Graphical User Interface to Solve the Burmester Problem

Julien S. Bourrelle* Chao Chen* Stéphane Caro† Jorge Angeles†

Departement of Mechanical Engineering
and Centre for Intelligent Machines
McGill University, Montreal, Canada

Abstract—*The classic Burmester problem aims at finding the geometric parameters of a planar four-bar linkage for a prescribed set of finitely separated poses. In this paper, we introduce a Matlab-based graphical user interface to support the solution of the classic Burmester problem. The synthesis deals with both revolute-revolute (RR) and prismatic-revolute (PR) dyads for a prescribed set of five poses. Issues such as numerical conditioning, multiplicity of solutions and singularities are considered. A case study shows the applicability and robustness of the numerical procedure in a design environment.*

Keywords: graphical user interface, Burmester problem, four-bar linkage, numerical conditioning, five-pose synthesis

I. Introduction

The Burmester problem consists in designing a planar four-bar linkage to guide a rigid-body through a set of finitely-separated poses. It is well known that four-bar linkages may be synthesized exactly for up to five prescribed poses [1].

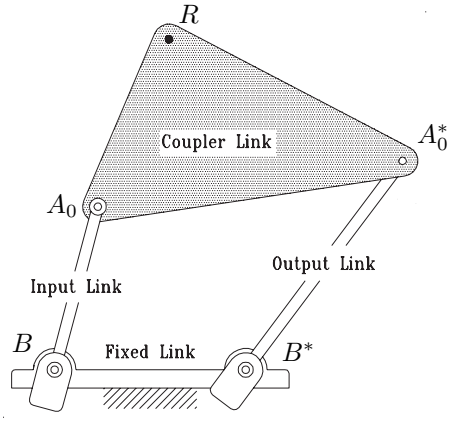

Fig. 1. A four-bar linkage with revolute-revolute dyads

The Burmester problem, in its full generality, can be stated as: *A rigid body, attached to the coupler link of a four-bar linkage, as shown in Fig. 1, is to be guided through a discrete set of m poses, given by $\{\mathbf{r}_j, \theta_j\}_0^m$, starting with a reference pose labeled 0, where $\mathbf{r}_j$ is the position vector of a landmark point R of the body at the jth pose and θ_j is the corresponding angle of a line of the body, as depicted in Fig. 2*, where (R_0, X_0, Y_0) is the reference frame. The problem consists in finding the joint centers A_0 and B that define the BA_0R *dyad* of the guiding four-bar linkage, dyad $B^*A_0^*R$ being determined likewise. As A_0 and A_0^* describe circles centered at B and B^*, respectively, the former are termed the *circlepoints*, the latter the *centerpoints* of the dyads.

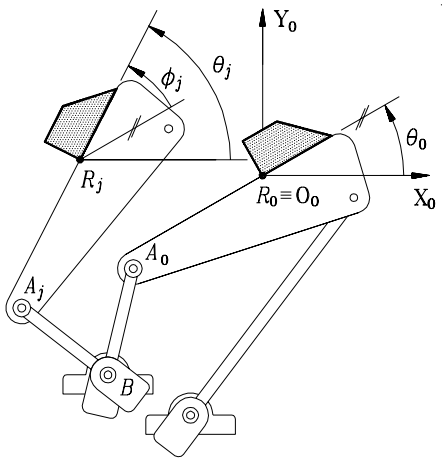

Fig. 2. Two finitely separated poses of a rigid body carried by the coupler link of a four-bar linkage

The four-pose problem is known to admit an infinity of solutions, each dyad being given by a pair of corresponding cubic curves, the *centerpoint curve* and the *circlepoint curve*. Commercial packages, such as LINCAGES 2000 [2], [3], are available for obtaining solutions to this problem, these packages dealing mostly with the synthesis of RR dyads. On the other hand, the five-pose problem is known to lead to the solution of a quartic equation, and hence, admits none, two or four real dyads [1]. Extensive research has been reported on the solution of the Burmester problem [4]–[11]. Among all these works, most focus on revolute-revolute RR dyads, few investigating the problem of prismatic-revolute PR dyads.

This paper introduces a graphical user interface (GUI) that solves the five-pose problem based on the robust computational algorithm developed by Angeles, Al-Widyan and Cervantes [12] and Angeles and Bai [13] for both RR dyads

*E-mail: {julien.bourrelle, chao.chen}@elf.mcgill.ca
†E-mail: {caro, angeles}@cim.mcgill.ca

and PR dyads. The various interface sections are discussed, while the underlying theory is outlined . We refer the user to several case studies available online that demonstrate the applicability of the package in a design environment.

II. Data Importing

The data-importing options permit the study of problems ranging from fully defined sets of five poses, entered directly from the keyboard, to design-oriented problems defined by up to fifteen poses, imported from a file. Within the interface, different sets of five poses may be selected from the imported data and tested independently. If no adequate solution is obtained for the given set of data poses, alternative poses may be tried from a rich set using an interactive graphical selection method, available in our GUI.

A. Cubic Spline Curves

Spline curves are used to interpolate data poses, thereby obtaining a continuous path for the landmark point and a continuous function for θ, as shown in Fig. 3. The set of landmark-point positions is interpolated by means of a *parametric periodic cubic spline*. The orientation angles are interpolated, in turn, by a *non-parametric periodic cubic spline*, proper for the interpolation of functions, as opposed to geometric curves. Moreover, the parameter s, defined as the length measured along the polygonal formed by the landmark points, is used as the abscissa of the non-parametric spline interpolating θ. Hard-to-meet pose orderings are readily identified with the aid of the spline interpolants prior to the synthesis.

B. Extraction of Poses from the Spline Interpolants

Design-oriented problems defined from client's needs often admit slight variations of the given poses. The spline plots permit the selection of new poses selected by the user by clicking, first, at an interpolated point on the spline-generated (pseudo[1]) coupler curve; the GUI then responds with a point on the spline-generated, plot of function $\theta = \theta(s)$. The selection, moreover, is not restricted to the spline-generated plots; full flexibility is given to the user. The point on the coupler-curve spline closest to the selection is used to compute the orientation angle by means of the orientation spline. Markers are added on both plots for validation and the new pose is added to the table.

III. Synthesis Equations

We start the synthesis with a general four-bar linkage, as shown in Fig. 2. Under the usual rigid-body assumption, the *synthesis equation* is readily derived as

$$\mathbf{b}^T(\mathbf{1}-\mathbf{Q}_j)\mathbf{a}_0+\mathbf{r}_j^T\mathbf{Q}_j\mathbf{a}_0-\mathbf{r}_j^T\mathbf{b}+\frac{\mathbf{r}_j^T\mathbf{r}_j}{2}=0\,,\ j=1,\ldots,m \tag{1}$$

[1]The closed curve appearing in the upper right window of Fig. 3 is not, properly speaking, a coupler curve, for it is not a sextic.

where $\mathbf{a}_0$ and $\mathbf{b}$ are the position vectors of points A_0 and B, the design parameters of the RR dyad, while $\mathbf{Q}_j$ denotes the rotation matrix carrying the guided body from pose 0 to pose j, i.e.,

$$\mathbf{Q}_j=\begin{bmatrix}\cos\phi_j & -\sin\phi_j\\ \sin\phi_j & \cos\phi_j\end{bmatrix},\ \text{with}\ \phi_j\equiv\theta_j-\theta_0 \tag{2}$$

The synthesis equations (1) allow us to compute the design parameters. The equations are obtained in symbolic form, free of round-off error by means of computer-algebra software.

A. Synthesis of RR dyads

In order to find the RR dyads of the four-bar linkage, we first eliminate $\mathbf{a}_0$ from Eq. (1), which can be achieved by rewriting this equation as:

$$\mathbf{G}\mathbf{z}=\mathbf{0}_m \tag{3}$$

where $\mathbf{0}_m$ is the m-dimensional zero vector, $\mathbf{G}$ is a $m\times 3$ matrix linear function of $\mathbf{a}_0$, and $\mathbf{z}=\begin{bmatrix}\mathbf{b}^T\ 1\end{bmatrix}^T$ is the *three-dimensional array of homogeneous coordinates* of B. Thus, for $j=1,\ldots,m$,

$$\mathbf{G}\equiv\begin{bmatrix}\mathbf{g}_1^T\\ \vdots\\ \mathbf{g}_m^T\end{bmatrix},\ \mathbf{g}_j=\begin{bmatrix}(\mathbf{1}-\mathbf{Q}_j)\mathbf{a}_0-\mathbf{r}_j\\ \mathbf{r}_j^T\mathbf{Q}_j\mathbf{a}_0+\mathbf{r}_j^T\mathbf{r}_j/2\end{bmatrix} \tag{4}$$

By the same token, $\mathbf{b}$ can be eliminated instead of $\mathbf{a}_0$ from Eq. (1) by rewriting this equation in the form

$$\mathbf{H}\mathbf{w}=\mathbf{0}_m \tag{5}$$

where $\mathbf{H}$ is a $m\times 3$ matrix linear function of $\mathbf{b}$, while $\mathbf{w}=\begin{bmatrix}\mathbf{a}_0^T\ 1\end{bmatrix}^T$ is the *three-dimensional array of homogeneous coordinates* of A_0. In this case, for $j=1,\ldots,m$,

$$\mathbf{H}\equiv\begin{bmatrix}\mathbf{h}_1^T\\ \vdots\\ \mathbf{h}_m^T\end{bmatrix},\mathbf{h}_j=\begin{bmatrix}(\mathbf{1}-\mathbf{Q}_j^T)\mathbf{b}+\mathbf{Q}_j^T\mathbf{r}_j\\ -\mathbf{r}_j^T\mathbf{b}+\mathbf{r}_j^T\mathbf{r}_j/2\end{bmatrix} \tag{6}$$

When five poses are specified, m is equal to 4, $\mathbf{G}$ becoming a 4×3 matrix. Given its definition, $\mathbf{z}$ cannot vanish, eq. (3) thus being required to admit nontrivial solutions. These are possible if $\mathbf{G}$ is rank-deficient, i.e. iff every 3×3 submatrix $\mathbf{G}_j$, for $j=1,\ldots,4$, of $\mathbf{G}$ is singular:

$$\mathcal{K}_j:\ \Delta_j(\mathbf{a}_0)=\Delta_j(x,y)=\det(\mathbf{G}_j)=0 \tag{7}$$

where $\mathbf{G}_j$ is formed by deleting the jth row from $\mathbf{G}$. Each of the four equations (7) defines one contour in the x-y plane, thereby leading to the four circlepoint curves $\mathcal{K}_j$ corresponding to the three synthesis equations obtained when deleting the jth equation from the given four in Eq. (1). Their common intersections yield the real circlepoints of the RR dyad.

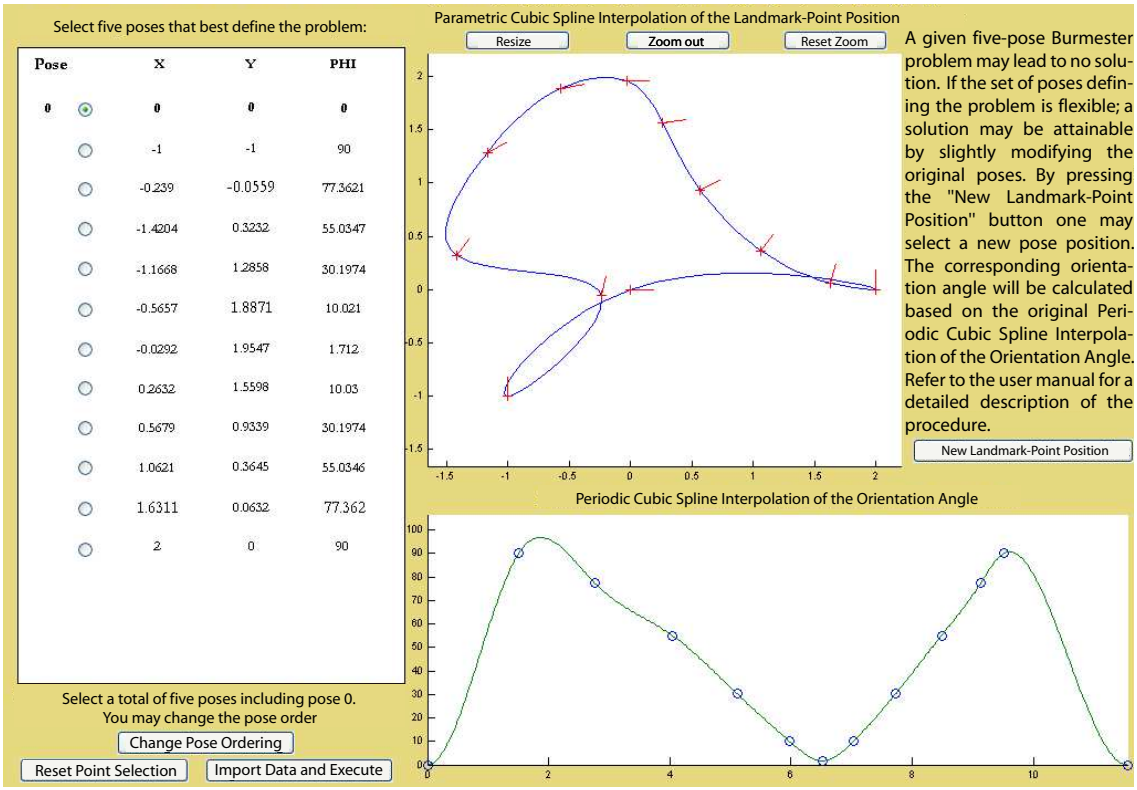

Fig. 3. Data-importating-window using cubic-splines

Likewise, the centerpoint curves $\mathcal{M}_j$ can be found from $\mathbf{H}$:

$$\mathcal{M}_j : \tilde{\Delta}_j(\mathbf{b}) = \tilde{\Delta}_j(u, v) = \det(\mathbf{H}_j) = 0 \qquad (8)$$

with $\mathbf{H}_j$, for $j = 1, \ldots, 4$, defined as its $\mathbf{G}_j$ counterpart, the common intersections of the four $\mathcal{M}_j$ curves yielding the centerpoints. Therefore, every triplet of synthesis equations (1) out of the given $m \geq 4$, defines one centerpoint and one circlepoint curve.

The expressions for the four circlepoint curves $\mathcal{K}_j$ and the four centerpoint curves $\mathcal{M}_j$ are again obtained in symbolic form. Symbolic calculations are done within *Matlab* using the *Maple Kernel*.

B. Synthesis of PR dyads

To simplify the derivation of the synthesis equations for PR dyads, we divide both sides of Eq. (1) by the Euclidean norm of $\mathbf{b}$, thus obtaining, for $j = 1, \ldots, m$,

$$[(\mathbf{1} - \mathbf{Q}_j)\mathbf{a}_0 - \mathbf{r}_j]^T \frac{\mathbf{b}}{\|\mathbf{b}\|} + \left(\mathbf{r}_j^T \mathbf{Q}_j \mathbf{a}_0 + \frac{\mathbf{r}_j^T \mathbf{r}_j}{2}\right) \frac{1}{\|\mathbf{b}\|} = 0 \qquad (9)$$

Furthermore, we define a unit vector $\boldsymbol{\beta}$ as

$$\boldsymbol{\beta} = \frac{\mathbf{b}}{\|\mathbf{b}\|} \qquad (10)$$

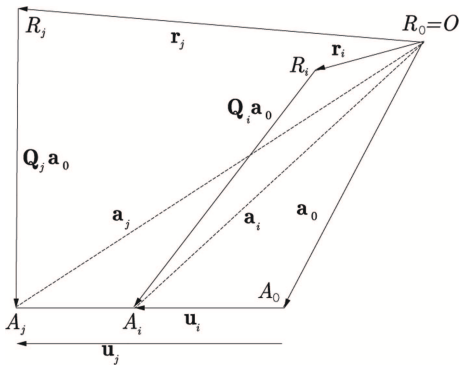

Fig. 4. Relation between the ith and jth poses and the circlepoints

When $\|\mathbf{b}\| \to \infty$, the centerpoint B goes to infinity, which leads to a PR dyad. Under this condition, Equation (9) can be rewritten as

$$\mathbf{u}_j^T \boldsymbol{\beta} = 0 \qquad (11)$$

where $\mathbf{u}_j \equiv \mathbf{a}_j - \mathbf{a}_0$ is the displacement of the circlepoint A_0 at the jth pose, i.e.,

$$\mathbf{u}_j = \mathbf{r}_j - (\mathbf{1} - \mathbf{Q}_j)\mathbf{a}_0, \quad j = 1, \ldots, m \qquad (12)$$

With reference to Fig. 4, $\{\mathbf{u}_i\}_1^m$ is the ith displacement vector of the circlepoint. From eq. (11), all m vectors $\mathbf{u}_i$

must be parallel. In other words, the cross product of any two displacement vectors must vanish. If we resort to the 2D representation of the cross product introduced in [14], then the cross product can be expressed as

$$\Delta_{ij} = \mathbf{u}_i^T \mathbf{E} \mathbf{u}_j = 0, \quad i,j = 1,\ldots,m, \quad i \neq j \quad (13)$$

where $\mathbf{E}$ is a 2×2 matrix rotating vectors in the plane $90°$ ccw. Equation (13) expands to

$$\begin{aligned}\Delta_{ij} = \mathbf{a}_0^T(-\mathbf{E}\mathbf{Q}_i - \mathbf{Q}_j^T\mathbf{E} + \mathbf{Q}_j^T\mathbf{E}\mathbf{Q}_i)\mathbf{a}_0 \\ + (\mathbf{E}\mathbf{r}_i - \mathbf{Q}_j^T\mathbf{E}\mathbf{r}_i - \mathbf{E}\mathbf{r}_j + \mathbf{Q}_i^T\mathbf{E}\mathbf{r}_j)^T\mathbf{a}_0 \\ + \mathbf{r}_j^T\mathbf{E}\mathbf{r}_i = 0 \end{aligned} \quad (14)$$

We develop below all quadratic terms of Eq. (14), those in the first line of this equation, by writing $\mathbf{Q}_i$ in the form $\mathbf{Q}_i = c_i\mathbf{1} + s_i\mathbf{E}$, in which $s_i \equiv \sin\phi_i$ and $c_i \equiv \cos\phi_i$. Hence,

$$-\mathbf{a}_0^T\mathbf{E}\mathbf{Q}_i\mathbf{a}_0 = s_i||\mathbf{a}_0||^2 \quad (15a)$$

$$-\mathbf{a}_0^T\mathbf{Q}_j^T\mathbf{E}\mathbf{a}_0 = -s_j||\mathbf{a}_0||^2 \quad (15b)$$

$$\mathbf{a}_0^T\mathbf{Q}_j^T\mathbf{E}\mathbf{Q}_i\mathbf{a}_0 = -\sin(\phi_i - \phi_j)||\mathbf{a}_0||^2 \quad (15c)$$

Further, let $\mathbf{v}_{ij} = -\mathbf{E}\mathbf{r}_i + \mathbf{Q}_j^T\mathbf{E}\mathbf{r}_i + \mathbf{E}\mathbf{r}_j - \mathbf{Q}_i^T\mathbf{E}\mathbf{r}_j$, as appearing in the second line of Eqs. (14), which are now rewritten for $i,j = 1,\ldots,m, i \neq j$, as,

$$\Delta_{ij} = (s_i - s_j - s_{i\bar{j}}) \parallel \mathbf{a}_0 \parallel^2 + \mathbf{v}_{ij}^T\mathbf{a}_0 + \mathbf{r}_j^T\mathbf{E}\mathbf{r}_i = 0 \quad (16)$$

and represent the loci of A_0, of position vector $\mathbf{a}_0$, namely, a family of circles, where $s_{i\bar{j}} = \sin(\phi_i - \phi_j)$.

IV. Determination of the Mechanism Type

To simplify the interface, the type of mechanism, for $m = 4$, is determined prior to the synthesis. By rewritting Eq. (16) as

$$\mathbf{N}\mathbf{x} = \mathbf{0}_4 \quad (17)$$

where $\mathbf{x} = [\ \ \|\mathbf{a}_0^T\|^2 \ \ \mathbf{a}_0^T \ \ 1\ \ ]^T$ and

$$\mathbf{N} = \begin{bmatrix} s_1 - s_2 + s_{1\bar{2}} & \mathbf{v}_{12}^T & \mathbf{r}_2^T\mathbf{E}\mathbf{r}_1 \\ s_2 - s_3 + s_{2\bar{3}} & \mathbf{v}_{23}^T & \mathbf{r}_3^T\mathbf{E}\mathbf{r}_2 \\ s_3 - s_4 + s_{3\bar{4}} & \mathbf{v}_{34}^T & \mathbf{r}_4^T\mathbf{E}\mathbf{r}_3 \\ s_4 - s_1 + s_{4\bar{1}} & \mathbf{v}_{41}^T & \mathbf{r}_1^T\mathbf{E}\mathbf{r}_4 \end{bmatrix} \quad (18)$$

the mechanism type can determined from the value of $\det(\mathbf{N})$. Two special cases can be identified for which $\mathbf{N}$ is singular, i.e. its determinant vanishes:

(a) The first column vanishes. In this case, the orientation of the coupler remains constant. A possible mechanism is the parallelogram.

(b) The last column vanishes. In this case, all given displacement vectors $\{\mathbf{u}_i\}_1^4$ are parallel. A mechanism containing a prismatic joint is possible iff all orientation angles are different. It is noteworthy that the landmark points may or not be aligned.

If $\mathbf{N}$ is nonsingular, then the synthesis produces only RR dyads.

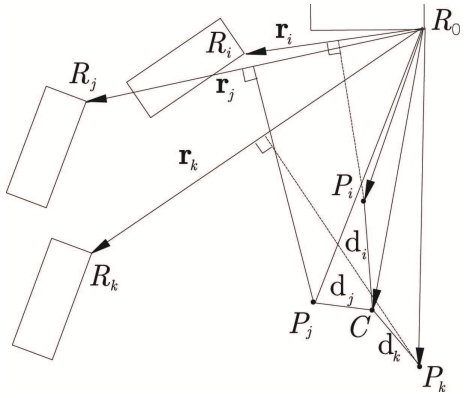

Fig. 5. Illustration of vector normalization: determination of the poles P_i and their centroid

A. Normalization of the Displacement Vector

To reduce the effect of data dimension on roundoff error, normalized position vectors are desirable. To this end, we make all $\{\mathbf{r}_j\}_1^4$ dimensionless by means of a suitable normalization [13].

As depicted in Fig. 5, the displacement from the reference pose to the ith pose can be considered as a pure rotation about the pole P_i. Distances $\{d_i\}_1^m$ between P_i and the centroid C of $\{P_i\}_1^m$ are first obtained. The rms value d_{rms} of these distances is considered as the *characteristic length* for normalization, such that the displacement vectors are normalized as

$$\boldsymbol{\rho}_i \equiv \frac{\mathbf{r}_i}{d_{rms}}, \quad i = 1,\ldots,m \quad (19)$$

with

$$d_{rms} = \sqrt{\frac{1}{m}\sum_{j=1}^{m} d_i^2}, \quad d_i = \parallel \mathbf{p}_i - \mathbf{c} \parallel, \quad \mathbf{c} = \frac{1}{m}\sum_{j=1}^{m}\mathbf{p}_i \quad (20)$$

To find $\mathbf{p}_i$, we refer to Fig. 5, whence,

$$\mathbf{p}_i = \frac{\mathbf{r}_i}{2} + l_i\mathbf{E}\hat{\mathbf{r}}_i, \quad \hat{\mathbf{r}}_i \equiv \frac{\mathbf{r}_i}{l_i}, \quad i = 1,\ldots,m \quad (21a)$$

and

$$l_i = \frac{\parallel \mathbf{r}_i \parallel}{2\tan(\phi_i/2)}, \quad \phi_i \neq 0 \quad (21b)$$

If the rigid-body displacement from the reference pose 0 to the ith pose is a pure translation, then P_i lies at infinity. In this case d_i cannot be computed from eq.(20) and, ϕ_i vanishing, l_i cannot be computed from eq.(21b). What we do in this case is define d_i as the absolute value of the translation displacement.

V. Determination of the Circlepoints and Centerpoints

The problem at hand reduces to finding the solution of an overdetermined nonlinear system of equations given by

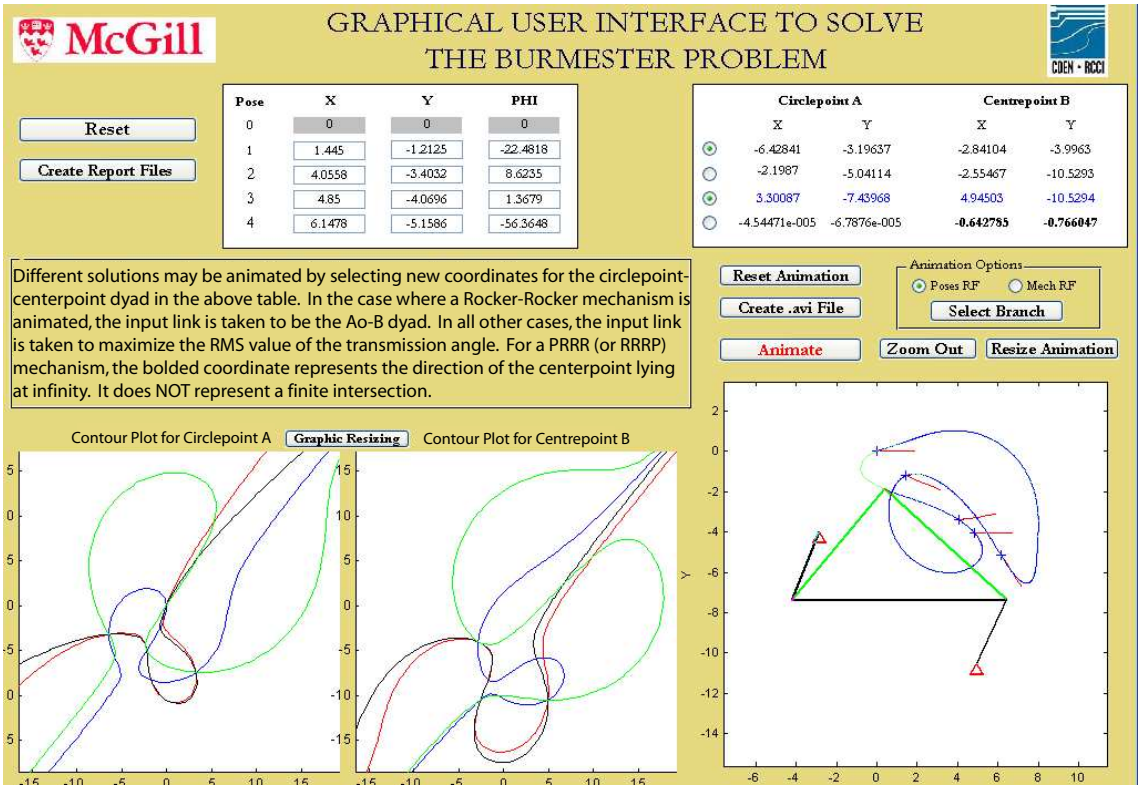

Fig. 6. Main window of the graphical user interface

eqs.(4) and (6) for $\mathbf{a}_0$ and $\mathbf{b}$, respectively. Rough estimates of the real solutions are to be user-produced by inspection of the intersections of the contour plots defined by each set of circlepoint curves $\mathcal{K}_j$ and centerpoint curves $\mathcal{M}_j$, as shown in Fig. 6. The estimates are refined up to machine accuracy by resorting to a nonlinear least-square approximation of an overdetermined system of four nonlinear equations in two unknowns namely,

$$\|\mathbf{f}\|^2 \equiv \frac{1}{2}\sum_{i=0}^{4}[\Delta_i(\mathbf{x})]^2 \quad \rightarrow \quad \min_{\mathbf{x}} \qquad (22)$$

with $\mathbf{x}$ denoting either $\mathbf{a}_0$ or $\mathbf{b}$ and $\mathbf{f}$ defined, correspondingly, in any of the two forms below:

$$\mathbf{f}(\mathbf{a}_0) = \begin{bmatrix} \Delta_1(\mathbf{a}_0) \\ \Delta_2(\mathbf{a}_0) \\ \Delta_3(\mathbf{a}_0) \\ \Delta_4(\mathbf{a}_0) \end{bmatrix} \qquad \mathbf{f}(\mathbf{b}) = \begin{bmatrix} \Delta_1(\mathbf{b}) \\ \Delta_2(\mathbf{b}) \\ \Delta_3(\mathbf{b}) \\ \Delta_4(\mathbf{b}) \end{bmatrix}$$

It is noteworthy that solving for the unknowns while taking into account all the redundant equations we enhance the algorithm robustness.

A. Calculations and Graphical Representation

The user-estimated solutions are refined using the above described algorithm, thus permitting the immediate identification of the real solutions. Both estimated and refined solutions are marked on the graph for this purpose.

B. Solution-Pairing and Centerpoint Calculation

Once all sets of real solutions $\mathbf{a}_0$ and $\mathbf{b}$ have been obtained, we resort to the synthesis equations to pair them. By referring to eq. (1), involving $\mathbf{b}$ linearly as an unknown once $\mathbf{a}_0$ is available, it is apparent that the number of real solutions $\mathbf{a}_0$ and $\mathbf{b}$ will be identical, the pairing of solutions thus being possible. Furthermore, it is apparent that $\mathbf{b}$ could be directly computed from eq. (1) once $\mathbf{a_0}$ is known. The problem thus reduces in finding the real circlepoints A_0, the centerpoints B being available.

The problem of roundoff error propagation is addressed here. Determining $\mathbf{b}$ from cascaded computations based on $\mathbf{a}_0$ would carry the roundoff error of $\mathbf{a}_0$ into $\mathbf{b}$. To prevent roundoff error propagation, the computed $\mathbf{b}$ is sent back to the nonlinear least-square algorithm, exactly as if it were independently estimated by inspection of the plots. Consequently, complete roundoff-error consistency is insured.

VI. Linkage Performance Evaluation

The mobility of the two links coupled to the frame is to be determined in order to assess the linkage performance. Notice that in the realm of the Burmester problem, there is no specification of input and output, which means that *any* of A_0B and $A_0^*B^*$ links can be used as input.

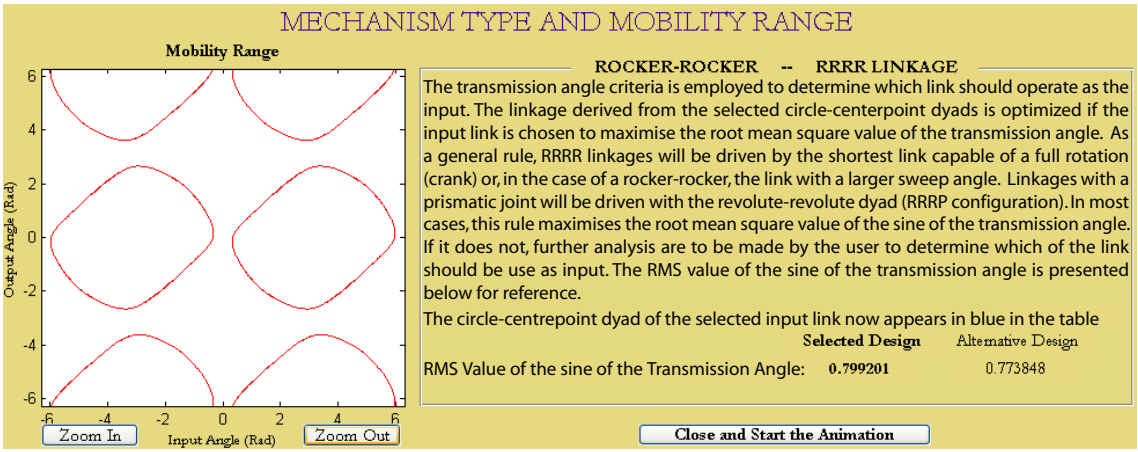

Fig. 7. Mechanism type and link-mobility range window

A. Mobility of the Input-Output Links

The input link of a four-bar linkage would better be capable of a full rotation, as four-bar linkages are frequently driven at constant angular velocity. For animation purposes, RRRR linkages are driven by the shortest link capable of a full rotation (crank); in the case of a rocker-rocker linkage, the link with a larger sweep angle is the designated input link. Linkages with a prismatic joint are driven by the revolute-revolute dyad. The link-mobility ranges of RRRR linkages are determined from the Freudenstein equation, aka the input-output (IO) equation:

$$F(\psi,\phi) = k_1 + k_2\cos(\phi) - k_3\cos(\psi) - \cos(\phi-\psi) = 0 \tag{23}$$

where

$$k_1 \equiv \frac{a_1^2 + a_2^2 - a_3^2 + a_4^2}{2a_2^2a_4^2},\quad k_2 \equiv \frac{a_1}{a_2},\quad k_3 \equiv \frac{a_1}{a_4}$$

Parameters $\{a_i\}_1^4$ are the link lengths while $\{k_i\}_1^3$ are the *Freudenstein parameters*, with links being numbered from 1 to 4 in the order: fixed, input, coupler and output.

B. Transmission Angle and Transmission Quality

A variable of merit which assesses the linkage performance is the transmission angle μ. This angle is defined as $A_0A_0^*B^*$ if BA_0 is the input link; else, as $A_0^*A_0B$. The *transmission quality*, defined as the root mean square (rms) value of the sine of the transmission angle, provides an overall performance criterion taking all possible linkage postures into consideration. We thus have the *transmission quality* Q defined as

$$Q \equiv \sqrt{\frac{1}{\Delta\psi}\int_{\psi_1}^{\psi_2}\sin^2\mu d\psi},\quad \Delta\psi \equiv \psi_2 - \psi_1 \tag{24}$$

were ψ is the input angle and $\Delta\psi$ the range of motion of the input link. An expression for $\sin\mu$ is derived in terms of the Freudenstein parameters as:

$$\sin^2\mu = \frac{k_3^2}{k_2^2 + k_3^2 + k_2^2k_3^2 - 2k_1k_2k_3}\Delta(\psi) \tag{25}$$

while the *linkage discriminant* $\Delta(\psi)$ is calculated as

$$\Delta(\psi) \equiv -k_3^2\cos^2\psi + 2(k_1k_3 - k_2)\cos\psi + (1 - k_1^2 + k_2^2) \geq 0 \tag{26}$$

The name of $\Delta(\psi)$ stems from its role in eq.(23), when this equation takes a quadratic form in $x = \tan(\phi/2)$. In this case, $\Delta(\psi)$ is the discriminant of this quadratic equation.

The transmission quality is computed after the synthesis of every RRRR linkage. One may compare the performance of all linkages solving a given problem via their transmission quality.

VII. Four-Bar Linkage Animation

A feature of the user interface is the animation function, permitting visualization of the synthesized linkage. Various design tools are offered; one may animate the linkage in a suitable frame, record an animation in an external movie file, select the starting position of the linkage, and resize the animation window. The animations are useful animation tools; a suitable[2] velocity profile is applied to a rocker input link; and the coupler-point trajectory is recorded, while animation speed is platform-independent.

A. Dyad Selection

It is well known that the five-pose synthesis of a four-bar linkage leads to none, two or four center-circlepoint dyads. Two dyads define a unique linkage, whereas four dyads lead up to six distinct four-bar linkages. One may animate any of the six linkages by selecting the desired set of

[2] A suitable velocity profile for a rocker link starts and ends with zero velocity and zero acceleration.

dyads, as shown in Fig. 8. Remember that within a center-circlepoint dyad, the centerpoint represents the base-joint position, while the circlepoint represents the coupler joint position in the reference linkage configuration. Once the selection has been confirmed and the linkage mobility determined, the dyad of the driving link is colored in blue. Furthermore, the "coordinates" of prismatic dyads are shown in boldface in the selection table, with the caveat that these putative coordinates do not represent here a point coordinate, but a unit vector pointing at the centerpoint lying at infinity.

	Circlepoint A		Centrepoint B	
	X	Y	X	Y
◉	-6.42841	-3.19637	-2.84104	-3.9963
○	-2.1987	-5.04114	-2.55467	-10.5293
◉	3.30087	-7.43968	4.94503	-10.5294
○	-4.54471e-005	-6.7876e-005	**-0.642755**	**-0.766047**

Fig. 8. Dyad selection table

B. Mechanism Frame

The *mechanism frame* is defined with origin at point B of Fig. 1, the X-axis containing B^* and directed from B to B^*, with the Y-axis pointing upwards.

Linkages may be animated in the coordinate frame of Fig. 2, in which the origin coincides with the landmark point at the reference pose. Animation in the *mechanism frame* is also allowed, in which case the whole linkage is rotated so as to have the X-axis horizontally on the display window.

C. Coupler curve

What we need here is an implicit function $F(x, y) = 0$, defined in either coordinate frame, free of linkage variables and having as parameters the link lengths. The desired function is obtained in terms of the Freudenstein parameters using computer algebra, and represents a closed, sextic curve, termed the coupler curve.

The coupler curve is computed independent from the animation. By superimposing the coupler-point trajectory, as displayed in Fig. 9 and produced during animation, with the coupler curve, we create a redundancy that increases the animation robustness. Animation problems can be readily identified this way.

D. Branch Selection

Due to the finite number of solutions of the five-pose problem, a situation known as *branch-defect* needs particular attention. Branch-defect happens if the five poses are not visited on the same branch. Without addressing directly the problem, the interface permits animation in every branch containing a pose. Branch-selection is done by graphical pose selection.

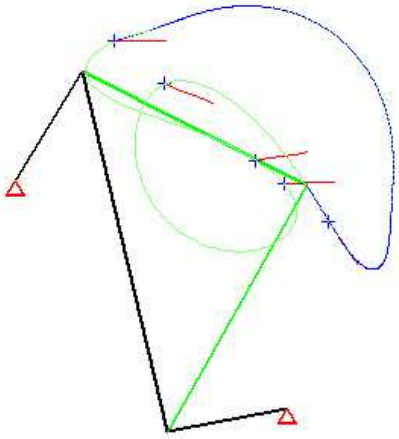

Fig. 9. RRRR linkage animation

Another branch-related problem happens if merging of branches occurs. During the animation, the direction of motion needs to be carefully determined to prevent branch-jumping. We accomplished this task by forcing the coupler-point to remain in the branch that preserves the smoothness of its acceleration.

E. Output angle

From the IO equation (23), we obtain the output angle for every input angle. A solution can be found by transforming the IO equation into an algebraic equation by means of the *tan-half* identities. This transformation leads to a quadratic equation yielding two possible solutions for each input angle. Rather than solving the foregoing quadratic equation for animation purposes, which is prone to singularities when the output angle approaches π, we used a geometric approach, free of the above singularities. These occur by virtue of the *tan-half* identities. As a matter of fact, the solutions are derived from the intersections between a line in the $u(\equiv \cos\phi) - v(\equiv \sin\phi)$ plane, stemming from the IO equation (23), and the unit circle centred at the origin of the same plane, which represents the constraints u and v. To ensure consistent frame display within the animation, the relevant output angle needs to be selected.

E.1 RRRR Linkage

As the IO equation admits two output-angle solutions, the correct output angle of RRRR linkages is determined by a proximity rule, by comparing the linkage configuration of the previous frame with the two possible configurations.

E.2 Parallelogram RRRR Linkage

In the parallelogram linkage, the input and output angles are identical. The output angle of the *kite* layout of the same linkage is the other solution of the IO equation. One may animate parallelogram linkages in their two layouts, parallelogram and kite.

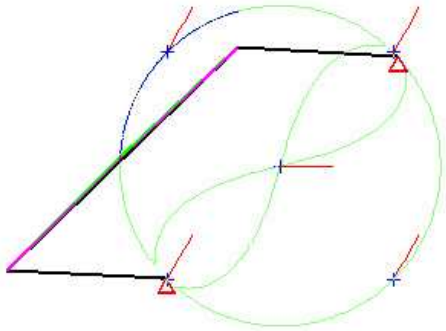

Fig. 10. Parallelogram linkage animation

E.3 RRRP Linkage

Linkages containing prismatic joints are always animated in their RRRP layout, the notion of output angle not being present here. The slider position is determined from a geometrical approach, leading again to two possible linkage configurations for each input angle. The desired configuration is determined similar to the RRRR linkage.

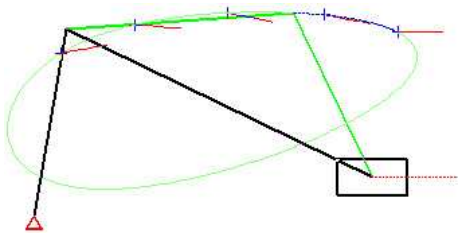

Fig. 11. RRRP linkage animation

VIII. Design Environment and Recording Tools

We included numerous functions aiming at creating a user interface useful for design. One may record the linkage animation, create an input file for reproduction of the experiment, produce a report file summarizing the synthesis and record the various graphs. Furthermore, zooming functions are available for precise analysis of all graphs and animation.

IX. Case Studies

Several case studies are available at the Centre for Intelligent Machines website at `http://www.cim.mcgill.ca/rmsl/Angeles_html/cden/module.html`. We encourage the reader to try the interface available on this same website.

X. Conclusions

We introduced a graphical user interface to solve the five-pose Burmester problem in the presence of both RRand PRdyads. The robust algorithm does not break down in the presence of special conditions of the prescribed pose. We integrated methods for the data conditioning that leads to a robust implementation of the algorithm.

XI. Acknowledgments

The work reported here is supported by the NSERC (Canada's Natural Sciences and Engineering Research Council) Design Engineering Chair at McGill University.

References

[1] L. Burmester. *Lehrbuch der Kinematik*. Arthur Felix Verlag, Leipzig, Germany, 1888.

[2] N. C. Chidambaran and A. G. Erdman. Lincages 2000 mechanism design software. In *Proc. Sixth Applied Mechanisms and Robotics Conference*, 2000.

[3] A. G. Erdman N. Yu and B. P. Byers. Lincages 2000 latest developments and case study. In *Proc. ASME DETC'2002*, 2002.

[4] R. Beyer. *Kinematische Getriebesynthese*. Springer-Verlag, Berlin, 1953.

[5] K. H. Modler. Beitrag zur theorie der burmesterschen mittelpunktkurve, teil 1. *Maschinenbautechnik*, 21(3):98–102, 1972.

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

[7] O. Bottema and B. Roth. *Theoretical Kinematics*. North-Holland Pub. Co., New York, 1979.

[8] B. Ravani and B. Roth. Motion synthesis using kinematic mappings. *ASME Journal of Mechanism, Transmissions, and Automation in Design*, 105:460–467, 1983.

[9] G. N. Sandor and A. Erdman. *Advanced Mechanism Design: Analysis and Synthesis*, volume 2. Prentice-Hall, New Jersey, 1984.

[10] K. Luck and K. H. Modler. *Getriebetechnik Analyse, Synthese, Optimierung*. Springer-Verlag, Berlin, 1995.

[11] J. M. McCarthy. *Geometric Design of Linkages*. Springer-Verlag, New York, 2000.

[12] Al-Widyan K., Angeles J., Jesus Cervantes-Sanchez J. A Numerically Robust Algorithm to Solve the Five-Pose Burmester Problem *ASME 2002 Design Engineering Technical Conferences and Computer and Information in Engineering Conference*, Montreal, Canada 2002.

[13] Angeles J., Bai S. The Determination of Planar PRDyads in Special Cases of Burmester Problem: Four and Five Poses *2005 ASME Design Engineering Technical Conferences*, Long Beach, USA 2005.

[14] Sinatra R., Angeles J. A novel approach to the teaching of planar mechanism dynamics. A case study *The International Journal of Mechanical Engineering Education* Vol. 31, No. 3, pp. 201–214, 2003