Computer-aided displacement analysis of spatial mechanisms using the $CH$ 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 $CH$ programming language. $CH$ is designed to be a superset of $C$ with all the programming features of FORTRAN. Formulas with dual numbers can be translated into $CH$ 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 $CH$ 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 $CH$ programming paradigm through displacement analysis of the RCRCR five-link spatial mechanism. A $CH$ program that can obtain both numerical and graphical results for complete displacement analysis of the RCRCR mechanism will be presented.

Key words: $CH$ 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. Recently, Cheng introduced the $CH$ programming language. $CH$ is a superset of ANSI C with incorporation of all programming features of FORTRAN. Due to our research interests, $CH$ 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 $CH$ 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 $CH$ is simpler than any other computer programming language. The linguistic features related
to dual numbers in C^H have been presented by Cheng.\textsuperscript{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.\textsuperscript{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\textsuperscript{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\textsuperscript{6} simplified the input–output relation to a quadratic equation. For complete displacement analysis in FORTRAN, Yang had to separate a concise dual input-output displacement equation 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.\textsuperscript{6} Using line coordinates with dual numbers, Yuan\textsuperscript{7} presented complete solutions for displacement analysis of the same mechanism. Using spherical trigonometry with dual numbers, Duffy & Habib-Olahi\textsuperscript{8,9} also studied the RCRCR mechanism. They showed that there are eight solutions for the RCRCR mechanism if the displacement d_4 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\textsuperscript{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 \( \theta_3 \) and \( \theta_5 \) and cylindrical joint values \( \theta_2 \) and \( \theta_4 \) for a given input angle \( \theta_1 \). For the RCRCR mechanism, link parameters \( \alpha_i = \alpha_i + e \alpha_i \) for \( i = 1, \ldots, 5 \), \( d_1, d_3, \) and \( d_5 \) are fixed. The parameters \( \alpha_i, \alpha_i, \alpha_i, \) 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 \( X_i Y_i Z_i O_i \) to \( X_i' Y_i' Z_i' O_i' \), an intermediate coordinate system,\textsuperscript{1} can be represented by the following dual transformation matrix:

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

(1)

where dual angle \( \hat{\theta}_i = \theta_i + e \theta_i \). Similarly, the transformation of the coordinate system from \( X_i Y_i Z_i O_i \) to \( X_{i+1} Y_{i+1} Z_{i+1} O_{i+1} \) can be represented by the dual transformation matrix

\[
\hat{\Lambda}_i = \begin{bmatrix}
1 & 0 & 0 \\
0 & \cos \hat{\alpha}_i & -\sin \hat{\alpha}_i \\
0 & \sin \hat{\alpha}_i & \cos \hat{\alpha}_i
\end{bmatrix},
\]

(2)

where dual angle \( \hat{\alpha}_i = \alpha_i + e \alpha_i \). The dual transformation matrix between two coordinate systems \( X_i Y_i Z_i O_i \) and \( X_{i+1} Y_{i+1} Z_{i+1} O_{i+1} \) can be derived by the successive screw transformations as follows:

\[
\hat{D}_i = \hat{\Theta}_i \hat{\Lambda}_i = \begin{bmatrix}
\cos \hat{\theta}_i & -\sin \hat{\theta}_i \cos \hat{\alpha}_i & \sin \hat{\theta}_i \sin \hat{\alpha}_i \\
\sin \hat{\theta}_i & \cos \hat{\theta}_i \cos \hat{\alpha}_i & -\cos \hat{\theta}_i \sin \hat{\alpha}_i \\
0 & \sin \hat{\alpha}_i & \cos \hat{\alpha}_i
\end{bmatrix},
\]

(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 \( O_1 X_1 Y_1 Z_1 \) and \( O_3 X_3 Y_3 Z_3 \) are attached to input
and output links, respectively. Formulas for calculating $\theta_3$, $\theta_4$, and $\theta_5$, in terms of the input angle $\theta_1$, can be derived as follows. According to the dual transformation matrices $\Theta_1$ and $\Lambda_1$, we can write the loop-closure equation for the RCRCR mechanism shown in Fig. 1 as

$$\Theta_4\Theta_3\Theta_2\Theta_1\Theta_5\Theta_5\Theta_4 = \mathbf{I},$$

(4)

where $\mathbf{I}$ is a $3 \times 3$ identity matrix. Equation (4) can be cyclically permuted.$^{10}$ 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 $\theta_1$. Equation (4) can be reformulated as

$$\Lambda_1\hat{\Theta}_1\Lambda_3\hat{\Theta}_2\hat{\Theta}_3\Lambda_5\hat{\Theta}_4 = \hat{\Theta}_3^T \Lambda_1^T \Lambda_5^T \Lambda_3^T \hat{\Theta}_2^T \hat{\Theta}_1^T,$$

(5)

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

$$\begin{bmatrix}
\hat{d} & \cos \hat{\theta}_4 & \hat{\theta}_4 \\
\hat{\theta}_4 & \hat{d} & \cos \hat{\theta}_3 \\
\cos \hat{\theta}_3 & -\sin \hat{\theta}_3 & \sin \hat{\theta}_3
\end{bmatrix}
\begin{bmatrix}
\cos \hat{\theta}_4 & \hat{d} & \hat{\theta}_4 \\
\hat{\theta}_4 & \cos \hat{\theta}_3 & \hat{d} \\
\cos \hat{\theta}_3 & -\sin \hat{\theta}_3 & \sin \hat{\theta}_3
\end{bmatrix}
\begin{bmatrix}
\theta_1 & \theta_2 & \theta_3 & \theta_4 & \theta_5
\end{bmatrix}
= \begin{bmatrix}
\hat{d} & \cos \hat{\theta}_4 & \hat{\theta}_4 \\
\hat{\theta}_4 & \cos \hat{\theta}_3 & \hat{d} \\
\cos \hat{\theta}_3 & -\sin \hat{\theta}_3 & \sin \hat{\theta}_3
\end{bmatrix}
\begin{bmatrix}
\cos \theta_1 & \hat{d} & \theta_4 \\
\theta_4 & \cos \theta_3 & \hat{d} \\
\cos \theta_3 & -\sin \theta_3 & \sin \theta_3
\end{bmatrix}
\begin{bmatrix}
\theta_1 & \theta_2 & \theta_3 & \theta_4 & \theta_5
\end{bmatrix}
= \begin{bmatrix}
\hat{d} & \cos \hat{\theta}_4 & \hat{\theta}_4 \\
\hat{\theta}_4 & \cos \hat{\theta}_3 & \hat{d} \\
\cos \hat{\theta}_3 & -\sin \hat{\theta}_3 & \sin \hat{\theta}_3
\end{bmatrix}
\begin{bmatrix}
\cos \theta_1 & \hat{d} & \theta_4 \\
\theta_4 & \cos \theta_3 & \hat{d} \\
\cos \theta_3 & -\sin \theta_3 & \sin \theta_3
\end{bmatrix}
\begin{bmatrix}
\theta_1 & \theta_2 & \theta_3 & \theta_4 & \theta_5
\end{bmatrix}
$$

(6)

where

$$\hat{\theta}_4 = \cos \hat{\theta}_1$$

(7)

$$\hat{d} = -\cos \hat{\theta}_1 \sin \hat{\theta}_1$$

(8)

$$\hat{\theta}_3 = \sin \hat{\theta}_1 \sin \hat{\theta}_1$$

(9)

$$\hat{\theta}_5 = \cos \hat{\theta}_1 \sin \hat{\theta}_1$$

(10)

$$\hat{\theta}_4 = \cos \hat{\theta}_1 \cos \hat{\theta}_3 \cos \hat{\theta}_5 - \sin \hat{\theta}_5 \sin \hat{\theta}_1$$

(11)

$$\hat{\theta}_3 = \sin \hat{\theta}_1 \cos \hat{\theta}_3 \cos \hat{\theta}_5 - \cos \hat{\theta}_1 \sin \hat{\theta}_5$$

(12)

$$\hat{\theta}_5 = \sin \hat{\theta}_1 \sin \hat{\theta}_3$$

(13)

$$\hat{\theta}_4 = \cos \hat{\theta}_1 \cos \hat{\theta}_3 \sin \hat{\theta}_5 + \sin \hat{\theta}_1 \cos \hat{\theta}_5$$

(14)

$$\hat{\theta}_3 = \sin \hat{\theta}_1 \cos \hat{\theta}_3 \sin \hat{\theta}_5 + \cos \hat{\theta}_1 \cos \hat{\theta}_5$$

(15)

$$\hat{\theta}_5 = \sin \hat{\theta}_1 \cos \hat{\theta}_3 \cos \hat{\theta}_5 + \cos \hat{\theta}_1 \sin \hat{\theta}_5$$

(16)

$$\hat{\theta}_4 = \cos \hat{\theta}_1 \sin \hat{\theta}_5$$

(17)

$$\hat{\theta}_3 = \sin \hat{\theta}_1 \cos \hat{\theta}_3$$

(18)

$$\hat{\theta}_5 = \cos \hat{\theta}_1$$

(19)

$$\hat{\theta}_4 = \sin \hat{\theta}_1 \cos \hat{\theta}_3$$

(20)

$$\hat{\theta}_5 = \cos \hat{\theta}_1$$

(21)

To determine solutions to the unknown values $\hat{\theta}_2$, $\hat{\theta}_3$, $\hat{\theta}_4$, and $\hat{\theta}_5$, the following steps are taken. First, the dual equation from element (3,3) of dual matrix eqn (6) is used to determine $\hat{\theta}_3$ and $\hat{\theta}_5$. Next, $\hat{\theta}_2$ is determined by solving equations from elements (1,3) and (2,3) with the given input value $\theta_1$ 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 $\theta_1$, $\hat{\theta}_3$, and $\hat{\theta}_5$.

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

$$\begin{bmatrix}
\cos \hat{\theta}_2 \cos \hat{\theta}_3 - \sin \hat{\theta}_2 \sin \hat{\theta}_3 \\
-\sin \hat{\theta}_2 \cos \hat{\theta}_3 - \cos \hat{\theta}_2 \sin \hat{\theta}_3 \\
\cos \hat{\theta}_2 \sin \hat{\theta}_3 + \sin \hat{\theta}_2 \cos \hat{\theta}_3
\end{bmatrix}
= \begin{bmatrix}
\cos \theta_2 \cos \theta_3 - \sin \theta_2 \sin \theta_3 \\
-\sin \theta_2 \cos \theta_3 - \cos \theta_2 \sin \theta_3 \\
\cos \theta_2 \sin \theta_3 + \sin \theta_2 \cos \theta_3
\end{bmatrix}
$$

(22)

Equation (22) contains three dual angles $\hat{\theta}_1$, $\hat{\theta}_3$, and $\hat{\theta}_5$. Both $\theta_1$ and $\theta_5$ are given so that there are only 2 unknowns, $\hat{\theta}_1$ and $\hat{\theta}_3$, in eqn (22). Therefore, we should further simplify the equation to contain only $\hat{\theta}_1$ and $\hat{\theta}_3$, and express the rest as parameters that can be easily handled by the CH programming language. Equation (22) can be easily rewritten as

$$\hat{\theta}_1 = \cos \hat{\theta}_3$$

(23)
where

\[ A = \cos \hat{\alpha}_2 \cos \hat{\alpha}_3 \]

\[ - \cos \hat{\alpha}_4 (\cos \hat{\alpha}_5 \cos \hat{\alpha}_1 + \sin \hat{\alpha}_5 \sin \hat{\alpha}_1 \cos \hat{\theta}_1) \quad (24) \]

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

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

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

These parameters are very easy to compute in CH. For example, parameter \( \hat{C} \) can be calculated in CH as

\[ \hat{C} = \sin(\alpha_1) \cdot \cos(\alpha_4) \cdot \sin(\theta_1) \quad . \]

Details on CH programming for the RCRCR mechanism will be described in the next section. There are two unknowns, \( \theta_3 \) and \( \theta_5 \), in eqn (23). These two unknowns are real parts of dual angles \( \theta_3 \) and \( \theta_5 \). Therefore, it is inevitable at this point to single out \( \theta_3 \) and \( \theta_5 \) from dual numbers \( \theta_3 \) and \( \theta_5 \), respectively. But, other dual parameters in the equation can remain intact. Furthermore, we can either solve for \( \theta_3 \) first, then for \( \theta_5 \), or solve for \( \theta_5 \) first, then for \( \theta_3 \). We choose to solve \( \theta_3 \) first. Because the resultant equation involves both \( \sin \theta_3 \) and \( \cos \theta_3 \), we introduce an additional parameter \( \chi = \tan(\theta_3/2) \) to express both \( \sin \theta_3 \) and \( \cos \theta_3 \) in terms of \( \chi \). After substituting the identities

\[ \sin \hat{\theta}_3 = \sin \theta_3 + \epsilon \hat{d}_3 \cos \theta_3 \quad (28) \]

\[ \cos \hat{\theta}_3 = \cos \theta_3 - \epsilon \hat{d}_3 \sin \theta_3 \quad (29) \]

\[ \sin \theta_3 = \frac{2 \chi}{1 + \chi^2} \quad (30) \]

\[ \cos \theta_3 = \frac{1 - \chi^2}{1 + \chi^2} \quad (31) \]

where

\[ \chi = \tan \frac{\theta_3}{2} \quad (32) \]

and making additional substitutions,

\[ \hat{E} = \frac{\hat{A} + \epsilon \hat{d}_3 \hat{C} - \hat{D}}{\hat{B}} \quad (33) \]

\[ \hat{F} = \frac{\hat{C} + \epsilon \hat{d}_3 \hat{D}}{\hat{B}} \quad (34) \]

\[ \hat{G} = \frac{\hat{A} - \epsilon \hat{d}_3 + \hat{C} + \hat{D}}{\hat{B}} \quad (35) \]

eqn (23) becomes

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

The identity \( \sin^2 \theta + \cos^2 \theta = 1 \) can be used to eliminate the variable \( \theta_3 \). Equation (36) can be separated into its real and imaginary parts,

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

\[ \sin \theta_3 = \frac{- \hat{E} \chi^2 - 2 \hat{F} \chi + \hat{G}}{- \hat{d}_3 (\chi^2 + 1)} \quad . \quad (38) \]

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 \( \hat{d} \) and \( d \) 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

\[ p_4 = \hat{E}^2 + \frac{\hat{F}^2}{\hat{d}_3^2} - 1 \quad (40) \]

\[ p_3 = -4 \left( \frac{\hat{E} \hat{F} + \hat{F} \hat{G}}{\hat{d}_3^2} \right) \quad (41) \]

\[ p_2 = 2 \hat{E} \hat{G} + 4 \hat{F}^2 + \frac{2 \hat{E} \hat{G} + 4 \hat{F}^2}{\hat{d}_3^2} - 2 \quad (42) \]

\[ p_1 = -4 \left( \frac{\hat{F} \hat{G} - \hat{F} \hat{G}}{\hat{d}_3^2} \right) \quad (43) \]

\[ p_0 = \hat{G}^2 + \frac{\hat{G}^2}{\hat{d}_3^2} - 1. \quad (44) \]

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

\[ p_4 = \text{real}(F) \ast \text{real}(F) + \text{dual}(F) \ast \text{dual}(F)/(\text{d}_3 \ast \text{d}_3) - 1; \]

The solution for \( \chi \) in eqn (39) can be computed by using a numerical solution. And \( \theta_3 \) then can be determined from the equation \( \theta_3 = 2 \arctan(\chi) \). Because eqn (39) is a fourth order polynomial equation, there are up to four real solutions for \( \chi \) for a given input angle \( \theta_1 \). Once \( \theta_3 \) is obtained, \( \theta_5 \) can be calculated by using eqn (36).

\[ \theta_3 = \pm \arccos \left( \frac{\hat{A} + \epsilon \hat{d}_3 \cos \hat{\theta}_3 - \hat{C} \sin \hat{\theta}_3}{\hat{B}} \right). \quad (45) \]

If \( \hat{d}_3 \) takes both positive and negative values, there are two solutions for \( \theta_3 \) from eqn (45) corresponding to a given \( \theta_5 \). There are up to four solutions for \( \theta_5 \) for a given \( \theta_1 \). 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 \( \hat{d}_3 \) is fixed. Therefore, only one of two solutions from eqn (45) is valid. This valid solution can be
obtained unambiguously by the formula \( \theta = 2 \arctan \left( \frac{\sin \theta_3}{\cos \theta_3 + 1} \right) \).

\[
\theta_3 = 2 \arctan \left( \frac{\sin \theta_3}{\cos \theta_3 + 1} \right)
= 2 \arctan \left( \frac{\dot{X}_X^2 - 2 \dot{X}_X + \dot{Y}}{-d_1(E + 1)X^2 - 2 \dot{X}_X + \dot{Y} + 1} \right)
\]

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

At this point, \( \theta_1, \theta_2, \) and \( \theta_3 \) all are known. The dual equation derived from elements (1,3) and (2,3) of dual matrix eqn (6) are in terms of \( \theta_1, \theta_2, \theta_3, \) and \( \theta_4 \). This leaves angle \( \theta_2 \) 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

\[
\begin{align*}
\dot{A}_1 &= \dot{D}_1 \sin \theta_2 + \dot{C}_1 \cos \theta_2 \\
\dot{B}_1 &= \dot{C}_1 \sin \theta_2 - \dot{D}_1 \cos \theta_2
\end{align*}
\]

respectively, where

\[
\begin{align*}
\dot{A}_1 &= (\cos \alpha_4 \sin \alpha_5 + \cos \alpha_5 \cos \theta_5) \sin \theta_1 \\
&\quad + \sin \alpha_4 \cos \theta_1 \sin \theta_5 \\
\dot{B}_1 &= (\cos \alpha_4 \sin \alpha_5 + \cos \alpha_5 \cos \theta_5) \cos \theta_1 \\
&\quad - [\sin \alpha_5 \sin \alpha_1 - \cos \alpha_5 \cos \alpha_1 \cos \theta_1]) \cos \theta_5 \\
&\quad + \cos \alpha_4 \sin \theta_1 \sin \theta_5 \sin \alpha_4 \\
\dot{C}_1 &= \sin \alpha_5 \sin \theta_3 \\
\dot{D}_1 &= \sin \alpha_5 \cos \alpha_3 + \cos \alpha_2 \sin \alpha_3 \cos \theta_3.
\end{align*}
\]

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

\[
\begin{align*}
\sin \dot{\theta}_2 &= \frac{\dot{A}_1 \dot{D}_1 + \dot{B}_1 \dot{C}_1}{C_1^2 + D_1^2} \\
\cos \dot{\theta}_2 &= \frac{A_1 \dot{C}_1 - \dot{B}_1 \dot{D}_1}{C_1^2 + D_1^2}
\end{align*}
\]

Then, the value of \( \dot{\theta}_2 \) can be calculated by

\[
\dot{\theta}_2 = 2 \arctan \left( \frac{\sin \dot{\theta}_2}{\cos \dot{\theta}_2 + 1} \right)
= 2 \arctan \left( \frac{\dot{A}_1 \dot{D}_1 + \dot{B}_1 \dot{C}_1}{C_1^2 + D_1^2 + A_1 \dot{C}_1 - B_1 \dot{D}_1} \right)
\]

Joint values \( \dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3, \) and \( \dot{\theta}_4 \) 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 \( \dot{\theta}_1, \dot{\theta}_3, \dot{\theta}_4, \) and \( \dot{\theta}_5 \). This leaves dual angle \( \dot{\theta}_4 \) as the only unknown left in the displacement analysis. Dual angle \( \dot{\theta}_4 \) is determined by using two equations in the same manner as in solving for \( \dot{\theta}_2 \). Elements (3,1) and (3,2) in eqn (6) give the following two equations:

\[
\begin{align*}
\dot{A}_2 &= \dot{D}_2 \sin \theta_4 + \dot{C}_2 \cos \theta_4 \\
\dot{B}_2 &= \dot{C}_2 \sin \theta_4 - \dot{D}_2 \cos \theta_4
\end{align*}
\]

respectively, where

\[
\begin{align*}
\dot{A}_2 &= (\sin \alpha_5 \cos \alpha_1 + \cos \alpha_5 \sin \alpha_1 \cos \theta_1) \sin \theta_5 \\
&\quad + \sin \alpha_1 \sin \theta_1 \sin \theta_5 \\
\dot{B}_2 &= (\cos \alpha_5 \cos \alpha_1 - \sin \alpha_5 \sin \alpha_1 \cos \theta_1) \sin \theta_5 \\
&\quad + [\sin \alpha_5 \cos \alpha_1 + \cos \alpha_5 \sin \alpha_1 \cos \theta_1] \cos \theta_5 \\
&\quad - \sin \alpha_1 \sin \theta_1 \sin \theta_5 \sin \alpha_4 \\
\dot{C}_2 &= \sin \alpha_5 \sin \theta_3 \\
\dot{D}_2 &= \cos \alpha_2 \sin \alpha_3 + \cos \alpha_2 \sin \alpha_3 \cos \theta_3.
\end{align*}
\]

Similar to \( \dot{\theta}_2 \) and value of \( \dot{\theta}_4 \) can be calculated by

\[
\dot{\theta}_4 = 2 \arctan \left( \frac{\sin \dot{\theta}_4}{\cos \dot{\theta}_4 + 1} \right)
= 2 \arctan \left( \frac{\dot{A}_2 \dot{D}_2 + \dot{B}_2 \dot{C}_2}{C_2^2 + D_2^2 + A_2 \dot{C}_2 - B_2 \dot{D}_2} \right)
\]

In the above formulations for calculating \( \theta_5, \theta_3, \dot{\theta}_2, \) and \( \dot{\theta}_4 \), we have kept the succinct dual format whenever possible. For example, dual twist angles \( \alpha_i \) (i = 1, . . . , 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. Such simplification of formulations is possible only because we can easily translate these formulas into a 

### 3 A C\textsuperscript{II} 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 C\textsuperscript{II} program by the following analysis problem.

Problem 1: Given the following link parameters for an RCRCR mechanism shown in Fig. 1: \( \alpha_1 = 30^\circ + \epsilon_1 10 \) cm, \( \alpha_2 = 35^\circ + \epsilon_4 40 \) cm, \( \alpha_3 = 45^\circ + \epsilon_3 30 \) cm, \( \alpha_4 = 60^\circ + \epsilon_5 25 \) cm, \( \alpha_5 = 10^\circ + \epsilon_5 32 \) cm, \( d_1 = 0 \), \( d_2 = 25 \) cm, and \( d_3 = 30 \) cm, find all possible
H. H. Cheng, S. Thompson

/* 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, "%/I theta 1 theta 2 d2(cm) theta 3 theta 4 d4(cm) theta 5 \n");

/************ INITIAL VARIABLES ***************I***********/
d1=0.0; d5=2.50; d5=3.00; /* fixed displacements of the RCRCR mechanism */
alphas1=dual(30*PI/180,1); /* fixed dual angles of the RCRCR mechanism */
alphas2=dual(35*PI/180,4.0); alphas3=dual(45*PI/180,3.0);
alphas4=dual(60*PI/180,2.5); alphas5=dual(10*PI/180,3.2);
for (thetal_i=60; thetal_i (~420; thetal_i++) /* 60 <= theta1 <= 420 */
{
    fprintf(stream, "% \n"); /* add space between sets of data */
    thetal=dual(thetal_i*PI/180,dl);
    A=cos(alpha2)*cos(alpha3)-(cos(alpha4)*(cos(alpha5)*cos(alpha1)-
        sin(alpha5)*sin(alpha1)*cos(thetal)));  /* (24) */
    B=sin(alpha2)*sin(alpha3);  /* (25) */
    C=sin(alpha1)*sin(alpha4)*sin(thetal);  /* (26) */
    D=sin(alpha4)*(sin(alpha5)*cos(alpha1)+
        cos(alpha5)*sin(alpha1)*cos(thetal));  /* (27) */
    E=(A+C*dual(O,d5)-D)/B;  /* (28) */
    F=(C+dual(O,d5)*D)/B;  /* (29) */
    G=(A-dual(O,d5)*C+D)/B;  /* (30) */
p[4]=pow(real(E),2)+pow(dual(E),2)/(d3*d3)-1; /* (40) */
p[3]=-4*(real(E)*real(F)+dnal(E)*dual~F)/(d3*d3);  /* (41) */
p[2]=2*real(E)*real(G)+4*pow(real(F),2)+(2*dnal(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;
rootsp(p,4,roots,1);  /* Solve polynomial equation */
for(i=0; i<3; i++) /* 4 branches of the RCRCR mechanism */
    x = roots[i];
    theta3=dual(2*atan(dual(E*x*x-l*F*x+G)/
        (-d3*real((E+l)*x*x-2*F*x+G+i))),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(alpha5);  /* (49) */
    B1=(cos(alpha5)*sin(alpha1)+sin(alpha5)*cos(alpha1)*cos(theta1)) *
        (sin(alpha4)-(sin(alpha5)*sin(alpha1)-cos(alpha5)*cos(alpha1)) * 
        cos(theta5)) * cos(alpha1)*sin(alpha1)*sin(theta1) *
        sin(theta6)) * sin(alpha4);  /* (50) */
    C1=sin(alpha3)*sin(theta3);  /* (51) */
    D1=(cos(alpha2)*cos(alpha3)*cos(alpha2)*sin(alpha5)*cos(theta3);  /* (52) */
    theta2=2*atan((Ai+D1*B1*C1)/(C1+D1+A1*B1)),D1)) ;  /* (53) */
    A2=(sin(alpha5)*cos(alpha1)+sin(alpha5)*cos(alpha1)*cos(theta1)) *
        sin(theta5)+sin(alpha1)*sin(theta1)*cos(theta5);  /* (54) */
    B2=(cos(alpha5)*cos(alpha1)+sin(alpha5)*cos(alpha1)*cos(theta1)) *
        sin(alpha4)+(sin(alpha5)*cos(alpha1)+cos(alpha5)*sin(alpha1)) * 
        cos(theta5)-sin(alpha1)*sin(theta1)*cos(theta5);  /* (55) */
    /* (56) */
Computer-aided displacement analysis

\[
\sin(\theta_S) \times \cos(\alpha_4); \quad \text{(* (57) *)}
\]
\[
C_I = \sin(\alpha_2) \times \sin(\theta_S); \quad \text{(* (58) *)}
\]
\[
D_2 = \cos(\alpha_2) \times \sin(\alpha_3) + \sin(\alpha_2) \times \cos(\alpha_3) \times \cos(\theta_3); \quad \text{(* (59) *)}
\]
\[
\theta_4 = \frac{\alpha_2 + \alpha_3}{(A_2 + D_2 + D_3 + C_2 + C_3 + D_2)}; \quad \text{(* (60) *)}
\]

/**************************** PRINT RESULTS ***************************/
if (real(theta5)*MO/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 ",
theta1_i,real(theta2)*180/PI,dual(theta2),real(theta3)*180/PI,
dual(theta4)*180/PI,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 CH program for displacement analysis of an RCRCR mechanism.

solutions for joint variables \( \theta_2, d_2, \theta_3, d_4, \theta_4, \) and \( \theta_5 \) when \( \theta_1 \) varies within its valid range. Plot joint variables vs input angle \( \theta_1 \) when joint \( \theta_1 \) 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 \( \theta_3 \) corresponding to each input angular position \( \theta_1 \). Subsequently, there are four solutions for \( \theta_3, \theta_2, \) and \( \theta_4 \) for a given \( \theta_1 \). 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 formulae 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 CH programming language numerically through the symbolic notation NaN. Any invalid value in CH is represented by the metanumber NaN. Therefore, we can compute joint variables when the input angle \( \theta_1 \) makes a complete 360° rotation. If the input angle \( \theta_1 \) is outside the valid input range, the values computed for all other joint variables will become NaNs.

Problem 1 can be solved by a CH 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 CH program in Program 1 is remarkably small. Most importantly, it is straightforward and easy to understand. We will highlight some critical points

<table>
<thead>
<tr>
<th>%</th>
<th>theta 1</th>
<th>theta 2</th>
<th>d2 (cm)</th>
<th>theta 3</th>
<th>theta 4</th>
<th>d4 (cm)</th>
<th>theta 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
</tr>
<tr>
<td>60</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
</tr>
<tr>
<td>60</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
</tr>
<tr>
<td>60</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
</tr>
<tr>
<td>120</td>
<td>18.489</td>
<td>-7.330</td>
<td>82.794</td>
<td>107.196</td>
<td>-2.998</td>
<td>-0.422</td>
<td>NaN</td>
</tr>
<tr>
<td>120</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
</tr>
<tr>
<td>120</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
</tr>
<tr>
<td>120</td>
<td>-146.343</td>
<td>-2.626</td>
<td>117.516</td>
<td>150.318</td>
<td>-5.743</td>
<td>111.508</td>
<td>NaN</td>
</tr>
<tr>
<td>180</td>
<td>-0.767</td>
<td>-7.427</td>
<td>99.701</td>
<td>111.059</td>
<td>0.278</td>
<td>-53.129</td>
<td>NaN</td>
</tr>
<tr>
<td>180</td>
<td>21.153</td>
<td>-2.840</td>
<td>-107.312</td>
<td>247.612</td>
<td>0.220</td>
<td>39.036</td>
<td>NaN</td>
</tr>
<tr>
<td>180</td>
<td>-146.419</td>
<td>-2.137</td>
<td>82.780</td>
<td>161.180</td>
<td>6.813</td>
<td>78.626</td>
<td>NaN</td>
</tr>
<tr>
<td>180</td>
<td>-131.176</td>
<td>-8.737</td>
<td>149.812</td>
<td>147.538</td>
<td>4.116</td>
<td>-169.184</td>
<td>NaN</td>
</tr>
<tr>
<td>360</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
</tr>
<tr>
<td>360</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
</tr>
<tr>
<td>360</td>
<td>-146.977</td>
<td>-3.758</td>
<td>96.887</td>
<td>-7.874</td>
<td>111.508</td>
<td>5.733</td>
<td>NaN</td>
</tr>
<tr>
<td>360</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
</tr>
<tr>
<td>360</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
<td>NaN</td>
</tr>
</tbody>
</table>
for the implementation of Program 1 in the following discussion.

In Program 1, dependent variable $\theta_1$ and joint displacements $d_1, d_2, d_3, d_4,$ and $d_5$ are declared as floats $\text{theta}_1, d_1, d_2, d_3, d_4,$ and $d_5$; twist dual angles $\alpha_1, \alpha_2, \alpha_3, \alpha_4,$ and $\alpha_5$ are declared as dual variables $\alpha_1, \alpha_2, \alpha_3, \alpha_4,$ and $\alpha_5$; dual angles $\theta_1, \theta_2, \theta_3, \theta_4,$ and $\theta_5$ are declared as $\theta_1, \theta_2, \theta_3, \theta_4,$ and $\theta_5$.
theta2, theta3, theta4, and theta5; dual parameters \( A, B, C, D, E, F \), and \( \bar{G} \) are declared as dual variables \( \theta_1, \theta_2, \theta_3, \theta_4, \theta_5 \) 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 \( k_i, B, C, D, E, F \), and \( \phi_1, \phi_2, \phi_3, \phi_4, \phi_5 \) are declared as dual variables \( A_1, B_1, C_1, D_1, A_2, B_2, C_2, \) and \( D_2 \) 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 CH programming statements verbatim. The analytical sequences of the algorithm is very clear in Program 1. The input angle \( \theta_1 \), defined as \( \theta_1 = \theta_1 - \theta_0 \), 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 \( \theta_1 \) 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 \( \theta_1 \) and columns 2–7 are the corresponding output angles or displacements for \( \theta_2, \theta_3, \theta_4, \phi_4, \) and \( \phi_5 \). For each input angle \( \theta_1 \), 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 \( \chi \) in eqn (39) are obtained by the function \( \text{zroots}(\cdot) \) using a function file \( \text{zroots.ff} \). The function \( \text{zroots}(\cdot) \) in CH is treated as if it was a system built-in function like \( \sin(\cdot) \) or \( \cos(\cdot) \). 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 \( \text{zroots}(\cdot) \) is based on Laguerre’s method. The function \( \text{zroots}(\cdot) \) will guarantee the delivery of \( n \) complex roots for an \( n \)th order polynomial equation. More details about the algorithm for this function can be found in Press. In CH, 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 \( \text{NaN} \) as described in Ref. 3. For the function call of \( \text{zroots}(P, 4, \text{roots}, 1) \) 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 \( \text{zroots}(\cdot) \) 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 \( \text{NaN} \). This indicates that no solution exists for this branch corresponding to the given input angle \( \theta_1 \). For example, there are only two sets of solutions for output joint variables when \( \theta_1 \) is at 120° or 360° as shown in Table 1. There is no solution when \( \theta_1 \) is 60° or 420°. The graphical output from Program 1 is shown in Fig. 2. There are six graphs that are produced by function \( \text{plotRCRCR}(\cdot) \) using the data saved in file RCRCR.out. The function \( \text{plotRCRCR}(\cdot) \) contains labels and ranges of \( x \) and \( y \) coordinates. It invokes two function files called separate(\cdot) and plotxyrf(\cdot). The function separate(\cdot) 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 \( \text{NaN} \) will also be ignored during separation. Based upon data in these six files, the function plotxyrf(\cdot) 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 \( \theta_1 \) 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 \( \theta_5 \) are computed within the domain from \( -180° \) to \( 180° \). To show the continuous graphs, the input range for angle \( \theta_1 \) and output ranges for \( \theta_2, \theta_3, \phi_4, \theta_5 \) are adjusted. \( \theta_1 \) varies from 60° to 420° instead of from 0° to 360°. No adjustment is made to \( \theta_2 \) and \( \theta_3 \). They are continuous in the range of \( -180° \) to \( 180° \). \( \theta_4 \) is adjusted to vary between 60° to 420° and \( \theta_5 \) 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 \( \theta_1 \) 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 \( \theta_2 \) when \( \theta_1 = 180° \). They are \( 1.767°, 96.887°, 21.153°, \) 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, \( \theta_2 \) can be \( 18.489° \) or \( -146.343° \) when \( \theta_1 = 120° \), as shown in Table 1. In this case, \( \text{NaN} \) 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