Kinematic Synthesis Using Clifford Algebras: Theory and Applications

Alba Perez-Gracia
Department of Mechanical Engineering
College of Engineering
Idaho State University
Pocatello, Idaho

Department of Mechanical Engineering

University of Utah

October 30, 2007

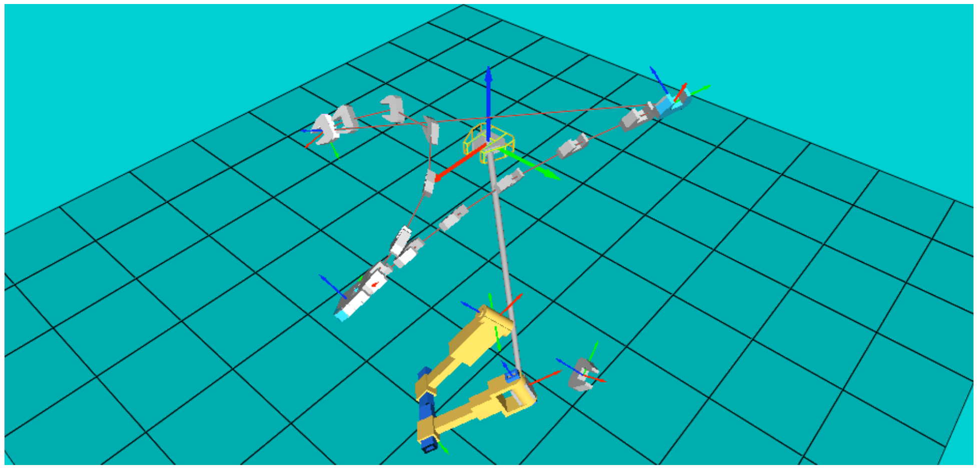
Kinematic Synthesis

Kinematic Synthesis :

- *Determine the mechanical constraints (i.e., links and joints) that provide a desired movement.*
- *It solves the function-to-form problem when dealing with motion.*

Finite-position Kinematic Synthesis:

- Identify a **set of task positions** that represent the desired movement of the workpiece.
- Developed for synthesis of **constrained serial open chains**. It can also be applied to parallel robots.

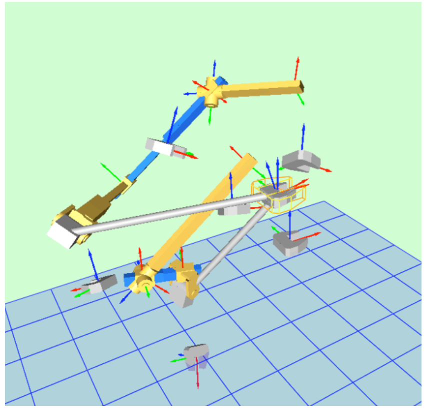


Constrained Spatial Robots

Constrained robotic system: A workpiece, or end-effector, supported by one or more serial chains such that **each one imposes at least one constraint** on its movement.

DOF	*Structure*
2	RP, RR
3	RPP, CP, RRP, RC, TP, RRR, TR
4	RPPP, CPP, RRPP, RCP, CC, TPP, RRRP, RRC, TRP, TC, SP, RRRR, TRR, TT, SR
5	RPPPP, CPPP, RRPPP, CRPP, CCP, TPPP, RRRPP, CRRP , CCR, TRPP, TCP, SPP, RRRRP , CRRR , TRRP, TTP, TCR, SRP, SC, RRRRR, TRRR, TTR, SRR, ST

Classification of constrained serial robots

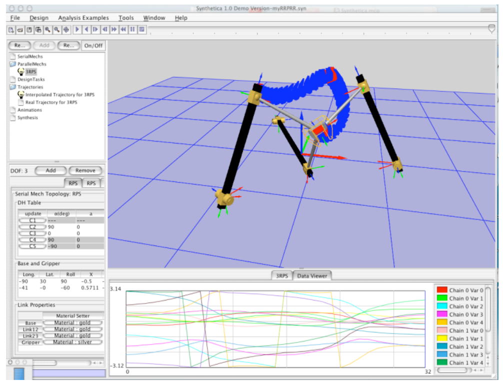


Parallel 2-TPR robot

- The constraints provide structural support in some directions, while allowing movement in the others.
- A constrained robot has less that six degrees of freedom. Its workspace is not a volume but rather a hypersurface of arbitrary shape.

Constraints	Assembly Categories	Total
5	5I, 3I-1II, 2I-1III, 1I-2II, 1I-1IV	487,990,859
4	4I, 2I-1II, 2II, 1I-1III, 1IV	16,734,569
3	3I, 1I-1II, 1III	464,417
2	2I, 1II	9,781
1	1I	139

Classification of constrained robotic systems

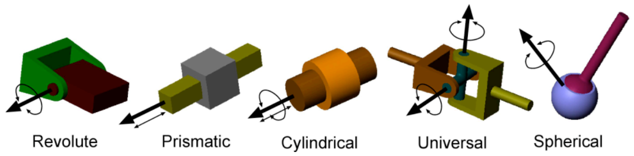


3-RPS constrained robot (category 3I, 3 degrees of freedom)

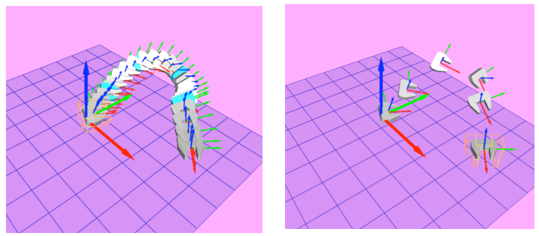
Finite-position Dimensional Synthesis

Finite-position Synthesis:

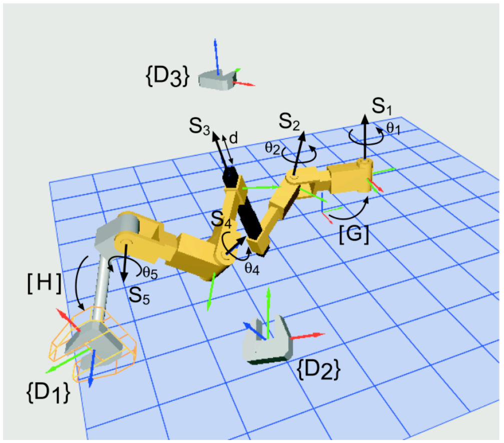
- **Given**: (a) a robot topology,



and (b) a task
(defined in terms of a set of positions
and orientations of a workpiece),



- **Find**: The location of the base, the location of the connection to the workpiece, and the dimensions of each link such the the chain reaches each task position exactly.

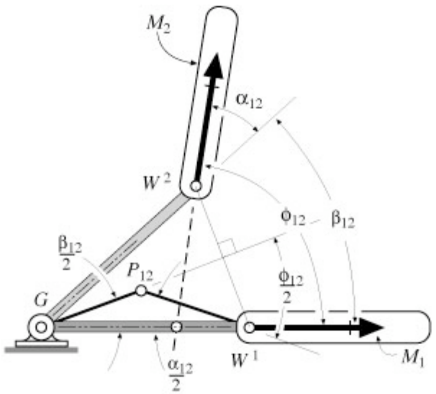


- A set of **design equations** evaluated at each of the task positions is used to determine the mechanism.
- There are different ways to formulate the set of design equations.

Finite Position Dimensional Synthesis
for Planar Robots

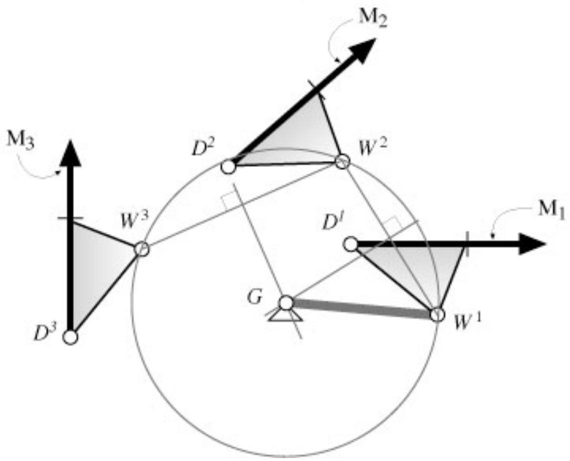
Approaches:

Graphical synthesis
Analytical constraint synthesis
Complex number formulation



Two position synthesis

$$(1 - e^{i\beta_{12}})\mathbf{G} + e^{i\beta_{12}}(1 - e^{i\alpha_{12}})\mathbf{W}^1 = (1 - e^{i\phi_{12}})\mathbf{P}_{12}$$



Three position synthesis

$$(1 - e^{i\beta_{12}})\mathbf{G} + e^{i\beta_{12}}(1 - e^{i\alpha_{12}})\mathbf{W}^1 = (1 - e^{i\phi_{12}})\mathbf{P}_{12}$$
$$(1 - e^{i\beta_{13}})\mathbf{G} + e^{i\beta_{13}}(1 - e^{i\alpha_{13}})\mathbf{W}^1 = (1 - e^{i\phi_{13}})\mathbf{P}_{13}$$

Finite Position Dimensional Synthesis for Spatial Robots

The Design Equations for Finite Position Synthesis can be obtained in several ways:

- **Geometric features** of the chain are used to formulate the algebraic constraint equations. (distance and angle constraints)

- **Kinematic geometry** based on the screw representation of the composition of displacements. (equivalent screw triangle)

- **Loop Closure Equations** along the chain from a reference configuration to each goal configuration.

- **Robot kinematics equations**, that define the set of positions reachable by the end-effector, are equated to each task position.

- **Relative kinematics equations using Clifford algebras (Dual Quaternion Synthesis):** Dual quaternion synthesis is a combination of Kinematic Geometry and Robot Kinematics Equations. It is, in addition, an extension of the complex number formulation to spatial robots.

Literature Review

Geometric features of the chain are used to formulate the algebraic constraint equations. (distance and angle constraints)

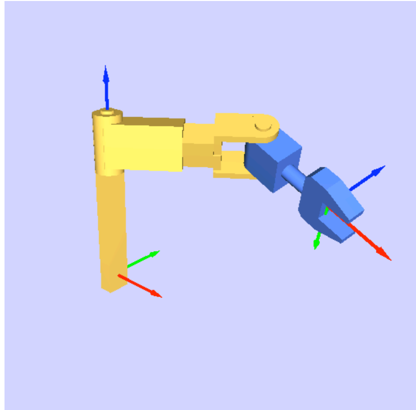
• Roth, B., 1968, "The design of binary cranks with revolute, cylindric, and prismatic joints", *J. Mechanisms*, 3(2):61-72.

• Chen, P., and Roth, B., 1969, ''Design Equations for the Finitely and Infinitesimally Separated Position Synthesis of Binary Links and Combined Link Chains,'' *ASME J. Eng. Ind*. 91(1):209–219.

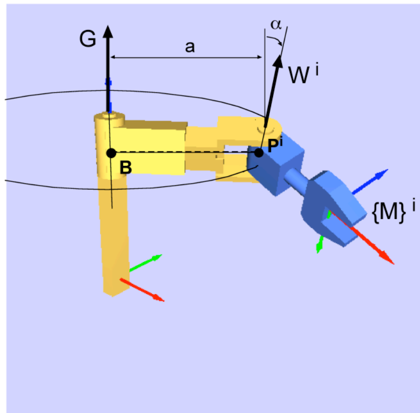
• Innocenti, C., 1994, ``Polynomial Solution of the Spatial Burmester Problem." *Mechanism Synthesis and Analysis,* ASME DE vol. 70.

• Nielsen, J. and Roth, B., 1995, "Elimination Methods for Spatial Synthesis," *Computational Kinematics*, (eds. J. P. Merlet and B. Ravani), Vol. 40 of Solid Mechanics and Its Applications, pp. 51-62, Kluwer Academic Publishers.

• Kim, H. S., and Tsai, L. W., 2002, "Kinematic Synthesis of Spatial 3-RPS Parallel Manipulators," *Proc. ASME Des. Eng. Tech. Conf.* paper no. DETC2002/MECH-34302, Sept. 29-Oct. 2, Montreal, Canada.

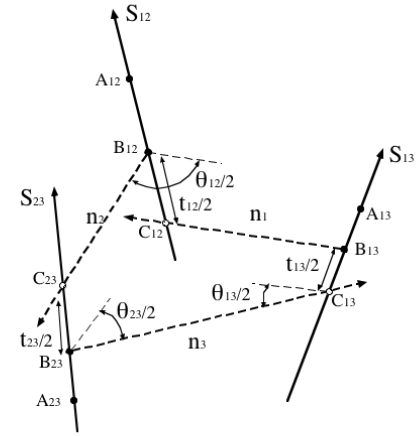


RR chain



Literature Review

- **Kinematic geometry** based on the screw representation of the composition of displacements. (equivalent screw triangle)



  - Tsai, L. W., and Roth, B., 1972, "Design of Dyads with Helical, Cylindrical, Spherical, Revolute and Prismatic Joints," *Mechanism and Machine Theory*, 7:591-598.

  - Tsai, L.W., and Roth, B., "A Note on the Design of Revolute-Revolute Cranks," *Mechanism and Machine Theory*, Vol. 8, pp. 23-31, 1973.

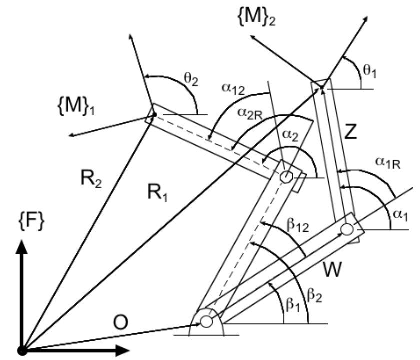
- **Loop closure equations** along the chain from a reference configuration to each goal configuration.

  - Sandor, G. N., and Erdman, A. G., 1984, *Advanced Mechanism Design: Analysis and Synthesis*, Vol. 2. Prentice-Hall, Englewood Cliffs, NJ

  .

  - Sandor, G.N., Xu, Y., and Weng, T.C., 1986, "Synthesis of 7-R Spatial Motion Generators with Prescribed Crank Rotations and Elimination of Branching", *The International Journal of Robotics Research*, 5(2):143-156.

  - Sandor, G.N., Weng, T.C., and Xu, Y., 1988, "The Synthesis of Spatial Motion Generators with Prismatic, Revolute and Cylindric Pairs without Branching Defect", Mechanism and Machine Theory, 23(4):269-274.



Literature Review

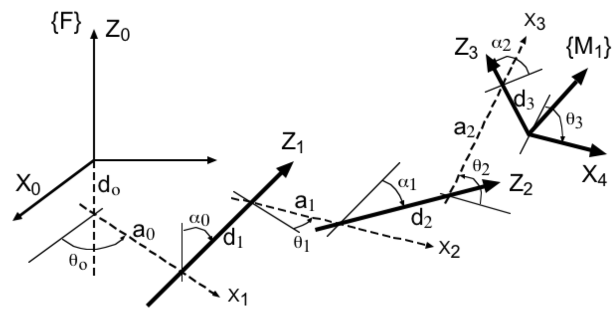
- **Robot kinematics equations** define the set of positions reachable by the end-effector. Equate to each task position to obtain design equations

- Park, F. C., and Bobrow, J. E., 1995, "Geometric Optimization Algorithms for Robot Kinematic Design". *Journal of Robotic Systems*, 12(6):453-463.

- 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, *J. Mechanical Design*, 123(1):58-67.

- Lee, E., and Mavroidis, D., 2002, "Solving the Geometric Design Problem of Spatial 3R Robot Manipulators Using Polynomial Homotopy Continuation", ASME J. of Mechanical Design, 124(4), pp.652-661.

- Lee, E., and Mavroidis, D., 2002c, "Geometric Design of Spatial PRR Manipulators Using Polynomial Elimination Techniques," Proc. ASME 2002 Design Eng. Tech. Conf., paper no. DETC2002/MECH-34314, Sept. 29-Oct. 2, Montreal, Canada.



$$[D] = [G][Z(\theta_1, d_1)][X(\alpha_1, a_1)] \ldots [Z(\theta_k, d_k)][H]$$

Literature Review

- **Relative kinematics equations using Clifford algebra** are used to obtain a formulation more directly related to the geometry of the problem.

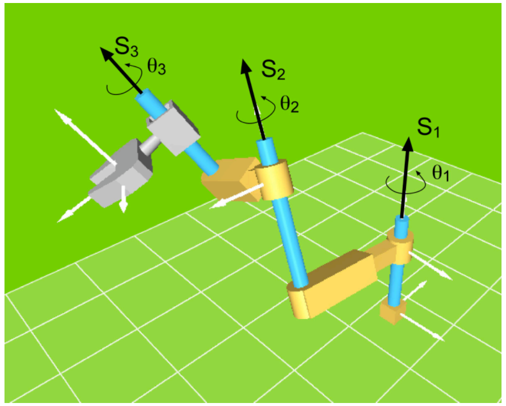
• Perez Gracia, A. and McCarthy, J.M., "The Kinematic Synthesis of Spatial Serial Chains Using Clifford Algebra Exponentials", Proceedings of the Institution of Mechanical Engineers, Part C, Journal of Mechanical Engineering Science, 220(7): 953-968, 2006.

• Perez, A. and McCarthy, J.M., "Clifford Algebra Exponentials and Planar Linkage Synthesis Equations", *ASME Journal of Mechanical Design*, 127(5): 931-940, September 2005.

•Perez, A. and McCarthy, J.M., "Geometric Design of RRP, RPR and PRR Serial Chains", *Mechanism and Machine Theory*, 40(11):1294-1311, November 2005.

•Perez, A. and McCarthy, J.M., "Sizing a Serial Chain to Fit a Task Trajectory Using Clifford Algebra Exponentials", *2005 IEEE International Conference on Robotics and Automation*, April 18-22, 2005, Barcelona.

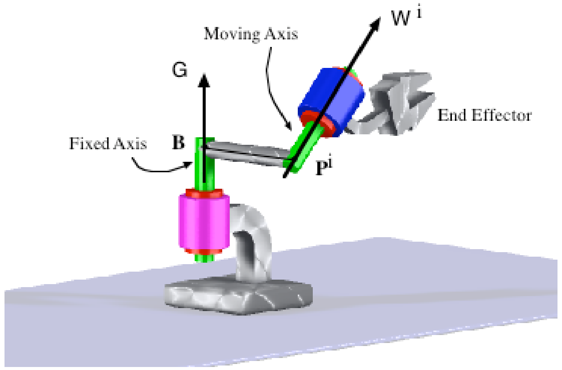
•Perez, A. and McCarthy, J.M., "Dual Quaternion Synthesis of Constrained Robotic Systems", *ASME Journal of Mechanical Design*, 126(3): 425-435, 2004.



Challenges of the Synthesis Problem

1. Stating the design equations

- Methods based on geometric constraints give simpler equations but lack a general methodology to find the constraints for all kinds of chains.
- Methods based on the kinematics equations are general but give a more complicated set of equations with extra variables.



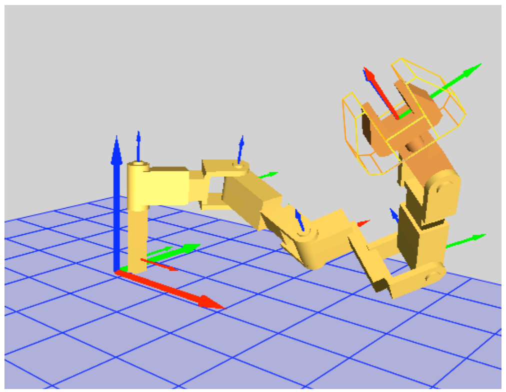
RR chain:
• 10 geometric constraints

$$\mathbf{G}\cdot[A_{1i}-I]\mathbf{V}^1+\mathbf{W}^1\cdot[A_{1i}-I]^T\mathbf{R}+\mathbf{G}\cdot[D_{1i}A_{1i}]\mathbf{W}^1=0,$$

$$\mathbf{G}\cdot[A_{1i}-I]\mathbf{W}^1=0,\quad i=2,3,$$

$$\mathbf{G}\cdot([T_{1i}]\mathbf{P}^1-\mathbf{B})=0,\quad \mathbf{W}^1\cdot(\mathbf{P}^1-[T_{1i}]^{-1}\mathbf{B})=0,$$

$$i=1,2,3.$$

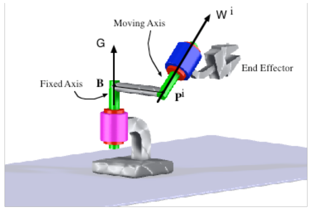


5R chain:
• geometric constraints? *(30 equations)*
• Using the kinematics equations, we obtain a set of 130 equations in 130 variables, including the joint angles.

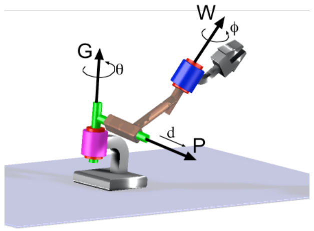
Challenges of the Synthesis Problem

2. Solving the design equations

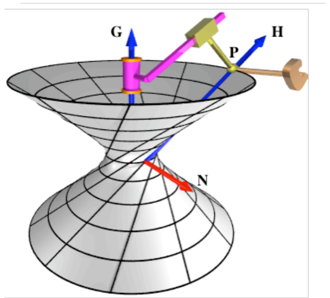
- Set of polynomial equations have a very high total degree.
- The joint variables may be eliminated to reduce the dimension of the problem.
- Due to internal structure, the equations have far less solutions than the Bezout bound.
- Some sample cases:



RR chain (2 dof robot):
- Initial total degree: $2^{10} = 1024$.
- Final solution: six roots, with only two real solutions.



RPR chain (3 dof robot):
- Initial total degree: $2^3 * 4^6 = 32768$.
- Final solution: 12 roots.



RPS chain (5 dof robot):
- Initial total degree: 262144.
- Final solution: 1024 roots?

Source: Hai-Jun Su

Synthesis of Robots
Using Clifford Algebras

Synthesis of Robots - Outline

1. **Create kinematics equations expressed as dual quaternions**
   - Composition of relative screw displacements from a reference position using Clifford product.
2. **Counting**
   - Compute n_{max}, maximum number of complete task positions for a desired topology.
3. **Create design equations**
   - Equate dual quaternion kinematics equations to n_{max} task displacements.
4. **Solve the design equations**
   - Solve numerically in parameterized form (including joint variables), or
   - Eliminate the joint variables to obtain a set of reduced equations
     - For those cases where it is possible, algebraic elimination leads to a closed solution:
       - Different algebraic methods (resultant, matrix eigenvalue, …) to create a univariate polynomial for finding all possible solutions.
     - For those cases that are too big for algebraic elimination, numerical methods to find solutions:
       - Polynomial continuation methods.
       - Newton-Raphson numerical methods.

Robot Kinematics Equations

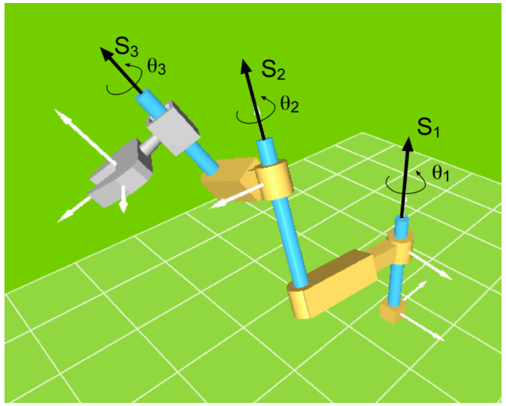
4x4 Homogeneous Transforms

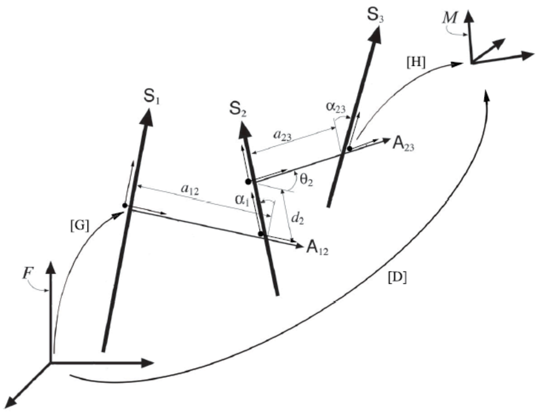
- The **kinematics equations** of the robot relate the motion of the end-effector to the composition of transformations about each joint axis.
- Matrix representation using Denavit-Hartenberg parameters:

$$[D_i] = [G][Z(\theta_1^i, d_1^i)][X(\alpha_{12}, a_{12})][Z(\theta_2^i, d_2^i)] \\ [X(\alpha_{n-1,n}, a_{n-1,n})][Z(\theta_m^i, d_m^i)][H], \quad i = 1, \dots n$$

$$[Z(\theta_i, d_i)] = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$[X(\alpha_{i,i+1}, a_{i,i+1})] = \begin{bmatrix} 1 & 0 & 0 & a_{i,i+1} \\ 0 & \cos\alpha_{i,i+1} & -\sin\alpha_{i,i+1} & 0 \\ 0 & \sin\alpha_{i,i+1} & \cos\alpha_{i,i+1} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$





- Relative displacements are used to represent the motion from a reference configuration $[D_0]$,

$$[D_{0i}] = [D_i][D_0]^{-1} = [T(\Delta\theta_1^i, \mathsf{S}_1)][T(\Delta\theta_2^i, \mathsf{S}_2)] \dots [T(\Delta\theta_m^i, \mathsf{S}_m)]$$

Robot Kinematics Equations
Matrix Exponential Form

- The transformations $[T(\Delta\theta_i, \Delta d_i, S_i)]$ can be seen as the matrix exponential of a Lie Algebra element $[J_i]$:
- Let J_i be the screw defined by joint axis S_i, $J_i = (\mathbf{S}_i, \mathbf{C}_i \times \mathbf{S}_i + \mu_i \mathbf{S}_i)$, with $\mu_i = \Delta d_i / \Delta\theta_i$ being the pitch of the screw.

$$[J_i] = \begin{bmatrix} 0 & -s_{z,i} & s_{y,i} & v_{x,i} \\ s_{z,i} & 0 & -s_{x,i} & v_{y,i} \\ -s_{y,i} & s_{x,i} & 0 & v_{z,i} \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

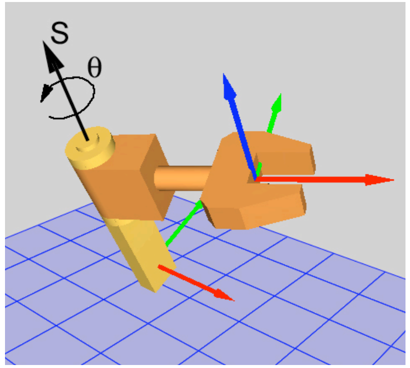
- The transformation can be written as:

$$[T(\Delta\theta_i, \Delta d_i, S_i)] = e^{\Delta\theta_i J_i}$$

- The relative kinematics equations, expressed as a product of exponentials, take the form:

$$[D_{0i}] = [D(\Delta\vec{\theta})] = e^{\Delta\theta_1 J_1} e^{\Delta\theta_2 J_2} \ldots e^{\Delta\theta_m J_m}$$

Clifford Algebra Formulation



- The Clifford algebra of P^3: sixteen-dimensional vector space with a Clifford, or geometric, product.
- The subalgebra $C^+(P^3)$ can be identified with the set of 4x4 homogeneous transforms.
- A spatial displacement is identified with a unit dual quaternion obtained as the Clifford algebra exponential of the screw axis S.
- For a rotation of angle θ and a slide d about and along the line of Plucker coordinates $\mathsf{S} = \mathbf{S} + \varepsilon\mathbf{S^0} = \mathbf{S} + \varepsilon\mathbf{C} \times \mathbf{S}$,

$$\hat{D}(\hat{\theta}) = e^{\frac{\hat{\theta}}{2}\mathsf{S}} = \cos\frac{\hat{\theta}}{2} + \sin\frac{\hat{\theta}}{2}\mathsf{S} = \begin{Bmatrix} \sin\frac{\theta}{2}s_1 \\ \sin\frac{\theta}{2}s_2 \\ \sin\frac{\theta}{2}s_3 \\ \cos\frac{\theta}{2} \end{Bmatrix} + \epsilon \begin{Bmatrix} \sin\frac{\theta}{2}s_1^0 + \frac{d}{2}\cos\frac{\theta}{2}s_1 \\ \sin\frac{\theta}{2}s_2^0 + \frac{d}{2}\cos\frac{\theta}{2}s_2 \\ \sin\frac{\theta}{2}s_3^0 + \frac{d}{2}\cos\frac{\theta}{2}s_3 \\ -\frac{d}{2}\sin\frac{\theta}{2} \end{Bmatrix}$$

- Clifford algebra kinematics equations for a relative displacement,

$$\begin{aligned} \hat{D}(\Delta\hat{\vec{\theta}}) &= e^{\frac{\Delta\hat{\theta}_1}{2}\mathsf{S}_1} e^{\frac{\Delta\hat{\theta}_2}{2}\mathsf{S}_2} \cdots e^{\frac{\Delta\hat{\theta}_n}{2}\mathsf{S}_n}, \\ &= (\mathrm{c}\frac{\Delta\hat{\theta}_1}{2} + \mathrm{s}\frac{\Delta\hat{\theta}_1}{2}\mathsf{S}_1)(\mathrm{c}\frac{\Delta\hat{\theta}_2}{2} + \mathrm{s}\frac{\Delta\hat{\theta}_2}{2}\mathsf{S}_2) \cdots (\mathrm{c}\frac{\Delta\hat{\theta}_n}{2} + \mathrm{s}\frac{\Delta\hat{\theta}_n}{2}\mathsf{S}_n). \end{aligned}$$

Unit Dual Quaternions

Geometric Meaning

- Another way of expressing elements of the group of spatial displacements SE(3)

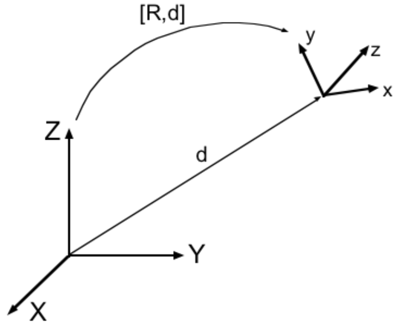
Matrix formulation:

$$\begin{Bmatrix} X \\ Y \\ Z \\ 1 \end{Bmatrix} = \begin{bmatrix} & [R] & & \mathbf{d} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x \\ y \\ z \\ 1 \end{Bmatrix}$$

Dual quaternion formulation:

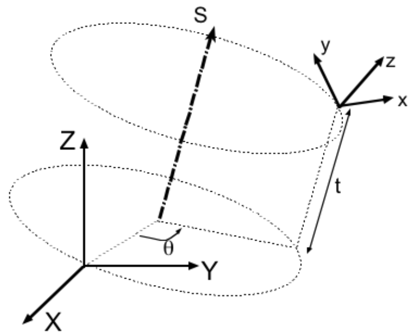
$$\hat{X} = \hat{Q}\hat{x}\hat{Q}^*$$

$$\hat{Q}(\hat{\theta}) = \sin(\frac{\hat{\theta}}{2})\mathsf{S} + \cos(\frac{\hat{\theta}}{2})$$



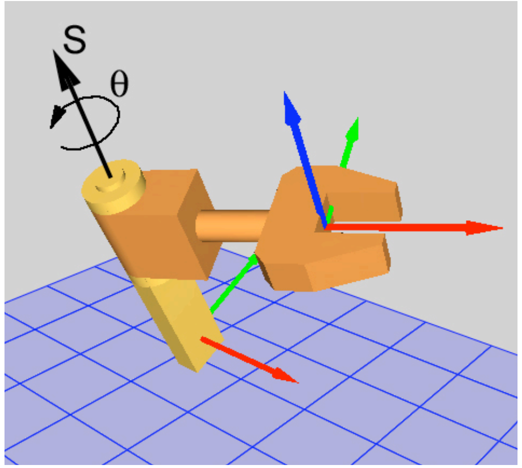
- We can write them as four-dimensional dual vectors,

$$\hat{Q} = \begin{Bmatrix} q_x \\ q_y \\ q_z \\ q_w \end{Bmatrix} + \epsilon \begin{Bmatrix} q_x^0 \\ q_y^0 \\ q_z^0 \\ q_w^0 \end{Bmatrix} = \begin{Bmatrix} \sin\frac{\theta}{2}s_x \\ \sin\frac{\theta}{2}s_y \\ \sin\frac{\theta}{2}s_z \\ \cos\frac{\theta}{2} \end{Bmatrix} + \epsilon \begin{Bmatrix} \sin\frac{\theta}{2}s_x^0 + \frac{d}{2}\cos\frac{\theta}{2}s_x \\ \sin\frac{\theta}{2}s_y^0 + \frac{d}{2}\cos\frac{\theta}{2}s_y \\ \sin\frac{\theta}{2}s_z^0 + \frac{d}{2}\cos\frac{\theta}{2}s_z \\ -\frac{d}{2}\sin\frac{\theta}{2} \end{Bmatrix}$$



- The dual vector S is the screw axis of the transformation, the dual numbers contain the information about the rotation and translation.

Unit Dual Quaternions
Geometric Meaning



In particular, expression of the rotation about a revolute joint of axis S and rotation angle θ:

- **Matrix formulation** (Lie algebra exponential):

$$e^{\theta J} = \begin{bmatrix} e^{\theta \mathbf{S}} & (I - e^{\theta \mathbf{S}})(\mathbf{s} \times (\mathbf{s} \times \mathbf{c})) \\ 0 & 1 \end{bmatrix}$$

expands to:

$$e^{\theta J} = \begin{bmatrix} 1-(1-c\theta)(s_2^2+s_3^2) & s_1 s_2(1-c\theta) - s_3 s\theta & s_1 s_3 (1-c\theta) + s_2 s\theta & c_1(1-c\theta) + (c_2 s_3 - c_3 s_2) s\theta \\ s_1 s_2 (1-c\theta) + s_3 s\theta & 1-(1-c\theta)(s_1^2+s_3^2) & s_2 s_3 (1-c\theta) - s_1 s\theta & c_2(1-c\theta) + (c_3 s_1 - c_1 s_3) s\theta \\ s_1 s_3 (1-c\theta) - s_2 s\theta & s_2 s_3 (1-c\theta) + s_1 s\theta & 1-(1-c\theta)(s_1^2+s_2^2) & c_3(1-c\theta) + (c_1 s_2 - c_2 s_1) s\theta \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- **Dual quaternion formulation** (Clifford algebra exponential):

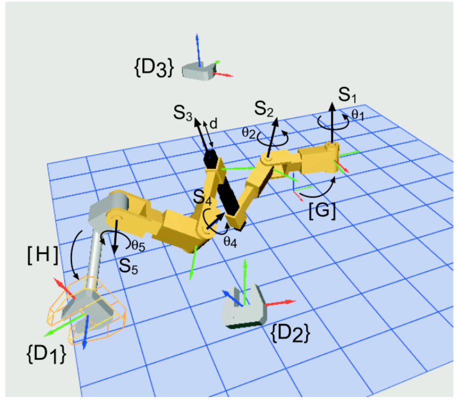
$$e^{\frac{\theta}{2} J} = \begin{Bmatrix} \sin\frac{\theta}{2}(s_1 + \epsilon(c_2 s_3 - c_3 s_2)) \\ \sin\frac{\theta}{2}(s_2 + \epsilon(c_3 s_1 - c_1 s_3)) \\ \sin\frac{\theta}{2}(s_3 + \epsilon(c_1 s_2 - c_2 s_1)) \\ \cos\frac{\theta}{2} \end{Bmatrix}$$

Dual Quaternion Design Equations

- Specify the m goal positions, P_1, …, P_m.

- Calculate the relative transformations from the first position and express them as elements of the Clifford algebra, P_{1j}, j=2,..,m.

- Equate the kinematics equations to each of the goal transformations,



$$\hat{D}^i = \hat{S}_1(\Delta\hat{\theta}^i_1)\hat{S}_2(\Delta\hat{\theta}^i_2)\ldots\hat{S}_m(\Delta\hat{\theta}^i_m)$$

- Create the **design equations**: equate the kinematics equations to each task position written in dual quaternion form:

$$\hat{S}_1(\Delta\hat{\theta}^i_1)\hat{S}_2(\Delta\hat{\theta}^i_2)\ldots\hat{S}_m(\Delta\hat{\theta}^i_m) - \hat{P}_{1i} = 0, \quad i = 2, \ldots, n$$

  - We obtain a set of 8-dim. vector equations where the variables to solve for are the **Plucker coordinates of the axes** S_j in the reference position.
  - The equations are parameterized by the joint variables θ_j, j=1,…m.

Counting

- How many complete **task positions** can we define?

  Consider a serial chain with r revolute joints and t prismatic joints, and n task positions.

Parameters:

- R joint-- 6 components of a dual vector, $6r$.
- P joint-- 3 components of a direction vector, $3t$.
- Joint variables, $(r+t)(n-1)$, measured relative to initial configuration.

Dual Quaternion design equations, $6(n-1)$

Associated constraint equations:

- R joint-- 2 constraints (Plucker conditions), $2r$.
- P joint-- 1 constraint (unit vector), t .

Imposed extra constraint equations, c.

Equations: $6(n-1)+2r+t+c$. Unknowns: $6r+3t+(r+t)(n-1)$.

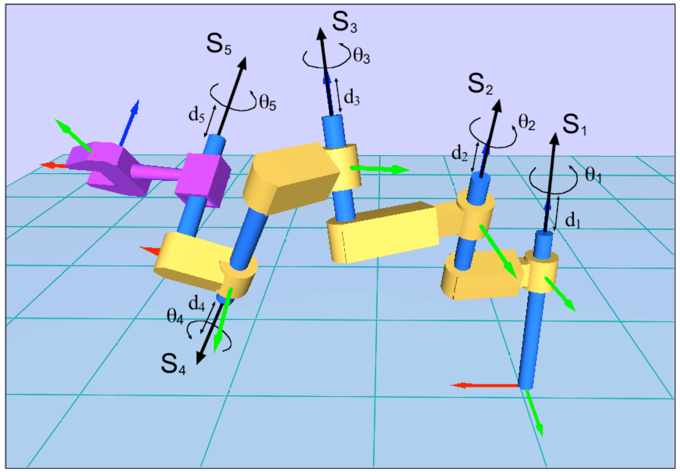
$$n_{max} = (6 + 3r + t - c)/(6 - r - t)$$

(note $r+t < 6$ for constrained robotic systems)

Structure of the Synthesis Equations

Consider the 5C Serial Chain

- 5 cylindrical joints, 10 degrees of freedom.
- Relative kinematics equations for the 5C chain, Clifford algebra exponential form:



$$\hat{Q}_{5C}(\vec{\theta}) = e^{\frac{\hat{\theta}_1}{2}\mathsf{S}_1} e^{\frac{\hat{\theta}_2}{2}\mathsf{S}_2} e^{\frac{\hat{\theta}_3}{2}\mathsf{S}_3} e^{\frac{\hat{\theta}_4}{2}\mathsf{S}_4} e^{\frac{\hat{\theta}_5}{2}\mathsf{S}_5}$$

- where each axis is defined as:

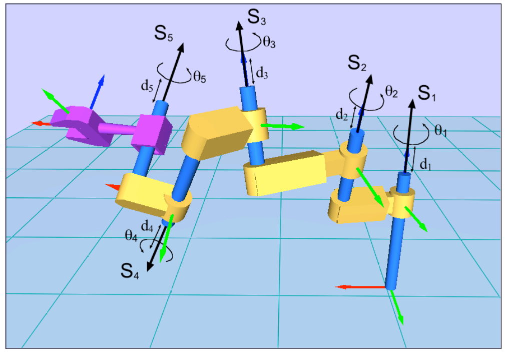
$$e^{\frac{\hat{\theta}_i}{2}\mathsf{S}_i} = \begin{Bmatrix} \sin\frac{\theta_i}{2}\mathbf{S}_i \\ \cos\frac{\theta_i}{2} \end{Bmatrix} + \epsilon \begin{Bmatrix} \sin\frac{\theta_i}{2}\mathbf{S}_i^0 + \frac{d_i}{2}\cos\frac{\theta_i}{2}\mathbf{S}_i \\ -\frac{d_i}{2}\sin\frac{\theta_i}{2} \end{Bmatrix}$$

Any 5-jointed serial chain can be derived by specializing the 5C expression.

Structure of the Synthesis Equations

- The equations can be collected as terms in the products of joint variables,
  - Products of the sines and cosines of the joint angles (32 monomials),

$$\mathbf{V} = \{s_1 s_2 s_3 s_4 s_5, (s_1 s_2 s_3 s_4 c_5)_5, (s_1 s_2 s_3 c_4 c_5)_{10}, (s_1 s_2 c_3 c_4 c_5)_{10}, (s_1 c_2 c_3 c_4 c_5)_5, c_1 c_2 c_3 c_4 c_5\}$$



  - plus the terms containing the slides. Total: 192 monomials,

$$\mathbf{M} = \{\mathbf{V}, \frac{\Delta d_1}{2}\mathbf{V}, \frac{\Delta d_2}{2}\mathbf{V}, \frac{\Delta d_3}{2}\mathbf{V}, \frac{\Delta d_4}{2}\mathbf{V}, \frac{\Delta d_5}{2}\mathbf{V}\}$$

- Write the kinematics equations as the sum of terms,

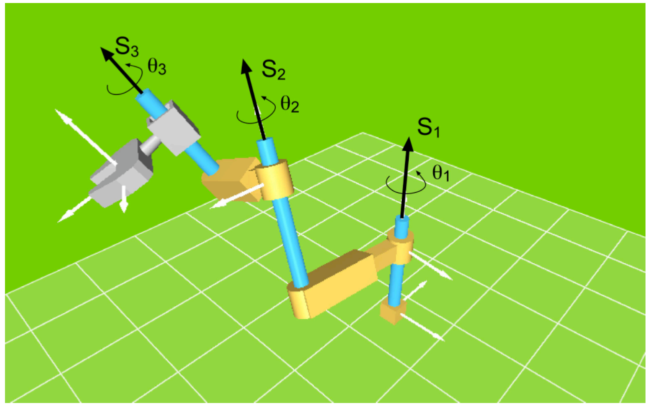
$$\hat{Q}_{5C} = \sum_{i=1}^{192} \mathbf{K}_i M_i, \quad M_i \in \mathbf{M},$$

where the 8-dimensional column vectors $\mathbf{K}_i$ contain the structural variables of the joint axes.

Joints	2	3	4	5
Terms	12	32	80	192

Example: The 3C Spatial Serial Chain

- Clifford algebra relative kinematics equations,



$$\hat{D}(\Delta\hat{\vec{\theta}}) = (\mathrm{c}\frac{\Delta\hat{\theta}_1}{2} + \mathrm{s}\frac{\Delta\hat{\theta}_1}{2}\mathsf{S}_1)(\mathrm{c}\frac{\Delta\hat{\theta}_2}{2} + \mathrm{s}\frac{\Delta\hat{\theta}_2}{2}\mathsf{S}_2)(\mathrm{c}\frac{\Delta\hat{\theta}_3}{2} + \mathrm{s}\frac{\Delta\hat{\theta}_3}{2}\mathsf{S}_3),$$

- Expand to:

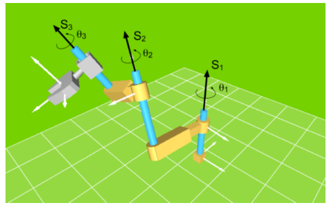
$$\begin{aligned}\hat{D}(\Delta\hat{\vec{\theta}}) =&(\hat{c}_1\hat{c}_2 - \hat{s}_1\hat{s}_2\mathsf{S}_1\cdot\mathsf{S}_2 + \hat{s}_1\hat{c}_2\mathsf{S}_1 + \hat{c}_1\hat{s}_2\mathsf{S}_2 + \hat{s}_1\hat{s}_2\mathsf{S}_1\times\mathsf{S}_2)(\hat{c}_3 + \hat{s}_3\mathsf{S}_3),\\ =&\hat{c}_1\hat{c}_2\hat{c}_3 - \hat{s}_1\hat{s}_2\hat{c}_3\mathsf{S}_1\cdot\mathsf{S}_2 - \hat{s}_1\hat{c}_2\hat{s}_3\mathsf{S}_1\cdot\mathsf{S}_3 - \hat{c}_1\hat{s}_2\hat{s}_3\mathsf{S}_2\cdot\mathsf{S}_3 - \hat{s}_1\hat{s}_2\hat{s}_3\mathsf{S}_1\times\mathsf{S}_2\cdot\mathsf{S}_3\\ &+ \hat{s}_1\hat{c}_2\hat{c}_3\mathsf{S}_1 + \hat{c}_1\hat{s}_2\hat{c}_3\mathsf{S}_2 + \hat{s}_1\hat{s}_2\hat{c}_3\mathsf{S}_1\times\mathsf{S}_2 + \hat{c}_1\hat{c}_2\hat{s}_3\mathsf{S}_3 - \hat{s}_1\hat{s}_2\hat{s}_3(\mathsf{S}_1\cdot\mathsf{S}_2)\,\mathsf{S}_3\\ &+ \hat{s}_1\hat{c}_2\hat{s}_3\mathsf{S}_1\times\mathsf{S}_3 + \hat{c}_1\hat{s}_2\hat{s}_3\mathsf{S}_2\times\mathsf{S}_3 + \hat{s}_1\hat{s}_2\hat{s}_3(\mathsf{S}_1\times\mathsf{S}_2)\times\mathsf{S}_3. \qquad (\end{aligned}$$

- The kinematics equations can be written as a linear combination of the products of joint variables:

$$\mathbf{M} = (\mathbf{V}, \frac{\Delta d_1}{2}\mathbf{V}, \frac{\Delta d_2}{2}\mathbf{V}, \frac{\Delta d_3}{2}\mathbf{V}),$$

$$\hat{\mathbf{V}} = (\hat{c}_1\hat{c}_2\hat{c}_3, \hat{s}_1\hat{c}_2\hat{c}_3, \hat{c}_1\hat{s}_2\hat{c}_3, \hat{s}_1\hat{s}_2\hat{c}_3, \hat{c}_1\hat{c}_2\hat{s}_3, \hat{s}_1\hat{c}_2\hat{s}_3, \hat{c}_1\hat{s}_2\hat{c}_3, \hat{s}_1\hat{s}_2\hat{s}_3)^T.$$

Example: The 3C Spatial Serial Chain



- Finally, we can write the equations as the linear system:

$$\hat{D}(\Delta\hat{\vec{\theta}}) =$$

$$\begin{bmatrix} 0 & S_1 & S_2 & S_1 \times S_1 & S_3 & S_1 \times S_3 & S_2 \times S_3 & -(S_1 \cdot S_2)\, S_3 + (S_1 \times S_2) \times S_3 \\ 1 & 0 & 0 & -S_1 \cdot S_2 & 0 & -S_1 \cdot S_3 & -S_2 \cdot S_3 & -S_1 \times S_2 \cdot S_3 \end{bmatrix} \hat{V}.$$

- The equations of the CCC chain can be specialized to those of any 3-jointed robot

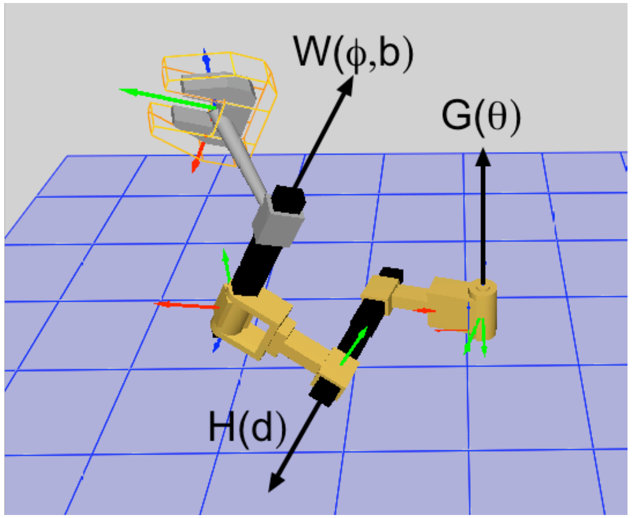
Serial Chain	Condition	Terms
CCR	d3 = 0	24
CRR	d2 = 0, d3 = 0	16
RRR	d1 = d2 = d3 = 0	8
CCP	s3 = 0	16
CPP	s2 = 0, s3 = 0	8
CPR	s2 = 0, d3 = 0	12
RRP	d1 = d2 = 0, s3 = 0	8
RPP	d1 = 0, s2 = s3 = 0	6
CT	d2 = 0, d3 = 0	16
RT	d1 = d2 = d3 = 0	8
PT	s1 =0, d2 = d3 = 0	8

Design Example: RPC Robot

- **Design equations**

$$\hat{Q}_{RPC}(\theta, d, \phi, b) = \hat{G}(\theta, 0)\hat{H}(0, d)\hat{W}(\phi, b)$$

$$\hat{Q}_{RPC}(\theta^i, d^i, \phi^i, b^i) = \hat{P}^i$$

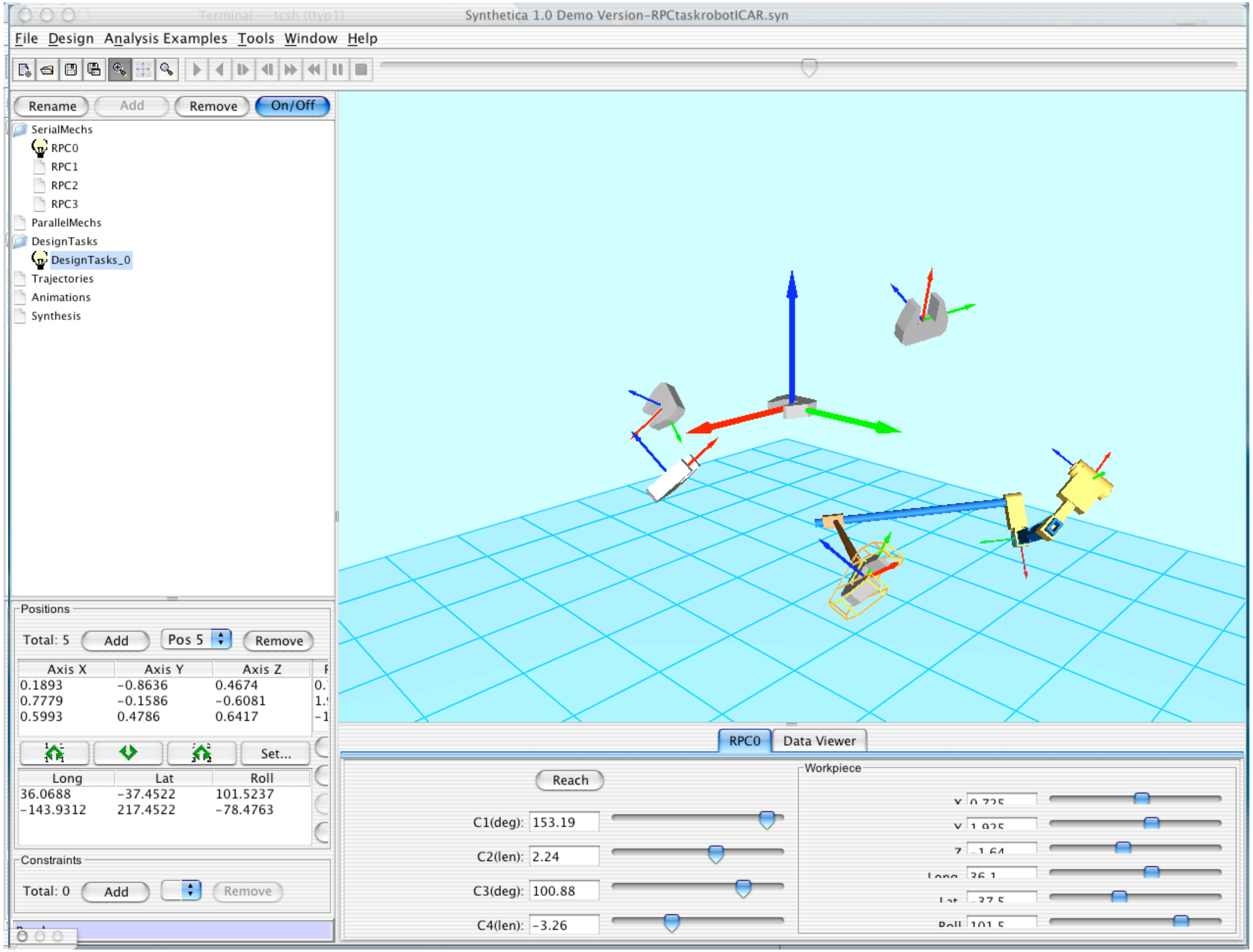


- We can define a **maximum** of n=5 task positions if we impose the conditions g.h=0, w.h=0.
- From the initial set of 34 equations, eliminate the joint variables to obtain a set of 15 equations in 15 parameters,

$$\begin{aligned}
&\{R_1, R_2\}^i, \quad i = 1, \ldots 4,\\
&\mathbf{g}\cdot\mathbf{g} = 1, \quad \mathbf{g}\cdot\mathbf{g}_0 = 0,\\
&\mathbf{w}\cdot\mathbf{w} = 1, \quad \mathbf{w}\cdot\mathbf{w}_0 = 0,\\
&\mathbf{h}\cdot\mathbf{h} = 1, \quad \mathbf{h}\cdot\mathbf{g} = 0, \quad \mathbf{h}\cdot\mathbf{w} = 0.
\end{aligned}$$

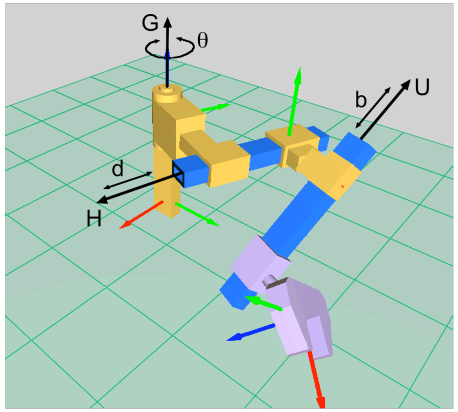
For the RPC chain, we can use resultant methods to obtain a sixth-degree univariate polynomial. The maximum number of solutions is six.

Design Example: RPC Robot

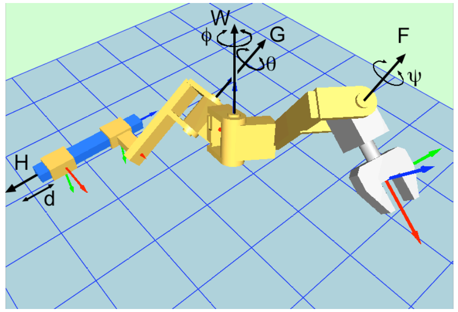


Summary of the Results

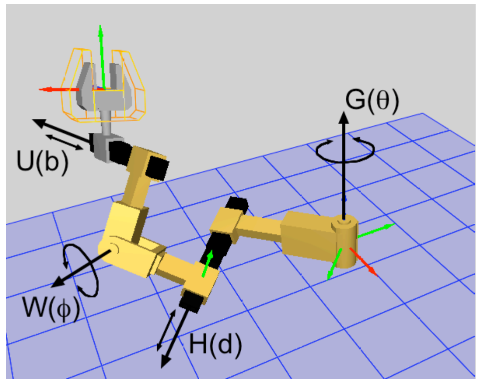
	Robot	DOF	n	e	structural vars.	joint vars.	***Previous results***
	C	2	2	0	6	2	
★	RP	2	2	1	9	3	
★	RR	2	3	0	12	4	Suh 1969, Tsai & Roth 1972.
	PC	3	2	3	9	9	
★	RPP	3	2	5	12	13	Partial solution: Tsai 1972.
	RC	3	$3\frac{2}{3}$	0	12	6	
★	TP	3	$3\frac{2}{3}$	0	12	6	
★	RRP	3	$4\frac{1}{3}$	0	15	9	
★	RRR	3	5	0	18	12	Partial: Sandor 1988, complete: Mavroidis 2000.
	CC	4	5	0	12	16	
★	RPC	4	5	2	15	20	Partial solution: Sandor 1988.
★	RRPP	4	5	4	18	24	
★	RRC	4	7	0	18	24	
★	TPR	4	7	0	18	24	
★	RRRP	4	8	0	21	28	
★	RRRR	4	9	0	24	32	Partial solution: Sandor 1988.
★	RCC	5	13	0	18	60	
★	TPT	5	15	0	21	70	
	RRPC	5	15	0	21	70	
	RRRPP	5	17	0	24	80	
	RRRC	5	17	0	24	80	
	RRRRP	5	19	0	27	90	
	RRRRR	5	21	0	30	100	



RPP chain



PRRR chain



RPRP chain

Problems to Solve

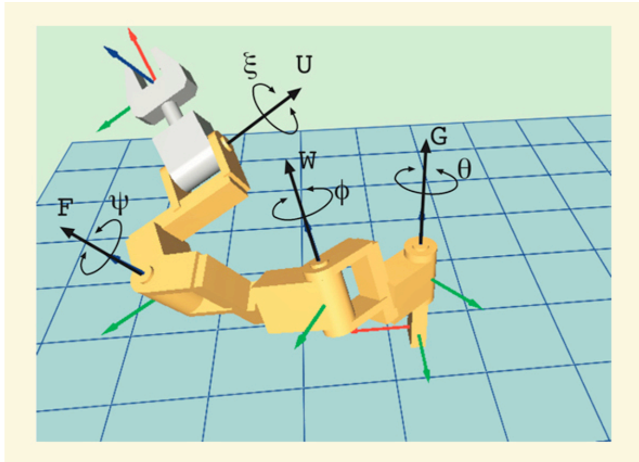
- For most of the five-dof serial chains, the complexity of the equations is too high even to eliminate the joint variables; numerical parameterized solution seems to be the only option so far.
- Numerical results seem to indicate a very high number of solutions:

(Example: RRRR serial robot)

RR: **2 real solutions**

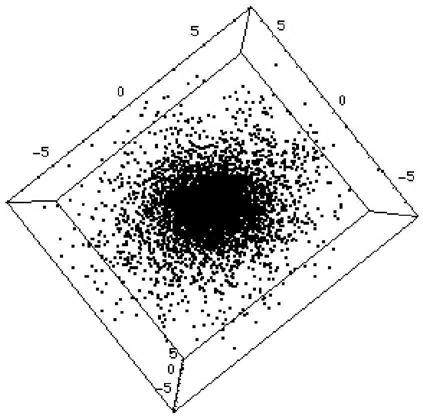
RRR: **approx. 13 real solutions?**

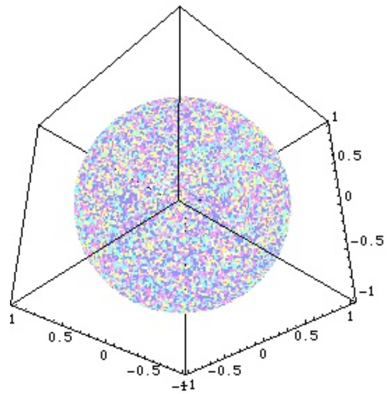
RRRR: **thousands?**



Numerical results:

total solutions	total non-repeated
167	167
382	381
459	456
652	642
4804	4366
5704	5123
6592	5826



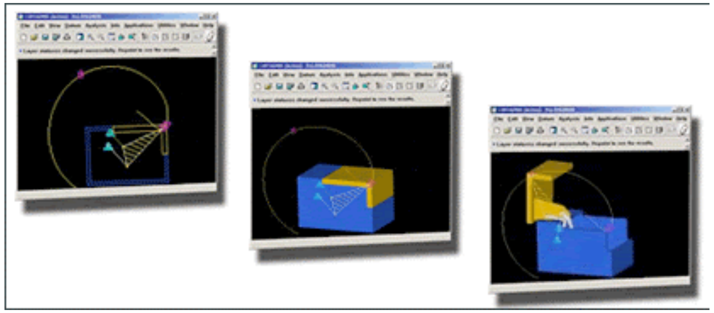


Some Applications of the Kinematic Synthesis Robots

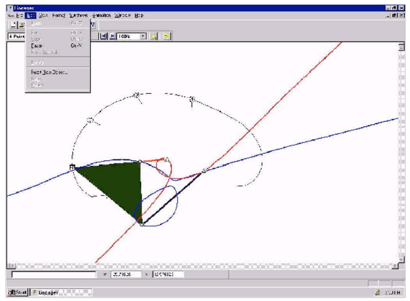
- Computer-aided design of robotic systems
  - Constrained serial chains
  - Optimization for 6 or more dof serial robots
- Identification of kinematic structures from video images
  - Avatar synthesis
  - Hand skeleton identification

Computer-aided Linkage Design

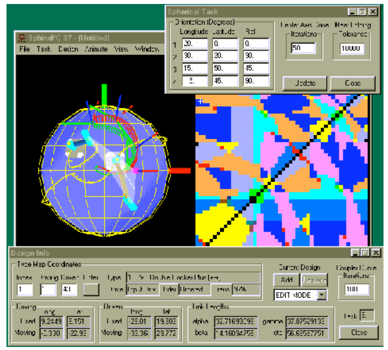
- **Commercial computer-aided design packages**
  - Drawing
  - CAM integration
  - Motion analysis / inverse kinematics
  - Force analysis / dynamic simulation
  - Stress/strain analysis / FEA



- **Commercially available synthesis software**
  - SyMech: planar 4 to 12-bar linkages.
  - WATT (Heron technologies): planar 4, 6, and 8-bar linkages.
  - SAM (Artas): planar 4-bar linkage, other for specific applications.



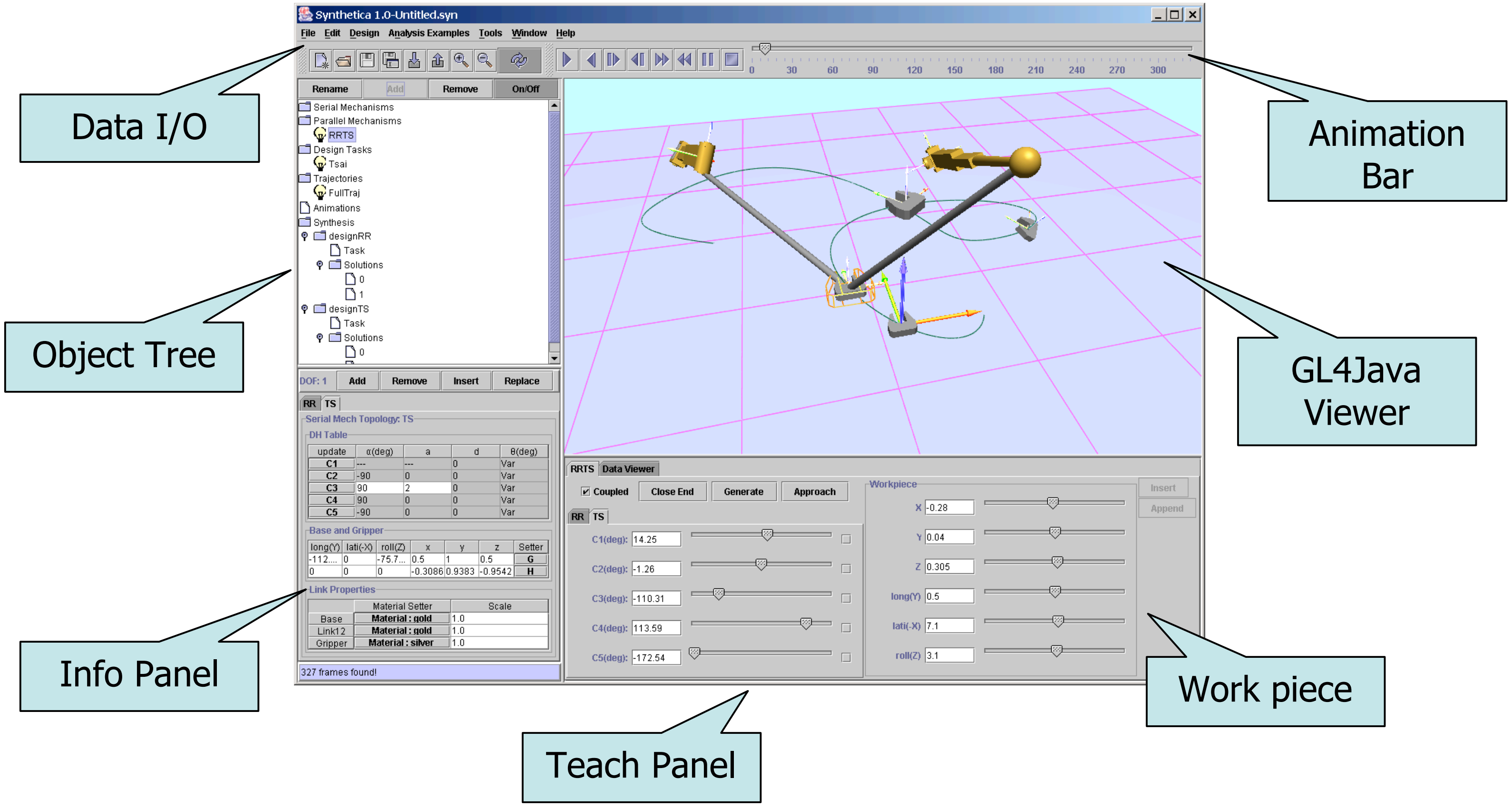
- **Research software**
  - Planar: LINCAGES, Erdman and Gustafson, 1977; RECSYN, Waldron and Song, 1981.
  - Spherical, spatial: SPHINX, Ruth and McCarthy, 1997; SPADES, Kihonge, Vance and Larochelle, 2002; Synthetica 1.0, Su, Collins and McCarthy, 2002.



Computer-aided Design of Robotic Systems:

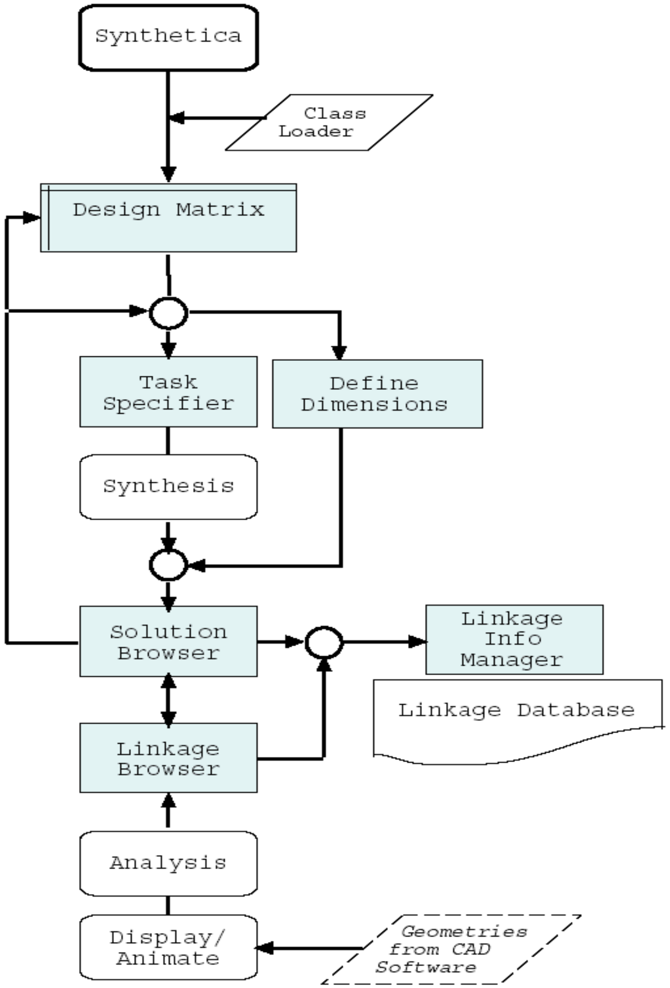
Extendable Synthesis and Analysis Software

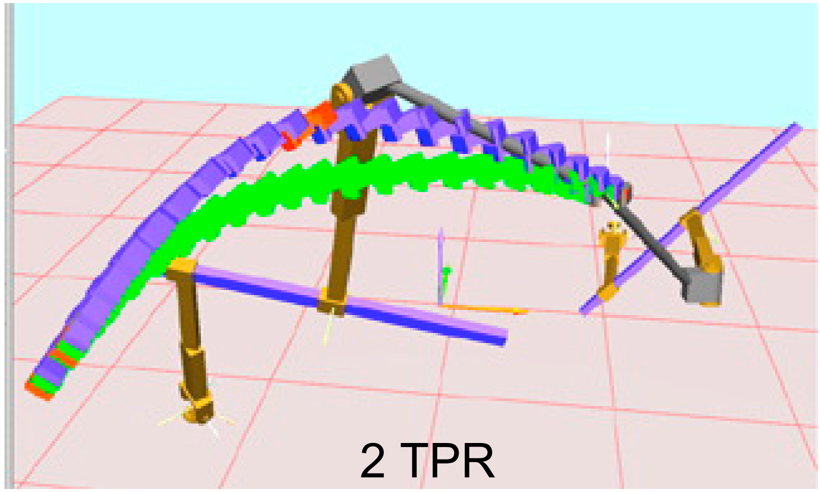
The results of the dual quaternion synthesis need to be provided to the mechanical designer of spatial mechanisms in an environment that allows him or her to define tasks, synthesize robots, simulate the movement and rank the possible solutions.

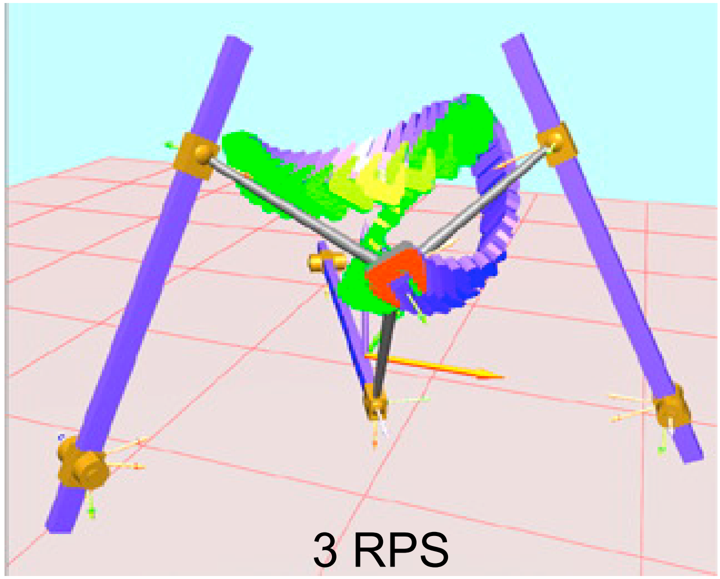


Source: Synthetica 1.0, developed by Hai Jun Su, Curtis Collins and J.M. McCarthy

Synthetica 2.0: Flow Chart

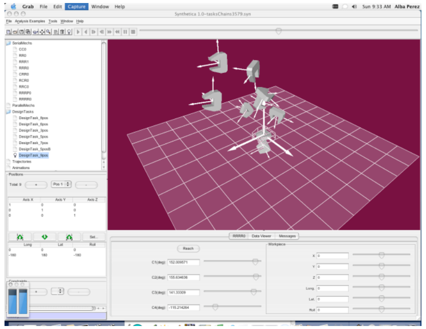
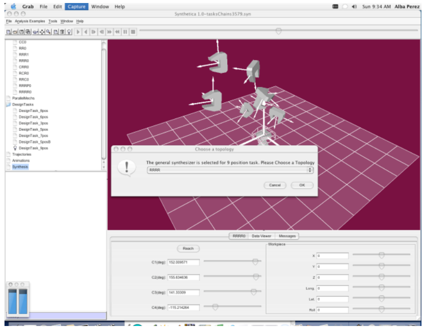
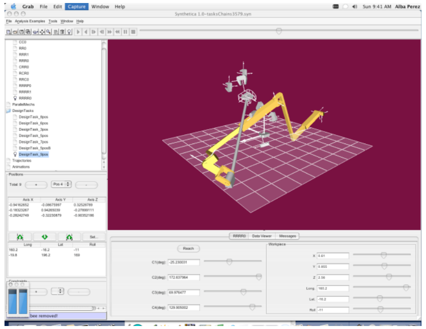
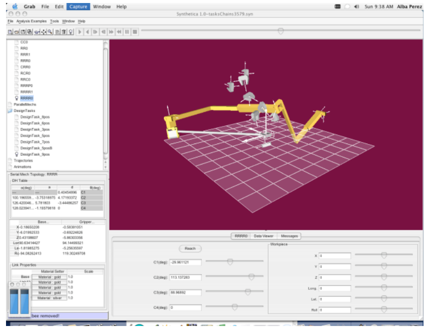






Synthetica 2.0: General Synthesizer

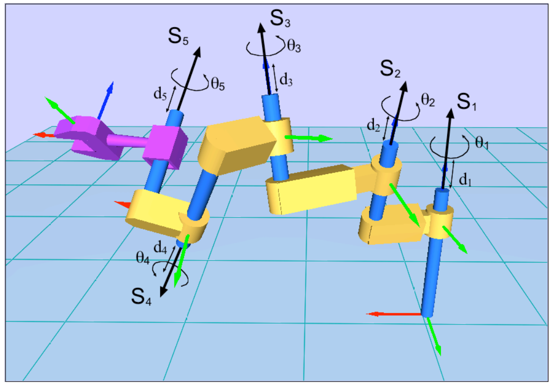
- Default design procedure to be used when there is no specialized design routine.
- A total of 120 topologies consisting of R, P, C, T and S joints and ranging from 2 to 5 degrees of freedom.

- Design equations are created by equating the dual quaternion synthesis equations to the task positions.
- The counting formula allows to assign serial chain topologies to each task.
- Java translation of FORTRAN Minpack numerical solver (Steve Verrill, translator).

Synthetica 2.0: General Synthesizer

To formulate the design equations, write the Clifford algebra kinematics equations,



$$\hat{KE}_m = \hat{S}_1(\theta_1, d_1)\hat{S}_2(\theta_2, d_2)\ldots\hat{S}_m(\theta_m, d_m)$$

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

specialize the joints if needed,

Joint Type	*Axes*	*Constraints*
R	$\mathsf{S}_i(\theta_i, d_i)$	$d_i = 0$
P	$\mathsf{S}_i(\theta_i, d_i)$	$\theta_i = 0$
C	$\mathsf{S}_i(\theta_i, d_i)$	none
T	$\mathsf{S}_i(\theta_i, d_i)$, $\mathsf{S}_{i+1}(\theta_{i+1}, d_{i+1})$	$d_i = 0$, $d_{i+1} = 0$, $\mathbf{s}_i \cdot \mathbf{s}_{i+1} = 0, (\mathbf{s}_i \cdot \mathbf{s}_{i+1})^0 = 0$
S	$\mathsf{S}_i(\theta_i, d_i)$, $\mathsf{S}_{i+1}(\theta_{i+1}, d_{i+1})$, $\mathsf{S}_{i+1}(\theta_{i+2}, d_{i+2})$	$d_i = 0$, $d_{i+1} = 0$, $d_{i+2} = 0$ $\mathbf{s}_i = \mathbf{x}$, $\mathbf{s}_{i+1} = \mathbf{y}$, $\mathbf{s}_{i+2} = \mathbf{z}$ $(\mathbf{s}_i \cdot \mathbf{s}_{i+1})^0 = 0$, $(\mathbf{s}_i \cdot \mathbf{s}_{i+2})^0 = 0$, $(\mathbf{s}_{i+1} \cdot \mathbf{s}_{i+2})^0 = 0$

Equate the kinematics equations to the set of task positions,

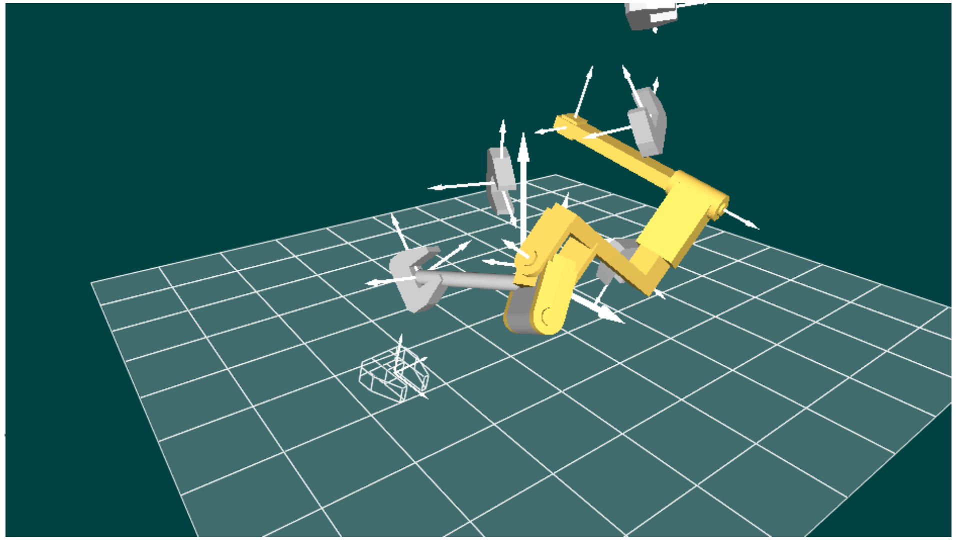
$$\hat{KE}_m^j = \hat{P}^j, \quad j = 2, \ldots, n.$$

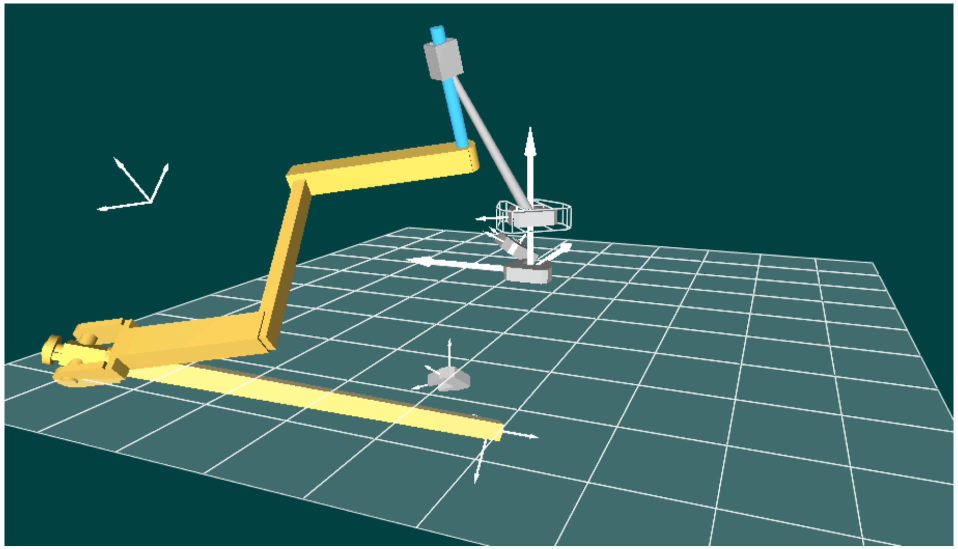
to solve numerically for the coordinates of the joint axes, Si, and the values of the joint variables to reach each task position.

Synthetica 2.0: General Synthesizer

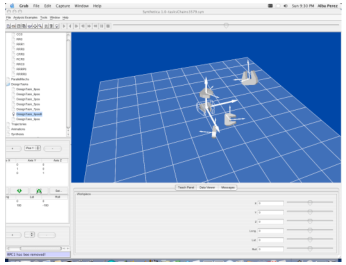
Some examples:

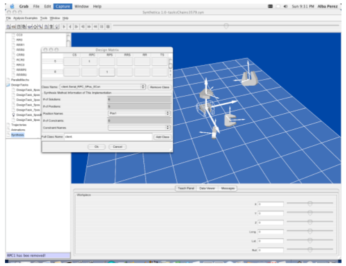
- RR, 3 positions: 83msec
- RRR, 5 positions: 1.55sec
- TC, 6 positions: 245 msec
- TRP, 7 positions: 4.73sec
- RCC, 7 positions: 2.26sec
- RPRR, 8 positions: 2.69sec
- SC, 8 positions: 6.76sec
- RTR, 8 positions: 5.20sec
- RRRR, 9 positions: 6.35sec.
- SPR, 10 positions: 49.72sec.

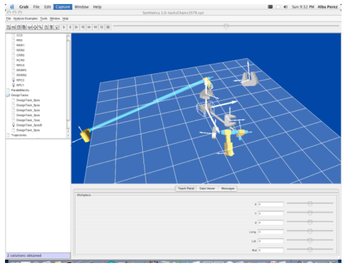


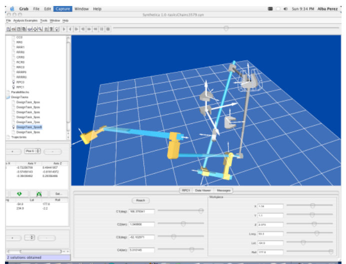


Synthetica 2.0: Specialized Synthesis Routines

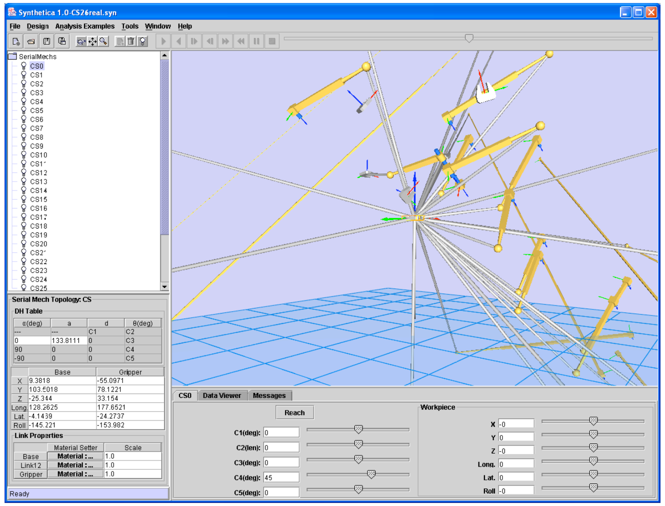






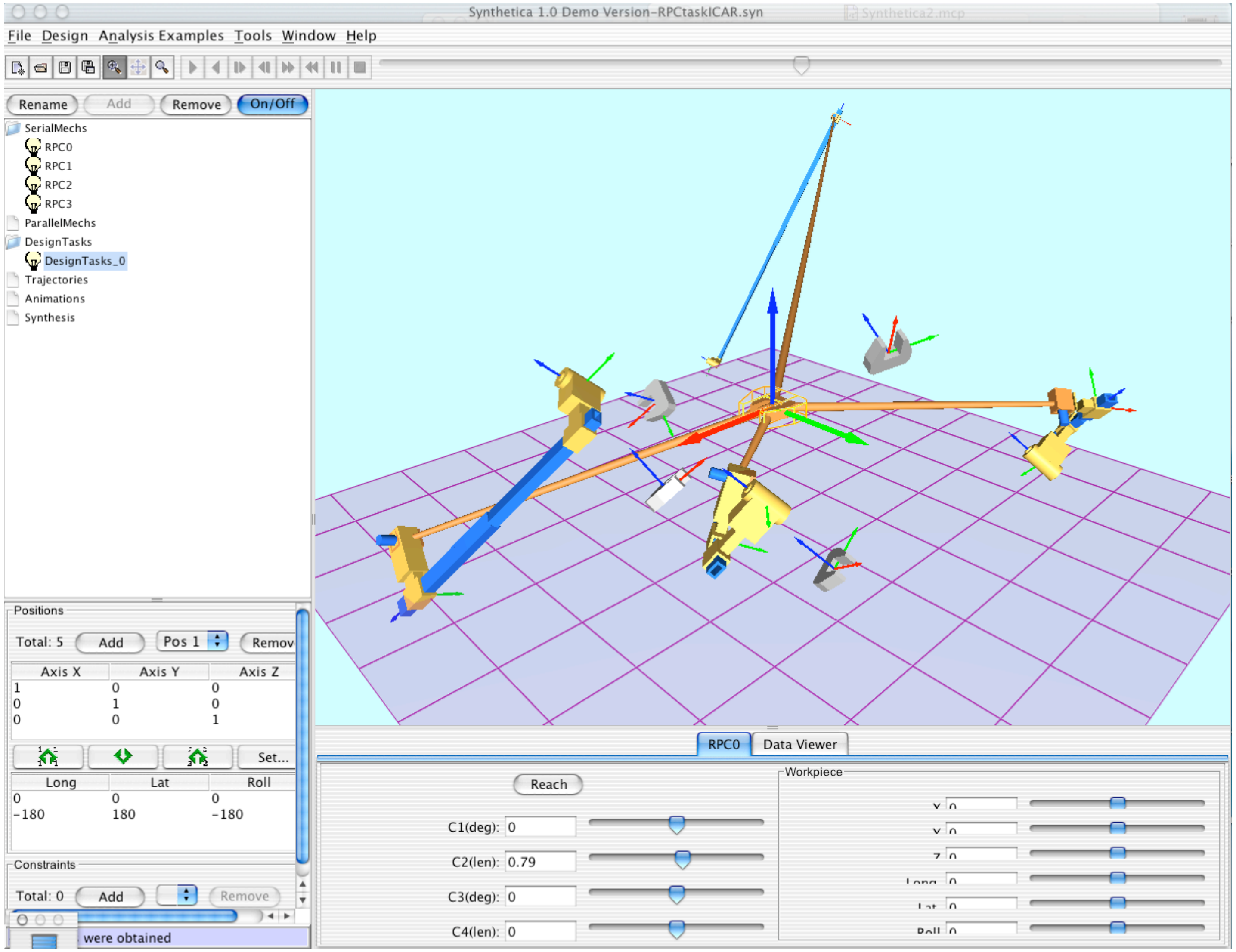


- Eight routines that solve specific topologies.
- Faster than numerical solver.
- Based on algebraic solutions, return all solutions for a given chain.
- New synthesis routines developed by external users can be integrated into the program.
- Implemented chains: RP-2, PR-2, RR-3, RPC-5, RP-2, PR-2, RR-3, RPC-5.



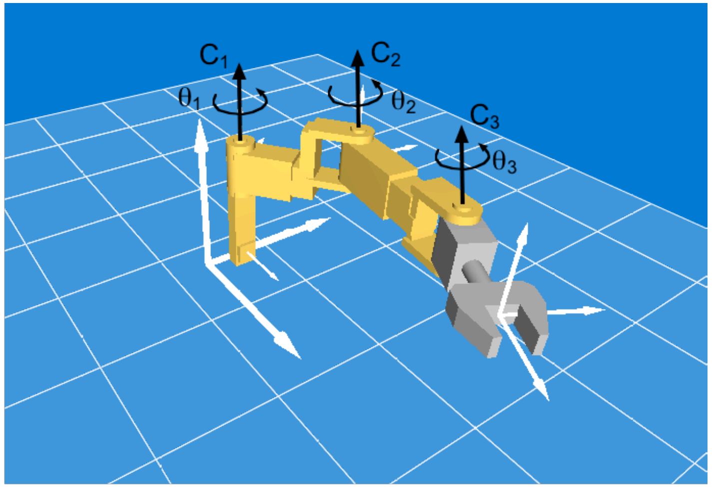
Synthetica 2.0: Specialized Synthesis Routines

Example: RPC Robot. Task definition: 5 positions. Result: 4 solutions.



Computer-aided Design of Robotic Systems:
Task optimization for general robots

- **Planar RRR Chain:**
  - 3 dof, it can reach any position within its extended configuration.
  - Impose conditions on the value of the joint angles to solve for a discrete task.



Chain	Pos.	Eqns.	Solution
RRR with fixed θ_i **(design with path planning)**	**3**	**4 translation 2 rotation**	**Fix the values of θ_1, θ_2 and the location of C_1. Solve linearly for θ_3, C_2, C_3.**
	5	**8 translation 4 rotation**	**Fix the values for θ_1 and the location of C_1. Solve for θ_2 ,θ_3, C_2, C_3.**
	7	**12 translation 6 rotation**	**Fix the values for θ_1, solve for θ_2 ,θ_3, C_1, C_2, and C_3.**
RRR with $\theta_i = k_i \theta_1$ **(coupled single dof chains)**	**n**	**2(n-1) translation (n-1) rotation**	**Solve for θ_1 in the rotation equations, solve linearly for (n-1) pivots C_i.**
	4	**6 translation 3 rotation**	**Solve for θ_1 in the rotation equations, solve linearly for 3 pivots C_1, C_2, C_3.**

Identification of kinematic structures:

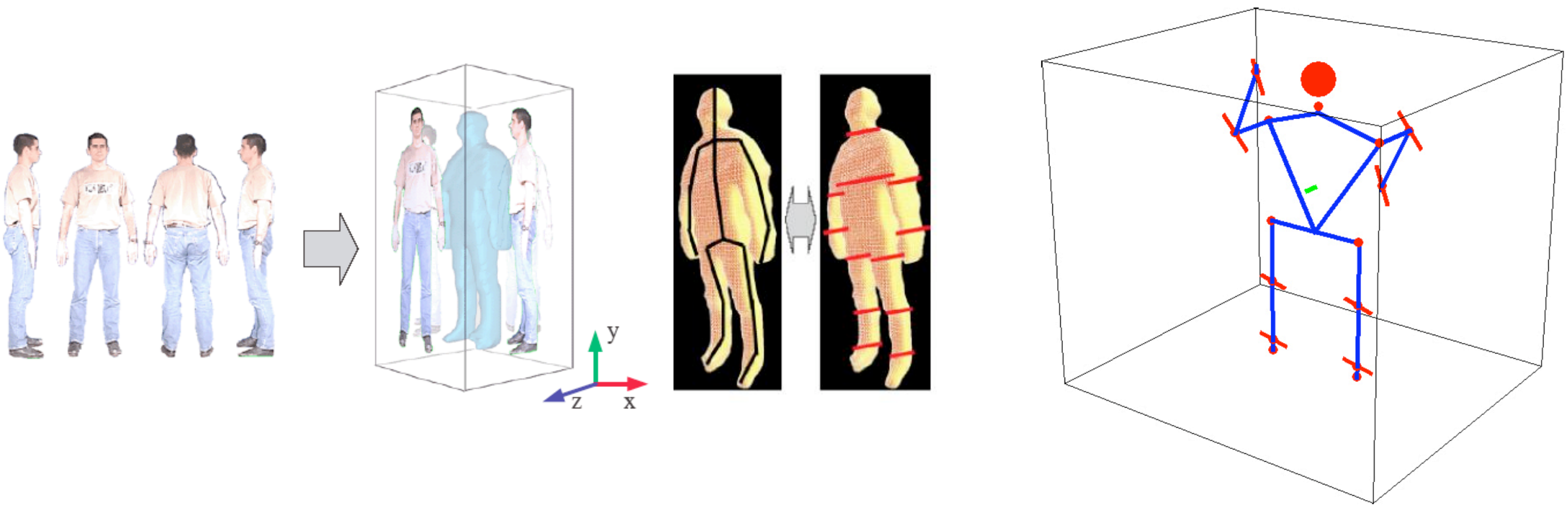
Generation of kinematic skeletons for avatars

(Collaboration with Dr Villa-Uriol, then at the Department of Computer Engineering, Univ. of California, Irvine)

When reconstructing 3-dimensional avatars from images (pictures or video), a procedure is needed to identify the kinematic structure necessary for pose estimation, tracking and movement generation.

- **Standard approach:** thinning algorithms based on finding the center of mass from a visual hull. Captures the three-dimensional structure but does not have information about the movement of the subject

- **Kinematic approach:** Synthesizing a skeleton composed of articulated rigid bodies joined by joints. It gives a compact representation of the movement of the avatar.



Kinematic Synthesis of the Avatar Skeleton

- **Goal:** To obtain a model for the human body that approximates human motion in a compact way.
- **Skeleton topology:** five serial chains with revolute and spherical joints

- **Forward kinematics:**
  - *Head:* S chain

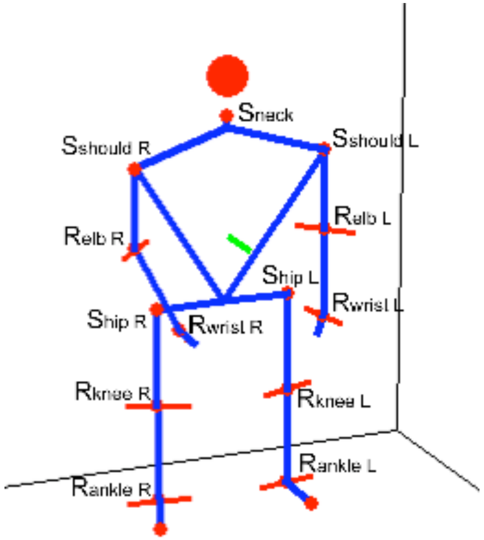
$$\hat{Q}_{head} = \hat{S}_{neck}(\theta_{h1}, \theta_{h2}, \theta_{h3}),$$

  - *Left Arm*
  - *Right Arm:* SRR chains

$$\hat{Q}_{arm} = \hat{S}_{should}(\theta_{a1}, \theta_{a2}, \theta_{a3})\hat{R}_{elb}(\theta_{a4})\hat{R}_{wrist}(\theta_{a5}),$$

  - *Left Leg*
  - *Right Leg:* SRR chains

$$\hat{Q}_{leg} = \hat{S}_{hip}(\theta_{l1}, \theta_{l2}, \theta_{l3})\hat{R}_{knee}(\theta_{l4})\hat{R}_{ankle}(\theta_{l5}).$$



Kinematic Synthesis of the Avatar Skeleton

- **Dual quaternion expression of the kinematics equations**
  - Revolute joint
  - (elbow, wrist, knee, ankle)

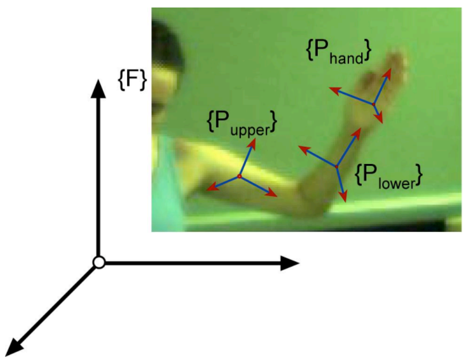
$$\hat{S}(\theta) = \begin{Bmatrix} \sin\frac{\theta}{2} s_x \\ \sin\frac{\theta}{2} s_y \\ \sin\frac{\theta}{2} s_z \\ \cos\frac{\theta}{2} \end{Bmatrix} + \epsilon \begin{Bmatrix} \sin\frac{\theta}{2} s_x^0 \\ \sin\frac{\theta}{2} s_y^0 \\ \sin\frac{\theta}{2} s_z^0 \\ 0 \end{Bmatrix}.$$

  - Spherical joint
  - (neck, shoulder, hip)

$$\hat{S}(\theta_1, \theta_2, \theta_3) = \begin{Bmatrix} \alpha_1 \mathbf{g}_1 + \alpha_2 \mathbf{g}_2 + \alpha_3 \mathbf{g}_3 \\ \alpha_4 \end{Bmatrix} + \epsilon \begin{Bmatrix} \alpha_1 \mathbf{g}_1^0 + \alpha_2 \mathbf{g}_2^0 + \alpha_3 \mathbf{g}_3^0 \\ 0 \end{Bmatrix}$$

- **Limb data**
  - Attaching a moving frame to each limb and for each snapshot, we obtain the set of positions used to dimension the skeleton,

P^j_{upper}, P^j_{lower}, P^j_{hand} for arms and legs, *j=1,..,n*
P^i_{head} for head, *i=1,..,m*.



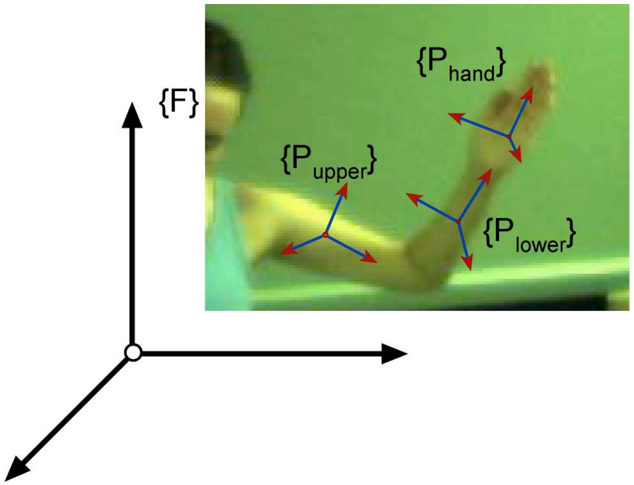
Kinematic Synthesis of the Avatar Skeleton

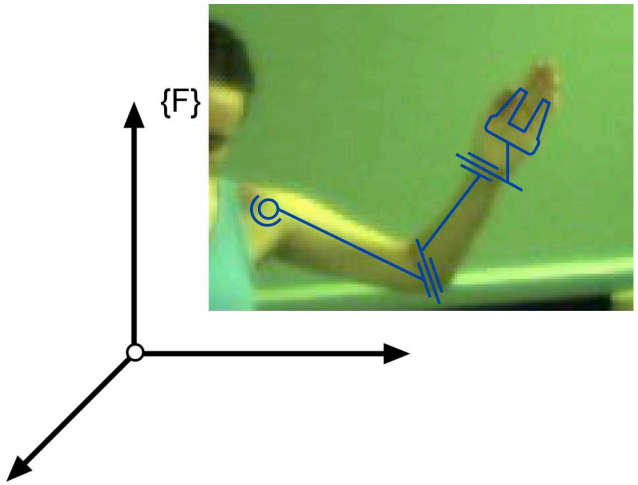
- **Adjustment of skeleton to limb data**
  - Equate the kinematics equations to experimental limb data

$$\hat{Q}_{head} = \hat{P}^{j}_{head}, \quad j = 2, \dots n,$$
$$\hat{Q}_{arm} = \hat{P}^{i}_{arm}, \quad \hat{Q}_{leg} = \hat{P}^{i}_{leg}, \quad i = 2, \dots m,$$

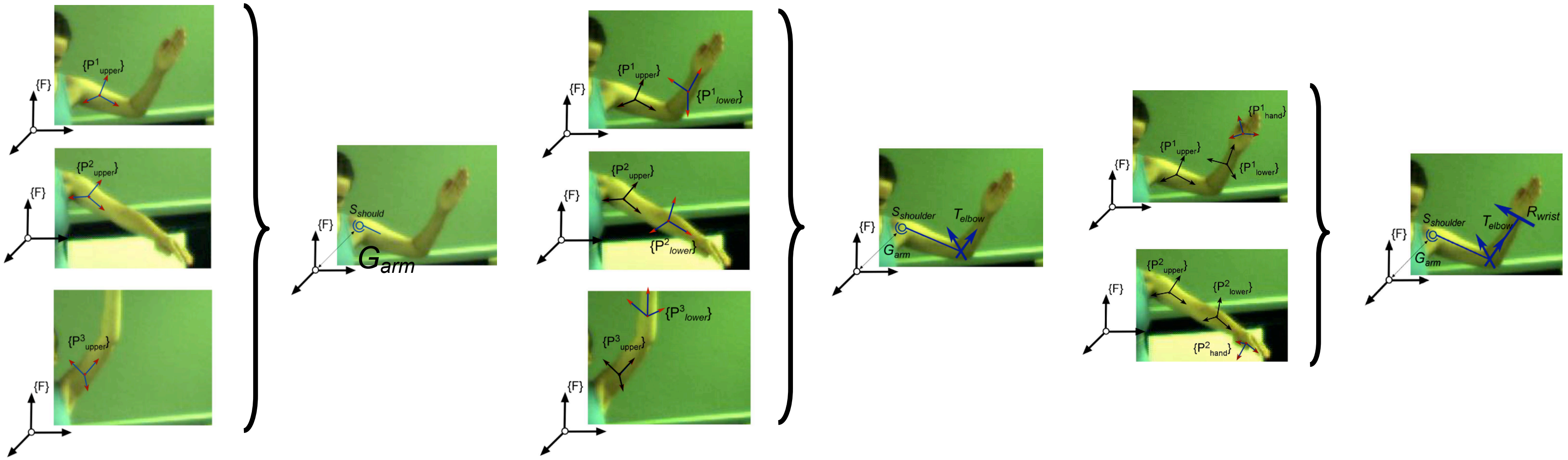
  - Minimize for the joint parameters (location and orientation) and the joint variables

- **Hierarchical solution process**
  - We have information for each link of the skeleton ➡ we can solve sequentially for each joint.





Kinematic Synthesis of the Avatar Skeleton



$\hat{S}_{shoulder}(\theta^i_{a1},\theta^i_{a2},\theta^i_{a3}) = \hat{G}^{i*}_{arm}\hat{P}^i_{upper}, \quad i=1,\dots n$ $\hat{T}_{elbow}(\theta^i_{a4},\theta^i_{a5}) = \hat{P}^{i*}_{upper}\hat{P}^i_{lower}, \quad i=1,\dots,n$ $\hat{R}_{wrist}(\theta^i_{a6}) = P^{i*}_{lower}\hat{P}^i_{hand}, \quad i=1,\dots,n$

3 input data frames required

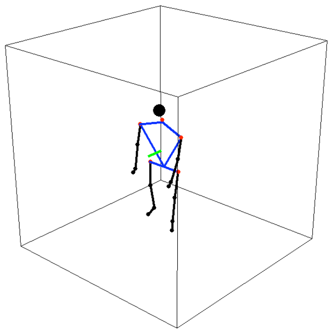
3 input data frames required

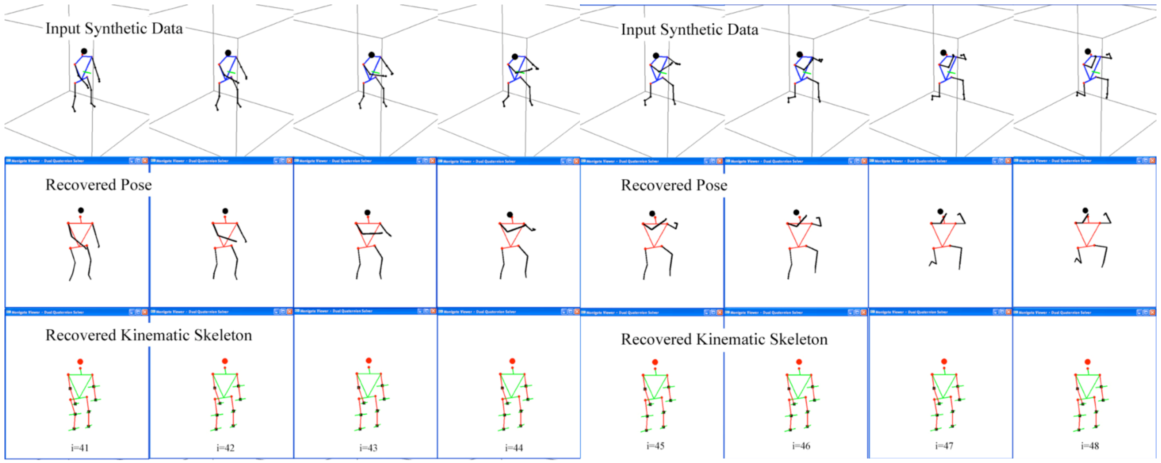
2 input data frames required

Repeat the process for each of the five serial chains. We obtain the Plücker coordinates of the joints and the joint variables to reach the first pose of the subject.

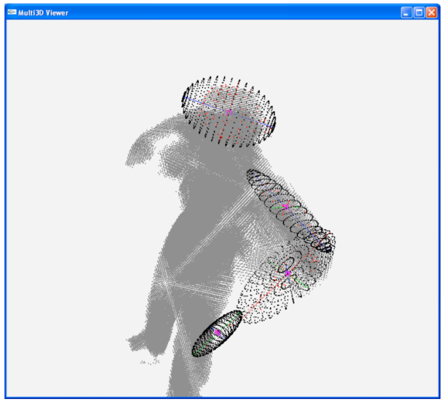
Kinematic Synthesis of the Avatar Skeleton
Experimental Results

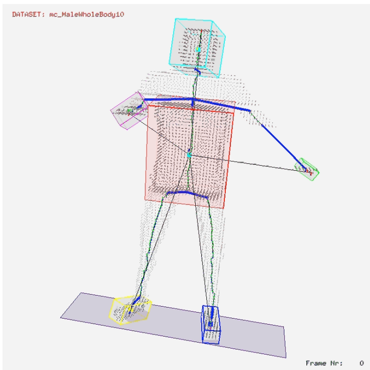
- **Algorithm testing:** synthetically generated frames. 15 frames were needed to obtain a good approximation (error 5e-3)





- **Real video data:** The dataset from raw visual images is very noisy. When motion capture images are used, the results are good.



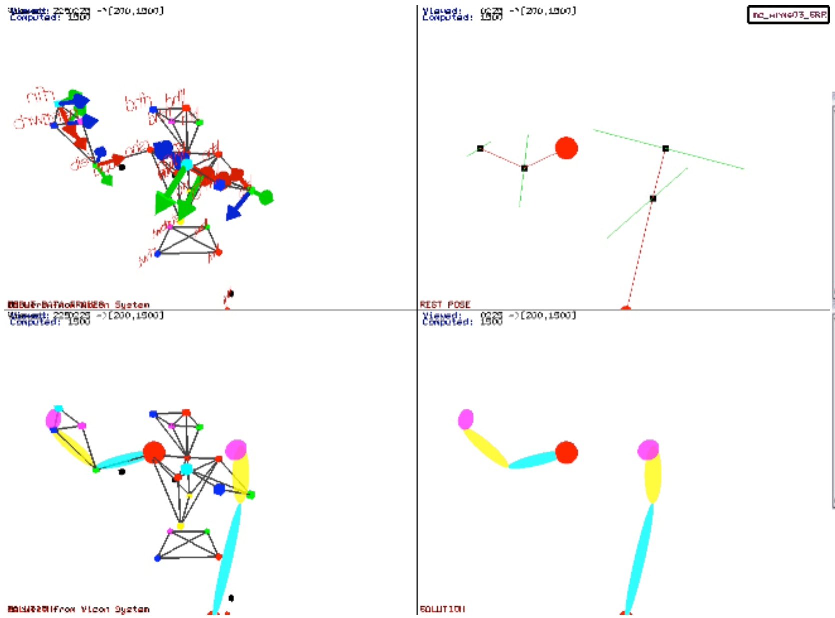


Kinematic Synthesis of the Avatar Skeleton
Experimental Results

- **Real video data:** The dataset from raw visual images is very noisy. When motion capture images are used, the results are good.

- Several iterations are needed for including all possible joint movements.

- Once the skeleton is defined, inverse kinematics is trivial. Criteria for termination of synthesis algorithm is needed.

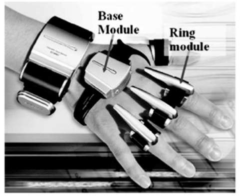


Hand Motion Identification

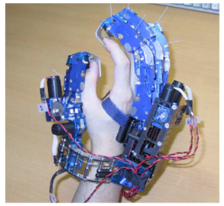
Other Approaches

- Glove-based devices
- Exoskeleton devices

Ekvall and Kragic, 2005



Kim, Soh and Lee, 2005



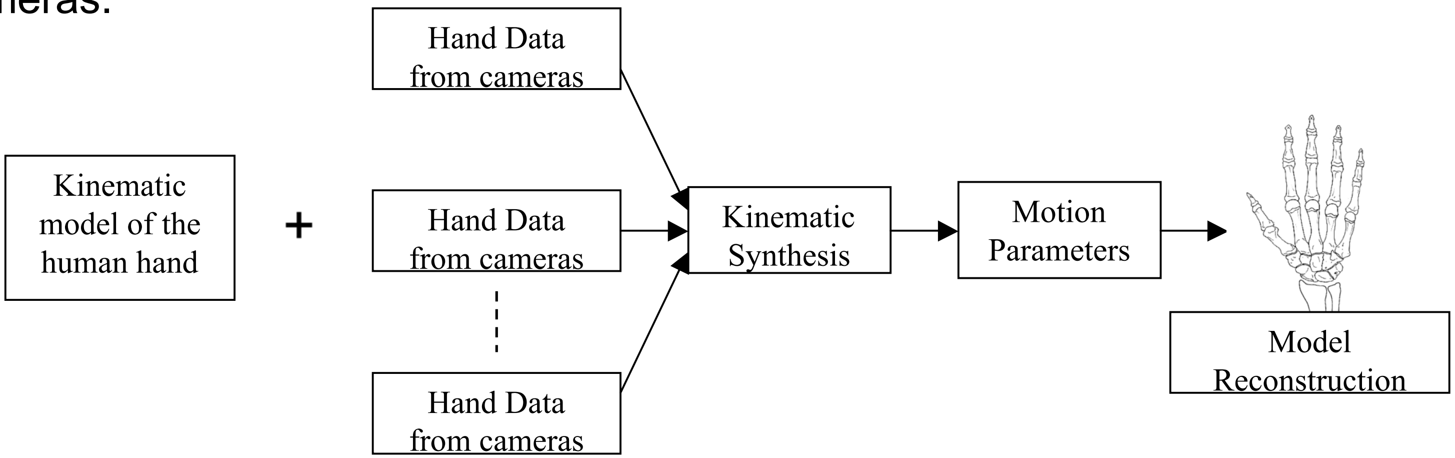
Nakawagara et al., 2005

- Motion capture methods:
  - Geometrical hand-shape models (motion library, thinning, solid fitting)
  - Pose estimation using kinematic hand models (full rotation at each joint)

Kinematic Synthesis Approach

- Use kinematic synthesis to define the skeleton structure and the joint angles for a given motion.
- 3-D motion obtained from a set of video cameras.

→ {
Non-intrusive system
Relatively cheap
Applicable to analyze other moving systems
Adaptable to human variability



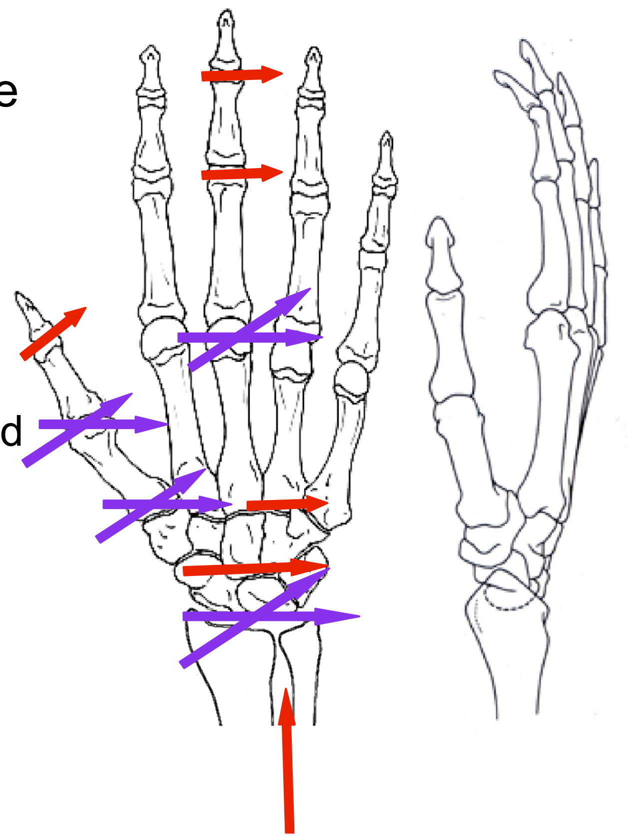
Kinematic Model of the Hand

Anatomy of the Human Hand: Degrees of Freedom

- Joints are formed at the surface of relative motion between two bones
- An accurate description of the relative motion uses the geometry and conjugation of the rubbing surfaces

- Classification of hand joints:
  - 1 DOF joints (rotation about an axes)
  - 2 DOF joints (rotation about 2 axes)
  - Immobile elements: Distal carpal bones, CMC, index and middle finger.

- Each finger is considered as a serial chain
  - Degrees of Freedom: each finger has 5 D.O.F, with 3 additional common D.O.F.



Kinematic Model of the Hand

Dual Quaternion Kinematic Model

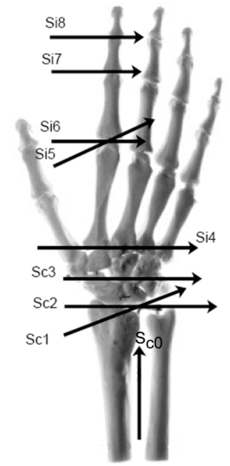
Index, Middle, Ring and Little Finger

$$\hat{Q}_{ind} = \hat{S}_{c0}(\theta_0)\hat{T}_{c12}(\theta_1,\theta_2)\hat{S}_{c3}(\theta_3)\hat{T}_{i56}(\theta_5,\theta_6)\hat{S}_{i7}(\theta_7)\hat{S}_{i8}(\theta_8)$$

$$\hat{Q}_{mid} = \hat{S}_{c0}(\theta_0)\hat{T}_{c12}(\theta_1,\theta_2)\hat{S}_{c3}(\theta_3)\hat{T}_{m56}(\theta_5,\theta_6)\hat{S}_{m7}(\theta_7)\hat{S}_{m8}(\theta_8)$$

$$\hat{Q}_{thir} = \hat{S}_{c0}(\theta_0)\hat{T}_{c12}(\theta_1,\theta_2)\hat{S}_{c3}(\theta_3)\hat{S}_{t4}(\theta_4)\hat{T}_{t56}(\theta_5,\theta_6)\hat{S}_{t7}(\theta_7)\,\hat{S}_{t8}(\theta_8)$$

$$\hat{Q}_{four} = \hat{S}_{c0}(\theta_0)\hat{T}_{c12}(\theta_1,\theta_2)\hat{S}_{c3}(\theta_3)\hat{S}_{f4}(\theta_4)\hat{T}_{f56}(\theta_5,\theta_6)\hat{S}_{f7}(\theta_7)\,\hat{S}_{f8}(\theta_8)$$

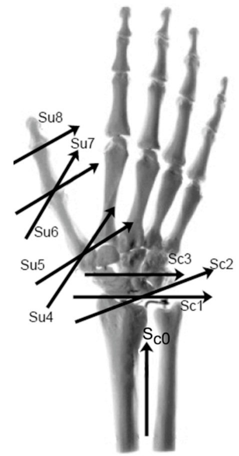


Thumb Finger

$$\hat{Q}_{thum} = \hat{S}_{c0}(\theta_0)\hat{T}_{c12}(\theta_1,\theta_2)\hat{S}_{c3}(\theta_3)\hat{T}_{t45}(\theta_4,\theta_5)\hat{T}_{t67}(\theta_6,\theta_7)\hat{S}_{t8}(\theta_8).$$

Total model complexity

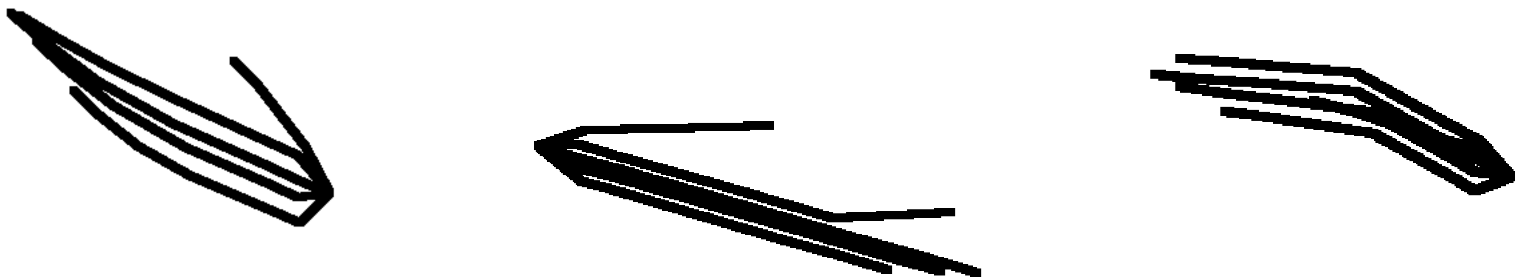
- Our hand model has 14 revolute joints, with 4 structural parameters each and 7 universal joints, with 6 structural parameters each.
- 28 joint variables (rotations qi),
- 98 structural parameters to define the joint axes Si.



Kinematic Model of the Hand

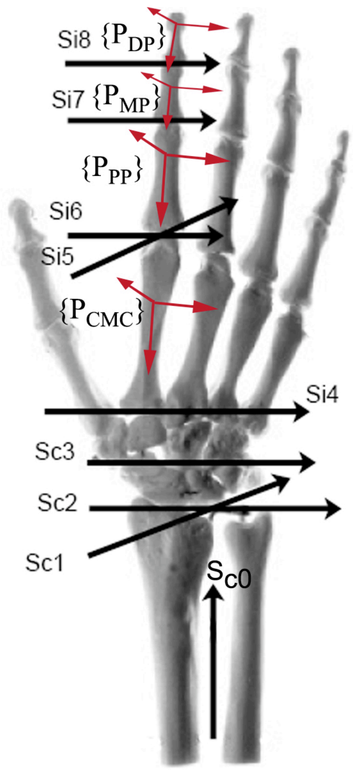
Motion Simulation

- Hand parameters taken from anatomy literature.
- Define joint trajectories for the 27 joint variables.
- Model was implemented in Maple to create kinematic simulation.
- Accurate geometry (angles and distances between joints) need not be correct; to be refined with synthesis process.



Kinematic Synthesis of the Human Hand

- For this application, we require precise values for the location of the joints and the joint rotations.
- Kinematic synthesis can be applied to dimension the human hand and to track its motion.
- The displacement of each hand limb can be expressed as
  - Lower arm P_{LA}
  - Palm, carpal area P_{CA}
  - Palm, carpo-metacarpal area P_{CMC}
  - Proximal phalanx P_{PP},
  - etc.
- The simultaneous solution for all joints in each ray presents some problems:
  - Complexity (33 different positions, 264 nonlinear equations)
  - Multiple solutions
- Instead, we solve for each joint in a hierarchical process.



Kinematic Synthesis of the Human Hand

Hierarchical Process

Solve for the proximal radio-ulnar joint with the motion of the lower arm

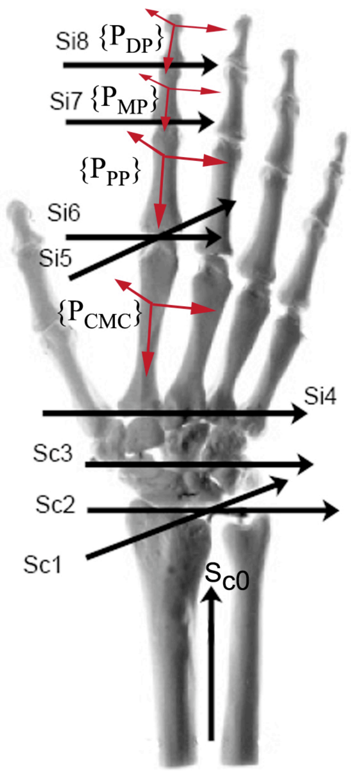
$$\hat{S}_{c0}(\theta_0^i) = \hat{P}_{LA}^i, \quad i = 2, \ldots n$$

→

Solve for the distal radio-ulnar joint and carpal joint using the motion of the palm carpal area

$$\hat{S}_{c0}(\theta_0^i)\hat{T}_{c12}(\theta_1^i, \theta_2^i)\hat{S}_{c3}(\theta_3^i) = \hat{P}_{CA}^i$$

$$\hat{T}_{c12}(\theta_1^i, \theta_2^i)\hat{S}_{c3}(\theta_3^i) = (\hat{P}_{LA}^i)^* \hat{P}_{CA}^i, \quad i = 2, \ldots n$$



Thumb

CMC joint (motion: thumb metacarpal area)
⇩
MCP joint (motion: proximal phalanx)
⇩
PP joint (motion: proximal phalanx)
⇩
DP joint (motion: distal phalanx)

Index, middle

MCP joint (motion: proximal phalanx)
⇩
PP joint (motion: proximal phalanx)
⇩
MP joint (motion: middle phalanx)
⇩
DP joint (motion: distal phalanx)

Third, fourth

CMC joint (motion: palm metacarpal area)
⇩
MCP joint (motion: proximal phalanx)
⇩
PP joint (motion: proximal phalanx)
⇩
MP joint (motion: middle phalanx)
⇩
DP joint (motion: distal phalanx)

Kinematic Synthesis of the Human Hand

Numerical Solver

- The numerical solution consists on the minimization of the distance between the real motion and the motion performed by the simplified joint,

$$\epsilon_i = \sqrt{\sum_{j=1}^{8}(\hat{Q}^i_j - \hat{P}^i_j)^2}$$

- We use a Levenberg-Marquardt nonlinear least squares solver implemented in C++. (http://www.ics.forth.gr/~lourakis/levmar , Manolis Lourakis, Institute of Computer Science, Foundation for Research and Technology-Hellas, Heraklion, Crete).

- For each joint, we have to solve a nonlinear system of equations. Tests show that the equations converge to a solution quickly.

```
cg = -0.1460830286e0 + s0x * sin (theta02 / 0.2e1);
cg0 = s0y * sin (theta02 / 0.2e1);
cg1 = s0z * sin (theta02 / 0.2e1);
cg2 = -0.9892723330e0 + cos (theta02 / 0.2e1);
cg3 = -0.2990407923e0 + s0x * sin (theta03 / 0.2e1);
cg4 = s0y * sin (theta03 / 0.2e1);
cg5 = s0z * sin (theta03 / 0.2e1);
cg6 = -0.9542403285e0 + cos (theta03 / 0.2e1);
cg7 = -0.4909037534e0 + s0x * sin (theta04 / 0.2e1);
cg8 = s0y * sin (theta04 / 0.2e1);
cg9 = s0z * sin (theta04 / 0.2e1);
cg10 = -0.8712138112e0 + cos (theta04 / 0.2e1);
cg11 = (c0y * s0z - c0z * s0y) * sin (theta02 / 0.2e1);
cg12 = (c0z * s0x - c0x * s0z) * sin (theta02 / 0.2e1);
cg13 = (c0x * s0y - c0y * s0x) * sin (theta02 / 0.2e1);
cg14 = (c0y * s0z - c0z * s0y) * sin (theta03 / 0.2e1);
cg15 = (c0z * s0x - c0x * s0z) * sin (theta03 / 0.2e1);
cg16 = (c0x * s0y - c0y * s0x) * sin (theta03 / 0.2e1);
cg17 = (c0y * s0z - c0z * s0y) * sin (theta04 / 0.2e1);
cg18 = (c0z * s0x - c0x * s0z) * sin (theta04 / 0.2e1);
cg19 = (c0x * s0y - c0y * s0x) * sin (theta04 / 0.2e1);
```

→

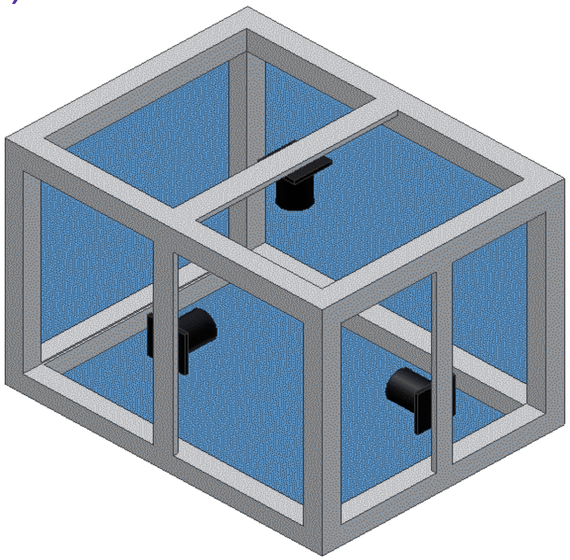
Results, middle finger joints **(3.45 GHz processor)**:

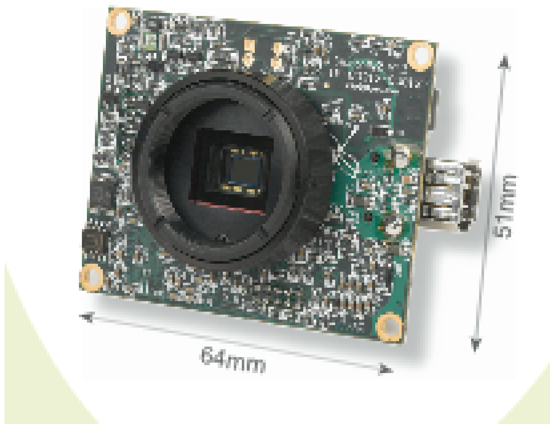
Axes	Error	Iteration	Time (sec)
S_{c0}	1.72259e-21	18	0.004
S_{c1} S_{c2}	2.00422e-19	247	0.012
S_{m4}	5.95462e-19	41	0.002
S_{m5} S_{m6}	2.59657e-18	634	0.024
S_{m7}	2.90753e-17	71	0.004
S_{m8}	7.91677e-17	51	0.002

Kinematic Synthesis of the Human Hand

Image acquisition: experimental setup (under construction)

- Actual motion capture takes place in a box of approx. dimensions 1mx1mx1m approx., equipped with three cameras.
- Cameras: Dragonfly2, 648x488 resolution, 60 fps.
- NI LabVIEW software for image acquisition and analysis.





Kinematic Synthesis of the Human Hand

Challenges of the image acquisition and processing

- 3-D model calculation from 2-D camera images.
- Motion segmentation (identify rigid bodies): Background subtraction, occultations, deformations, etc.
- Object classification (identify which rigid body corresponds to which limb)
- Tracking (identify the same rigid body in consecutive frames)
- We are dealing with more than 20 “rigid bodies” in the hand, some of them with very subtle motion and with visible deformation.
- For obtaining good results using kinematic synthesis, we need to be able to isolate the rigid motion associated to the skeleton.

Synthesis strategy

- Newton-based solver very fast when close to a solution.
- Usually, inverse kinematic solutions are not very close to each other.
- Use hierarchical synthesis to obtain an approximate solution
- Use the solution from the hierarchical synthesis as initial conditions for solving each finger as a complete RTRR or TTR serial chain.
- Input data: an easy feature of each finger (fingertip).

Conclusions

- The finite-position synthesis for general topologies is still an open problem.
- Equating the relative transformation of the chain to the set of desired relative transformations yields a general method to formulate design equations.
- Expressing the displacements using the Clifford algebra of dual quaternions helps in reducing the complexity of the equations.
- For serial chains with four or more joints, complexity may be a real problem (independently of the method used to state the equations).
- Efficient implementation in CAD software requires:
    - Solutions for yet unsolved problems.
    - User interface strategies for specifying spatial linkage tasks.
    - Numerical algorithms to select among solutions according to other criteria.
- Application to the identification of kinematic structures from visual data:
    - Solution is very dependent on the quality of the data.
    - The method is being tested for precise identification of anatomy and joint motion.