COMPUTER-AIDED DISPLACEMENT ANALYSIS OF SPATIAL MECHANISMS

Sean Thompson
Department of Mechanical
and Aeronautical Engineering
University of California
Davis, California

Harry H. Cheng[1]
Department of Mechanical
and Aeronautical Engineering
University of California
Davis, California

ABSTRACT[1]

Recently, Cheng (1993) introduced the C^H programming language. C^H is designed to be a superset of ANSI C with all programming features of FORTRAN. Many programming features in C^H are specifically designed and implemented for design automation. Handling dual number as a basic built-in data type in the language is one example. 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 within the programming paradigm of C^H. 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 programming paradigm of C^H through a 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.

INTRODUCTION

Dual number is a powerful mathematical tool for analysis of spatial mechanisms. It has attracted attention of generations of kinematicians [1, 10-18, 20, 22-24]. Dual number was first introduced by Clifford [9] to study non-Euclidean geometry. Study treated a dual number as a dual angle to specify the relation between two lines in space [21]. Dimentberg [12], Denavit [10], and Yang and Freudenstein [22-23] pioneered analysis of spatial mechanisms using dual numbers. Dual number is widely used for analytical treatment of spatial mechanisms nowadays [13-18, 20, 22-24]. 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 in thinking 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 the 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, as pointed out by Cheng [5-7].

Recently, Cheng introduced the C^H programming language [2-8]. C^H is a superset of ANSI C with incorporation of all programming features of FORTRAN. Due to our re-

[1] Address all correspondence to this author.

search interests, $\mathrm{C^H}$ 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 $\mathrm{C^H}$ 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 $\mathrm{C^H}$ is simpler than any other computer programming language. The linguistic features related to dual numbers in $\mathrm{C^H}$ have been presented by Cheng [5-7]. Dual numbers are treated in the same manner as real or complex numbers in $\mathrm{C^H}$. The simplicity of programming with dual numbers in $\mathrm{C^H}$ has been demonstrated by kinematic analysis of an RCCC mechanism in [5, 7]. Translating dual formulas for displacement analysis of the RCCC mechanism into a $\mathrm{C^H}$ program is so simple that some researchers have commented, based upon their reading of [5-7], that it is useful to demonstrate how the $\mathrm{C^H}$ 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 $\mathrm{C^H}$ are remarkably simple for analysis of spatial mechanisms that do not have closed-form solutions. Programming with dual numbers in $\mathrm{C^H}$ 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 in the programming paradigm of $\mathrm{C^H}$.

The RCRCR spatial mechanism will be used to demonstrate the simplicity of formulation and programming with dual numbers under the programming paradigm of $\mathrm{C^H}$. The displacement analysis of the RCRCR mechanism has been studied extensively over the last two decades. Based upon dual numbers, Dimentberg [12] 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 transformation matrices, Yang [23] 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-pronecated [23]. Using line coordinates with dual numbers, Yuan [24] presented complete solutions for the displacement analysis of the same mechanism. Using spherical trigonometry with dual numbers, Duffy and Habib-Olahi [13-14] 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 $\mathrm{C^H}$, 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 [23] and obtain Duffy and Habib-Olahi's result on eight solutions for the RCRCR in an alternate form. Because dual formulas can be readily translated into a $\mathrm{C^H}$ program, formulations presented in this paper is very simple and easy to follow. Most importantly, based upon these dual formulas, we can easily obtain both numerical and graphical results in the $\mathrm{C^H}$ programming language.

DUAL TRANSFORMATION MATRIX

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 $X'Y'Z'O'$ can be represented by the following dual transformation matrix [5, 7].

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

where dual angle $\widehat{\theta}_i = \theta_i + \varepsilon d_i$. 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

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

where dual angle $\widehat{\alpha}_i = \alpha_i + \varepsilon a_i$. 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.

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

where $\widehat{\mathbf{\Theta}}_i$ and $\widehat{\mathbf{\Lambda}}_i$ are given in equation (1) and equation (2), respectively.

DUAL FORMULATION FOR DISPLACEMENT ANALYSIS OF THE RCRCR MECHANISM

In this section, a displacement analysis of the RCRCR mechanism shown in Figure 1 will be used to demonstrate how to formulate dual equations for proper implementation in the $\mathrm{C^H}$ programming language. The analysis of this mechanism consists of determining the revolute joint values θ_3 and θ_5 and cylindrical joint values $\widehat{\theta}_2$ and $\widehat{\theta}_4$ for a given input angle θ_1. For the RCRCR mechanism, link parameters $\widehat{\alpha}_i = \alpha_i + \varepsilon a_i (i = 1, \cdots, 5)$, d_1, d_3, and d_5 are fixed. The parameters θ_i, α_i, a_i, and d_i are shown in Figure 1. In Figure 1 the revolute axes Z_1 and Z_5 are rotated with respect to the ground. In other words, the coordinate systems

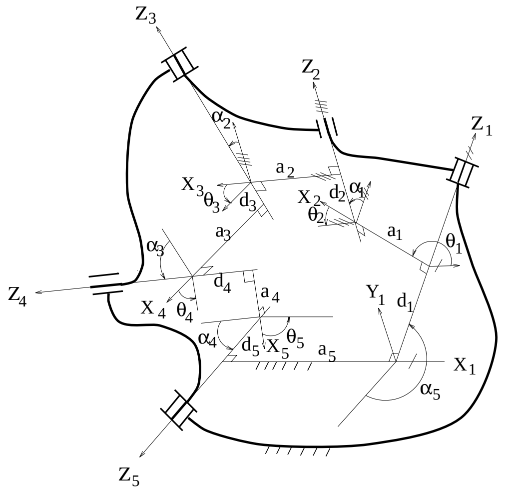

FIGURE 1: THE RCRCR MECHANISM.

$O_1X_1Y_1Z_1$ and $O_5X_5Y_5Z_5$ are attached to input and output links, respectively.

Formulas for calculating θ_3, θ_5, $\widehat{\theta}_2$, and $\widehat{\theta}_4$ in terms of the input angle θ_1 can be derived as follows. According to the dual transformation matrix $\widehat{\mathbf{D}}_i$ in equation (3), we can write the loop-closure equation for the RCRCR mechanism shown in Figure 1 as

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

where transformation matrices $\widehat{\mathbf{\Theta}}_i$ and $\widehat{\mathbf{\Lambda}}_i$ are defined in equations (1) and (2), respectively, and $\mathbf{I}$ is a 3x3 identity matrix. Equation (4) can be cyclically permutated [15, 23]. The goal of the reformulation is to create a loop-closure matrix equation that simplifies the algebraic manipulations needed. Reformulation of the loop-closure equation (4) for different five-link spatial mechanisms is determined by knowledge of the characteristics of the resultant matrix equation. Each element of the matrix equation (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 [23].

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

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

$$\begin{bmatrix} \widehat{a}\cos\widehat{\theta}_5 - \widehat{d}\sin\widehat{\theta}_5 & \widehat{o}(\widehat{a}\sin\widehat{\theta}_5 + \widehat{d}\cos\widehat{\theta}_5) - \widehat{gn} & \widehat{n}(\widehat{a}\sin\widehat{\theta}_5 + \widehat{d}\cos\widehat{\theta}_5) + \widehat{go} \\ \widehat{b}\cos\widehat{\theta}_5 - \widehat{e}\sin\widehat{\theta}_5 & \widehat{o}(\widehat{b}\sin\widehat{\theta}_5 + \widehat{e}\cos\widehat{\theta}_5) - \widehat{hn} & \widehat{n}(\widehat{b}\sin\widehat{\theta}_5 + \widehat{e}\cos\widehat{\theta}_5) + \widehat{ho} \\ \widehat{c}\cos\widehat{\theta}_5 - \widehat{f}\sin\widehat{\theta}_5 & \widehat{o}(\widehat{c}\sin\widehat{\theta}_5 + \widehat{f}\cos\widehat{\theta}_5) - \widehat{in} & \widehat{n}(\widehat{c}\sin\widehat{\theta}_5 + \widehat{f}\cos\widehat{\theta}_5) + \widehat{io} \end{bmatrix} =$$

$$\begin{bmatrix} (\cos\widehat{\theta}_2\cos\widehat{\theta}_4 - (\widehat{k}(\sin\widehat{\theta}_3\cos\widehat{\theta}_4 + \widehat{m}\sin\widehat{\theta}_3\sin\widehat{\theta}_4) - \widehat{kl}\sin\widehat{\theta}_4)\sin\widehat{\theta}_2 & \sin\widehat{\theta}_4\cos\widehat{\theta}_4 - ((\widehat{m}\cos\widehat{\theta}_3\cos\widehat{\theta}_4 - \sin\widehat{\theta}_3\sin\widehat{\theta}_4)\widehat{k} - \widehat{jl}\cos\widehat{\theta}_4)\sin\widehat{\theta}_2 & \widehat{l}\cos\widehat{\theta}_2\sin\widehat{\theta}_3 + (\widehat{kl}\cos\widehat{\theta}_3 + \widehat{jm})\sin\widehat{\theta}_2 \\ (\sin\widehat{\theta}_2\cos\widehat{\theta}_4 + (\widehat{k}(\sin\widehat{\theta}_3\cos\widehat{\theta}_4 + \widehat{m}\sin\widehat{\theta}_3\sin\widehat{\theta}_4) - \widehat{kl}\sin\widehat{\theta}_4)\cos\widehat{\theta}_2 & \sin\widehat{\theta}_4\cos\widehat{\theta}_4 + ((\widehat{m}\cos\widehat{\theta}_3\cos\widehat{\theta}_4 - \sin\widehat{\theta}_3\sin\widehat{\theta}_4)\widehat{k} - \widehat{jl}\cos\widehat{\theta}_4)\cos\widehat{\theta}_2 & \widehat{l}\sin\widehat{\theta}_2\sin\widehat{\theta}_3 + (\widehat{kl}\cos\widehat{\theta}_3 + \widehat{jm})\cos\widehat{\theta}_2 \\ \widehat{j}(\sin\widehat{\theta}_3\cos\widehat{\theta}_4 + \widehat{m}\sin\widehat{\theta}_3\sin\widehat{\theta}_4) + \widehat{kl}\sin\widehat{\theta}_4 & -\widehat{j}(\sin\widehat{\theta}_3\sin\widehat{\theta}_4 - \widehat{m}\cos\widehat{\theta}_3\cos\widehat{\theta}_4) + \widehat{kl}\cos\widehat{\theta}_4 & \widehat{km} - \widehat{jl}\cos\widehat{\theta}_3 \end{bmatrix} \tag{6}$$

where

$$\begin{aligned}
\widehat{a} &= \cos\widehat{\theta}_1 && (7)\\
\widehat{b} &= -\cos\widehat{\alpha}_1\sin\widehat{\theta}_1 && (8)\\
\widehat{c} &= \sin\widehat{\alpha}_1\sin\widehat{\theta}_1 && (9)\\
\widehat{d} &= \cos\widehat{\alpha}_5\sin\widehat{\theta}_1 && (10)\\
\widehat{e} &= \cos\widehat{\alpha}_1\cos\widehat{\theta}_1\cos\widehat{\alpha}_5 - \sin\widehat{\alpha}_5\sin\widehat{\alpha}_1 && (11)\\
\widehat{f} &= -\sin\widehat{\alpha}_1\cos\widehat{\theta}_1\cos\widehat{\alpha}_5 - \cos\widehat{\alpha}_1\sin\widehat{\alpha}_5 && (12)\\
\widehat{g} &= \sin\widehat{\theta}_1\sin\widehat{\alpha}_5 && (13)\\
\widehat{h} &= \cos\widehat{\alpha}_1\cos\widehat{\theta}_1\sin\widehat{\alpha}_5 + \sin\widehat{\alpha}_1\cos\widehat{\alpha}_5 && (14)\\
\widehat{i} &= -\sin\widehat{\alpha}_1\cos\widehat{\theta}_1\sin\widehat{\alpha}_5 + \cos\widehat{\alpha}_1\cos\widehat{\alpha}_5 && (15)\\
\widehat{j} &= \sin\widehat{\alpha}_2 && (16)\\
\widehat{k} &= \cos\widehat{\alpha}_2 && (17)\\
\widehat{l} &= \sin\widehat{\alpha}_3 && (18)\\
\widehat{m} &= \cos\widehat{\alpha}_3 && (19)\\
\widehat{n} &= \sin\widehat{\alpha}_4 && (20)\\
\widehat{o} &= \cos\widehat{\alpha}_4. && (21)
\end{aligned}$$

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

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

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

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

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

where

$$\widehat{A} = \cos\widehat{\alpha}_2\cos\widehat{\alpha}_3 -$$

$$\cos\widehat{\alpha}_4(\cos\widehat{\alpha}_3\cos\widehat{\alpha}_1+\sin\widehat{\alpha}_5\sin\widehat{\alpha}_1\cos\widehat{\theta}_1) \quad (24)$$

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

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

$$\widehat{D} = \sin\widehat{\alpha}_4(\sin\widehat{\alpha}_3\cos\widehat{\alpha}_1+\cos\widehat{\alpha}_5\sin\widehat{\alpha}_1\cos\widehat{\theta}_1) \quad (27)$$

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

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

Details on C$^{\mathrm{H}}$ programming for the RCRCR mechanism will be described in the next section. There are two unknowns θ_3 and θ_5 in equation (23). These two unknowns are real parts of dual angles $\widehat{\theta}_3$ and $\widehat{\theta}_5$. Therefore, it is inevitable at this point to single out θ_3 and θ_5 from dual numbers $\widehat{\theta}_3$ and $\widehat{\theta}_5$, 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

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

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

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

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

where

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

and making additional substitutions,

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

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

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

equation (23) becomes

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

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 and accent grave "ˋ" stands for the real part of the dual number, whereas a dual variable topped with an accent acute "ˊ" stands for the imaginary part. For example, if $\widehat{d}=x+\varepsilon y$, 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 $(i=0,\cdots,4)$ of the polynomial equation (39) can be easily computed in the C$^{\mathrm{H}}$ programming language. For example, coefficient p_4 in equation (40) can be programmed in C$^{\mathrm{H}}$ as

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

The solution for χ in equation (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 equation (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 equation (36).

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

If d_3 takes both positive and negative values, there are two solutions for $\widehat{\theta}_3$ from equation (45) corresponding to a given $\widehat{\theta}_5$. There are up to four solutions for $\widehat{\theta}_5$ for a given θ_1. Therefore, there are up to eight solutions for the RCRCR mechanism as pointed out by Duffy and Habib-Olahi [13]. However, for a given RCRCR mechanism, the sign of d_3 is fixed. Therefore, only one of two solutions from equation (45) is valid. This valid solution can be obtained unambiguously by the formula $\theta=2\mathrm{atan}(\sin\theta/(\cos\theta+1))$ [15].

$$\begin{aligned}\theta_3 &= 2\mathrm{atan}\left(\frac{\sin\theta_3}{\cos\theta_3+1}\right)\\ &= 2\mathrm{atan}\left(\frac{\grave{E}\chi^2-2\grave{F}\chi+\grave{G}}{-d_3(\acute{E}+1)\chi^2-2\acute{F}\chi+\acute{G}+1}\right)\end{aligned} \quad (46)$$

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

At this point, $\widehat{\theta}_1$, $\widehat{\theta}_3$ and $\widehat{\theta}_5$ all are known. The dual equations derived from elements (1,3) and (2,3) of dual matrix equation (6) are in terms of $\widehat{\theta}_1$, $\widehat{\theta}_2$, $\widehat{\theta}_3$, and $\widehat{\theta}_5$. 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

$$\begin{aligned}\widehat{A}_1 &= \widehat{D}_1 \sin\widehat{\theta}_2 + \widehat{C}_1 \cos\widehat{\theta}_2 &(47)\\ \widehat{B}_1 &= \widehat{C}_1 \sin\widehat{\theta}_2 - \widehat{D}_1 \cos\widehat{\theta}_2, &(48)\end{aligned}$$

respectively, where

$$\begin{aligned}\widehat{A}_1 &= (\cos\widehat{\alpha}_4 \sin\widehat{\alpha}_5 + \sin\widehat{\alpha}_4 \cos\widehat{\alpha}_5 \cos\widehat{\theta}_5) \sin\widehat{\theta}_1\\ &\quad + \sin\widehat{\alpha}_4 \cos\widehat{\theta}_1 \sin\widehat{\theta}_5 &(49)\\ \widehat{B}_1 &= (\cos\widehat{\alpha}_5 \sin\widehat{\alpha}_1 + \sin\widehat{\alpha}_5 \cos\widehat{\alpha}_1 \cos\widehat{\theta}_1) \cos\widehat{\alpha}_4\\ &\quad -[(\sin\widehat{\alpha}_5 \sin\widehat{\alpha}_1 - \cos\widehat{\alpha}_5 \cos\widehat{\alpha}_1 \cos\widehat{\theta}_1)) \cos\widehat{\theta}_5\\ &\quad + \cos\widehat{\alpha}_1 \sin\widehat{\theta}_1 \sin\widehat{\theta}_5] \sin\widehat{\alpha}_4 &(50)\\ \widehat{C}_1 &= \sin\widehat{\alpha}_3 \sin\widehat{\theta}_3 &(51)\\ \widehat{D}_1 &= \sin\widehat{\alpha}_2 \cos\widehat{\alpha}_3 + \cos\widehat{\alpha}_2 \sin\widehat{\alpha}_3 \cos\widehat{\theta}_3. &(52)\end{aligned}$$

$\widehat{\theta}_2$ can be calculated either by equation (47) or by equation (48). For example, from equation (47), we get

$$\hat{\theta}_2 = 2\text{atan}\left(\frac{-\widehat{D}_1 \pm \sqrt{\widehat{D}_1^2 + \widehat{C}_1^2 - \widehat{A}_1^2}}{-\widehat{A}_1 - \widehat{C}_1}\right) \quad (53)$$

which can be implemented in $\mathrm{C^H}$ as

```
theta2 = 2*atan(-D1+sqrt(D1*D1+C1*C1-A1*A1))
         /(-A1-C1);
theta2 = 2*atan(-D1-sqrt(D1*D1+C1*C1-A1*A1))
         /(-A1-C1);
```

But only one of solutions for $\widehat{\theta}_2$ in equation (53) is valid. The valid solution can be determined by the secondary equation (48). In our software implementation, we compute the residual error of equation (48) for a calculated $\widehat{\theta}_2$ from equation (53). $\widehat{\theta}_2$ that gives a residual error smaller than a preset tolerance is selected as the valid solution.

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

$$\begin{aligned}\widehat{A}_2 &= \widehat{D}_2 \sin\widehat{\theta}_4 + \widehat{C}_2 \cos\widehat{\theta}_4 &(54)\\ \widehat{B}_2 &= \widehat{C}_2 \sin\widehat{\theta}_4 - \widehat{D}_2 \cos\widehat{\theta}_4, &(55)\end{aligned}$$

respectively, where

$$\begin{aligned}\widehat{A}_2 &= (\sin\widehat{\alpha}_5 \cos\widehat{\alpha}_1 + \cos\widehat{\alpha}_5 \sin\widehat{\alpha}_1 \cos\widehat{\theta}_1) \sin\widehat{\theta}_5\\ &\quad + \sin\widehat{\alpha}_1 \sin\widehat{\theta}_1 \sin\widehat{\theta}_5 &(56)\\ \widehat{B}_2 &= (\cos\widehat{\alpha}_5 \cos\widehat{\alpha}_1 - \sin\widehat{\alpha}_5 \sin\widehat{\alpha}_1 \cos\widehat{\theta}_1) \sin\widehat{\alpha}_4\\ &\quad +[(\sin\widehat{\alpha}_5 \cos\widehat{\alpha}_1 + \cos\widehat{\alpha}_5 \sin\widehat{\alpha}_1 \cos\widehat{\theta}_1) \cos\widehat{\theta}_5\\ &\quad - \sin\widehat{\alpha}_1 \sin\widehat{\theta}_1 \sin\widehat{\theta}_5] \cos\widehat{\alpha}_4 &(57)\\ \widehat{C}_2 &= \sin\widehat{\alpha}_2 \sin\widehat{\theta}_3 &(58)\\ \widehat{D}_2 &= \cos\widehat{\alpha}_2 \sin\widehat{\alpha}_3 + \sin\widehat{\alpha}_2 \cos\widehat{\alpha}_3 \cos\widehat{\theta}_3. &(59)\end{aligned}$$

Two values for $\widehat{\theta}_4$ can be calculated by using either equation (54) or equation (55). Using equation (54), we get

$$\hat{\theta}_4 = 2\text{atan}\left(\frac{-\widehat{D}_2 \pm \sqrt{\widehat{D}_2^2 + \widehat{C}_2^2 - \widehat{A}_2^2}}{-\widehat{A}_2 - \widehat{C}_2}\right). \quad (60)$$

The valid value for $\widehat{\theta}_4$ can be determined by the secondary equation (55).

In the above formulations for calculating θ_5, θ_3, $\widehat{\theta}_2$, and $\widehat{\theta}_4$, we have kept the succinct dual format whenever possible. For example, dual twist angles $\widehat{\alpha}_i$ ($i = 1, \cdots, 5$) 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 [23]. 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.

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 [13-14,23-24].

> Problem 1: Given the following link parameters for an RCRCR mechanism shown in Figure 1: $\widehat{\alpha}_1 = 30° + \varepsilon 10$ cm, $\widehat{\alpha}_2 = 35° + \varepsilon 40$ cm, $\widehat{\alpha}_3 = 45° + \varepsilon 30$ cm, $\widehat{\alpha}_4 = 60° + \varepsilon 25$ cm, $\widehat{\alpha}_5 = 10° + \varepsilon 32$ cm, $d_1 = 0, d_3 = 25$ cm, and $d_5 = 30$ cm, find all possible 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 versus 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 equation (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, $\widehat{\theta}_2$, and $\widehat{\theta}_4$ for a given θ_1. In the above derivation, we haven't done any input rotability analysis. Unlike planar four-bar linkages, the determination of input rotability and valid input range

PROGRAM 1: A C^H PROGRAM FOR DISPLACEMENT ANALYSIS OF AN RCRCR MECHANISM.

```
/* RCRCR */
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[5],tolerance = 0.001;
  complex roots[5]
  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(imaginary(E),2)/(d3*d3)-1;               /# (40)
    p[3]=-4*(real(E)*real(F)+imaginary(E)*imaginary(F)/(d3*d3));     /# (41)
    p[2]=2*real(E)*real(G)+4*pow(real(F),2)+(2*imaginary(E)*imaginary(G)+
         4*pow(imaginary(F),2))/(d3*d3)-2;                           /# (42)
    p[1]=-4*(real(F)*real(G)+imaginary(F)*imaginary(G)/(d3*d3));     /# (43)
    p[0]=pow(real(G),2)+pow(imaginary(G),2)/(d3*d3)-1;               /# (44)
    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(imaginary(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((-D1-sqrt(D1*D1+C1*C1-A1*A1))/(-A1-C1));         /# (53)
      if (abs(C1*sin(theta2)-D1*cos(theta2)-B1) > tolerance)
        theta2=2*atan((-D1+sqrt(D1*D1+C1*C1-A1*A1))/(-A1-C1));       /# (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((-D2+sqrt(D2*D2+C2*C2-A2*A2))/(-A2-C2));              /# (60)
     if (abs(C2*sin(theta4)-D2*cos(theta4)-B2) > tolerance)
       theta4=2*atan((-D2-sqrt(D2*D2+C2*C2-A2*A2))/(-A2-C2));            /# (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,imaginary(theta2),real(theta3)*180/PI,
           real(theta4)*180/PI,imaginary(theta4),real(theta5)*180/PI);
   }
  }
  fclose(stream);                        /# close file RCRCR.out
  plotRCRCR("RCRCR.out");                /# Makes labels and plots 6 graphs
}
```

TABLE 1: SAMPLE NUMERICAL OUTPUT FOR JOINT VARIABLES θ_2, d_2, θ_3, θ_4, d_4, AND θ_5 VERSUS 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	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
	120	18.489	-7.330	82.794	107.196	-2.998	-0.422
%							
	180	-146.419	-2.137	82.780	161.180	-6.813	78.626
	180	21.153	-2.840	-107.312	247.612	0.220	39.036
	180	96.887	-7.874	-115.760	224.105	5.733	-15.340
	180	-0.767	-7.427	99.701	111.059	0.278	-53.129
%							
	240	-135.796	-1.998	42.565	180.722	-5.417	38.355
	240	-29.049	-0.732	-75.879	249.613	0.321	7.678
	240	82.004	-8.971	-126.521	244.676	8.864	-63.397
	240	-41.643	-7.397	126.398	115.033	2.905	-92.071
%							
	300	-121.255	-1.790	11.814	205.620	-1.825	-16.539
	300	-30.551	-1.780	-80.543	263.154	3.552	-54.732
	300	27.019	-6.720	-120.878	271.843	7.713	-91.292
	300	-90.487	-8.573	145.319	130.602	5.005	-129.962
%							
	360	-146.978	-3.758	41.281	197.792	-0.535	-78.752
	360	NaN	NaN	NaN	NaN	NaN	NaN
	360	NaN	NaN	NaN	NaN	NaN	NaN
	360	-131.176	-8.737	149.812	147.538	4.116	-169.184
%							
	420	NaN	NaN	NaN	NaN	NaN	NaN
	420	NaN	NaN	NaN	NaN	NaN	NaN
	420	NaN	NaN	NaN	NaN	NaN	NaN
	420	NaN	NaN	NaN	NaN	NaN	NaN

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 $\mathrm{C^H}$ programming language numerically through the symbolic notation **NaN**. Any invalid value in $\mathrm{C^H}$ is represented by the metanumber **NaN**. Therefore, we can compute joint variables when the input angle θ_1 makes a complete 360 degree rotation. If the input angle θ_1 is outside the valid input range, the values computed for all other joint variables will become **NaN**s.

Problem 1 can be solved by a $\mathrm{C^H}$ program shown in Program 1. The sample numerical and graphical output of Program 1 are given in Table 1 and Figure 2, respectively. Considering the complexity of the problem and completeness of the presented solutions, the size of the $\mathrm{C^H}$ program in Program 1 is remarkably small. Most importantly, it is straightforward and easy to understand. We will highlight some critical points 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 `theta1_i`, `d1`, `d2`, `d3`, `d4`, and `d5`; twist dual angles $\widehat{\alpha}_1$, $\widehat{\alpha}_2$, $\widehat{\alpha}_3$, $\widehat{\alpha}_4$, and $\widehat{\alpha}_5$ are declared as dual variables `alpha1`, `alpha3`, `alpha4`, and `alpha5`; dual angles $\widehat{\theta}_1$, $\widehat{\theta}_2$, $\widehat{\theta}_3$, $\widehat{\theta}_4$, and $\widehat{\theta}_5$ are declared as `theta1`, `theta2`, `theta3`, `theta4`, and `theta5`; dual parameters $\widehat{A}$, $\widehat{B}$, $\widehat{C}$, $\widehat{D}$, $\widehat{E}$, $\widehat{F}$, and $\widehat{G}$ are declared as dual variables `A`, `B`, `C`, `D`, `E`, `F`, and `G`; respectively. Coefficients p_i in the fourth order polynomial equation (39) is treated as an element of the array `p` of five floats. Temporary dual numbers $\widehat{A}_1$, $\widehat{B}_1$, $\widehat{C}_1$, $\widehat{D}_1$, $\widehat{A}_2$, $\widehat{B}_2$, $\widehat{C}_2$, and $\widehat{D}_2$ are declared as dual variables A1, B1, C1, D1, A2, B2, and C2 respectively. In addition to the ANSI C standard conforming comment symbols "`/*`" and "`*/`" that can be used to comment out a segment of $\mathrm{C^H}$ code, in $\mathrm{C^H}$ the symbol "`/#`" can also be used to comment out the code terminated at the end of the line as shown in Program 1. For cross comparison, equation numbers for formulas derived in the previous section are given as comments at the end of the corresponding assignment statements in Program 1.

Dual mathematical formulas are almost translated into $\mathrm{C^H}$ 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 two to seven 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 equation (39).

Solution for χ in equation (39) are obtained by the function `zroots()` using a function file *zroots.ff*. Function `zroots()` in $\mathrm{C^H}$ 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 [19]. In $\mathrm{C^H}$, 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** [4]. For the function call of `zroots(P,4,roots,0)` in Program 1, array `p` of five floats for coefficients of the polynomial is cast to a complex array

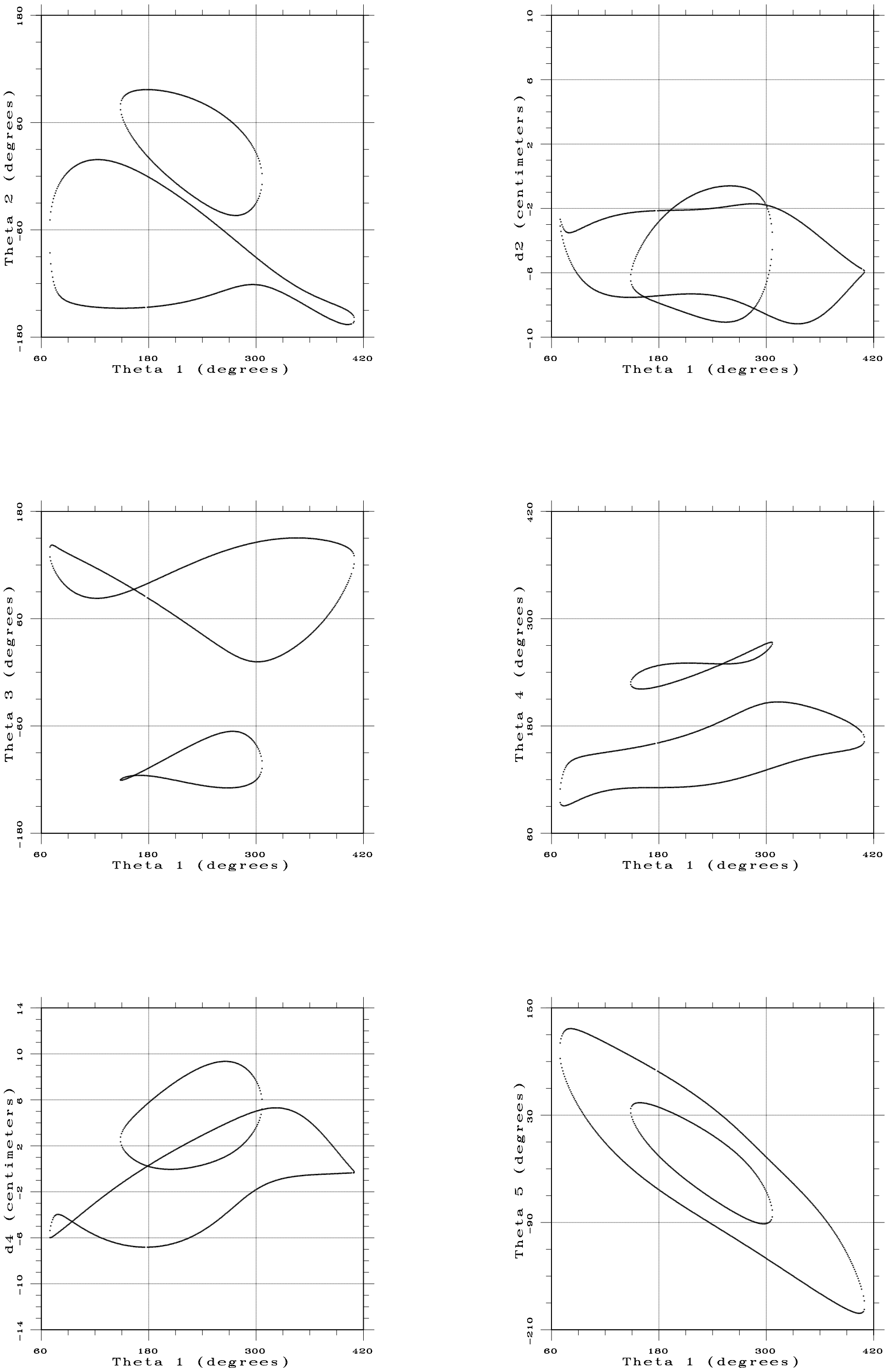

FIGURE 2: GRAPHICAL OUTPUT FOR JOINT VARAIBLES θ_2, d_2, θ_3, θ_4, d_4, AND θ_5 VERSUS INPUT θ_1 FROM PROGRAM 1.

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 Figure 2. There are six graphs that are produced by function `plotRCRCR()` using the data saved in file RCRCR.out. The function `plotRCRCR()` contains the x and y labels and graph dimensions. It invokes two function files called separate() and plotXYR(). 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 gives the y-coordinates. As the data is 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 `plotXYR()` 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 $\theta_2, \theta_3, \theta_4$, 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, **NaN**s are used as symbolic values for the other two non-existing solutions.

CONCLUSIONS

The $\mathrm{C^H}$ programming language is a general-purpose high-level programming language. $\mathrm{C^H}$ merges ANSI C and FORTRAN, and is designed to be especially suitable for applications in mechanical engineering. Many programming features are specially designed and implemented for design automation. Handling dual number as a basic built-in data type is one example. Dual formulas can be easily translated into $\mathrm{C^H}$ 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 in the programming paradigm of $\mathrm{C^H}$ are significantly simpler than those derived for implementation in programming languages without dual data type. A remarkably small $\mathrm{C^H}$ program can give a complete numerical and graphical displacement solution of an RCRCR mechanism. The $\mathrm{C^H}$ program is very simple and easy to implement. Although the presentation on the derivation of formulas and the developed $\mathrm{C^H}$ 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. The $\mathrm{C^H}$ programming language and its programming paradigm simplify the derivation and programming of formulas with dual numbers. With conciseness of formulation and programming simplicity, it is hoped that dual numbers will be used not only by kinematicians, but also by practicing engineers as a powerful mathematical tool to solve practical engineering analysis and design problems.

ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation as a Research Initiation Award under Grant DDM-9309207 and by University of California, Davis through an Undergraduate Instructional Improvement Grant, a Faculty Research Grant, and a Junior Faculty Research Fellowship to the second author.

REFERENCES

1. Bottema, O. and Roth, B., *Theoretical Kinematics*, North-Holland Publishing Comp., Amsterdam, 1979.

2. Cheng, H. H., *The C^H Programming Language*, revision 1.1, Department of Mechanical and Aeronautical Engineering, University of California, Davis, CA, November 30, 1993.

3. Cheng, H. H., Scientific Computing in the $\mathrm{C^H}$ Programming Language, *Scientific Programming*, Vol. 2, No. 3, Fall 1993, pp. 49-75.

4. Cheng, H. H., Handling of Complex Numbers in the $\mathrm{C^H}$ Programming Language, *Scientific Programming*, Vol. 2, No. 3, Fall 1993, pp. 77-106.

5. Cheng, H. H., Programming with Dual Numbers and its Applications in Mechanism Design, *Engineering with Computers*, 1994 (in press).

6. Cheng, H. H., Computations of Dual Numbers in the Extended Finite Dual Plane, *Proc. of ASME Design Automation Conference*, Albuquerque, NM, Sept. 19-22, 1993, vol. 2, pp. 73-80.

7. Cheng, H. H., Numerical Computations in the C^H Programming Language with Applications in Mechanisms and Robotics, *Proceedings of the Third National Conference on Applied Mechanisms and Robotics*, Cincinnati, OH, Nov. 8-10, vol. 1, pp AMR-19-17-1 to pp. AMR-93-19-15.

8. Cheng, H. H., Extending C and Fortran for Design Automation, *Proceedings of ASME Design Automation Conference*, Minneapolis, MN, Sept. 11-14, 1994.

9. Clifford, W. K., Preliminary Sketch of Bi-quaternions, *Proceedings of London Mathematical Society*, Vol. 4, 1873, pp. 361-395.

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

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

12. Dimentberg, F. M., The Determination of the Positions of Spatial Mechanisms, *Izdatel'stvo Akademii Nauk*, Moscow, USSR, 1950.

13. Duffy, J. and Habib-Olahi, H. Y., A Displacement Analysis of Spatial Five-Link 3R-2C Mechanisms; Part I: On the Closures of the RCRCR Mechanism, *Journal of Mechanisms*, Vol. 6, 1970, pp. 289-301.

14. Duffy, J., *Analysis of Mechanisms and Robot Manipulators,* Edward Arnold Ltd., 1980, pp. 235-242.

15. Gupta K. C., Spatial Kinematics. ME 409 Lecture Notes. Department of Mechanical Engineering, University of Illinois at Chicago, 1989.

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

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

18. Pennock, G. R. and Yang, A. T., Dynamic Analysis of Multi-Rigid-Body Open-Chain System, *ASME Trans., J. of Mechanisms, Transmissions, and Automation in Design*, Vol.105, March 1983, pp. 28-34.

19. Press, W. H., Flannery, B.P., Teukolshy, S. A., and Vetterling, W., *Numerical Recipes in C : The Art of Scientific Computing*, Cambridge University Press, NY, 1988.

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

21. Study, E., *Geometrie der Dynamen*, Leipzig, 1903.

22. Yang, A. T. and Freudenstein, F., Application of Dual-Number Quaternion 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.

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

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