ELSEVIER

0965-9978(95)00074-7

Advances in Engineering Software 23 (1995) 163–172
Elsevier Science Limited
Printed in Great Britain

Computer-aided displacement analysis of spatial mechanisms using the C^H programming language

Harry H. Cheng & Sean Thompson

Integration Engineering Laboratory, Department of Mechanical & Aeronautical Engineering, and Graduate Group in Computer Science, University of California, Davis, CA 95616, USA

(Received 15 May 1994; accepted 1 September 1995)

Recently, Cheng in 1993 introduced the C^H programming language. C^H is designed to be a superset of C with all the programming features of FORTRAN. Formulas with dual numbers can be translated into C^H programming statements as easily as formulas with real and complex numbers. In this paper we will show that both formulation and programming with dual numbers are remarkably simple for analysis of complicated spatial mechanisms under the C^H programming paradigm. With computational capabilities for dual formulas in mind, formulas for analysis of spatial mechanisms are derived differently from those intended for implementation in computer programming languages without dual data type. We will demonstrate some formulation and programming techniques in the C^H programming paradigm through displacement analysis of the RCRCR five-link spatial mechanism. A C^H program that can obtain both numerical and graphical results for complete displacement analysis of the RCRCR mechanism will be presented.

Key words: C^H language, dual number, RCRCR mechanism.

1 INTRODUCTION

Dual number is a powerful mathematical tool for analysis and design of spatial mechanisms. Although considerable progress has been made in using dual numbers for various applications over the last century, progress in numerical handling and simplification of scientific programming with dual numbers is relatively slow. FORTRAN is the first general-purpose high-level programming language ever developed. It was developed for FORmula TRANslation in the 1950s. FORTRAN has revolutionarily changed the way scientists and engineers think about formulating problems. Formulas with real and complex numbers can be easily translated into FORTRAN programming statements. However, formulas with dual numbers are difficult to handle in FORTRAN or any other programming language because dual number is not a basic data type in the language. Although succinct dual formulations for mechanism analysis and design can be derived, their numerical computation tends to become cumbersome. For numerical implementation, dual equations have to be reformulated as complicated formulas with only real numbers. The process of rewriting equations with only real numbers is complicated and error-prone.

To use dual numbers effectively, we need a programming language that treats dual number as a basic data type. Although dual number can be treated as if it was a built-in data type in C++ by using operator and function overloading through a class, developing such a class in C++ is not a trivial task for novice users and beginners. Besides, many desirable programming features related to dual numbers simply cannot be implemented in C++ at the user's level. They should be treated as one of language primitives.[1] Recently, Cheng introduced the C^H programming language.[2-4] C^H is a superset of ANSI C with incorporation of all programming features of FORTRAN. Due to our research interests, C^H is designed to be specially suitable for applications in mechanical engineering, although it is useful in many other disciplines as well. Many programming features in C^H are specially designed and implemented for design automation. Handling dual number as a basic built-in data type in the language is one example. It is believed that handling of dual numbers in C^H is simpler than any other computer programming language. The linguistic features related to dual numbers in C^H have been presented by Cheng.[1] Dual numbers are treated in the same manner as real or complex numbers in C^H. The simplicity of programming with dual numbers in C^H has been demonstrated by kinematic analysis of an RCCC mechanism.[1] Translating dual formulas for displacement analysis of the RCCC mechanism into a C^H program is so simple that some researchers have commented, based upon their reading of Ref. 1, that it is useful to demonstrate how the C^H language can be used to simplify the solution of more complicated spatial mechanisms that do not have closed-form dual solutions. The purpose of this paper is to show that both formulation and programming with dual numbers under the programming paradigm of C^H are remarkably simple for analysis of spatial mechanisms that do not have closed-form solutions. Programming with dual numbers in C^H is very simple and is quite different from programming in any other programming languages. In this paper we will demonstrate some formulation and programming techniques for analysis of spatial mechanisms under the C^H programming paradigm.

The RCRCR spatial mechanism will be used to demonstrate the simplicity of formulation and programming with dual numbers in C^H. The displacement analysis of the RCRCR mechanism has been studied extensively over the last two decades. Based upon dual numbers, Dimentberg[5] used screw algebra to obtain an input–output displacement equation of degree eight in half-tangents of the output angular displacement for this mechanism. Using dual formulation matrices, Yang[6] simplified the input–output relation to a quadratic equation. For complete displacement analysis in FORTRAN, Yang had to separate a concise dual formula into two equations, one for the real part and the other for the dual part. Experience indicates that transforming original dual formulas into formulas with only real numbers is very complicated and error-prone.[6] Using line coordinates with dual numbers, Yuan[7] presented complete solutions for displacement analysis of the same mechanism. Using spherical trigonometry with dual numbers, Duffy & Habib-Olahi[8,9] also studied the RCRCR mechanism. They showed that there are eight solutions for the RCRCR mechanism if the displacement d_3 for joint 3 takes both positive and negative values. Derivation of displacement solutions in all the aforementioned different methods have used dual number as a mathematical tool in one way or another. But, all their final analytical solutions have to be expressed in real numbers after tremendous paper work. Within the programming paradigm of C^H, this is no longer the case and we can leave dual numbers intact in the final analytical solutions. In this paper we will derive analytical solutions for the displacement analysis of the RCRCR mechanism within the framework of Yang[6] and obtain Duffy & Habib-Olahi's result on eight solutions for the RCRCR in an alternative form. Because dual formulas can be readily translated into a C^H program, formulations presented in this paper are very simple and easy to follow. Most importantly, based upon these dual formulas, we can easily obtain both numerical and graphical results in the C^H programming language.

2 DUAL FORMULATION FOR DISPLACEMENT ANALYSIS OF THE RCRCR MECHANISM

In this section, a displacement analysis of the RCRCR mechanism shown in Fig. 1 will be used to demonstrate how to formulate dual equations for proper implementation in the C^H programming language. The analysis of this mechanism consists of determining the revolute joint values θ_3 and θ_5 and cylindrical joint values $\hat{\theta}_2$ and $\hat{\theta}_4$ for a given input angle θ_1. For the RCRCR mechanism, link parameters $\hat{\alpha}_i = \alpha_i + \epsilon a_i$ $(i = 1, \ldots, 5)$, d_1, d_3, and d_5 are fixed. The parameters θ_i, α_i, a_i, and d_i are shown in Fig. 1. The dual part of a dual angle is defined as the mutually perpendicular distance between two arbitrary lines in space and the real part is defined as the angle between these two lines as seen along the perpendicular line. The transformation of the coordinate system from $X_iY_iZ_iO_i$ to $X'Y'Z'O'$, an intermediate coordinate system,[1] can be represented by the following dual transformation matrix

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

where dual angle $\hat{\theta}_i = \theta_i + \epsilon d_i$. Similarly, the transformation of the coordinate system from $X'Y'Z'$ to $X_{i+1}Y_{i+1}Z_{i+1}O_{i+1}$ can be represented by the dual transformation matrix

$$\hat{\Lambda}_i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\hat{\alpha}_i & -\sin\hat{\alpha}_i \\ 0 & \sin\hat{\alpha}_i & \cos\hat{\alpha}_i \end{bmatrix}, \tag{2}$$

where dual angle $\hat{\alpha}_i = \alpha_i + \epsilon a_i$. The dual transformation matrix between two coordinate systems $X_iY_iZ_iO_i$ and $X_{i+1}Y_{i+1}Z_{i+1}O_{i+1}$ can be derived by the successive screw transformations as follows:

$$\begin{aligned} \hat{\mathbf{D}}_i &= \hat{\Theta}_i\hat{\Lambda}_i \\ &= \begin{bmatrix} \cos\hat{\theta}_i & -\sin\hat{\theta}_i\cos\hat{\alpha}_i & \sin\hat{\theta}_i\sin\hat{\alpha}_i \\ \sin\hat{\theta}_i & \cos\hat{\theta}_i\cos\hat{\alpha}_i & -\cos\hat{\theta}_i\sin\hat{\alpha}_i \\ 0 & \sin\hat{\alpha}_i & \cos\hat{\alpha}_i \end{bmatrix}, \end{aligned} \tag{3}$$

where $\hat{\Theta}_i$ and $\hat{\Lambda}_i$ are given in eqns (1) and (2), respectively.

In Fig. 1 the revolute axes Z_1 and Z_5 are rotated with respect to the ground. In other words, the coordinate systems $O_1X_1Y_1Z_1$ and $O_5X_5Y_5Z_5$ are attached to input

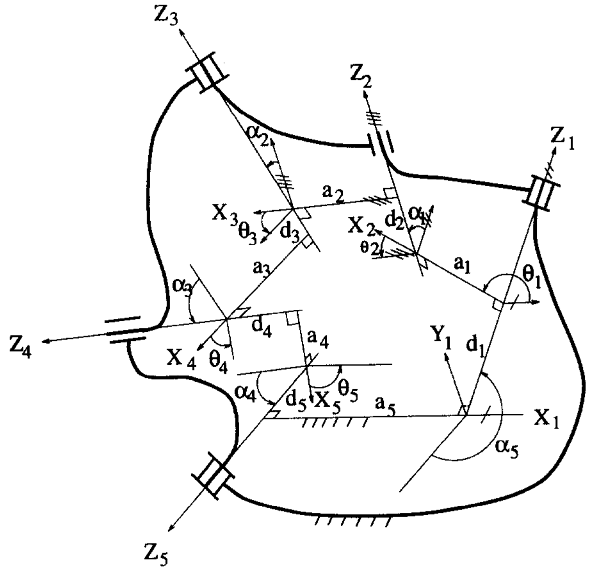

Fig. 1. The RCRCR mechanism.

and output links, respectively. Formulas for calculating θ_3, θ_5, $\hat{\theta}_2$, and $\hat{\theta}_4$ in terms of the input angle θ_1 can be derived as follows. According to the dual transformation matrices $\hat{\Theta}_i$ and $\hat{\Lambda}_i$, we can write the loop-closure equation for the RCRCR mechanism shown in Fig. 1 as

$$\hat{\Theta}_4\hat{\Lambda}_3\hat{\Theta}_3\hat{\Lambda}_2\hat{\Theta}_2\hat{\Lambda}_1\hat{\Theta}_1\hat{\Lambda}_5\hat{\Theta}_5\hat{\Lambda}_4 = \mathbf{I}, \tag{4}$$

where $\mathbf{I}$ is a 3×3 identity matrix. Equation (4) can be cyclically permutated.[6,10] The goal of the reformulation is to create a loop-closure matrix equation that simplifies the algebraic manipulations needed. Reformulation of the loop-closure eqn (4) for different five-link spatial mechanisms is determined by knowledge of the characteristics of the resultant matrix equation. Each element of the matrix eqn (4) gives a dual equation. Therefore, there are nine dual equations that can be used to obtain relations between the output variables and input variable θ_1. Equation (4) can be reformulated as[6]

$$\hat{\Lambda}_1\hat{\Theta}_1\hat{\Lambda}_5\hat{\Theta}_5\hat{\Lambda}_4 = \hat{\Theta}_4^T\hat{\Lambda}_3^T\hat{\Theta}_3^T\hat{\Lambda}_2^T\hat{\Theta}_2^T, \tag{5}$$

where $\hat{\Theta}_i^T$ and $\hat{\Lambda}_i^T$ are the transposes of matrices $\hat{\Theta}_i$ and $\hat{\Lambda}_i$, respectively. Multiplying matrices, eqn (5) becomes

$$\begin{bmatrix} \hat{a}\cos\hat{\theta}_5 - \hat{d}\sin\hat{\theta}_5 & \hat{o}(\hat{a}\sin\hat{\theta}_5 + \hat{d}\cos\hat{\theta}_5) - \hat{g}\hat{n} & \hat{n}(\hat{a}\sin\hat{\theta}_5 + \hat{d}\cos\hat{\theta}_5) + \hat{g}\hat{o} \\ \hat{b}\cos\hat{\theta}_5 - \hat{e}\sin\hat{\theta}_5 & \hat{o}(\hat{b}\sin\hat{\theta}_5 + \hat{e}\cos\hat{\theta}_5) - \hat{h}\hat{n} & \hat{n}(\hat{b}\sin\hat{\theta}_5 + \hat{e}\cos\hat{\theta}_5) + \hat{h}\hat{o} \\ \hat{c}\cos\hat{\theta}_5 - \hat{f}\sin\hat{\theta}_5 & \hat{o}(\hat{c}\sin\hat{\theta}_5 + \hat{f}\cos\hat{\theta}_5) - \hat{i}\hat{n} & \hat{n}(\hat{c}\sin\hat{\theta}_5 + \hat{f}\cos\hat{\theta}_5) + \hat{i}\hat{o} \end{bmatrix}$$

$$= \begin{bmatrix} (\cos\hat{\theta}_2\cos\hat{\theta}_4 - (\hat{k}(\sin\hat{\theta}_3\cos\hat{\theta}_4 + \hat{m}\sin\hat{\theta}_3\sin\hat{\theta}_4) - \hat{k}\hat{l}\sin\hat{\theta}_4)\sin\hat{\theta}_2 \\ (\sin\hat{\theta}_2\cos\hat{\theta}_4 + (\hat{k}(\sin\hat{\theta}_3\cos\hat{\theta}_4 + \hat{m}\sin\hat{\theta}_3\sin\hat{\theta}_4) - \hat{k}\hat{l}\sin\hat{\theta}_4)\cos\hat{\theta}_2 \\ \hat{j}(\sin\hat{\theta}_3\cos\hat{\theta}_4 + \hat{m}\sin\hat{\theta}_3\sin\hat{\theta}_4) + \hat{k}\hat{l}\sin\hat{\theta}_4 \end{bmatrix.$$

$$\begin{matrix} \sin\hat{\theta}_4\cos\hat{\theta}_4 - ((\hat{m}\cos\hat{\theta}_3\cos\hat{\theta}_4 - \sin\hat{\theta}_3\sin\hat{\theta}_4)\hat{k} - \hat{j}\hat{l}\cos\hat{\theta}_4)\sin\hat{\theta}_2 \\ \sin\hat{\theta}_4\cos\hat{\theta}_4 + ((\hat{m}\cos\hat{\theta}_3\cos\hat{\theta}_4 - \sin\hat{\theta}_3\sin\hat{\theta}_4)\hat{k} - \hat{j}\hat{l}\cos\hat{\theta}_4)\cos\hat{\theta}_2 \\ -\hat{j}(\sin\hat{\theta}_3\sin\hat{\theta}_4 - \hat{m}\cos\hat{\theta}_3\cos\hat{\theta}_4) + \hat{k}\hat{l}\cos\hat{\theta}_4 \end{matrix}$$

$$\left.\begin{matrix} \hat{l}\cos\hat{\theta}_2\sin\hat{\theta}_3 + (\hat{k}\hat{l}\cos\hat{\theta}_3 + \hat{j}\hat{m})\sin\hat{\theta}_2 \\ \hat{l}\sin\hat{\theta}_2\sin\hat{\theta}_3 + (\hat{k}\hat{l}\cos\hat{\theta}_3 + \hat{j}\hat{m})\cos\hat{\theta}_2 \\ \hat{k}\hat{m} - \hat{j}\hat{l}\cos\hat{\theta}_3 \end{matrix}\right] \tag{6}$$

where

$$\hat{a} = \cos\hat{\theta}_1 \tag{7}$$

$$\hat{b} = -\cos\hat{\alpha}_1\sin\hat{\theta}_1 \tag{8}$$

$$\hat{c} = \sin\hat{\alpha}_1\sin\hat{\theta}_1 \tag{9}$$

$$\hat{d} = \cos\hat{\alpha}_5\sin\hat{\theta}_1 \tag{10}$$

$$\hat{e} = \cos\hat{\alpha}_1\cos\hat{\theta}_1\cos\hat{\alpha}_5 - \sin\hat{\alpha}_5\sin\hat{\alpha}_1 \tag{11}$$

$$\hat{f} = -\sin\hat{\alpha}_1\cos\hat{\theta}_1\cos\hat{\alpha}_5 - \cos\hat{\alpha}_1\sin\hat{\alpha}_5 \tag{12}$$

$$\hat{g} = \sin\hat{\theta}_1\sin\hat{\alpha}_5 \tag{13}$$

$$\hat{h} = \cos\hat{\alpha}_1\cos\hat{\theta}_1\sin\hat{\alpha}_5 + \sin\hat{\alpha}_1\cos\hat{\alpha}_5 \tag{14}$$

$$\hat{i} = -\sin\hat{\alpha}_1\cos\hat{\theta}_1\sin\hat{\alpha}_5 + \cos\hat{\alpha}_1\cos\hat{\alpha}_5 \tag{15}$$

$$\hat{j} = \sin\hat{\alpha}_2 \tag{16}$$

$$\hat{k} = \cos\hat{\alpha}_2 \tag{17}$$

$$\hat{l} = \sin\hat{\alpha}_3 \tag{18}$$

$$\hat{m} = \cos\hat{\alpha}_3 \tag{19}$$

$$\hat{n} = \sin\hat{\alpha}_4 \tag{20}$$

$$\hat{o} = \cos\hat{\alpha}_4 \tag{21}$$

To determine solutions to the unknown values $\hat{\theta}_2$, θ_3, $\hat{\theta}_4$ and θ_5, the following steps are taken. First, the dual equation from element (3,3) of dual matrix eqn (6) is used to determine θ_3 and θ_5. Next, $\hat{\theta}_2$ is determined by solving equations from elements (1,3) and (2,3) with the given input value $\hat{\theta}_1$ and newly determined values $\hat{\theta}_3$ and $\hat{\theta}_5$. Finally, $\hat{\theta}_4$ is determined similarly by solving equations from elements (3,1) and (3,2) with the known values $\hat{\theta}_1$, $\hat{\theta}_3$, and $\hat{\theta}_5$.

Element (3,3) of dual matrix eqn (6) gives the following input–output relation of the RCRCR spatial mechanism.

$$\begin{aligned} \cos\hat{\alpha}_2\cos\hat{\alpha}_3 - \sin\hat{\alpha}_2\sin\hat{\alpha}_3\cos\hat{\theta}_3 &= (\sin\hat{\alpha}_1\sin\hat{\alpha}_4\sin\hat{\theta}_1)\sin\hat{\theta}_5 \\ &\quad - [\sin\hat{\alpha}_4(\sin\hat{\alpha}_3\cos\hat{\alpha}_1 \\ &\quad + \cos\hat{\alpha}_3\sin\hat{\alpha}_1\cos\hat{\theta}_1)]\cos\hat{\theta}_5 \\ &\quad + \cos\hat{\alpha}_4(\cos\hat{\alpha}_5\cos\hat{\alpha}_1 - \sin\hat{\alpha}_5\sin\hat{\alpha}_1\cos\hat{\theta}_1). \end{aligned} \tag{22}$$

Equation (22) contains three dual angles $\hat{\theta}_1$, $\hat{\theta}_3$, and $\hat{\theta}_5$. Both d_3 and d_5 are given so that there are only 2 unknowns, θ_3 and θ_5, in eqn (22). Therefore, we should further simplify the equation to contain only $\hat{\theta}_3$ and $\hat{\theta}_5$, and express the rest as parameters that can be easily handled by the C^{H} programming language. Equation (22) can be easily rewritten as

$$\hat{A} - \hat{B}\cos\hat{\theta}_3 = \hat{C}\sin\hat{\theta}_5 - \hat{D}\cos\hat{\theta}_5, \tag{23}$$

where

$$\hat{A} = \cos\hat{\alpha}_2 \cos\hat{\alpha}_3 - \cos\hat{\alpha}_4(\cos\hat{\alpha}_5 \cos\hat{\alpha}_1 + \sin\hat{\alpha}_5 \sin\hat{\alpha}_1 \cos\hat{\theta}_1) \quad (24)$$

$$\hat{B} = \sin\hat{\alpha}_2 \sin\hat{\alpha}_3 \quad (25)$$

$$\hat{C} = \sin\hat{\alpha}_1 \cos\hat{\alpha}_4 \sin\hat{\theta}_1 \quad (26)$$

$$\hat{D} = \sin\hat{\alpha}_4(\sin\hat{\alpha}_5 \cos\hat{\alpha}_1 + \cos\hat{\alpha}_5 \sin\hat{\alpha}_1 \cos\hat{\theta}_1). \quad (27)$$

These parameters are very easy to compute in C^H. For example, parameter $\hat{C}$ can be calculated in C^H as

```
C = sin(alpha1) * cos(alpha4) * sin(theta1);.
```

Details on C^H programming for the RCRCR mechanism will be described in the next section. There are two unknowns, θ_3 and θ_5, in eqn (23). These two unknowns are real parts of dual angles $\hat{\theta}_3$ and $\hat{\theta}_5$. Therefore, it is inevitable at this point to single out θ_3 and θ_5 from dual numbers $\hat{\theta}_3$ and $\hat{\theta}_5$, respectively. But, other dual parameters in the equation can remain intact. Furthermore, we can either solve for θ_3 first, then for θ_5; or solve for θ_5 first, then for θ_3. We choose to solve θ_5 first. Because the resultant equation involves both $\sin\theta_5$ and $\cos\theta_5$, we introduce an additional parameter $\chi = \tan(\theta_5/2)$ to express both $\sin\theta_5$ and $\cos\theta_5$ in terms of χ. After substituting the identities

$$\sin\hat{\theta}_5 = \sin\theta_5 + \epsilon d_5 \cos\theta_5 \quad (28)$$

$$\cos\hat{\theta}_5 = \cos\theta_5 - \epsilon d_5 \sin\theta_5 \quad (29)$$

$$\sin\theta_5 = \frac{2\chi}{1+\chi^2} \quad (30)$$

$$\cos\theta_5 = \frac{1-\chi^2}{1+\chi^2} \quad (31)$$

where

$$\chi = \tan\frac{\theta_5}{2}, \quad (32)$$

and making additional substitutions,

$$\hat{E} = \frac{\hat{A} + \epsilon d_5 \hat{C} - \hat{D}}{\hat{B}} \quad (33)$$

$$\hat{F} = \frac{\hat{C} + \epsilon d_5 \hat{D}}{\hat{B}} \quad (34)$$

$$\hat{G} = \frac{\hat{A} - \epsilon d_5 + \hat{C} + \hat{D}}{\hat{B}} \quad (35)$$

eqn (23) becomes

$$\cos\hat{\theta}_3 = \cos\theta_3 - \epsilon d_3 \sin\theta_3 = \frac{\hat{E}\chi^2 - 2\hat{F}\chi + \hat{G}}{(\chi^2+1)}. \quad (36)$$

The identity $\sin^2\theta + \cos^2\theta = 1$ can be used to eliminate the variable θ_3. Equation (36) can be separated into its real and imaginary parts,

$$\cos\theta_3 = \frac{\acute{E}\chi^2 - 2\acute{F}\chi + \acute{G}}{(\chi^2+1)} \quad (37)$$

$$\sin\theta_3 = \frac{\grave{E}\chi^2 - 2\grave{F}\chi + \grave{G}}{-d_3(\chi^2+1)}. \quad (38)$$

A dual variable topped with an accent grave '\`' stands for the real part of the dual number, whereas a dual variable topped with an accent acute '´' stands for the dual part. For example, if $\hat{d} = x + \epsilon y$, then $\acute{d}$ and $\grave{d}$ are defined to be x and y, respectively. Substituting $\cos\theta_3$ and $\sin\theta_3$ into the aforementioned identity equation, we get

$$p_4\chi^4 + p_3\chi^3 + p_2\chi^2 + p_1\chi + p_0 = 0, \quad (39)$$

where

$$p_4 = \acute{E}^2 + \frac{\grave{E}^2}{d_3^2} - 1 \quad (40)$$

$$p_3 = -4\left(\acute{E}\acute{F} + \frac{\grave{E}\grave{F}}{d_3^2}\right) \quad (41)$$

$$p_2 = 2\acute{E}\acute{G} + 4\acute{F}^2 + \frac{2\grave{E}\grave{G} + 4\grave{F}^2}{d_3^2} - 2 \quad (42)$$

$$p_1 = -4\left(\acute{F}\acute{G} + \frac{\grave{F}\grave{G}}{d_3^2}\right) \quad (43)$$

$$p_0 = \acute{G}^2 + \frac{\grave{G}^2}{d_3^2} - 1. \quad (44)$$

Coefficients p_i $(i = 0, \ldots, 4)$ of the polynomial eqn (39) can be easily computed in the C^H programming language. For example, coefficient p_4 in eqn (40) can be programmed in C^H as

```
p4 = real(F) * real(F)
     + dual(F) * dual(F)/(d3 * d3) - 1;
```

The solution for χ in eqn (39) can be computed by using a numerical solution. And θ_5 then can be determined from the equation $\theta_5 = 2\mathrm{atan}(\chi)$. Because eqn (39) 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 eqn (36).

$$\hat{\theta}_3 = \pm\,\mathrm{acos}\left(\frac{\hat{A} + \hat{D}\cos\hat{\theta}_5 - \hat{C}\sin\hat{\theta}_5}{\hat{B}}\right). \quad (45)$$

If d_3 takes both positive and negative values, there are two solutions for $\hat{\theta}_3$ from eqn (45) corresponding to a given $\hat{\theta}_5$. There are up to four solutions for $\hat{\theta}_5$ for a given θ_1. Therefore, there are up to eight solutions for the RCRCR mechanism as pointed out by Duffy & Habib-Olahi.[8] However, for a given RCRCR mechanism, the sign of d_3 is fixed. Therefore, only one of two solutions from eqn (45) is valid. This valid solution can be obtained unambiguously by the formula[10] $\theta = 2\,\mathrm{atan}\,(\sin\theta/(\cos\theta + 1))$.

$$\theta_3 = 2\,\mathrm{atan}\left(\frac{\sin\theta_3}{\cos\theta_3 + 1}\right)$$

$$= 2\,\mathrm{atan}\left(\frac{\grave{E}\chi^2 - 2\grave{F}\chi + \grave{G}}{-d_3(\acute{E} + 1)\chi^2 - 2\acute{F}\chi + \acute{G} + 1}\right) \qquad (46)$$

where $\cos\theta_3$ and $\sin\theta_3$ are from eqns (37) and (38), respectively.

At this point, $\hat{\theta}_1$, $\hat{\theta}_3$ and $\hat{\theta}_5$ all are known. The dual equation derived from elements (1,3) and (2,3) of dual matrix eqn (6) are in terms of $\hat{\theta}_1$, $\hat{\theta}_2$, $\hat{\theta}_3$, and $\hat{\theta}_5$. This leaves angle θ_2 and displacement d_2 as the only two unknowns in these two equations. The two unknowns can be solved by using two equations. Equations derived from elements (1,3) and (2,3) can be written as

$$\hat{A}_1 = \hat{D}_1 \sin\hat{\theta}_2 + \hat{C}_1 \cos\hat{\theta}_2 \qquad (47)$$

$$\hat{B}_1 = \hat{C}_1 \sin\hat{\theta}_2 - \hat{D}_1 \cos\hat{\theta}_2 \qquad (48)$$

respectively, where

$$\hat{A}_1 = (\cos\hat{\alpha}_4 \sin\hat{\alpha}_5 + \sin\hat{\alpha}_4 \cos\hat{\alpha}_5 \cos\hat{\theta}_5) \sin\hat{\theta}_1 + \sin\hat{\alpha}_4 \cos\hat{\theta}_1 \sin\hat{\theta}_5 \qquad (49)$$

$$\hat{B}_1 = (\cos\hat{\alpha}_5 \sin\hat{\alpha}_1 + \sin\hat{\alpha}_5 \cos\hat{\alpha}_1 \cos\hat{\theta}_1) \cos\hat{\alpha}_4 - [(\sin\hat{\alpha}_5 \sin\hat{\alpha}_1 - \cos\hat{\alpha}_5 \cos\hat{\alpha}_1 \cos\hat{\theta}_1)) \cos\hat{\theta}_5 + \cos\hat{\alpha}_1 \sin\hat{\theta}_1 \sin\hat{\theta}_5] \sin\hat{\alpha}_4 \qquad (50)$$

$$\hat{C}_1 = \sin\hat{\alpha}_3 \sin\hat{\theta}_3 \qquad (51)$$

$$\hat{D}_1 = \sin\hat{\alpha}_2 \cos\hat{\alpha}_3 + \cos\hat{\alpha}_2 \sin\hat{\alpha}_3 \cos\hat{\theta}_3. \qquad (52)$$

From eqns (47) and (48), we obtain

$$\sin\hat{\theta}_2 = \left(\frac{\hat{A}_1\hat{D}_1 + \hat{B}_1\hat{C}_1}{\hat{C}_1^2 + \hat{D}_1^2}\right),$$

$$\cos\hat{\theta}_2 = \left(\frac{\hat{A}_1\hat{C}_1 - \hat{B}_1\hat{D}_1}{\hat{C}_1^2 + \hat{D}_1^2}\right).$$

Then, the value of $\hat{\theta}_2$ can be calculated by

$$\hat{\theta}_2 = 2\,\mathrm{atan}\frac{\sin\hat{\theta}_2}{\cos\hat{\theta}_2 + 1}$$

$$= 2\,\mathrm{atan}\left(\frac{\hat{A}_1\hat{D}_1 + \hat{B}_1\hat{C}_1}{\hat{C}_1^2 + \hat{D}_1^2 + \hat{A}_1\hat{C}_1 - \hat{B}_1\hat{D}_1}\right). \qquad (53)$$

Joint values $\hat{\theta}_1$, $\hat{\theta}_2$, $\hat{\theta}_3$ and $\hat{\theta}_5$ are available at this point. The dual equations derived from elements (3,1) and (3,2) of dual matrix eqn (6) are in terms of $\hat{\theta}_1$, $\hat{\theta}_3$, $\hat{\theta}_4$, and $\hat{\theta}_5$. This leaves dual angle $\hat{\theta}_4$ as the only unknown left in the displacement analysis. Dual angle $\hat{\theta}_4$ is determined by using two equations in the same manner as in solving for $\hat{\theta}_2$. Elements (3,1) and (3,2) in eqn (6) give the following two equations:

$$\hat{A}_2 = \hat{D}_2 \sin\hat{\theta}_4 + \hat{C}_2 \cos\hat{\theta}_4 \qquad (54)$$

$$\hat{B}_2 = \hat{C}_2 \sin\hat{\theta}_4 - \hat{D}_2 \cos\hat{\theta}_4, \qquad (55)$$

respectively, where

$$\hat{A}_2 = (\sin\hat{\alpha}_5 \cos\hat{\alpha}_1 + \cos\hat{\alpha}_5 \sin\hat{\alpha}_1 \cos\hat{\theta}_1) \sin\hat{\theta}_5 + \sin\hat{\alpha}_1 \sin\hat{\theta}_1 \sin\hat{\theta}_5 \qquad (56)$$

$$\hat{B}_2 = (\cos\hat{\alpha}_5 \cos\hat{\alpha}_1 - \sin\hat{\alpha}_5 \sin\hat{\alpha}_1 \cos\hat{\theta}_1) \sin\hat{\alpha}_4 + [(\sin\hat{\alpha}_5 \cos\hat{\alpha}_1 + \cos\hat{\alpha}_5 \sin\hat{\alpha}_1 \cos\hat{\theta}_1) \cos\hat{\theta}_5 - \sin\hat{\alpha}_1 \sin\hat{\theta}_1 \sin\hat{\theta}_5] \cos\hat{\alpha}_4 \qquad (57)$$

$$\hat{C}_2 = \sin\hat{\alpha}_2 \sin\hat{\theta}_3 \qquad (58)$$

$$\hat{D}_2 = \cos\hat{\alpha}_2 \sin\hat{\alpha}_3 + \sin\hat{\alpha}_2 \cos\hat{\alpha}_3 \cos\hat{\theta}_3. \qquad (59)$$

Similar to $\hat{\theta}_2$ and value of $\hat{\theta}_4$ can be calculated by

$$\hat{\theta}_4 = 2\,\mathrm{atan}\frac{\sin\hat{\theta}_4}{\cos\hat{\theta}_4 + 1}$$

$$= 2\,\mathrm{atan}\left(\frac{\hat{A}_2\hat{D}_2 + \hat{B}_2\hat{C}_2}{\hat{C}_2^2 + \hat{D}_2^2 + \hat{A}_2\hat{C}_2 - \hat{B}_2\hat{D}_2}\right) \qquad (60)$$

In the above formulations for calculating θ_5, θ_3, $\hat{\theta}_2$, and $\hat{\theta}_4$, we have kept the succinct dual format whenever possible. For example, dual twist angles $\hat{\alpha}_i$ $(i = 1, \ldots, 5)$ are never partitioned into their real and imaginary parts in the formulation. Therefore, these formulas are very compact and their analytical sequences are easy to follow. Our solution procedure is significantly simpler than formulations based upon only real numbers presented by Yang.[6] Such simplification of formulations is possible only because we can easily translate these formulas into a $\mathrm{C^H}$ program to get both numerical and graphical results. In the next section, we will describe the software implementation of the formulas derived in this section.

3 A $\mathrm{C^H}$ PROGRAM FOR DISPLACEMENT ANALYSIS OF AN RCRCR MECHANISM BASED ON DUAL FORMULATION

Formulations for displacement analysis of the RCRCR mechanism presented in the previous section are quite general. Displacement analysis of many other spatial mechanisms can be performed in the same manner. In this section, we will demonstrate how dual formulations mixed with real numbers are handled in a $\mathrm{C^H}$ program by the following analysis problem.[6–9]

Problem 1: Given the following link parameters for an RCRCR mechanism shown in Fig. 1: $\hat{\alpha}_1 = 30° + \epsilon 10\,\mathrm{cm}$, $\hat{\alpha}_2 = 35° + \epsilon 40\,\mathrm{cm}$, $\hat{\alpha}_3 = 45° + \epsilon 30\,\mathrm{cm}$, $\hat{\alpha}_4 = 60° + \epsilon 25\,\mathrm{cm}$, $\hat{\alpha}_5 = 10° + \epsilon 32\,\mathrm{cm}$, $d_1 = 0$, $d_3 = 25\,\mathrm{cm}$, and $d_5 = 30\,\mathrm{cm}$, find all possible

```
/* RCRCR */
#include <linkage.h>
main()
{
  dual   alpha1,alpha2,alpha3,alpha4,alpha5,theta1,theta2,theta3,theta4,theta5;
  dual   A,B,C,D,E,F,G,A1,B1,C1,D1,A2,B2,C2,D2;
  float  theta1_i,d1,d2,d3,d4,d5,x,p[5],roots[4],tolerance = 0.001;
  int    i;                          /* branch number */
  FILE   *stream;

  stream = fopen("RCRCR.out","w");   /* open file RCRCR.out */
  fprintf(stream, "%% theta 1  theta 2  d2(cm)   theta 3  theta 4  d4(cm)   theta 5 \n");
  /*******************   INITIALIZE VARIABLES   ******************************/
  d1=0.0; d3=2.50; d5=3.00;          /* fixed displacements of the RCRCR mechanism */
  alpha1=dual(30*PI/180,1);          /* fixed dual angles of the RCRCR mechanism */
  alpha2=dual(35*PI/180,4.0); alpha3=dual(45*PI/180,3.0);
  alpha4=dual(60*PI/180,2.5); alpha5=dual(10*PI/180,3.2);
  for (theta1_i=60; theta1_i <=420; theta1_i++) /* 60 <= theta1 <=420 */
  {
    fprintf(stream,"%% \n");         /* add space between sets of data */
    theta1=dual(theta1_i*PI/180,d1);
    A=cos(alpha2)*cos(alpha3)-(cos(alpha4)*(cos(alpha5)*cos(alpha1)-
      sin(alpha5)*sin(alpha1)*cos(theta1)));                           /* (24) */
    B=sin(alpha2)*sin(alpha3);                                         /* (25) */
    C=sin(alpha1)*sin(alpha4)*sin(theta1);                             /* (26) */
    D=sin(alpha4)*(sin(alpha5)*cos(alpha1)+
      cos(alpha5)*sin(alpha1)*cos(theta1));                            /* (27) */
    E=(A+C*dual(0,d5)-D)/B;                                            /* (33) */
    F=(C+dual(0,d5)*D)/B;                                              /* (34) */
    G=(A-dual(0,d5)*C+D)/B;                                            /* (35) */
    p[4]=pow(real(E),2)+pow(dual(E),2)/(d3*d3)-1;                      /* (40) */
    p[3]=-4*(real(E)*real(F)+dual(E)*dual(F)/(d3*d3));                 /* (41) */
    p[2]=2*real(E)*real(G)+4*pow(real(F),2)+(2*dual(E)*dual(G)+
         4*pow(dual(F),2))/(d3*d3)-2;                                  /* (42) */
    p[1]=-4*(real(F)*real(G)+dual(F)*dual(G)/(d3*d3));                 /* (43) */
    p[0]=pow(real(G),2)+pow(dual(G),2)/(d3*d3)-1;                      /* (44) */
    roots[0] = 0; roots[1] = 0; roots[2] =0; roots[3] =0;
    zroots(p,4,roots,1);            /* Solve polynomial equation       (39) */
    for(i=0; i<=3; i++)             /* 4 branches of the RCRCR mechanism */
    {
      x = roots[i];
      theta3=dual(2*atan(dual(E*x*x-2*F*x+G)/
                  (-d3*real((E+1)*x*x-2*F*x+G+1))),d3);                /* (46) */
      theta5=dual((2*atan(x)),d5);
      A1=(cos(alpha4)*sin(alpha5)+sin(alpha4)*cos(alpha5)*cos(theta5))*
         sin(theta1)+sin(alpha4)*cos(theta1)*sin(theta5);              /* (49) */
      B1=(cos(alpha5)*sin(alpha1)+sin(alpha5)*cos(alpha1)*cos(theta1))*
         cos(alpha4)-((sin(alpha5)*sin(alpha1)-cos(alpha5)*cos(alpha1)*
         cos(theta1))*cos(theta5)+cos(alpha1)*sin(theta1)
         *sin(theta5))*sin(alpha4);                                    /* (50) */
      C1=sin(alpha3)*sin(theta3);                                      /* (51) */
      D1=sin(alpha2)*cos(alpha3)+cos(alpha2)*sin(alpha3)*cos(theta3);  /* (52) */
      theta2=2*atan((A1*D1+B1*C1)/(C1*C1+D1*D1+A1*C1-B1*D1));          /* (53) */
      A2=(sin(alpha5)*cos(alpha1)+cos(alpha5)*sin(alpha1)*cos(theta1))*
         sin(theta5)+sin(alpha1)*sin(theta1)*cos(theta5);              /* (56) */
      B2=(cos(alpha5)*cos(alpha1)-sin(alpha5)*sin(alpha1)*cos(theta1))*
         sin(alpha4)+((sin(alpha5)*cos(alpha1)+cos(alpha5)*sin(alpha1)*
         cos(theta1))*cos(theta5)-sin(alpha1)*sin(theta1)*
      sin(theta5))*cos(alpha4);                                          /* (57) */
    C2=sin(alpha2)*sin(theta3);                                          /* (58) */
    D2=cos(alpha2)*sin(alpha3)+sin(alpha2)*cos(alpha3)*cos(theta3);      /* (59) */
    theta4=2*atan((A2*D2+B2*C2)/(C2*C2+D2*D2+A2*C2-B2*D2));              /* (60) */
  /*********************    PRINT RESULTS    ***********************/
    if (real(theta5)*180/PI > 150)      /* adjust for clear plots */
      real(theta5) = real(theta5-2*PI); /* adjust real part of theta5 */
    if (real(theta4)*180/PI < 0)
      real(theta4) = real(theta4+2*PI); /* adjust real part of theta4 */
    fprintf(stream," %8.3f %8.3f %8.3f %8.3f %8.3f %8.3f %8.3f \n",
           theta1_i,real(theta2)*180/PI,dual(theta2),real(theta3)*180/PI,
           real(theta4)*180/PI,dual(theta4),real(theta5)*180/PI);
   }
  }
  fclose(stream);                       /* close file RCRCR.out */
  plotRCRCR("RCRCR.out");               /* make labels and plots 6 graphs */
  remove("RCRCR.out");                  /* remove RCRCR.out file  */
}
```

Program 1. A C^H program for displacement analysis of an RCRCR mechanism.

solutions for joint variables $\theta_2, d_2, \theta_3, \theta_4, d_4$, and θ_5 when θ_1 varies within its valid range. Plot joint variables vs input angle θ_1 when joint θ_1 is within its valid range.

Problem 1 is by no means a simple problem. As shown in the fourth order polynomial eqn (39), there are up to four solutions for θ_5 corresponding to each input angular position θ_1. Subsequently, there are four solutions for θ_3, $\hat{\theta}_2$, and $\hat{\theta}_4$ for a given θ_1. In the above derivation, we have not done any input rotability analysis. Unlike planar four-bar linkages, the determination of input rotability and valid input range is a very complicated task for spatial mechanisms. Analytical formulas for different mechanisms must be derived differently. However, the determination of the input rotability and valid input range of a spatial mechanism can be easily handled in the C^H programming language numerically through the symbolic notation **NaN**. Any invalid value in C^H is represented by the metanumber **NaN**. Therefore, we can compute joint variables when the input angle θ_1 makes a complete 360° rotation. If the input angle θ_1 is outside the valid input range, the values computed for all other joint variables will become **Nans**.

Problem 1 can be solved by a C^H program shown in Program 1. The sample numerical and graphical output of Program 1 are given in Table 1 and Fig. 2, respectively. Considering the complexity of the problem and completeness of the presented solutions, the size of the C^H program in Program 1 is remarkably small. Most importantly, it is straightforward and easy to understand. We will highlight some critical points

Table 1. Sample numerical output for joint variables θ_2, d_2, θ_3, θ_4, d_4 and θ_5 vs input θ_1 from Program 1

%	theta 1	theta 2	d2 (cm)	theta 3	theta 4	d4 (cm)	theta 5
%							
	60	NaN	NaN	NaN	NaN	NaN	NaN
	60	NaN	NaN	NaN	NaN	NaN	NaN
	60	NaN	NaN	NaN	NaN	NaN	NaN
	60	NaN	NaN	NaN	NaN	NaN	NaN
%							
	120	18.489	−7.330	82.794	107.196	−2.998	−0.422
	120	NaN	NaN	NaN	NaN	NaN	NaN
	120	NaN	NaN	NaN	NaN	NaN	NaN
	120	−146.343	−2.626	117.516	150.318	−5.743	111.508
%							
	180	−0.767	−7.427	99.701	111.059	0.278	−53.129
	180	96.887	−7.874	−115.760	224.105	5.733	−15.340
	180	21.153	−2.840	−107.312	247.612	0.220	39.036
	180	−146.419	−2.137	82.780	161.180	−6.813	78.626
%							
	360	−131.176	−8.737	149.812	147.538	4.116	−169.184
	360	NaN	NaN	NaN	NaN	NaN	NaN
	360	−146.977	−3.758	41.280	197.793	−0.535	−78.753
	360	NaN	NaN	NaN	NaN	NaN	NaN

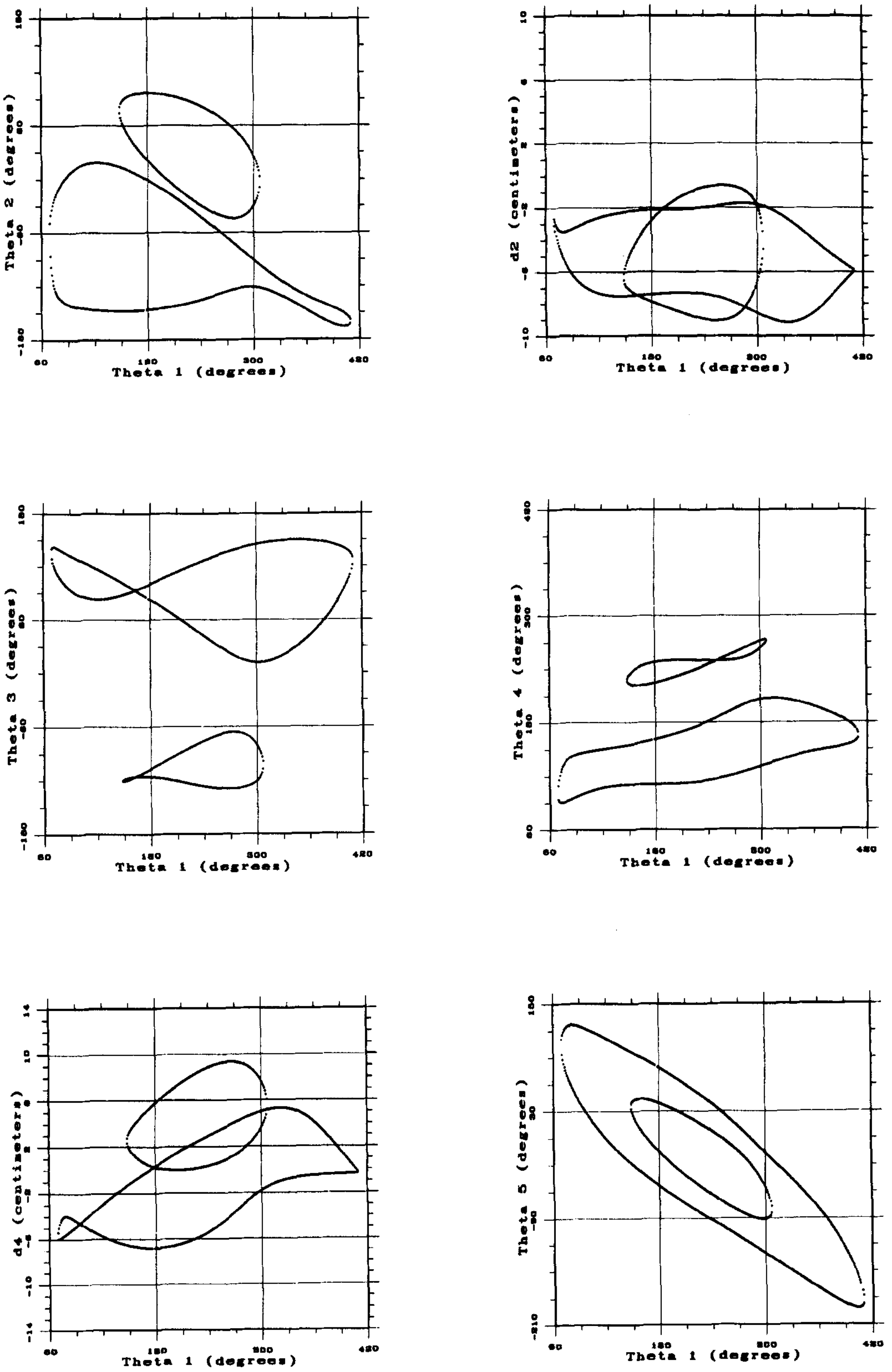

Fig. 2. Graphical output for joint variables θ_2, d_2, θ_3, θ_4, d_4 and θ_5 vs input θ_1 from Program 1.

for the implementation of Program 1 in the following discussion.

In Program 1, dependent variable θ_1 and joint displacements d_1, d_2, d_3, d_4, and d_5 are declared as floats theta_i, d1, d2, d3, d4, and d5; twist dual angles $\hat{\alpha}_1$, $\hat{\alpha}_2$, $\hat{\alpha}_3$, $\hat{\alpha}_4$, and $\hat{\alpha}_5$ are declared as dual variables alpha1, alpha2, alpha3, alpha4, and alpha5; dual angles $\hat{\theta}_1$, $\hat{\theta}_2$, $\hat{\theta}_3$, $\hat{\theta}_4$, and $\hat{\theta}_5$ are declared as theta1, theta2, theta3, theta4, and theta5; dual parameters $\hat{A}$, $\hat{B}$, $\hat{C}$, $\hat{D}$, $\hat{E}$, $\hat{F}$, and $\hat{G}$ are declared as dual variables A, B, C, D, E, F, and G; respectively. Coefficients p_i in the fourth order polynomial eqn (39) is treated as an element of the array p of five floats. Temporary dual numbers $\hat{A}_1$, $\hat{B}_1$, $\hat{C}_1$, $\hat{D}_1$, $\hat{A}_2$, $\hat{B}_2$, $\hat{C}_2$, and $\hat{D}_2$ are declared as dual variables A1, B1, C1, D1, A2, B2, and C2 respectively. For cross comparison, equation numbers for formulas derived in the previous section are given as comments at the end of the corresponding programming statements in Program 1.

Dual mathematical formulas are almost translated into C^H programming statements verbatim. The analytical sequences of the algorithm is very clear in Program 1. The input angle θ_1, defined as theta1_i, varies from 60° to 420° incremented with a step size of 1° by using a for-loop. The reason for choosing the range of 60° to 420° rather than 0° to 360° for θ_1 is for clarifying graphical output. This will be further explained later. Output of Program 1 is saved in the data file *RCRCR.out*. Sample data from the data file are given in Table 1. There are seven columns in Table 1. The first column is the input angle θ_1 and columns 2–7 are the corresponding output angles or displacements for θ_2, d_2, θ_3, θ_4, d_4, and θ_5. For each input angle θ_1, there are up to four possible solutions for these joint variables; each corresponds to one of four solutions for the fourth order polynomial eqn (39).

Solutions for χ in eqn (39) are obtained by the function zroots() using a function file *zroots.ff*. Function zroots() in C^H is treated as if it was a system built-in function like sin() or cos(). This function can compute all solutions of a polynomial equation using complex numbers. There are four input arguments for this function. The first argument is a complex array that contains the coefficients of the polynomial. The second one is an integer that indicates the order of the polynomial. The third one is a complex array that returns the roots of the polynomial. The last one is a boolean number. If it is true, the algorithm will be polished with more accuracy. The algorithm for solving a polynomial equation inside function zroots() is based on Laguerre's method. The function zroots() will guarantee the delivery of n complex roots for an nth order polynomial equation. More details about the algorithm for this function can be found in Press.[11] In C^H, real and complex numbers can be converted either explicitly or implicitly. When a real number is converted to a complex number, the imaginary part of the resultant complex number is zero. When a complex number is converted to a real number, if the imaginary part of the complex number is identically zero, the real number takes the real part of the complex number, otherwise, the real number becomes **NaN** as described in Ref. 3. For the function call of zroots(P,4,roots,1) in Program 1, array p of five floats for coefficients of the polynomial and initial guesses of the algorithm passed to the function through array of 4 float roots are cast to complex arrays implicitly by the system. Similarly, the complex roots obtained by function zroots() are cast to real roots implicitly and are saved in array roots of four floats. If a root of the polynomial equation is a complex number, it becomes **NaN**. This indicates that no solution exists for this branch corresponding to the given input angle θ_1. For example, there are only two sets of solutions for output joint variables when θ_1 is at 120° or 360° as shown in Table 1. There is no solution when θ_1 is 60° or 420°. The graphical output from Program 1 is shown in Fig. 2. There are six graphs that are produced by function plotRCRCR() using the data saved in file RCRCR.out. The function plotRCRCR() contains labels and ranges of x and y coordinates. It invokes two function files called separate() and plotxyrf(). The function separate () is used to read in and separate the RCRCR.out data file into six separate files. Each file has two columns. The first column of data forms the x-coordinates whereas the data in the second column give the y-coordinates. As the data are being separated, lines with a leading symbol '%' are ignored. In other words, the symbol '%' can be used to comment a data file as shown in Table 1. Also, lines containing the value **NaN** will also be ignored during separation. Based upon data in these six files, the function plotxyrf() plots six graphs with appropriate x and y labels as well as valid ranges and steps.

Each graph shows a relationship between the input angle θ_1 and a corresponding output angle or displacement. The graphs are continuous in various ranges. According to the formulas derived in the previous section, all joint angles θ_2, θ_3, θ_4, and θ_5 are computed within the domain from −180° to 180°. To show the continuous graphs, the input range for angle θ_1 and output ranges for θ_4 to θ_5 are adjusted. θ_1 varies from 60° to 420° instead of from 0° to 360°. No adjustment is made to θ_2 and θ_3. They are continuous in the range of −180° to 180°. θ_4 is adjusted to vary between 60° to 420° and θ_5 is in the range from −210° to 150°. These adjustments are made in the 'PRINT RESULTS' section of Program 1. The input–output relationship between the input angle θ_1 and each of the output angles and displacements form two closed loops. One loop has an input range of $69.350° \leq \theta_1 \leq 410.471°$ and the other loop is within the input range of $148.788° \leq \theta_1 \leq 308.299°$. In the range where both loops exist, there are four solutions for each output joint variable. For example, there are four solutions for θ_2 when $\theta_1 = 180°$. They are −0.767°, 96.887°, 21.153°, and −146.419° as shown in Table 1. In the ranges where only one of the loops exist, there are two solutions only. For example, θ_2 can be 18.489° or −146.343° when $\theta_1 = 120°$, as shown in Table 1. In this case, **Nans** are used as symbolic values for the other two non-existing solutions.

4 CONCLUSIONS

Dual numbers are handled as a built-in data type in the C^H programming language. Dual formulas can be easily translated into C^H programming statements in the same manner as real or complex formulas. It is shown in this paper that formulations for displacement analysis of the RCRCR mechanism under the C^H programming paradigm are significantly simpler than those derived for implementation in programming languages without dual data type. A remarkably small C^H program can give a complete numerical and graphical displacement solution of an RCRCR mechanism. The C^H program is very simple and easy to implement. Although the presentation on the derivation of formulas and the developed C^H program based on these formulas are specific for displacement analysis of the RCRCR mechanism, the ideas, solution procedures, and programming techniques presented in this paper are general. They are applicable to analysis of a wide variety of spatial mechanisms.

ACKNOWLEDGEMENTS

This work was supported in part by the National Science Foundation under Grant DMII-9309207 and by the University of California, Davis through an undergraduate instructional improvement grant, a faculty research grant, and a junior faculty research fellowship.

REFERENCES

1. Cheng, H. H. Programming with dual numbers and its applications in mechanism design. *Engineering with Computers*, 1994, **10** (4), 212–29.
2. Cheng, H. H. Scientific computing in the C^H programming language. *Scientific Programming*, 1993, **2** (3), 49–75.
3. Cheng, H. H. Handling of complex numbers in the C^H programming language. *Scientific Programming*, 1993, **2** (3), 77–106.
4. Cheng, H. H. Extending C and FORTRAN for design automation. *ASME Trans., J. Mech. Design*, 1995, **117**(3), 390–95.
5. Dimentberg, F. M. The determination of the positions of spatial mechanisms. *Izdatel'stvo Akademii Nauk*, Moscow, USSR, 1950.
6. Yang, A. T. Displacement analysis of spatial five-link mechanisms using (3 × 3) matrices with dual-number elements. *J. Engng Indust.*, 1969, **91**, 152–57.
7. Yuan, S. C. Displacement analysis of the RCRCR five-link spatial mechanism. *J. Mech.*, 1970, **6**, 119–34.
8. Duffy, J. & Habib-Olahi, H. Y. A displacement analysis of spatial five-link 3R-2C mechanisms; Part I: On the closures of the RCRCR mechanism. *J. Mech.*, 1970, **6**, 289–301.
9. Duffy, J. *Analysis of Mechanisms and Robot Manipulators*. Edward Arnold, 1980, pp. 235–42.
10. Gupta, K. C. Spatial kinematics. ME 409 Lecture Notes. Department of Mechanical Engineering, University of Illinois at Chicago, 1989.
11. Press, W. H., Flannery, B. P., Teukolshy, S. A. & Vetterling, W. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, NY, 1988.