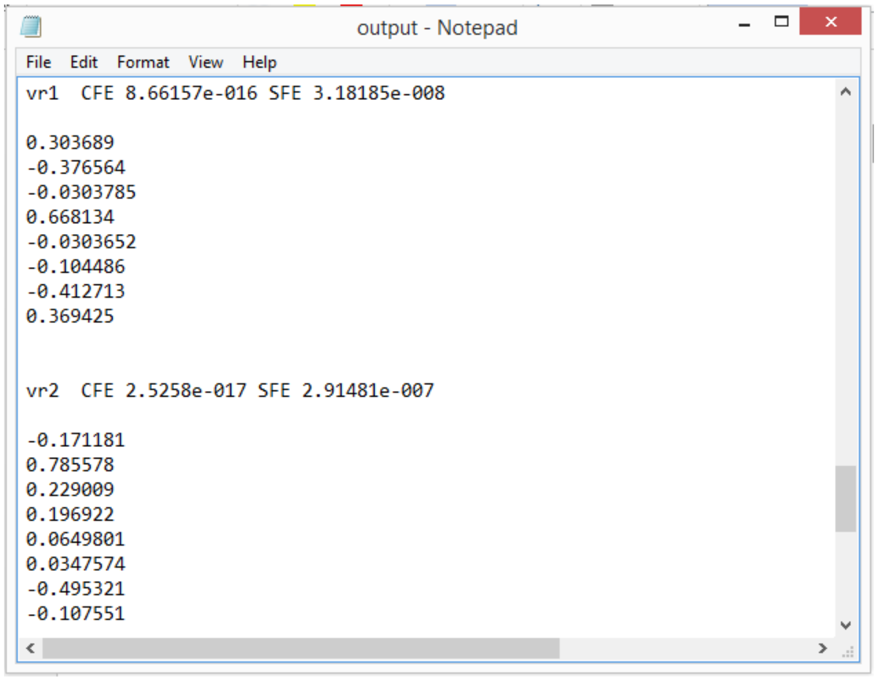


```
Real Roots of a

-1854.61
-8.3425
0.317924
0.558595


Real Roots of b
-182.333
246.376
0.271395
-3.03087
```

Figure 21: Example 1, 10 positions, real roots

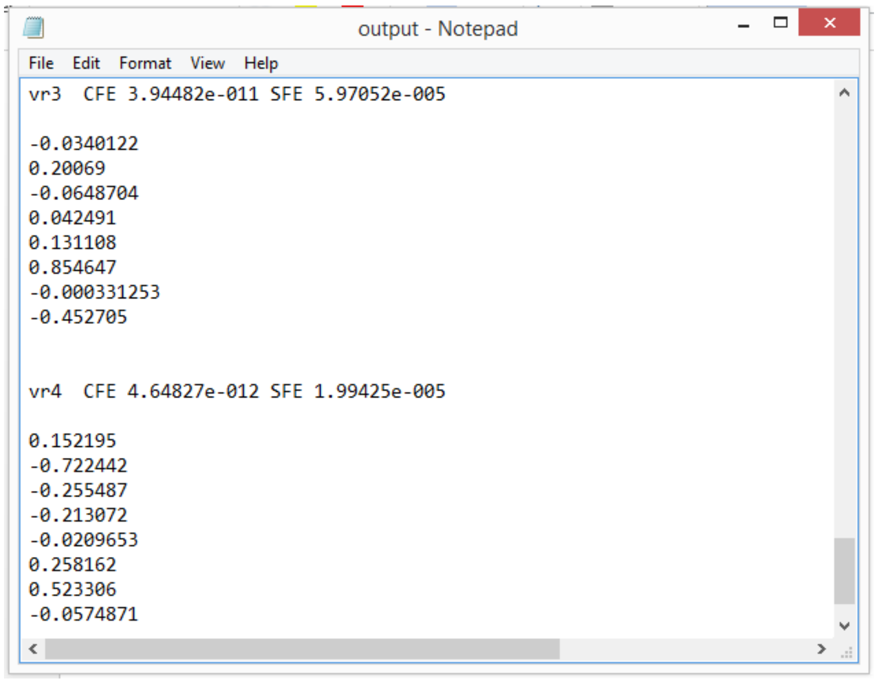
As four real pairs are obtained, we obtain 4 C-Manifolds each corresponding to a dyad.

The C-Manifolds are shown in the image below. It can be clearly seen that the constraint fitting error is minimized and from the pencil of G-Manifolds, they indeed represent C-Manifolds.



```
vr1  CFE 8.66157e-016 SFE 3.18185e-008

0.303689
-0.376564
-0.0303785
0.668134
-0.0303652
-0.104486
-0.412713
0.369425


vr2  CFE 2.5258e-017 SFE 2.91481e-007

-0.171181
0.785578
0.229009
0.196922
0.0649801
0.0347574
-0.495321
-0.107551
```

Figure 22: Example 1, vr1 and vr2



```
vr3  CFE 3.94482e-011 SFE 5.97052e-005

-0.0340122
0.20069
-0.0648704
0.042491
0.131108
0.854647
-0.000331253
-0.452705


vr4  CFE 4.64827e-012 SFE 1.99425e-005

0.152195
-0.722442
-0.255487
-0.213072
-0.0209653
0.258162
0.523306
-0.0574871
```

Figure 23: Example 1, 10 positions, vr3 and vr4

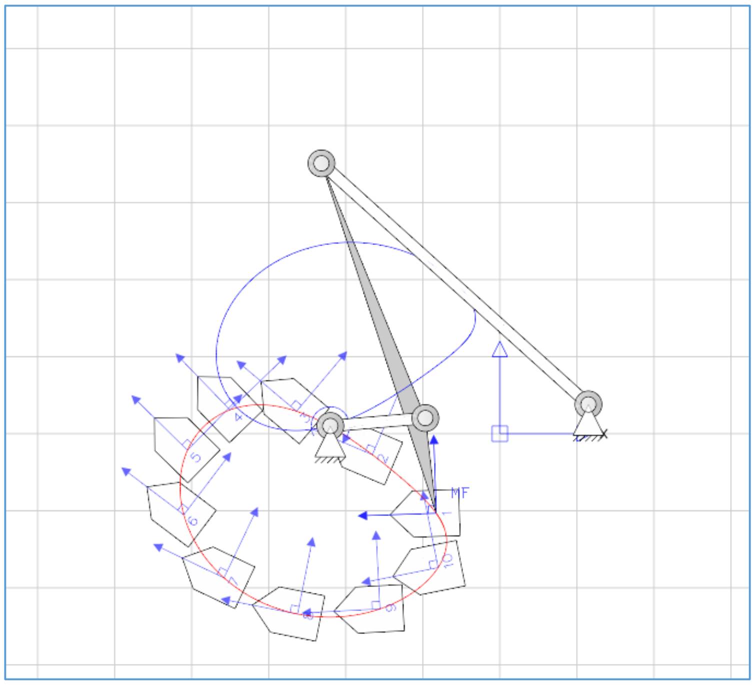


Figure 24: Example 1, 10 positions, Mechanism 1

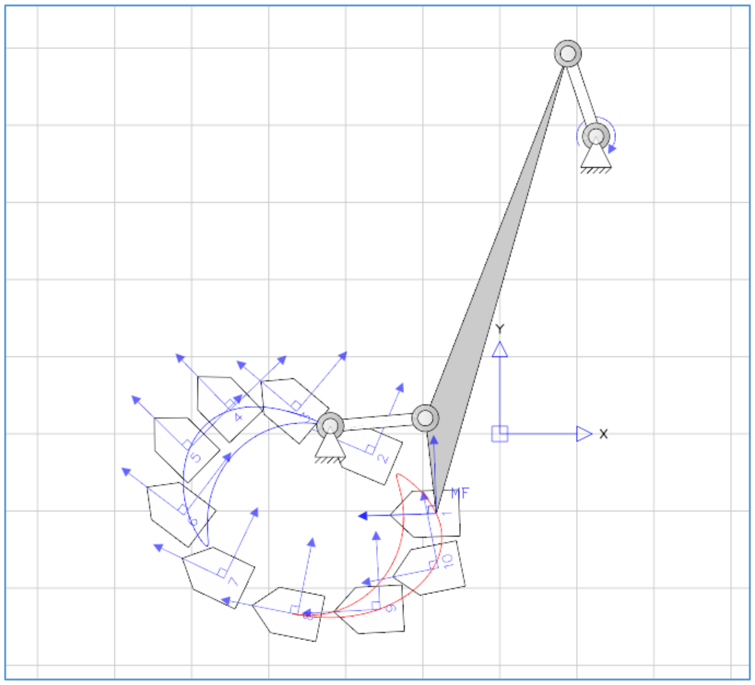


Figure 25: Example 1, 10 positions, Mechanism 2

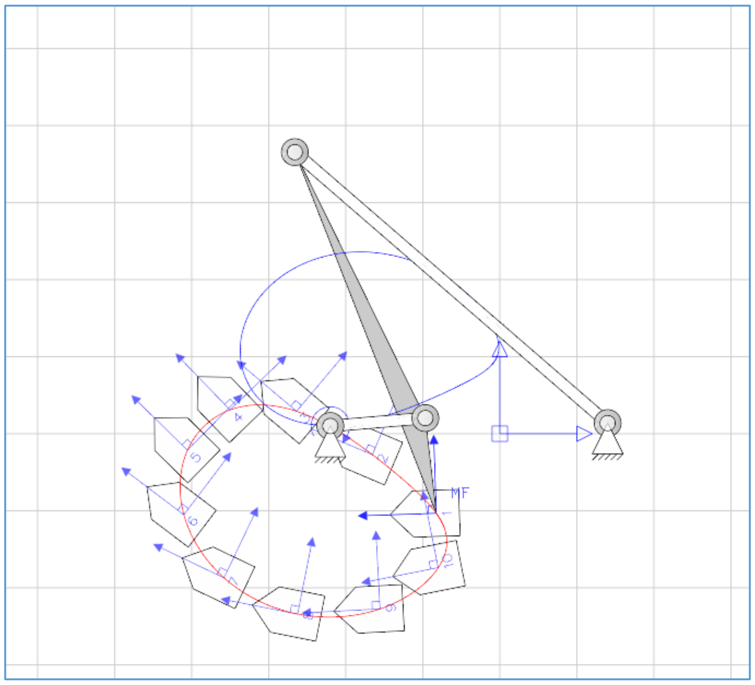


Figure 26: Example 1, 10 positions, Mechanism 3

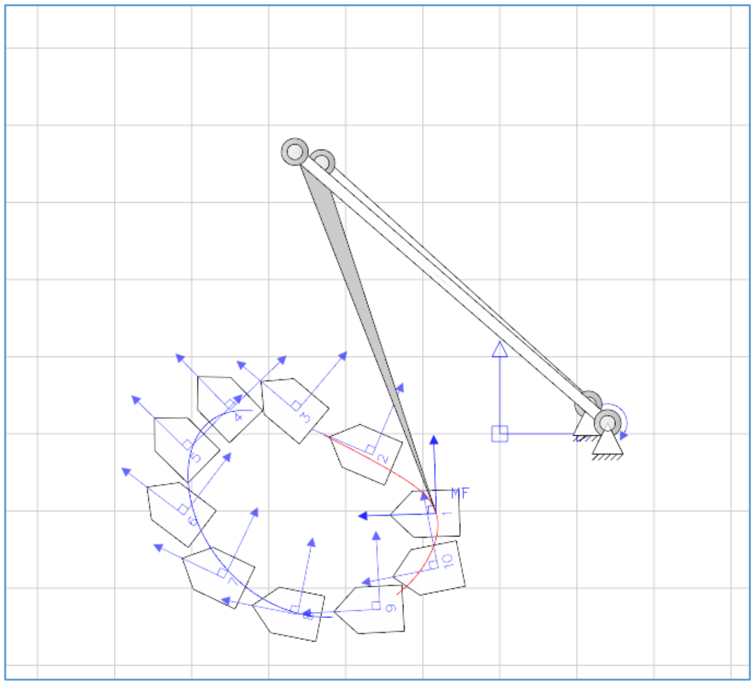


Figure 27: Example 1, 10 positions, Mechanism 4

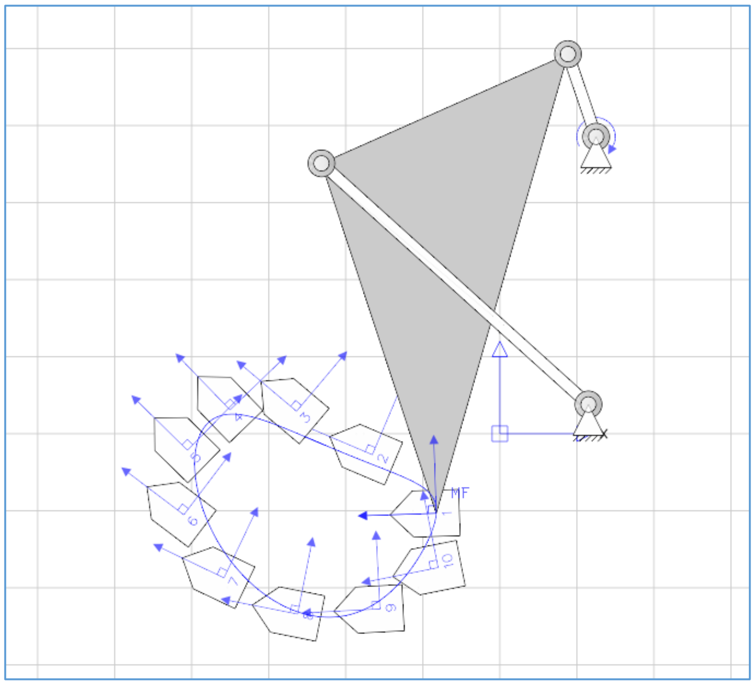


Figure 28: Example 1, 10 positions, Mechanism 5

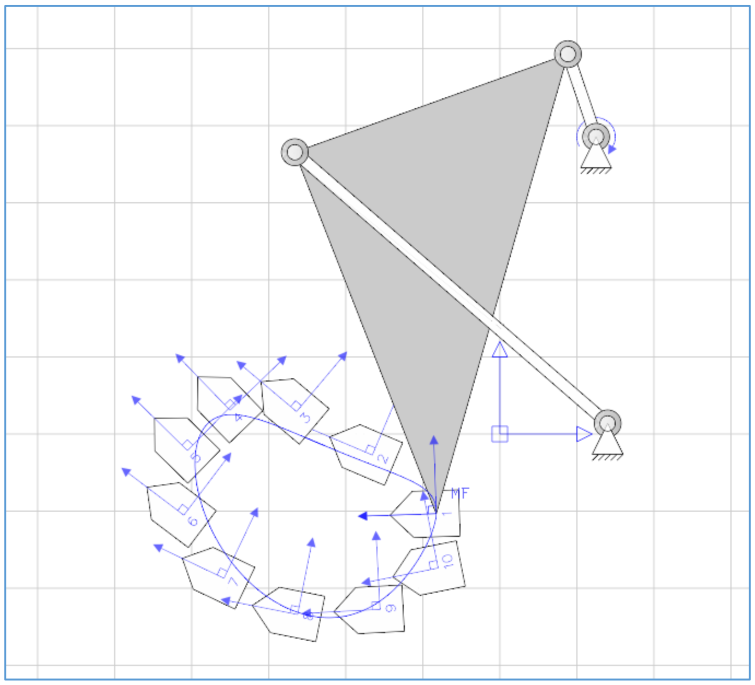


Figure 29: Example 1, 10 positions, Mechanism 1

Following figure shows image space points corresponding to input dyads and constraint manifolds corresponding to two dyads of a mechanism. As constraint fitting error is zero for both the dyads, image space points lie exactly lie on the intersection curve of two manifolds. Also, it is clear that constraint manifold representing RR dyad is hyperboloid of single sheet.

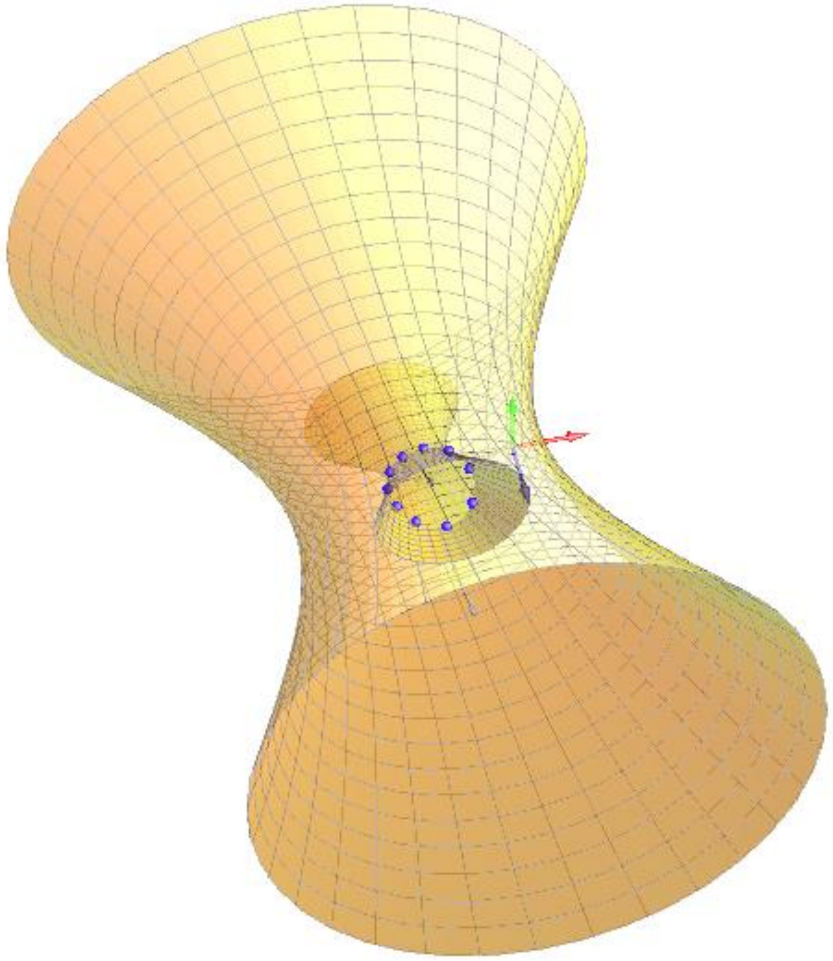


Figure 30: Example 1, 10 positions, constraint manifolds

2. Seven input positions (inputted interactively)

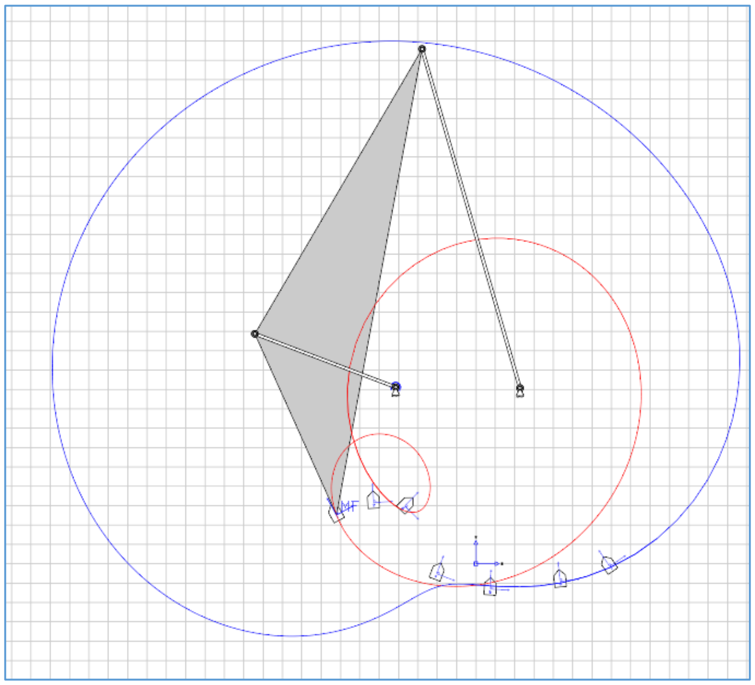


Figure 31: Example 2, seven input positions

3. 10 Input Positions: Mechanism Obtained RRRP

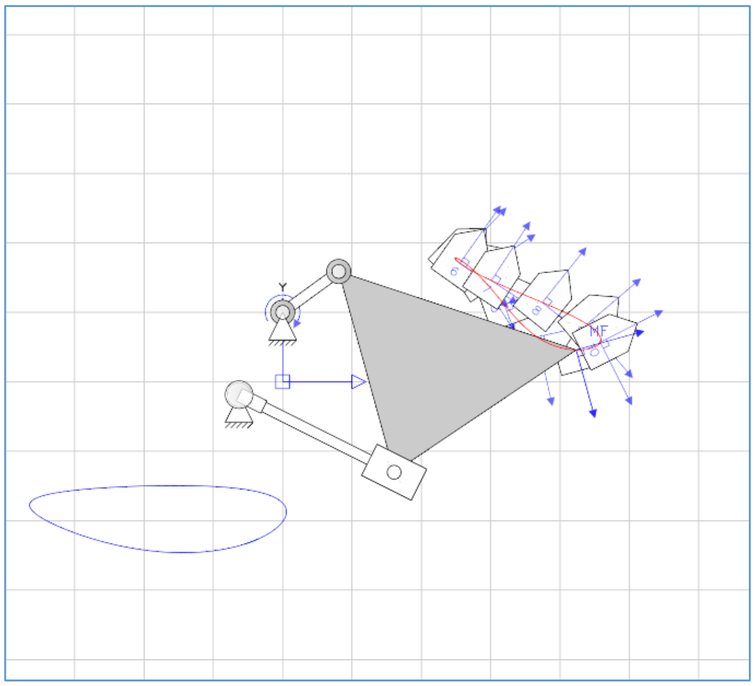


Figure 32: Example 2, 10 input positions, RRRP

4. 5 Input Positions: Shows all 4 dyads obtained

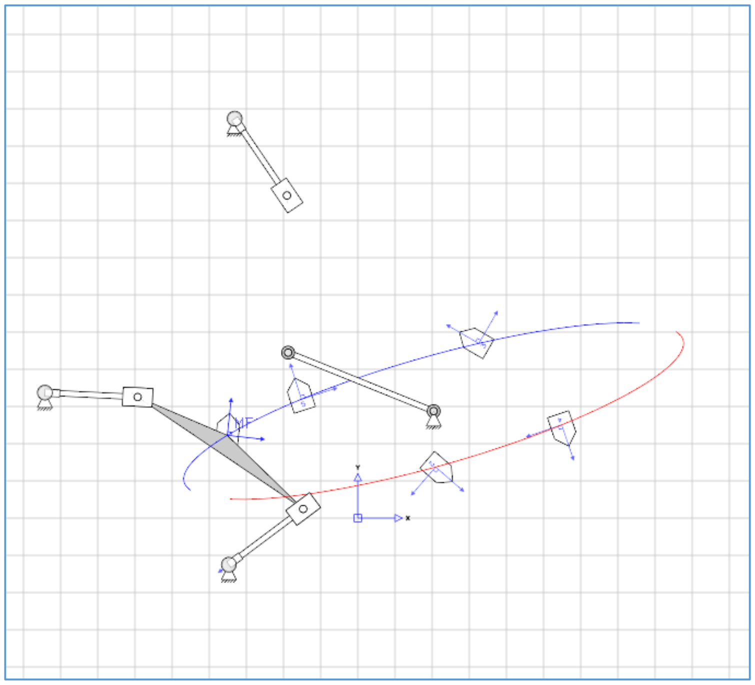


Figure 33: Example 4, five input positions

6.3 Simulation:

RRRR Mechanism

1) Grashof - Crank- crank:

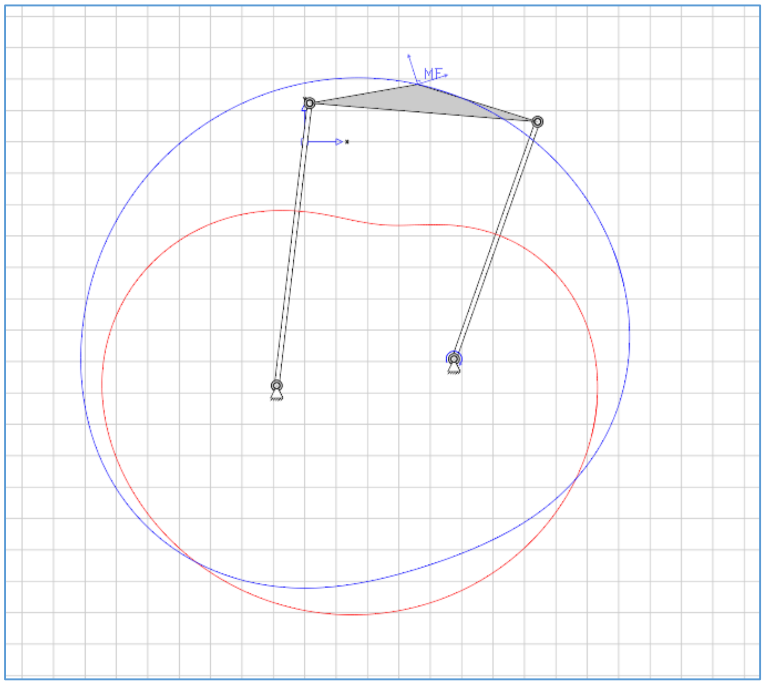


Figure 34: Grashof crank-crank

2) Grashof Crank rocker:

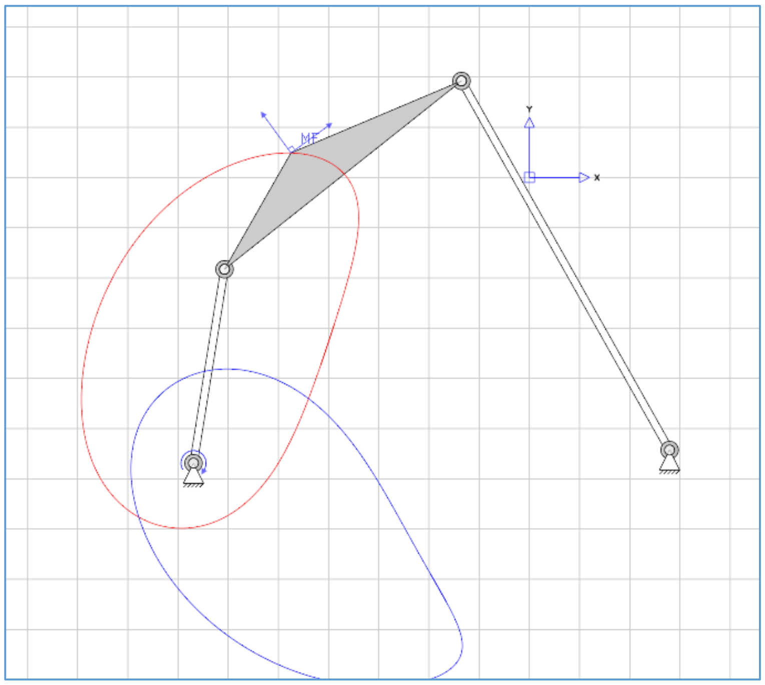


Figure 35: Grashof crank-rocker

3) Grashof Rocker-crank:

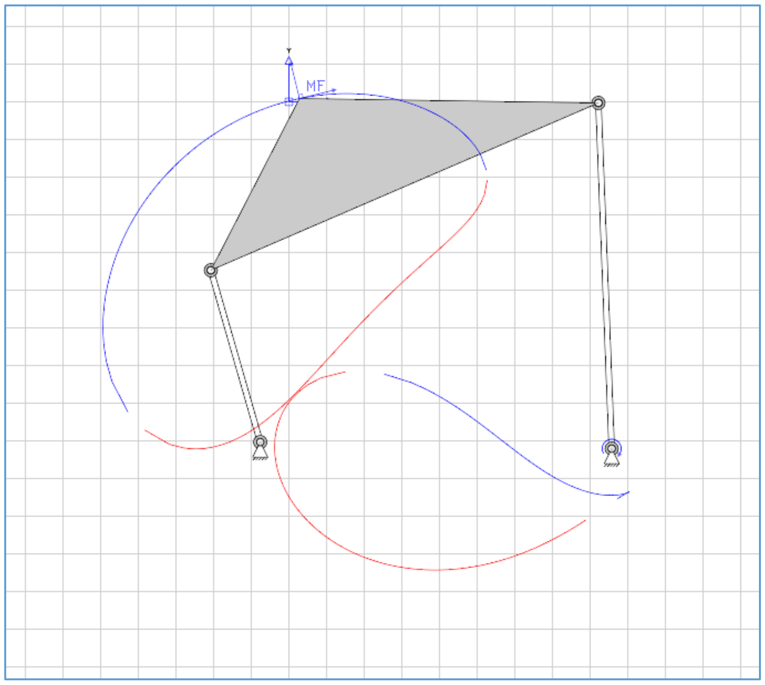


Figure 36: Grashof rocker-crank

4) Grashof double rocker

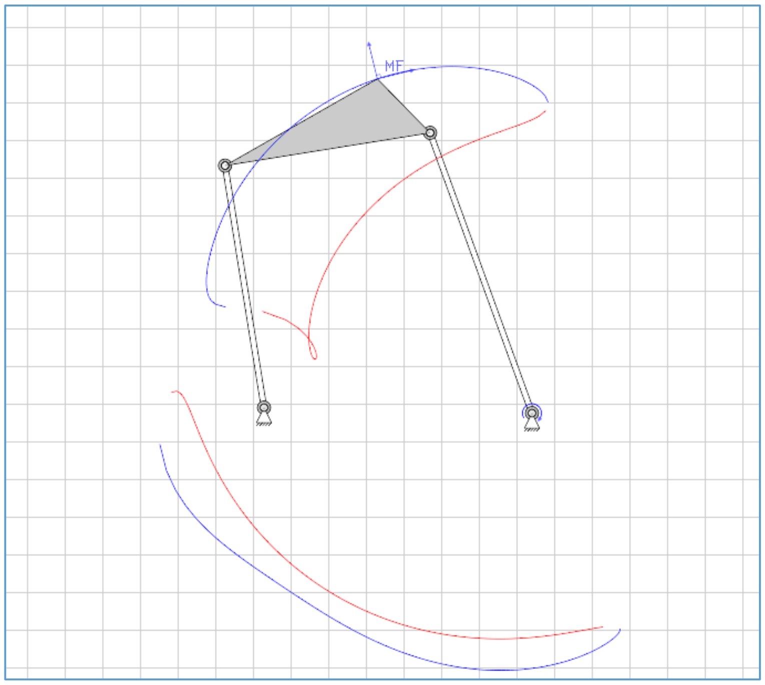


Figure 37: Grashof Double Rocker

5) Non Grashof- Inward-Inward Triple Rocker:

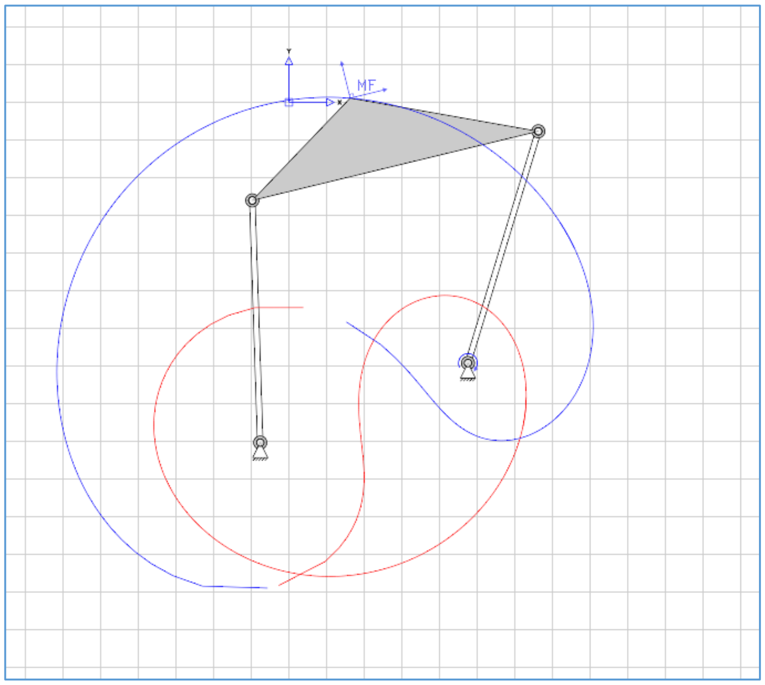


Figure 38: Non Grashof Inward-Inward Triple Rocker

6) Non Grashof- Inward-Outward Triple Rocker:

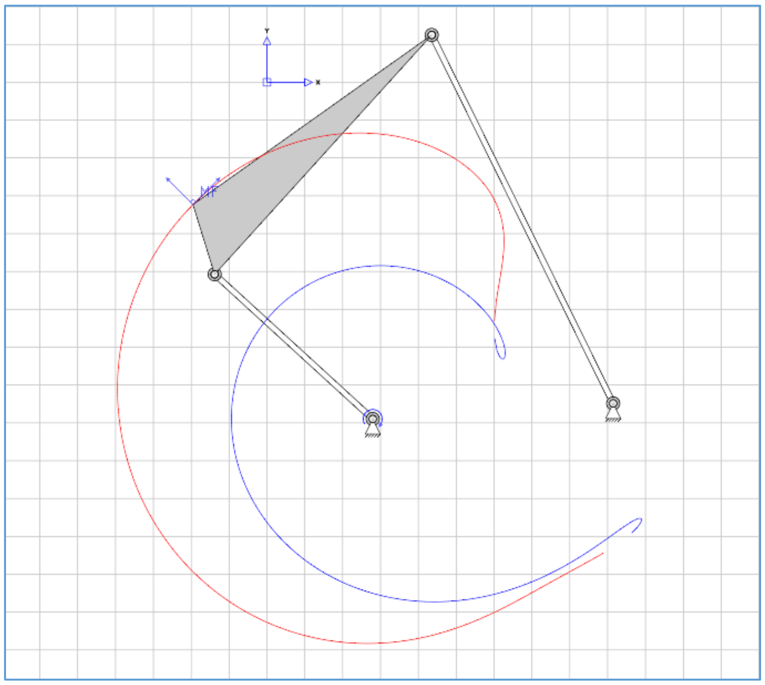


Figure 39: Non Grashof Inward-outward Triple Rocker

7) Non Grashof- Outward-Inward Triple Rocker:

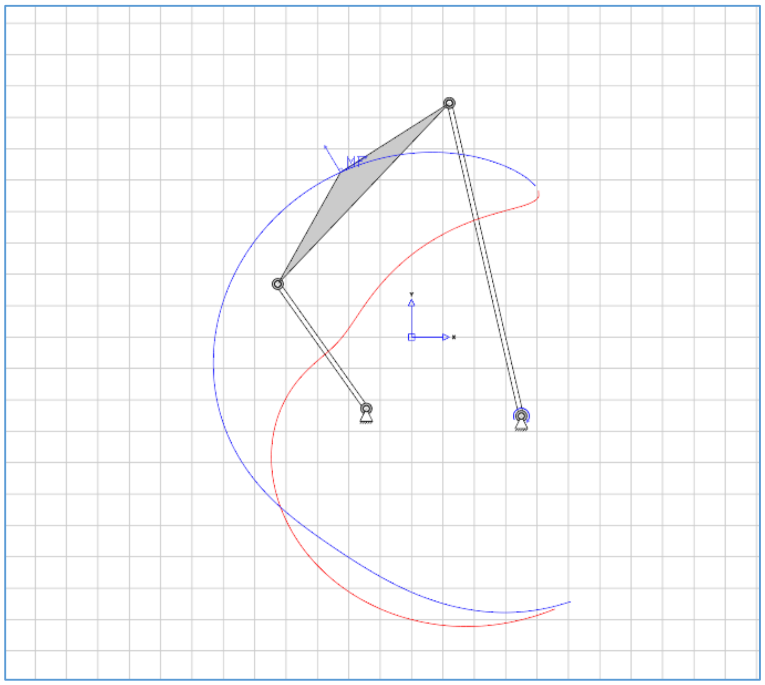


Figure 40: Non Grashof Outward-Inward Triple Rocker

8) Non Grashof Outward-Outward Triple Rocker:

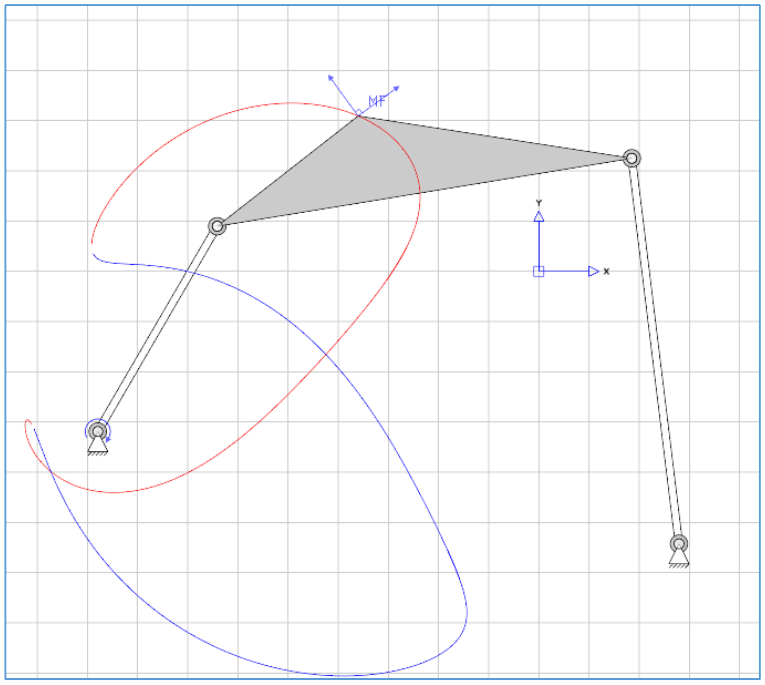


Figure 41: Non Grashof Outward-Outward Triple Rocker

RRPR: Swing Block

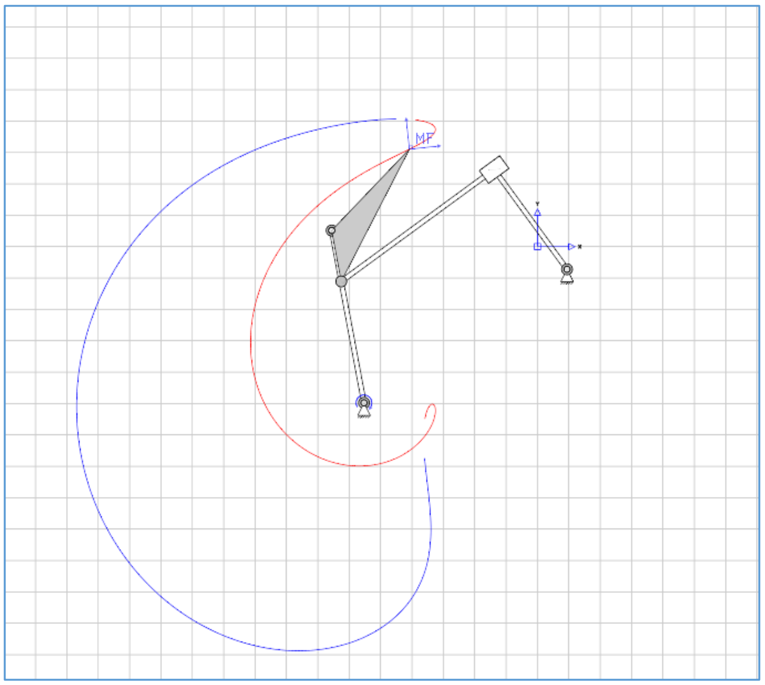


Figure 42: RRPR- swing block mechanism

RRRP: Crank Slider

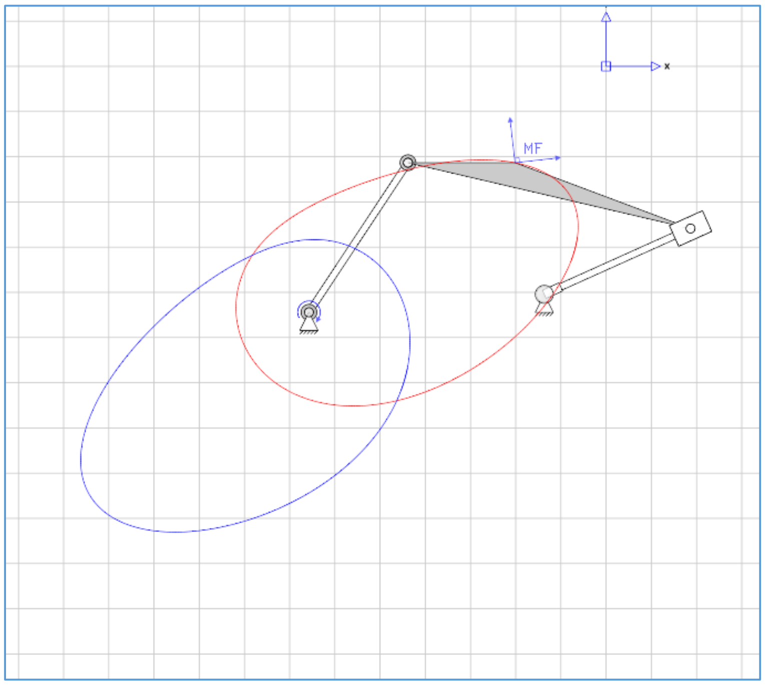


Figure 43: RRRP, crank slider

PRRP: Double Slider Mechanism

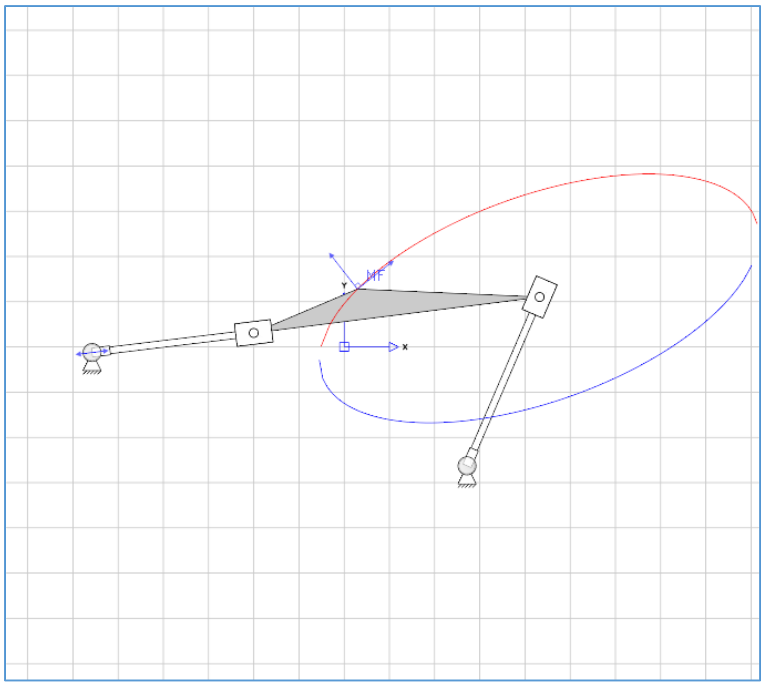


Figure 44: PRRP, double slider

PRPR: Slider Swing Block Mechanism

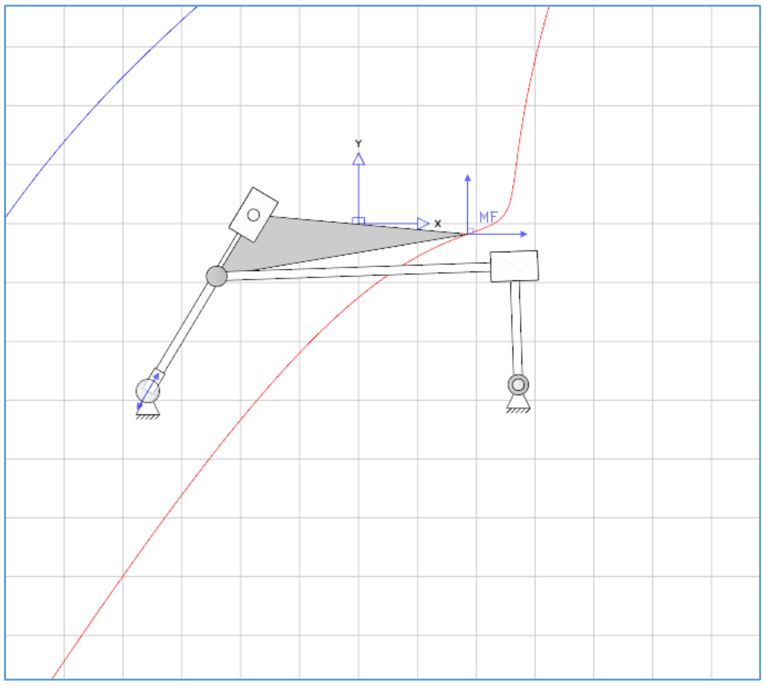


Figure 45: PRPR, slider swing block

RPPR: Double swing block

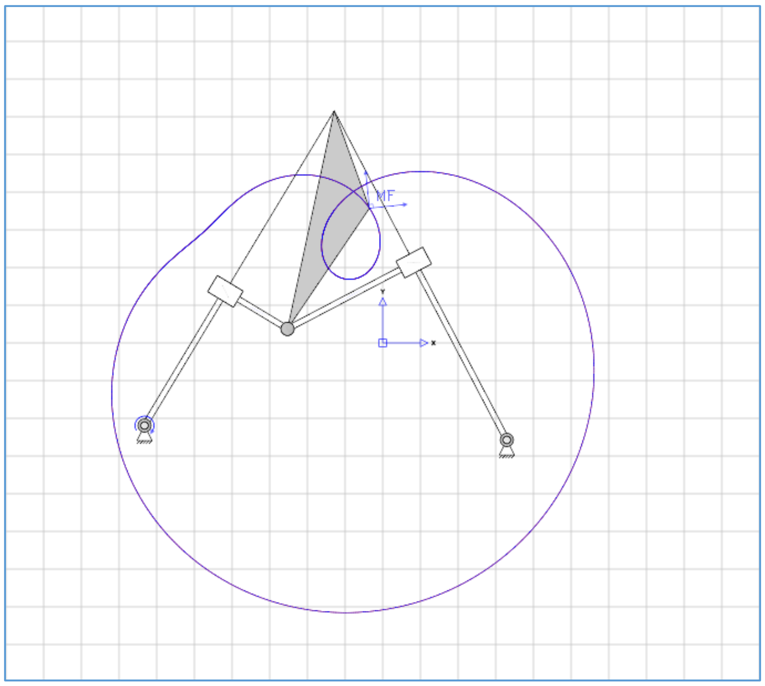


Figure 46: RPPR, double swing block

6.4 Verification of Algorithm used for Simulation:

In this section, 5 positions are sampled out from the input mechanism and stored in a text file. Same text file is then used as an input for synthesis. It is observed that indeed, the original mechanism is regenerated (and may generate extra dyads passing through sample position) through synthesis.

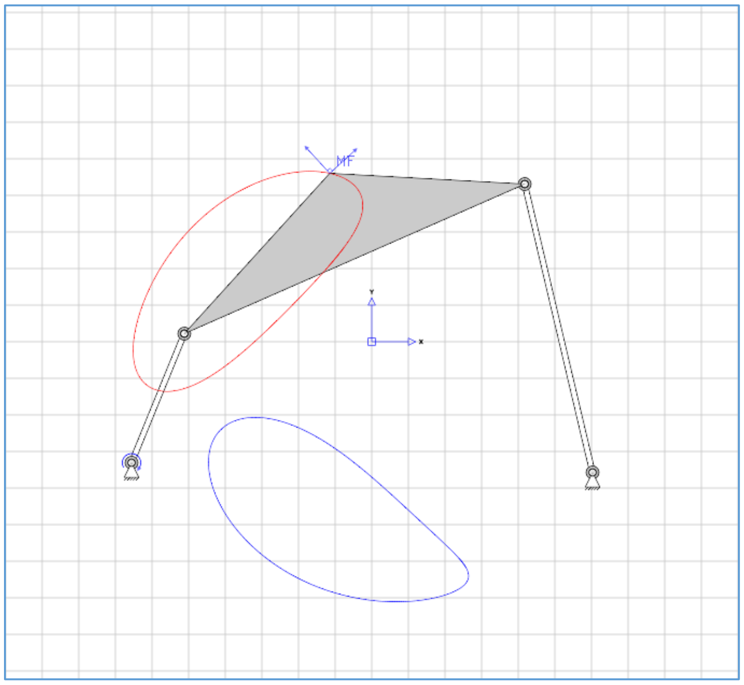


Figure 47: Input Mechanism, verification

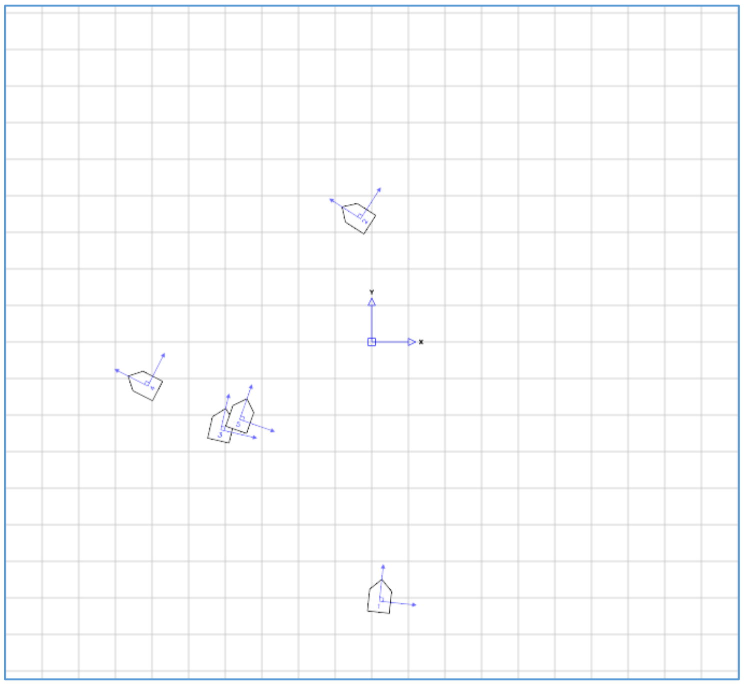


Figure 48: Random Position sampled out, verification

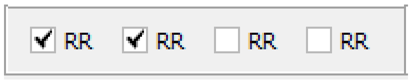

[✓ RR] [✓ RR] RR RR

Figure 49: Dyads obtained after synthesis, verification

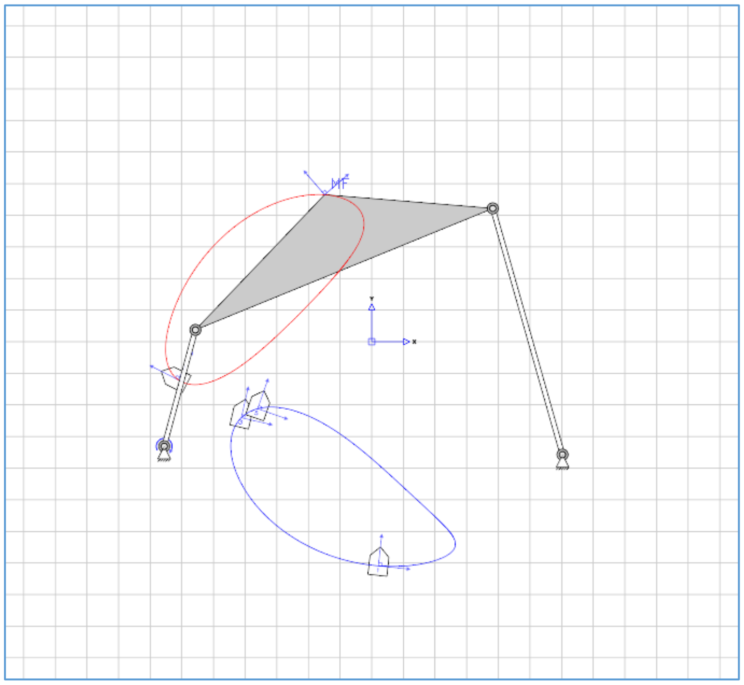


Figure 50: Original Mechanism Regenerated, verification

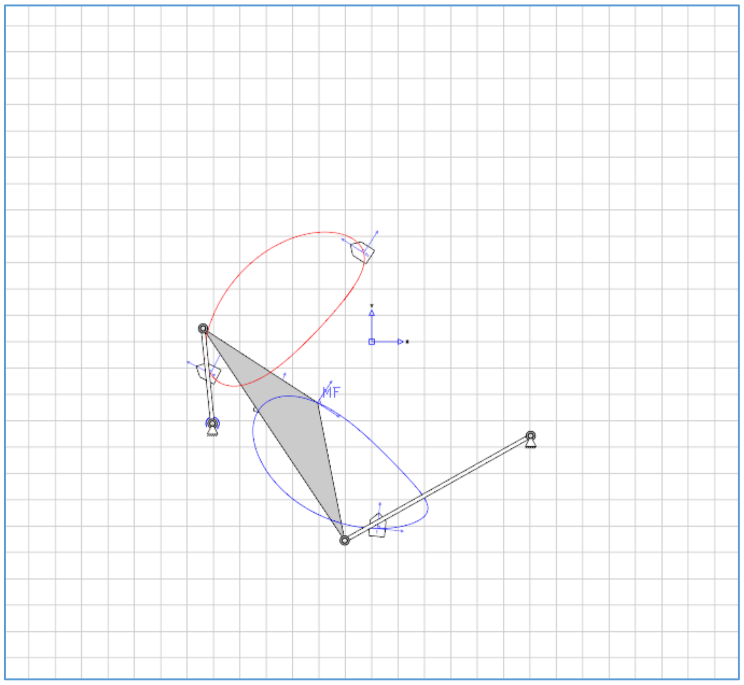


Figure 51: Other mechanism obtained, which goes thruogh sample positions

7. Conclusion and Future Work

7.1 Conclusion

In the software, a novel method for synthesizing planar motion using kinematic mapping is used. Instead of finding two special quadric constraint manifolds associated with a four-bar linkage with nonlinear (quadratic) coefficients, which makes the problem difficult to solve, the algorithm uses more general form of quadric such that its coefficients are linear. The algorithm used for planar four-bar linkage synthesis is not only vastly more efficient but also unifies the treatment of dyads composed from revolute joints and sliding joints (unified type and dimension synthesis).

As the software architecture is object oriented, it has advantage of being easy to understand, versatile and extendible to other types of mechanism such as spatial and spherical type. As Mechanism is a ‘class’; it is possible to incorporate more than one mechanism at a time by simply creating multiple objects of that class.

As ‘Qt’ is cross platform language. It is possible to create executables for other platforms such as OS X or Linux by compiling the source code for those platforms.

The current tool will act as a foundation to an extensive commercial application which will be an important asset to the broader cyber-infrastructure for engineering design. It will not be only a research tool but also for academic purposes and non-expert users.

7.2 Future Work

This part enlists possible improvements and the features that could be added in the software.

- Fixing issues concerning synthesis of RP dyad.
- Plotting Hyperbolic-Paraboloid (corresponding to RP and PR dyads) in Image Space.
- The algorithm for synthesis currently works well for more than 5 positions. Incorporate algorithm for synthesis using less than 5 positions.
- Incorporating more than one mechanism (as Mechanism is a class- It is possible with creating multiple objects of same class).
- Velocity and acceleration plot for the coupler point.
- A ‘wizard’ for entering mechanism and motion data and to assist the user.
- Increment and Decrement button issues.

- Improving the display and drawings.
- Change orientation of Moving Frame.
- Use of other Icons.
- Visual Feedback for icons.
- Saving the mechanism data for construction etc.
- Turn on Labels for mechanisms (show/hide).
- Saving animation (in the form of .avi or other video formats).
- Printing mechanism and motion to a printer or pdf.
- Help files and tutorials.
- Allow a filter to select mechanisms on the basis of Grashof/Non-Grashof, circuit defect, branch defect, order defects.
- Constraint Input: extra position, fixed pivot line, moving pivot line, link lengths.
- To Show other image space surfaces (singular vectors and the constraint manifold), ability to show them individually.
- Set Workspace size.
- Setting tolerances for input- positions or other constraints.
- Path Generation and Function Generation

References

[1] Q.J.Ge, Ping Zhao, Anurag Purwar, A Task Driven Approach to Unified Synthesis of Planar Four-bar Linkages using Algebraic Fitting of a Pencil of G-manifolds, In ASME Design Engineering Technical Conferences, 2013. DETC2013-12977.

[2] A. J. Rubel and R. E. Kaufman. Kinsyn iii: A new human-engineered system for interactive Computer-aided design of planar linkages. ASME Journal of Engineering for Industry, 99:440-448, 1977.

[3] G. Erdman and J. Gustafson. Lincages: Linkage interactive computer analysis and graphically enhanced synthesis packages. Technical report, 1977.

[4] G. Erdman and D. Riley. Computer-aided linkage design using the lincages package. In
ASME Design Engineering Technical Conferences, 1981. 81-DET-121.

[5] J. Kihonge, J. Vance, and P. Larochelle. Spatial mechanism design in virtual reality with
Networking. In ASME 2001 Design Engineering Technical Conferences, 2001.

[6] P.M. Larochelle. Spades: Software for synthesizing spatial 4c linkages. In CD-ROM Proc. Of the ASME DETC'98, 1998. Paper no.DETC98/Mech- 5889.

[7] P. Larochelle, J. Dooley, A. Murray, and J. M. McCarthy. Sphinx: Software for synthesizing spherical 4r mechanisms. In Proc. of the 1993 NSF Design and Manufacturing Systems Conference, volume 1, pages 607{611, 1993.

[8] D. Ruth and J.M. McCarthy. Sphinxpc: An implementation of four position synthesis for Planar and spherical 4r linkages. ASME Design Engineering Technical Conferences, 1997.

[9] D. Tse and P.M. Larochelle. Osiris: a new generation spherical and spatial mechanism cad program. In Florida Conference on Recent Advancements in Robotics, 1999.

[10] H.-J. Su, C.L. Collins, and J.M. McCarthy. An extensible java applet for spatial linkage synthesis. In ASME International Design Engineering Technical Conferences, 2002.

[11] Thomas R. Chase, Gary L. Kinzel, and Arthur G. Erdman.; advances in Mechanisms, Robotics and Design Education and Research; Mechanisms and Machine Science Volume 14, 2013, pp 17-33

[12] Freudenstein, F., Sandor, G.N.: On the Burmester Points of a Plane. ASME Journal of Applied Mechanics Series E 28(1), 41–49 (1961)

[13] Petuya, V.; Macho, E.; Altuzarra, O.; Pinto, C. and Hernández, A. “Educational Software Tools for the Kinematic Analysis of Mechanisms”. Comp. Appl. Eng. Education. First published online: February 24, 2011. DOI: 10.1002 cae.20532. ISSN: 1061-3773.

[14] Burmester, L. E. H., 1888. Lehrbuch der kinematik. A. Felix, Leipzig, Germany. 23004756 diagrs. 24 cm.

[15] McCarthy, J. M., 1990. Introduction to Theoretical Kinematics. MIT, Cambridge, Mass.

[16] Sandor, G. N., and Erdman, A. G., 1997. Advanced Mechanism Design: Analysis and Synthesis Vol. 2. Prentice-Hall, Englewood Cliffs, NJ.

[17] Hunt, K., 1978. Kinematic Geometry of Mechanisms. Oxford University Press, New York.

[18] Hartenberg, R. S., and Denavit, J., 1964. Kinematic Synthesis of Linkages. McGraw-Hill, New York.

[19] Suh, C. H., and Radcliffe, C.W., 1978. Kinematics and Mechanism Design. Wiley, New York.

[20] David H. Myszka (2009*), Kinematic synthesis and analysis techniques to improve planar rigid body guidance, Ph.D. thesis, University of Dayton, USA*

[21] Enginerdery available at http://www.enginerdery.org/w/Grashof's_criteria

[22] Ravani, B., and Roth, B., 1983. “Motion synthesis using kinematic mappings”. Journal of Mechanisms Transmissions and Automation in Design-Transactions of the ASME, 105(3), pp. 460–467. Article.

[23] SyMech Inc. Symech, http://www.symech.com/.

[24] Heron Technologies. Watt, http://www.heron-technologies.com/watt/.

[25] Artas Engineering. Sam (synthesis and analysis of mechanisms), http://www.artas.nl/en.

[26] McCarthy Design Associates. Mechgen, http://mechanicaldesign101.com/ Mechanism-generator-2-0/#MechGen3

[27] Dassault Systems. Solidworks, http://www.solidworks.com/.

[28] Design Simulation Technologies. Working model 2d, http://www.design-simulation.com/WM2D/.

[29] SoftIntegration. Ch mechanism toolkit, http://www.softintegration.com/products/toolkit/mechanism/

[30] Tinkercad Inc. Tinkercad, https://tinkercad.com.

[31] 3DTin. 3dtin, http://www.3dtin.com/.

[32] Plugworks Inc. Padcad, https://plugworks.com/.

[33] Benjamin Nortier. Shapesmith, http://shapesmith.net/.

[34] Bottema, O., and Roth, B., 1979. Theoretical Kinematics. North Holland, Amsterdam.

[35] Golub, G., and Van Loan, C., 1996. Matrix Computations. Johns Hopkins Univ Press, Baltimore, MD.

[36] Armadillo, available at http://arma.sourceforge.net/