Proceedings of DETC'02:
27th Biennial Mechanisms and Robotics Conference
Montreal, Canada, September 29-October 2, 2002

DETC2002/MECH-34272

FIVE PRECISION POINTS SYNTHESIS OF SPATIAL RRR MANIPULATORS USING INTERVAL ANALYSIS

Eric Lee, Constantinos Mavroidis*
Dept. of Mechanical and Aerospace Engineering
Rutgers University
98 Brett Rd., Piscataway, NJ 08854, USA
TEL: 732 – 445 – 0732, FAX: 732 – 445 – 3124
EMAIL: chingkui@eden.rutgers.edu,
mavro@jove.rutgers.edu

Jean Pierre Merlet*
INRIA Sophia Antipolis
2004 Route des Lucioles, BP 93
06902 Sophia Antipolis Cedex, FRANCE
TEL: 04 92 38 7761, FAX: 04 92 38 7643
EMAIL: Jean-Pierre.Merlet@sophia.inria.fr

ABSTRACT
In this paper, the geometric design problem of serial-link robot manipulators with three revolute (R) joints is solved for the first time using an interval analysis method. In this problem, five spatial positions and orientations are defined and the dimensions of the geometric parameters of the 3-R manipulator are computed so that the manipulator will be able to place its end-effector at these pre-specified locations. Denavit and Hartenberg parameters and 4x4 homogeneous matrices are used to formulate the problem and obtain the design equations and an interval method is used to search for design solutions within a predetermined domain. At the time of writing this paper, six design solutions within the search domain and an additional twenty solutions outside the domain have been found.

KEYWORDS
Geometric Design, Robot Manipulators, Interval Analysis

INTRODUCTION
The calculation of the geometric parameters of a multi-articulated mechanical or robotic system so that it guides a rigid body in a number of specified spatial locations or *precision points* is known as the *Rigid Body Guidance Problem*. In this paper, it will also be called the *Geometric Design Problem*. The precision points are described by six parameters: three for position and three for orientation. This problem has been studied extensively for planar mechanisms and robotic systems and has recently drawn much attention to researchers for spatial multi-articulated systems. Solution techniques for the geometric design problem may be classified into two categories: *exact synthesis* and *approximate synthesis*.

Exact synthesis methods result in mechanisms and manipulators, which guide a rigid body exactly through the specified precision points. Solutions in the exact synthesis exist only if the number of independent design equations obtained by the precision points is less than or equal to the number of design parameters. The number of precision points that may be prescribed for a given mechanism or manipulator is limited by the system type [1]. This number depends on the number of design parameters and the type of joints and can be calculated using Tsai and Roth's formula [2], [3].

In *approximate synthesis*, using an optimization algorithm, a mechanism is found that, although not guiding a rigid body exactly through the desired poses, it optimizes an objective function defined using information from all the desired poses. Approximate synthesis is mainly used in *over-determined* geometric design problems where more precision points are defined than required for exact synthesis and therefore no exact solution exists. A complete listing of the extensive amount of research that has been performed in the geometric design of spatial mechanisms and robotic systems, both exact and approximate, can be found in [4].

The equations for the geometric design problem of mechanisms and manipulators are mathematically represented by a set of non-linear, highly coupled multivariate polynomial equations. The solutions of these equations can be obtained by either numerical methods or algebraic methods [5]. Algebraic methods solve the polynomial system by eliminating all but one variable that gives a polynomial equation in one variable. All the solutions are then obtained by solving for the roots of the final polynomial. While algebraic solutions are usually very difficult to obtain, numerical methods serve as an alternative to solve these nonlinear equations. Examples of these methods are polynomial continuation and interval methods.

* Corresponding Authors

Using algebraic methods, the exact synthesis of planar mechanisms for rigid body guidance has been studied extensively by many researchers and is described in most textbooks on mechanism synthesis [6], [7]. The exact synthesis of a few spatial mechanisms and manipulators has been solved using algebraic methods. The spatial geometric design problems that captured the most attention were the spatial revolute-revolute (R-R) [8]-[11] and the cylindrical-cylindrical (C-C) manipulators [12]-[14]. Other than these two dyads, the geometric design problem has been solved algebraically for the following spatial manipulators/mechanisms. Innocenti [15] solved the geometric design problem for the sphere-sphere binary link. Neilsen and Roth [14] solved the slider-slider sphere dyad, cylinder-cylinder binary link, revolute-slider-sphere dyad and cylinder-sphere binary link design problem. McCarthy [16] also solved the exact synthesis problem for several types of dyads. Even though algebraic methods had been demonstrated to be very effective in solving several geometric design problems for spatial mechanical systems there exist many types of robotic and mechanical systems that are used very often in practical applications for which the exact synthesis of the geometric design problem has not been solved before. The main reason for this is the high complexity of the non-linear polynomial design equations that are obtained.

Polynomial continuation methods have been used extensively in the geometric analysis and design of mechanisms and robotic systems [17]. Roth and Freudenstein [18] were the first to use continuation methods to solve polynomial systems obtained in the kinematic synthesis of mechanisms. Morgan and Wampler [19] and Wampler, Morgan and Sommese [20] solved the path following design problem of 4R closed loop planar mechanisms using continuation method. Dhingra, Cheng and Kohli [24] solved several design problems for six link, slider crank and four-link planar mechanisms using polynomial continuation methods. Lee and Mavroidis [22] used continuation methods to solve the spatial 3R geometric design problem when 3 precision points are specified.

Interval analysis is a numerical method based on interval arithmetic [23]. It was developed for error control [24] and had been used in optimization. It can be used as a method for solving system of non-linear multivariate polynomial equations [25], but it has never been used in kinematic design of spatial mechanisms.

In this paper, the geometric design problem of 3R spatial manipulators when five precision points are defined is solved for the first time using an interval analysis method. Prior work related to the synthesis of spatial 3R chains is very limited. Tsai [2] and Roth [9] used screw theory to obtain the design equations for this problem but did not solve them. Lee and Mavroidis [22] used continuation methods to solve the three precision point synthesis problem for the spatial 3R when 6 design parameters are selected as free choices. They considered two different schemes for free choice selection and showed that in both cases the 3R synthesis problem using 3 precision points can have at the most 8 distinct solutions. In this paper, we solve the more difficult synthesis problem for the 3R when five precision points are selected. In this case no free choices are selected and hence, the number of design unknowns to calculate is much larger than in the cases with 3 or 4 precision points. Five spatial positions and orientations are defined and the dimensions of the geometric parameters of the 3-R manipulator are computed so that the manipulator will be able to place its end-effector at these five pre-specified locations. Denavit and Hartenberg (DH) parameters and 4x4 homogeneous matrices are used to formulate the problem and obtain the design equations. Interval method is used to search for design solutions of the design equations within a predetermined domain. At the time of writing this paper, six design solutions within the search domain and an additional twenty solutions outside the domain are found.

INTERVAL METHOD

The basic principle of interval method relies on interval arithmetic to determine bounds for the minimum and maximum value of a given function when the unknowns lie in some given ranges. One of the possible ways to obtain these bounds is to replace the mathematical operators of the function by their equivalent in term of interval arithmetic. For example if we consider the function $f(x)=x^2-x$ when x lie in the range [2,3], then we may write:

$$f([2,3]) = [2,3]^2 - [2,3] = [4,9] - [2,3] = [1,7] \tag{1}$$

The value we get here are lower and upper bound for the real value of the minimum and maximum of the function. In other words we guarantee that whatever is the value of x in the range [2,3], then:

$$1 \le f(x) \le 7 \tag{2}$$

An interesting feature of interval arithmetic is that it can be implemented to take into account round-off errors i.e. the bounds we get are guaranteed to include the exact value of the minimum and maximum of the function. Furthermore interval arithmetic can be used for almost any mathematical operator such as the trigonometric functions.

On the other hand a bad point is that the bounds we get may be over-estimated: in our example the real bounds are [2,6]. But the error decreases with the width of the input ranges. A basic solving algorithm relying on interval arithmetic will use the fact that if the bounds returned by the interval evaluation of an equation does not include 0, then the equation has no solution in the range of the unknowns (e.g. in our example we can insure that there is no root of the equation $x^2-x=0$ for x in the range [2,3]).

In a solving algorithm based on interval analysis we will assume that we are looking for all the solutions within given ranges for the unknowns (a set of range for the unknowns will be called a "box") and the algorithm will use a list of boxes, initialized with the box in which we are looking for solutions. The algorithm will proceed along the following steps:

1. Compute the interval evaluation of the equations for the current box.
2. If one of the interval evaluation does not include 0, then there is no solution in this box and we consider the next box in the list.
3. If all the interval evaluation of the equations include 0:
   - If the width of at least one range in the box is greater than a given threshold, then bisect one of the range of the box: 2 new boxes will be created and will be put at the end of the list.
   - If the width of all the ranges in the box is lower than the threshold, then store the box as a solution.

- Restart at 1 with the next box in the list.

4. Stop if all the boxes in the list have been considered.

Such algorithm is general in the sense that it can deal with equations involving any mathematical operators (we are not restricted for example to polynomial equations).

It must be noted that the previous algorithm can be implemented in a distributed way. Indeed the treatment of a box does not depend on the other boxes in the list. Hence a parallel implementation may be used, with a master sending the current box to another computer that will process it.

But there are many ways to improve this basic algorithm. Two main types of operators may be used:

(1) Filtering operators: these operators take as input a box and will return either the same box or a smaller box i.e. a box in which at least one of the range has a lower width than the one of the input box. In the latter case the eliminated parts of the input box do not include a solution.

(2) Existence and uniqueness operator: these operators take as input a box and may return a box with the following properties:

(i) There is a solution in this box.
(ii) This solution is unique.
(iii) The solution can be computed safely with an iterative algorithm with as initial guess any point within the box.

There are numerous operators that can be used. As an example of filtering operator let us mention the 2B method [26] that we will illustrate on the example. Let us define 2 new variables y_1 and y_2 with $y_1=x^2$ and $y_2=x$. Clearly if $f(x)=x^2-x$ has a solution for x within some given range, then we will have $y_1=y_2$. As x lie in a range, then y_1, y_2 also lie in a range: if x is in [2,3], then y_1, y_2 are in [4,9], [2,3]. A solution of the equation may be found only at the intersection of y_1 and y_2, which is empty in our case, meaning that there is no solution for the equation. If we have assumed that x was in [-4,1], then y_1, y_2 are in [0,16], [-4,1] and a possible solution lie in the range [0,1]. As y_2 is x we may reduce the interval for x from [-4,1] to [0,1].

Existence and uniqueness operator are very useful as they guarantee the solution and enable one to avoid a large number of bisection. An example of such operator is the Kantorovitch operator [27]. This operator needs to be able to compute the Jacobian and Hessian matrices of the system of equations and, provided that some conditions on the Jacobian and Hessian are fulfilled, allows to state that a unique solution exists within some given box and that this solution can be found with the Newton scheme.

PROBLEM FORMULATION

In this work, the relative position of links and joints in mechanisms and manipulators is described using the variant of DH notation that was introduced by Pieper and Roth [29]. In this formulation, the parameters a_i, α_i, d_i and θ_i are defined so that: a_i is the length of link i, α_i is the twist angle between the axes of joints i and i+1, d_i is the offset along joint i and θ_i is the rotation angle about joint axis i as shown in Figure 1. When joint i is revolute, then a_i, α_i and d_i are constants and are called structural parameters, while the value for θ_i depends on the configurations and is called the joint variable.

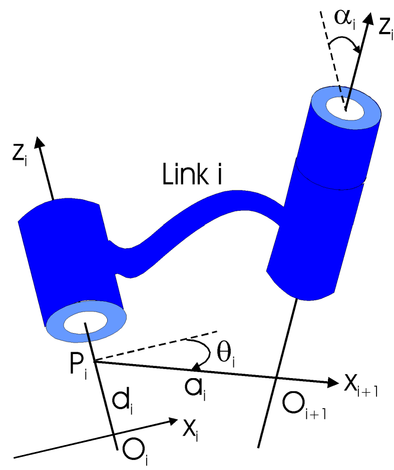

Figure 1: Denavit and Hartenberg Parameters

Reference frame R_i is attached at link i and its origin O_i is the intersection point of the common perpendicular between axes i and i-1 with joint axis i. Unit vector $\mathbf{z_i}$ of frame R_i is along joint axis i unit vector $\mathbf{x_i}$ is along the common perpendicular of joint axes i and i-1. Positive directions for $\mathbf{x_i}$ and $\mathbf{z_i}$ are arbitrarily selected. (Note: letters in bold indicate vectors and matrices.) The homogeneous transformation matrix $\mathbf{A_i}$ that describes reference frame R_{i+1} into R_i, and its inverse matrix $\mathbf{A_i}^{-1}$ are found to be equal to:

$$\mathbf{A_i}=\begin{pmatrix} c_i & -s_i c_{\alpha_i} & s_i s_{\alpha_i} & a_i c_i \\ s_i & c_i c_{\alpha_i} & -c_i s_{\alpha_i} & a_i s_i \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \mathbf{A_i^{-1}}=\begin{pmatrix} c_i & s_i & 0 & -a_i \\ -s_i c_{\alpha_i} & c_i c_{\alpha_i} & s_{\alpha_i} & -d_i s_{\alpha_i} \\ s_i s_{\alpha_i} & -c_i s_{\alpha_i} & c_{\alpha_i} & -d_i c_{\alpha_i} \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3)$$

where: $c_i=\cos(\theta_i)$, $s_i=\sin(\theta_i)$, $c_{\alpha i}=\cos(\alpha_i)$ and $s_{\alpha i}=\sin(\alpha_i)$.

Consider the three-link open loop spatial chain with revolute (R) joints shown in Figure 2. Two frames are selected arbitrarily: a fixed reference frame R_0 and a moving end-effector frame R_e. Frame R_e will be defined in three distinct spatial locations. In addition to the three links of the manipulator, a stationary virtual link 0 is also assumed between axis z_0 of frame R_0 and the first revolute joint axis. Frames are defined at each link using the DH procedure described above. Frame R_1 that is stationary is defined attached at link 0 having its z_1 axis along the first revolute joint and its x_1 axis along the common perpendicular of z_0 and z_1. Frame R_{i+1} is attached at the tip of link i (where i=1, 2, 3). The axis z_4 is coincident with the axis z_e of the end-effector frame. The axis x_4 is defined along the common perpendicular of z_3 and z_e and the origin O_4 of R_4 is the point of intersection of z_e with its common perpendicular with z_3. So frames R_4 and R_e have the same z-axis.

The homogeneous transformation matrices $\mathbf{A_i}$, with i=0, 1, 2, 3 describe frame R_{i+1} relative to R_i. The homogeneous transformation matrix $\mathbf{A_c}$ relates R_e to R_4. The relationship between these frames is a screw displacement: a rotation ϕ around the z_4 axis and a translation d along the z_4 axis. Homogeneous transformation matrix $\mathbf{A_h}$ relates directly the end-effector reference frame R_e to the frame R_0. Matrices $\mathbf{A_c}$ and $\mathbf{A_h}$ are written as:

$$\mathbf{A_c}=\begin{pmatrix} c_\phi & -s_\phi & 0 & 0 \\ s_\phi & c_\phi & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \mathbf{A_h}=\begin{pmatrix} l_1 & m_1 & n_1 & x_d \\ l_2 & m_2 & n_2 & y_d \\ l_3 & m_3 & n_3 & z_d \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4}$$

where $\mathbf{l}=[l_1, l_2, l_3]^T$, $\mathbf{m}=[m_1, m_2, m_3]^T$, and $\mathbf{n}=[n_1, n_2, n_3]^T$, are the 3 by 1 vectors of the direction cosines of R_e in R_0. The parameters x_d, y_d, and z_d are the coordinates of the origin of R_e in R_0.

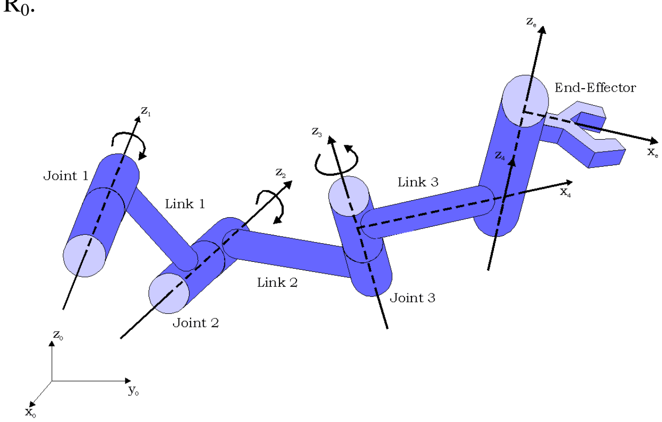

Figure 2: 3R Open loop Spatial Manipulator

An important feature in the matrix definition above is the use of matrix $\mathbf{A_c}$. In general, six parameters are needed to describe one reference frame relative to another. The DH parameterization succeeds in using four parameters for the relative transformation between frames within the serial kinematic chain itself only after the various motion axes are fixed. However, a special treatment is required, either at the origin or at the end-effector of the serial chain, the latter case being used in this paper. Assuming directions for axes z_1, z_2 and z_3 relative to the fixed reference frame, then the displacement described by the product of matrices $\mathbf{A_0A_1A_2}$ can be treated as a displacement of the fixed reference frame to the location of frame R_3. At this stage, a general six-parameter displacement is needed to transform frame R_3 into the end-effector frame R_e. The transformations described by the matrices $\mathbf{A_3}$ and $\mathbf{A_c}$ provide the complete set of six parameters.

The loop closure equation of the manipulator is used to obtain the design equations:

$$\mathbf{A_0\,A_1\,A_2\,A_3A_c = A_h} \tag{5}$$

Equation (5) is a 4 by 4 matrix equation that results in six independent scalar equations. The right side of Equation (5), i.e. the elements of matrix $\mathbf{A_h}$, is known since they represent the position and orientation of frame R_e at each precision point. The left side of Equation (5) contains all the unknown geometric parameters of the manipulator which are the DH parameters a_i, α_i, d_i and θ_i for i=0, 1, 2, 3, and parameters ϕ and d of matrix $\mathbf{A_c}$. Joint angles θ_1, θ_2 and θ_3 have a different value for each precision point while all other 15 geometric parameters are constant. Thus for five precision points there are 30 unknown parameters in total, and there are 30 scalar equation that are obtained. Therefore, the maximum number of precision points for exact synthesis is five.

DESIGN EQUATIONS AT EACH PRECISION POINT

Using the loop closure equation of the manipulator (Equation 5), six scalar design equations are obtained at each precision point. The unknowns in these equations are the manipulator constant structural parameters and the joint variables θ_1, θ_2 and θ_3, which vary from precision point to precision point. To simplify the solution process, we eliminate the joint variables from the design equations at each precision point. Once the joint variables are eliminated, the new set of equations contains only unknowns that do not change from precision point to precision point. In this way, for each new precision point that is defined, new equations are added that have exactly the same form as for the first precision point. In this section we present the method to obtain design equations devoid of the joint variables.

From Equation (3), it can be seen that the 3rd and 4th columns of matrix $\mathbf{A_i^{-1}}$ are independent of joint angle θ_i. Therefore, if Equation (5) is written as:

$$\mathbf{A_1\,A_2 = A_0^{-1}\,A_h\,A_c^{-1}\,A_3^{-1}} \tag{6}$$

then the scalar equations that are obtained by equating the left and right side of the third and fourth columns of matrix Equation (6) will be devoid of joint angle θ_3.

From the third column of Equation (6), three scalar equations are obtained:

$$s_{\alpha_2}c_1s_2 + c_{\alpha_1}s_{\alpha_2}s_1c_2 + s_{\alpha_1}c_{\alpha_2}s_1 = c_0L_1 + s_0L_2 \tag{7}$$

$$s_{\alpha_2}s_1s_2 - c_{\alpha_1}s_{\alpha_2}c_1c_2 - s_{\alpha_1}c_{\alpha_2}c_1 = -c_{\alpha_0}s_0L_1 + c_{\alpha_0}c_0L_2 + s_{\alpha_0}L_3 \tag{8}$$

$$-s_{\alpha_1}s_{\alpha_2}c_2 + c_{\alpha_1}c_{\alpha_2} = s_{\alpha_0}s_0L_1 - s_{\alpha_0}c_0L_2 + c_{\alpha_0}L_3 \tag{9}$$

where $L_i=l_iA+m_iB+n_iC$, with i = 1, 2, 3, and $A=s\phi s\alpha_3$, $B=c\phi s\alpha_3$ and $C=c\alpha_3$ with $c\phi=\cos(\phi)$ and $s\phi=\sin(\phi)$.

From the fourth column of Equation (6), another three scalar equations are obtained:

$$\begin{aligned} &a_2c_1c_2 - a_2c_{\alpha_1}s_1s_2 + a_1c_1 + d_1s_{\alpha_1}s_1 \\ &= c_0(M_1+x) + s_0(M_2+y) - a_0 \end{aligned} \tag{10}$$

$$\begin{aligned} &a_2s_1c_2 + a_2c_{\alpha_1}c_1s_2 - d_2s_{\alpha_1}c_1 + a_1s_1 = -c_{\alpha_0}s_0(M_1+x) \\ &+c_{\alpha_0}c_0(M_2+y) + s_{\alpha_0}(M_3+z) - d_0s_{\alpha_0} \end{aligned} \tag{11}$$

$$\begin{aligned} &a_2s_{\alpha_1}s_2 + d_1 + d_2c_{\alpha_1} = s_{\alpha_0}s_0(M_1+x) \\ &-s_{\alpha_0}c_0(M_2+y) + c_{\alpha_0}(M_3+z) - d_0c_{\alpha_0} \end{aligned} \tag{12}$$

where: $M_i=l_iP+m_iQ+n_iR$, with i = 1, 2, 3 and $P=-a_3c\phi-d_3s\phi s\alpha_3$, $Q=a_3s\phi-d_3c\phi s\alpha_3$ and $R=-d_3c\alpha_3-d$.

Note that Equations (9) and (12) are free of θ_1, thus, c_2 and s_2 can be computed by these two equations and their analytical expressions are free of θ_1 also. Using this result, θ_2 is essentially eliminated, for c_2 and s_2 can be eliminated from any equation by substituting the above result.

The final step is to obtain equations free of θ_1. To obtain such equations, we will consider the matrix Equation (6) again, written here as,

$$\mathbf{A_L = A_R} \tag{13}$$

where $\mathbf{A_L = A_1\,A_2}$ and $\mathbf{A_R = A_0^{-1}\,A_h\,A_c^{-1}\,A_3^{-1}}$.

We will denote the third column vector of $\mathbf{A_L}$ and $\mathbf{A_R}$ as $\mathbf{U_L}$ and $\mathbf{U_R}$, respectively, and the fourth column vector of $\mathbf{A_L}$ and $\mathbf{A_R}$ as $\mathbf{V_L}$ and $\mathbf{V_R}$, respectively (Note: vectors $\mathbf{U_L}$, $\mathbf{U_R}$, $\mathbf{V_L}$ and $\mathbf{V_R}$ are 3 by 1; i.e. we neglect the fourth component which is the homogeneous coordinate). Then, we form the following three vector equations:

$$\mathbf{U_L \cdot V_L = U_R \cdot V_R} \tag{14}$$

$$\mathbf{V_L \cdot V_L = V_R \cdot V_R} \tag{15}$$

$$\mathbf{U_L \times V_L = U_R \times V_R} \tag{16}$$

Equations (14), (15) and (16) were originally proposed by Raghavan and Roth to solve the inverse kinematics problem of general six degree of freedom serial link manipulators [5]. The same equations are used here for the geometric design of 3R manipulators.

Equations (14), (15) and (16) give a total of five scalar equations. For Equation (16), only the third component is used, i.e.

$$\mathbf{U_L}(1)\mathbf{V_L}(2)-\mathbf{U_L}(2)\mathbf{V_L}(1) = \mathbf{U_R}(1)\mathbf{V_R}(2)-\mathbf{U_R}(2)\mathbf{V_R}(1) \quad (17)$$

It was found that Equations (14), (15) and (17) are naturally devoid of θ_1. With θ_2 eliminated by using the expressions of c_2 and s_2 calculated from Equations (9) and (12), the three equations are free of θ_1, θ_2 and θ_3 and have, respectively, the following form:

$$\sum_{X_j,X_k \in W} f_{X_j,X_k}(\alpha_0,\theta_0,\alpha_1)X_jX_k = 0 \quad (18)$$

$$\sum_{X_j,X_k \in W} g_{X_j,X_k}(\alpha_0,\theta_0,\alpha_1)X_jX_k = 0 \quad (19)$$

$$\sum_{X_j,X_k \in W} h_{X_j,X_k}(\alpha_0,\theta_0,\alpha_1)X_jX_k = 0 \quad (20)$$

Where W={λA, λB, λC, $\lambda c\alpha_2$, P, Q, R, d_2, a_0, a_1, d_0, d_1, 1} and $\lambda=a_2/s\alpha_2$.

Note that Equations (18), (19) and (20) depend also on the parameters l_i, m_i, n_i, (i=1, 2, 3) and x, y and z, which are defined at each precision point and vary from precision point to precision point. Therefore, each precision point contributes three design equations (i.e. Equation (18), (19) and (20)), which are devoid of the joint variables and have as unknowns only the 15 constant structural parameters.

SOLUTION PROCEDURE USING INTERVAL METHOD

To solve the five points synthesis problem we propose to use an interval analysis based approach. Interval analysis requires a careful analysis of the problem to solve. Indeed there are numerous ways to transform a given problem in a set of equations with different number of unknowns and various complexity for the equations. Finding the more appropriate formulation of the problem is one of the main difficulties when using interval analysis. Indeed one may assume that having the least number of unknowns is the best choice as this will reduce the number of bisection. This is often true but may be wrong if the associated equations are very complex: indeed due to the over-estimation of interval arithmetic, complex equations will require a large number of bisection before we can insure that there is no solution within a given box. On the other hand, having more unknowns but simpler equations may lead to very accurate interval evaluation of the equations and, on the whole, to a better efficiency.

In our problem, we may consider either a problem with the smallest number of unknowns or a problem with the maximum number of unknowns. In the former case, we use all five Equation (9), (12) and one of five Equation (17). This problem is denoted the F_{11} problem and we have a set of 11 unknowns V_{11}={a_2, d_1, θ_0, α_0, α_1, α_2, $\theta_2(1)$, $\theta_2(2)$, $\theta_2(3)$, $\theta_2(4)$, $\theta_2(5)$}. The difficulty of using this system of equations is that the equations are very complex and the computation of the Jacobian and Hessian is very complicated. In the later case, we may define a problem with 30 unknowns, denoted as the F_{30} problem, where the set of unknowns V_{30} are a_0, d_0, a_1, d_1, a_2, d_2, P, Q, R, A. B, C and the sine and cosine of the angles θ_0, α_0, α_1, α_2 and of the 5 joint angles $\theta_2(i)$. The equations used are five equations from each of (9), (12), (14), (15) and (17), together with the trigonometric identity $\cos^2\rho+\sin^2\rho=1$ for each of the angles used in V_{30}. Note that there is more than 30 equations in F_{30}.These equations are structurally quite simple compare with those used in F_{11}. There is a one-to-one relation between the unknowns in the F_{11} problem and the unknowns in the F_{30} problem: being given a set of variable for F_{11} we may calculate uniquely the corresponding set of unknowns for F_{30}.

For solving the 5 points problem we have decided to use a hybrid approach: the basic set of unknowns will be the unknowns of F_{11} but we will also use the equations of F_{30}. The procedure of this is:

1. The filtering operation will be used first on the unknowns of F_{11}.
2. Interval evaluation will be performed first on the equations of F_{11}.
3. If a given box of F_{11} is not eliminated, then the unknowns are converted to the unknowns of F_{30}.
4. Filtering and interval evaluation of the equations are performed for the F_{30} problem.
5. Existence and uniqueness operator will then be used for F_{30}.
6. Eventual improvement on V_{30} will be used to obtain a new set for V_{11}.

The detailed implementation of this algorithm is described below.

SEARCH DOMAIN

The variables in the F_{11} problem are a_2, d_1, while the other parameters are angles. For the later parameters an evident choice for the search domain is [-π, π]. For a_2, we can restrict ourselves to positive values (negative solution exist but they will lead to the same robot design) and, clearly, a_2 cannot be 0.

As for the maximum value, we have decided to use roughly the maximum distance D between the precision points and the origin: hence the search domain for a_2 has been fixed to [0.8, D]. For d_1, the search domain is fixed to [-D,D].

We have also fixed a search domain for the variables in the F_{30} problems, using the same rule. Hence, boxes for the F_{11} problem that lead to variables for F_{30} outside the search domain will be rejected.

GETTING THE F_{30} UNKNOWNS

As mentioned previously there is a one-to-one relation ψ between the variables V_{11} of F_{11} and the variables V_{30} of F_{30}. Hence being given ranges for V_{11} we are able to compute ranges for V_{30}. As ψ is relatively simple we may compute the derivatives of each variable in V_{30} with respect to the variable of V_{11}. We can also compute the interval evaluation of these derivatives using the intervals of the unknowns. Let S_{ij} be the derivative of the variable u_i in V_{30} with respect to the variable v_j in V_{11}. If the interval evaluation of one derivative S_{ij} has a constant sign, for example as the lower bound of the evaluation is positive, then a better evaluation of the variable u_i of V_{30} may be obtained. Indeed its minimum will be obtained by fixing the value of the variable v_j of V_{11} to its lower bound and the maximum to the upper bound and the interval evaluation of the variable u_i will be performed with v_j having now a constant value instead of a range, thereby possibly leading to a restricted range. Note that the computation has to be done recursively as fixing the value of the variable v_j in V_{11} may imply that another derivative which was not of constant sign when computed with the range for the variable v_j may have a constant sign when computed with a constant value for the variable v_j.

FILTERING WITH THE 2B METHODS

The 2B method is implemented in F_{11} by using equation (12) that may be written as $H_{1k}a_2+d_1+H_{2k}=0$, where H_{1k}, H_{2k} are functions of the others unknowns and of the precision points k. We first write $d_1=-H_{2k}-H_{1k}a_2$ and consider as range for d_1 the intersection between the interval evaluation of the left and right terms. In a second step, if the interval evaluation of H_{1k} does not include 0, we write $a_2=(-d_1-H_{2k})/H_{1k}$ and update a_2 in the same manner.

Now consider two equations (12) obtained for the precision points k and j. If we subtract these two equations we get $(H_{1k}-H_{1j})a_2 + (H_{2k}-H_{2j})=0$.

Provided that the interval evaluation of $(H_{1k}-H_{1j})$ does not include 0, we write $a_2=-(H_{2k}-H_{2j})/(H_{1k}-H_{1j})$ and update eventually the range for a_2 by the intersection of the current range of a_2 with the range of $-(H_{2k}-H_{2j})/(H_{1k}-H_{1j})$.

On the other hand the 2B method can be used in F_{30} for any variables in equations (9) and (12) which are polynomial of degree 1 in each of the variable.

Note that the 2B method may be used more than once: indeed as soon as a range for a variable is changed at one step of the process, other variables that were not modified at a previous step, may now be improved. However, the rate of improvement is usually decreasing very fast and hence we repeat the 2B method only if the change in at least one variable is greater than a fixed threshold.

FILTERING USING THE SIMPLEX METHOD

A drawback of filtering only with the interval evaluation of the equations or by using the 2B method is that each equation is considered independently (these methods are often called “local” method). It would be interesting to use a method that consider the whole set of equations or, at least, a subset of the equations (this type of method is usually called a ”global” method). In our solving procedure we use a global method initially proposed by Yamamura [28]. Let x_i be a variable in V_{30} and let x_{i1}, x_{i2} denote the lower and upper bound of the current range for this variable. We define now a new variable u_i such that $u_i = x_i-x_{i1}$ which has a range $[0,x_{i2}-x_{i1}]$. By substituting x_i by u_i+x_{i1} for every variable of V_{30} in equation (9) or (12), we get a polynomial equation in the variables u_i. Each of these equations F_j may be written as:

$$F_j = G_j + \sum_{i=1}^{30} b_i u_i \quad (21)$$

where b_i are constants and G_j are non linear function of the u_i’s. Using interval arithmetic, we may find bounds for G_j, that is, $L_j \le G_j \le U_j$.

We define now new variables y_j as $y_j= G_j$ and the set of equations (9), (12) is now a set of linear equations in the variables y_j, u_i. These variables are also submitted to linear constraints, defined by the previous inequalities and the inequalities provided by the range on u_i. Hence we may use the simplex method that, in its first step, allows us to determine if there is a feasible region for the system (otherwise the current box can be eliminated as it will not include a solution). We may also use the simplex algorithm as an optimization method that will try to find successively the minimum and maximum value of the variable u_i. If a value greater than 0 is obtained for the minimum and a value lower than $x_{2i}-x_{1i}$ is obtained for the maximum, then the range on the variable u_i is improved. If such case occurred it is necessary to compute again the value of the coefficients b_i’s together with the interval evaluation of the G_j’s.

USING THE NEWTON SCHEME

When processing a box we apply systematically the Newton iterative scheme on the F_{30} problem with as initial guess the center of the box and allowing only a limited number of iteration. If the scheme converge we then apply the inflation method of Neumaier that enable one to verify that the solution found by the Newton scheme is a real solution of the system and to determine a box that include only this solution.

This box may be outside our search domain in which case we just store the solution for later analysis. If the box is included in the search domain then the solution is stored although the solution may not belong to the current box. Hence before processing a box we examine if one of the solution intersects the current box or even covers the current box. In the later case we just skip the processing of the current box. If there is only an intersection between a solution and the current box we modify one of the range of the current box in order to avoid getting the same solution. More precisely if [a,b] is a range of x_i for the current box and [u,v] the corresponding range for the solution:

1-if u is in [a,b] and v is not in [a,b], then [a,b] is changed to [u,b]

2-if u is not in [a,b] and v is in [a,b], then [a,b] is changed to [a,v]

3-if u and v are in [a,b], then we change the range of the the current box to [a,u] and we create a new box which has the same ranges than the current box, except for the variable x_i which has the range [v,b].

IMPLEMENTATION AND NUMERICAL EXAMPLE

Our solving program has been written using our C++ interval analysis library ALIAS. This library has a Maple interface that enable one to produce most of the necessary C++ code directly within Maple.

The solving program is run on a cluster of PC's. A master program manages the list of boxes and distributes the load among the various slaves using PVM. As soon as a slave is free, the master program will send the next box to the slave. If no slave is available the master program will process the current box, this processing being stopped as soon as a slave has emitted a message indicating that it is free. The slaves run the same slave program which takes as input a box and returns as soon as either it has been determined that the box does not include a solution or that a fixed number of new boxes are present in the list of the slave, in which case these boxes are returned to the master.

The five precision points selected have the following $\mathbf{A_{hi}}$ matrices:

$$\mathbf{A_{h1}} = \begin{pmatrix} -.6396094375 & .1435961208 & 0.755168803 & 8.310644971 \\ -.6265434807 & .4717800207 & -.6203764008 & -1.993959918 \\ -.4453571983 & -.869944691 & -.2117857403 & 4.525646630 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$\mathbf{A_{h2}} = \begin{pmatrix} .4273095207 & -.3048426696 & .8511624523 & 8.462432080 \\ .7180580935 & -.4576191690 & -.5243827518 & 3.909344844 \\ .5493624920 & .8352578302 & .02334971838 & 3.781393231 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$\mathbf{A_{h3}} = \begin{pmatrix} .2085023533 & .2490486651 & .9457809106 & 8.213357066 \\ -.4704189878 & -.8222864878 & .3202357065 & 4.720930002 \\ .8574571385 & -.5116831972 & -.0542914469 & 1.906020548 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$\mathbf{A_{h4}} = \begin{pmatrix} -.2651650429 & .5540136722 & .7891491309 & 6.610088080 \\ -.8775374786 & .2004602816 & -.4355957403 & -.9786178219 \\ -.3995190528 & -.8080127018 & .4330127018 & 7.933012701 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$\mathbf{A_{h5}} = \begin{pmatrix} -.5451561411 & -.5421432835 & .6394415077 & 7.498628082 \\ -.2838098567 & -.5983617170 & -.7492764648 & -2.362107226 \\ .7888325214 & -.5899524689 & .1723349570 & -.5803329915 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

When writing this paper, full results for the five precision points problem were not available. However, after 5 days of computation the algorithm has already found 6 solutions that were inside our search domain and 20 solutions that were outside. The 6 solutions that are inside the search domain are reported in Table 1. The 20 solutions outside the search domain are reported in Table 2.

CONCLUSIONS

In this paper, the geometric design problem of serial-link spatial robot manipulators with three revolute (R) joints is studied using an interval method. Five spatial positions and orientations are defined and the DH parameters of the 3-R manipulator are computed so that the manipulator will be able to place its end-effector at these five pre-specified locations. Interval method is used to search for design solutions of the design equations within a predetermined domain. At the time of writing this paper, six design solutions within the search domain and an additional twenty solutions outside the domain are found. This is an important new result for a very difficult problem related to the exact synthesis of spatial manipulators, that has not solved before. It will be useful because it can give insight on both the number and the nature of design solutions for the synthesis of the 3R.

ACKNOWLEDGEMENTS

This work was supported by a National Science Foundation CAREER Award to Professor Mavroidis under the grant DMI-9984051. Mr. Eric Lee was supported by a Computational Sciences Graduate Fellowship from the Department of Energy.

REFERENCES

[1] Suh, C. H., and Radcliffe, C. W., 1978, *Kinematics and Mechanism Design*, Wiley and Sons, New York, New York.

[2] Tsai, L. W., 1972, *Design of Open Loop Chains for Rigid Body Guidance*, Ph.D. Thesis, Department of Mechanical Engineering, Stanford University

[3] Roth, B., 1986, "Analytic Design of Open Chains," *Proceedings of the Third International Symposium of Robotic Research,* Editors: O. Faugeras and G. Giralt, MIT Press.

[4] Bodduluri, M., Ge, J., McCarthy, M. J. and Roth, B., 1993, "The Synthesis of Spatial Linkages," in *Modern Kinematics: Developments in the Last Forty Years*, Editor Erdman A., John Willey and Sons.

[5] Raghavan, M. and Roth, B., 1995, "Solving Polynomial Systems for the Kinematic Analysis and Synthesis of Mechanisms and Robot Manipulators," *Transactions of the ASME, Journal of Mechanical Design*, Vol. 117, pp.71-78.

[6] Erdman, A. G. and Sandor, G. N., 1997, *Mechanism Design: Analysis and Synthesis,* Vol. 1. Third Edition, Prentice-Hall, Englewood Cliffs, N.J.

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

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

[9] Roth, B., 1986, "Analytical Design of Two-Revolute Open Chains," *Proceedings of the Sixth CISM-IFToMM Symposium on the Theory and Practice of Robots and Manipulators,* The MIT Press, Cambridge, MA, Malczyk G. and Morecki A. (eds.), pp. 207-214.

[10] Perez, A. and McCarthy, M. J., 2000, "Dimensional Synthesis of Spatial RR Robots," *Advances in Robot Kinematics*, Lenarčič J. (ed.), Piran-Portoroz, Slovenia, June 26-30, 2000.

[11] Mavroidis, C., Lee, E. and Alam, M., 2001, "A New Polynomial Solution to the Geometric Design Problem of Spatial R-R Robot Manipulators Using the Denavit and Hartenberg Parameters," *Transactions of the ASME, Journal of Mechanical Design*, Vol. 123.

[12] Huang, C. and Chang, Y.-J., 2000, "Polynomial Solution to the Five-Position Synthesis of Spatial CC Dyads via Dialytic Elimination," *Proceedings of the ASME Design Technical Conferences,* September 10-13, 2000, Baltimore MD, Paper Number DETC2000/MECH-14102.

[13] Murray, A. P. and McCarthy, J. M., 1999, "Burmester Lines of a Spatial Five Position Synthesis from the Analysis of a 3-CPC Platform," *Transactions of the ASME, Journal of Mechanical Design*, Vol. 121, pp. 45-49.

[14] Neilsen, J. and Roth, B., 1995, "Elimination Methods for Spatial Synthesis," *1995 Computational Kinematics*, Edited by Merlet J. P. and Ravani, B., Vol. 40 of *Solid Mechanics and It's Applications*, pp. 51-62, Kluwer Academic Publishers.

[15] Innocenti, C., 1994, "Polynomial Solution of the Spatial Burmester Problem," *Mechanism Synthesis and Analysis*, ASME DE-Vol. 70,.pp. 161-166.

[16] McCarthy, M., 2000, "Chapter 11: Algebraic Synthesis of Spatial Chains," in *The Geometric Design of Linkages,* New York, McGraw Hill Book Company.

[17] Morgan, A. P., Wampler, C. W. and Sommese, A. J., 1990, "Numerical Continuation Methods for Solving Polynomial Systems Arising in Kinematics," *Transactions of the ASME, Journal of Mechanical Design,* Vol. 112, pp. 59-68.

[18] Roth, B. and Freudenstein, F., 1963, "Synthesis of Path Generating Mechanisms by Numerical Methods," *Transactions of the ASME, Journal of Engineering for Industry,* Vol. 85B, pp. 298-306.

[19] Morgan, A. P. and Wampler, C. W. 1990, "Solving a Planar Four-Bar Design Problem Using Continuation,"

Transactions of the ASME, Journal of Mechanical Design, Vol. 112, pp. 544-550.

[20] Wampler, C. W., Morgan, A. P. and Sommese, A.J., 1992, "Complete Solution of the Nine-Point Path Synthesis Problem for Four-Bar Linkages," *Transactions of the ASME, Journal of Mechanical Design*, Vol. 114, pp. 153-159.

[21] Dhingra, A. K., Cheng, J. C. and Kohli, D., 1994, "Synthesis of Six-Link, Slider-Crank and Four-Link Mechanisms for Function, Path and Motion Generation Using Homotopy With m-Homegenization," *Transactions of the ASME, Journal of Mechanical Design,* Vol. 116, pp. 1122-1130.

[22] Lee E. and Mavroidis C., “Solving the Geometric Design Problem of Spatial 3R Robot Manipulators Using Polynomial Continuation,” *Journal of Mechanical Design, Transactions of the ASME*, 2002 (in press).

[23] Moore, R. E., *Interval Analysis*, Prentice Hall, Englewood Cliffs, NJ, 1966.

[24] Moore, R. E., “Automatic error analysis in digital computation”, Lockheed Missiles and Space Co. Technical Report LMSD-48421, Palo Alto, CA, 1959.

[25] Neumaier, A., *Interval Methods for Systems of Equations*, PHI Series in Computer Science, Cambridge University Press, Cambridge, 1990.

[26] Collavizza, M., Deloble F. and Rueher, M., “Comparing partial consistencies”, *Reliable Computing*, Vol. 5, pp 1-16, 1999.

[27] Tapia, R.A, “The Kantorovitch theorem for Newton's method”, *American Mathematic Monthly*, Vol. 78, Number 1.ea, pp 389-392, 1971.

[28] Yamamura, K. and Kawata, H. and Tokue, A., “Interval solution of nonlinear equations using linear programming”, *BIT*, Vol. 38, Number 1, pp 186-199, 1998.

[29] Pieper, D. and Roth, B., 1969, "The Kinematics of Manipulators under Computer Control," *Proceedings of the Second World Congress on the Theory of Machines and Mechanisms*, Zakopane, Poland, Vol. 2, pp. 159-16

Table 1: The Six Solutions that are Inside the Search Domain

	#1	#2	#3	#4	#5	#6
θ_0	.775233248e-1	-.558620402	-.887597446	.785398163	1.02815630	.467426668
α_0	-1.35006277	-1.55503283	-1.77055159	1.57079633	-1.32132713	.317368142
α_1	.778968214	.755785230	.857624579	1.04719755	2.39035747	.985347661
α_2	-1.47839137	-1.75037402	-1.52621309	-1.04719755	-2.79636601	-1.53692129
α_3	-1.34748462	1.35240628	1.37593855	.523598775	-.298230906	1.43586836
ϕ	-.677568166	-.824382301e-1	-.169340204e-2	1.04719755	-1.25152760	-.544689358
a_0	3.669802832	1.899572388	1.593159687	2.000000002	1.716349561	4.279703528
a_1	3.210372245	1.553570225	1.347395742	1.999999999	3.758534112	4.083721705
a_2	3.706618339	5.082353433	6.233230137	3.000000001	3.467544655	4.238238536
a_3	2.45411115	1.97078235	1.92083172	1.00000000	2.59356743	-2.34566763
d_0	-.1387432034e-1	.5374031497	.6756925237	2	1.497453273	-.6540743372
d_1	1.163390591	-.3839799612	-.1402507203	2	.7137420168	1.416388129
d_2	3.133574696	5.410782491	5.00638551	2.999999999	3.064435366	1.363225287
d_3	1.57538769	-.253103746	-1.21138132	1.00000001	-3.14060054	.4598868e-1
d	4.75122014	2.72334255	1.75524749	1.00000000	7.64504452	4.75277166
$\theta_1(1)$	-.917815361	-.569661836	-.832679011	1.04719754	-1.36859116	2.71876708
$\theta_1(2)$	-.404034827	1.35742784	1.19051562	1.04719755	.398669826	.978875535
$\theta_1(3)$	-2.44319716	-2.61861726	-2.66945034	.523598786	.660884293	.840817132
$\theta_1(4)$	-1.50428225	-.969955998	-1.07498897	1.61596841	-.932546645	2.79046807
$\theta_1(5)$	2.29730358	3.05743854	-2.96386783	-.785398162	-.511547197e-1	-1.15675133
$\theta_2(1)$	.171354876	.396771184	.695755672	.785398163	-.322866871	2.42698301
$\theta_2(2)$	2.60768841	2.31728491	2.57239980	-.785398163	.376399945	.820193236
$\theta_2(3)$	2.75808653	1.93194838	1.65401499	-.785398163	1.97888095	2.58794420
$\theta_2(4)$	.629554463	.259163421	.343466040	-.618054498e-10	1.21020939	2.17836416
$\theta_2(5)$	-2.13340824	3.10075861	2.69936837	.785398163	-1.58493203	-2.66693630
$\theta_3(1)$	-1.33918732	2.16147876	2.08098896	1.04719756	-.276517905	2.12652332
$\theta_3(2)$	1.49897572	-.727847406	-.439250834	-1.04719755	1.82387667	-.454995625
$\theta_3(3)$	-.854266259	1.11440653	1.16238938	1.04719754	-1.24575318	-.801880494
$\theta_3(4)$	-.453046561	1.84747794	1.80566240	1.00202547	.520333244	1.52123150
$\theta_3(5)$	3.13388122	1.82531601	1.85742067	1.57079633	1.75461360	-1.19894860

Table 2: The Twenty Solutions that are Outside the Search Domain

	#1	#2	#3	#4	#5
θ_0	1.17359169	.350585393	-.292390040	-.142843941	1.31232855
α_0	1.34412025	1.89862600	1.78652178	1.91911139	1.11513011
α_1	1.91230361	-2.35784644	-2.37655993	-.701471841	1.47064063
α_2	-.937977154	-2.69377363	-2.41642142	-.767524643	-1.62767951
α_3	-1.41662619	.455415593	.996433853	1.18822166	-1.35199268
ϕ	-.745781731	1.04168476	.317250175	.619059260e-1	1.03764795
a_0	7.126804225	4.27568738	2.971134202	3.026250654	6.023782775
a_1	-3.707867014	.8880056112	-.4624035243	-1.42860136	2.114621109
a_2	27.85066177	-3.961430431	1.939695187	-2.34430932	-16.01751106
a_3	-2.01055899	-1.64100629	1.56015173	2.24599679	-.350620268
d_0	-.6695703009	2.940274108	2.573218103	2.132502749	1.851980345
d_1	-24.0706679	-.3707389167	.4965308263	1.80980333	4.429964231
d_2	-14.21480943	10.31621001	8.419468479	-8.785348923	.6608324781
d_3	12.0995048	16.1870022	5.08611217	3.95906716	-1.92840395
d	2.70904159	-3.3846047	2.96813554	4.84414996	20.1528657
$\theta_1(1)$	-.995452858	.578529478	-.102045718	-2.73713153	-1.92972170
$\theta_1(2)$	2.00164877	2.58446365	3.06137652	.419498681	1.07428276
$\theta_1(3)$	-2.70905922	1.88226784	2.18270335	-.731924687	2.80839672
$\theta_1(4)$	-1.08037080	1.56341412	.638698219	-1.98672560	-2.10695605
$\theta_1(5)$	-2.61159757	-2.34784500	-3.10175623	.348761073	2.44830379
$\theta_2(1)$	1.28514037	-3.08310587	-.783063040	-.640327467e-1	.541734542
$\theta_2(2)$	.948668527	.738182258	.607244431	-.844897873	1.86208389
$\theta_2(3)$	1.05017352	.830886476e-2	1.25139530	-1.89953929	2.08267817
$\theta_2(4)$	1.33937359	-2.22159638	-.639168487	-.341148070	1.21065138
$\theta_2(5)$	.880369960	2.08374094	3.05114118	2.85315477	1.01346249
$\theta_3(1)$	-1.63932155	-1.32236390	1.41701205	-.943541356	-1.83934055
$\theta_3(2)$	-.668241253	-2.16887660	-1.34999730	1.78535324	-1.60749241
$\theta_3(3)$	-.198324045	-.891430329	.727867650	-2.04330587	-1.44094816
$\theta_3(4)$	-.995363093	-1.37042179	1.13755827	-1.23775513	-1.88333557
$\theta_3(5)$	-1.31229080	-.411928253	1.77900763	-1.21664090	-1.12714850

	#6	#7	#8	#9	#10
θ_0	.927600151	1.02999095	2.28936614	.586377149	2.02157383
α_0	1.54767820	1.64906511	2.43598637	1.79938177	1.88960132
α_1	-1.66415034	-2.57695764	-1.13349102	-2.47370576	-1.28960042
α_2	-1.29370977	-.745271301	-2.04995340	2.74389168	-.655366712
α_3	1.04735761	.891045311	.818680133	.620867904	.665620439
ϕ	.105632773	.730099304	.702998044	.841836522	.998618002e-1
a_0	.8844625893e-2	-.971537803e-1	-1.448107272	3.81236446	13.83458197
a_1	-.8066292833	4.963493871	.2027717983	.501006262	11.32977506
a_2	1.484299168	2.977955859	5.82555414	5.711073585	-.915858381
a_3	1.55979013	3.76856410	-.782345593	-.846028499	.291419623
d_0	2.035728545	.9673033976	7.750182015	3.466461772	-4.17632569
d_1	3.559738666	2.956152612	5.369348706	.8677974031	-.7277756598
d_2	-6.395541634	-9.853808516	-.6410145232	-3.447388904	10.86116344
d_3	-4.19783069	-4.38765427	2.09771516	1.32515612	-4.83191096
d	3.94570694	-2.41707714	-.32855449	-1.68753796	9.26166420
$\theta_1(1)$	1.62682146	.502843887	-.474452718	.672363475	1.86389817
$\theta_1(2)$	2.26857384	2.09325636	2.65751543	2.33643502	1.53185818
$\theta_1(3)$	.975227772	.453400860	-2.84076395	1.42827331	2.06287515
$\theta_1(4)$	1.91312169	1.32606720	-.692379002	1.82167171	1.88263083
$\theta_1(5)$	.137499569	-1.38964909	-2.24494572	-1.68520039	2.25680144
$\theta_2(1)$	-1.26907727	-1.94613927	2.91236347	-1.00580571	2.47423642
$\theta_2(2)$	1.01132874	1.77122427	-1.33709575	2.62873797	-1.16467520
$\theta_2(3)$	-.392248449	.111866097	-2.04425701	1.67552078	-2.72813665
$\theta_2(4)$	-.726977517	-.567372400	2.32906881	.531420032e-1	-2.70664285
$\theta_2(5)$	-1.18782335	-2.57381227	-1.04200746	-1.40042239	-.692091120
$\theta_3(1)$	-1.22601192	-.788178953	1.54657584	1.06051979	-.448546190
$\theta_3(2)$	2.21010746	.503958442	.392445721	.337340366e-1	.488655967
$\theta_3(3)$	2.82798025	1.76932088	1.71606193	1.36012109	2.96476296
$\theta_3(4)$	-1.12495801	-1.51666623	1.37097954	.972497968	-1.62522234
$\theta_3(5)$	-2.79999852	1.85311182	2.44974237	1.92330435	1.60155175

	#11	#12	#13	#14	#15
θ_0	2.05302298	1.36641900	1.34164380	-1.91462601	.864014281
α_0	2.10879236	-1.95348150	1.23703833	-1.49486428	1.83384086
α_1	2.26491086	-.756482027	-2.41634764	-.688456290	2.48297335
α_2	-1.79915152	-1.09529645	-2.09308102	-.651601726	-1.98733455
α_3	1.49271830	.835926185	.836784366	.641073663	.538724277
ϕ	-1.43383475	.545492482	.525565339	1.01880571	-.103772235
a_0	.1600848345	1.820282011	1.89935463	-2.617465405	1.751354712
a_1	-1.702207236	.5560028088	-.6317532668	-1.044325929	5.72716366
a_2	5.587558626	-1.601129656	1.525986228	-1.090043242	-6.392799157
a_3	-1.70323216	-.885874496	-.929809543	-.489824572	-2.82372970
d_0	4.692743264	2.177907462	2.347897549	3.638019137	2.440896476
d_1	10.28982073	-3.25431734	3.102100845	3.124552794	-15.04441253
d_2	5.471912845	-6.886364081	-7.156971637	8.748927392	-10.29254414
d_3	-.98992359e-1	-.74866297e-1	-.385237823	-4.20459045	7.32349530
d	7.47862199	-1.09011307	-1.18229780	-.5452035e-1	6.86349095
$\theta_1(1)$	1.57424012	-1.93210740	-1.16250163	-1.33658343	.363308623
$\theta_1(2)$	-2.96295738	1.52729490	1.70935740	1.81337490	-1.76288735
$\theta_1(3)$	-.905544422	1.67811101	1.50408607	1.36418796	.701153441
$\theta_1(4)$	.974382355	-1.06382723	-2.06493707	-2.37210987	1.52801781
$\theta_1(5)$	-.630064274	-2.51778788	-.678353633	-1.34902689	-2.14842832
$\theta_2(1)$	-.853817880	-1.11319280	-1.11730469	1.21152471	-.150219571
$\theta_2(2)$	-2.39916423	-2.56237172	-2.46036481	2.27688668	.695305372
$\theta_2(3)$	-1.00298655	2.65648366	2.69393826	2.85088941	-.336738261
$\theta_2(4)$	-1.10241616	-2.13912080	-2.18432756	2.29109169	-.255847913
$\theta_2(5)$	-.184257553	1.16120196	1.09443778	-.985958189	-.284436577
$\theta_3(1)$	2.94758274	1.03062998	2.90261458	-.577476221	-3.03981571
$\theta_3(2)$	-1.49328345	.943337677	1.73776772	-1.62051249	2.96746511
$\theta_3(3)$	-2.32663102	-2.92016982	2.84533060	.228988792e-1	1.01714501
$\theta_3(4)$	2.66343866	.623156777	2.65122839	-.580969055	1.88295702
$\theta_3(5)$	-1.33471457	1.99590876	-2.63296598	.492764825	-1.87554440

	#16	#17	#18	#19	#20
θ_0	.946128931	1.06744276	1.42443877	1.00427511	.916076368
α_0	-1.38292767	-1.50654078	1.16098641	1.49206501	1.49722981
α_1	.222963132	-.367090838	2.42213668	-2.61126521	1.64091400
α_2	-1.79637794	-2.58056397	-.911709880	-.778875246	-2.91343232
α_3	1.49495006	.693470188	.588524816	1.29724338	1.56866510
ϕ	-1.00187384	-.678057038e-1	.965486696	-.730254557	-1.23410648
a_0	15.47654258	6.529515575	5.396430835	6.147850817	5.11206029
a_1	9.453350669	17.71415622	-2.195047774	-9.459050441	13.49236541
a_2	-10.58316335	1.575019224	10.17630488	-4.259134281	4.956707472
a_3	-5.26523936	-6.74819300	-1.67817804	-5.89701657	10.1276314
d_0	15.38042453	-10.38341138	2.317895327	-4.290465927	-3.98137681
d_1	-66.13075549	-18.62759207	3.414708243	50.00923713	7.016307489
d_2	47.3481895	18.00853108	7.812650287	51.82135966	-35.78528046
d_3	1.16852754	5.41643391	-14.0345036	22.2732549	31.9844308
d	-12.6649384	-.61162223	27.6477603	-4.99668554	-4.70704071
$\theta_1(1)$	-2.61998683	-2.62304591	1.76592587	.974499806	1.52127262
$\theta_1(2)$	2.78515628	-1.35930588	-2.86076872	-1.47643365	-1.15373351
$\theta_1(3)$	1.54497786	-2.02446506	-.957408939	-.299537667	.184596289
$\theta_1(4)$	-1.72624727	-2.55819802	.760710378	.875099098	1.50392961
$\theta_1(5)$	-3.03432324	-2.20139415	.389883118	-.525980262e-1	.293318748
$\theta_2(1)$	1.17300655	1.90009881	-.985519437	2.93719232	2.24794540
$\theta_2(2)$	-1.23707796	-2.05252938	3.03536348	-2.67453741	-2.34164688
$\theta_2(3)$	-1.59599660	.846294187	2.95956937	2.74024239	-.502361134
$\theta_2(4)$	.226573061	.657678558	-2.06400445	2.51977153	.998789140
$\theta_2(5)$	3.02960299	-2.65773057	-1.36410453	-2.76391158	-3.02128679
$\theta_3(1)$	-2.90317444	2.96758641	.490925524	1.44624621	-1.64601022
$\theta_3(2)$	3.14132240	-2.65890997	-.199768025	1.45549735	.658659166e-1
$\theta_3(3)$	1.95972936	.718407721	-2.79854815	-.739469468	1.61870876
$\theta_3(4)$	2.54772866	1.61757990	-.249798958	1.74702713	-.518615869
$\theta_3(5)$	-2.88247034	-2.39157160	2.32856680	-.646110298e-1	.378326100