9

Certified Solving and Synthesis on Modeling of the Kinematics. Problems of Gough-Type Parallel Manipulators with an Exact Algebraic Method

Luc Rolland
University of Central Lancashire
*United Kingdom**

1. Introduction

The significant advantages of parallel robots over serial manipulators are now well known. However, they still pose a serious challenge when considering their kinematics. This paper covers the state-of-the-art on modeling issues and certified solving of kinematics problems. Parallel manipulator architectures can be divided into two categories: planar and spatial. Firstly, the typical planar parallel manipulator contains three kinematics chains lying on one plane where the resulting end-effector displacements are restricted. The majority of these mechanisms fall into the category of the **3-RPR** generic planar manipulator, [Gosselin 1994, Rolland 2006]. Secondly, the typical spatial parallel manipulator is an hexapod constituted by six kinematics chains and a sensor number corresponding to the actuator number, namely the **6-6** general manipulator, fig. 1.

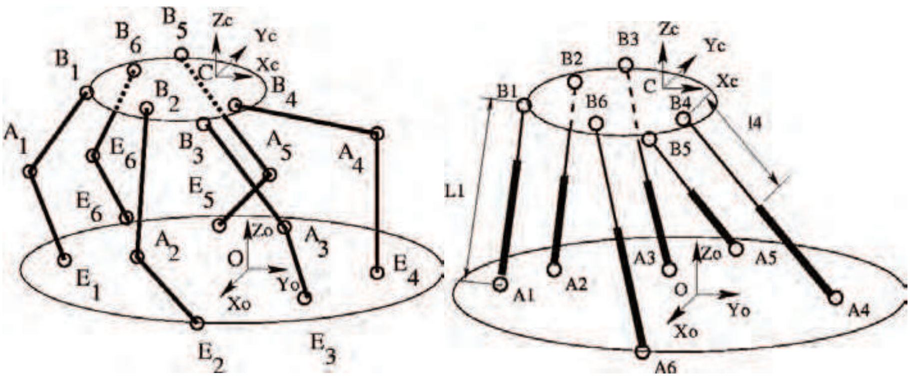

Fig. 1. The general **6-6** hexapod manipulators

Solving the **FKP** of general parallel manipulators was identified as finding the real roots of a system of non-linear equations with a finite number of complex roots. For the **3-RPR**, 8 assembly modes were first counted, [Primerose and Freudenstein 1969]. Hunt geometrically demonstrated that the **3-RPR** could yield 6 assembly modes, [Hunt 1983]. The numeric

Source: Parallel Manipulators, Towards New Applications, Book edited by: Huapeng Wu, ISBN 978-3-902613-40-0, pp. 506, April 2008, I-Tech Education and Publishing, Vienna, Austria

iteration methods such as the very popular Newton one were first implemented, [Dieudonne 1972, Merlet 1987, Sugimoto 1987]. They only converge on one real root and the method can even fail to compute it. To compute all the solutions, polynomial equations were justified, [Gosselin and Angeles 1988]. Ronga, Lazard and Mourrain have established that the general **6-6** hexapod **FKP** has 40 complex solutions using respectively Gröbner bases, Chern classes of vector bundles and explicit elimination techniques, [Ronga and Vust 1992, Lazard 1993, Mourrain 1993a]. The continuation method was then applied to find the solutions, [Raghavan 1993], however, it will be explained why they are prone to miss some solutions, [Rolland 2003]. Computer algebra was then selected in order to manipulate exact intermediate results and solve the issue of numeric instabilities related to round-off errors so common with purely numerical methods. Using variable elimination, for the **3-RPR**, 6 complex solutions were calculated [Gosselin 1994] and, for the **6-6**, Husty and Wampler applied resultants to solve the **FKP** with success, [Husty 1996, Wampler 96]. However, resultant or dialytic elimination can add spurious solutions, [Rolland 2003] and it will be demonstrated how these can be hidden in the polynomial leading coefficients. Inasmuch, a sole univariate polynomial cannot be proven equivalent to a complete system of several polynomials. Intervals analyses were also implemented with the Newton method to certify results, [Didrit et al. 1998, Merlet 2004]. However, these methods are often plagued by the usual Jacobian inversion problems and thus cannot guarantee to find solutions in all non-singular instances. The geometric iterative method has shown promises, [Petuya et al. 2005], but, as for any other iterative methods, it needs a proper initial guess.
Hence, this justified the implementation of an exact method based on proven variable elimination leading to an equivalent system preserving original system properties. The proposed method uses Gröbner bases and the rational univariate representation, [Faugère 1999, Rouillier 1999, Rouillier and Zimmermann 2001], implementing specific techniques in the specific context of the **FKP**, [Rolland 2005]. Three journal articles have been covering this question for the general planar and spatial manipulators [Rolland 2005, Rolland 2006, Rolland 2007]. This algebraic method will be fully detailed in this chapter.
This document is divided into 3 main topics distributed into five sections. The first part describes the kinematics fundamentals and definitions upon which the exact models are built. The second section details the two models for the inverse kinematics problem, addresses the issue of the kinematics modeling aimed at its adequate algebraic resolution. The third section describes the ten formulations for the forward kinematics problem. They are classified into two families: the displacement based models and position based ones. The fourth section gives a brief description of the theoretical information about the selected exact algebraic method. The method implements proven variable elimination and the algorithms compute two important mathematical objects which shall be described: a Gröbner Basis and the Rational Univariate Representation including a univariate equation. In the fifth section, one **FKP** typical example shall be solved implementing the ten identified kinematics models. Comparing the results, three kinematics models shall be retained. The selected manipulator is a generic **6-6** in a realistic configuration, measured on a real parallel robot prototype constructed from a theoretically singularity-free design. Further computation trials shall be performed on the effective **6-6** and theoretical one to improve response times and result files sizes. Consequently, the effective configuration does not feature the geometric properties specified on the theoretical design. Hence, the **FKP** of theoretical designs shall be studied and their kinematics results compared and analyzed. Moreover, the posture analysis or assembly mode issue shall be covered.

2. Kinematics of parallel manipulator

2.1 Kinematics notations and hypotheses

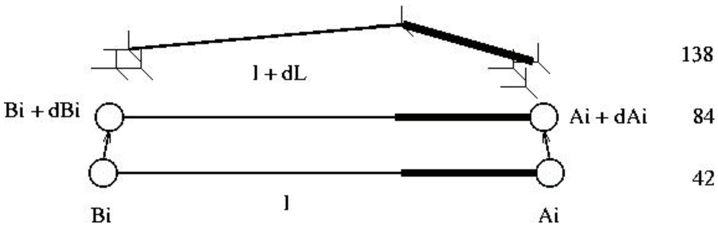

Fig. 2. Typical kinematics chains

The parallel Gough platform, namely **6-6**, is constituted by six kinematics chains, fig. 2. It is characterized by its mechanical configuration parameters and the joint variables. The configuration parameters are thus $\mathbf{OA}_{Rf}$ as the base geometry and $\mathbf{CB}_{Rm}$ as the mobile platform geometry. The joint variables are described as ρ the joint actuator positions (angular or linear). Lets assume rigid kinematics chains, a rigid mobile platform, a rigid base and frictionless ball joints between platforms and kinematics chains.

2.2 Hexapod exact modeling

Stringent applications such as milling or surgery require kinematics models as close as possible to exactness. Realistically, any effective configuration always comprises small but significant manufacturing errors, [Vischer 1996, Patel & Ehmann 1997]. Hence, any constructed parallel manipulator never corresponds to the theoretical one where specific geometric properties may have been chosen, for example, to alleviate singularities or to simplify kinematics solving. Two prismatic actuator axes may be neither collinear nor parallel and may not even intersect. Whilst knowing joints prone to many imperfections, then rotation axes are not intersecting and the angles between them are never perpendicular. Moreover, real ball joints differ from a perfectly circular shape and friction induces unforeseeable joint shape modification, which results into unknown axis changes. However, the joint axis angles stay almost perpendicular and any rotation combination shall be feasible. In a similar fashion, the Cardan joint axes are not perpendicular and may be separated by a small offset. Finally, the articulation center is not crossed by any axis.

Identified the **hexapode 138**, the exact geometric model is then characterized by 138 configuration parameters. Each kinematics chain is described by 23 parameters, as shown on fig. 2 and defined hereafter:

- the 3 parameters of each base joint A_i with their error vector δA_i ,
- the 3 joint Ai inter-axis distances ϵ_1^a , ϵ_2^a and ϵ_3^a
- each prismatic joint measured position l_i with its error coordinate δL_i ,
- the 3 parameters of the minimum distance between the two prismatic actuator axes: $\vec{d}_r$,
- the angular deviation between the two prismatic actuator axes: φ,
- the 3 parameters of the platform joint B_i with their error vector δB_i,
- the 3 joint B_i inter-axis distances and ϵ_1^b, ϵ_2^b and ϵ_3^b

To solve this model includes the determination of parameters which cannot be measured neither determined. Moreover, the model includes more variables than equations and therefore, its resolution would then only be possible through optimization methods. Relying on a calibration procedure would only determine configuration parameters by specifying an error margin consisting of a radius around joint positions and would not indicate the direction of the error vector. Hence, only an error ball becomes applicable to the model. In practice, the δA_i and δB_i joint error vectors shall reposition the respective kinematics chains by adding an offset to the joint centers. Thus, a random function shall compute the δA_i and δB_i vectors with the maximum being the error ball radius. Finally, the selected model, namely the *hexapod 84*, is effectively based on the *hexapod 42* model with errors added to the configuration data and joint variables.

2.3 Kinematics problems

Definition 2.1 *The kinematics model is an implicit relation between the configuration parameters and the posture variables,* $F(\vec{X}, \Gamma, \mathbf{OA}_{|Rf}, \mathbf{CB}_{|Rm})=0$ *where* $\Gamma = \{\rho_1, \rho_2, \ldots, \rho_6\}$.

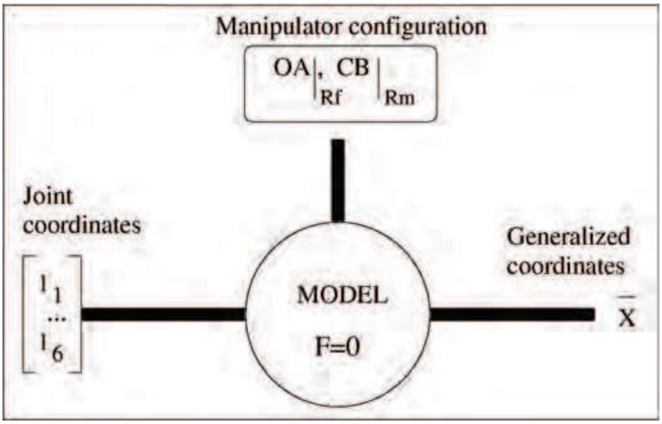

Fig. 3. Kinematics model

Three problems can be derived from the above relation: the forward kinematics problem (**FKP**), the inverse kinematics problem (**IKP**) and the kinematics calibration problem, fig. 3. The two first problems shall be covered in this article. The inverse kinematics problem (**IKP**) is defined as:

Definition 2.2 *Given the generalized coordinates of the manipulator end-effector, find the joint positions.*

The **6-6 IKP** yields explicit solutions from vector $\Gamma = G(\vec{X}, \mathbf{OA}_{|Rf}, \mathbf{CB}_{|Rm})$ and is used to prepare the **FKP** which is defined as:

Definition 2.3 *Given the joint positions* Γ*, find the generalized coordinates* $\vec{X}$ *of the manipulator end-effector.*

The **6-6 FKP** is a difficult problem, [Merlet 1994, Raghavan and Roth 1995] and explicit solutions $\vec{X} = G(\Gamma, \mathbf{OA}_{|Rf}, \mathbf{CB}_{|Rm})$ have not yet been established. The difficulties in solving the **FKP** have hampered the application of parallel robot in the milling industry.

2.4 Vectorial formulation of the basic kinematics model

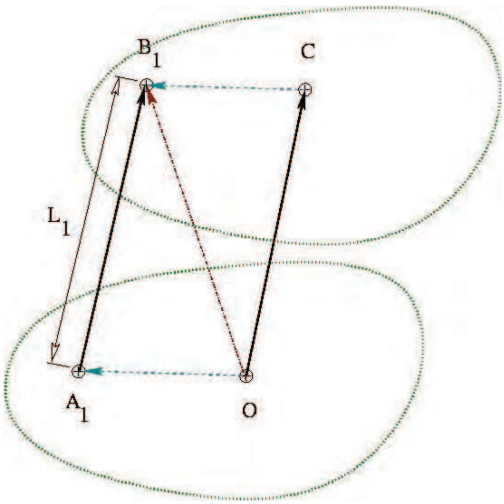

Fig 4. The vectorial formulation

The vectorial formulation produces an equation system which contains the same number of equations as the number of variables, fig. (4), [Dieudonne et al. 1972]. A closed vector cycle is constituted between the manipulator characteristic points: A_i and B_i, kinematics chain attachment points, O the fixed base reference frame and C the mobile platform reference frame. For each kinematics chain, a function between points A_i and B_i expresses the generalized coordinates X, such as $\overrightarrow{A_iB_i} = U_1(X)$. Inasmuch, vector $\overrightarrow{A_iB_i}$ is determined with the joint coordinates Γ and X giving a function $U_2(X, \Gamma)$. Finally, the following equality has to be solved: $U_1(X) = U_2(X, \Gamma)$.

3. The inverse kinematics problem

For each kinematics chain, $i = 1, ..., 6$, each platform point $\overrightarrow{OB}_{i|Rf}$ can be expressed in terms of the distance constraint, [Merlet 1997]:

$$l_i^2 = \|A_iB_i\|^2, i = 1 \ldots 6 \tag{1}$$

Using the vectorial formulation, two equation families can be derived: displacement-based and position-based equations.

3.1 Displacement based equations

Any mobile platform position $\overrightarrow{\mathrm{OB}}_{|Rf}$ which meets constraints 1 has a rotation matrix $\Re$ such that:

$$\overrightarrow{OB}_{i|R_f} = \overrightarrow{OC}_{i|R_f} + \Re \cdot \overrightarrow{CB}_{i|R_m}, i = 1 \ldots 6 \tag{2}$$

Substituting 2 in 1, we obtain:

$$l_i^2 = \left\| \overrightarrow{OC}_{|R_f} + \Re \cdot \overrightarrow{CB}_{i|R_m} - \overrightarrow{OA}_{i|R_f} \right\|^2, i = 1 \ldots 6 \tag{3}$$

This last equation system can be developed and simplified, leading to the **IKP** :

$$l_i^2 = \left(\overrightarrow{OC}_{|R_f} - \overrightarrow{OA}_{i|R_f} \right)^2 + \left(\overrightarrow{OC}_{|R_f} - \overrightarrow{OA}_{i|R_f} \right) \Re \cdot \overrightarrow{CB}_{i|R_m} + \overrightarrow{CB}_i^2 \tag{4}$$

3.2 Position based equations

In 3D space, any rigid body can be positioned by 3 of its distinct non-colinear points, [Fischer and Daniel 1992, Lazard 1992b]. The 3 mobile platform distinct points are usually selected as the 3 joint centers B_1, B_2, B_3, fig. 5. The 6 variables are set as: $\overrightarrow{OB}_{i|Rf} = [x_i, y_i, z_i]$ for $i = 1 \ldots 3$. The $\overrightarrow{OB}_{i|Rf}$ parameters define the reference frame R_{b1} relative to the mobile platform and B_1 is chosen as its center. The frame axes u_1, u_2 and u_3 are determined by the 3 platform points:

$$u_1 = \frac{\overrightarrow{B_1B_2}}{\left\| \overrightarrow{B_1B_2} \right\|}, u_2 = \frac{\overrightarrow{B_1B_3}}{\left\| \overrightarrow{B_1B_3} \right\|}, u_3 = u_1 \wedge u_2 \tag{5}$$

Any platform point M can be expressed by $\overrightarrow{B_1M} = a_M u_1 + b_M u_2 + c_M u_3$ where a_M, b_M, c_M are constants in terms of these three points. Hence, in the case of the **IKP**, the constants are noted a_{Bi}, b_{Bi}, c_{Bi}, i = i . . . 6 and can explicitly be deduced from $\mathbf{CB}_{|Rm}$ by solving the following linear system of equations:

$$\overrightarrow{B_1B}_{i|R_{b_1}} = a_{B_i} u_1 + b_{B_i} u_2 + c_{B_i} u_3, i = 1 \ldots 6 \tag{6}$$

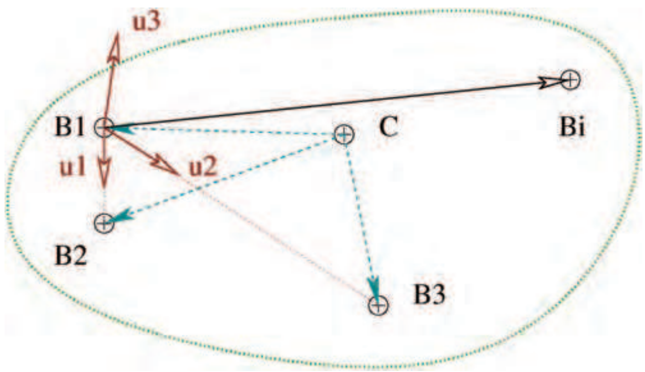

Fig. 5. The platform three point coordinate system

Substituting relations 6 in the distance equations $l_i^2 = \| \overrightarrow{A_iB_i}_{|Rf} \|$, $i = 1 \dots 6$, the system can be expressed with respect to the variables x_i, y_i, z_i , $i = 1, 2, 3$. Thus, for i = 1 . . . 6, the **IKP** is obtained by isolating the ρ_i or l_i linear actuator variables in the six following equations:

$$l_i^2 = \left(x_i - \overrightarrow{OA}_{ix}\right)^2 + \left(y_i - \overrightarrow{OA}_{iy}\right)^2, i = 1...3 \quad (7)$$

$$l_i^2 = \left\| \vec{B}_{k|R_{b_1}} - \overrightarrow{OA}_{k|R_f} \right\|^2, i = 4 \dots 6 \quad (8)$$

4. The forward kinematics problem

4.1 Displacement based equations

There exist various formulations of the displacement based equation models.

4.1.1 AFD1 - formulation with the position and the trigonometry identity

The AFD1 formulation is obtained by replacing each trigonometric function of the **IKP** rotation matrix, 2, by one distinct variable, [Merlet 1987], for $j = 1, 2, 3$, then $c_j = cos(\Theta j)$, $s_j = sin(\Theta_j)$. The end-effector position variables are retained. The 9 unknowns are then: $\{x_c, y_c, z_c, c_1, c_2, c_3, s_1, s_2, s_3\}$. The orientation variables can either be any Euler angles or the navigation ones (pitch, yaw and roll). The orientation variables are linked by the 3 trigonometric identities, for $j = 1 \dots 3$, then $c^2_j + s^2_j = 1$ which complete the equation system:

$$F_i = \left(\overrightarrow{OC}_{|R_f} - \overrightarrow{OA}_{i|R_f}\right)^2 + \left(\overrightarrow{OC}_{|R_f} - \overrightarrow{OA}_{i|R_f}\right)\Re \cdot \overrightarrow{CB}_{i|R_m} - L_i^2, i = 1, \dots, 6 \quad (9)$$

$$F_j = c_j^2 + s_j^2 - 1, j = 1, 2, 3 \quad (10)$$

The system is constituted of 9 equations with 6 polynomials of degree 6 and 3 quadratics. The model is simply build by variable substitution without any computation. Thus, the coefficients remain unchanged. The number of variables is not minimal.

4.1.2 AFD2 - formulation with the position and the trigonometric function change

The end-effector position variables are retained. Rotation variable changes can apply the following trigonometric relations, [Griffis & Duffy 1989, Parenti-Castelli & C. Innocenti 1990, Lazard 1993]. For $i = 1, 2, 3$:

$$t_i = \tan\left(\frac{\theta_i}{2}\right), \sin\left(\theta_i\right) = \frac{2t_i}{\left(1 + t_i^2\right)}, \cos\left(\theta_i\right) = \frac{\left(1 - t_i^2\right)}{\left(1 + t_i^2\right)} \quad (11)$$

The 6 variables become $\{x_c, y_c, z_c, t_1, t_2, t_3\}$. The **IKP** equations (2) are rewritten to obtain the 6 following equations:

$$F_i = \left(\overrightarrow{OC}_{|R_f} - \overrightarrow{OA}_{i|R_f}\right)^2 + \left(\overrightarrow{OC}_{|R_f} - \overrightarrow{OA}_{i|R_f}\right)\Re \cdot \overrightarrow{CB}_{i|R_m} - L_i^2, i = 1, \dots, 6 \quad (12)$$

The final equation system comprises 6 equations of order 8 with the high degree monomial being $x_i^2 x_j^2 x_k^2 x_n^2$. This model has a minimal variable number. The polynomials coefficients are expanding due to variable change computation. Moreover, this representation is not intuitive.

4.1.3 AFD3 - formulation with the translation and rotation matrix

The intuitive way to set an algebraic equation system from the **IKP** equations 2 is to straightforwardly use all the rotation matrix parameters and the vector $\overrightarrow{OC}_{|Rf}$ coordinates as unknowns, [Lazard 1993, Sreenivasan et al. 1994, Bruyninckx and DeSchutter 1996]. The variables are then $\{X_c, Y_c, Z_c, r_{ij}, j=1...3, i=1...3\}$. Since $\mathfrak{R}$ is a rotation matrix, the following relations hold: $\mathfrak{R}^t\mathfrak{R} = Id$ or $det(\mathfrak{R}) = 1$. These relations are redundant since $\mathfrak{R}^t\mathfrak{R}$ is symmetrical and they generate the 7 following equations:

$$\begin{cases} 1 = r_{11}^2 + r_{12}^2 + r_{13}^2, 1 = r_{21}^2 + r_{22}^2 + r_{23}^2, 1 = r_{31}^2 + r_{32}^2 + r_{33}^2 \\ 0 = r_{11}r_{21} + r_{12}r_{22} + r_{13}r_{23}, 0 = r_{11}r_{31} + r_{12}r_{32} + r_{13}r_{33}, 0 = r_{21}r_{31} + r_{22}r_{32} + r_{23}r_{33} \\ 1 = r_{11}r_{22}r_{33} - r_{11}r_{23}r_{32} - r_{21}r_{12}r_{33} + r_{21}r_{13}r_{32} + r_{31}r_{12}r_{23} - r_{31}r_{13}r_{22} \end{cases} \quad (13)$$

Six rotation matrix constraints are then selected and preferably with the lowest degree polynomials. This leads to an algebraic system with 12 polynomial equations (13 and 1) in 12 unknowns.

$$F_i = \left(\overrightarrow{OC}_{|R_f} - \overrightarrow{OA}_{i|R_f}\right)^2 + \left(\overrightarrow{OC}_{|R_f} - \overrightarrow{OA}_{i|R_f}\right)\mathfrak{R}\cdot\overrightarrow{CB}_{i|R_m} - L_i^2, i = 1,\ldots,6 \quad (14)$$

$$F_7 = r_{11}^2 + r_{12}^2 + r_{13}^2 - 1 \quad (15)$$

$$F_8 = r_{21}^2 + r_{22}^2 + r_{23}^2 - 1 \quad (16)$$

$$F_9 = r_{31}^2 + r_{32}^2 + r_{33}^2 - 1 \quad (17)$$

$$F_{10} = r_{11}r_{21} + r_{12}r_{22} + r_{13}r_{23} \quad (18)$$

$$F_{11} = r_{11}r_{31} + r_{12}r_{32} + r_{13}r_{33} \quad (19)$$

$$F_{12} = r_{21}r_{31} + r_{22}r_{32} + r_{23}r_{33} \quad (20)$$

Finally, the model polynomials are quadratic and minimal. They are obtained by substitution and no computations are required. The coefficients are then unchanged. There is a very large number of variables.

4.1.4 AFD4 - formulation with the translation and *Gröbner Basis* on the rotation matrix

The rotation matrix constraints are not depending on the end-effector position variables. Hence, if one *Gröbner Basis* is computed from the rotation constraints, the *Gröbner Basis* is also independent of the position variables and thus constant for any **FKP** pose. Therefore, one preliminary *Gröbner Basis* can be calculated and saved into a file for later reuse.
Hence, the rotation matrix constraints in the system 20 can be replaced by their *Gröbner Basis* comprising 24 equations where the coefficients are only unity. Thus, the algebraic system involves 30 equations and 12 variables.

4.1.5 AFD5 - translation and quaternion algebraic model

Based on equation (2), quaternions can express mobile platform rotation, [Lazard 1993, Mourrain 1993b, Egner 1996, Murray et al. 1997]. The quaternion representation includes 4 variables $\{q_0; q_1; q_2; q_3\}$ where the vector $\overline{q} = q_1\,\mathbf{i} + q_2\,\mathbf{j} + q_3\,\mathbf{k}$ defines the platform specific rotation axis and $q_0 = cos(\alpha/2)$ determines the coordinate expressing the rotation α along that axis. Thus, the rotation matrix $\Re$ used in equations 4 may then be expressed in terms of the quaternion coordinates and with $\Delta^2 = q_0^2 + q_1^2 + q_2^2 + q_3^2$, we can write:

$$\Re = \Delta^{-2}\begin{pmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2-q_0q_3) & 2(q_1q_3+q_0q_2) \\ 2(q_1q_2+q_0q_3) & q_0^2-q_1^2+q_2^2-q_3^2 & 2(q_2q_3-q_0q_1) \\ 2(q_1q_3-q_0q_2) & 2(q_2q_3+q_0q_1) & q_0^2-q_1^2-q_2^2+q_3^2 \end{pmatrix} \tag{21}$$

The end-effector position variables are retained. Moreover, one may implement a unitary quaternion: $\Delta^2 = 1$. Rewriting the **IKP** equations 4, we obtain 7 polynomial equations in the 7 unknowns $\{X_c; Y_c; Z_c; q_0; q_1; q_2; q_3\}$:

$$F_i = \left(\overrightarrow{OC}_{|R_f} - \overrightarrow{OA}_{i|R_f}\right)^2 + \left(\overrightarrow{OC}_{|R_f} - \overrightarrow{OA}_{i|R_f}\right)\Re\cdot\overrightarrow{CB}_{i|R_m} - L_i^2, i = 1,\ldots,6 \tag{22}$$

$$F_7 = q_0^2 + q_1^2 + q_2^2 + q_3^2 - 1 \tag{23}$$

The system contains 6 polynomials of degree 6 and 1 quadratic. The highest degree monomial is $x_i^2\ x_j^2$. The quaternion has intrinsic coordinate redundancy which allows avoiding typical mathematical singularities seen in other representations. The number of variable is almost minimal. The rotation matrix system must be recomputed leading to larger resulting polynomial coefficients.

4.1.6 AFD6 - translation and dual quaternion algebraic model

Not only orientations can be formulated using quaternions, but also positions, [Husty 1996, Wampler 96]. The $\Re$ rotation matrix is then expressed in terms of the first *quaternion* $\Phi = \{c_0; c_1; c_2; c_3\}$. In a sense, the second $\Psi = \{g_1, g_2, g_3, g_4\}$ represents the end-effector position. Moreover, one relation can be written between the two quaternions: $\Phi = \overline{OC}\ \Psi$. This relation unfolds in the following equations from which two constraint equations, noted $FC_1 = 0$ and $FC_2 = 0$, are selected. Lets $s_i = \mathbf{OA}_{|Rf}$ and $t_i = \mathbf{CB}_{|Rm}$, then:

$$\begin{aligned} &\overline{c}^t\overline{c} = 1 \\ &\overline{g}^t\overline{c} = 0 \\ &\overline{g}^t\overline{g} - l_1\overline{c}^t\overline{c} = 0 \\ &\overline{c}^t s_i\overline{c} + 2\overline{g}^t t_i\overline{c} = 0 \end{aligned} \qquad \text{For } i = 2,\ldots,6 \tag{24}$$

The dual quaternion system is thus constituted by the 8 following equations, for $i = 1 \ldots 6$:

$$F_i = \left(\overrightarrow{OC}_{|R_f} - \overrightarrow{OA}_{i|R_f}\right)^2 + \left(\overrightarrow{OC}_{|R_f} - \overrightarrow{OA}_{i|R_f}\right)\Re\cdot\overrightarrow{CB}_{i|R_m} - L_i^2 \tag{25}$$

$$F_7 = FC_1 \tag{26}$$

$$F_8 = FC_2 \tag{27}$$

The system comprises 6 polynomials of degree 4 and 2 quadratics. The highest degree monomials are either x_i^4; $x_i^3\ x_j$ or $x_i^2\ x_j^2$. One more variable is added over the former quaternion model. The variable choice is not intuitive.

4.2 Position based equations

We shall examine four formulations derived from the position based equations. Every variable has the same units and their range is equivalent.

4.2.1 AFP1 - three point model with platform dimensional constraints

The 3 platform distinct points are usually selected as the three joint centers B_1, B_2 and B_3, fig. 5. The 6 variables are set as: $\overrightarrow{OB}_{i\,|\,Rf} = [x_i, y_i, z_i]$ for $i = 1 \ldots 3$.

Using the relations 6, the constraint equations $L_i^2 = \|\overrightarrow{A_iB_i}_{\,|Rf}\|^2$, $i = 1, \ldots, 6$ can be expressed with respect to the variables x_i, y_i, z_i, $i = 1, 2, 3$. Together with equations 30, they define an algebraic system with 9 equations in 9 unknowns $\{x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3\}$. The resulting kinematics chain system becomes:

$$F_i = (x_i - OA_{ix})^2 + (y_i - OA_{iy})^2 + (x_i - OA_{ix})^2 - L_i^2, i = 1 \ldots 3 \tag{28}$$

$$F_j = \left\| \vec{B}_{j|R_{b_1}} - \overrightarrow{OA}_{j|R_f} \right\|^2 - L_j^2, j = 4 \ldots 6 \tag{29}$$

The mobile platform geometry yields the following three distance equations:

$$\begin{aligned}
F_7 &= \left\| \overrightarrow{B_2B}_{1|R_f} \right\|^2 - (x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2 = \left\| \overrightarrow{B_2B}_{1|R_m} \right\|^2 \\
F_8 &= \left\| \overrightarrow{B_3B}_{1|R_f} \right\|^2 - (x_3 - x_1)^2 + (y_3 - y_1)^2 + (z_3 - z_1)^2 = \left\| \overrightarrow{B_3B}_{1|R_m} \right\|^2 \\
F_9 &= \left\| \overrightarrow{B_3B}_{2|R_f} \right\|^2 - (x_3 - x_2)^2 + (y_3 - y_2)^2 + (z_3 - z_2)^2 = \left\| \overrightarrow{B_3B}_{2|R_m} \right\|^2
\end{aligned} \tag{30}$$

Together with equations 30, they produce an algebraic system with 9 equations with 9 unknowns $\{x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3\}$. In all instances, it can be easily proven that this **6-6 FKP** formulation yields 9 quadratic polynomials.

The system variable choice is relatively intuitive. Each equation polynomial is always quadratic. However, the b_1 reference frame and the platform points B_i in the b_1 frame require computations, which usually result into coefficient size explosion. The variable number is not minimal.

4.2.2 AFP2 - the three point model with platform constraints

The former system can be slightly modified by replacing the last mobile platform constraint with a platform normal vector one. Hence, lets take the two mobile platform vectors $\overrightarrow{B_1B_2}$ and $\overrightarrow{B_1B_3}$, then the last constraint is calculated from these two vector multiplication:

$$\begin{aligned}
F_7 &= \left\|\overrightarrow{B_2B}_{1|R_f}\right\|^2 - (x_2-x_1)^2 + (y_2-y_1)^2 + (z_2-z_1)^2 = \left\|\overrightarrow{B_2B}_{1|R_m}\right\|^2 \\
F_8 &= \left\|\overrightarrow{B_3B}_{1|R_f}\right\|^2 - (x_3-x_1)^2 + (y_3-y_1)^2 + (z_3-z_1)^2 = \left\|\overrightarrow{B_3B}_{1|R_m}\right\|^2 \\
F_9 &= (x_3-x_1)*(x_2-x_1)+(y_3-y_1)*(y_2-y_1)+(z_3-z_1)*(z_2-z_1)-\left\|\overrightarrow{B_3B}_{2|R_m}\right\| \wedge \left\|\overrightarrow{B_3B}_{1|R_m}\right\|
\end{aligned} \tag{31}$$

The result is still an algebraic system with nine equations in the former nine unknowns $\{x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3\}$. The **6-6 FKP** formulation using this three point model is constituted by nine quadratic polynomials.

4.2.3 AFP3 - the three point model with constraints and function recombination

By rewriting the **IKP** as functions, the algebraic system comprises three equations and three functions in terms of the nine variables: $x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3$, equation (29).

$$F_i = (x_i - OA_{ix})^2 + (y_i - OA_{iy})^2 - l_i^2, i = 1...3 \tag{32}$$

$$C_i = \left\|\vec{B}_{k|R_{b_1}} - \overrightarrow{OA}_{k|R_f}\right\|^2 - l_i^2, i = 4\dots 6 \tag{33}$$

Hence, three constraints are derived from the following three functions, [Faugère and Lazard 1995]. Two functions can be written using two characteristic platform vector norms between the B_1,B_2 distinct points and the B_1,B_3 ones. The last function comes from these vector multiplication.

$$\begin{aligned}
F_7 &= \left\|\overrightarrow{B_2B}_{1|R_f}\right\|^2 - (x_2-x_1)^2 + (y_2-y_1)^2 + (z_2-z_1)^2 = \left\|\overrightarrow{B_2B}_{1|R_m}\right\|^2 \\
F_8 &= \left\|\overrightarrow{B_3B}_{1|R_f}\right\|^2 - (x_3-x_1)^2 + (y_3-y_1)^2 + (z_3-z_1)^2 = \left\|\overrightarrow{B_3B}_{1|R_m}\right\|^2 \\
F_9 &= (x_3-x_1)*(x_2-x_1)+(y_3-y_1)*(y_2-y_1)+(z_3-z_1)*(z_2-z_1)-\left\|\overrightarrow{B_3B}_{2|R_m}\right\| \wedge \left\|\overrightarrow{B_3B}_{1|R_m}\right\|
\end{aligned} \tag{34}$$

Furthermore, the three last equations (F_7, F_8, F_9) are computed by the following function sequential combinations:

$$\begin{aligned}
F_7 &= -C_7 + F_1 + F_2 \\
F_8 &= -C_8 + F_1 + F_3 \\
F_9 &= 2*C_9 + F_7 + F_8 - 2*F_1
\end{aligned} \tag{35}$$

The formulation is completed with other function combinations obtained by the following algorithm leading to three middle equations (F_4, F_5, F_6). Let $d_7 = \|\overrightarrow{B_2B_1}_{|\mathrm{Rmj}}\|$, $d_8 = \|\overrightarrow{B_3B_1}_{|Rm}\|^2$ and $d_9 = \|\overrightarrow{B_3B_2}\ |Rm\| \wedge \|\overrightarrow{B_3B_1}_{|\mathrm{Rm}}\|$, then for $i = 4, 5, 6$, we compute:

$$\begin{aligned}
C_i &= C_i - a_{B_i}^2 * C_7 - b_{B_i}^2 * C_8 - c_{B_i}^2 * \left(C_7 * C_8 - C_9^2\right) - a_{B_i} * b_{B_i} * (2 * C_9) \\
C_i &= 2 * C_i - a_{B_i} * (F_7 - 2 * F_1) - b_{B_i} * (F_1 + F_2 - F_7) \\
F_i &= C_i - 2 * c_{B_i}^2 * d_7 * (F_1 + F_2 - F_8) - 2 * c_{B_i}^2 * d_8 * (F_1 + F_2 - F_7) \\
&\quad - 2 * c_{B_i}^2 * d_9 * (F_7 - F_9 + F_8 - 2 * F_1) + 2 * \left(a_{B_i} + b_{Bi} - 1\right) * F_1 - F_7 * a_{B_i} - F_8 * b_{B_i}
\end{aligned} \tag{36}$$

The result is an algebraic system with nine equations with the nine unknowns. The **6-6 FKP** formulation using this modified three point model includes six quadratic and three quartic polynomials. The system includes polynomials of higher degree than for the former two position based models. Computations cause to coefficient expansion.

4.2.4 AFP4 - the six point model

The six mobile platform B_i joints can be used in defining 18 variables, [Rolland 2003]. Taking the **IKP** equations (8), a position based variation is obtained:

$$l_i^2 = (x_i - OA_{ix})^2 + (y_i - OA_{iy})^2 + (x_i - OA_{ix})^2, i = 1 \ldots 6 \quad (37)$$

The system is completed with 12 distance constraint equations selected among the distinct B_i passive platform joints. Here are some examples:

$$\begin{aligned} \left\| \overrightarrow{B_i B}_{1|R_f} \right\|^2 &= (x_i - x_1)^2 + (y_i - y_1)^2 + (z_i - z_1)^2 = \left\| \overrightarrow{B_i B}_{1|R_m} \right\|^2, i = 1 \ldots 6 \\ \left\| \overrightarrow{B_j B}_{2|R_f} \right\|^2 &= (x_j - x_2)^2 + (y_j - y_2)^2 + (z_j - z_2)^2 = \left\| \overrightarrow{B_j B}_{2|R_m} \right\|^2, j = 3 \ldots 6 \\ \left\| \overrightarrow{B_k B}_{3|R_f} \right\|^2 &= (x_k - x_3)^2 + (y_k - y_3)^2 + (z_k - z_3)^2 = \left\| \overrightarrow{B_k B}_{3|R_m} \right\|^2, k = 4 \ldots 6 \end{aligned} \quad (38)$$

The formulation results in 18 polynomials in the 18 unknowns:
$\{x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3, x_4, y_4, z_4, x_5, y_5, z_5, x_6, y_6, z_6\}$. The system is then constituted of quadratic polynomials. This variable choice is intuitive and the system yields minimal degree. Finally, the number of variables is maximal.

5. Solving polynomial systems using exact computation

5.1 Mathematical system solving

Kinematics problems contain systems of several equations containing non-linear functions with various variable numbers. These systems can be difficult to solve, especially in the general **6-6** cases and response times actually makes them inappropriate for implementations in design, simulation or control. In some instances, the results may appear to be faulty bringing doubts to the reliability of the methods.

If left without any reliable and performing methods, the tendency, in engineering practice, would be to convert the difficult models into simpler linearized ones. In material handling, this proposal might suffice, however, in high speed milling where the accuracy requirements are more severe, any simplification can have a dramatic impact, whereby result certification becomes an important issue.

However, with proper polynomial formulation, algebraic methods can lead to at least certified and even exact results, whereas numeric methods, unless they implement proper interval analysis, cannot actually obtain certified results since they are prone to numeric instabilities or matrix inversion problems. Therefore, although time consuming, algebraic methods are preferred since they handle integer, rational and symbolic values as such without any truncation or approximation, even when manipulating intermediate results. Hence, there will be no loss of information.

Solving non-linear equation systems will usually result in several complex solutions, out of which a certain subset are real solutions. However, only the real solutions bear practical significance, since they correspond to effective manipulator poses.

5.2 Calculation accuracies

The calculation accuracies are depending upon the type of arithmetic, the behavior of the calculation methods and the quality of the implemented algorithms.

Definition 5.1 *An exact calculation is defined as a calculation which always produces the same exact result to the same specific mathematical problem.*

The result does not contain any error. Its representation is also exact.

Definition 5.2 *A reliable computation is defined as one which will always give the same result from the same initial input data presented in the same format.*

Definition 5.3 *A certified calculation is defined as a reliable computation giving a result distant from the true solution by a certain maximum known accuracy.*

Hence, such a calculation may not be exact. However, the result contains some exact digits. Hence, we shall try to apply a method that computes certified results and if possible exact ones.

For example, lets take the univariate function $f_1(x) = x_2 - 4/25$. Computing $f_1 = 0$, we obtain the exact response: {–2/5, 2/5}. The closed-form resolution calculates exact results with rational numbers. Therefore, the result is certified without any error.

Lets consider $f_2(x) = x_2 - 5$. Solving $f_2 = 0$, the result will be two irrational numbers which can only be represented by truncation. However an interval can be certified to contain the exact result: {[2, 5/2], [–5/2, 2]}. Wherefore, exact computations keep intermediate results in symbolic format whenever possible and only revert to rational or floating boundary numbers for display purposes.

Therefore, any real number can be coded by an interval which width corresponds to the required accuracy. However, the difficulty lies in insuring that the interval contains the exact result which is not known a priori.

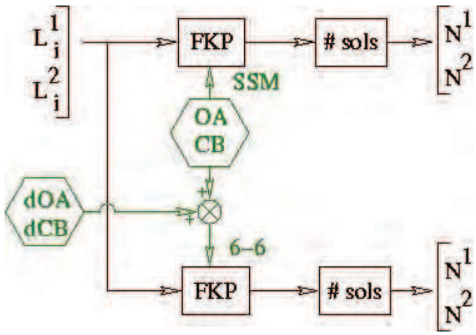

Fig. 6. Bloc Diagram of the Continuation Method

5.3 Solving a non-linear system

Two method groups have been advocated to find all solutions of the **FKP**, namely: continuation methods and variable elimination ones, [Raghavan and Roth 1995]. The first approach is usually realized in a numeric environment and the later algebraic.

5.4 Continuation method with homothopy

In order to compute several solutions, the continuation method can be implemented with a homothopy process. The Continuation approach implements a numerical iterative method which is successively repeated in order to progressively transfer from an original equation system which solutions are predetermined to another system relatively close to the former, Fig. 6.

Let a system of equation be $F(X) = 0$ with variables $X = \{x_1,\ldots,x_n\}$; we wish to find the solution to this equation system. Let $G(X) = 0$ be a similar equation system which roots are already known, namely the variety $V\mathbf{r}\,(I)^G$; then, we set the continuation process as $H(X, \lambda) = G(X) + \lambda\,(F(X)-G(X))$ and commence with $\lambda = 0$. It provides for a mechanism to convert an original equation system into a final one through several steps. At each step, $H(X, \lambda)$ is successively computed with a new value of λ which is increased by a small arbitrary increment $\delta\lambda$ such that $\lambda \in \{0,\ldots, 1\}$. The homothopy principle assumes connectivity between solutions of each system computed with the various λ. More generally, if the system $H(X, \lambda)$ with λ as a variable would be solved, it would result in paths as solutions.

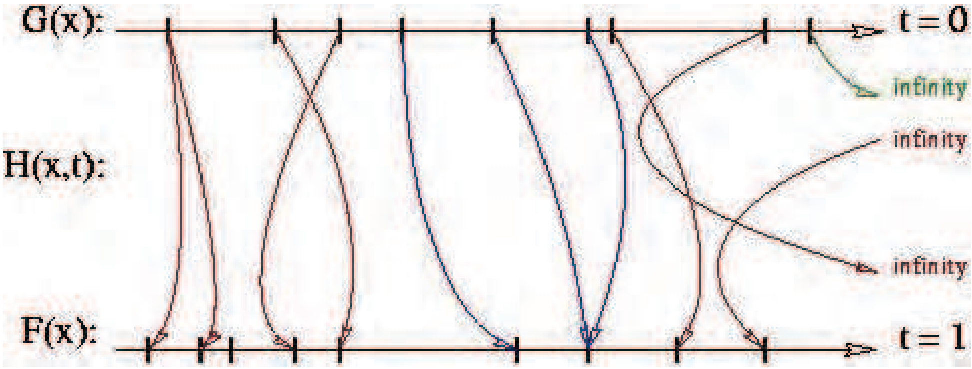

Fig. 7. Examples of path following with the Continuation Method

The continuation method cannot solve any equation system as such and, at each step, when λ is instanciated, the $H(X, \lambda)$ system roots are computed by a typical iterative method, either the Newton one or the new *Geometric Iterative Method* which is a potentially good alternative, [Petuya et al. 2005].

This method was first applied to classical robotics kinematics, [Tsai & Morgan 1984] and then applied to solve the parallel manipulator **FKP**, [Kholi et al. 1992, Sreenivasan and P. Nanua 1992].

Advocating that a little change on parameters of one system shall cause only a small change on solutions, the continuation method could be used to find the 40 solutions on some **6-6 FKP** cases, [Raghavan 1993].

It is feasible to construct an efficient and reliable method; however, the method is still unproven. Moreover, continuation does not alleviate the problems related to the application of a numeric iterative method.

This method can be somewhat delicate to implement. There exist several scenarios which might pose significant problems depending on how the solution paths evolve from $\lambda = 0$ to $\lambda = 1$, see fig. 7, where solutions:

- go to or come from infinity,
- merge or split,
- start complex and become real,
- start real and become complex.

Therefore, proper implementation would require a priori continuation process verification which is still an open question, since this would require solving a one-dimensional system $H(X,\lambda)$ where λ is left as a variable and this is an even more difficult problem. Inasmuch, in many instances, finding a nearby equation system with known roots may not be always be feasible. Then, there is also an issue on what constitutes a sufficiently similar system.
However, it is very difficult to determine precisely what the meaning of sufficiently similar is.

5.5 Variable elimination

5.5.1 Introduction

Most algebraic methods which were implemented to solve the parallel manipulator **FKP** apply one form of variable elimination. Let an algebraic system $F(X) = 0$ be a system of polynomial functions $f_i(X)$, $i = 1,\ldots,m$ with variables $X = \{x_1,\ldots,x_n\}$, the variable elimination approach consists in the transformation of the original system $F(X) = 0$ into another system $H(Y) = 0$ with functions $g_j(X)$, $j = 1,\ldots,p$ with variables $Y = \{y_1,\ldots,y_r\}$ where $r < n$. Ultimately, the goal is to find a method which allows to compute an equation system $H(Y)$ in either triangular format or preferably in univariate form which would be the easiest to solve.
Most variable elimination methods are usually divided into four steps:

- Step 1: Variable elimination.
- Step 2: Solving the univariate equation.
- Step 3: Return or extension to original system variables.

We will examine the variable elimination methods which were successfully applied to solve the **FKP** from which two can be identified:

- method based on resultant calculation including the so-called dialytic elimination,
- method based on Gröbner basis calculation.

5.5.2 Resultant method

Variable elimination can be implemented through a recursive method based on resultants. As input, we give a system of equations with rational coefficients. The output will be one univariate polynomial equation in terms of one of the original variables. Each elimination step involves two polynomial equations which results in one equation with the number of variable reduced by one.

Definition 5.4 *[Cox et al. 1992] Let a system be $F(X) = \{f1,\ldots,fn\} \in Q[x1, \ldots , xn]$; let $P = fi$ and $R = fj$ where $fi = apx1p +\ldots+a0$ and $fj = bqx1q +\ldots+bq$ with $i, j \in \{1, 2,\ldots,n\}$ and $ai, bi \in Q[x2,\ldots,xn]$, let $p = deg(P)$ et $r = deg(R)$ knowing that $p, r \in N^*$; suppose that $a0 \neq 0$ and $b0 \neq 0$; then $Res(f, g, x1) = det(M)$ which is the resultant of P and R in terms of x1 where M is the identified as the Sylvester matrix.*

Then, the Sylvester matrix can be expressed in terms of the polynomial coefficients:

$$M = \begin{pmatrix} a_0 & 0 & \cdots & 0 & b_0 & 0 & \cdots & 0 \\ a_1 & a_0 & \cdots & 0 & b_1 & b_0 & \cdots & 0 \\ \vdots & & \ddots & & \vdots & & \ddots & \\ a_l & & \cdots & a_1 & b_m & & \cdots & b_1 \\ 0 & a_l & \cdots & a_2 & 0 & b_m & \cdots & b_2 \\ \vdots & & \ddots & & \vdots & & \ddots & \\ 0 & & \cdots & a_l & 0 & & \cdots & b_1 \end{pmatrix} \qquad (39)$$

Inasmuch, we can write: $Res(P, R, x_1) = det(Sylv(P, R, x_1))$. If we examine the Sylvester matrix, we can observe that part with the a_i parameters contains m columns and the one with b_i n columns.The following proposition holds and its proof is described in [Cox et al. 1992]: *The resultant $Res(f, g, x)$ is the first ideal of elimination $< f, g > \cap k[x^2,\ldots,x_n]$; moreover, $Res(f, g, x) = 0$ iff f, g have a common factor in $k[x_1,\ldots,x_n]$ which has a positive degree in x*. The nature of this factor has to be determined and we wish to establish if it is only one root of functions *f* and *g*. To answer that question, the following corollary will be employed: *If $f, g \in C[X]$ then $Res(f, g, x) = 0$ iff f and g contain a common root in C*. This common root is determined by computing $det(Sylv(P, R, x_1) = 0$. The nature of this root has to be determined, notably if it is a partial one and the answer will come from the following proposition, [Cox et al. 1992] : *Knowing that $f, g \in C[X]$, let $a_0, b_0 \neq 0$ and $a_0, b_0 \in C[x_2,\ldots, x_n]$, if $Res(f, g, x) \in C[x_2,\ldots,x_n]$ cancels at $(c_2,\ldots,c_n)$, then we obtain that either $a_0 b_0 = 0$ at $(c_2,\ldots,c_n)$ or either $\exists\, c_1 \in C$ such as f and g cancel.*
In certain instances, the head terms of the polynomials can cancel which will result in the cancellation of the determinant and the process consequently adds one extraneous root.
In order to obtain the univariate equation, a recursive algorithm will be applied. Firstly, we calculate $n-1$ resultants $h_k = Res(f_{k+1}, f_1, x_1)$ on variable x_1 for $k = 1,\ldots,n-1$. Secondly, we compute $n-2$ resultants $h(2)_j = Res(h_{j+1}, h_1, x_2)$ on variable x_2 for $j = 1,\ldots,n-2$ and we continue in the same fashion until the univariate equation is determined. An almost triangular equation system is constructed.

$$\begin{bmatrix} f_1(X)=0,\ldots,f_{n-2}(X)=0, f_{n-1}(X)=0, f_n(X)=0 \\ h_1(x_2,\ldots,x_n),\ldots,h_{n-2}(x_2,\ldots,x_n), h_{n-1}(x_2,\ldots,x_n) \\ h_1^{(2)}(x_3,\ldots,x_n),\ldots,h_{n-2}^{(2)}(x_3,\ldots,x_n) \\ \vdots \\ h_1^{n-2}(x_{n-1},x_n), h_2^{(n-2)}(x_{n-1},x_n) \\ H(x_n) \end{bmatrix} \qquad (40)$$

The last $H(x_n)$ is the targeted univariate equation. However, this equation cannot be considered equivalent to the initial algebraic system because the head terms can cancel.
The return step to original variables is performed by substituting back through the triangular system. The equation is solved $H(x_n) = 0$ and we obtain a series of w roots $\{x_n\}$. We take the w roots, one by one, which is introduced in one of the equations $h_1^{(n-2)}(x_n-1) = 0$ or $h_2^{(n-2)}(x_n-1) = 0$ and obtain the w roots $\{x_n-1\}$. We continue until x_1 is isolated.
The **6-6 FKP** has been solved applying resultants, [Husty 1996], in a computer algebra environment to avoid intermediate result truncation, since, in a sense, parameter truncation can be envisioned as changing the manipulator configuration.

A variation to the resultant method is called the dialytic elimination. Let the variable set be $X = \{x_1,\ldots,x_n\}$ of the algebraic system $F(X) = 0$; then select any variable x_i and set it as the hidden variable, then a monomial vector is constructed around x_i for the system $F(X)$ which is expressed as $\overline{W} = (1, x_i, x_i^2,\ldots)$. The **FKP** is rewritten as a linear system in terms of $\overline{W}$:

$$A\overline{W} = 0 \tag{41}$$

where : $\overline{W} \neq 0$

Being a generalization of $Res(P, Q, x_1) = det(Sylv(P, Q, x_1)$, it is subjected to the same risks of root addition through the head term cancellation. Dialytic elimination has been implemented to solve the **FKP** of the **3-RS** or **MSSM** parallel manipulators, [Griffis and Duffy 1989, Dedieu and Norton 1990, Innocenti and Parenti-Castelli 1990]. Satisfactory results were produced on simple parallel manipulators, [Raghavan and Roth 1995].

5.6 Gröbner Bases

Lets denote by $Q[x_1,\ldots,x_n]$ the ring of polynomials with rational coefficients. For any n-uple $\mu = (\mu_1,\ldots, \mu_n) \in N^n$, lets denote by X^μ the monomial $X_1^{\mu_1} \cdot\ldots\cdot X_n^{\mu_n}$. If < is an admissible monomial ordering and $P = \sum_{i=0}^{r} a_i X^{\mu^{(i)}}$ any polynomial in $Q[X_1,\ldots,X_n]$, the following polynomial notations are necessary :

- $LM(P,<) = max_{i=0\ldots r}, <X^{\mu(i)}$ is the leading monomial of P for the order <,
- $LC(P,<) = a_i$ with i such that $LT(P) = X^{\mu(i)}$ is the leading coefficient of P for <,
- $LT(P,<) = LC(P,<) \cdot LM(P,<)$ is the leading term of P for <.

Lets denote by $x_1,\ldots,x_n$ the unknowns and $S = \{P_1 =,\ldots,P_s\}$ any polynomial system as a subset of $Q[x_1,\ldots,x_n]$. A point $\alpha \in C^n$ is a zero of S if $P_i(\alpha) = 0\ \forall i = 1\ldots s$. Actually, any large polynomial equation system cannot be directly or explicitly solved. Thus, it is necessary to revert to mathematical objects containing sufficient information for resolution. Any polynomial system is then described by an ideal:

Definition 5.5 *[Cox et al. 1992] An ideal **I** is defined as the set of all polynomial P(X) that can be constructed by multiplying and adding all polynomials in the ring of polynomials with the original polynomials in the set S.*

A *Gröbner Basis G* is then as a computable polynomial generator set of a selected polynomial set $S = \{P_1,\ldots,P_s\}$ with good algorithmic properties and defined with respect to a monomial ordering. This basis is a mathematical object including the ideal ***I*** information. The *lexicographic* and *degree reverse lexicographic* (*DRL*) orders are usually implemented, [Cox et al. 1992, Geddes et al. 1994]. Given any admissible monomial ordering, the classical Euclidean division can be extended to reduce a polynomial by another one in $Q[X_1,\ldots,X_n]$. This polynomial reduction can be generalized to the reduction by a polynomial list. The reduction output depends on the monomial ordering < and the polynomial order.

Definition 5.6 *Given any admissible monomial ordering, <, a Gröbner Basis G with respect to < of an ideal **I** $\subset Q[X_1,\ldots,X_n]$ is a finite subset of **I** such that: $\forall f \in$ **I** , $\exists g \in G$ such that $LM(g,<)$ divides $LM(f,<)$.*

Some useful *Gröbner Basis* properties are described in the following theorem:

Theorem 5.1 *Let G be a Gröbner Basis G of an ideal **I** $\subset Q[X_1,\ldots,X_n]$ for any < monomial ordering, then a polynomial $p \in Q[X_1,\ldots,X_n]$ belongs to **I** if and only if the reduction algorithm Reduce (p, G,<) = 0; the reduction does not depend on the order of the polynomials in the list of G; it can be used as a simplification function.*