Proceedings of the 5th Applied Mechanisms and Robotics Conference, Cincinnati, OH, October 12-15, 1997,
pp. AMR97-046-001/010

Development of a Software Library of Functions for Analysis of Spatial Mechanisms

Harry H. Cheng, Associate Professor
Pablo Gonzalez, Graduate Research Assistant
Integration Engineering Laboratory
Department of Mechanical and Aeronautical Engineering
University of California
Davis, CA 95616
Email: hhcheng@ucdavis.edu

Abstract

An integrated method for symbolical derivation of equations and numerical computations using dual numbers for analysis of spatial mechanisms is presented in this paper. The formulation is based upon 3x3 dual transformation matrices and derived symbolically using software package Mathematica. Based on the solution procedure presented in this paper, a software library of functions for displacement analysis of spatial mechanisms has been developed. Functions in this software library can be readily used in the C^H language environment where dual number is treated as a first-class object. Displacement analysis of the RCRCR spatial mechanism is used as an example to illustrate the solution procedure and programming details.

1 Introduction

Spatial mechanisms are widely used in various industrial applications. Displacement analysis of spatial mechanisms can be performed using a variety of displacement operators or coordinate transformation forms such as 3x3 dual matrices (Soni and Harrisberger, 1968; Yang, 1969; Ravani and Roth, 1984; Pennock and Yang, 1985; McCarthy, 1990), 4x4 Denavit-Hartenberg (DH) matrices (Denavit and Hartenberg, 1955), 4x4 screw matrices (Suh and Radcliffe, 1978), 6x6 line coordinate transformation matrices (Yuan, 1970), 2x2 dual matrices (Denavit, 1958), dual quaternions (Hamilton, 1899; Yang and Freudenstein, 1964), etc. A 3x3 dual transformation matrix is orthogonal and similar to a 3x3 pure rotational matrix with real elements. Because of their compact nature, the 3x3 dual matrices are widely used for analytical treatment of spatial mechanisms. However, programming with dual numbers is cumbersome in traditional programming languages. When numerical computations are needed, dual formulas may have to be partitioned into separate real and dual parts. The partition process is tedious and error-prone, the conciseness of dual formulas diminishes.

We have developed the C^H language environment (Cheng, 1993, 1997). C^H is a superset of ANSI C with incorporation of all programming features of FORTRAN. Because dual number is treated as a first-class object in the language, a dual formula can be translated into a C^H programming statement almost verbatim. We have demonstrated that numerical implementation of dual formulas for analysis of spatial mechanisms in the C^H language environment is remarkably simple (Cheng, 1994). In this paper, we will show how symbolic manipulation software package Mathematica (Wolfram, 1991) can be integrated with the C^H language environment for derivation of dual equations for analysis of spatial mechanisms. The output from a Mathematica program can be readily used as a function file in the C^H language environment. Based on this technique, we have developed a software library of functions for analysis of spatial mechanisms. Automatic derivation of functions for displacement analysis of spatial mechanisms for computation in the C^H language environment will be illustrated by derivation of dual formulas for displacement analysis of the RCRCR spatial mechanism. For comparison study and illustrative purpose, formulas presented in (Cheng and Thompson, 1995) for analysis of the RCRCR mechanism will be derived symbolically using a Mathematica program in this paper.

2 Related Mathematica Commands for Symbolic Derivation of Equations for Analysis of Spatial Mechanisms

Important features related to symbolic derivation of equations for analysis of spatial mechanisms in Mathematica will be briefly described in this section. When Mathematica is started in interactive mode, the prompt "In[1]:=" will appear on the screen. When the user types in an expression, the output starting with the output prompt "Out[1]=" from the execution of the expression will be displayed. These input and output prompt allow us to distinguish an input expression and an output from execution of the expression. The numerical number inside a pair of brackets keeps track of the number of input in the session. In Mathematica, values of the prompt and expression on the right hand side of the equal sign is the same as shown in the following session of Mathematica command execution.

Function	Application
Expand[arg]	Expand an expression defined by arg.
ExpandAll[arg]	Expand a fraction defined by arg.
Together[arg]	Simplify a fraction defined by arg.
Simplify[arg]	Simplify the expression defined by arg.
Nominator[arg]	Get the nominator of a fraction defined by arg.
Denominator[arg]	Get the denominator of a fraction defined by arg.
Transpose[arg]	Take the transpose of an array defined by arg.
Inverse[arg]	Take the inverse of a square array defined by arg.
D[arg,x]	Take derivative of the expression defined by arg with respect to x.
Collect[arg,$x_1,\cdots,x_n$]	Collect terms defined by $x_1,\cdots,x_n$ of arg.
Solve[{arg_1==0,$\cdots$,arg_n==0},{$x_1,\cdots,x_n$}]	Equation solver that can solve arg_n expressions in x_n unknowns.
TrigReduce[arg]	Reduce a trigonometric expression defined by arg.
Coefficient[arg,x,n]	Collect terms defined by arg with respect to x^n.
CForm[arg]	Convert Mathematica expression defined by arg to C-language.
Splice["filename"]	Run external Mathematica program defined by filename.

TABLE 1: Mathematica functions for analysis of spatial mechanisms.

Operation	Definition
+, -, *, /, and .	Addition, subtraction, multiplication, division, and matrix multiplication operators.
{{ $arg_{11},\cdots,arg_{1n}$ },{ $\cdots$ },{$arg_{n1},\cdots,arg_{nn}$}}	Define an nxn matrix.
arg_1 . arg_2	Multiplication of two matrices arg_1 and arg_2.
arg[[n]]	Get nth element from a column vector arg.
arg[[n,m]]	Get element (n,m) from matrix arg.
arg /. a -> b	Replace a by b in the expression arg
arg >> filemname	Save output of expression arg in filename.
<< filename	Read in Mathematica filename.
arg /. a	Replace arg by a.

TABLE 2: Mathematica operations for analysis of spatial mechanisms.

In[1]:= a
Out[1]= a
In[2]:= b+c-d
Out[2]= b + c - d
In[3]:= In[2]+In[1]
Out[3]= a + b + c - d
In[4]:= Out[2]+In[1]
Out[4]= a + b + c - d

where symbolic values of output Out[3] and Out[4] are the same even though the input expressions are different. In addition to Mathematica prompt In[i] and Out[i], variables can also be used to keep symbolic values as shown in the following example.

In[1]:= eq1=a+b+c
Out[1]= a + b + c
In[2]:= eq1
Out[2]= a + b + c

where variable `eq1` in the above example can be used inside a Mathematica function or symbolic expression. In Mathematica, there are many functions for symbolic manipulation. They can be used for solving equations, finding roots of equation, manipulation of trigonometric expressions and matrices, etc. Tables 1 and 2 list major functions and operations that are useful for analysis of spatial mechanisms, respectively.

Mathematica can be used for symbolic derivation of dual equations for application of spatial mechanisms. For a dual number $\widehat{d} = x + \varepsilon y$, x is called the real part and y the dual part. A variable $\widehat{d} = x + \varepsilon y$ in Mathematica can be used to represent the dual number. The arithmetic operations such as addition, subtraction, multiplication, and division can be carried out as if x, y, and ε were real numbers. The function `Coefficient[ ]` then can be used to obtain real and dual parts of an expression for dual number. The first argument of function `Coefficient[ ]` is the result of a dual expression, the second argument the variable in the expression, and the third the power of the variable. Because $\varepsilon^2 = \varepsilon^3 = \varepsilon^4 = \cdots = 0$, the real and dual parts of the dual result can be obtained by functions `Coefficient[d,e,0]` and `Coefficient[d,e,1]`, respectively. Examples of symbolic manipulation of the dual operations for dual numbers are given in Table 3.

3 Symbolic Derivation of Equations for Analysis of Spatial Mechanisms

The detailed derivation of formulations for displacement analysis of the RCRCR spatial mechanism will be used to illustrate the technique of using Mathematica to symbolically derive equations for analysis of spatial mechanisms. For a cross comparison, the solution procedure similar to

Mathematica Command	Description
d1=x1+e*y1	Define dual number $\widehat{d_1}$
d2=x2+e*y2	Define dual number $\widehat{d_2}$
d3=x3+e*y3	Define dual number $\widehat{d_3}$
d=Expand[d1*d2*d3]	Multiplication of $\widehat{d_1}\widehat{d_2}\widehat{d_3}$
$x1*x2*x3+e*x2*x3*y1+e*x1*x3*y2+e^2*x3*y1*y2+e*x1*x2*y3+e^2*x2*y1*y3+e^2*x1*y2*y3+e^3*y1*y2*y3$	Mathematica output of d.
x=Coefficient[d,e,0]	Get real part of dual number d.
$x1*x2*x3$	Mathematica output of the real part of d.
y=Coefficient[d,e,1]	Get dual part of dual number d.
$x2*x3*y1+x1*x3*y2+x1*x2*y3$	Mathematica output of the dual part of d.
d=x+e*y	Redefine dual number d.

TABLE 3: Examples of symbolic manipulation of the dual operations for dual numbers.

one presented in (Cheng and Thompson, 1995) will be used. The Mathematica commands for symbolic derivation at each step will be given. A schematic diagram with Denavit-Hartenberg parameters for the adjacent links of the RCRCR spatial mechanism is shown in Figure 1. In Figure 1, dual angles $\widehat{\alpha}_i = \alpha_i + \varepsilon a_i$ $(i = 1,\ 2,\ 3,\ 4,\ 5\ )$ and d_1, d_3, d_5 are given link parameters. The revolute axes Z_1 and Z_5 are attached to the ground link. Coordinate systems $O_1X_1Y_1Z_1$ and $O_5X_5Y_5Z_5$ are attached to the input and output links, respectively. The joint variable θ_1 is the input parameter. The displacement analysis of this RCRCR mechanism is that, given link parameters $\widehat{\alpha}_i = \alpha_i + \varepsilon\alpha_i(i = 1,2,3,4,5)$ and d_1, d_3, d_5, determine the joint values $\theta_2, \theta_3, \theta_4, \theta_5, d_2, d_4$ in terms of the input angle θ_1. The formulations for displacement analysis of the RCRCR mechanism will be derived as follows. According to displacement transformation matrices, we can write the loop-closure equation for the RCRCR mechanism shown in Figure 1 as

$$\widehat{\mathbf{\Lambda}}_5\widehat{\mathbf{\Theta}}_5\widehat{\mathbf{\Lambda}}_4\widehat{\mathbf{\Theta}}_4\widehat{\mathbf{\Lambda}}_3\widehat{\mathbf{\Theta}}_3\widehat{\mathbf{\Lambda}}_2\widehat{\mathbf{\Theta}}_2\widehat{\mathbf{\Lambda}}_1\widehat{\mathbf{\Theta}}_1 = \mathbf{I} \tag{1}$$

where $\mathbf{I}$ is a 3×3 identity matrix, and dual transformation matrices $\widehat{\mathbf{\Theta}}_i$ and $\widehat{\mathbf{\Lambda}}_i$ are given as follows:

$$\widehat{\mathbf{\Theta}}_i = \begin{bmatrix} \cos\widehat{\theta}_i & -\sin\widehat{\theta}_i & 0 \\ \sin\widehat{\theta}_i & \cos\widehat{\theta}_i & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2}$$

$$\widehat{\mathbf{\Lambda}}_i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\widehat{\alpha}_i & -\sin\widehat{\alpha}_i \\ 0 & \sin\widehat{\alpha}_i & \cos\widehat{\alpha}_i \end{bmatrix}. \tag{3}$$

Equation (1) can be reformulated as

$$\widehat{\mathbf{\Lambda}}_1\widehat{\mathbf{\Theta}}_1\widehat{\mathbf{\Lambda}}_5\widehat{\mathbf{\Theta}}_5\widehat{\mathbf{\Lambda}}_4 - (\widehat{\mathbf{\Theta}}_4\widehat{\mathbf{\Lambda}}_3\widehat{\mathbf{\Theta}}_3\widehat{\mathbf{\Lambda}}_2\widehat{\mathbf{\Theta}}_2)^T = \mathbf{0} \tag{4}$$

The dual matrix equation (4) can be expanded symbolically using the following commands.

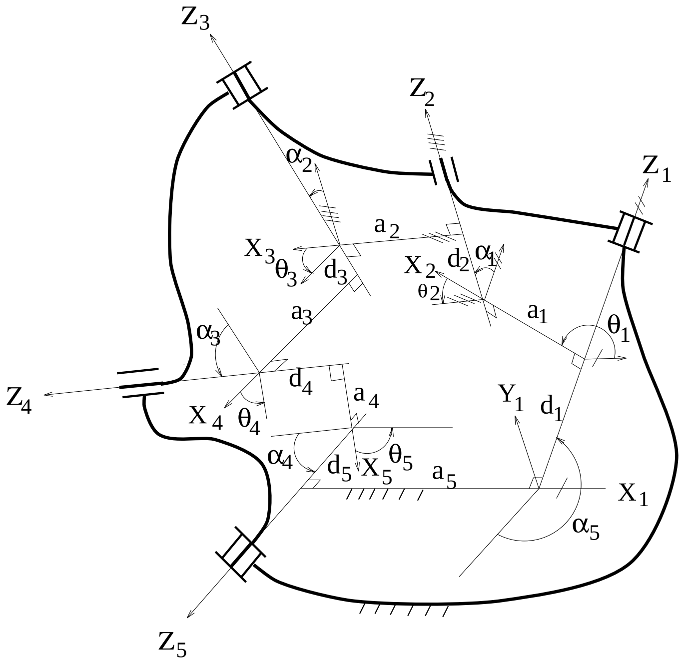

FIGURE 1: The RCRCR spatial mechanism.

Mathematica Command	Description
T1={{C1,-S1,0},{S1,C1,0},{0,0,1}}	equation (2)
T2={{C2,-S2,0},{S2,C2,0},{0,0,1}}	
T3={{C3,-S3,0},{S3,C3,0},{0,0,1}}	
T4={{C4,-S4,0},{S4,C4,0},{0,0,1}}	
T5={{C5,-S5,0},{S5,C5,0},{0,0,1}}	
A1={{1,0,0},{0,Ca1,-Sa1},{0,Sa1,Ca1}}	equation (3)
A2={{1,0,0},{0,Ca2,-Sa2},{0,Sa2,Ca2}}	
A3={{1,0,0},{0,Ca3,-Sa3},{0,Sa3,Ca3}}	
A4={{1,0,0},{0,Ca4,-Sa4},{0,Sa4,Ca4}}	
A5={{1,0,0},{0,Ca5,-Sa5},{0,Sa5,Ca5}}	
IO =A1.T1.A5.T5.A4 - Transpose[T4.A3.T3.A2.T2]	equation (4)
eq1=Expand[IO[[1,1]]]	element (1,1)
eq2=Expand[IO[[1,2]]]	element (1,2)
eq3=Expand[IO[[1,3]]]	element (1,3)
eq4=Expand[IO[[2,1]]]	element (2,1)
eq5=Expand[IO[[2,2]]]	element (2,2)
eq6=Expand[IO[[2,3]]]	element (2,3)
eq7=Expand[IO[[3,1]]]	element (3,1)
eq8=Expand[IO[[3,2]]]	element (3,2)
eq9=Expand[IO[[3,3]]]	element (3,3)

Variable `eq9` in the above Mathematica commands, corresponding to element (3,3) in equation (4), gives the following input-output relation of the mechanism

$$C\widehat{\alpha}_2 C\widehat{\alpha}_3 - C\widehat{\alpha}_1 C\widehat{\alpha}_4 C\widehat{\alpha}_5 - C\widehat{\theta}_3 S\widehat{\alpha}_2 S\widehat{\alpha}_3 + C\widehat{\alpha}_5 C\widehat{\theta}_1 C\widehat{\theta}_5 S\widehat{\alpha}_1 S\widehat{\alpha}_4 + C\widehat{\alpha}_4 C\widehat{\theta}_1 S\widehat{\alpha}_1 S\widehat{\alpha}_5 + C\widehat{\alpha}_1 C\widehat{\theta}_5 S\widehat{\alpha}_4 S\widehat{\alpha}_5 - S\widehat{\alpha}_1 S\widehat{\alpha}_4 S\widehat{\theta}_1 S\widehat{\theta}_5 = 0 \quad (5)$$

where $S\widehat{\theta}_i$, $C\widehat{\theta}_i$, $S\widehat{\alpha}_i$, and $C\widehat{\alpha}_i$ represent $\sin(\widehat{\theta}_i)$, $\cos(\widehat{\theta}_i)$, $\sin(\widehat{\alpha}_i)$, and $\cos(\widehat{\alpha}_i)$, respectively. To isolate unknown variables θ_3 and θ_5, dual equation (5) can be reformulated as

$$\widehat{A} + \widehat{B}(C\widehat{\theta}_3) + \widehat{C}(S\widehat{\theta}_5) + \widehat{D}(C\widehat{\theta}_5) = 0 \quad (6)$$

with dual coefficients $\widehat{A}, \widehat{B}, \widehat{C}, \widehat{D}$ as follows:

$$\widehat{A} = -C\widehat{\alpha}_2 C\widehat{\alpha}_3 + C\widehat{\alpha}_1 C\widehat{\alpha}_4 C\widehat{\alpha}_5 - C\widehat{\alpha}_4 C\widehat{\theta}_1 S\widehat{\alpha}_1 S\widehat{\alpha}_5 \quad (7)$$

$$\widehat{B} = S\widehat{\alpha}_2 S\widehat{\alpha}_3 \quad (8)$$

$$\widehat{C} = S\widehat{\alpha}_1 S\widehat{\alpha}_4 S\widehat{\theta}_1 \quad (9)$$

$$\widehat{D} = -C\widehat{\alpha}_5 C\widehat{\theta}_1 S\widehat{\alpha}_1 S\widehat{\alpha}_4 - C\widehat{\alpha}_1 S\widehat{\alpha}_4 S\widehat{\alpha}_5 \quad (10)$$

The above dual coefficients can be derived symbolically by the following Mathematica commands.

Mathematica Command
BBB=CForm[Coefficient[eq9,C3,1]]
CCC=CForm[Coefficient[eq9,S5,1]]
DDD=CForm[Coefficient[eq9,C5,1]]
AAA=CForm[Expand[eq9-BBB*C3-CCC*S5-DDD*C5]]

Some capital letters in Mathematica have special meanings. For example, variables C and D in Mathematica are protected variables and I is used as imaginary part of a complex number. Therefore, variables with three repeated capital letters are used in our Mathematica program. Variables AAA, BBB, CCC, and DDD in the above Mathematica commands correspond to $\widehat{A}$, $\widehat{B}$, $\widehat{C}$, and $\widehat{D}$ in equations (7) to (10), respectively. The corresponding dual variables AA, BB, CC, and DD will be used in a C^H program which will be discussed in the next section.

In order to solve for the unknown variables θ_3 and θ_5, equation (6) needs to be separated into the real and dual parts with the following two equations with real numbers only.

$$\acute{A} + \acute{B}(C\theta_3) + \acute{C}(S\theta_5) + \acute{D}(C\theta_5) = 0 \quad (11)$$

$$\grave{A} + \grave{B}(C\theta_3) - \acute{B}d_3(S\theta_3) + \grave{C}(S\theta_5) + \acute{C}d_5(C\theta_5) + \grave{D}(C\theta_5) - \acute{D}d_5(S\theta_5) = 0 \quad (12)$$

In the above formulation, a dual variable topped with an accent grave “ˊ” stands for the real part of the dual number, whereas a dual variable topped with an accent acute “ˋ” stands for the dual part. For example, $\acute{A}$ and $\grave{A}$ stands for real and dual parts of $\widehat{A}$, respectively. Equations (11) and (12) can be derived by the following Mathematica commands.

Mathematica Command
AA=Ax+e*Ay
BB=Bx+e*By
CC=Cx+e*Cy
DD=Dx+e*Dy
CC3=c3-e*d3*s3
SS5=s5+e*d5*c5
CC5=c5-e*d5*s5
eq9=Expand[AA+BB*CC3+CC*SS5+DD*CC5]
Req9=Coefficient[eq9,e,0]
Deq9=Coefficient[eq9,e,1]

where variables SSi and CCi represent the dual values of $S\widehat{\theta}_i$ and $C\widehat{\theta}_i$, respectively. Variable eq9 in Mathematica contains the dual expression for element (3,3) in dual equation (6). Variables Req9 and Deq9, corresponding to equations (11) and (12), give the real and dual parts of dual equation (6), respectively. We introduce an additional parameter $\chi = \tan(\theta_5/2)$ to express both $\sin\theta_5$ and $\cos\theta_5$ in terms of χ as follows.

$$S\theta_5 = \frac{2\chi}{1+\chi^2}, \quad C\theta_5 = \frac{1-\chi^2}{1+\chi^2} \quad (13)$$

The results of $\sin(\theta_3)$ and $\cos(\theta_3)$ then can be expressed in terms of the unknown parameter χ and other constants in the following equations.

$$S\theta_3 = \frac{-\grave{C}\acute{B}(2\chi) - \acute{C}\acute{B}d_5(1-\chi^2) - \grave{D}\acute{B}(1-\chi^2) + \acute{x}_4\acute{B}d_5(2\chi)}{-(1+\chi^2)\acute{B}d_3} + \frac{-\grave{A}\acute{B}(1+\chi^2) + \grave{B}\acute{C}(2\chi) + \grave{B}\acute{D}(1-\chi^2) + \grave{B}\acute{A}(1+\chi^2)}{-(1+\chi^2)\acute{B}d_3} \quad (14)$$

$$C\theta_3 = \frac{-\acute{C}(2\chi) - \acute{D}(1-\chi^2) - \acute{A}(1+\chi^2)}{\acute{B}(1+\chi^2)} \quad (15)$$

These results can be derived symbolically using the following Mathematica commands.

Mathematica Command
temp=Solve[Req9==0,c3]
c3=c3 /. temp; c3=c3[[1]]
temp=Solve[Deq9==0,s3]
s3=s3 /. temp; s3=s3[[1]]
s5=2*X/(1+X*X)
c5=(1-X*X)/(1+X*X)
s3=Together[Expand[s3]]*(1+X*X)
c3=Together[Expand[c3]]*(1+X*X)

Because the identity equation $\sin^2\theta_3 + \cos^2\theta_3 = 1$ will be used to form a polynomial equation in terms of variable χ, both variables c3 and s3 for $\sin\theta_3$ and $\cos\theta_3$, respectively, are multiplied by the common denominator $(1+\chi^2)$ to simplify the derivation of the symbolic equation. Equations (14) and (15) can be reformulated as

$$S\theta_3 = \frac{E\chi^2 + F\chi + G}{(1+\chi^2)} \quad (16)$$

$$C\theta_3 = \frac{H\chi^2 + I\chi + J}{(1+\chi^2)} \quad (17)$$

where coefficients E, F, G, H, I and J are defined as follows.

$$E = \frac{-\grave{B}\acute{A} + \grave{A}\acute{B} - \grave{D}\acute{B} - d_5\acute{B}\acute{C} + \grave{B}\acute{D}}{d_3\acute{B}^2} \quad (18)$$

$$F = \frac{2(\grave{C}\acute{B} - \grave{B}\acute{C} - d_5\acute{B}\acute{D})}{d_3\acute{B}^2} \quad (19)$$

$$G = \frac{-\grave{B}\acute{A} + \grave{A}\acute{B} + \grave{D}\acute{B} + d_5\acute{B}\acute{C} - \grave{B}\acute{D}}{d_3\acute{B}^2} \quad (20)$$

$$H = \frac{-\acute{A} + \acute{D}}{\acute{B}} \quad (21)$$

$$I = \frac{-2\acute{C}}{\acute{B}} \quad (22)$$

$$J = \frac{-\acute{A} - \acute{D}}{\acute{B}} \quad (23)$$

These coefficients can be obtained by the following Mathematica commands.

Mathematica Command
EEE=CForm[Together[Coefficient[s3,X,2]]]
FFF=CForm[Together[Coefficient[s3,X,1]]]
GGG=CForm[Together[Coefficient[s3,X,0]]]
HHH=CForm[Together[Coefficient[c3,X,2]]]
III=CForm[Together[Coefficient[c3,X,1]]]
JJJ=CForm[Together[Coefficient[c3,X,0]]]

Substitute $S\theta_3$ and $C\theta_3$ in equations (14) and (15), respectively, into the identity equation

$$\sin\theta_3^2 + \cos\theta_3^2 - 1 = 0, \tag{24}$$

the following fourth-order polynomial equation for the RCRCR mechanism can be obtained.

$$poly = p_4\chi^4 + p_3\chi^3 + p_2\chi^2 + p_1\chi + p_0 = 0 \tag{25}$$

where the coefficients of the polynomial are defined as

$$p_4 = -1 + E^2 + H^2 \tag{26}$$
$$p_3 = 2(EF + HI) \tag{27}$$
$$p_2 = -2 + F^2 + 2EG + 2HJ + I^2 \tag{28}$$
$$p_1 = 2(FG + JI) \tag{29}$$
$$p_0 = -1 + J^2 + G^2 \tag{30}$$

These coefficients can be derived symbolically by the following Mathematica commands.

Mathematica Command
s3=EE*X*X+FF*X+GG
c3=HH*X*X+II*X+JJ
poly=Together[Expand[c3*c3+s3*s3-((1+X*X)*(1+X*X))]]
p4=CForm[Together[Coefficient[poly,X,4]]]
p3=CForm[Together[Coefficient[poly,X,3]]]
p2=CForm[Together[Coefficient[poly,X,2]]]
p1=CForm[Together[Coefficient[poly,X,1]]]
p0=CForm[Together[Coefficient[poly,X,0]]]

Variables $c3$ and $s3$ in the above Mathematica commands have been multiplied with $(1+\chi^2)$ in the previous symbolic derivation, which is equivalent to multiplication of $(1+\chi^2)^2$ in equation (24). Polynomial equation (25) can be solved using a numerical solution. And θ_5 then can be determined from the equation

$$\theta_5 = 2\mathrm{atan}\,(\chi)\,. \tag{31}$$

Because equation (25) is a fourth-order polynomial equation, there are up to four real solutions for χ for a given input angle θ_1. Once θ_5 is obtained, θ_3 can be calculated by using the following formula

$$\theta_3 = 2\tan^{-1}\left(\frac{S\theta_3}{1 + C\theta_3}\right) \tag{32}$$

where $S\theta_3$ and $C\theta_3$ are given in equations (16) and (17), respectively.

Once values for θ_5 and θ_3 are obtained, we can use the other elements of the dual matrix equation (4) to find the remaining unknowns $\widehat{\theta}_2$ and $\widehat{\theta}_4$. The elements (1,3) and (2,3) corresponding to Mathematica variables eq3 and eq6 in the previous derivation can be derived as follows.

$$\begin{aligned} eq3 \;=\;& C\widehat{\alpha}_5 C\widehat{\theta}_5 S\widehat{\alpha}_4 S\widehat{\theta}_1 + C\widehat{\alpha}_4 S\widehat{\alpha}_5 S\widehat{\theta}_1 - \\ & C\widehat{\alpha}_3 S\widehat{\alpha}_2 S\widehat{\theta}_2 - C\widehat{\alpha}_2 C\widehat{\theta}_3 S\widehat{\alpha}_3 S\widehat{\theta}_2 - \\ & C\widehat{\theta}_2 S\widehat{\alpha}_3 S\widehat{\theta}_3 + C\widehat{\theta}_1 S\widehat{\alpha}_4 S\widehat{\theta}_5 = 0 \end{aligned} \tag{33}$$

$$\begin{aligned} eq6 \;=\;& C\widehat{\alpha}_4 C\widehat{\alpha}_5 S\widehat{\alpha}_1 + C\widehat{\alpha}_3 C\widehat{\theta}_2 S\widehat{\alpha}_2 + C\widehat{\alpha}_2 C\widehat{\theta}_2 C\widehat{\theta}_3 S\widehat{\alpha}_3 + \\ & C\widehat{\alpha}_1 C\widehat{\alpha}_5 C\widehat{\theta}_1 C\widehat{\theta}_5 S\widehat{\alpha}_4 + C\widehat{\alpha}_1 C\widehat{\alpha}_4 C\widehat{\theta}_1 S\widehat{\alpha}_5 - \\ & C\widehat{\theta}_5 S\widehat{\alpha}_1 S\widehat{\alpha}_4 S\widehat{\alpha}_5 - S\widehat{\alpha}_3 S\widehat{\theta}_2 S\widehat{\theta}_3 - \\ & C\widehat{\alpha}_1 S\widehat{\alpha}_4 S\widehat{\theta}_1 S\widehat{\theta}_5 = 0 \end{aligned} \tag{34}$$

where $S\widehat{\theta}_i$, $C\widehat{\theta}_i$, $S\widehat{\alpha}_i$, and $C\widehat{\alpha}_i$ represent $\sin(\widehat{\theta}_i)$, $\cos(\widehat{\theta}_i)$, $\sin(\widehat{\alpha}_i)$, and $\cos(\widehat{\alpha}_i)$, respectively. Equations (34) and (38) can be rearranged as follows.

$$\widehat{C}_1(S\widehat{\theta}_2) + \widehat{D}_1(C\widehat{\theta}_2) = \widehat{A}_1 \tag{35}$$
$$\widehat{E}_1(S\widehat{\theta}_2) + \widehat{F}_1(C\widehat{\theta}_2) = \widehat{B}_1 \tag{36}$$

where dual constants $\widehat{A}_1$, $\widehat{B}_1$, $\widehat{C}_1$, $\widehat{D}_1$ $\widehat{E}_1$, and $\widehat{F}_1$ can be derived by the following Mathematica commands.

Mathematica Command
temp=Coefficient[eq3,S2,0]; AA1=-Coefficient[temp,C2,0]
DD1=Coefficient[eq3,S2,1]; CC1=Coefficient[eq3,C4,1]
temp=Coefficient[eq6,S2,0]; BB1=-Coefficient[temp,C2,0]
EE1=Coefficient[eq6,S2,1]; FF1=Coefficient[eq6,C4,1]

where variables AA1, BB1, CC1, DD1, EE1, and FF1 correspond to dual variables $\widehat{A}_1$, $\widehat{B}_1$, $\widehat{C}_1$, $\widehat{D}_1$, $\widehat{E}_1$, and$\widehat{F}_1$, respectively. The dual angle θ_2 then can be calculated by the formula

$$\begin{aligned} \widehat{\theta}_2 &= 2\tan^{-1}\left(\frac{\sin\widehat{\theta}_2}{\cos\widehat{\theta}_2 + 1}\right) \\ &= 2\tan^{-1}\left(\frac{\widehat{A}_1\widehat{F}_1 - \widehat{B}_1\widehat{C}_1}{\widehat{D}_1\widehat{F}_1 - \widehat{C}_1\widehat{E}_1 + \widehat{B}_1\widehat{D}_1 - \widehat{A}_1\widehat{E}_1}\right) \end{aligned} \tag{37}$$

Similarly, the solution for $\widehat{\theta}_4$ can be obtained using the elements (3,1) and (3,2) of equation (4), corresponding to Mathematica variables eq7 and eq8, respectively, as follows:

$$\begin{aligned} eq7 \;=\;& C\widehat{\theta}_5 S\widehat{\alpha}_1 S\widehat{\theta}_1 - C\widehat{\theta}_4 S\widehat{\alpha}_2 S\widehat{\theta}_3 - \\ & C\widehat{\alpha}_3 C\widehat{\theta}_3 S\widehat{\alpha}_2 S\widehat{\theta}_4 - C\widehat{\alpha}_2 S\widehat{\alpha}_3 S\widehat{\theta}_4 + \\ & C\widehat{\alpha}_5 C\widehat{\theta}_1 S\widehat{\alpha}_1 S\widehat{\theta}_5 + C\widehat{\alpha}_1 S\widehat{\alpha}_5 S\widehat{\theta}_5 = 0 \end{aligned} \tag{38}$$

$$\begin{aligned} eq8 \;=\;& C\widehat{\alpha}_4 C\widehat{\alpha}_5 C\widehat{\theta}_1 C\widehat{\theta}_5 S\widehat{\alpha}_1 + C\widehat{\alpha}_3 C\widehat{\theta}_3 C\widehat{\theta}_4 S\widehat{\alpha}_2 + \\ & C\widehat{\alpha}_2 C\widehat{\theta}_4 S\widehat{\alpha}_3 + C\widehat{\alpha}_1 C\widehat{\alpha}_5 S\widehat{\alpha}_4 + \\ & C\widehat{\alpha}_1 C\widehat{\alpha}_4 C\widehat{\theta}_5 S\widehat{\alpha}_5 - C\widehat{\theta}_1 S\widehat{\alpha}_1 S\widehat{\alpha}_4 S\widehat{\alpha}_5 - \\ & S\widehat{\alpha}_2 S\widehat{\theta}_3 S\widehat{\theta}_4 - C\widehat{\alpha}_4 S\widehat{\alpha}_1 S\widehat{\theta}_1 S\widehat{\theta}_5 = 0 \end{aligned} \tag{39}$$

Equations (38) and (39) can be simplified as

$$\widehat{C}_2(S\widehat{\theta}_4) + \widehat{D}_2(C\widehat{\theta}_4) = \widehat{A}_2 \tag{40}$$
$$\widehat{E}_2(S\widehat{\theta}_4) + \widehat{F}_2(C\widehat{\theta}_4) = \widehat{B}_2 \tag{41}$$

where dual coefficients $\widehat{A}_2$, $\widehat{B}_2$, $\widehat{C}_2$, $\widehat{D}_2$ $\widehat{E}_2$, and $\widehat{F}_2$ can be derived by the following Mathematica commands.

Mathematica Command
temp=Coefficient[eq7,S4,0]; AA2=-Coefficient[temp,C4,0]
DD2=Coefficient[eq7,S4,1]; CC2=Coefficient[eq7,C4,1]
temp=Coefficient[eq8,S4,0]; BB2=-Coefficient[temp,C4,0]
EE2=Coefficient[eq8,S4,1]; FF2=Coefficient[eq8,C4,1]

where variables AA2, BB2, CC2, DD2, EE2, and FF2 correspond to dual variables $\widehat{A}_2$, $\widehat{B}_2$, $\widehat{C}_2$, $\widehat{D}_2$, $\widehat{E}_2$, and $\widehat{F}_2$, respectively. The dual angle θ_4 can then be calculated by the formula

$$\widehat{\theta}_4 = 2\tan^{-1}\left(\frac{\widehat{A}_2\widehat{F}_2 - \widehat{B}_2\widehat{C}_2}{\widehat{D}_2\widehat{F}_2 - \widehat{C}_2\widehat{E}_2 + \widehat{B}_2\widehat{D}_2 - \widehat{A}_2\widehat{E}_2}\right) \tag{42}$$

4 Automatic Generation of Functions for Analysis of Spatial Mechanisms

In the previous section, we have derived formulations for displacement analysis of the RCRCR spatial mechanism using the Mathematica commands. In this section, we will demonstrate how these Mathematica commands can be integrated to automatically generata a displacement analysis function for numerical computation in the C^H language environment.

For the RCRCR mechanism, function RCRCR() shall have the following function prototype.

```
void RCRCR(double dual a[5],double d[3],
     double dual theta1, double dual theta2[4],
     theta3[4],theta4[4],theta5[4]);
```

The first argument of the function is an array of five elements of double dual data type for the D-H parameter $\widehat{a}_i$. The second argument is an array of three elements of double data type for the given D-H parameters d_1, d_3, and d_5. The third argument is for $\widehat{\theta}_1$. The remaining arguments of arrays of four elements of double dual data type are for unknown joint variables $\widehat{\theta}_i$ for i from 2 to 5. When the function RCRCR() is called, the set of four solutions for calculated joint variables will be passed to the calling function through these arrays of double dual data type. For a given input value θ_1, there are up to four set of solutions. An array of four dual elements is used for each joint variable θ_i. A program with file extension `.ff` is treated a function file in C^H. For a function file, the file name and function name inside the program shall be the same. A function in the function file is treated as if it were a built-in function like sin() in C^H. For displacement analysis of the RCRCR mechanism in C^H, all we need is the function file RCRCR.ff.

The function file RCRCR.ff can be generated automatically using the Mathematica program shown in Appendix A. The delimiters `(*` and `*)` are used to comment out section of code in a Mathematica program. A variable of a Mathematica program inside delimiters `<*` and `*>` will be replaced with its symbolic expression when the program is executed. Like in C, delimiters `/*` and `*/` are used for comments in C^H. For cross comparison, equation numbers for formulas presented in the previous section are given as comments at the end of the corresponding programming statements in RCRCR.mff. When program RCRCR.mff in Appendix A is executed by the following command at a shell prompt, a function file named RCRCR.ff shown in Appendix B will be created.

```
math -run 'Splice["RCRCR.mff"]; Quit[0];'
```

The command `math` will invoke the Mathematica, and option `-run` indicates that the specified Mathematica commands `Splice[]` and `Quit[]` will be executed. The command `Splice["RCRCR.mff"]` will run program RCRCR.mff and generate the output in an external function file RCRCR.ff with file extension `.ff` for C^H, whereas command `Quit[0]` will quit the Mathematica environment.

The procedure described in the above presentation for creation of function RCRCR() can be applied to derivation of other spatial mechanisms. For example, based upon this solution procedure, we have symbolically derived formulations and created C^H functions RCCC(), RCCRR(), RCRCP(), RPCCR(), RPPPRRR(), RPRCPR(), RPRPRPR(), RRPCC(), RRPRPPR(), RCPCR(), RPCPRR(), RPPRRC(), RPRPCR(), RPRRPC(), RRPPPRR(), RRRCC(), RCPPRR(), RCRPPR(), RPCRC(), RPPRRPR(), RPRPPRR(), RRCPC(), RRPPRC(), RRRPPC(), RCPRC(), RCRRC(), RPPCRR(), RPRCC(), RPRPRC(), RRCRC(), and RRPRPC() for analysis of these spatial mechanisms (Gonzalez, 1996). Additional functions for analysis of other spatial mechanisms can be derived and added to the software library.

5 An Example of Using Functions in the Software Library for Analysis of Spatial Mechanisms

In the previous section, we have presented the programming techniques in Mathematica for automatically generating a software library of functions for displacement analysis of spatial mechanisms. In this section, we will demonstrate how this library of functions can be conveniently used in the C^H language environment by solving the following displacement analysis problem for the RCRCR spatial mechanism:

> Given the following link parameters for an RCRCR mechanism shown in Figure 1: $\widehat{\alpha}_1 = 30°+\varepsilon 1$ cm, $\widehat{\alpha}_2 = 35°+\varepsilon 4$ cm, $\widehat{\alpha}_3 = 45°+\varepsilon 3$ cm, $\widehat{\alpha}_4 = 60° + \varepsilon 2.5$ cm, $\widehat{\alpha}_5 = 10° + \varepsilon 3.2$ cm, $d_1 = 0, d_3 = 2.5$ cm, and $d_5 = 3$ cm, find all possible solutions for joint variables $\theta_2, d_2, \theta_3, \theta_4, d_4$, and θ_5 when θ_1 is 180 degree.

This problem can be easily solved using a C^H program shown in Program 1. Function RCRCR() is shown in Appendix B. In function RCRCR(), the Denavit-Hartenberg parameters d_i, $\widehat{\alpha}_i$, and $\widehat{\theta}_i$ are represented as variables d[i], a[i], and theta*i*, respectively. For a given input value θ_1, there are up

Program 1: C^H program RCRCR.

```
#include <stdio.h>
#include <mechanism3D.h>
int main() {
  int i;
  double RAD2DEG=180/PI, DEG2RAD=PI/180;
  double d[3]={0,2.5,3}; /* D-H parameters */
  double dual theta1,theta2[4],theta3[4],theta4[4],
      theta5[4], a[5]={dual(30*DEG2RAD,1),
      dual(35*DEG2RAD,4), dual(45*DEG2RAD,3),
      dual(60*DEG2RAD,2.5), dual(10*DEG2RAD,3.2)};

  theta1=dual(PI,d[0]); /* 180 degree for theta1 */
  RCRCR(a,d,theta1,theta2,theta3,theta4,theta5);
  printf("%8s %8s %8s %8s %8s %8s %8s\n","theta1",
    "theta2","d2","theta3","theta4","d4","theta5");
  for (i = 0; i<4; i++)
    printf("%8i %8.2f %8.2f %8.2f %8.2f %8.2f %8.2f\n",
      180, real(theta2[i])*RAD2DEG, dual(theta2[i]),
      real(theta3[i])*RAD2DEG, real(theta4[i])*RAD2DEG,
      dual(theta4[i]), real(theta5[i])*RAD2DEG);
}
```

TABLE 4: The output from execution of program RCRCR.

theta1	theta2	d2	theta3	theta4	d4	theta5
180	96.89	-7.87	-115.76	-135.89	5.73	-15.34
180	21.15	-2.84	-107.31	-112.39	0.22	39.04
180	-0.77	-7.43	99.70	111.06	0.28	-53.13
180	-146.42	-2.14	82.78	161.18	-6.81	78.63

to four set of solutions. The result for each joint variable θ_i for i from 2 to 5 is passed back from function RCRCR() through an argument of array of four dual elements. The macro PI for π is defined in header file `mechanism3D.h`. Many macros for calculation of sine and cosine functions, such as S1 for sin(T1) and C1 for cos(T1), are also defined in `mechanism3D.h`. When Program 1 is executed, the output is given in Table 4. There are four sets of solutions for dependent joint variables $\theta_2, d_2, \theta_3, \theta_4, d_4$, and θ_5 when θ_1 is 180 degree.

6 Conclusions

We have developed a software library of functions for displacement analysis of spatial mechanisms. Functions in this library can be readily used in the C^H language environment. C^H is a superset of C and it treats dual number as a first-class object, which significantly simplifies the formulation for analysis of spatial mechanisms using dual numbers. The symbolic derivation details for function RCRCR() in the library for displacement analysis of the RCRCR spatial mechanism have been presented in this paper. Based on this solution procedure, additional functions for analysis of other spatial mechanisms can be derived and added to the library. As shown by an example of displacement analysis of the RCRCR spatial mechanism, analysis of spatial mechanisms using this software library is extremely simple.

7 Acknowledgements

This work was supported in part by the National Science Foundation under Grant DMI-9309207 and Grant DMI-9622588, and by industrial affiliates of the Integration Engineering Laboratory of the University of California, Davis.

8 References

1. Cheng, H. H., *Scientific Computing in the* C^H *Programming Language, Scientific Programming*, Vol. 2, No. 3, Fall 1993, pp. 49-75.

2. Cheng, H. H., Programming with Dual Numbers and its Applications in Mechanism Design, *Engineering with Computers*, 1994.

3. Cheng, H. H., *The* C^H *Language Environment*, http://iel.ucdavis.edu/CH, 1997.

4. Cheng, H. H., and Thompson, S., Computer-Aided Displacement Analysis of Spatial Mechanisms Using the C^H Programming Language, *Advances in Engineering Software*, Vol. 23, No. 3, 1995, pp. 163-172.

5. Denavit, J., Displacement Analysis of Mechanisms Based on (2x2) Matrices of Dual Numbers, *VDI-Berichte*, Vol. 29, 1958, pp. 81-89.

6. Denavit, J. and Hartenberg, R. S., A Kinematic Notation for Lower Pair Mechanisms Based on Matrices, *ASME Trans., Journal of Applied Mech.*, Vol. 22, 1955, pp. 215-221.

7. Gonzalez, P., *A Software Library for Displacement Analysis of Spatial Mechanisms*, MS Thesis, University of California, Davis, CA, 1996.

8. Hamilton, W. R., *Elements of Quaternions*, 2nd ed., C. J. Jolly, ed., Vol. 1, 1899; Vol. 2, 1901, Longmans, Green and Co., London.

9. McCarthy, M., *Introduction to Theoretical Kinematics*, MIT Press, Cambridge, MA, 1990.

10. Pennock, G. R. and Yang, A. T., Application of Dual Number Matrices to the Inverse Kinematic Problem of Robot Manipulators, *ASME Trans., Journal of Mechanisms, Transmissions, and Automation in Design*, Vol. 107, June 1985, pp. 201-208.

11. Ravani, B. and Roth, B., Mappings of Spatial Kinematics, *ASME Trans., Journal of Mechanisms, Transmissions, and Automation in Design*, Vol. 106, Sept. 1984, pp. 341-347.

12. Soni, A. H. and Harrisburger, L., Die Anwendung der 3C Schraubungs Matrix auf die kinematische und dynamische Analyse von Raumlichen Getriben, *V.D.I. Berichte*, No. 127, 1968.

13. Suh, C. H. and Radcliffe, C. W., *Kinematics and Mechanism Design*, John Wiley and Sons Inc., NY, 1978.

14. Wolfram, Stephen, *Mathematica, A system for doing mathematics by computer*, Addison Wesley Pub. Co., Advanced book program, 2^{nd} edition, 1991.

15. Yang, A. T. and Freudenstein, F., Application of Dual Number Quaternions Algebra to the Analysis of Spatial Mechanisms, *ASME Journal of Applied Mechanics*, Vol. 31, *Trans. ASME*, Vol. 86, Series E, No. 2, June 1964, pp. 300-308.

16. Yang, A. T., Displacement Analysis of Spatial Five-Link Mechanisms Using (3x3) Matrices with Dual-Number Elements, *Journal of Engineering for Industry*, Vol. 91, Feb. 1969, pp. 152-157.

17. Yuan, S. C., Analysis of the RCRCR Five-Link Spatial Mechanism, *Journal of Mechanisms*, Vol. 6, 1970, pp. 119-134.

Appendix A: Mathematica program RCRCR.mff.

```
/*============= Mathematica program RCRCR.mff ==================
  <*
    (* Denavit-Hartenberg matrices *)
    T1={{C1,-S1,0},{S1,C1,0},{0,0,1}};                        (* eq(2)  *)
    T2={{C2,-S2,0},{S2,C2,0},{0,0,1}};
    T3={{C3,-S3,0},{S3,C3,0},{0,0,1}};
    T4={{C4,-S4,0},{S4,C4,0},{0,0,1}};
    T5={{C5,-S5,0},{S5,C5,0},{0,0,1}};
    A1={{1,0,0},{0,Ca1,-Sa1},{0,Sa1,Ca1}};                    (* eq(3)  *)
    A2={{1,0,0},{0,Ca2,-Sa2},{0,Sa2,Ca2}};
    A3={{1,0,0},{0,Ca3,-Sa3},{0,Sa3,Ca3}};
    A4={{1,0,0},{0,Ca4,-Sa4},{0,Sa4,Ca4}};
    A5={{1,0,0},{0,Ca5,-Sa5},{0,Sa5,Ca5}};

    (* I/O loop-closure displacement equation *)
    IO=A1.T1.A5.T5.A4-Transpose[T4.A3.T3.A2.T2]               (* eq(4)  *)
    eq1=Expand[IO[[1,1]]]
    eq2=Expand[IO[[1,2]]]
    eq3=Expand[IO[[1,3]]]
    eq4=Expand[IO[[2,1]]]
    eq5=Expand[IO[[2,2]]]
    eq6=Expand[IO[[2,3]]]
    eq7=Expand[IO[[3,1]]]
    eq8=Expand[IO[[3,2]]]
    eq9=Expand[IO[[3,3]]]

    (* Get dual coefficients for element (3,3) in eq(5) *)
    BBB=Coefficient[eq9,C3,1];                                (* eq(8)  *)
    CCC=Coefficient[eq9,S5,1];                                (* eq(9)  *)
    DDD=Coefficient[eq9,C5,1];                                (* eq(10) *)
    AAA=Expand[eq9-BBB*C3-CCC*S5-DDD*C5];                     (* eq(7)  *)

    (* Real and dual number substitution *)
    AA=Ax+e*Ay; BB=Bx+e*By; CC=Cx+e*Cy; DD=Dx+e*Dy;
    CC3=c3-e*d3*s3; SS5=s5+e*d5*c5; CC5=c5-e*d5*s5;

    (* Separate real and dual parst of eq9 *)
    eq9=Expand[AA+BB*CC3+CC*SS5+DD*CC5];                      (* eq(6) *)
    Req9=Coefficient[eq9,e,0];                                (* eq(11) *)
    Deq9=Coefficient[eq9,e,1];                                (* eq(12) *)

    (* Solve for c3 and s3 from Req9 and Deq9 *)
    temp=Solve[Req9==0,c3];
    c3=c3 /. temp; c3=c3[[1]];
    temp=Solve[Deq9==0,s3];
    s3=s3 /. temp; s3=s3[[1]];

    (* Substitue half angle tangent for s5 and c5 *)
    s5=2*X/(1+X*X); c5=(1-X*X)/(1+X*X);                       (* eq(13) *)
    s3=Together[Expand[s3]]*(1+X*X);                          (* eq(14) *)
    c3=Together[Expand[c3]]*(1+X*X);                          (* eq(15) *)

    (* Get constants multiplied by X from s3 and c3 *)
    EEE=Together[Coefficient[s3,X,2]];                        (* eq(18) *)
    FFF=Together[Coefficient[s3,X,1]];                        (* eq(19) *)
    GGG=Together[Coefficient[s3,X,0]];                        (* eq(20) *)
    HHH=Together[Coefficient[c3,X,2]];                        (* eq(21) *)
    III=Together[Coefficient[c3,X,1]];                        (* eq(22) *)
    JJJ=Together[Coefficient[c3,X,0]];                        (* eq(23) *)

    (* Use c3^2 + s3^2 -1 = 0 to form a 4th order polynomial *)
    s3=(EE*X*X+FF*X+GG); c3=(HH*X*X+II*X+JJ);                 (* eq(16-17) *)
    poly=Together[Expand[c3*c3+s3*s3-(1+X*X)*(1+X*X)]];       (* eq(25) *)
    p4=Together[Coefficient[poly,X,4]];                       (* eq(26) *)
    p3=Together[Coefficient[poly,X,3]];                       (* eq(27) *)
    p2=Together[Coefficient[poly,X,2]];                       (* eq(28) *)
```

```
    p1=Together[Coefficient[poly,X,1]];                            (* eq(29) *)
    p0=Together[Coefficient[poly,X,0]];                            (* eq(40) *)

    (* Use element (1,3) and (2,3) to find theta2 *)
    temp=Coefficient[eq3,S2,0]; AA1=-Coefficient[temp,C2,0];       (* eq(35) *)
    CC1=Coefficient[eq3,S2,1];  DD1=Coefficient[eq3,C2,1];         (* eq(35) *)
    temp=Coefficient[eq6,S2,0]; BB1=-Coefficient[temp,C2,0];       (* eq(36) *)
    EE1=Coefficient[eq6,S2,1];  FF1=Coefficient[eq6,C2,1];         (* eq(36) *)

    (* Use element (3,1) and (3,2) to find theta4 *)
    temp=Coefficient[eq7,S4,0]; AA2=-Coefficient[temp,C4,0];       (* eq(40) *)
    CC2=Coefficient[eq7,S4,1];  DD2=Coefficient[eq7,C4,1];         (* eq(40) *)
    temp=Coefficient[eq8,S4,0]; BB2=-Coefficient[temp,C4,0];       (* eq(41) *)
    EE2=Coefficient[eq8,S4,1];  FF2=Coefficient[eq8,C4,1];         (* eq(41) *)
  *>
=================================================================*/
void RCRCR(double dual a[5],double d[3],double dual theta1
           double dual theta2[4],theta3[4],theta4[4],theta5[4])
{
  #define Power pow
  int j;
  double t1,d1,d3,d5,X,EE,FF,GG,HH,II,JJ,
         SIN3,COS3, Ax,Ay,Bx,By,Cx,Cy,Dx,Dy;
  double dual T1,T2,T3,T4,T5,a1,a2,a3,a4,a5,AA,BB,CC,DD,AA1,
              BB1,CC1,DD1,EE1,FF1,AA2,BB2,CC2,DD2,EE2,FF2;
  double complex roots[4],p[5];

  T1=theta1;
  a1=a[0]; a2=a[1]; a3=a[2]; a4=a[3]; a5=a[4];
  d1=d[0]; d3=d[1]; d5=d[2];
  t1=real(T1);
  AA=<*AAA*>;  BB=<*BBB*>;                                         /* eq(7-8) */
  CC=<*CCC*>;  DD=<*DDD*>;                                         /* eq(9-10) */
  Ax=real(AA); Ay=dual(AA); Bx=real(BB); By=dual(BB);
  Cx=real(CC); Cy=dual(CC); Dx=real(DD); Dy=dual(DD);
  EE=<*EEE*>;                                                      /* eq(18) */
  FF=<*FFF*>;                                                      /* eq(19) */
  GG=<*GGG*>;                                                      /* eq(20) */
  HH=<*HHH*>; II=<*III*>;  JJ=<*JJJ*>;                             /* eq(21-23) */
  p[4]=<*p4*>;                                                     /* eq(26) */
  p[3]=<*p3*>;                                                     /* eq(27) */
  p[2]=<*p2*>;                                                     /* eq(28) */
  p[1]=<*p1*>;                                                     /* eq(29) */
  p[0]=<*p0*>;                                                     /* eq(30) */
  zroots(p,4,roots,1);  /* solve polynomial equation */
  for(j=0;j<=3;j++) {
    X=real(roots[j]);
    T5=dual(2*atan(X), d5);                                        /* eq(31) */
    SIN3=(<*s3*>)/(1+X*X);                                         /* eq(16) */
    COS3=(<*c3*>)/(1+X*X);                                         /* eq(17) */
    T3=dual(2*atan(SIN3/(1+COS3)), d3);                            /* eq(32) */
    AA1=<*AA1*>;
    BB1=<*BB1*>;
    CC1=<*CC1*>; DD1=<*DD1*>;
    EE1=<*EE1*>; FF1=<*FF1*>;
    T2=2*atan((AA1*FF1-BB1*DD1)/(CC1*FF1-DD1*EE1+BB1*CC1-AA1*EE1));     /* eq(37) */
    AA2=<*AA2*>;
    BB2=<*BB2*>;
    CC2=<*CC2*>; DD2=<*DD2*>;
    EE2=<*EE2*>; FF2=<*FF2*>;
    T4=2*atan((AA2*FF2-BB2*DD2)/(CC2*FF2-DD2*EE2+BB2*CC2-AA2*EE2));     /* eq(42) */
    theta2[j]=T2; theta3[j]=T3; theta4[j]=T4; theta5[j]=T5;
  }
}
```

Appendix B: C^H function file RCRCR.ff automatically generated by Mathematica program RCRCR.mff in Appendix A.

```
/*============= Mathematica program RCRCR.mff =================
  Null
===============================================================*/
void RCRCR(double dual a[5],double d[3],double dual theta1
           double dual theta2[4],theta3[4],theta4[4],theta5[4])
{
  #define Power pow
  int j;
  double t1,d1,d3,d5,X,EE,FF,GG,HH,II,JJ,
         SIN3,COS3, Ax,Ay,Bx,By,Cx,Cy,Dx,Dy;
  double dual T1,T2,T3,T4,T5,a1,a2,a3,a4,a5,AA,BB,CC,DD,AA1,
              BB1,CC1,DD1,EE1,FF1,AA2,BB2,CC2,DD2,EE2,FF2;
  double complex roots[4],p[5];

  T1=theta1;
  a1=a[0]; a2=a[1]; a3=a[2]; a4=a[3]; a5=a[4];
  d1=d[0]; d3=d[1]; d5=d[2];
  t1=real(T1);
  AA=-(Ca2*Ca3) + Ca1*Ca4*Ca5 - C1*Ca4*Sa1*Sa5;  BB=Sa2*Sa3;             /* eq(7-8) */
  CC=S1*Sa1*Sa4;  DD=-(C1*Ca5*Sa1*Sa4) - Ca1*Sa4*Sa5;                    /* eq(9-10) */
  Ax=real(AA); Ay=dual(AA); Bx=real(BB); By=dual(BB);
  Cx=real(CC); Cy=dual(CC); Dx=real(DD); Dy=dual(DD);
  EE=(Ay*Bx - Ax*By - Bx*Cx*d5 + By*Dx - Bx*Dy)/(Power(Bx,2)*d3);        /* eq(18) */
  FF=(2*(-(By*Cx) + Bx*Cy - Bx*d5*Dx))/(Power(Bx,2)*d3);                 /* eq(19) */
  GG=(Ay*Bx - Ax*By + Bx*Cx*d5 - By*Dx + Bx*Dy)/(Power(Bx,2)*d3);        /* eq(20) */
  HH=(-Ax + Dx)/Bx; II=(-2*Cx)/Bx;  JJ=(-Ax - Dx)/Bx;                    /* eq(21-23) */
  p[4]=-1 + Power(EE,2) + Power(HH,2);                                   /* eq(26) */
  p[3]=2*(EE*FF + HH*II);                                                /* eq(27) */
  p[2]=-2 + Power(FF,2) + 2*EE*GG + Power(II,2) + 2*HH*JJ;               /* eq(28) */
  p[1]=2*(FF*GG + II*JJ);                                                /* eq(29) */
  p[0]=-1 + Power(GG,2) + Power(JJ,2);                                   /* eq(30) */
  zroots(p,4,roots,1);  /* solve polynomial equation */
  for(j=0;j<=3;j++) {
    X=real(roots[j]);
    T5=dual(2*atan(X), d5);                                              /* eq(31) */
    SIN3=(GG + FF*X + EE*Power(X,2))/(1+X*X);                            /* eq(16) */
    COS3=(JJ + II*X + HH*Power(X,2))/(1+X*X);                            /* eq(17) */
    T3=dual(2*atan(SIN3/(1+COS3)), d3);                                  /* eq(32) */
    AA1=-(C5*Ca5*S1*Sa4) - C1*S5*Sa4 - Ca4*S1*Sa5;
    BB1=Ca4*Ca5*Sa1 + C1*C5*Ca1*Ca5*Sa4 - Ca1*S1*S5*Sa4 + C1*Ca1*Ca4*Sa5 - C5*Sa1*Sa4*Sa5;
    CC1=-(Ca3*Sa2) - C3*Ca2*Sa3; DD1=-(S3*Sa3);
    EE1=S3*Sa3; FF1=-(Ca3*Sa2) - C3*Ca2*Sa3;
    T2=2*atan((AA1*FF1-BB1*DD1)/(CC1*FF1-DD1*EE1+BB1*CC1-AA1*EE1));      /* eq(37) */
    AA2=-(C5*S1*Sa1) - C1*Ca5*S5*Sa1 - Ca1*S5*Sa5;
    BB2=-(C1*C5*Ca4*Ca5*Sa1) + Ca4*S1*S5*Sa1 - Ca1*Ca5*Sa4 - C5*Ca1*Ca4*Sa5 + C1*Sa1*Sa4*Sa5;
    CC2=-(C3*Ca3*Sa2) - Ca2*Sa3; DD2=-(S3*Sa2);
    EE2=-(S3*Sa2); FF2=C3*Ca3*Sa2 + Ca2*Sa3;
    T4=2*atan((AA2*FF2-BB2*DD2)/(CC2*FF2-DD2*EE2+BB2*CC2-AA2*EE2));      /* eq(42) */
    theta2[j]=T2; theta3[j]=T3; theta4[j]=T4; theta5[j]=T5;
  }
}
```