Inverse kinematics of six-degree of freedom “general” and “special” manipulators using symbolic computation

Article in Robotica · September 1994

DOI: 10.1017/S0263574700017975

2 authors:

F.B. Ouezdou
Université de Versailles Saint-Quentin, VEDECOM Institute

Philippe Bidaud
Sorbonne Université

Robotica (1994) volume 12, pp 421–430. © 1994 Cambridge University Press

Inverse kinematics of six-degree of freedom "general" and "special" manipulators using symbolic computation

C. Mavroidis,* F. B. Ouezdou and P. Bidaud

Laboratoire de Robotique de Paris, Université Pierre et Marie Curie, Tour 66, 2ème étage, 4 Place Jussieu, 75252, Paris France

(Received in Final Form: December 2, 1993)

SUMMARY

This paper presents an algorithm that solves the inverse kinematics problem of all six degrees of freedom manipulators, "general" or "special". A manipulator is represented by a chain of characters that symbolizes the position of prismatic and revolute joints in the manipulator and the special geometry that may exist between its joint axes. One form of the loop closure equation is chosen and the Raghavan and Roth method is used to obtain symbolically a square matrix. The determinant of this matrix yields the characteristic polynomial of the manipulator in one of the kinematic variables. As an example of the use of this algorithm we present the solution to the inverse kinematics problem of the **GMF Arc Mate** welding manipulator. In spite of its geometry, this industrial manipulator has a non-trivial solution to its inverse kinematics problem.

KEYWORDS: Inverse kinematics; Serial manipulators; "Special manipulators"; Symbolic computation.

1. INTRODUCTION

Inverse kinematics of serial manipulators is the problem of finding the values of the joint variables (joint angles or offsets) when the manipulator geometry is known and an end effector position and orientation are specified. The methods that calculate the solutions of this problem are either numerical or algebraic.

We need to define the following terms: A univariate polynomial in one of the kinematic variables that gives all the solutions to the inverse kinematics problem for this variable is called **characteristic polynomial**. If certain special conditions of the Denavit and Hartenberg parameters exist then the manipulator is said to have a **special geometry**. If, in addition, these special conditions cause a manipulator to have a lower degree characteristic polynomial than the general manipulator of the same type, we call this manipulator: **special manipulator**; otherwise the manipulator is called: **general manipulator**.

Numerical iterative schemes based on Newton–Rampson modified techniques,[1,2] or on optimization based methods[3,4] have been used to obtain some of the solutions to the inverse kinematics problem. In general, these methods are very fast but can't give all the solutions. Tsai and Morgan[5] applied continuation methods to solve inverse kinematics and they succeeded to find numerically 16 distinct solutions for a **6R** of general geometry which is the correct number of solutions.

Algebraic methods are of interest because they give all the solutions. These methods can also be used for symbolic computation of the solutions. The first correct and complete algebraic solution has been given by Lee and Liang[6–8] for **6R** and **5R1P** manipulators of general geometry. They calculated the 16th degree characteristic polynomial in the tangent of the half angle of one of the joint variables, using spherical trigonometry.

Raghavan and Roth[9–11] presented the first method that may be used to calculate the characteristic polynomial of all general geometry manipulators (i.e. **6R, 5R1P, 4R2P, 3R3P**). The works of Manocha and Canney[12] and Kohli and Osvatic[13,14] are variants of this method. In both papers the writers succeeded in reducing the execution time of the method.

The main concept of the previous method can be applied to all general types of manipulators. However, when inverse kinematics of two different manipulators has to be solved, two different mathematical analyses are needed in each case (i.e. the loop closure equation can be written with a different form, the unknowns are different, special geometry into one of them may demand the modification of the method).

Our aim in this paper is to present a general algorithm based on the previous method that solves inverse kinematics of all six degree of freedom manipulators (**6R, 5R1P, 4R2P, 3R3P**), general or special with the same mathematical analysis.

Inverse kinematics of special manipulators has been studied in a systematic way by Pieper and Roth,[15] Smith,[16] and Mavroidis and Roth.[17] The main concern of these papers is to uncover general conditions on the structural parameters of a manipulator that will reduce the degree of its characteristic polynomial. Very often, for special manipulators, the Raghavan and Roth method has to be modified properly. These modifications have been taken into account by the present algorithm.

Symbolic computation substitutes the "hand made" mathematical development of equations that precedes the numerical programming. Symbolic calculation software packages for the inverse kinematics of manipulators are presented in Khalil and Bennis,[18] Rieseler and Wahl.[19] Both software packages deal with classes of

* Currently at Mech. Eng. Dept., MIT, Cambridge, Mass. (USA).

special manipulators. The symbolic computation program presented here, solves the inverse kinematics of both general and special 6 degree of freedom manipulators.

2. METHOD OF RAGHAVAN AND ROTH

In this paper we use the variant of the Denavit and Hartenberg notation[2] in which the parameters a_i, α_i, d_i, θ_i are defined as: a_i is the length of link $i+1$, 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. In this notation the ground is link 1 and the end-effector is link 7.

A complete description of the Raghavan and Roth method can be found in Raghavan and Roth.[9,11] Here we describe this method in outline form. The main steps of the method are:

(i) The homogeneous matrix transformation equation of a 6 degree of freedom serial manipulator is written under the form:

$$A_3A_4A_5 = A_2^{-1}A_1^{-1}A_HA_6^{-1} \tag{1}$$

(ii) The following 14 scalar equations which are devoid of θ_6 are calculated analytically:

$$l,\ p,\ lp,\ pp,\ lxp,\ l(pp)-2p(lp).$$

(Note: l and p are the vectors formed by the first three elements of columns 3 and 4 respectively of equation (1))

(iii) These equations form the system:

$$MX_l = NY \tag{2}$$

where M is a 14×9 matrix and N is a 14×8 matrix. Every element in matrix M is a quadratic in $x_3 = tan\,(\theta_3/2)$ while every element in N is constant. The vectors X_l and Y for the **6R** manipulator are equal to:

$$X_l = [s_4s_5,\ s_4c_5,\ c_4s_5,\ c_4c_5,\ s_4,\ c_4,\ s_5,\ c_5,\ 1]'$$

and

$$Y = [s_1s_2,\ s_1c_2,\ c_1s_2,\ c_1c_2,\ s_1,\ c_1,\ s_2,\ c_2]'$$

with $s_i = sin\,(\theta_i)$ and $c_i = cos\,(\theta_i)$. Every element of the vectors X_l and Y is called a **"power product"** of equations (2).

(iv) 8 of the 14 equations are used to eliminate Y; the resulting system of 6 equations takes the form:

$$\Sigma_l X_l = 0 \tag{3}$$

where Σ_l is a 6×9 matrix.

(v) The following substitution $sin\,(\theta_i) = 2x_i/(1+x_i^2)$ and $cos\,(\theta_i) = (1-x_i^2)/(1+x_i^2)$, where $i = 3, 4, 5$ and $x_i = tan\,(\theta_i/2)$, lead to the system form:

$$\Sigma_2 X_2 = 0 \tag{4}$$

where $X_2 = [x_4^2x_5^2,\ x_4^2x_5,\ x_4^2,\ x_4x_5^2,\ x_4x_5,\ x_4,\ x_5^2,\ x_5,\ 1]'$.

(vi) The 6 equations of (4) are multiplied by x_4 and the following 12×12 homogeneous system is obtained:

$$\Sigma X = 0 \tag{5}$$

where:

$$X = [x_4^3x_5^2,\ x_4^3x_5,\ x_4^3,\ x_4^2x_5^2,\ x_4^2x_5,\ x_4^2,\ x_4x_5^2,\ x_4x_5,\ x_4,\ x_5^2,\ x_5,\ 1]'$$

(for the **6R**)

(vii) the condition $det\,\Sigma = 0$ gives the characteristic polynomial of the manipulator in x_3. For a **6R** general manipulator the degree of this polynomial is 16.

(viii) We back substitute each real solution of θ_3 in step vi and calculate a unique value for θ_4 and θ_5 from the solution of a linear system. Further substitution of θ_3, θ_4, θ_5 in step iii gives a unique value for θ_1, θ_2. Finally, θ_6 follows from two elements of the first two columns of the closure equation.

It has been shown[10] that the same method can be applied to solve the inverse kinematics problem of general **5R1P**, **4R2P** and **3R3P** manipulators. The only difference is that, in the elements of vectors X_l and Y of equation (2), for a prismatic joint i, c_i is replaced by d_i and s_i is replaced by d_i^2.

This method has been used by Mavroidis and Roth[17] to identify six conditions between the Denavit and Hartenberg parameters that make a manipulator to become special. The inverse kinematic problem for these manipulators can be solved with the Raghavan and Roth method as well.

3. GENERAL ALGORITHM

Although the Raghavan and Roth method is general, its application on the inverse kinematics problem depends on the kinematic type and the geometry of the manipulator. In this paper we propose an algorithm that solves the inverse kinematics of all types of manipulators, general or special, with a unified way. We used Maple™, a symbolic computation software, for programming this algorithm. The main steps of the algorithm, that are represented in Figure 1 of Appendix 2, are the following:

3.1 Type of manipulator

A manipulator is represented with a chain of characters. Using a notation introduced by Mavroidis and Roth[17] (see Appendix 1) the joint type and the geometry of the manipulator are determined. For example two consecutive parallel revolute joints are represented as **R'R'**.

In every chain of characters there must be 6 characters equal to **R** or **P** (a **C** character is counted as one **R** and one **P**). These characters are stored in an additional six-dimensional vector-array, ***ch***. This internal variable, is used to represent the general type of the manipulator and to identify the number and the position of prismatic joints in the manipulator. We can't have more than 3 **P** characters in a 6 degree of freedom manipulator representation.

In this **"interpretor"** part of the program the special geometry between the axes of the manipulator is identified and special values[20] that satisfy this geometry are set for the manipulator Denavit and Hartenberg parameters. For example, for the **RR'R'RR"R"** chain of characters the program will identify that this is a **6R** manipulator with joints 2 and 3 parallel, and joints 5 and 6 parallel, and automatically will set $\alpha_2 = 0°$ and $\alpha_5 = 0°$.

In addition, the "interpretor" detects special geometries that have no physical sense (ex.: **R' + R'** i.e. two parallel joints can't be perpendicular) or geometries that cause the manipulator to have less than 6 degrees of freedom[17] (ex. **R′ × R′**, two revolute axes are coincident).

3.2 Choice of the equation

There are six different ways of writing the loop closure equation:

$$A_1A_2A_3 = A_hA_6^{-1}A_5^{-1}A_4^{-1} \quad (6)$$

$$A_2A_3A_4 = A_1^{-1}A_hA_6^{-1}A_5^{-1} \quad (7)$$

$$A_3A_4A_5 = A_2^{-1}A_1^{-1}A_hA_6^{-1} \quad (8)$$

$$A_4A_5A_6A_h^{-1} = A_3^{-1}A_2^{-1}A_1^{-1} \quad (9)$$

$$A_5A_6A_h^{-1}A_1 = A_4^{-1}A_3^{-1}A_2^{-1} \quad (10)$$

$$A_6A_h^{-1}A_1A_2 = A_5^{-1}A_4^{-1}A_3^{-1} \quad (11)$$

These equations have the same general form which is described by the following algorithm:

```
# i is the subscript of the first matrix of the left
# hand side of equations (6) to (11). It can take the
# values from 1 to 6. All other subscripts i.e. j1, j2,
# j3, k1, k2, k3, m, are defined from i. These eight
# subscripts determine the order of multiplications
# of homogeneous matrices in the left and right
# part of equation (12). For each value of i equation
# (12) is one equation from equations (6) to (11).
for  i:=1  to  6  do
     m:=i+3;
     if  m>6  then  m:=m-6;
     fi;
     if  member(i, [1, 2, 3])  then
     j1:=u;         j2:=i+1;
     j3:=i+2;       k1:=m+3;
     k2:=m+2;  k3:=m+1;  else
     j1:=i+1;       j2:=i+2;
     j3:=i+3;       k1:=u;
     k2:=m+2;  k3:=m+1;                         fi;
     if  (k1>7)  then  k1:=k1-7;                fi;
     if  (k2>7)  then  k2:=k2-7;                fi;
     if  (j2>7)  then  j2:=j2-7;                fi;
     if  (j3>7)  then  j3:=j3-7;                fi;
     A_i A_j1 A_j2 A_j3 = A_k1^-1 A_k2^-1 A_k3^-1 A_m^-1;    (12)
od;
# Note: member(i, [1, 2, 3]) is true if i is 1 or 2 or 3.
```

In the general equation (12) when one of the subscripts is equal to u then the matrix A_u is equal to the 4 × 4 unitary matrix and when one of the subscripts is equal to 7 then the correspondent matrix A_7^{-1} is equal to A_h.

Anyone of the equations (6) to (11), will yield a characteristic polynomial which is expressed in a different unknown x_i depending what the kinematic variable of the matrix A_i of the general equation (12) is. The following test determines x_i:

if *ch[i]=R* **then** $x_i = \tan(\theta_i/2)$ **else** $x_i = d_i$

In general, all equations give the same final result and there is no reason to prefer one equation or the other. However, it turns out that for each equation there are some manipulators for which no characteristic polynomial is obtained. These manipulators are the following:

(i) The matrix A_m^{-1} must not have a prismatic joint kinematic variable

If the mth joint is revolute then the inverse homogeneous matrix A_m^{-1} of equation (12) has the 3rd and the 4th columns independent of the unknown θ_m. Thus the 14 scalar equations of step ii in section 2 are devoid of the kinematic variable of the matrix A_m^{-1} (i.e. θ_m). If this joint is a prismatic one, then the 14 equations are dependent of the kinematic variable of the matrix A_m^{-1} (i.e. d_m). In that case the number of power products in the right hand side of equation (12) is bigger and we'll need more equations than 14 to solve the inverse kinematic problem. For example, for the **RRRPRR** general type manipulator, equation (12) is used to solve inverse kinematics with the restriction that $i \neq 1$.

(ii) Manipulators 4R2P

General manipulators of the **4R2P** type have a characteristic polynomial of degree 8. This polynomial can be obtained as the determinant of either a 12 × 12 matrix or a 6 × 6 matrix[10] depending what the value of the subscript i of equation (12) is. For **4R2P** manipulators the following test identifies the dimensions of Σ matrix:

if $x_i = d_i$ **then** "Σ *is* 12 × 12" **else** "Σ 6 × 6"

The dimensions of Σ for **4R2P** manipulators decrease when $x_i = \tan(\theta_i/2)$. This is explained by the fact that less power products are in the left or right hand side and less equations are needed to solve inverse kinematics.[10] For these **4R2P** manipulators, only the first 11 of the 14 equations described in step ii of section 2 are used. So when inverse kinematics for a **4R2P** manipulator has to be solved, the program must identify if the equation chosen leads to 12 × 12 or to a 6 × 6 Σ matrix in order to avoid singular Σ matrices.

(iii) Manipulators 3R2P

The characteristic polynomial of **3R3P** manipulators has a degree equal to 2. For the same reasons as for the 4R2P manipulators, this polynomial is either obtained from the determinant of a 6 × 6 or 3 × 3 Σ matrix and the following test is used:

if $x_i = d_i$ **then** "Σ *is* 6 × 6" **else** "Σ *is* 3 × 3"

(iv) The Σ matrix has linear dependent columns

When the Denavit and Hartenberg constant parameters of a manipulator have special values, then the columns of the Σ matrix may become linear dependent and all the characteristic polynomial coefficients turn to 0.

The algebraic conditions that make the columns of Σ matrix to be linear dependent, correspond to the Denavit and Hartenberg parameters of matrices A_{j2} and A_{j3}

(equation (12)) as the following algorithm presents. The only exception is when $j_3 = 7$. In this case the previous conditions are satisfied with the Denavit and Hartenberg parameters of matrices A_{j1} and A_{j2}.

```
# This algorithm is a test for the Denavit and
# Hartenberg parameters that make the columns of Σ
# matrix linear dependent. We distinguish two groups
# of conditions:
# a) If Σ matrix dimensions are 12x12:

if ch[j2]=R and ch[j3]=R and ch[j3+1]=R
  then
    if ((α_j2 = 0°) and (α_j3 = 0°))
      or (a_j2 = 0) and (a_j3 = 0) and (d_3 = 0)
      or ((sin(α_j2)/a_j2 = sin(α_j3)/a_j3) and (d_j3 = 0))
    then "Σ has linear dependent columns";
    fi;
if ch[j2]=P and ch[j3]=R and ch[j3+1]=R
  then
    if (((α_j2 = 90°) or (α_j2 = 0°)) and (α_j3 = 0°))
      then "Σ has linear dependent columns";
    fi;
fi;
# b) If Σ matrix dimensions are 6×6:

if ch[j2]=R and ch[j3]=R and ch[j3+1]=R
  then
    if ((α_j2 = 0°) or (α_j3 = 0°))
    or ((a_j2 = 0) and (a_j3 = 0) and (d_3 = 0))
    or ((sin(α_j2)/a_j2 = sin(α_j3)/a_j3) and (d_j3 = 0))
      then "Σ has linear dependent columns";
    fi;
fi;
if ch[j2]=P and ch[j3]=R and ch[j3+1]=R
  then
    if (α_j3 = 0°)
      then "Σ has linear dependent columns";
    fi;
fi;
```

Similar conditions exist for the case when the prismatic joint is either the joint j_3 or the joint $j_3 + 1$.

We have to note that the method of finding the special values of Denavit and Hartenberg parameters that cause the columns of Σ matrix to be linear dependent is described in reference 20.

(v) The Σ matrix has linear dependent lines

In general, manipulators with less than 6 degrees of freedom (degenerate manipulators) will cause that all equations from (6) to (11), yield Σ matrices that have linear dependent lines.[20] That is the reason why in section 3.1 degenerate manipulators are excluded from the present study.

However, even if manipulators with less than 6 degrees of freedom are not studied from the program, for each equation, due to **numerical** reasons there are still some conditions on the Denavit and Hartenberg parameters that will lead to the result that the Σ matrix has linear dependent lines.[20] These conditions are described with the following algorithm:

```
# This algorithm is a test for the Denavit and
# Hartenberg parameters that make the lines of Σ
# matrix linear dependent. The variable o is used to
# determine for each equation the subscript that
# specifies these Denavit and Hartenberg parameters.

o := k3;
if k3=7     then o := 1;        fi;
if k3=6     then o := 5;        fi;
if ch[o]=R  and ch[o+1]=R  then
   if ((α_o = 0°) or (a_o = 0))   then
      "Σ has linear dependent lines";
   fi;
fi;
if ((ch[o]=P) and (ch[o+1]=R))
   or ((ch[o]=R) and (ch[o+1]=P)) then
   if ((α_o = 90°) then
      "Σ has linear dependent lines";
   fi;
fi;
```

If for a given value of the subscript i a manipulator satisfies the previous conditions, then the equation (12) for this value of i, can't be used to solve inverse kinematics for this manipulator. This result doesn't mean that the manipulator has less than 6 degrees of freedom, or that it is a special manipulator. It simply means that for this value of i, equation (12) will yield a Σ matrix with linear dependent lines.

The following example shows the way that the algorithm identifies the proper equation to solve inverse kinematics among equations (6) to (11), using the previous conditions (i) to (v):

Consider the manipulator **R⊥PRPR×R.** Despite the special geometric conditions between the joints, this manipulator is general. The 2nd and 4th joints are prismatic and therefore, due to condition (i), equations (6) and (10) can't be used. Due to condition (v), the special geometry of this manipulator will make the program not to use equations (7) and (8). So for this manipulator only equations (9) and (11) can be used to solve inverse kinematics. Due to condition (ii), equation (9) will calculate the 8th degree characteristic polynomial of this manipulator in d_4, through a 12×12 Σ matrix while equation (11) will calculate the 8th degree characteristic polynomial in $x_6 = \tan(\theta_6/2)$, from a 6×6 Σ matrix.

3.3 Definition of the vectors of power products

The vectors of power products of equations (2) to (5), i.e. the vectors X_1 and Y of equation (2), the vector X_2 of equation (4) and the vector X of equation (5) depend on the value of the subscript i of equation (12) and on the type of joints of the manipulator.

The power products of vector X_1 are defined with the following algorithm:

```
# The variables o1 and o2 specify for each equation
# the subscripts of the Denavit and Hartenberg
# unknown parameters that participate in X_1.
```

```
if  member(i,[1,2,3,6])
                then  o1:=j2;  o2:=j3;  fi;
if  i=4         then  o1:=j1;  o2:=j2;  fi;
if  i=5         then  o1:=j1;  o2:=j3;  fi;
if  ch[o]=R     then  e_o1:=sin(θ_o1); f_o1:=cos(θ_o1);
                else  e_o1:=d_o1^2; f_o1:=d_o1; fi;
if  ch[o2]=R    then  e_o2:=sin(θ_o2); f_o2:=cos(θ_o2);
                else  e_o2:=d_o2^2; f_o2:=d_o2;  fi;
X_1:=[e_o1 e_o2, e_o1 f_o2, f_o1 e_o2, f_o1 f_o2, e_o1, f_o1, e_o2, f_o2, 1]'
```

The power products of vector Y are defined with a similar procedure.

For **4R2P** manipulators that the Σ matrix dimensions are 6×6 and for **3R3P** manipulators, the vectors X_2 and X are not defined because equations (4) and (5) are not needed.

For **6R, 5R1P** and **4R2P** manipulators that Σ is a 12×12 matrix, the vectors X_2 and X have the following form:

```
g_o1:=x_o1;
#x_o1=tan(θ_o1/2)
if  ch[o2]=R  then g_o2:=x_o2 else g_o2:=d_o2; fi;
X_2:=[g_o1^2 g_o2^2, g_o1^2 g_o2, g_o1^2, g_o1 g_o2^2, g_o1 g_o2, g_o1, g_o2^2, g_o2, 1]'
X:=[g_o1^3 g_o2^2, g_o1^3 g_o2, g_o1^3, g_o1^2 g_o2^2, g_o1^2 g_o2, g_o1^2, g_o1 g_o2^2,
g_o1 g_o2, g_o1, g_o2^2, g_o2, 1]'
```

For these manipulators the joints o_1 and o_2 (these variables were defined in the beginning of the previous algorithm), can't be prismatic at the same time and with no loss of generality, joint o_1 can always be considered as a revolute one.

For the example of the previous section, i.e. the manipulator **R⊥PRPR×R**, equations (9) or (11) were chosen to solve inverse kinematics. For equation (11) the power products X_1 and Y are as follows:

$$X_1=[s_2d_2^2,\ s_1d_2,\ c_1d_2^2,\ c_1d_2,\ s_1,\ c_1,\ d_2^2,\ d_2,\ 1]'$$
$$Y=[s_5d_4^2,\ s_5d_4,\ c_5d_4^2,\ c_5d_4,\ s_5,\ c_5,\ d_4^2,\ d_4,\ 1]'$$

For this manipulator and this equation the Σ matrix dimensions are 6×6 and therefore there are no vectors X_2 and X.

3.4 Symbolic calculation of matrices M and N

For two different types of manipulators that the subscript i of equation (12) has the same value, the 14 equations of step ii in section 2 are exactly the same. On the contrary, for each manipulator type the vectors of power products X_1 and Y are different (see section 3.3), and therefore their coefficients (i.e. matrices M and N) are different.

The following example shows the way that from an expression u the vector of coefficients m is obtained when a vector of power product w is known:

Consider the expression: $u=axy^2+bx^2y+c$, and two different vectors of power products: $w_1=[y^2,y,1]'$ and $w_2=[x^2,x,1]'$. The same expression can be written in two different ways: $u=m_1w_1$ and $u=m_2w_2$ where $m_1=[ax,bx^2,c]$ and $m_2=[by,ay^2,c]$. Each time the vector w_i represent the unknowns of the problem and the vector m_i represent the constant coefficients. In this section we present the algorithm that calculates the vectors m_i when w_i is specified:

```
for  i:=1 to nops(u) do
  for j:=1 to vectdim(w) do
    if  ged(op(i, u), w[j]=w[j]      then      m[j]:=
        u[i]/w[j];
            u:=u-op(i, u); fi; od; od;

# gcd means great common divisor,
# nops means number of operands,
# op(i, u) means the ith term of u
# vectdim(w) is the dimension of w.
```

So for $u[1]=axy^2$ and $w[1]=y^2$, $m[1]=ax$. Each term of u that contributed with a coefficient is not going to be reexamined (for example the term $u[1]=axy^2$ that contributed with a coefficient of $w[1]$ can't contribute again with a coefficient of $w[2]$ or $w[3]$).

For inverse kinematics of manipulators the expression u is one of the 14 scalar equations defined in step 2 of section 2, the vector w is one of the vectors X_1 and Y, and the vector m is one line of the matrices M and N.

3.5 Symbolic calculation of matrix Σ_1

This step is called **"elimination phase"** because the vector Y of the right hand side power products is expressed as a linear function X_1.

The "elimination phase" presented in Raghavan and Roth[9,11] is valid only for **6R** manipulators and when equation (8) is used to solve inverse kinematics. In the present algorithm a general way to eliminate Y is used.

Two equations from the 14 of the system (2) are chosen for which two non-zero products can be specified, for example a and b, in the right hand side of both of them. Then, a 2×2 linear system in the unknowns a and b is solved. These expressions of a and b are linear functions of the left hand side power products (i.e. the components of X_1) and linear functions of the right hand side power products other than a and b (i.e. $Y-\{a,b\}$). If the expressions of a and b are substituted in the 12 equations that were not used then these equations are devoid of a and b. If this procedure is continued in the same way, then 6 equations are obtained that have 0 right hand side. These equations form the system (3).

The symbolic computation of matrices Σ_2 and Σ follows the steps v and vi of section 2.

3.6 Numerical part

It is impossible, till now, to obtain symbolically the characteristic polynomial. So steps (vii) and (viii) of section 2 are performed with the numerical values for the constant Denavit and Hartenberg parameters and for the desired A_h matrix.

From the analytical expression of Σ matrix, its determinant is determined for the numerical values of the known Denavit and Hartenberg parameters and for the desired A_h matrix. This determinant yields the characteristic polynomial of the manipulator. The computation of the determinant of Σ matrix is based on the multiple use of the Laplacian development of a determinant in 3×3 subdeterminants and thus the computation time is reduced.

The characteristic polynomial may have special characteristics if the manipulator is special. These characteristics,[17,20] may be: (i) existence of extraneous roots $\pm i$ represented by a number of factors of (x^2+1) (ii) some of the first or last coefficients are 0 (iii) the coefficients of the characteristic polynomial have symmetry properties. Each one of these characteristics reduces the degree of the characteristic polynomial. The program can identify the special type of manipulator and can eliminate each one of these characteristics. So the final characteristic polynomial in the chosen hidden variable has its minimum degree.

4. SYMBOLIC VERSUS NUMERICAL COMPUTATION OF INVERSE KINEMATICS

One of the questions that we had to answer was the following: "Where does symbolic computation stop and where do we start to perform computations with the numerical values of all known parameters?".

There is a minimum limit of symbolic computation that is needed. This limit corresponds at step iii of section 2, i.e. computation of matrices M and N. We need to have the analytical expression of these matrices. There is also a maximum limit of symbolic computation. This limit is the computation of Σ matrix i.e. step vi of section 2. Due to practical reasons (time, memory capacity) it is not possible to calculate symbolically the characteristic polynomial.

Thus, steps iv, v and vi of section 2 can be computed either symbolically or with the numerical values of the known parameters. There is no general rule where to separate the computations and just personal criteria can be set. From personal experience we can remark the following statements:

(i) For the most general manipulators of all types, it is better to stop the symbolic computation just after the computation of matrices M and N or at least after the computation of matrix Σ_l. Symbolic computations after these steps increase the number of terms and therefore the number of operations.

(ii) For manipulators with a lot of special geometric characteristics (as the manipulator presented in section 6), the analytical expressions of matrices Σ_2 and Σ are simple enough so as to be able to continue the symbolic computation until step 6 of section ii.

(iii) For all manipulators, general or special, an analytic expression of matrices M, N, Σ_l, Σ_2, Σ, is needed if a study on the conditions on the Denavit and Hartenberg parameters that make a manipulator to be special has to be made.[17,20]

5. LIMITS OF THE METHOD

For some manipulators with special geometry the method can't solve inverse kinematics. In addition, there are some special manipulators for which the method solves inverse kinematics but can't calculate the characteristic polynomial of minimum degree. These manipulators that are presented in the next paragraphs, are the exceptions in the generality of the algorithm proposed in this paper.

Manipulators with a characteristic polynomial of fourth degree or lower are called **analytic**. All known analytic manipulators are described in reference 20. It turns out that most of the analytic manipulators have a 4th degree polynomial in some of the kinematic variables, and for each value of the polynomial unknown there exist two values for some of the rest of the kinematic variables. For example **"wrist"** manipulators have a fourth degree characteristic polynomial in one of the **"arm"** kinematic variables. For each value of the polynomial unknown there is one value for the other "arm" kinematic variables and two values for the "wrist" kinematic variables. If the characteristic polynomial unknown were one of the wrist kinematic variables then its degree would have been 8.

The algorithm proposed in section 3 can solve inverse kinematics of all manipulators, analytic manipulators included. However for those analytic manipulators, like wrist manipulators, that the characteristic polynomial can have two different degrees, i.e. 4 and 8 the program can calculate only the characteristic polynomial of higher degree i.e. 8. The reason for this is that if the polynomial degree were 4, then in step viii of section 2 where a linear system is solved in order to calculate the rest of the variables, the linear system becomes singular; for one value of the 4th degree characteristic polynomial unknown, the linear system must calculate two distinct values for some of its unknowns which is impossible. The methods that yield the lower degree characteristic polynomial for all analytic manipulators depend on the manipulator type and geometry and are presented in reference 20.

For some manipulators with special geometry, inverse kinematics is not solved by the algorithm proposed in this paper. This is due to the fact that according to the conditions presented in section 3.2 no equation from equations (6) to (11) can be used to solve the inverse kinematics problem. For example, due to condition (4.2.v) the algorithm can't solve inverse kinematics for the manipulator $R \times R \times R \times R \times R \times R$. In general any manipulator with all pairs of consecutive revolute joints either parallel or intersected and all pairs of consecutive revolute and prismatic joints orthogonal, can't be solved by this algorithm.

6. EXAMPLE: GMF ARC MATE INVERSE KINEMATICS

In this section we present the inverse kinematics of the **GMF Arc Mate** welding manipulator as an example of the use of the algorithm.

The GMF Arc Mate is an industrial **6R** manipulator, shown schematically in Figure 2 (Appendix 3). Although the mechanical architecture is very similar to that of other industrial manipulators, the inverse kinematics solution for this manipulator is very different and much more complicated.

This special geometry manipulator is represented by the chain of characters: $\mathbf{R \perp R'(0)R' \perp R+R+R}$. The non-special Denavit and Hartenberg parameters as they are defined by the constructor are the following:

$a_1 = 200$ mm, $a_2 = 600$ mm, $a_3 = 130$ mm, $d_1 = 810$ mm, $d_3 = 30$ mm, $d_4 = 550$ mm, $d_5 = 100$ mm, $d_6 = 100$ mm and $\alpha_6 = 0°$. The special values of the Denavit and Hartenberg parameters due to the previous special geometry are defined in the following table:

	α	a	d
1	90°	a_1	d_1
2	0°	a_2	0
3	90°	a_3	d_3
4	90°	0	d_4
5	90°	0	d_5
6	α_6	a_6	d_6

The special geometric characteristics of this manipulator don't reduce the degree of its characteristic polynomial and so this manipulator is general i.e. the degree of the characteristic polynomial would be 16. The program chose equation (8) to solve inverse kinematics. This equation will give the characteristic polynomial of the manipulator in $x_3 = \tan(\theta_3/2)$.

The Table I in Appendix 4 shows the Σ_1 matrix for this manipulator in symbolic form. Each entry for this matrix is a linear function in c_3, s_3. The first line of the table shows which component of the vector X_1 corresponds to each column. (Note that in this table, $\lambda_6 = \cos(\alpha_6)$ and $\mu_6 = \sin(\alpha_6)$)

The Table II in Appendix 5 shows the Σ_2 matrix for this manipulator. Each entry of this matrix is a quadratic in $x_3 = \tan(\theta_3/2)$. Normally (in the program) this table is in symbolic form. As the available space for this paper is limited it was not possible to write this matrix in its complete symbolic form. The numerical values for the constant Denavit and Hartenberg parameters were considered as they are specified by the constructor. Thus every element of the Σ_2 matrix depends only on the task dependent terms, described in Appendix 4.

As we explained in section 4 the characteristic polynomial of the manipulator is calculated for the numerical values of the constant Denavit and Hartenberg parameters and for a specific A_h matrix.

Let us suppose that the following A_h matrix is specified:

$$A_h = \begin{pmatrix} 0.92474 & -0.023662 & -0.375612 & 0.772271 \\ -0.079567 & 0.963147 & -0.256934 & 0.122903 \\ 0.367850 & 0.267929 & 0.890449 & 1.079209 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

For this A_h matrix the 16th degree characteristic polynomial in $x_3 = \tan(\theta_3/2)$ is the following:

$$1x_3^{16} + 29.742x_3^{15} + 258.533x_3^{14} + 552.768x_3^{13} - 1194.379x_3^{12} - 6618.041x_3^{11} - 7774.368x_3^{10} + 7491.943x_3^{9} + 30752.031x_3^{8} + 37208.590x_3^{7} + 22719.151x_3^{6} + 6350.533x_3^{5} - 232.829x_3^{4} - 609.108x_3^{3} - 104.471x_3^{2} + 10.086x_3 + 3.005 = 0$$

This polynomial has 8 real roots: -15.794, -9.909, -0.434, -0.276, 0.174, 0.217, 2.156, 2.373. The real roots give the following eight configurations:

θ_1	θ_2	θ_3	θ_4	θ_5	θ_6
5.76°	−38.25°	−172.75°	15.211°	123.85°	−18.77°
19.40°	−37.45°	−168.47°	−171.48°	−127.49°	152.11°
12°	73°	−47.°	86°	10°	70°
18.50°	69.40°	−30.95°	−149.46°	−14.17°	−172.09°
−164.82°	−163.19°	19.84°	9.69°	−117.25°	156.66°
−173.42°	−163.70°	24.59°	−164.21°	115.01°	−13.03°
−164.82°	143.16°	130.24°	9.83°	−61.18°	165.93°
−178.39°	143.58°	134.30°	−163.46°	59.91°	2.21°

7. CONCLUSIONS

In this paper we presented a general algorithm that solves in a unified way the inverse kinematic problem for all types of manipulators general or special. There are 42 types of general manipulators (**6R, 5R1P, 4R2P, 3R3P**) and an undetermined number of special manipulators or of manipulators with special geometry. Each one of these manipulators needs a different mathematical analysis and therefore a different program to solve inverse kinematics. In our algorithm there is a symbolic computation part that does the analysis adapted to the specified manipulator and that calculates the equations needed to specify the characteristic polynomial of the manipulator. The second part of the algorithm is using the numerical data to calculate the characteristic polynomial of the manipulator and all the solutions to the inverse kinematics problem for the kinematic variables of the manipulator. The inverse kinematics of the **GMF Arc Mate** welding manipulator has been presented as an example to illustrate the algorithm.

Acknowledgment

The authors gratefully acknowledge the theoretical help and advice provided by Professor B. Roth of Stanford University. The authors would like also to thank Professor J.C. Guinot of the University of Paris VI, for his support and pleasant arrangements during the development of this work.

Bibliography

1. K.C. Gupta and V.K. Singh, "A Numerical Algorithm for Solving Inverse Kinematics" *Robotica* **7**, part 3, 159–164.
2. D.L. Pieper, *The kinematics of Manipulators under Computer Control* (Ph.D. thesis, Stanford University 1968).
3. J. Angeles, "On the numerical solution of the inverse kinematic problem." *Int. J. Robotics Research* **4**, *No.* 2, 21–37 (1985).
4. K. Kazerounian, "On the Numerical Inverse Kinematics of Robotic Manipulators," *ASME J. Mechanisms, Transmissions, and Automation in Design* **109**/3, 8–13 (1987).
5. L.W. Tsai and A. Morgan, "Solving the Kinematics of the Most General Six-and-Five-Degree-of-Freedom Manipulators by Continuation Methods" *Transactions of ASME, J. Mechanisms, Transmission, and Automation in Design* **107**, 189–200 (1985).
6. H.Y. Lee and C.G. Liang, "Displacement Analysis of the Spatial 7-link 6R-P Linkages" *Mechanism and Machine Theory* **22**, 1–11 (1987).

7. H.Y. Lee and C.G. Liang, "Displacement Analysis of the General Spatial 7-link 7R Mechanism" *Mechanism and Machine Theory* **23**, No. 3, 211–226 (1988).
8. H.Y. Lee and C.G. Liang, "A New Vector Theory for the Analysis of Spatial Mechanisms" *Mechanism and Machine Theory* **23**, No. 3, 209–213 (1988).
9. M. Raghavan and B. Roth, "Kinematic Analysis of the 6R Manipulator of General Geometry" *Proceedings of the 5th International Symposium on Robotics Research* (edited by H.Miura et S. Arimoto) (MIT press, Cambridge (1990) pp. 263–270.
10. M. Raghavan and B. Roth, "A General Solution for the Inverse Kinematics of all Series Chains" *Proceedings of the 8th CISM-IFToMM Symposium on Robots and Manipulators* (Romansy 90), Cracow, Poland (1990) pp. 24–31.
11. M. Raghavan and B. Roth, "Inverse Kinematics of the General 6R Manipulator and Related Linkages" *Proceedings of the ASME, Design Technical Conference, Chicago, Illinois, DE* **25**, 59–65 (1992).
12. D. Manocha and J.F. Canney, "Real Time Inverse Kinematics for General 6R Manipulators" *Proceedings of the 1992 IEEE International Conference on Robotics and Automation, Nice, France* (1992) pp. 383–388.
13. D. Kohlli and M. Osvatic, "Inverse Kinematics of General 6R and 5R,P serial manipulators" *Flexible Mechanisms, Dynamics and Analysis, ASME DE* **47**, 619–629 (1992).
14. D. Kohli and M. Osvatic, "Inverse Kinematics of General 4R2P, 3R3P, 4R1C, 2R2C, and 3C serial manipulators" *Robotics, Spatial Mechanisms, and Mechanical Systems, ASME DE* **45**, 129–137 (1992).
15. D.L. Peiper and B. Roth, "The Kinematics of Manipulators under computer Control" *Proceedings of the 2nd International Congress for the Theory of Machines and Mechanisms, Zakopane, Poland* (1969) pp. 159–160.
16. D.R. Smith, *Design of Solvable 6R Manipulators* (Ph.D. Thesis, Georgia, Institute of Technology, 1990).
17. C. Mavroidis and B. Roth, "Structural Parameters which Reduce the Number of Manipulators Configurations" *Robotics, Spatial Mechanisms, and Mechanical Systems, ASME DE* **45**, 359–366 (1992).
18. W. Khalil and F. Bennis, "Automatic Generation of the Inverse Geometric Model of Robots" *Robotics and Autonomous Systems* **7**, 47–56 (1991).
19. H. Rieseler and F.M. Wahl, "Fast Symbolic Computation of the Inverse Kinematics of Robots" *Proceedings of the 1990 IEEE Conference on Robotics and Automation, San Diego* (1990) pp. 462–467.
20. C. Mavroidis, *Résolution du problème géométrique inverse pour les manipulateurs série à 6 degrés de liberté* (Thèse de Doctorat, Univérsité Pierre et marie Curie, Paris, 1993).

APPENDIX 1. NOTATION OF SPECIAL MANIPULATORS

The notation that represents special geometry in a manipulator is the following:

R,P,C denote revolute, prismatic and cylindric joints respectively.

′, ″ are used to indicate parallel joints. For example **RR′R′RR″R″** is a **6R** manipulator with axes 2 and 3 parallel, and axes 5 and 6 parallel.

⊥ indicates two orthogonal (not necessarily intersecting) axes.

\+ indicates two orthogonal and intersecting axes.

x indicates two intersecting axes.

b as a subscript, indicates that this joint is part of a Bennett group of axes (i.e. three revolute axes whose Denavit and Hartenberg parameters satisfy the Bennett condition).

s as a subscript, indicates that this axis is one of three concurrent revolute axes (i.e. part of an equivalent spherical joint).

R(0) indicates that the offset along this revolute axis is zero.

P(0° or 90°) indicates that the angle θ about this prismatic axis is 0° or 90°.

C(90,0) means that the angle θ is 90° for the prismatic portion of the coaxial **PR** pair that forms the **C** joint, and the offset is 0 for the revolute portion.

APPENDIX 2. GENERAL ALGORITHM THAT SOLVES INVERSE KINEMATICS

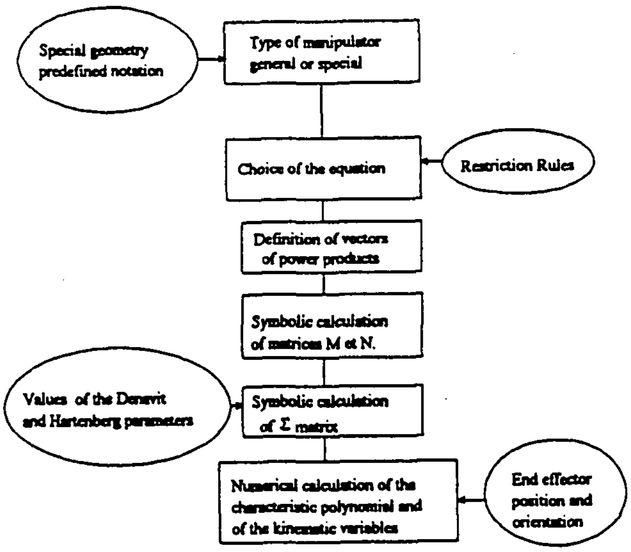

Fig. 1. General algorithm that solves inverse kinematics.

APPENDIX 3. GMF ARC MATE INDUSTRIAL MANIPULATOR

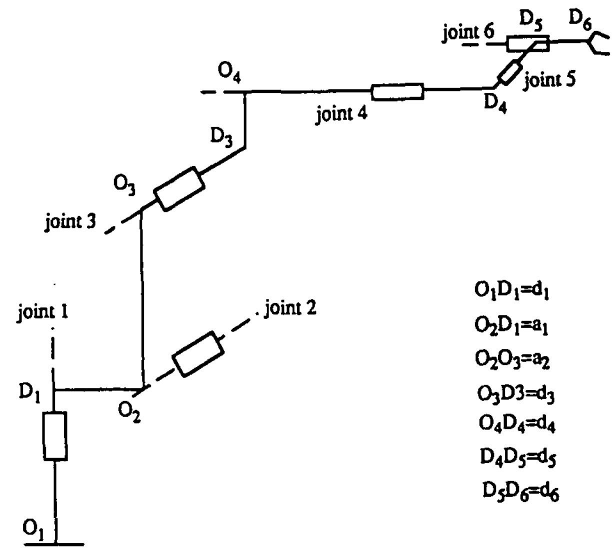

Fig. 2. GMF Arc Mate.

APPENDIX 4. Σ_1 MATRIX FOR THE GMF ARC MATE MANIPULATOR

Table I Σ_1 matrix for the GMF Arc Mate Manupulator.

$s_4 s_5$	$s_4 c_5$	$c_4 s_5$	$c_4 c_5$	s_4	c_4	s_5	c_5	1
$d_3 - t_2 a_1/t_1$	0	$a_3 + a_2 C_3$	0	0	$(-t_4 a_1 d_5)/t_1$	0	$-d_4 - a_2 S_3$	$-(t_1 w r_d + t_1 t_2 - t_4 d_3 a_1)/t_1$
$-(t_2 r_d - t_3 w)/t_1$	d_5	$d_4 + a_2 S_3$	0	0	$-(-t_2 d_5 w + t_4 d_5 r_d)/t_1$	0	$a_3 + a_2 C_3$	$-(t_2 d_3 w - t_4 d_3 r_d - t_1 w a_1)/t_1$
$-2t_3 a_1/t_1$	0	0	0	$2(d_5 a_3 + a_2 d_5 C_3)$	$-(2t_2 a_1 d_5 + 2t_1 d_5 d_3)/t_1$	0	0	K_1
$-2d_3 S_3 d_4 - 2d_3 a_3 C_3 - 2t_2 a_1 a_2/t_1$	$2d_4 d_5 C_3 - 2d_5 a_3 S_3$	K_2	$2d_5 d_3 S_3$	$2(-a_1 w d_5 S_3 + t_2 d_5 C_3 + d_5 C_3 w r_d)$	$-2t_4 a_1 d_5/t_1$	$2d_3 d_5 C_3$	K_3	$-2(w a_1 a_3 S_3 - w a_1 d_4 C_3 - t_2 S_3 d_4 -$ $t_2 a_3 C_3 - w r_d S_3 d_4 - w r_d a_3 C_3 - t_4 a_1 a_2 d_3/t_1)$
$-2d_3 C_3 d_4 + 2d_3 a_3 S_3$ $2(-t_2 a_2 r_d + t_3 a_2 w)t_1$	$-2d_4 d_5 S_3 - 2d_5 a_3 C_3$	K_4	$2d_5 d_2 C_3$	$2(-d_5 S_3 w r_d - t_2 d_5 S_3 - a_1 w d_5 C_3)$	$-2t_4 a_2 d_5 r_d/t_1$ $+ 2t_2 w a_2 d_5/t_1$	$2d_3 d_5 S_3$	K_5	$-2(w a_1 a_3 C_3 w a_1 d_4 S_3 t_2 a_3 S_3 - w r_d C_3 d_4$ $- t_2 C_3 d_4 + w r_d a_3 S_3 + w a_2 d_3 t_2/t_1$ $- t_4 r_d a_2 d_3/t_1)$
K_6	0	$-2d_3(a_5 + a_2 C_3)$	$-2d_5(d_4 + a_2 S_3)$	0	$-2d_5(w r_d + t_2)$	K_8	$2d_3(d_4 + a_2 S_3)$	$2(d_3 w r_d + t_2 d_3 - t_1 a_1)$

where:

$c_i = cos\,(\theta_1),\ s_i = sin\,(\theta_i)$

$p = -l_x a_6 - (m_x \mu_6 + n_x \lambda_6) d_6 + p_x,\ q = -l_y a_6 - (m_y \mu_6 + n_y \lambda_6) d_6 + p_y,\ r = -l_z a_6 - (m_x \mu_6 + n_z \lambda_6) d_6 + p_z$

$r_d = r - d_1,\ u = m_x \mu_6 + n_x \lambda_6,\ p_z = m_y \mu_6 + n_y \lambda_6,\ w = m_z \mu_6 + n_z \lambda_6$

$t_1 = pv - qu,\ t_2 = pu + qv,\ t_3 = p^2 - q^2,\ t_4 = u^2 + v^2$

$K_1 = -(-d_4^2 t_1 - 2t_1 a_2 a_3 C_3 - 2t_2 a_1 d_3 - t_2 pq + t_1 r_d^2 - t_1 d_5^2 - t_1 d_3^2 + 2t_1 q^2 + t_1 p^2 - t_1 a_2^2 - 2t_1 a_2 S_3 d_4 + t_1 a_1^2 - t_1 a_3^2 + t_3 qu)/t_1$

$K_2 = -2a_3 S_3 d_4 - a_3^2 C_3 + d_4^2 C_3 - C_3 a_2^2 + 2a_1 S_3 r_d - C_3 r_d^2 + C_3 a_1^2 + d_5^2 C_3 + d_3^2 C_3 - t_3 C_3$

$K_3 = d_4^2 S_3 + 2d_4 a_3 C_3 - d_5^2 S_3 - d_1^2 S_3 - d_3^2 S_3 + S_3 a_2^2 + 2a_1 C_3 r_d + S_3 r_d^2 - S_3 a_1^2 + t_3 S_3$

$K_4 = -2a_3 C_3 d_4 + a_3^2 S_3 - d_4^2 S_3 - S_3 a_2^2 + 2a_1 C_3 r_d + S_3 r_d^2 - S_3 a_1^2 - d_5^2 S_3 - d_5^2 S_3 - d_3^2 S_3 + t_3 S_3$

$K_5 = d_4^2 C_3 - 2d_4 a_3 S_3 - d_5^2 C_3 - d_5^2 C_3 - d_2^2 C_3 + C_3 a_2^2 - 2a_1 S_3 r_d + C_3 r_d^2 - C_3 a_1^2 + t_3 C_3$

$K_6 = -d_3^2 + 2a_2 S_3 d_4 + d_4^2 + a_3^2 - 2a_2 a_3 C_3 - d_5^2 + a_2^2 - r_d^2 - a_1^2 - t_3$

$K_8 = 2d_5 a_3 + 2a_2 d_5 C_3$

In the previous expressions we considered that the A_h matrix has the following general form: $\begin{pmatrix} l_x & m_x & n_x & p_x \\ l_y & m_y & n_y & p_y \\ l_z & m_z & n_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$. The terms $u,\ v,\ w,\ p,\ q,\ r,\ t_1,\ t_2,\ t_3,\ t_4$ are "task dependent".

APPENDIX 5. Σ_2 MATRIX FOR THE GMF ARC MATE MANIPULATOR

Table II Σ_2 matrix for the GMF Arc Mate Manipulator.

$x_4^2x_5^2$	$x_4^2x_5$	x_4^2	$x_4x_5^2$	x_4x_5	x_4	x_5^2	x_5	1
$K_1x_3^2+12x_3+K_1$	$9.4x_3^2-14.6$	$K_2x_3^2-12x_3+K_2$	0	$(x_3^2+1)*(1.2-8t_2/t_1)$	0	$K_3x_3^2+12x_3+K_3$	$-9.4x_3^2+14.6$	$K_4x_3^2-12x_3+K_4$
$(K_5+4.7)x_3^2+(K_5-7.3)$	$-11x_3^2-24x_3-11$	$(K_4-4.7)*x_3^2+K_5-7.3$	$-2*(x_3^2+1)$	$K_6*(x_3^2+1)$	$2*(x_3^2+1)$	$(K_7+4.7)x_3^2+K_7-7.3$	$11x_3^2+2.4x_3+11$	$(K_7-4.7)x_3^2+K_7+7.3$
$(K_8-15.58)x_3^2+132x_3+K_8+15.62$	0	$(K_8-15.8)*x_3^2+132x_3$ $+(K_8+15.62)$	$-18.8*x_3^2+29.2$	$(-16t_3/t_1)(x_3^2+1)$	$-18.8*x_3^2+29.2$	$(K_9-16.78)x_3^2+132x_3$ $+K_9+14.42$	0	$(K_9-16.78)x_3^2+132x_3+K_9+$ 14.42
$(K_{10}+31.2t_4/t_1)x_3^2+K_{11}x_3$ $+(-K_{10}+31.2t_4/t_1)$	$K_{12}x_3^2+K_{13}x_3$ $-K_{12}$	$(K_{14}+31.2t_4/t_1)x_3^2+K_{15}x_3$ $+(-K_{14}+31.2t_4/t_1)$	$K_{16}x_3^2+K_{17}x_3$ $-K_{16}$	$(3.12-96t_2/t_1)*x_3^2$ $-26.4x_3$ $-3.12-96t_2/t_1$	$K_{18}x_3^2+K_{19}x_3$ $-K_{18}$	$(K_{10}-16.8t_4/t_1)x_3^2$ $+K_{11}x_3$ $-(-K_{10}-16.8t_4/t_1)$	$(-K_{12}+2.4)x_3^2$ $-K_{13}x_3$ $+(K_{12}-2.4)$	$(K_{14}-16.8t_4/t_1)*x_3^2+K_{15}x_3$ $+-K_{14}-16.8t_4/t_1$
$(K_{20}+15.6(t_4r-t_2w)/$ $t_1-126.34t_4)*x_3^2$ $+K_{21}x_3-K_{15}+15.6(rt_4-t_2w)/t_1$	$K_{22}x_3^2+K_{23}x_3$ $-K_{22}$	$K_{24}x_3^2+K_{25}x_3+K_{26}$	$K_{27}x_3^2+K_{28}x_3$ $-K_{27}$	$(K_{29}+13.2)*x_3^2$ $+6.24x_3$ $+(K_{29}-13.2)$	$K_{30}x_3^2+K_{31}x_3$ $-K_{30}$	$K_{32}x_3^2+K_{33}x_3+K_{34}$	$K_{35}x_3^2+K_{36}x_3-K_{35}$	$K_{37}x_3^2+K_{38}x_3+K_{39}$
$(K_{40}-14.3)x_3^2-31.2x_3$ $+(K_{40}-14.3)$	$-24.44x_3^2+37.96$	$(K_{40}+14.3)x_3^2-31.2x_3$ $+(K_{40}+14.3)$	0	$(K_{41}-65.44)*x_3^2$ $+528x_3$ $K_{41}+59.36$	0	$(K_{42}+7.7)x_3^2-16.8x_3$ $+(K_{42}+7.7)$	$-13.16x_3^2+20.44$	$(K_{42}-7.7)x_3^2-16.8x_3$ $+(K_{42}-7.7)$

$K_1=2.64t_4/t_1+5.5-wr-t_2+8.1w$; $K_2=2.6t_4/t_1-5.5-wr-t_2+8.1w$; $K_3=-1.4t_4/t_1+5.5-wr-t_2+8.1w$;
$K_4=-1.4t_4/t_1-5.5-wr-t_2+8.1w$; $K_5=-1.3t_2w/t_1+1.3t_4r/t_1+2w-10.53t_4$; $K_6=4(-t_2r+8.1w+t_3w)/t_1$;
$K_7=0.7t_2w/t_1-0.7t_4r/t_1+2w+5.67t_4$; $K_8=-r^2-po^2-2qo^2+5.2t_2/t_1+t_2pq/t_1-t_3qu/t_1+16.2r$;
$K_9=-r^2-po^2-2qo^2-2.8t_2/t_1+t_2pq/t_1-t_3qu/t_1+16.2r$;
$K_{10}=2(-1.3wr-9.05+2r-1.3t_2-0.47w)$; $K_{11}=2(-94.3w-52.48-ro^2-t_3+11t_2+11wr+16.2r)$;
$K_{12}=2(4.64-t_3-r^2+16.2r)$; $K_{13}=2(-8r+93.4)$; $K_{14}=-2(1.3wr-9.05+2r+1.3t_2+0.47w)$;
$K_{15}=-2(94.3w-52.48-ro^2-t_3-11t_2-11wr+16.2r)$; $K_{16}=-4(-5.5+wr+t_2-8.1w)$;
$K_{17}=-4(4w-2.6)$; $K_{18}=-4(5.5+wr+t_2-8.1w)$; $K_{19}=-4(4w+2.6)$;
$K_{20}=-(11wr-t_3-r^2-124.48+11t_2-94.3w+16.2r)$; $K_{21}=-(5.2t_2-36.2+5.2rw+1.88w-8r)$;
$K_{22}=4(-23.35+2r)$; $K_{23}=4(-67.36-r^2-t_3+16.2r)$;
$K_{24}=(-124.48-t_3-r^2-15.6t_2w/t_1+15.6t_4r/t_1-11t_2-11wr+94.3w-126.36t_4+16.2r)$;
$K_{25}=(-5.2t_2-5.2wr-1.88w-8r+36.2)$; $K_{26}=(r^2+124.48-15.6t_2w/t_1+15.6t_4r/t_1+11t_2+t_3+11wr-94.3w-126.36t_4-16.2r$;
$K_{27}=4(2w-1.3)$; $K_{28}=4(-2wr+11-2t_2-16.2w)$;
$K_{29}=8(6t_3w/t_1-6t_2r/t_1+48.6t_2/t_1)$; $K_{30}=4(2w+1.3)$; $K_{31}=4(-2wr-11-2t_2+16.2u)$;
$K_{32}=-(-125.68-t_3-r^2-8.4t_2w/t_1+8.4t_4r/t_1+11t_2+11wr-94.3w-68.04t_4/t_1+16.2r)$;
$K_{33}=-(5.2t_2+5.2wr+1.88w-8r+36.2)$;
$K_{34}=-(125.68+t_3+r^2-8.4t_2w/t_1+8.4t_4r/t_1-11t_2-11wr+94.3w-68.04t_4/t_1-16.2r)$;
$K_{35}=-4(-23.35+2r)$; $K_{36}=-4(-68.56-r^2-t_3+16.2)$;
$K_{37}=(-125.68-t_3-r^2+8.4t_2w/t_1-8.4t_4r/t_1-11t_2-11wr+94.3w+68.04t_4/t_1+16,2r)$;
$K_{38}=(-5.2t_2-5.2wr-1.88w-8r+36.2)$;
$K_{39}=(125.68+t_3+r_2+8.4t_2w/t_1-8.4t_4r/t_1+11t_2+11wr-94.3w+68.04t_4/t_1-16.2r)$;
$K_{40}=-2(-1.3wr-1.3t_2+2t_1+10.53w)$; $K_{41}=-4(r^2+t_3-16.2r)$; $K_{42}=-2(0.7wr+0.7t_2+2t_1+2t-5.67w)$;