Proceedings of the ASME 2014 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2014
August 17-20, 2014, Buffalo, NY, USA

DETC2014-35235

4MDS: A GEOMETRIC CONSTRAINT BASED MOTION DESIGN SOFTWARE FOR SYNTHESIS AND SIMULATION OF PLANAR FOUR-BAR LINKAGES

Anurag Purwar, Abhijit Toravi, Q.J. Ge

anurag.purwar@stonybrook.edu
Computational Design Kinematics Lab
Department of Mechanical Engineering
Stony Brook University
Stony Brook, New York, 11794-2300

ABSTRACT

*This paper presents our recent work on designing and developing a geometric constraint based motion design software system for planar four-bar linkages. Given a motion task, the software computes possible four-bar linkage topologies as well as its dimensions. This capability to analyze the given task and find the best type of the linkage and the dimensions simultaneously sets it apart from any other linkage design software. The **F**our-Bar **M**otion **D**esign **S**ystem (4MDS) makes the synthesis and simulation capabilities available to mechanism designers in an intuitive graphical user interface (GUI) environment. Instead of taking a black box approach to mechanism design, wherein the user simply enters the motion requirements and the software outputs parameters of mechanisms, this software facilitates a dialog with the designer by providing various paths to simulation and synthesis in a design session. The designer has complete control over the specification of motion task, interactive tweaking of the motion, choice of linkage topology computed, dimensional changes, and their apparent effect on motion, all done in real time. This interactivity enhances designers kinematic experience. The underlying theoretical foundation of this paper is based on our earlier work on a task-driven approach to unified type and dimensional synthesis of planar four-bar linkage mechanisms. Instead of treating a planar four-bar mechanism as a set of connected rigid links and joints, we treat them as line or circle constraint generators. With that view, the synthesis problem is reduced to extracting geometric constraints hidden in a given motion task and the simulation is reduced to assembling constraints realizable by mechanical dyads. The algorithm employed is simple and efficient and permits real-time computation, and thus precludes using a limiting database-oriented approach. This tool should make innovation of mechanical motion generating devices accessible to novice and experienced designers alike.*

1 Introduction

1.1 Kinematic Synthesis of Mechanisms

This work deals with the kinematic synthesis of planar four-bar linkages for a given motion. A kinematic mechanism is a collection of links connected with kinematic pairs (or joints). Although not always visible, kinematic mechanisms are widely embedded in engineered products and systems such as automobiles, robots, biomechanical devices, as well as production systems for these products. Planar four-bar mechanisms are the most widely used closed loop, single degree of freedom linkages in mechanical systems and are almost always the first choice of designers because of their simplicity. Some key texts that describe established methods and theory in kinematic synthesis of machines are by McCarthy and Soh [1], Sandor and Erdman [2], Hunt [3], Hartenberg and Denavit [4], and Suh and Radcliffe [5].

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 [6]. The focus of this work is on kinematic synthesis of linkages for motion generation.

Kinematic synthesis, in its broadest sense, consists of type, structural, and dimensional synthesis. *Type synthesis* is concerned with the decision regarding the *type* of mechanism to be employed, whether to choose lower- or higher-pairs, such as the choice between a cam mechanism and a linkage; *structural synthesis*, also known as *topological synthesis* and sometimes *number synthesis*, is concerned with the number of links and connections as well as their patterns of connections; *dimensional synthesis* is concerned with the determination of those dimensions that affect the kinematic behavior of a mechanism. In the context of this work, we refer to type synthesis problem as determining the best planar four-bar linkage topologies for a given motion.

1.2 Theoretical Underpinning

This software system's core algorithm is based on our work on a task driven approach to unified type and dimensional synthesis of planar four-bar linkage mechanisms (Ge and Purwar [7]). In that work, using a kinematic mapping (Blaschke [8] and Grunwald [9]) of planar kinematics, we proposed a general algebraic method for unified type and dimensional synthesis of planar four-bar linkages, which reveals the geometric constraints implicit in the given motion via a two-step data and constraint fitting process. The algorithm proposed is fast and efficient and provides type and dimensions of the mechanisms simulatenously, which enables real-time computation in the software. A modern treatment of kinematic mapping can be found in the formative texts of Bottema and Roth [10] and McCarthy [11] and in the work of Ravani and Roth [12, 13]. 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 (precision position synthesis). For a continuous motion or for more than five positions, typically only an approximate motion synthesis can be performed. This software system can handle either of the cases.

In our kinematic mapping approach to synthesis, planar displacements in Cartesian Space are mapped into points in a three-dimensional projective space (called Image Space of Planar Kinematics), while workspace constraints of a mechanism map into algebraic manifolds (called Constraint Manifold) in the same space. A four-bar linkage has two dyads connected at the ends. Each dyad's geometric constraint maps into an algebraic surface in the image space. In this way, a single degree of freedom motion of a planar four-bar linkage is represented by the intersection curve of two algebraic surfaces in the image space. A unified representation for all possible dyadal constraints is obtained which allows best *type* determination. The problem of motion synthesis is transformed into a algebraic curve fitting problem in the image space. Using a Singular Value Decomposition (SVD) process, the image points are fitted to a pencil of quadrics and then application of two simple quadratic conditions on the pencil reveals the constraint manifolds of the mechanical dyads that can execute the given motion the best. The signature of the obtained constraint manifolds represents the type of the mechanical dyad in the linkage, while their geometry gives away the dimensions of dyads.

1.3 Computer Aided Mechanism Design

Modern commercial CAD/CAM/CAE softwares are fairly mature and advanced in providing their users capabilities to model shapes, prepare assemblies and detailed drawings, interface with computer controlled machining tools, and perform engineering simulations and analysis. They have also begun to provide mechanism design capabilities by leveraging the geometric constraint capabilities, such as parallelism, perpendicularity, connections, etc. However, they lack to provide machine and mechanism designers a path to innovation wherein both the mechanism topology and the dimensions may be determined simultaneously. An early stage, critical goal in the machine design process is generation, evaluation, and analysis of mechanism design concepts that can realize a given task. To this end, there have been many academic research efforts in the development of software systems for the synthesis of planar, spherical, and spatial mechanisms (KINSYN III from Rubel and Kaufamn [15], LINCAGES from Erdman et al. [16, 17], Kihonge et al. [18], Spades from Larochelle [19], Sphinx from Larochelle et al. [20], Sphinxpc from Ruth and McCarthy [21], Osiris from Tse and Larochelle [22], and Synthetica from Su et al. [23]). Among these, Synthetica, a spatial linkage design and animation software, is probably the only one under active development.

More recently, Purwar et al. [24], Wu et al. [25] and Purwar and Gupta [26] have demonstrated a visual, computer graphics approach for multi-degrees of freedom mechanism design that exploits the constraint manifold geometry and its apparent effect on the parameters of a mechanism to interactively perform kinematic synthesis. These tools, although useful to designers, have a strong research focus.

SyMech [27] and WATT [28] from Heron Technologies used to be two well-known software systems for planar mechanisms design. Merely a decade after a special session on *Computer Aided Linkage Synthesis* was held at the 2002 ASME International Design Engineering Technical Conferences [29], where both Heron Technologies and SyMech presented, neither of the softwares are available any longer. Fortunately, there have been some recent developments on providing mechanism design softwares to a broad class of users. SAM (Synthesis and Analysis of Mechanisms) from Artas Engineering [30] can perform at most three position synthesis for planar linkages, but expects users to rely on their design experience, or consult handbooks for anything beyond. Despite these critical limitations, the software is polished and has extensive modeling and simulation capabilities. MechGen from McCarthy Design Associates [31] provides a plug-in to Solidworks [32] for five position planar four-bar and six-bar linkage synthesis. An attractive feature of this plug-in is that it harnesses the CAD modeling capabilities of the host software, and therefore, extends beyond stick kinematic diagrams to provide a rich context-oriented interaction to the user. Unfortunately, the total cost of ownership of both Solidworks and plug-ins can be prohibitive. `Linkages` from Norton Associates Engineering [33] is an educational software bundled with the well-known Design of Machinery textbook (Norton [6]), which synthesizes planar linkages for two- and three-positions of the coupler. `Linkages` has replaced Working Model 2D from Design Simulation Technologies [34] and other synthesis tools found in earlier versions of the text. However, both of them are primarily mechanism modeling and simulation tools. Ch Mechanism Toolkit (SoftIntegration [35]) written in Ch, a C/C++ interpreter and scripting language environment, is a similar tool for mainly animating and analyzing planar linkages. Some of the well-known softwares, such as Adams [36], NX Unigraphics [37], PTC Creo [38], and Solidworks Motion [39], primarily known for their CAD prowess also have the capabilities to simulate the motion of the mechanism when appropriate geometric- and dimensional-constraints are imposed on the links of a mechanism model. In a similar vein, with a focus on the educational market, Geometer's Sketchpad [40] and Geogebra [41], both interactive geometry and algebra software, GrafiCalc [42], Molecular Workbench [43], Solvespace [44], and Analytix [45] can also solve the mathematical relationships to determine positions of various links of a four-bar linkage to achieve animation. Kinzel et al. [46] proposed a geometric constraint programming approach for mechanism synthesis similar to what is found in Geogebra, albeit in a host CAD environment. MechDesigner (PSMotion [47]) is a specialized tool for packaging, assembly and textile machine design. Similarly, Universal Mechanism [48] is another specialized CAE tool intended for simulation of kinematics and dynamics of transport machines, road vehicles, railway vehicles, space structures, robots and manipulator, and defense system.

Unfortunately, none of the aforementioned available stand alone CAx tools provide true motion design simultaneous type and dimensional determination capabilities. Either they limit users to 3-4 task positions or require synthesis via a simulation based trial and error process by pre-supposing the type of the mechanism. The 4MDS provides an environment where the designer is not constrained by these limitations and can explore a multitude of options before converging to one or many possible mechanism design solutions. 4MDS allows designers to specify a motion via 5 or more task positions and it can compute linkage topologies and dimensions that can execute the given motion with the least structural error. It does so by analyzing the given task and extracting the geometric constraints implicit in the tasks. The simulation of known linkages is performed by by an assembly of geometric constraints of mechanical dyads. This tool not only permits innovation of new mechanical motion generating devices, but also allows designers to reverse engineering known mechanisms. The software (see Fig. 1) is a work in progress and is being released as a beta software at the time of the writing of this paper at `http://cadcam.eng.sunysb.edu/software`.

The organization of the rest of the paper is as follows. Section 2 reviews kinematic mapping concept, geometric constraints of planar dyads, and a unifying representation of constraints. Section 3 reviews design methodology for planar four-bar linkages. Section 4 presents the features of the software system, and section 5 presents the architecture of the software. Finally, we present an example design scenario before concluding and making some remarks on future work.

2 Theoretical Fundamentals

The basis of this work is contained in [7]. In this section, we review some fundamentals from that work in so far as necessary for placing this work in a proper context and for understanding of the features and capabilities of the software.

2.1 Kinematic Mapping of Planar Displacements

A planar displacement can be decomposed into the translation of a point (d_1, d_2) on the moving body as well as rotation of the body by an angle ϕ. Let M denote a coordinate frame attached to the moving body and F be a fixed reference frame. Then, a planar displacement can be represented as a transformation of point or line coordinates from M to F. The kinematic mapping from Cartesian space parameters (d_1, d_2, ϕ) to Image Space coordinates $\mathbf{Z} = (Z_1, Z_2, Z_3, Z_4)$ is defined as (see Ravani and Roth [12]),

$$\begin{aligned} Z_1 &= \frac{1}{2}(d_1 \sin\frac{\phi}{2} - d_2 \cos\frac{\phi}{2}), \\ Z_2 &= \frac{1}{2}(d_1 \cos\frac{\phi}{2} + d_2 \sin\frac{\phi}{2}), \\ Z_3 &= \sin\frac{\phi}{2}, \\ Z_4 &= \cos\frac{\phi}{2}. \end{aligned} \tag{1}$$

A point $\mathbf{x}$ in M given by its homogeneous coordinates (x_1, x_2, x_3) (with $x_3 \neq 0$) is mapped to $\mathbf{X} = (X_1, X_2, X_3)$ (with $X_3 \neq 0$) in F

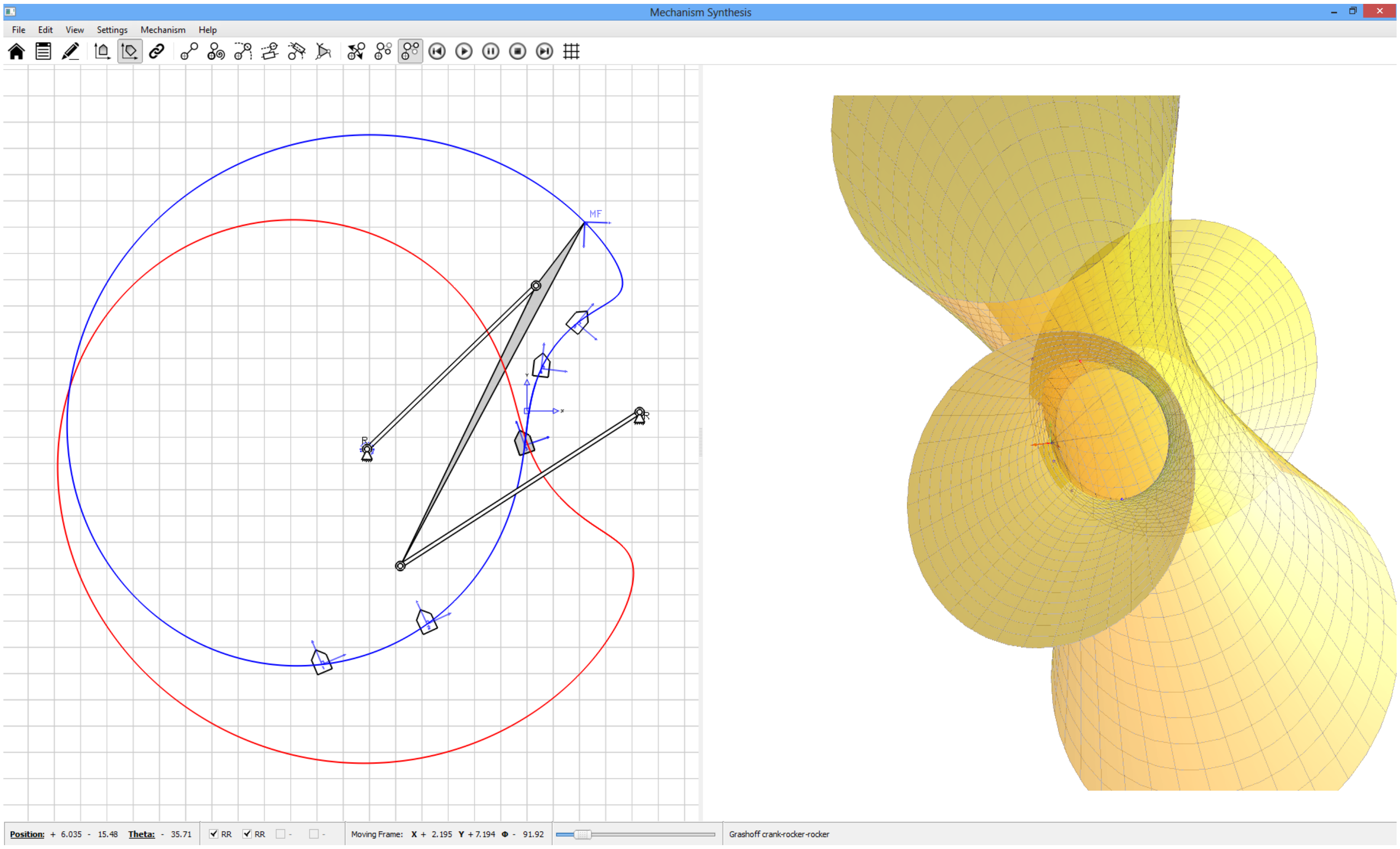


FIGURE 1. A screenshot of the 4MDS showing Design Space Window (left), Image Space Window (right), main menu and toolbar (top), status bar and dyad selection (bottom)

by the following homogeneous transform:

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

where $Z_3^2 + Z_4^2 = 1$. Similarly, for a line with homogeneous coordinates $\mathbf{l} = (l_1, l_2, l_3)$ in M and its corresponding coordinates $\mathbf{L} = (L_1, L_2, L_3)$ in F, we have

$$\mathbf{L} = [\overline{H}]\mathbf{l}, \quad [\overline{H}] = \begin{bmatrix} Z_4^2 - Z_3^2 & -2Z_3Z_4 & 0 \\ 2Z_3Z_4 & Z_4^2 - Z_3^2 & 0 \\ 2(Z_1Z_3 - Z_2Z_4) & 2(Z_2Z_3 + Z_1Z_4) & Z_3^2 + Z_4^2 \end{bmatrix}. \quad (3)$$

The four-dimensional coordinates $\mathbf{Z} = (Z_1, Z_2, Z_3, Z_4)$ are said to define a point in a projective three-space called the *Image Space* of planar displacement, denoted as Σ. In this way, a planar displacement is represented by a point in Σ; a single degree of freedom (DOF) motion is represented by a curve and a two DOF motion is represented by a surface in Σ [12].

2.2 Geometric constraint of RR–, PR– and RP– Dyads

Planar four-bar linkages have two dyads of types RR, PR, RP or PP connected at the ends, where R denotes a revolute– and P denotes a prismatic– joint. We do not discuss PP dyads any further as it can not effect a change in orientation. Each dyad imposes a geometric constraint on the end-effector. Fig. 2 shows the geometric constraints for three dyads. For RR, there is one point on the end-effector that lies on a circle; for PR, there is one point on the end-effector that lies on a straight line; and for RP, there is a line on the end-effector that is always tangent to a circle (the radius of this circle is zero for a swinging block like configuration). A planar motion subject to any two of the three constraints (including two of the same types) results in a 1-DOF motion called planar four-bar motion. Using kinematic mapping, these geometric constraints can be written in a unifying algebraic form. In this section, we review representations of circular and linear constraints that lead to a unified representation of planar dyad motions.

Let $\mathbf{a} = (a_1, a_2, a_0)$, where $a_0 \neq 0$, denote homogeneous coordinates of the center of a circle C in F. Then a point with

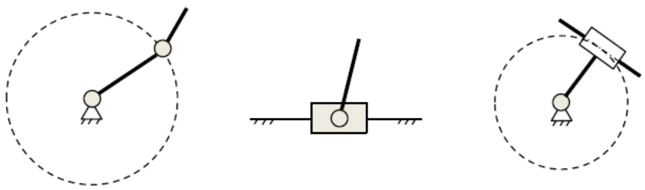


FIGURE 2. RR, PR and RP dyads and their constraints

homogeneous coordinates $\mathbf{X} = (X_1, X_2, X_3)$ lies on C if

$$2a_1X_1 + 2a_2X_2 + a_3X_3 = a_0\left(\frac{X_1^2 + X_2^2}{X_3}\right). \tag{4}$$

The radius r of the circle is given by

$$r^2 = (a_1/a_0)^2 + (a_2/a_0)^2 + a_3/a_0. \tag{5}$$

When $a_0 = 0$, Eq.(4) becomes linear,

$$L_1X_1 + L_2X_2 + L_3X_3 = 0, \tag{6}$$

which represents a line with homogeneous coordinates $\mathbf{L} = (2a_1, 2a_2, a_3)$. Thus, Eq.(4) is a unified presentation for both a circle and a line, and therefore, could lead to a unified representation of the constraints of *RR* and *PR* dyads.

For an *RP* dyad, a line with homogeneous coordinates $\mathbf{L} = (L_1, L_2, L_3)$ passes through a fixed point $\mathbf{X} = (X_1, X_2, X_3)$. In other words, they also satisfy (6).

Thus, we may conclude that all three dyadal constraints can be represented by Eq. (4) and that when $a_0 = 0$, the dyad has at least one prismatic joint.

2.3 Unified Representation of Dyadal Constraints

A planar motion subject to any two constraints listed above (including two of the same types) results in a 1-DOF motion called planar four-bar motion. By substituting (2) into (4), we have shown in [7] that the constraint manifold of an *RR* dyad is the following quadric surface

$$\begin{aligned} p_1(Z_1^2 + Z_2^2) &+ p_2(Z_1Z_3 - Z_2Z_4) + p_3(Z_2Z_3 + Z_1Z_4) \\ &+ p_4(Z_1Z_3 + Z_2Z_4) + p_5(Z_2Z_3 - Z_1Z_4) + p_6Z_3Z_4 \\ &+ p_7(Z_3^2 - Z_4^2) + p_8(Z_3^2 + Z_4^2) = 0, \end{aligned} \tag{7}$$

where the eight coefficients p_i are not independent but must satisfy two quadratic conditions

$$p_1p_6 + p_2p_5 - p_3p_4 = 0, \;\; 2p_1p_7 - p_2p_4 - p_3p_5 = 0. \tag{8}$$

This is because p_i are related to the geometric parameters of the dyad by

$$\begin{aligned} &p_1 = -a_0, \quad p_2 = a_0x \quad p_3 = a_0y, \quad p_4 = a_1, \quad p_5 = a_2, \\ &p_6 = -a_1y + a_2x, \quad p_7 = -(a_1x + a_2y)/2, \\ &p_8 = (a_3 - a_0(x^2 + y^2))/4, \end{aligned} \tag{9}$$

where (a_0, a_1, a_2, a_3) are the homogeneous coordinates of the constraint circle and (x, y) are the coordinates of the circle point. For a *PR* dyad, we have $a_0 = 0$ and therefore, $p_1 = p_2 = p_3 = 0$. Eqns. (7) and (8) are said to define the constraint manifold of *RR* and *PR* dyads.

By substituting (3) into (6), it is found that for *RP* dyad, the constraint manifold has the same form as Eqns. (7) and (8), however we now have $p_1 = p_4 = p_5 = 0$. Incidentally, for a *PP* dyad, we have $p_1 = p_2 = p_3 = p_4 = p_5 = 0$. Thus, all planar dyads can be represented in the same form by Eqns. (7) and (8), and we can determine the type of a planar dyad by looking at the zeros in the coefficients p_i (called signature of a dyad). In the image space, the general form of the quadric in Eq. (7) represents a hyperboloid of one sheet, which for RP or PR dyads degenerates into a hyperbolic paraboloid.

Lastly, Eq.(9) may be inverted to obtain the coordinates of a circle (or a line), (a_0, a_1, a_2, a_3), as well as the circle point (x, y). Let, $u = p_4^2 + p_5^2$. For *RR* and *PR* dyads, we have $u \neq 0$ and

$$\begin{aligned} &a_0 : a_1 : a_2 : a_3 = -p_1u : p_4u : p_5u : (4p_8u - p_1(p_6^2 + 4p_7^2)), \\ &x : y : 1 = (p_6p_5 - 2p_7p_4) : -(p_6p_4 + 2p_7p_5) : u. \end{aligned} \tag{10}$$

For an *RP* dyad, we have $u = 0$ and

$$\begin{aligned} &a_0 : a_1 : a_2 = (p_2^2 + p_3^2) : (-p_3p_6 - 2p_2p_7) : 2(p_2p_6 - 2p_3p_7), \\ &l_1 : l_2 : l_3 = p_2 : p_3 : 2p_8, \end{aligned} \tag{11}$$

where $\mathbf{l} = (l_1, l_2, l_3)$ are the homogenous line coordinates of a line in M, which passes through a fixed point (a_1, a_2, a_0) in F.

In our approach, we first obtain the homogeneous coordinates p_i, determine the dyad type from the signature of coefficient array p_i, and then compute the dyad parameters using (10) or (11).

3 Design Methodology

In this section, we review the synthesis methodology given in [7]. The synthesis of planar four-bar linkage in 4MDS is based on two fundamental steps: 1. fitting of given task image points to a pencil of quadrics, 2. imposing additional constraints on the pencil to reveal constraint manifold of dyads.

3.1 Data Fitting

Given N task positions ($N \geq 5$), they are first converted into N image points. Let $\mathbf{Z}_i = (Z_{i1}, Z_{i2}, Z_{i3}, Z_{i4})$ $(i = 1, 2, \ldots, N)$ denote the image points associated with specified task positions of a rigid body. If a four-bar linkage has to go through these positions, then these image points should satisfy Eq. (7). However, Burmester showed that, in general, up to only five task positions may be exactly reached by a planar four-bar linkage. For more than 5 positions, a linkage, at best, may follow given positions only approximately. This can be formulated as an over-constrained linear problem $[A]\mathbf{p} = 0$ obtained by substituting for the given values of the image points in Eq. (7), where $\mathbf{p}$ is the column vector of homogeneous coefficients $p_i (i = 1 \ldots 8)$. The coefficient matrix $[A]$ is given by:

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

where, for the ith image points, we have

$$\begin{aligned} A_{i1} &= Z_{i1}^2 + Z_{i2}^2, \quad A_{i2} = Z_{i1}Z_{i3} - Z_{i2}Z_{i4}, \\ A_{i3} &= Z_{i2}Z_{i3} + Z_{i1}Z_{i4}, \quad A_{i4} = Z_{i1}Z_{i3} + Z_{i2}Z_{i4}, \\ A_{i5} &= Z_{i2}Z_{i3} - Z_{i1}Z_{i4}, \quad A_{i6} = Z_{i3}Z_{i4}, \\ A_{i7} &= Z_{i3}^2 - Z_{i4}^2, \quad A_{i8} = Z_{i3}^2 + Z_{i4}^2. \end{aligned} \tag{13}$$

In linear algebra, the Singular Value Decomposition (SVD) (see Golub [49]) of an $N \times 8$ matrix $[A]$ is a factorization of the form:

$$[A] = [U][S][V]^T, \tag{14}$$

where $[U]$ is an $N \times N$ orthonormal matrix, $[S]$ is an $N \times 8$ rectangular diagonal matrix with 8 non-negative real numbers on the diagonal, and $[V]^T$ is an 8×8 orthonormal matrix, whose 8 columns are called the *right singular vectors*.

The over-constrained system of linear equations, $[A]\mathbf{p} = 0$, can be solved as a total least squares minimization problem with the constraint $\mathbf{p}^T\mathbf{p} = 1$. The solution turns out to be the right singular vectors of $[A]$ corresponding to the least singular values. These vectors form an orthonormal set of basis vectors spanning the nullspace of $[A]$, or in other words, solutions to $[A]\mathbf{p} = 0$. Therefore, the rank of matrix $[A]$, and consequently, its nullity (8−rank) will determine the number of zero singular values. Since the rank of $[A]$ is five, the matrix $[A]^T[A]$ has three zero eigenvalues and the corresponding eigenvectors, $\mathbf{v}_\alpha$, $\mathbf{v}_\beta$ and $\mathbf{v}_\gamma$, define the basis for the null space. Let α, β, γ denote three real parameters. Then, any vector in the null space is given by

$$\mathbf{p} = \alpha\mathbf{v}_\alpha + \beta\mathbf{v}_\beta + \gamma\mathbf{v}_\gamma. \tag{15}$$

The above also defines a pencil of quadrics in the Image Space as α, β, and γ are varied.

3.2 Constraint Imposition

For vector $\mathbf{p}$ to satisfy Eq. (8), we substitute (15) into (8) and obtain two homogeneous quadratic equations in (α, β, γ):

$$\begin{aligned} K_{10}\alpha^2 + K_{11}\beta^2 + K_{12}\alpha\beta + K_{13}\alpha\gamma + K_{14}\beta\gamma + K_{15}\gamma^2 = 0, \\ K_{20}\alpha^2 + K_{21}\beta^2 + K_{22}\alpha\beta + K_{23}\alpha\gamma + K_{24}\beta\gamma + K_{25}\gamma^2 = 0, \end{aligned} \tag{16}$$

where K_{ij} are defined by components of the three eigenvectors obtained from SVD algorithm. These two quadratic equations can be further reduced to a single quartic equation in one unknown parameter in terms of the ratio of two of the three parameters (α, β, γ) and thus can be analytically solved. Since a quartic equation may have four real roots, two real roots or no real roots, there could be four solutions, two solutions, or no solutions for the coefficients $\mathbf{p}$ of the constraint manifold of planar dyads. As coefficients $\mathbf{p}$ are homogeneous, in this paper, we normalize them such that $\mathbf{p} \cdot \mathbf{p} = 1$.

Furthermore, by investigating whether the solution $\mathbf{p}$ falls into one of the following four signature patterns, we can determine the type of the resulting dyads:

1. if $p_1 = p_2 = p_3 = 0$, the resulting dyad is a PR dyad;
2. if $p_1 = p_4 = p_5 = 0$, the resulting dyad is a RP dyad;
3. if $p_1 \neq 0$, the resulting dyad is a RR dyad.

4 Software Functionality

At its very basic, 4MDS provides a GUI-driven environment for designers to input a motion, generate candidate mechanism solutions, and evaluate and simulate possible linkage solutions. However, it is the constraint based input and the dialog based nature of environment that enables a natural user-software interaction. Figure 3 shows a flow chart of the process and the paths to synthesis and simulation. We now describe each functionality in detail:

4.1 Task-Driven Mechanism Synthesis

The primary and most innovative function of 4MDS is interactive task-driven mechanism synthesis. We describe each action during the synthesis process.

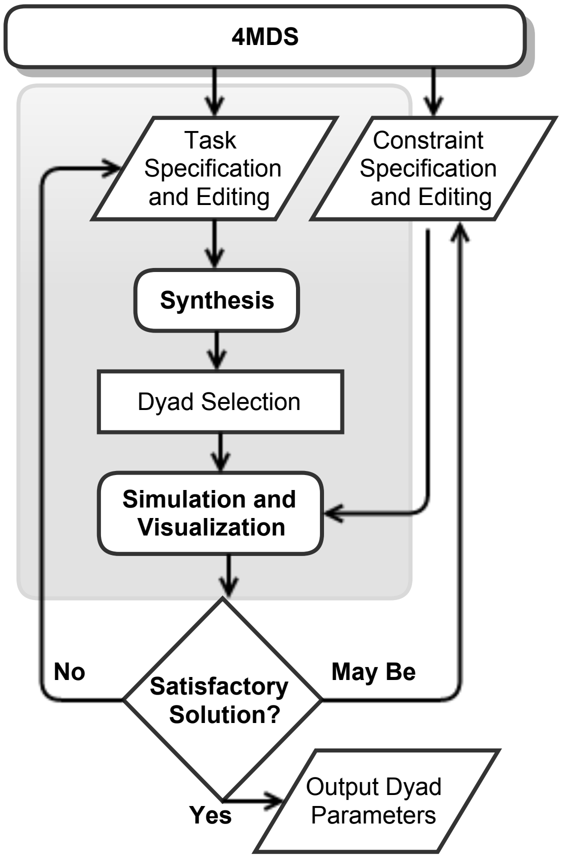


FIGURE 3. A flow chart illustrating the synthesis and simulation paths available to designer. Shaded region encompasses actions related to the task-driven synthesis process.

FIGURE 4. Icons that govern task specification and editing; from left to right: home (reset), read input file, write output, input task position, edit task position, synthesize

Task Specification and Editing Designer can input a set of tasks ($N \geq 5$) either interactively using mouse or by reading from a text file containing Cartesian parameters (d_1, d_2, ϕ). Fig. 4 shows icons relevant to this operation. Individual task positions can be edited by first triple clicking on them (selected positions are shown in grey shade), and then moving (click and drag) and re-orienting (rotate the axes) them in the workspace. When the user inputs task positions interactively, instead of reading from a file, the task positions are automatically saved in a file *data_value matrix.txt* in the folder of the executable.

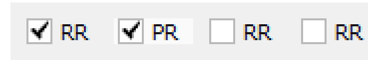


FIGURE 5. Checkboxes enable designer to select two dyads at a time.

Synthesis As soon as five tasks positions are entered by the designer interactively, dyads that can interpolate through the given positions are computed and their types are displayed at the bottom. The designer can choose two dyads (see Fig. 5) to display a four-bar linkage. As discussed before in the Design Methodology section, up to four dyads may be obtained resulting in six possible four-bar linkages. If more than five positions are entered or read from a text file, the best approximating dyads that minimize the least squares error between the given position and the moving frame attached to the end-effector are computed and shown. If designer reads task positions from an input file, then the synthesize button (right most icon in Fig. 4) should be pressed to see computed dyads.

Simulation and Visualization The interface of the software is split in two main panels (Fig. 1): Design Space Window (DSW) on the left and the Image Space Window (ISW) on the right. The designer's active interaction is only with the DSW, while the ISW shows given task image points, the constraint manifold of computed dyads, and the current position of the moving frame as a red ball. ISW facilitates intuitive, geometric visualization of image space entities and how they map to their cartesian space counterparts. For example, it is fascinating to see different intersections of the manifolds and how they correspond to different assemblies of the mechanisms. This may be of interest to researchers only, not to a practitioner, and therefore, the software permits minimizing this window.

As soon as two dyads are selected, all branches of the coupler curve in both open and crossed configurations are shown; circuits are shown in blue and red colors. The user can use play controls (Fig. 6) to animate the linkage and verify that the mechanism satisfies the motion requirements. Designer can also switch actuated dyad and see its apparent effect on how branches change. The status bar shows the classification of the chosen mechanism: Grashof, Non-Grashof, Crank-Rocker, Rocker-Rocker, etc. If none of the computed mechanisms are found satisfactory on the basis of various criteria, such as Grashof vs non-Grashof, circuit/branch/order defects, the user can interactively edit the task positions and observe the newly computed mechanisms in real time. The current version of the software does not explicitly warn the user of circuit/branch/order defects, but merely shows the user that a linkage may have such defects through the color coding of the motion path and animations.

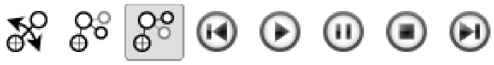


FIGURE 6. Post synthesis options; from left to right: switch actuated dyad, crossed configuration, open configuration, step backward, play, pause, stop, step forward

FIGURE 7. These icons allow designer to input dyadal constraints interactively; from left to right: assemble a mechanism mode, edit assembled mechanism mode, circle (RR-dyad) constraint, fixed-line (PR-dyad) constraint, moving-line (RP-dyad) constrain, moving frame location

Constraint Specification and Editing Constraint Specification and Editing feature of the software allows a designer to assemble a planar four-bar linkage by choosing two geometric constraints of possible mechanical dyads. The 4MDS is a geometric constraint based software not only in the sense of extracting geometric constraints from a given task, but also in specifying geometric constraints that determine a motion. The traditional approach to assembling and simulation of the mechanism is based on connecting a combination of joints and links. By focusing on the geometric constraints as means to assemble mechanisms, we not only simplify the problem, but also bring the simulation methodology closer to the CAD systems. Another advantage that accrues from this view is that the approach is scalable for spherical or spatial mechanisms. For example, for a spatial RR dyad, the geometric constraint is that of a point on a 3D sphere. From an intuition and visualization point of view, the designer would find it easier to draw a sphere describing a constraint rather than a spatial RR robot. If more than one mechanical dyads or triads can represent that geometric constraint, then we will have also achieved data reduction and simplification of assembly process.

Figure 7 shows icons for assembling a mechanism and inputting dyadal constraints.

The *May Be* decision in the flow chart of Fig. 3 may look disconcerting to the reader, however, its purpose is to reflect designer's partial satisfaction with the mechanism. For example, say, the designer is satisfied with every aspect of the mechanism except for the location of one of the fixed pivots. If the motion requirements are not exacting, he can invoke the constraint editing mode wherein every parameter of the mechanism can be edited interactively. Since the mechanism-to-motion change is well-behaved, this dialog between the user and the software leads to satisfaction of practical, geometric or otherwise, constraints imposed in a variety of applications.

This feature of the software can also be used for reverse-engineering a mechanism. If a designer wants to generate alternative mechanism concepts for a known mechanism, he can assemble the known mechanism and the software will automatically store five arbitrary task positions in generated motion in a text file *GeneratedMotion.txt*. Next, the designer can switch to synthesis mode and read input task positions from this file. The software can provide up to six new mechanisms (including the original one) that can generate this motion. Admittedly, the new mechanisms are highly dependent on the chosen five positions and a different choice of the task positions may lead to different mechanisms. At the same time, arbitrary choices may not be the best either. Characterizing a motion by a few task positions that truly represent a motion is a topic of further research. However, from a user interface perspective, we could easily incorporate designer specified task positions critical to the generated motion.

```
5
-0.8253    -1.0329    91.91
-1.6554    -0.2721    66.99
-2.6392     0.2920    49.90
-3.5027     0.3088    43.94
-4.0549    -0.2141    45.58
```

FIGURE 8. The first line should contain number of task positions, and the subsequent lines contain translation and orientation data (d_{1i}, d_{2i}, ϕ_i).

File I/O For file input, task positions can be read from a text file (Fig. 8). When dyads are synthesized, all the computed dyads and their parameters are automatically saved in a file named *dyad parameters.txt* inside the folder of the program executable. The user can also save individual synthesized linkages' parameters using save icon. The symbols and named parameters in the output files follow the convention in [7].

5 Software Architecture

4MDS implemented using standard C++ for core programming, OpenGL for 3D graphics, LAPACK for linear algebra operations, and Qt for GUI design employs an Object-Oriented Programming (OOP) and Design approach. The software's extensible and scalable architecture implemented using essential OOP traits such as inheritance, composition, operator overloading, and polymorphism allows extension to more complex planar as well as spatial linkages. The software has following four main subkernels.

1. **GUI**: Graphical User Interface of 4MDS is implemented using open source cross platform UI framework Qt under LGPL.
2. **Mechanism and Motion**: This package contains an abstract base class (ABC) Dyad from which more concrete RP, PR, and RR dyad classes inherit. There is a Mecha-

nism ABC from which PlanarFourBarMechanism class inherits. The PlanarFourBarMechanism contains two Dyad classes by composition. These classes interact with Motion and Branch classes.
3. **Geometry and Math:** This contains classes for creation of geometric objects, such as links, joints, manifold for the image space as well as classes for vector and matrix computation. This also includes third party library LAPACK for SVD operation.
4. **Kinematics:** This package contains classes for Quaternion operations and conversion to matrix form.

6 Example

We now present a step by step design scenario for a five task positions example.

TABLE 1.

d_1	d_2	ϕ (degree)
-7.81	-9.65	22.9
-3.80	-8.11	24.7
-0.088	-1.30	18.9
0.54	1.61	-6.59
1.90	3.33	-39.9

1. Select Input Task icon and Enter five positions interactively; data for the five positions entered are automatically saved in the file *data_value matrix.txt*. Table 1 gives this data.
2. As soon as the positions are entered, dyads are computed. Select two dyads (there are only two computed in this case) from the check boxes in the bottom row.
3. Swap actuating dyad, switch between open and crossed configuration and run simulation to observe the mechanism behavior: examine circuit, position traversal order, and type of the computed mechanism
4. If the mechanism does not satisfy designer's criteria, tweak task positions or modify the mechanism directly using Constraint Editing Mode
5. Save dyad parameters data; this data is automatically saved in a file *dyad parameters.txt*. Table 2 contains the computed dyad parameters data.

Figure 1 shows the final synthesized mechanism.

For assembling a mechanism manually, the user has to use the strip of icons shown in Fig. 7. First, enter the Constraint Specification/Editing mode (first icon) and draw two constraints – the strip has three icons for drawing circle–, fixed line–, or moving line–constraints – one at a time. After entering two constraints, the designer has to specify the moving frame location (last icon). See Fig. 9 for an example of an RRRR mechanism assembled. To edit the constraints, use the second icon and interactively click and drag the elements. Once the assembly is complete, play controls can be used, actuating dyad can be swapped, and different circuits can be examined.

TABLE 2.

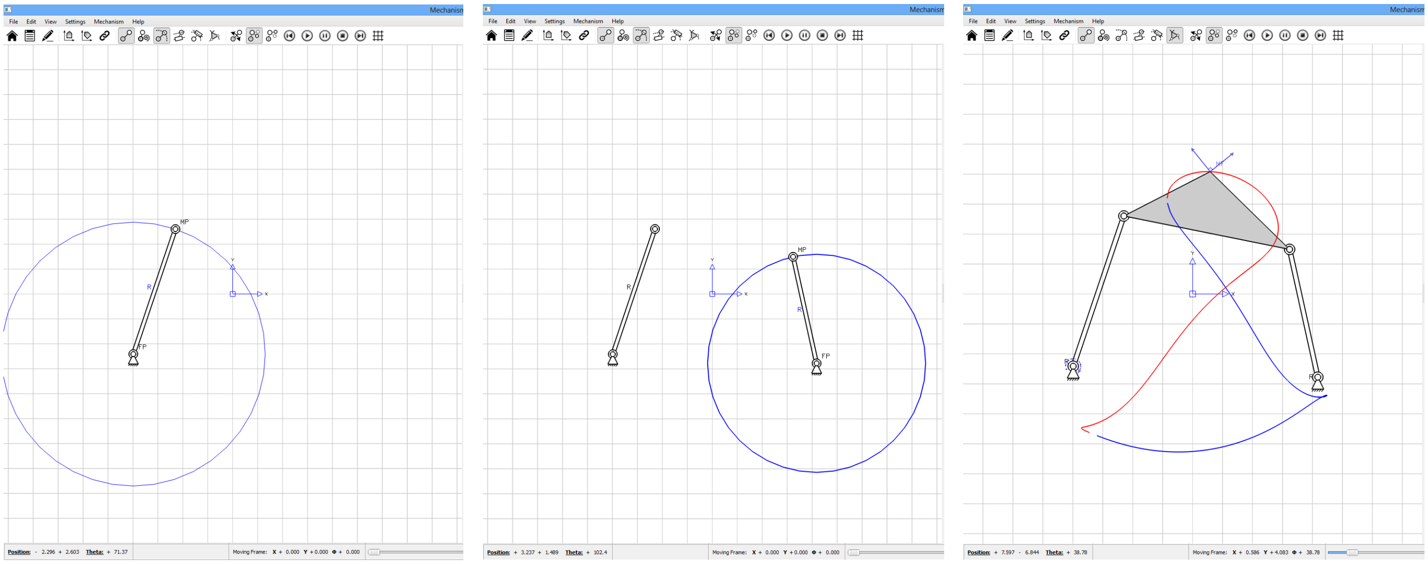
Dyad Type	a_1/a_0	a_2/a_0	x_1/x_3	x_2/x_3	r
RR	-6.09	-1.45	2.48	-1.77	8.95
RR	4.28	-0.042	13.32	-6.57	10.83

7 Conclusions and Future Work

This paper presented a novel and unique planar four bar mechanism software system 4MDS, which can compute both type and dimensions simultaneously for a given motion task. The underlying theory is based on algebraic fitting of a pencil of manifolds in the image space of planar displacements which results into an efficient algorithm for task-driven mechanism synthesis. This, in turn, facilitates real-time computation and a highly responsive interactive system wherein the designer can freely change controlling parameters and see results change dynamically on the screen. Both the simulation and synthesis process are completely geometric constraint driven and makes the designing process transparent to user. Multiple paths to motion generation enable a more natural and fluid dialog between user and the software, where the designer is free to exercise his creativity and is not constrained by artificial limitations of a database or off-line computation. The software, at the time of writing of this paper, is being released as a beta software. The immediate future work would entail improving GUI, allowing designer to input other practical constraints, such as geometric ones, less than 5 positions synthesis, automated evaluation and filtering of computed linkages based on criteria such as circuit/branch/order defect, transmission angle, adding tolerance to task positions for creating a large number of linkages, and exact numerical input for assembling mechanisms.

ACKNOWLEDGMENT

The work has been financially supported by a Technology Accelerator Fund Award by Research Foundation of SUNY to Anurag Purwar. All findings and results presented in this paper are those of the authors and do not represent those of the funding agencies. We also acknowledge the help of graduate students Xiangyun Li and Xin Ge, who are working on an iOS version of a linkage design software.



(a) First Circle Constraint (b) Second Circle Constraint (c) Moving frame specified and assembly complete

FIGURE 9. Assembling a mechanism requires drawing two constraints and specifying moving frame location

REFERENCES

[1] McCarthy, J. M. and Soh, G. S., 2010, Geometric Design of Linkages, Springer, 2nd edition.

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

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

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

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

[6] Norton, R., 2011, Design of Machinery: An Introduction To The Synthesis and Analysis of Mechanisms and Machines, McGraw Hill, 5th edition.

[7] Ge, Q. J., Zhao, P., and Purwar, A., 2013, "A Task Driven Approach to Unified Synthesis of Planar Four-bar Linkages using Algebraic Fitting of a Pencil of G-manifolds", ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, dETC2013-12977.

[8] Blaschke, W., 1911, "Euklidische Kinematik und nichteuklidische Geometrie", Zeitschr. Math. Phys., **60**, pp. 61–91 and 203–204.

[9] Grunwald, J., 1911, "Ein Abbildungsprinzip, welches die ebene Geometrie und Kinematik mit der raumlichen Geometrie verknupft", Sitzber. Ak. Wiss. Wien, **120**, pp. 677–741.

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

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

[12] 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.

[13] Ravani, B. and Roth, B., 1984, "Mappings of spatial kinematics", Journal of Mechanisms Transmissions and Automation in Design-Transactions of the ASME, **106(3)**, pp. 341–347.

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

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

[16] Erdman, A. and Gustafson, J., 1977, "LINCAGES: Linkage INteractive Computer Analysis and Graphically Enhanced Synthesis Packages,", Technical report.

[17] Erdman, A. G. and Riley, D., "Computer-aided linkage design using the LINCAGES package", ASME Design Engineering Technical Conferences, ASME Press, Hartford, CN, USA, paper Number 81-DET-121.

[18] Kihonge, J., Vance, J., and Larochelle, P., "Spatial mechanism design in virtual reality with networking", ASME 2001 Design Engineering Technical Conferences.

[19] Larochelle, P., 1998, "Spades: Software for Synthesizing Spatial 4C Linkages", CD-ROM Proc. of the ASME

DETC'98, paper no.DETC98/Mech- 5889.
[20] Larochelle, P., Dooley, J., Murray, A., and McCarthy, J. M., 1993, "SPHINX: Software for Synthesizing Spherical 4R Mechanisms", Proc. of the 1993 NSF Design and Manufacturing Systems Conference, volume 1, pp. 607–611.
[21] Ruth, D. and McCarthy, J., 1997, "Sphinxpc: An implementation of four position synthesis for planar and spherical 4r linkages", ASME Design Engineering Technical Conferences.
[22] Tse, D. and Larochelle, P., 1999, "Osiris: a new generation spherical and spatial mechanism cad program", Florida Conference on Recent Advancements in Robotics.
[23] Su, H.-J., Collins, C., and McCarthy, J., 2002, "An Extensible Java Applet for Spatial Linkage Synthesis", ASME International Design Engineering Technical Conferences.
[24] Purwar, A., Anantwar, S., and Zhao, P., "An Interactive Approach to Designing Planar Parallel Manipulators using Image Space Representation", ASME 2012 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, ASME, paper No. DETC2012-70880.
[25] Wu, J., Purwar, A., and Ge, Q. J., 2010, "Interactive Dimensional Synthesis and Motion Design of Planar 6R Single-Loop Closed Chains via Constraint Manifold Modification", ASME Journal of Mechanisms and Robotics, **2**, p. 31012(8 pages).
[26] Purwar, A. and Gupta, A., "Visual Synthesis of RRR- and RPR-legged Planar Parallel Manipulators using Constraint Manifold Geometry", ASME 2011 International Design Engineering Technical Conferences & Computers and Information in Engineering, ASME, paper No. DETC2011-48830.
[27] SyMech, "http://www.symech.com/", .
[28] Technologies, H., "WATT, `http://www.heron-technologies.com/watt/`", .
[29] "Special Session on Computer Aided Linkage Synthesis", ASME IDETC.
[30] Engineering, A., "SAM (Synthesis and Analysis of Mechanisms), `http://www.artas.nl/en`", .
[31] Associates, M. D., "MechGen, `http://mechanicaldesign101.com/mechanism-generator-2-0/#MechGen3`", .
[32] Dassault Systems, "SolidWorks Software", `http://www.solidworks.com`, accessed Jan 27, 2014.
[33] Engineering, N. A., "Linkages, `http://www.designofmachinery.com/Linkage/index.html`", .
[34] Technologies, D. S., "Working Model 2D, `http://www.design-simulation.com/WM2D/`", .
[35] SoftIntegration, "Ch Mechanism Toolkit, `http://www.softintegration.com/products/toolkit/mechanism/`", .
[36] Software, M., "Adams, `http://www.mscsoftware.com/product/adams`", .
[37] Software, S. P., "NX unigraphics, `http://www.plm.automation.siemens.com`", .
[38] Corp, P. T., "PTC Creo, `http://www.ptc.com/product/creo/`", .
[39] Systems, D., "Solidworks Motion, `http://www.solidworks.com/sw/products/simulation/motion-analysis.htm`", .
[40] Technologies, K., "The Geometer's Sketchpad, `http://www.dynamicgeometry.com/`", .
[41] Institute, I. G., "Geogebra, `http://www.geogebra.org/cms/`", .
[42] Company, G., "GrafiCalc, `http://www.graficalc.com`", .
[43] Consortium, T. C., "Molecular Workbench, `http://mw.concord.org/modeler/`", .
[44] Useful Subset, L., "Solvespace, `http://solvespace.com`", .
[45] Software, S., "Analytix, `http://www.saltire.com/ax.html`", .
[46] Kinzel, E. C., Schmiedeler, J. P., and Pennock, G. R., 2006, "Kinematic Synthesis for Finitely Separated Positions Using Geometric Constraint Programming", ASME Journal of Mechanical Design, **128(5)**, pp. 1070–1079.
[47] Ltd., P. M., "MechDesigner, `http://www.psmotion.com/`", .
[48] Laboratory of Computational Mechanics, R., Bryansk State Technical University, "Universal Mechanism", .
[49] Golub, G. and Van Loan, C., 1996, Matrix Computations, Johns Hopkins Univ Press, Baltimore, MD.