

**DETC2004-57524**

## **SYNTHETICA 2.0: SOFTWARE FOR THE SYNTHESIS OF CONSTRAINED SERIAL CHAINS**

**Alba Perez, Hai-Jun Su, J. Michael McCarthy\***

Robotics and Automation Laboratory  
Department of Mechanical and Aerospace Engineering  
University of California  
Irvine, California 92697

Email: maperez@uci.edu, suh@eng.uci.edu, jmmccart@uci.edu

### **ABSTRACT**

*A spatial linkage is defined by a workpiece supported by one or more serial chains. We say that the spatial linkage can perform a given task when this task belongs to its workspace. A task can be specified as a discrete set of positions or as a set of curves or surfaces. Spatial serial chains can be synthesized to perform specified tasks, and combined to form more complicated linkages.*

*Recently we developed Synthetica 1.0, a Java-based architecture for the computer-aided-design of spatial linkages that contains task definition and visualization, and linkage visualization, analysis and synthesis modules.*

*In this paper, we present the new version, Synthetica 2.0. Among other additions, the new version includes task interpolation, a selection matrix, a generic synthesizer and several specialized synthesis procedures for constrained spatial linkages.*

### **1 INTRODUCTION**

The focus of commercial computer-aided design (CAD) software has been on tools for shape representation and manipulation, assembly and detail drawing creation, and integration with computer-aided manufacturing (CAM) and analysis packages. This focus is clearly represented in high-end systems such as Dassaults CATIA, and PTCs Pro/ENGINEER. Mid-range modeling applications such as SolidWorks, AutoCAD, IronCad, and

Vellum Solids have export capabilities or additional plugin interfaces to CAM packages such as SurfCam or MasterCam, FEA packages such as Algor or Nastran, and dynamic simulation packages such as WorkingModel or ADAMS.

While this software is extremely powerful and versatile, it does not help automate the step in the design process described as the function-to-form transformation. This is the creative phase in the product development cycle where the designer synthesizes the geometric model of individual parts and assemblies to realize the particular function and satisfy the particular requirements.

One specific area of design research that has well developed tools for this function-to-form transformation is the theory of mechanism synthesis. Mechanism synthesis theory can be divided into methodologies for type and number synthesis, and dimensional synthesis, see Sandor and Erdman, 1984 [1]. Type and number synthesis allow the designer to match the input requirements to a space of possible mechanism types. Once a mechanism topology is chosen, the designer can use the tools of dimensional synthesis to identify the values of geometric parameters that satisfy the motion requirements.

#### **1.1 Linkage Design Software**

In the dimensional synthesis problem, the input specification, or task, describes the desired workspace or motion capabilities, and is typically represented by a set of discrete positions. From this description and for the desired topology, design equations are created and solved to determine the physical dimensions of the

---

\*Address all correspondence to this author.

mechanism. Solution algorithms have been presented for a variety of mechanism architectures and input conditions (see for example Suh and Radcliffe, 1978 [2] and McCarthy, 2000 [3]). Recent developments in the area of kinematic synthesis (Mavroidis et al., 2001 [4], Perez and McCarthy, 2003 [5]) define systematic methodologies for the synthesis of constrained mechanisms. These provide ways to obtain the design equations for any serial chain topology.

Research software has been developed to aid in the design of planar (Erdman and Gustafson, 1977 [6], Waldron and Song, 1981 [7]), spherical (Ruth and McCarthy, 1997 [8]), and spatial (Kihonge, Vance and Larochelle, 2001 [9]) linkages. Two commercial packages for planar linkage design are also available, SyMech software (2002) and WATT software (2002).

Synthetica 1.0 was developed and presented in 2002 as a software architecture that can incorporate a variety of planar and spatial mechanism synthesis routines into an interactive Java-based design system. It provided a graphical user interface to integrate features common to linkage design systems. One of its main highlights was the possibility of incorporating user-defined synthesis routines. As examples, synthesis routines for the RR and the TS chains were incorporated into the program.

## 1.2 Paper Overview

In this paper, we present the new version of the software, Synthetica 2.0, which incorporates new functions based on developments in the kinematic synthesis theory and improvements in the user tools. The task specification module incorporates in this new version interpolation capabilities in order to create trajectories from a set of discrete task positions. In the synthesis module, a selection matrix and a generic synthesizer have been added to the software, based on the dual quaternion kinematic synthesis theory.

In addition to the generic synthesizer, the new version provides specialized synthesis algorithms for 17 different topologies. These algorithms have been created and written by different researchers to be incorporated into Synthetica 2.0 independently, according to its extensibility properties. They provide quick, algebraic formulations that calculate the complete set of multiple solutions for the specific problem.

In section 2 we present the design process and the structure of the software. In section 3, we discuss in detail the theory and implementation of the new features added in the new version, Synthetica 2.0. Finally, future developments and conclusions are stated.

## 2 SOFTWARE DESCRIPTION

Synthetica is a software for the synthesis of serial chain mechanisms and the specification, analysis and animation of both serial and parallel chains. Synthetica can run as a standalone ap-

plication/applet, but can also be used as a programming package, in which interested users can define their own mechanisms and implement their own synthesis or kinematics algorithms.

The design methodology follows a procedure for dimensional synthesis with a finite number of rigid body positions, in which the designer specifies a set of positions as the desired task. Then he or she is presented with the selection matrix, that shows the topologies that can be synthesized for the given task, together with the number of extra constraints to define for each case. The designer can select from there the particular type or types of linkages to design. The selection matrix presents the open-chain candidates, which we call linkage primitives. The solutions can be combined later to create parallel mechanisms; their primitives can have the same topology, or different. If they are designed to reach the same set of positions, the combined closed chain is guaranteed to, at least, be assembled at those positions.

Once the primitives are selected, specialized synthesis routines are used if they exist; otherwise, the generic synthesizer is used. The specialized synthesis routines provide with all possible solutions, while in the generic synthesizer, the designer must specify the desired number of solutions, that will be found running the synthesizer several times.

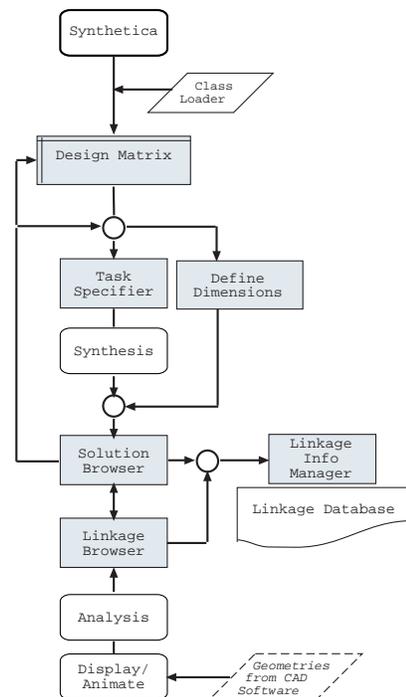


Figure 1. Flowchart for Synthetica. The gray boxes denote user-interface routines.

## 2.1 Structure of Synthetica

The key components and flow of information for Synthetica are shown in Figure 1; for more detailed information, we refer to Collins et al., 2002 [10] and Su et al., 2002 [11].

The Java-based software is designed to run under Windows, Mac OS, and Linux operating systems.

The Task Specier module allows the user to input the required position information, and any additional constraints required by the synthesis method, see Figure 2.

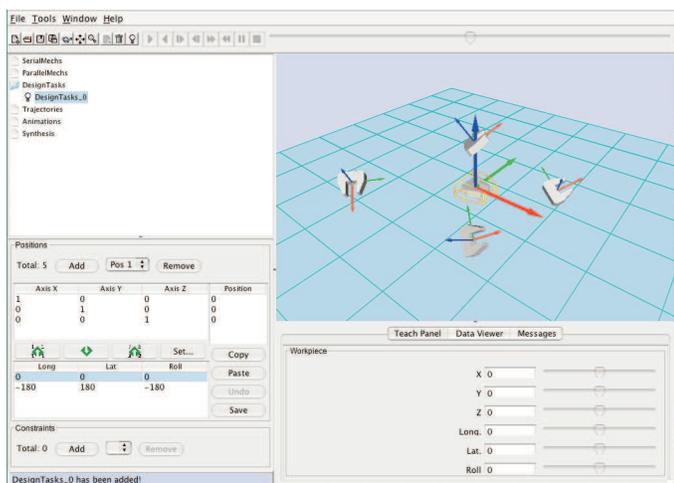


Figure 2. Task dialog box and task visualization

Once this information is provided, the designer accesses the Decision Matrix. This is a tabular representation of the mechanism topologies and synthesis routines available for the specified task, dynamically constructed from information about the available classes and the task. The user can choose as many available topologies as desired in order to perform the synthesis.

For each of the chosen candidates, the synthesis method is called, and a list of solutions is generated and made available in the Solution Browser. A 3D model of the solution linkages is generated, displayed, and animated to perform the task.

## 2.2 Synthetica Packages

Synthetica has four major components organized into Java packages. The basic packages were presented with Synthetica 1.0 [10; 11] and will be only briefly described here.

The main components are *GUI Modules*, that defines the interfaces that allow users to modify/view the data, including the Design Task Dialog and the Linkage Specifier Dialog; *glprimitives*, that provides the 3D graphics engine (see the mechanism window in Figure 3); *kinemath*, providing basic mathematical resources used by other parts of the program; and *mechanism*, containing the data classes for extending new mechanisms. A mech-

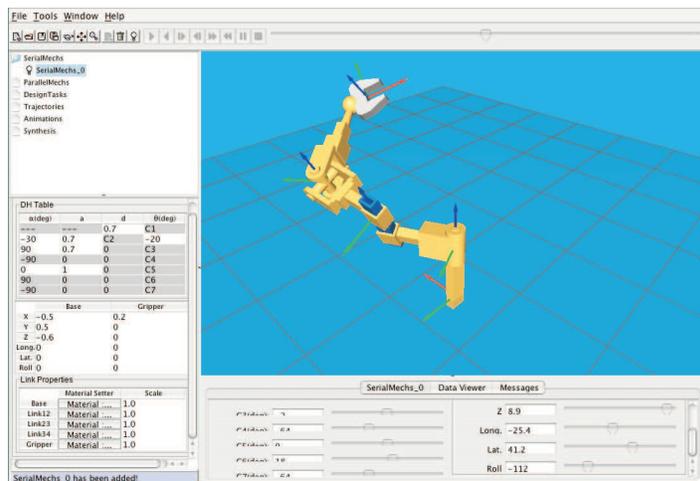


Figure 3. Mechanism Viewer Window showing an RPTS serial chain

anism is dened by serial chains composed of links and joints. A design tasks is dened by a set of discrete positions and a list of additional constraints.

## 3 NEW FEATURES OF SYNTHETICA 2.0

In this section, we present the theory and implementation for the new features introduced in Synthetica 2.0. The addition of a generic synthesizer has been possible thanks to the development of systematic synthesis procedures. In our approach, we use dual quaternion kinematics equations to create design equations. The same theory provides with formulas for counting the maximum number of task positions for each topology, formulas that are implemented in the decision matrix. In addition, we have incorporated specialized synthesis results that provide faster algorithms and more complete solutions. In the area of task definition and visualization, improved interpolation and visualization capabilities have been added.

### 3.1 Generic Synthesizer

The generic synthesizer is intended as a default design procedure in which the synthesis equations of a constrained chain are formulated and solved numerically to obtain one solution.

To formulate design equations, we write the kinematics equations of a generic chain composed of  $m$  cylindrical joints,

$${}^K E_m = \hat{S}_1(\theta_1, d_1) \hat{S}_2(\theta_2, d_2) \dots \hat{S}_m(\theta_m, d_m), \quad (1)$$

where each of the displacements of the individual axes,  $\hat{S}_i$ , are

$$\hat{S}_i(\theta_i, d_i) = \begin{Bmatrix} \sin \frac{\theta_i}{2} \mathbf{s}_i \\ \cos \frac{\theta_i}{2} \end{Bmatrix} + \varepsilon \begin{Bmatrix} \sin \frac{\theta_i}{2} \mathbf{s}_i^0 + \frac{d_i}{2} \cos \frac{\theta_i}{2} \mathbf{s}_i \\ -\frac{d_i}{2} \sin \frac{\theta_i}{2} \end{Bmatrix}, \quad (2)$$

where  $S_i = \mathbf{s}_i + \varepsilon \mathbf{s}_i^0$  are the Plücker coordinates of the joint axis,  $\theta_i$  is the rotation about the axis (for a revolute joint) and  $d_i$  is the translation along the axis (for a prismatic joint). All of these are unknowns for the design problem.

To synthesize a generic robot  $C_m$  composed of  $m$  cylindrical joints, the set of transformations defining the task are also expressed as dual quaternions relative to the first, or reference, position. We denote by  $\hat{P}^j$  the dual quaternion expression of the relative task position  $[M_{1,j}] = [M_j][M_1]^{-1}$ . To create the design equations, we equate  $\hat{K}E_m$  to each of the positions,

$$\hat{K}E_m^j = \hat{P}^j, \quad j = 2, \dots, n. \quad (3)$$

For each task position we can define eight equations, six of which are independent.

The generic  $C_m$  robot can be specialized to any robot with that same number of joints by setting to zero some of the joint variables. For the transformations defined by an axis  $\hat{S}_1(\theta_1, d_1)$ , we would use  $\hat{S}_1(\theta_1, 0)$  to represent a revolute (R) joint and  $\hat{S}_1(0, d_1)$  to represent a prismatic (P) joint. A universal (T) joint can be created as  $\hat{S}_1(\theta_1, 0)\hat{S}_2(\theta_2, 0)$  with the extra constraints  $S_1 \cdot S_2 = (0, 0)$ , and a spherical joint can be created similarly, taking into account that the directions of the axes can be set arbitrarily, see Perez, 2003 [12].

In order to avoid the computational charge of setting to zero the appropriate terms every time, the general kinematics equations  $KE_m$  are collected as coefficients of the joint variables and set in matrix form. When a certain topology is specified, the kinematics equations are assembled as sums of the terms that are nonzero multiplied by the products of joint variables.

These generic kinematics equations are used both for compute the inverse kinematics –in which case the parameters of the axes are known and the equations can be solved linearly for the joint variables– and as the synthesis design equations.

To compute the design parameters, the design equations are solved numerically for both the joint parameters and the joint variables, using a numerical algorithm based on the implementation of the Levenberg-Marquardt method for Minpack. This solver returns only one solution, while usually the designer is interested in obtaining several solutions that could be ranked using different performance criteria. In order to offer this possibility to the designer, we are planning to implement a dialog box in which the designer will be able to input the desired number of solutions, and the solver will run accordingly as many times as needed.

### 3.2 Selection Matrix

The selection matrix is based on counting formulas [5] that compute the maximum number of arbitrary positions that a given topology can reach. These formulas are based on counting the number of equations and unknowns for each case.

For doing this, we distinguish axes  $S_i$  that define revolute joints from those that define prismatic joints. The axis of a revolute joint is defined by four independent parameters in the associated Plücker coordinate vector: six elements, three for the direction and three for the moment, plus two constraints, the unit direction vector and the moment being perpendicular to the direction. A prismatic joint depends on two parameters defining the direction of the slide: three coordinates plus the unit vector constraint.

Let  $r$  and  $t$  be the number of revolute and prismatic joints in the chain, where  $m = r + t$ , then for  $n$  task positions, we have the total number of design parameters  $4r + 2t + (n - 1)r + (n - 1)t$ .

Only six of the eight components of a dual quaternion are independent, therefore we have  $6(n - 1)$  equations for each dual quaternion, plus any number of additional constraints  $c$  that we may want to add. We can equate the number of equations and the number of unknowns to obtain

$$n_{\max} = \frac{3r + t + 6 - c}{6 - r - t}, \quad r + t \leq 5, \quad (4)$$

which defines the maximum number of arbitrary task positions for a given topology. Doing a similar computation for the number of orientations, we obtain that

$$n_R = \frac{3 + r - c}{3 - r}, \quad r \leq 2. \quad (5)$$

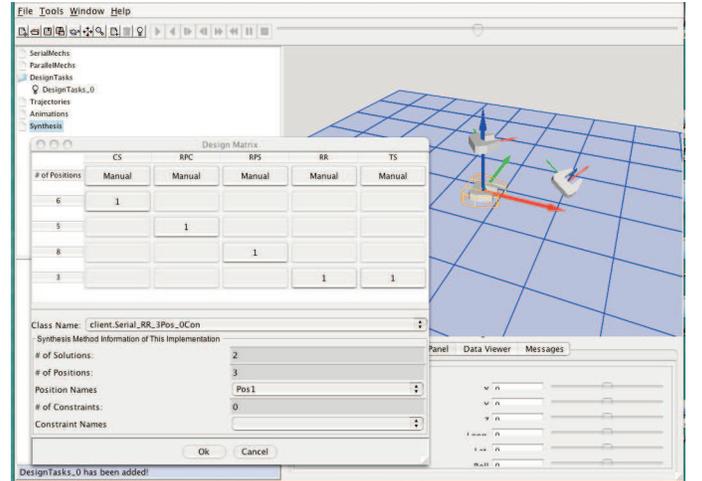


Figure 4. The selection matrix for some of the specialized synthesis procedures

For constrained serial chains in which  $n_R < n_{\max}$ , we can solve the orientation equations independently, in which case we

can add some constraints to the chain or add some extra translations to the task, the total number of translations being

$$n_T = \frac{2r+t+3-c}{3-t}, \quad t \leq 2. \quad (6)$$

If  $n_R < n_{max}$ , we can solve the chain for  $n_R$  complete task positions plus  $e = n_T - n_R$  additional task positions with arbitrary translational terms and rotational terms within the workspace of the chain, see [5] for further discussion on the counting formulas.

For a design task consisting of  $n$  positions, the selection matrix highlights all those topologies that admit  $m$  positions, for all  $m \geq n$ . For those cases in which  $m = n$ , the chains can be designed to obtain a finite number of positions, while for those topologies with  $m > n$ , additional structural constraints must be defined for the chains. The selection matrix allows to associate constraints with the topologies and allows the designer to select as many of the highlighted topologies as desired to be synthesized.

### 3.3 Specialized Synthesis Routines

We have added a number of specific synthesis routines that apply recent results in the field. The main reason to incorporate these chain-specific synthesis methods is the existence of algebraic synthesis methods that are fast to compute, reliable, and that output all the solutions.

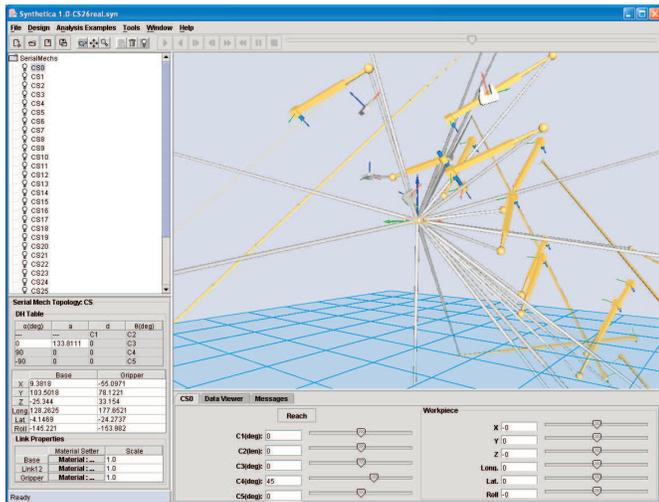


Figure 5. Twenty-six real solutions for the CS robot

Efficient synthesis methods for several chains with a spherical joint in the wrist were developed in Su, Wampler and McCarthy, 2004 [13]. Of these, we have implemented the cases for which solutions can be found quickly. This means either

closed algebraic formulas or resultant methods leading to univariate polynomials.

In particular, Synthetica 2.0 has optimized routines to solve for the TS, CS and RPS chains, see Figure 5. The TS algebraic synthesis method of Innocenti, 1995 [14] shows that there are up to 20 real solutions for a maximum of 7 positions; in addition, a specialized synthesis procedure with 3 task positions has been implemented in order to synthesize RRSS closed chains.

For the CS, the 6-position synthesis has been implemented, which uses resultant methods to yield a maximum of 26 solutions. The 8-position synthesis of RPS chains is solved by using matrix eigenvalue elimination [13] to yield up to 64 solutions.

More complicated cases, such as the RRS chain, need polynomial continuation and parallel computing in order to find all the solutions, and have not been implemented in Synthetica yet.

The results of the dual quaternion kinematic synthesis developed in [5] have also been implemented in Synthetica 2.0. The basic equations are like those presented in the previous section, but for these specific cases, a method was used to eliminate the joint variables and obtain algebraic equations with only the parameters of the joint axes, that could be further simplified to the solution of univariate polynomials via resultant methods. The following optimized routines are available: PPR, RC, TP, RRP, CC, and RPC, and chains obtained by permutation of their joints.

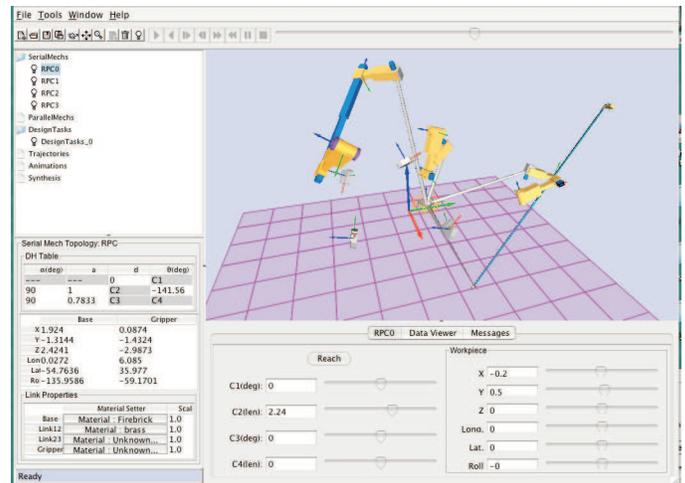


Figure 6. Four real solutions for the RPC robot

In summary, Synthetica 2.0 includes so far 17 specialized synthesis routines.

### 3.4 Interpolation

The task positions defined by the designer can be interpolated when expressed as dual quaternions by generalizing known

techniques for Bezier interpolation of quaternions. Ge and Ravani, 1994 [15] generalized results by Shoemake, 1985 [16] to spatial displacements using double quaternions. These results were refined to ensure smooth transitions at each key frame by Ge and Kang, 1995 [17].

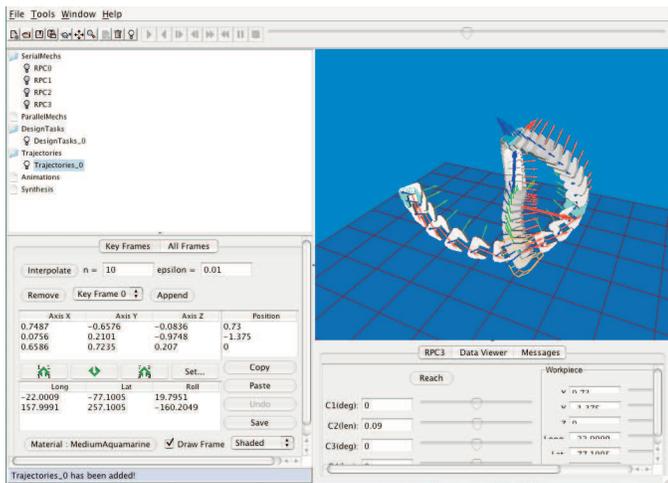


Figure 7. Interpolation with five key positions, shown in light blue

In Synthetic 2.0, the user specifies the key positions and the number of intermediate frames between key positions to obtain an interpolated trajectory as the one shown in Figure 7.

### 3.5 Animation

Synthetic 2.0 allows the designer to animate the linkages to closely follow a given trajectory. In the case of constrained chains, it may happen that the trajectory does not lie on the workspace of the chain; in this case, the movement of the chain will approximate the trajectory.

The trajectory can be created as an interpolation of the task positions. For each of the individual frames forming the trajectory, the program computes the values of the joint angles that locate the robot in the closest position to the given frame, by performing a minimization for the inverse kinematics.

## 4 CONCLUSIONS

This paper presents Synthetic 2.0, the new version of the Java-based software for the design, analysis and visualization of spatial mechanisms. Synthetic 1.0 contained modules for the construction and visualization of serial and parallel chains, and the possibility of accepting external synthesis routines; two of them were provided as examples.

The new release incorporates a generic synthesizer to be used as default when no user-supplied synthesis methods are

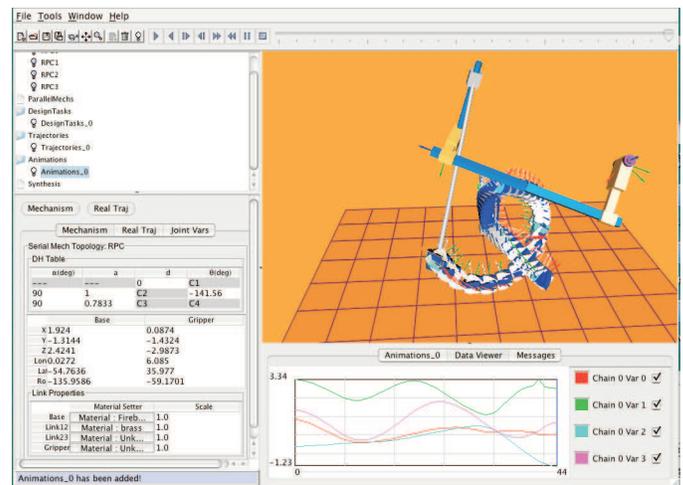


Figure 8. Animation of an RPC chain to follow a trajectory. The desired trajectory is represented by the white frames and the real trajectory of the linkage is given by the blue frames

available. This aims to generate the synthesis equations from a general robot that is tailored to the choice of the designer. A decision matrix informs the designer with the candidate topologies for a specified task, from which the primitives can be chosen. In addition, Synthetic 2.0 incorporates the latest advances in synthesis results for a series of constrained chains that add up to 17. This release contains also new and improved visualization tools for both the definition of the task and the display of the chains.

## ACKNOWLEDGMENT

The authors gratefully acknowledge the support of the National Science Foundation.

## REFERENCES

- Sandor, G. N., and Erdman, A. G., 1984, *Advanced Mechanism Design: Analysis and Synthesis, Vol. 2*. Prentice-Hall, Englewood Cliffs, NJ.
- Suh, C.H. and Radcliffe, C.W., 1978, *Kinematics and Mechanisms Design*, John Wiley & Sons, New York.
- McCarthy, J.M., 2000, *Geometric Design of Linkages*, Springer-Verlag, New York.
- Mavroidis, C., Lee, E., and Alam, M., 2001, "A New Polynomial Solution to the Geometric Design Problem of Spatial RR Robot Manipulators Using the Denavit-Hartenberg Parameters," *ASME J. Mechanical Design*, 123(1):58-67.
- Perez, A. and McCarthy, J.M. "Dual Quaternion Synthesis of Constrained Robotic Systems", accepted for publication at the *Journal of Mechanical Design*, August 2003.
- Erdman, A., and Gustafson, J., 1977, "LINCAGES Linkages

Interactive Computer Analysis and Graphically Enhanced Synthesis Packages,” Technical Report 77-DET-5, ASME.

Waldron, K. J., and Song, S. M., 1981, “Theoretical and Numerical Improvements to an Interactive Linkage Design Program, RECSYN,” *Proc. of the Seventh Applied Mechanisms*, Kansas City, Missouri.

Ruth, D. A., and McCarthy, J. M., 1997, “SPHINXPC: An Implementation of Four Position Synthesis For Planar And Spherical 4R Linkages,” *Proc. 1997 ASEM Design Engineering Technical Conferences: Design and Automation Conference*, Sacramento, California.

Kihonge, J.N., Vance, J. M., and Larochelle, P. M., 2002, “Spatial Mechanism Design in Virtual Reality with Networking,” *ASME J. Mechanical Design*, **124**(3):435-440.

Collins, C. L., McCarthy, J. M., Perez, A. and Su, H., “The Structure of an Extensible Java Applet for Spatial Linkage Synthesis,” *ASME Journal of Computing and Information Science and Engineering*, 2(1):45-49, March 2002.

Su, H., Collins, C.L., and McCarthy, J.M., “An Extensible Java Applet for Spatial Linkage Synthesis”, *ASME Design Engineering Technical Conferences*, Sept.29-Oct.2, 2002, Montreal, Quebec, Canada.

Perez, A., “Dual Quaternion Synthesis of Constrained Robotic Systems”, *PhD Dissertation*, September 2003.

Su, H., Wampler, C. W., and McCarthy, J. M., 2004, “Geometric Design of Cylindric PRS Serial Chains,” *ASME Journal of Mechanical Design*, 126(1):269-277.

Innocenti, C., 1995, “Polynomial Solution of the Spatial Burmester Problem”, *ASME J. Mech. Design* 117(1).

Ge, Q. J., and Ravani, B., “Geometric Construction of Bézier Motions,” *ASME Journal of Mechanical Design*, 116:749-755, 1994.

Shoemake, K., “Animating Rotation with Quaternion Curves,” *ACM Siggraph*, 19(3):245-254, 1985.

Ge, Q. J., and Kang, D., “Motion Interpolation With  $G^2$  Composite Bezier Motions,” *ASME Journal of Mechanical Design*, 117:520-525, 1995.