Kinematic Simulation of Planar and Spatial Mechanisms Using a Polynomial constraints Solver

Michael Barton[1], Nadav Shragai[2], Gershon Elber[3]

[1]Technion, Israel Institute of Technology, barton@cs.technion.ac.il
[2]Technion, Israel Institute of Technology, shnadav@cs.technion.ac.il
[3]Technion, Israel Institute of Technology, gershon@cs.technion.ac.il

ABSTRACT

The connection between kinematics and mechanisms to algebraic constraints is well known. This work presents a general kinematics simulator that allows end users to define planar and/or spatial arrangements, even along freeform curves and surfaces. The mechanical arrangement is then converted into a set of algebraic constraints and the motion of the arrangements is computed with the aid of a multivariate polynomial constraint solver.

.

Keywords: freeform shapes, 2D and 3D mechanisms, splines
DOI: 10.3722/cadaps.2009.xxx-yyy

1. INTRODUCTION

For many mechanical systems, the possible configuration of the mechanism may be considered as a zero set of, typically non-linear, polynomial system where its equations express geometrical constraints among components of the mechanism [13]. The motion of the mechanism is than understood as a function of only geometric relations among its parts. Searching for all feasible positions of the mechanism, known as *kinematic/mechanism synthesis*, has been of major interest in recent decades [11][19][21]. With over two thousand electro and/or mechanical samples of mechanisms [2], the vast majority of them are planar [9], of which more than half are linkages [21] and mechanisms with fixed-axes.

Since the synthesis is made possible by solving polynomial systems, two main streams of solution were followed. The first reaches the solutions algebraically, for instance by a sequential elimination of variables using resultants or by transforming original system to simpler one via, for example, Gröbner basis [6][8].

The second approach is based on numerical solvers, mostly exploiting interval analysis [18] or polynomial continuation [22][19]. Numerous work has been published in this field, among others we mention [20] which seems to be the first to employ numerical continuation for kinematic synthesis purposes.

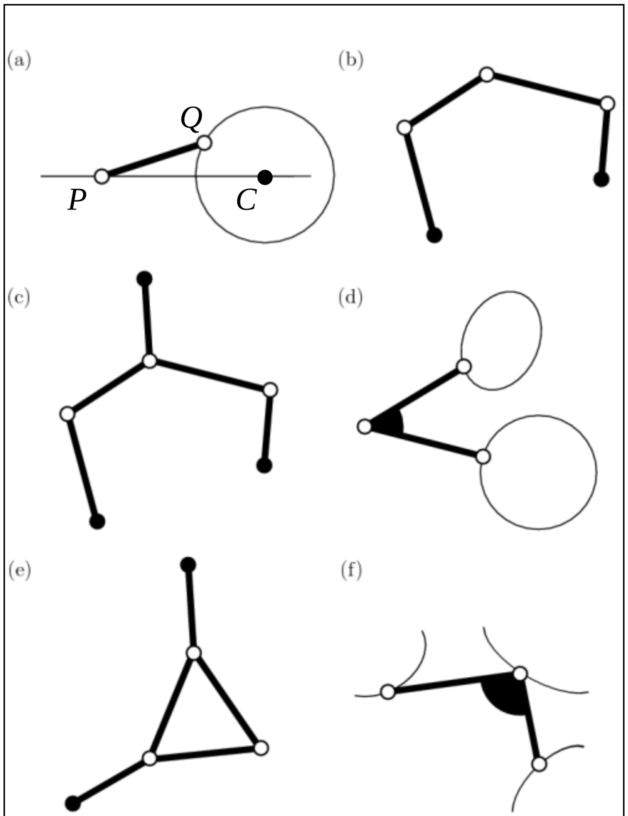

Figure 1. Examples of planar mechanisms: a) A piston: point *P* moves horizontally, *Q* moves along a circle, and their distance is preserved (black bar). b) An underconstrained mechanism: 6 unknowns -- coordinates of three movable points (white) -- in 4 bar-length preserving constraints. The solution space is a two-variate. c) Adding one more constraint makes the system "well defined". d) Mechanism is defined as points along two trajectories (circle & ellipse), two bar-length preserving constraints and one angle-preserving constraint. e) "Moving triangle": mechanism is anchored by two fixed points (black) while the triangle moves, five bar-length constraints are preserved. f) An overconstrained mechanism with 3 constraints with 3 unknowns (the parameters of the trajectory curves) – mechanism does not move - only finitely many placements could be obtained.

In general, solving a large non-linear system is computationally expensive and hence the speed-up procedures are demanded. [15] presents a decomposition technique, *called degrees of freedom analysis*, which is based on finding a sequence of actions which move the rigid body from the initial to the desired position. Geometrical constraints are solved locally instead of simultaneous solving set of equations.

Similarly to the degrees of freedom analysis, the graph-directed algebraic solvers [5][21] construct a graph whose vertices are geometric elements (typically points, lines and circles) and edges are the constraints between them. Every edge and its two end points, or a *cluster*, is a basic "decomposition" unit. The graph is then segmented into mutually independent low-degree subsystems (a union of some clusters) and those are subsequently solved. At the end of the algorithm, a cluster-merging process is required to connect them in congruent solution.

Some mechanisms perform a motion even though their movability estimations expect only finite number of its configurations. Such *overconstrained* mechanisms, whose motion typically corresponds to a zero set of a well constrained system, received special attention and treatment [19][15][4].

This work assumes well constrained mechanisms, whose motion is described by $n \times (n+1)$ (piecewise) polynomial constraints, and is presenting a 2D/3D kinematic simulation which supports motion along free form curves and surfaces.

In industry related area [7], many of the real-life mechanism's motion are computed with the aid of the ADAMS (Automatic Dynamic Analysis of Mechanical Systems) [1] commercial package. This environment simulates the motion by solving the first order Euler-Lagrange equations.

2. BACKGROUND

In this section, we present our employed tools and representations toward the definition of mechanisms:

Definition 2.1: A *kinematic mechanism* **M** = { **E**, **C** }, contains **E**, a set of *elements* from which the mechanism is built, and **C**, a set of *constraints* among them.

Under motion of the mechanism, the constraints **C** are to be preserved. That is, in every valid position of the mechanism all constraints must be satisfied. We denote such valid position by a *placement*.

We say mechanism **M** is *movable* if there exist infinitely many (continuous) *placements* of **E**. The vast majority of mechanisms yield a motion space that is a univariate. Stated differently, in most placements, the local solution space is a curve. We will denote such a mechanism a univariate-motion mechanism or UMM. In order to compute all placements of a UMM, we expect a system of n-1 constraints in n unknowns. We say that the UMM is *well defined* iff the solution of the system is a univariate or a finite set of univariates. For instance, an *underconstrained* mechanism will have two or more degrees of freedom. One can clearly note that such a count can be insufficient as a system of n-1 constraints in n unknowns could be overconstrained at some locations and underconstrained at others. While conditions for a fully constrained mechanism in the plane are known [16], this problem is, in fact, open for the spatial case. For now, we, again, assume that if the problem has n-1 constraints in n unknowns it is well defined.

In this work, we mainly focus at the simulation of UMMs, either planar or spatial. Figure 1 presents a few examples of planar mechanisms. In the rest of this section, we present the elements (in Section 2.1) of a mechanism and its constraints (in Section 2.2). Finally, in Section 2.3, we briefly discuss the subdivision based solver we employ.

2.1 The Elements of a Mechanism

The basic build block of the mechanism is a *kinematic point* which is a 2D/3D point. A point could be fixed or anchored, or it could be allowed to move along some specific trajectories. The following types of kinematic points are supported:

- **An anchored point** -- point that lies fixed and does not change its position during the motion. All black points in Figure 1 are anchored.
- **Point on a Curve** -- point that can move while its trajectory is constrained to a (not necessary planar) curve. For example point Q in Figure 1 (a) is constrained to a circular motion.
- **Point on a Surface** -- point that can move while its trajectory is constrained to be on a surface.
- **A free point** -- point that can move in any direction.

Nothing in the point's definition constraints the space the point is embedded in. The point could be in $\Re^n$ and be constrained to move along a curve or a surface in $\Re^n$. Interested in planar and spatial kinematics only, we handle points on curves or surfaces only but if higher dimensions are to be handled, points along general manifolds could be considered as well.

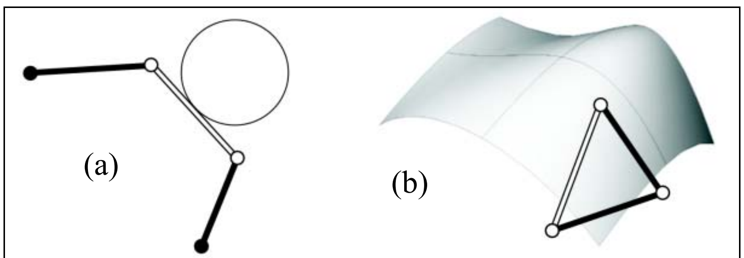

Figure 2. a) Mechanism with a flexible bar: Contrary to the outer black bars with length-preserving constraint, the middle white bar is allowed to stretch while moving; tangency constraint of the flexible bar to the circle makes the mechanism well defined. b) Part of the spatial mechanism: a triplet of kinematic points, mutually linked together with one flexible and two fixed bars, forms a kinematic face that remains tangent to the surface during the motion.

Special cases, such as a point along the X axis, or a spatial point constraint to the XY plane, are already covered by the above definitions. Yet, they could possibly be handled more efficiently if handled specifically.

Having the kinematic points' building block, the other elements of the mechanism, *kinematic bars* and *kinematic faces*, are defined as a pairs and triplets of kinematic points, respectively. The length of a bar can be specified by the user to be *fixed*, in which case an implicit distance constraint between the two points defining the bar is introduced. Figure 1 presents fixed length bars (black). In contrast, bars whose length may vary, are plotted (See Figure 2) as white bar. The kinematic face is analogously defined.

2.2 The Constraints in a Mechanism

In order to simulate a motion of a mechanism, one must build a set of constraints that bind the different elements of the mechanisms together. The following types of constraints are readily available:

1. **Distance Constraints:**
   - point--point (black bars in Figure 1),
   - point--bar,
   - bar--bar,
   - point--curve,
   - point--surface,
   - bar--curve,
   - bar--surface,
2. **Angular Constraints:**
   - bar--bar (see Figure 1 (d)),
   - bar--plane,
3. **Tangency:**
   - bar--curve (see Figure 2 (a)),
   - bar--surface,
   - face--surface (see Figure 2 (b)),
4. **Parallelism:**
   - bar--bar,

One should recall, when considering these constraints, that the *curve* or *surface* element is merely a point constraint to the curve or surface. Moreover, due to the fact that we employ geometrically oriented solver (see Section 2.3) regular (piecewise) polynomial curves and surfaces could be employed.

Working in the space of piecewise polynomial curves and surfaces, this set of constraints is piecewise polynomial as well. For example, the d distance--preserving constraint between points P and Q is expressed by:

$$\| P-Q \|^2 - d^2 \tag{2.1}$$

Similarly, an angle α angular constraint between two bars $\underline{PQ}$ and $\underline{RT}$, can be written as:

$$\frac{\langle P-Q, R-T \rangle^2}{\| P-Q \|^2 \| R-T \|^2} - \cos^2(\alpha) = 0. \tag{2.2}$$

Some constraints, such as point--curve/surface distance, are expressed by two or even three individual equations since both metric and orthogonality factors are concerned. Consider point P and surface $Q(u, v)$, (see Figure 3 (b)). Their point--surface distance constraint is expressed as

$$\begin{aligned} \|P-Q\|^2 &= d^2, \\ \left\langle \frac{\partial Q}{\partial u}, P-Q \right\rangle &= 0, \\ \left\langle \frac{\partial Q}{\partial v}, P-Q \right\rangle &= 0, \end{aligned} \tag{2.3}$$

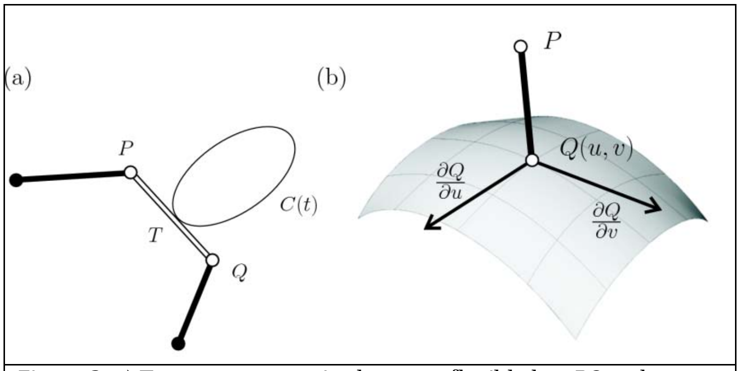

Figure 3. a) Tangency constraint between flexible bar *PQ* and curve *C*(*t*) is expressed by two equations. b) Point *P* preserves its distance from surface *Q*(*u*, *v*); this constraint is expressed by three equations (See Equations (2.3)).

where d is again the requested distance.

2.3 The Multivariate Polynomial Solver

We exploit a multivariate (piecewise) polynomial solver [3] that is capable of handling n-1(piecewise) polynomial constraints in n unknowns. The result is typically a univariate solution that is prescribing all possible placements of the UMM. Given the system of constraints

$$F(\mathrm{x}) = 0 \qquad (2.4)$$

the solver is required to solve for the simultaneous zero set of a (piecewise) polynomial system of n-1 equations with n unknowns or degrees of freedom.

The employed solver [3][12] is a subdivision based solver. That is, the univariate solution is sought in some domain $D \in \mathfrak{R}^n$, by recursively dividing D until a condition for the existence of a single univariate solution segment can be met. Then, having a cognizance of the starting and ending points of the isolated curve segment on the sub-domain's boundary, the segment is numerically traced up to a user-defined accuracy.

Two placements of the mechanism are considered *disjoint* if no path in the solution space connects them. The solution seeking approach of solver [3] ensures the topological consistency of the solution and hence one can also analyze the number of disjoint placements of the mechanism using the solver.

In the next section, we will show how a geometric formulation of a mechanism is mapped to a set of (piecewise-polynomial) constraints so solver [3] can be employed and solve for the resulting motion.

3. BUILDING THE (ALGEBRAIC) CONSTRAINTS

In this work, we only focus on piecewise-polynomial constraints that solver [3] can handle. The process of build the constraints could be divided into the following steps:

- Counting the number of degrees of freedom.
- Assigning parameters to degrees of freedoms.
- Defining the domain of the constrained problem.
- Building the constraints as piecewise-polynomial multivariates.

Every kinematic point which is a 2D or 3D, is assigned between zero and three degrees of freedom as follows (Recall Section 2.1):

- An anchored point is assigned no degrees of freedom.
- A point on a curve is assigned one degree of freedom, the parameter of the curve.
- A point on a surface is assigned two degrees of freedom, the parameters of the surface.
- A free point is assigned two degrees of freedom if planar (x and y) and three degrees of freedom if spatial (x, y and z).

Having k kinematic points in a mechanism, each point can be assigned at most three degrees of freedom. The $n \le 3k$ degrees of freedom are then assigned in sequence. Every degree of freedom that is on a curve or on a surface possesses a domain that is inherited from the curve or surface. For free points, the domain is prescribed via a bounding box of the working space as defined by the user.

Consider the example in Figure 1 (a). Let P be the first kinematic point that is assigned the first degree of freedom, t_1, to move along a horizontal line. Then, Q is assigned the second degree of freedom of the problem, t_2, to move along a circular curve. While P is independent of t_2 and Q is independent of t_1, conceptually one can make all kinematic points be functions of all degrees of freedoms. Hence, in the end of this process, $P = P(t_1, t_2)$ and $Q = Q(t_1, t_2)$. Then, the single point—point distance constraint of the problem is

$$\left\| P(t_1 - t_2) - Q(t_1 - t_2) \right\|^2 - d^2 = 0 \qquad (3.1)$$

With two degrees of freedom and one constraint, the UMM is well defined. Here is an alternative consideration of the example in Figure 1 (a). Let P be the first kinematic point that is assigned the first degree of freedom, t_1, to move along a horizontal line. Then, let Q be a free point with two additional degrees of freedom, t_2 and t_3. Q must be at a fixed distance from the fixed kinematic circle center point, C, so we now have two constraints in three unknowns:

$$\begin{aligned} &\left\| Q(t_1, t_2, t_3) - C \right\|^2 - R^2 = 0, \\ &\left\| P(t_1, t_2, t_3) - Q(t_1, t_2, t_3) \right\|^2 - d^2 = 0, \end{aligned} \tag{3.2}$$

where R is the circle's radius. Here again the (same) UMM is well defined. This alternative consideration should hint at the advantage of having kinematic points defined over curves and surfaces that not only allows precise univariate and/or bivariate motion of points (i.e. a point moving along a mechanical CAM) but also reduces the dimensionality of the problem at times.

Once the problem is fully prescribed as n-1 constraints in n unknowns, it is fed to the solver. The solution, one curve or a set of curves hinting to the existence of disjoint components, is returned as vector curve(s) in $\Re^n$. Each point on a solution curve is defining one placement of the mechanism. In the example of Equation (3.1), the vector curve is in $\Re^2$ and each point of the curve defines a (t_1, t_2) pair. This (t_1, t_2) pair is then used to further position the kinematic points P and Q (and display the entire placement). Similarly, for the example of Equation (3.2), a vector curve in $\Re^3$ is returned and each point of the curve defines a (t_1, t_2, t_3) triplet. One can either display individual placements or alternatively animate the motion of the mechanism by stepping along the solution curve(s) in small increments.

Typically, going from 2D to 3D can brings some complications and inconveniences. In the presented approach, this generalization presents virtually no difficulties and has minor impact on the size of the system (3) since the number of equations is directly related to the complexity of the mechanism, namely to the number and type of its components, and not to the dimension. The next section provides some examples, planar and spatial, of mechanisms that were simulated using the presented scheme.

4. EXAMPLES

In this section we present several examples of computed mechanisms. All examples were created using the GuIrit GUI user interface (www.cs.technion.ac.il/~gershon/GuIrit) of the Irit solid modeling system (www.cs.technion.ac.il/~irit). This Kinematic simulator was implemented as a shared library extension in GuIrit.

While the result could be animated, herein only samples (i.e. placements) of the (univariate) solution space could be shown. Hence, all figures in this section are shown as pairs with two densities of samples of placements. The dense version shows the general expected motions of the mechanism while a second figure, with only a few placements, allows one to follow each placement precisely. As an example, Figure 4 shows a simple planar mechanism while the points are moving along piecewise polynomial curves.

Clearly the strength of this approach is seen when considering kinematic points over freeform curves and surface in 3D. Figures 5-7 shows a few such a examples.

All presented data were tested on PC with an Intel(R) Pentium(R) CPU (2.8GHz), 1 GB of RAM. Time values might differ depending on the topological complexity of each specific example and on a numerical precision which is required in the stage where the solver is called. Requiring smooth simulations with the “motion” step no longer than 0.01. In the most demanding 3D examples it took about a minute to compute.

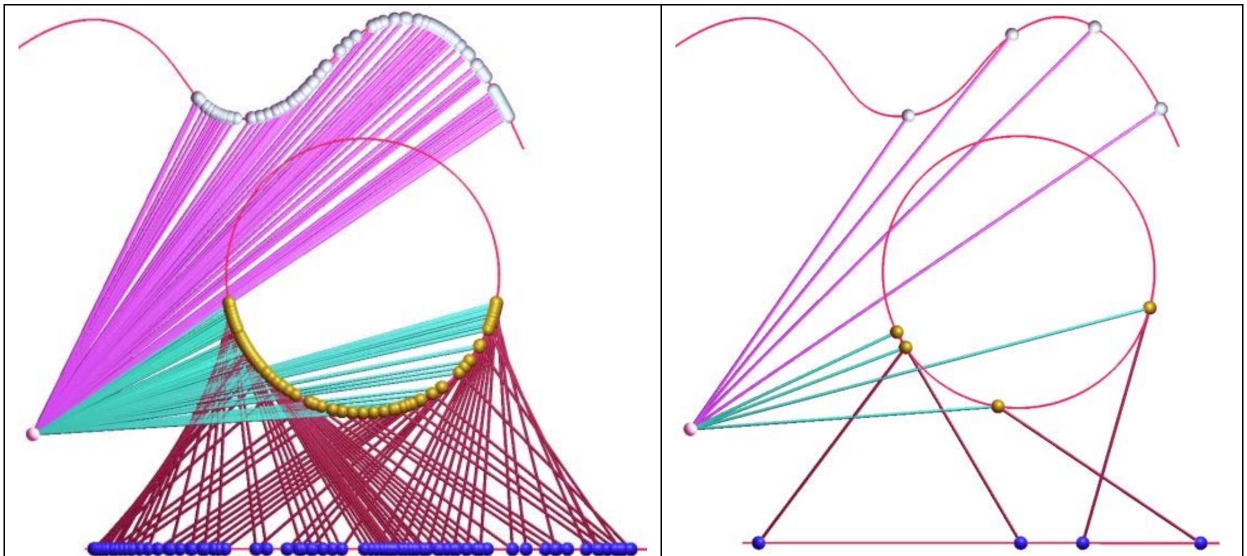

Figure 4. Planar motion converter: Mechanism consists of four points and three consecutive bars. Blue (bottom) point is moving along a horizontal line, the yellow point is constrained to a circle, the pink point is fixed and the top white point is on a B-spline curve. The bottom magenta bar is fixed while the other two are flexible, while preserving an angle of 30 degrees between them. Four different placements are shown on the right.

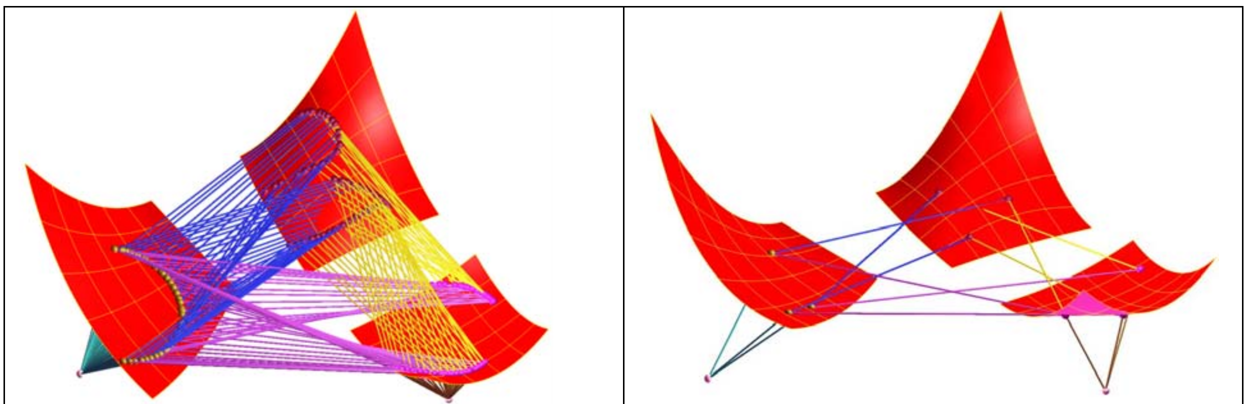

Figure 5. A spatial linkage is defined by two fixed points (at the bottom) and three along-a-surface-movable points. Corresponding polynomial system consist of five constraints (lengths of links) and six unknowns (surface's parameter). Three different placements are shown on the right.

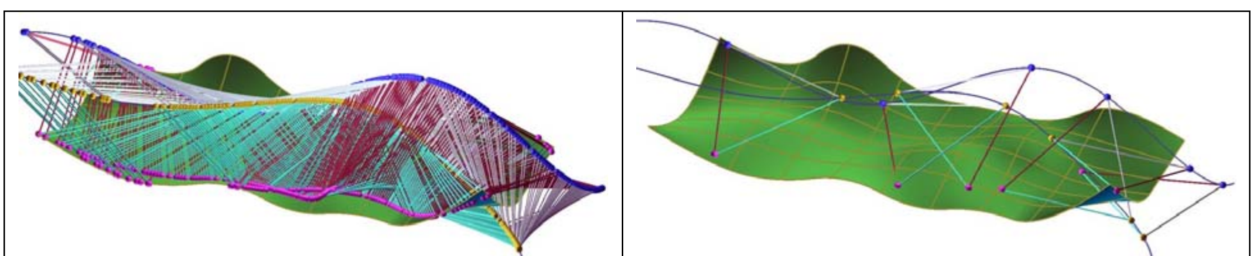

Figure 6. Two vertices (in blue and yellow) of a rigid triangle are constrained to move along the two (blue) curve trajectories. The third vertex must follow the bottom surface in green. Six different placements are shown on the right.

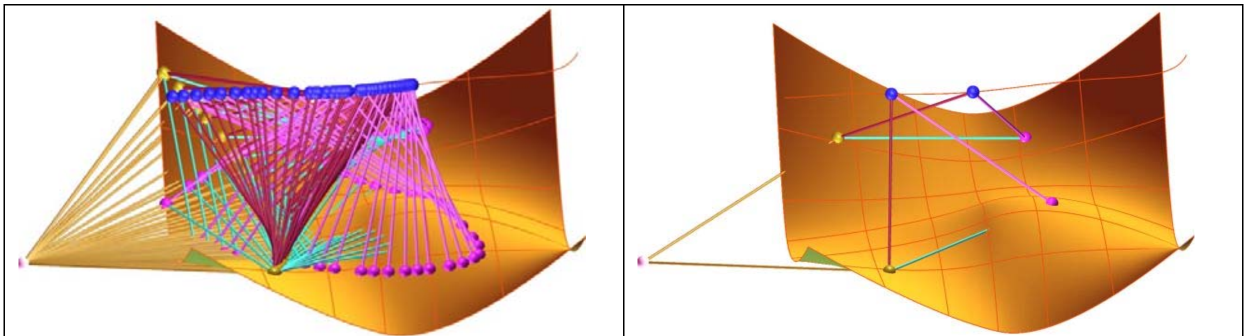

Figure 7. Two points (in magenta and yellow) are allowed to move along the same B-spline surface while the blue top point is constrained to move the top curve, in a rigid triangle. In addition, the white point on the left is fixed and a fixed bar connects it to the yellow point. Two different placements are shown on the right.

5. CONCLUSIONS AND FUTURE WORK

In this work, we have presented an application of simulated motion of planar and spatial mechanisms using a subdivision based solver. The fact that this solver is geometrically oriented makes it well suited to handle geometric constraints. Hence, the ability to handle freeform motion along a freeform curve and/or surface in the plane or in space provides this simulator with unique capabilities.

In [5], a scheme was presented to decompose a large constrained problem into numerous small problems. Because, in principle, the complexity of the subdivision based solvers grows exponentially with the dimension, such decomposition might be highly beneficial. Similarly, in [10] a scheme that represents the constraints as expression trees is introduced that shows only polynomial grows with the dimension of the problem. The use of expression trees here could be beneficial as well.

As an additional future work, any algebraic constraint may be formulated as a function of time, introducing time as an additional parameter, which would allow, for instance, the mechanism to change dynamically its shape during its motion.

Acknowledgement

This research was partially supported in part at the Technion by a fellowship from the Lady Davis Foundation and in part by the Israel Science Foundation (grant No.346/07).

6. REFERENCES

[1] The ADAMS system, http://www.mscsoftware.com/products/adams.cfm

[2] Artobolevsky I.: Mechanisms in Modern Engineering Design, Vols 1-4, MIR Publishers, Moscow, 1979.

[3] Barton M; Hanniel I., Elber G.: Topologically guaranteed solution of surface--surface intersection via no--loop and single--component tests, Submitted for publication in Solid and Physical Modeling 2009.

[4] Beddows M.; Mavroidis C.; A spatial overconstrained mechanism that can be used in practical applications, Proceedings of the 5th Applied Mechanisms and Robotics Conference, Cincinnati, OH, October 1997

[5] Bouma W.; Fudos I.; Hoffmann C. M.; Cai J.; Paige R.: Geometric constraint solver, Computer-Aided Design, 27(6), 1995, 487-501.

[6] Buchberger B.; Collins G.; Kutzler B.: Algebraic methods for geometric reasoning, Annual Reviews in Computer Science, 3, 1988, 85-120.

[7] Car A. C.; Esaki M. T.: Virtual prototyping of a wheelchair lift for the disabled fitted to a CL-600 Chalenger Aircraft, International ADAMS User Conference, 1998.

[8] Cox D. A.; Little J. B.; O'Shea D.: Using algebraic geometry, Springer 2005.

[9] Dar T.; Joskowicz L.; Rivlin E.: Understanding mechanical motion: From images to behaviors, Artificial Intelligence, 1999.

[10] Elber G.; Grandine T.: Efficient Solution to Systems of Multivariate Polynomials using Expression Trees. SMI'08, Stony Brooks, NY, June 2008, 163-169.

[11] Erdman A. G.; Sandor G. N.: Mechanism Design: Analysis and Synthesis, Prenctic-Hall International Editions, 1991.

[12] Hanniel I.; Elber G.: Subdivision Termination Criteria in Subdivision Multivariate Solvers, Computer Aided Design, 39, 2007, 369-378.

[13] Hopcroft J. E.; Joseph D. A.; Whitesides S. H.: Movement Problems for 2-Dimensional Linkages, SIAM J. on Computing, 13(3), 1984, 610-629.

[14] Karger A.; Husty M.: Self-motions of Griffis-Duffy type parallel manipulators, Proceedings 2000 IEEE International Conference Robotics and Automation, San Francisco, CA, 2000, 24-28.

[15] Kramer G. A.: Solving geometric constraint systems, In Proceedings of the National Conference on Artificial Intelligence, AAAI, 1990, 708-714.

[16] Laman G.: On Graphs and Rigidity of Plane Skeletal Structures, Journal of Engineering Mathematics, 4(4), October 1970.

[17] Latombe J. C.: Robot Motion Planning, Kluwer Academic Publishers, 1991.

[18] Lee E.; Mavroidis C.; J. P. Merlet: Five precision points synthesis of spatial RRR manipulators using interval analysis, ASME J. of Mechanical Design, 126(5), 2004, 842-849.

[19] Morgan A. P.; Wampler C. W.; Sommese A. J.: Numerical continuation methods for solving polynomial systems arising in kinematics, Transactions of the ASME, J. of Mechanical Design, 112, 2004, 59-68.

[20] Roth B.; Freudenstein F.: Synthesis of path generating mechanisms by numerical methods, Transactions of the ASME, J. of Engineering for Industry, 85B, 1963, 298-306.

[21] Shrobe H. E.: Understanding linkages, National Conference on Artificial Intelligence, 1993.

[22] Sommese A. J.; Verschelde J.; Wampler C. W.: Advances in polynomial continuation for solving problems in kinematics, ASME J. of Mechanical Design, 126(2), 2004, 262-268.