

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.

For more information visit www.intechopen.com



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.

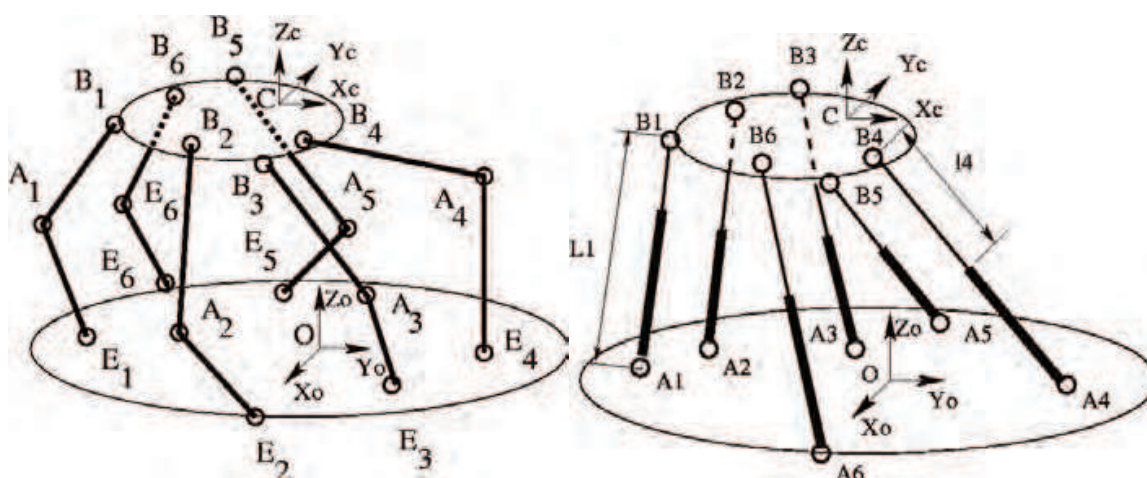


Fig. 1. The general 6-6 hexapod manipulators

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

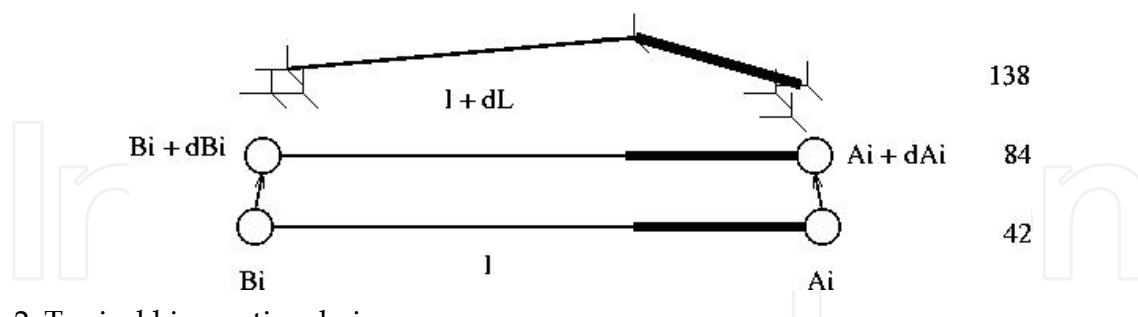


Fig. 2. Typical kinematics chains

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 A_i inter-axis distances e_{1^a} , e_{2^a} and e_{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 e_{1^b} , e_{2^b} and e_{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 $\Gamma = \{\rho_1, \rho_2, \dots, \rho_6\}$.

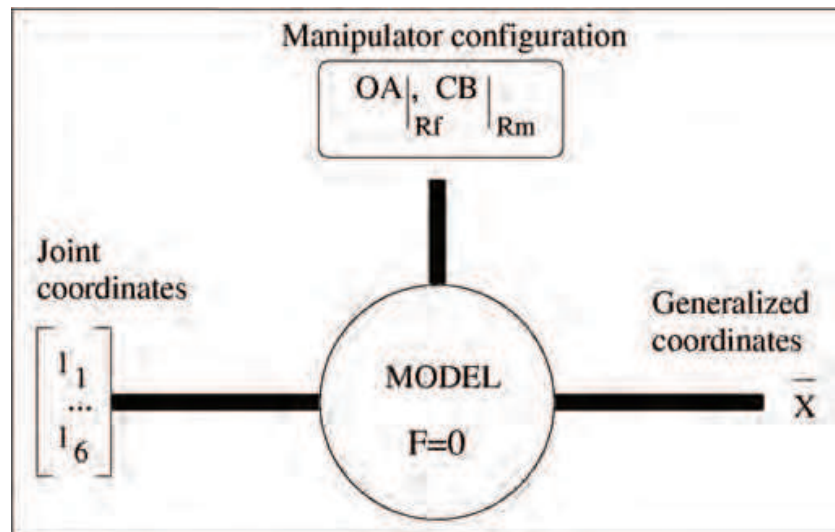


Fig. 3. Kinematics model

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

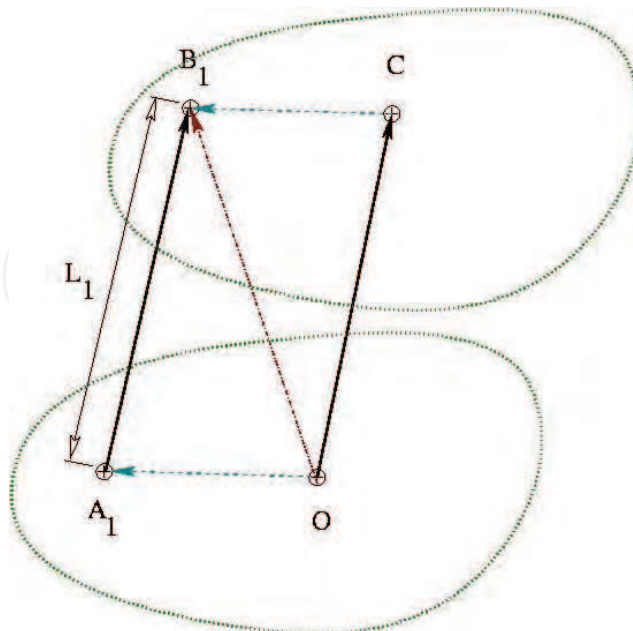


Fig 4. The vectorial formulation

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 $\overrightarrow{A_i B_i} = U_1(X)$. Inasmuch, vector $\overrightarrow{A_i B_i}$ 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, $i = 1, \dots, 6$, each platform point $\overrightarrow{OB_{i|R_f}}$ can be expressed in terms of the distance constraint, [Merlet 1997]:

$$l_i^2 = \|A_i B_i\|^2, i = 1 \dots 6 \tag{1}$$

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 $\overrightarrow{OB_{i|R_f}}$ which meets constraints 1 has a rotation matrix \mathfrak{R} such that:

$$\overrightarrow{OB_{i|R_f}} = \overrightarrow{OC_{i|R_f}} + \mathfrak{R} \cdot \overrightarrow{CB_{i|R_m}}, i = 1 \dots 6 \tag{2}$$

Substituting 2 in 1, we obtain:

$$l_i^2 = \left\| \overrightarrow{OC}_{|R_f} + \mathfrak{R} \cdot \overrightarrow{CB}_{i|R_m} - \overrightarrow{OA}_{i|R_f} \right\|^2, i = 1 \dots 6 \quad (3)$$

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

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

3.2 Position based equations

In 3D space, any rigid body can be positioned by 3 of its distinct non-collinear 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: $\overrightarrow{OB}_{i|R_f} = [x_i, y_i, z_i]$ for $i = 1 \dots 3$. The $\overrightarrow{OB}_{i|R_f}$ 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:

$$u_1 = \frac{\overrightarrow{B_1 B_2}}{\|\overrightarrow{B_1 B_2}\|}, u_2 = \frac{\overrightarrow{B_1 B_3}}{\|\overrightarrow{B_1 B_3}\|}, u_3 = u_1 \wedge u_2 \quad (5)$$

Any platform point M can be expressed by $\overrightarrow{B_1 M} = a_M u_1 + b_M u_2 + c_M u_3$ 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 $a_{B_i}, b_{B_i}, c_{B_i}, i = 1 \dots 6$ and can explicitly be deduced from $\overrightarrow{CB}_{i|R_m}$ by solving the following linear system of equations:

$$\overrightarrow{B_1 B}_{i|R_{b1}} = a_{B_i} u_1 + b_{B_i} u_2 + c_{B_i} u_3, i = 1 \dots 6 \quad (6)$$

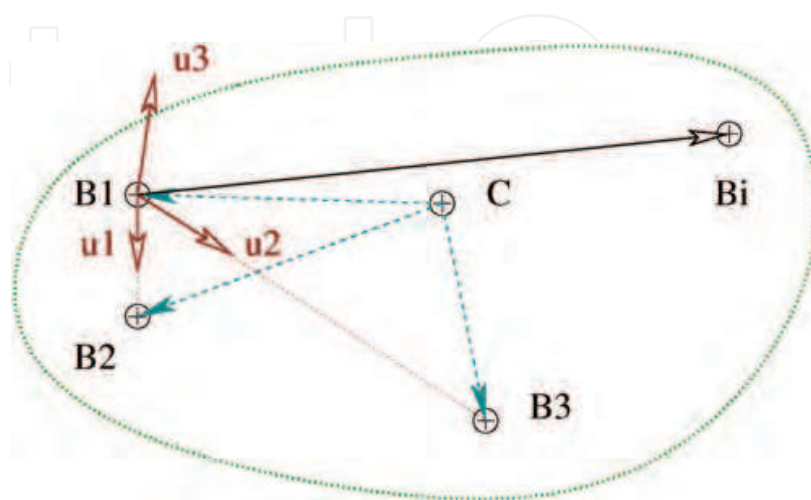


Fig. 5. The platform three point coordinate system

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

$$l_i^2 = (x_i - \overrightarrow{OA_{ix}})^2 + (y_i - \overrightarrow{OA_{iy}})^2, i = 1 \dots 3 \tag{7}$$

$$l_i^2 = \|\overrightarrow{B_k}|_{R_{b_1}} - \overrightarrow{OA_k}|_{R_f}\|^2, i = 4 \dots 6 \tag{8}$$

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 $c_j^2 + s_j^2 = 1$ which complete the equation system:

$$F_i = (\overrightarrow{OC}|_{R_f} - \overrightarrow{OA_i}|_{R_f})^2 + (\overrightarrow{OC}|_{R_f} - \overrightarrow{OA_i}|_{R_f}) \Re \cdot \overrightarrow{CB_i}|_{R_m} - L_i^2, i = 1, \dots, 6 \tag{9}$$

$$F_j = c_j^2 + s_j^2 - 1, j = 1, 2, 3 \tag{10}$$

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 $i = 1, 2, 3$:

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

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:

$$F_i = (\overrightarrow{OC}|_{R_f} - \overrightarrow{OA_i}|_{R_f})^2 + (\overrightarrow{OC}|_{R_f} - \overrightarrow{OA_i}|_{R_f}) \Re \cdot \overrightarrow{CB_i}|_{R_m} - L_i^2, i = 1, \dots, 6 \tag{12}$$

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 $\overrightarrow{OC}_{|R_f}$ coordinates as unknowns, [Lazard 1993, Sreenivasan et al. 1994, Bruyninckx and DeSchutter 1996]. The variables are then $\{X_c, Y_c, Z_c, r_{ij}, j=1\dots 3, i=1\dots 3\}$. Since \mathcal{R} is a rotation matrix, the following relations hold: $\mathcal{R}\mathcal{R}^T = Id$ or $\det(\mathcal{R}) = 1$. These relations are redundant since $\mathcal{R}\mathcal{R}^T$ 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.

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

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

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

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

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

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

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

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 $\bar{q} = q_1 \mathbf{i} + q_2 \mathbf{j} + q_3 \mathbf{k}$ 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 \mathfrak{R} 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:

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

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 $\{X_c; Y_c; Z_c; q_0; q_1; q_2; q_3\}$:

$$F_i = \left(\overline{OC}_{|R_f} - \overline{OA}_{i|R_f} \right)^2 + \left(\overline{OC}_{|R_f} - \overline{OA}_{i|R_f} \right) \mathfrak{R} \cdot \overline{CB}_{i|R_m} - L_i^2, i = 1, \dots, 6 \quad (22)$$

$$F_7 = q_0^2 + q_1^2 + q_2^2 + q_3^2 - 1 \quad (23)$$

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 \mathfrak{R} 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 $s_i = \mathbf{OA}_{|R_f}$ and $t_i = \mathbf{CB}_{|R_m}$, then:

$$\begin{aligned} \bar{c}^t \bar{c} &= 1 \\ \bar{g}^t \bar{c} &= 0 \\ \bar{g}^t \bar{g} - I_4 \bar{c}^t \bar{c} &= 0 \\ \bar{c}^t s_i \bar{c} + 2 \bar{g}^t t_i \bar{c} &= 0 \end{aligned} \quad \text{For } i = 2, \dots, 6 \quad (24)$$

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

$$F_i = \left(\overline{OC}_{|R_f} - \overline{OA}_{i|R_f} \right)^2 + \left(\overline{OC}_{|R_f} - \overline{OA}_{i|R_f} \right) \mathfrak{R} \cdot \overline{CB}_{i|R_m} - L_i^2 \quad (25)$$

$$F_7 = FC_1 \quad (26)$$

$$F_8 = FC_2 \quad (27)$$

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: $\overrightarrow{OB}_{i|R_f} = [x_i, y_i, z_i]$ for $i = 1 \dots 3$.

Using the relations 6, the constraint equations $L_i^2 = \|\overrightarrow{A_i B_i}_{|R_f}\|^2, i = 1, \dots, 6$ 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:

$$F_i = (x_i - OA_{ix})^2 + (y_i - OA_{iy})^2 + (z_i - OA_{iz})^2 - L_i^2, i = 1 \dots 3 \quad (28)$$

$$F_j = \|\overrightarrow{B_j R_{b_1}} - \overrightarrow{OA_j}_{|R_f}\|^2 - L_j^2, j = 4 \dots 6 \quad (29)$$

The mobile platform geometry yields the following three distance equations:

$$\begin{aligned} F_7 &= \|\overrightarrow{B_2 B_1}_{|R_f}\|^2 - (x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2 = \|\overrightarrow{B_2 B_1}_{|R_m}\|^2 \\ F_8 &= \|\overrightarrow{B_3 B_1}_{|R_f}\|^2 - (x_3 - x_1)^2 + (y_3 - y_1)^2 + (z_3 - z_1)^2 = \|\overrightarrow{B_3 B_1}_{|R_m}\|^2 \\ F_9 &= \|\overrightarrow{B_3 B_2}_{|R_f}\|^2 - (x_3 - x_2)^2 + (y_3 - y_2)^2 + (z_3 - z_2)^2 = \|\overrightarrow{B_3 B_2}_{|R_m}\|^2 \end{aligned} \quad (30)$$

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, let's take the two mobile platform vectors $\overrightarrow{B_1 B_2}$ and $\overrightarrow{B_1 B_3}$, then the last constraint is calculated from these two vector multiplication:

$$\begin{aligned}
 F_7 &= \left\| \overrightarrow{B_2 B_{1|R_f}} \right\|^2 - (x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2 = \left\| \overrightarrow{B_2 B_{1|R_m}} \right\|^2 \\
 F_8 &= \left\| \overrightarrow{B_3 B_{1|R_f}} \right\|^2 - (x_3 - x_1)^2 + (y_3 - y_1)^2 + (z_3 - z_1)^2 = \left\| \overrightarrow{B_3 B_{1|R_m}} \right\|^2 \\
 F_9 &= (x_3 - x_1) * (x_2 - x_1) + (y_3 - y_1) * (y_2 - y_1) + (z_3 - z_1) * (z_2 - z_1) - \left\| \overrightarrow{B_3 B_{2|R_m}} \right\| \wedge \left\| \overrightarrow{B_3 B_{1|R_m}} \right\|
 \end{aligned}
 \tag{31}$$

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).

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

$$C_i = \left\| \overrightarrow{B_{k|R_{b_1}}} - \overrightarrow{OA_{k|R_f}} \right\|^2 - l_i^2, i = 4...6 \tag{33}$$

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.

$$\begin{aligned}
 F_7 &= \left\| \overrightarrow{B_2 B_{1|R_f}} \right\|^2 - (x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2 = \left\| \overrightarrow{B_2 B_{1|R_m}} \right\|^2 \\
 F_8 &= \left\| \overrightarrow{B_3 B_{1|R_f}} \right\|^2 - (x_3 - x_1)^2 + (y_3 - y_1)^2 + (z_3 - z_1)^2 = \left\| \overrightarrow{B_3 B_{1|R_m}} \right\|^2 \\
 F_9 &= (x_3 - x_1) * (x_2 - x_1) + (y_3 - y_1) * (y_2 - y_1) + (z_3 - z_1) * (z_2 - z_1) - \left\| \overrightarrow{B_3 B_{2|R_m}} \right\| \wedge \left\| \overrightarrow{B_3 B_{1|R_m}} \right\|
 \end{aligned}
 \tag{34}$$

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 $d_7 = \left\| \overrightarrow{B_2 B_{1|R_m}} \right\|$, $d_8 = \left\| \overrightarrow{B_3 B_{1|R_m}} \right\|^2$ and $d_9 = \left\| \overrightarrow{B_3 B_{2|R_m}} \right\| \wedge \left\| \overrightarrow{B_3 B_{1|R_m}} \right\|$, then for $i = 4, 5, 6$, we compute:

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

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:

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

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

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

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.

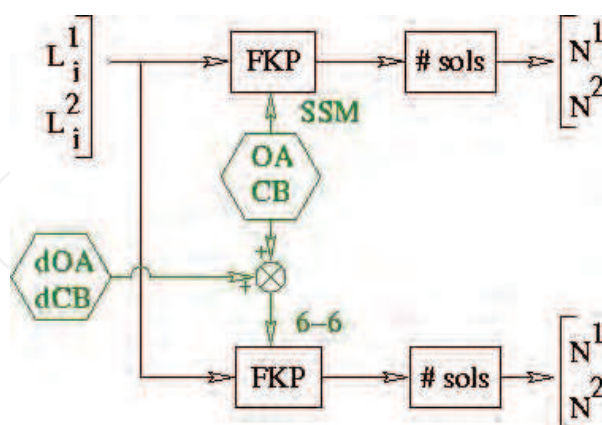


Fig. 6. Bloc Diagram of the Continuation Method

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 $X = \{x_1, \dots, x_n\}$; 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 $Vr(I)^G$; 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 $\lambda \in \{0, \dots, 1\}$. 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.

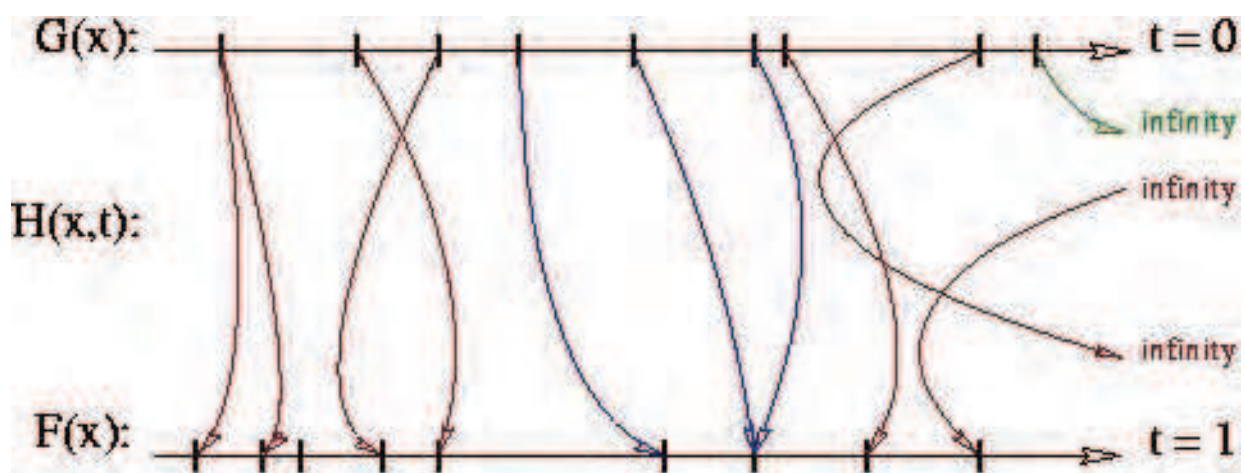


Fig. 7. Examples of path following with the Continuation Method

The continuation method cannot solve any equation system as such and, at each step, when λ is instantiated, 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 $f_i(X)$, $i = 1, \dots, m$ with variables $X = \{x_1, \dots, x_n\}$, the variable elimination approach consists in the transformation of the original system $F(X) = 0$ into another system $H(Y) = 0$ with functions $g_j(X)$, $j = 1, \dots, p$ with variables $Y = \{y_1, \dots, y_r\}$ 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 $F(X) = \{f_1, \dots, f_n\} \in Q[x_1, \dots, x_n]$; let $P = f_i$ and $R = f_j$ where $f_i = a_p x^p + \dots + a_0$ and $f_j = b_q x^q + \dots + b_0$ with $i, j \in \{1, 2, \dots, n\}$ and $a_i, b_i \in Q[x_2, \dots, x_n]$, let $p = \deg(P)$ et $r = \deg(R)$ knowing that $p, r \in \mathbb{N}^*$; suppose that $a_0 \neq 0$ and $b_0 \neq 0$; then $\text{Res}(f, g, x_1) = \det(M)$ which is the resultant of P and R in terms of x_1 where M is the identified as the Sylvester matrix.

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

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

Inasmuch, we can write: $\text{Res}(P, R, x_1) = \det(\text{Sylv}(P, R, x_1))$. 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 $\text{Res}(f, g, x)$ is the first ideal of elimination $\langle f, g \rangle \cap k[x^2, \dots, x_n]$; moreover, $\text{Res}(f, g, x) = 0$ iff f, g have a common factor in $k[x_1, \dots, x_n]$ 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 $\text{Res}(f, g, x) = 0$ iff f and g contain a common root in C .* This common root is determined by computing $\det(\text{Sylv}(P, R, x_1)) = 0$. 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 $a_0, b_0 \in C[x_2, \dots, x_n]$, if $\text{Res}(f, g, x) \in C[x_2, \dots, x_n]$ cancels at (c_2, \dots, c_n) , then we obtain that either $a_0 b_0 = 0$ at (c_2, \dots, 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 $h_k = \text{Res}(f_{k+1}, f_1, x_1)$ on variable x_1 for $k = 1, \dots, n - 1$. Secondly, we compute $n - 2$ resultants $h^{(2)}_j = \text{Res}(h_{j+1}, h_1, x_2)$ on variable x_2 for $j = 1, \dots, n - 2$ and we continue in the same fashion until the univariate equation is determined. An almost triangular equation system is constructed.

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

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 $X = \{x_1, \dots, x_n\}$ 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 $\bar{W} = (1, x_i, x_i^2, \dots)$. The **FKP** is rewritten as a linear system in terms of \bar{W} :

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

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

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

- $LM(P, <) = max_{i=0..r} < X^{\mu^{(i)}}$ 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, \dots, x_n the unknowns and $S = \{P_1, \dots, P_s\}$ any polynomial system as a subset of $Q[x_1, \dots, x_n]$. A point $\alpha \in C^n$ is a zero of S if $P_i(\alpha) = 0 \forall i = 1..s$. 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 $S = \{P_1, \dots, P_s\}$ 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 $Q[X_1, \dots, X_n]$. 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 $I \subset Q[X_1, \dots, X_n]$ 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 $I \subset Q[X_1, \dots, X_n]$ for any $<$ monomial ordering, then a polynomial $p \in Q[X_1, \dots, X_n]$ 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.*

The classical method for computing a *Gröbner Basis* is based on Buchberger's algorithm [Buchberger et al. 1982, Buchberger 1985]. Recently, Faugère proposed more powerful algorithms, namely *accel* and *F4* implemented in the *FGb* software, [Faugère 1999].

5.7 Gröbner Bases and zero dimensional systems

Definition 5.7 [Lazard 1992a] A zero-dimensional system is defined as a mathematical equation system with a variety (solution set) constituted by a finite number of complex points.

From a mathematical point of view, any variety is a valid result. From an engineering one, we seek a variety which represents an exploitable result. For any parallel manipulator **FKP**, the zero-dimensional systems allow pose analysis, since each *real* solution corresponds to each manipulator pose. Otherwise the problems fall in the category of rarely usable singular configurations. Detection of zero-dimensional systems is performed by implementing the following theorem:

Theorem 5.2 Let $G = \{g_1, \dots, g_l\}$ be a Gröbner Basis for any ordering $<$ of any system $S = \{P_1, \dots, P_s\} \in Q[X_1, \dots, X_n]^s$. The two following properties are equivalent:

- For all index $i, i = 1 \dots n$, there exists a polynomial $g_j \in G$ and a positive integer n_j such that $X_i^{n_j} = LM(g_j, <)$;
- The system $\{P_1 = 0, \dots, P_s = 0\}$ has a finite number of solutions in C^n .

Hence, one can determine if the **FKP** resolution yields a finite or infinite number of solutions. This is obviously an important issue, since an algebraic system with an infinite number of solutions (variety of degree one or higher) cannot be directly exploited by a control system and failure is usually the outcome.

5.8 The lexicographic Gröbner Basis

The choice of the monomial ordering is critical for the *Gröbner Basis* computation efficiency, which is evaluated by the computation times, the size and the shape of the result. A *lexicographic Gröbner Basis* with $X_1 < X_2 < \dots < X_n$ as variable order has always the following general shape:

$$\begin{cases} f_1(X_1) = 0, f_2(X_1, X_2) = 0, \dots \\ f_{k_2}(X_1, X_2) = 0, f_{k_2+1}(X_1, X_2, X_3) = 0, \dots \\ f_{k_{n-1}+1}(X_1, \dots, X_n) = 0, \dots \\ f_{k_n}(X_1, \dots, X_n) = 0 \end{cases} \quad (42)$$

Whilst computing a *lexicographic Gröbner Basis* using Buchberger's algorithm yields doubly exponential complexity in the worst case, an order change will save computation time. Therefore, one *DRL Gröbner Basis* may first be computed in d^n polynomial time where d is the maximum total degree of the equations. Then, the *DRL Gröbner Basis* can be converted into a *lexicographic* one in $O(nD^3)$ arithmetic operations, [Faugère et al. 1991]. In the case of any zero-dimensional system, this can be done using exclusively linear algebra techniques, [Faugère et al. 1991].

Definition 5.8 Let S be a system constituted by the polynomials $p_1, \dots, p_s \in Q[x_1, \dots, x_n]$, without square factors. Let $G = \{g_1, \dots, g_l\}$ be a Gröbner Basis computed for any ordering $<$ on the system S . If the univariate polynomial $f(x_1)$ is without any multiple factors, then the equation is square-free, the first coordinates of all solutions are distinct and the solutions yield no multiple root. Then, the system is considered in **Shape Position**.

If the polynomial system is in **Shape Position**, the Gröbner Basis has the following simplified univariate format:

$$f_1(X_1) = 0, X_2 = f_2(X_1) = 0, \dots, X_n = f_n(X_1) \tag{43}$$

In this format, the *lexicographic Gröbner Basis* can be computed directly using the F4 algorithm, [Faugère 1999].

5.9 The rational univariate representation

Another root representation is given by the *Rational Univariate Representation*, hereafter identified *Rational Univariate Representation*, [Rouillier 1999]. In the particular case of systems in **Shape Position**, the univariate variable is the first one in the original system ones, so that the *Rational Univariate Representation* can be written in the following format:

$$\{h(X_k) = 0, X_2 = h_2(X_k)/h_0(X_k), \dots, X_n = h_n(X_k)/h_0(X_k)\} \tag{44}$$

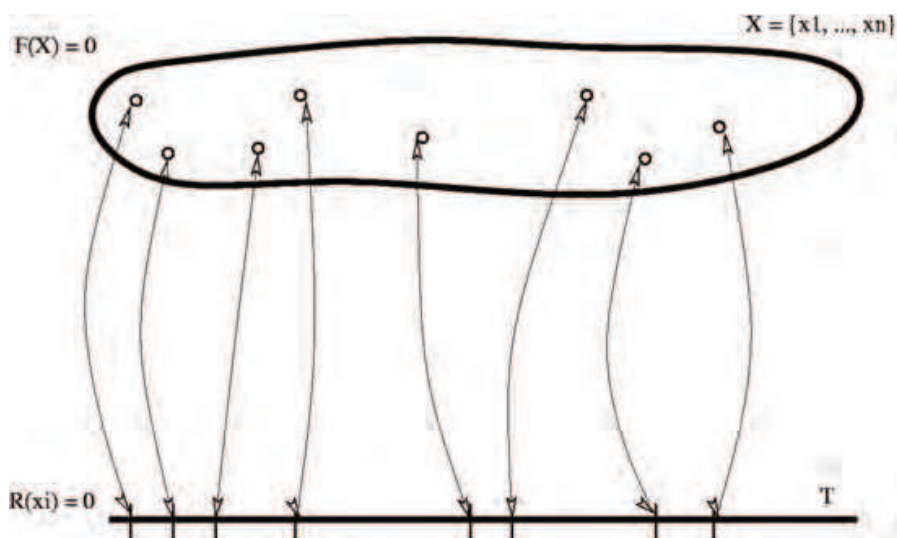


Fig. 8. The correspondence between the *Rational Univariate Representation* and original system varieties.

$h(X_k)$ where $1 \leq k \leq n$ denotes the univariate equation, h_i $i = 0 \dots n$ the univariate return functions. These polynomials are elements of $Q[X_k]$. Inasmuch, h and h_0 are coprime. Moreover, if the *Rational Univariate Representation* system is in **Shape Position** and if $h(X_k)$ is square-free, the *Rational Univariate Representation* can be converted into a *lexicographic Gröbner Basis* and the *Rational Univariate Representation* expression can be simply computed from this basis. This situation is generally the case in most **FKP** instantiations:

$$h(X_k) = f(X_k), g(X_k) = f'(X_k) \tag{45}$$

Moreover, since f and f_0 are coprimes, f_0 is invertible modulo f ; this leads to: $h_i = f_i \cdot (f)^{-1}$ modulo f , $i = 2 \dots n$.

Definition 5.9 Let S be a system constituted by the polynomials $p_1, \dots, p_s \in Q[x_1, \dots, x_n]$ and let $G = \{g_1, \dots, g_l\}$ be a Gröbner Basis for any ordering $<$ of the system S . Then, the **RR** form is defined as the system $R = \{p_1, p_1^{-1}p_2 \text{ mod } G, p_1^{-1}p_3 \text{ mod } G, \dots, p_1^{-1}p_k \text{ mod } G\} \in Q[x_1, \dots, x_n]$ where p_1^{-1} is calculated such that $(p_1 p_1^{-1} \text{ modulo } G) = 1$.

The rationale behind the *RR* form computation is to reduce the polynomial system S . The computed *Gröbner Basis* G on any ordering $<$ of the system S is used for the system reduction. Then, the reduced system R resolution shall be less difficult.

Definition 5.10 Let S be a system constituted by the polynomials $p_1, \dots, p_s \in Q[x_1, \dots, x_n]$, if the system S is without square factors and in **Shape Position** and if a lexicographic *Gröbner Basis* is computable, then the system *RR* form R' can be deduced.

Since the S system reduction by a lexicographic *Gröbner Basis* leads to a further reduced system R , one can conjecture faster **FKP** resolution. However, in some instances, p_1^{-1} can hardly be computed and the *RR* form cannot be deduced. Hence, the **FKP** is solved by first computing a *Gröbner Basis* on a **DRL** order and then we rely on the aforementioned order change to obtain a lexicographic *Gröbner Basis*. In the rare instances where the **FKP** system is not in **Shape Position** and a lexicographic *Gröbner Basis* is not computable either by the *RR* form or by order change, the *Rational Univariate Representation* can be always be computed by reverting to the more robust and lengthy algorithm taking as input any monomial ordering *Gröbner Basis* and calculating the multiplication tables and matrices, [Rouillier 1999].

In practice, the *Rational Univariate Representation* coefficients are smaller than the lexicographic *Gröbner Basis* ones, [Alonso et al. 1996]. This is the rationale behind the preference of the *Rational Univariate Representation* expression which can be computed using the algorithm described in [Rouillier 1999]. Anyhow, it should be deduced from a lexicographic *Gröbner Basis* if the system is in **Shape Position**. Hence, in any case, the determination of a lexicographic *Gröbner Basis* is first tried and then, a *Rational Univariate Representation* is computed. In fact, for any parallel manipulator **FKP**, the computation of the *RR* form is preferred. Finally, the method insures that the *Rational Univariate Representation* system is equivalent to the former polynomial system. Each *Rational Univariate Representation* system solution corresponds to one and only one original system solution and the properties are preserved, fig. 8.

5.10 Real roots isolation

Once the *Rational Univariate Representation* is computed, then the real roots of the univariate equation $h(X_k)$ roots can be isolated either using rapid numeric methods or exact algebraic methods. As far as algebra is concerned, such computations can be performed by a method such as the Uspensky one based on Descartes' sign rule and Sturm's theorem. An improved algorithm is introduced in [Rouillier and Zimmermann 2001] applying one Vincent theorem. In computer algebra and in particular in the proposed method, a natural way for coding a real algebraic number α is given by either a square-free polynomial f with rational coefficients such that $f(\alpha) = 0$ or an interval $[l, r]$ with rational bounds containing α . Inasmuch, if $\beta \neq \alpha$ and $f(\beta) = 0$ then $\beta \notin [l, r]$.

Let α be a root of $h(T)$ represented by $(\bar{h}, [l, r])$, then the resolution of $h(T) = 0$ produces a solution list $\{\alpha_1, \dots, \alpha_r\} = [(\bar{h}, [l_i, r_i]), i = 1 \dots r]$ where \bar{h} is the square-free part of h . Moreover, it is very easy to refine the intervals $[l_i, r_i]$ by dichotomy since \bar{h} is square-free which induce that $\text{sign}(\bar{h}(l_i)) \neq \text{sign}(\bar{h}(r_i))$ if $l_i \neq r_i$.

Since h and h_0 are coprime by definition of the *Rational Univariate Representation*, one can refine $[l, r]$ so that it doesn't contain any real root of G . One can then simply use interval arithmetic to compute $[l^{(i)}, r^{(i)}] = h_i/h_0([l, r]); i = 1 \dots n$. The complete method allows one solution of the equation $h(T) = 0$ to correspond to each solution of the system $F(X) = 0$ since the bijection is guaranteed and the properties are preserved, fig. 8.

Then, using the return functions h_i/h_0 , for each α , we compute an interval product $\prod_{i=0}^n [l^{(i)}, r^{(i)}]$ containing $[h_1/h_0(\alpha), \dots, h_n/h_0(\alpha)]$. For refining $\prod_{i=0}^n [l^{(i)}, r^{(i)}]$, it is sufficient to refine and introduce again $[l, r]$ in the coordinate functions h_i/h_0 ; $i = 1 \dots n$.

The *Rational Univariate Representation* system real root isolation requires a second step where the original system roots are determined in terms of the original variables using the return functions provided by the *Rational Univariate Representation*.

By computing a *Rational Univariate Representation* and then isolating the real roots as described above, we obtain an exact method for isolating all the real roots of any polynomial system with a finite number of complex roots verified during the computations using theorem 5.2. The *Rational Univariate Representation* has rational coefficients and the isolating intervals have rational bounds. For each solution $\alpha = (\alpha_1, \dots, \alpha_n)$, the method produces :

- a box $B_\alpha \prod_{i=0}^n [l^{(i)}, r^{(i)}] \in R^n$ such that $\alpha \in B_\alpha$ ($l_i, r_i \in Q^2$ with exact (rational) values and if $\beta \neq \alpha$ is a root of the system, then $\beta \notin B_\alpha$;
- a function for refining B_α up to any precision $\epsilon > 0$ ($|r_i - l_i| < \epsilon, \forall i = 1 \dots n$).

5.11 The certified method description

Assembling the aforementioned algorithms, we obtain a complete certified method since it handles the intermediate results with exact representations. Taking into account the fact that most parallel manipulator FKP are in **Shape Position**, the method has been simplified, fig. 9.

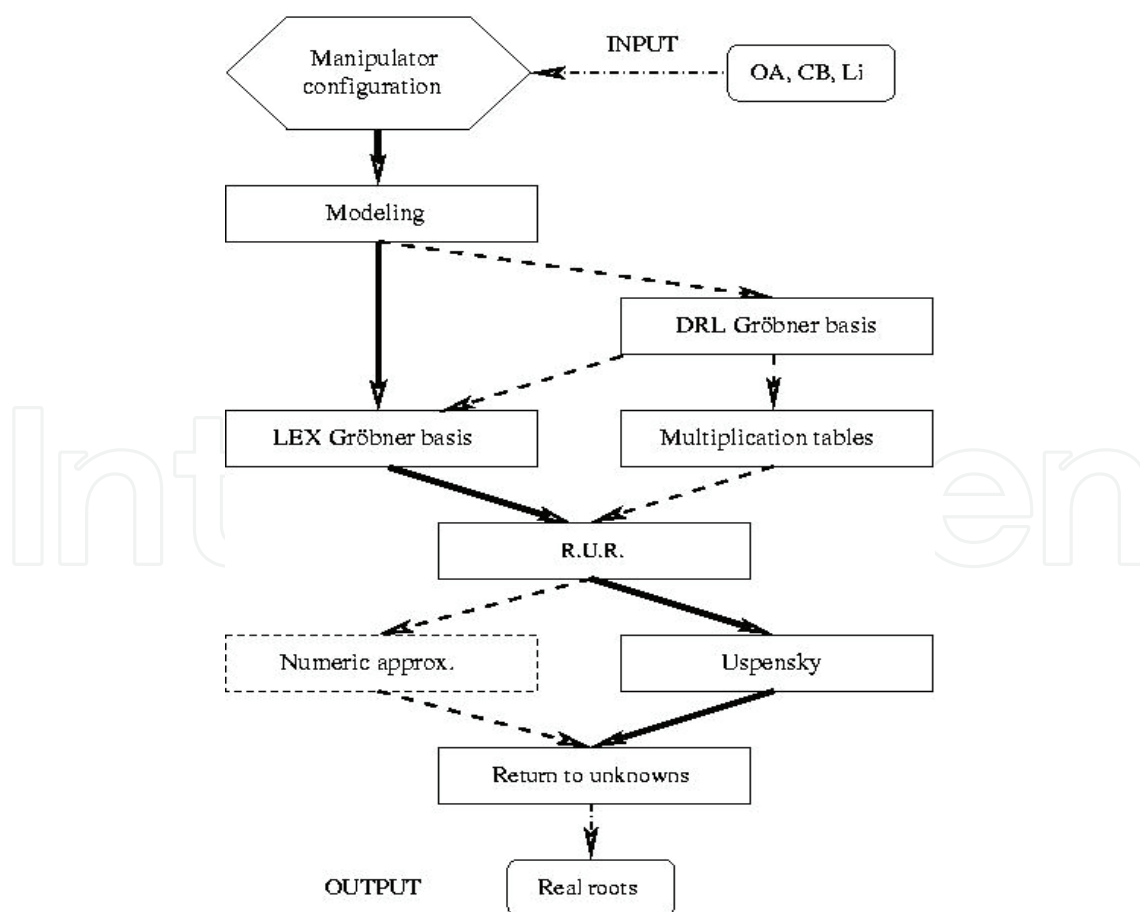


Fig. 9. Bloc diagram of the exact method

6. Solving the forward kinematics problem

6.1 Solving the general 6-6

The selected manipulator is a generic 6-6 in a realistic configuration, measured on a real parallel robot prototype, trying to reproduce a singularity-free theoretical SSM design. This example is a typical one among several successful trials which were performed to test the method without any failures. Even when the 6-6 FKP was in a singular pose, the method returned that the solution is a variety of degree one (infinite number of solutions), thus in singularity.

Computations were done on a personal computer equipped with a 400 MHz Pentium II microprocessor including 128 bytes of random access memory. The operating system was Red Hat LINUX. Thus, this conservative approach allows the user to expect better performance.

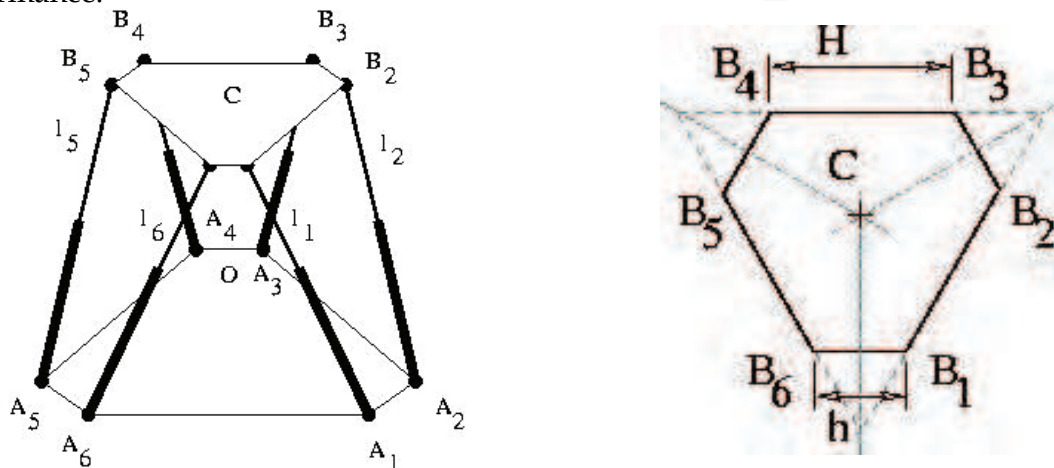


Fig. 10. The selected 6-6 hexapod manipulator

6.1.1 Modeling the 6-6 using the three point model

The parallel manipulator corresponding to the configuration data is shown in fig. 10. Lets take a typical 6-6 configuration example where construction realism introduce manipulator configuration deformations. It is then written in a configuration text file which includes the manipulator essential parameters: the coordinates of the joint center positions $OA_{i|R_f}$ in the fixed base reference frame R_f and the coordinates of the joint center positions $CB_{i|R_m}$ in the mobile platform reference frame R_m . In the computations, we use their simplified format, respectively identified as A and B. The unit is the millimeter:

```
B := [ [ 68410/1000, 393588/1000, 236459/1000 ],
        [ 375094/1000, -137623/1000, 236456/1000 ],
        [ 306664/1000, -256012/1000, 236461/1000 ],
        [ -306664/1000, -255912/1000, 236342/1000 ],
        [ -375057/1000, -137509/1000, 236464/1000 ],
        [ -68228/1000, 393620/1000, 236400/1000 ] ]:
A := [ [ 464141/1000, 389512/1000, -178804/1000 ],
        [ 569471/1000, 207131/1000, -178791/1000 ],
        [ 1052905/10000, -597151/1000, -178741/1000 ],
        [ -1052905/10000, -597200/1000, -178601/1000 ],
        [ -569744/1000, 206972/1000, -178460/1000 ],
        [ -464454/1000, 389384/1000, -178441/1000 ] ]:
Li := [(1250^2)\$i=1..6]:
```

6.2 The ten formulation comparison

Solving the ten **FKP** proposed formulations has been tried on the aforementioned example. Performance results are displayed on table 1 where the total real root computation times along with model code and name are given. This table specifies if the method terminated or was interrupted by the user if computations lasted more than one hour.

Five **FKP** formulations led to computation termination. However, only the AFD4, AFD5, AFP2 and AFP3 models brought response times within an hour. For the AFD1 formulation, only the *DRL Gröbner Basis* was successfully computed. Therefore, only the three following models were retained:

- AFD4 - Formulation with the translation and *Gröbner Basis* of the rotation matrix,
- AFD5 - Formulation with Translation and Quaternion variables,
- AFP3 - The three point model with constraints and function recombination.

Code	Method	Resp. time seconds	Ending	Comment
AFD1	trigo. ident.	148	FGb yes	stopped
AFD2	trigo relations	3600+	no	stopped
AFD3	\mathcal{R}	3600	yes	
AFD4	GB(\mathcal{R})	4,8	yes	
AFD5	quaternion	59,6	yes	
AFD6	somas	3600	no	stopped
AFP1	3-points+dist	3600+	no	stopped
AFP2	3pts+dist+normal	125,0	yes	
AFP3	3pts+modified	10,9	yes	
AFP4	6-points	3600+	no	stopped

Table 1. Response times for each formulations

Table 2 shows information which can be deduced from system observation. It gives the variable number, equation number and the maximum degree with respect to one variable, file size (*ASCII* size in bytes) and the maximum number of digits in one coefficient.

Code	Method	Variable number	Equations number	Size bytes
AFD4	GB on \mathcal{R}	12	30	7100
AFD5	Quaternion	7	7	6100
AFP3	3 pts & recomb	9	9	4500

Table 2. The selected formulation characteristics

6.3 System model analysis and selection

Solving more examples was carried using the three selected formulations in order to evaluate computations. Performance is studied on the *Gröbner Basis* being the largest computed mathematical object. We compare the resulting file sizes and the computation response times. The computations are performed on the preceding theoretical **SSM**, identified *theo*, and on the corresponding real general **6-6**, identified *eff*.

Robot + model	Strategy tgrob (s)	Accel (s)	F4 (s)	Gröbner Basis Size (Kbytes)	Complex Solution	Comment
theo_AFP3.fgb	29,06	6,53	0,6	130	36	stopped
theo_AFD5.fgb	124,29	36,24	2,45	150	72	
theo_AFD4.fgb	28,22	3600	0,31	85	36	
eff_AFP3.fgb	NA	1200	33,0	800	40	
eff_AFD5.fgb	NA	NA	31,9	790	80	
eff_AFD4.fgb	NA	NA	38,9	768	40	

Table 3. DRL Gröbner basis computation times and memory usage

As input, the manipulator configuration data include coefficients which are rational numbers with a six digit numerator and the number 1000 as denominator. Thus, the manipulator is measured at the micron accuracy. Let's note that computations are proceeding with the *RR* form.

6.4 DRL Gröbner Basis computations

Three algorithms shall be applied, namely *tgrob*, *Accel* and *F4* respectively the Buchberger, the *Gb* and the *FGb* implementations, [Faugère 1999]). Table 3 presents the computations times in seconds and memory space in kilobytes for the various techniques to compute a DRL Gröbner Basis for the three selected formulations.

The response time discrepancy between the **SSM** and **6-6** is significant. Practically, a small even infinitesimal coefficient difference such as platform non-planarity leads to response times which can be either dramatically longer (15, 55 or 100 times). The *Accel* algorithm is more sensible to formulation than *F4*. The *Accel* algorithm produces lengthy computations using the *AFD4* model. The quaternion formulation produces contradicting results, since it leads to significantly longer response times for the **SSM**, but slightly faster ones for the **6-6**. For the **6-6**, the response times and file sizes are almost equivalent.

Applying the fast *F4* algorithm, the rotation matrix model is preferred since one part of the system equations (13) does not depend on the studied configuration.

6.5 Rational univariate representation computations

The computed *DRL Gröbner Basis* is provided as the input to the *Rational Univariate Representation* computation algorithms. Knowing that the *Rational Univariate Representation* computation behaviour from the three formulation Gröbner bases cannot be predicted and lead to comparable *Gröbner Basis* sizes, the three *Rational Univariate Representation* computations have been performed. The following figures come from the direct calculations of the *RR* form using Faugère's *F4* algorithm for the lexicographic order followed by a division leading to a *Rational Univariate Representation* as the end-result. This strategy is compared with the original one which computes the *Rational Univariate Representation* by an order change (*F4+FGLM*). Table 4 gives the response times in seconds, memory space usage in kbytes, the number of complex and real roots.

Name	F4+FGLM (s)	RR (s)	File (Kbytes)	Complex #	Real #
theo_AFP3.rur	NA	2,95	210	36	8
theo_AFD5.rur	NA	NA	NA	72	16
theo_AFD4.rur	NA	3,31	270	36	8
eff_AFP3.rur	36,0	10,9	320	40	8
eff_AFD5.rur	3600	59,6	290	80	16
eff_AFD4.rur	17,9	4,8	300	40	8

Table 4. Rational Univariate Representation response times from the RR form

Problems were encountered with the quaternion based formulation to solve the **SSM**. This proves that even simpler hexapod configurations can be challenging. For the real 6-6 manipulator, the RR form calculation is always faster than the computation with an order change. Inasmuch, our results indicate that the RR Rational Univariate Representation calculation using the AFD4 rotation matrix formulation is the most effective one. For the quaternion formulation, two solutions express the same manipulator posture, this is confirmed by the obtained univariate polynomial which has a total degree 80 and the number of real solutions also doubled. This behavior is well known in practice since the redundant nature of quaternions leads to two results for the same pose. Finally, the direct computation of a *lexicographic Gröbner Basis* is the right choice when using recent algorithms such as F4. Many instantiations were tested on various manipulator configurations and all FKP systems were in **Shape Position** and the RR form could be computed.

6.6 FKP of typical hexapod manipulators

6.6.1 Hexapod architecture descriptions

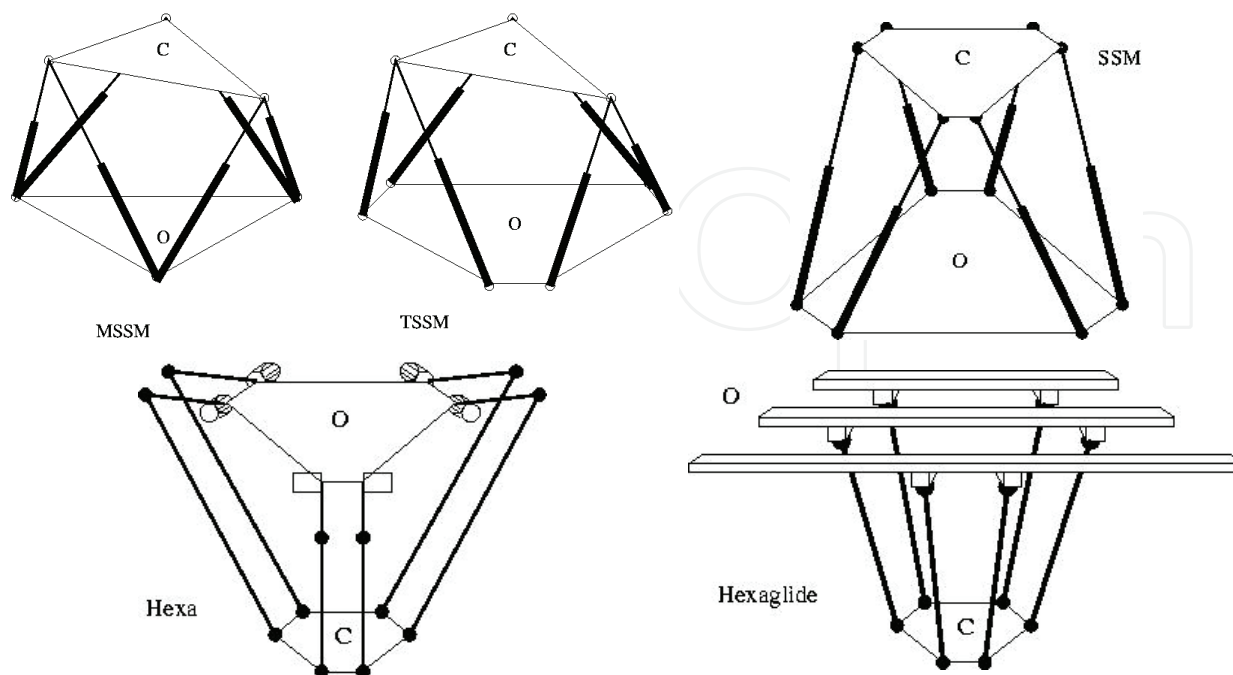


Fig. 11. The various hexapod alternatives

Although the method has been shown to solve the **6-6** and **SSM** manipulator **FKP**, it is relevant to compare results for various hexapods which feature specific and related geometric properties.

Several well-known parallel manipulators feature theoretical architectures, fig. 11. In order to clearly examine the configuration change impact, we have selected the first five hexapods as the slightly modified versions of each other, from the theoretical **MSSM** until the realistic **6-6**. The **6-6p** design comprises the joint positions lying on the same plane for either the fixed base or mobile platform. The **6-6pp** design is characterized by the two platforms being planar. The **SSM** design repeats the **6-6pp** along with planar hexagonal platforms which are made by equally truncating an equilateral triangular platform, [Nanua et al. 1990]. The **TSSM** manipulators are designed by merging the ball or Cardan joints in pairs to obtain a truly triangular mobile platform. The **MSSM** is merging the ball or Cardan joints of the fixed base by pairs to obtain an octahedron. These theoretical design **FKP** are computed and their kinematics results compared with the general **6-6** case. For each manipulator, the selected configuration and joint variables of each manipulator are selected to correspond.

6.6.2 FKP solving results

On table 5, the results are including the manipulator type, the manipulator morphology, the base and platform morphology where *T*, *S*, *P* and *NP* stands respectively for triangular, truncated triangular, planar and not planar, the complete real root computation time, the number of complex roots (the univariate polynomial degree), the number of real roots and the *Gröbner Basis* size. Only the *Gröbner Basis* sizes are compared, since the *Gröbner Basis* is the largest mathematical object calculated by method.

Robot type	Mechanism config	Base	Platform	Time (s)	Complex sols #	Real sols #	Gröbner (kbytes)
MSSM	octahedron	T,P	T,P	0,07	16	16	60
TSSM	hexapod	S,P	T,P	0,08	16	16	76
SSM	(6)	S,P	S,P	0,67	36	16	238
6-6pp	(6)	P	P	1,1	40	16	390
6-6p	(6)	NP	P	1,8	40	16	308
6-6	(6)	NP	NP	10,4	40	16	402
DIET	(6)	NP	NP	9,9	40	40	392
Hexa	6R-6R-6R	P	S,P	2,0	40	8	346
Hexaglide	6T-6R-6R	P	S,P	0,5	36	8	180

Table 5. Hexapod **FKP** overall results and performances

6.6.3 Discussion on the results

The gradual passage to geometrically simpler parallel structures leads to significantly shorter response times. In the **6-6** case, the introduction of one planar platform reduces computation times by 20. Inasmuch, the passage from a theoretical **SSM** to a realistic **6-6** leads to computation times which are 40 times longer.

With model simplification, it is notable that the real solution number is maintained. One could conjecture homothopy in those cases. The number of complex roots varies when the manipulator type changes; the **MSSM** and **TSSM** only have 16 solutions whereas the **SSM** has 36. This last result raises an important classification issue which was not formerly identified, [Faugère and Lazard 1995, Merlet 1997]. The **SSM** manipulator does not go in the class of **6-6** manipulators and becomes a class by itself.

The table second section comprises **FKP** solving results about the hexapod obtained with the Dietmaier's method, [Dietmaier 1998], the **Hexa** and the **Hexaglide** with a **SSM** type mobile platform. In all instances, the **Hexa**, [Pierrot et al. 1991], and **Hexaglide**, [Hebsacker 1998] can feature either 36 or 40 complex solutions, this only depends on the mobile platform configuration. In fact, when the truncated equilateral triangle is used, then the **FKP** yield 36 solutions.

Knowing that the **6-6pp** and **6-6p** lead to 40 complex solutions, the new **SSM** class is characterized not only by its complex solution number but by the mobile platform and fixed base peculiar type.

Therefore, the manipulators which fall into the **SSM** category are the ones which feature one specific truncated platform.

Although Dietmaier's **6-6 FKP** yields the largest number of real solutions, it does not necessarily lead to the longest computation times and largest result files.

Finally, various tests were performed where leg lengths were changed such as moving the robot on a straight line or circles with the same **6-6** manipulator configuration. The real root numbers have all been an even number in the set {4,8,12,16} depending on the location inside the workspace. This could be considered a conjecture. The only case where only one real solution has been found is when a theoretical **6-6pp** has its actuator values bringing the manipulator mobile platform to lie on the fixed base plane. This solution corresponds to two real coincident roots and this case can be identified as a singularity with the loss of three *DOF* since the manipulator can then only move in a plane.

6.7 Assembly mode analyses

The exact method allows addressing the question of assembly modes. This problem is also referred to as posture analysis. Assembly modes are defined as follows:

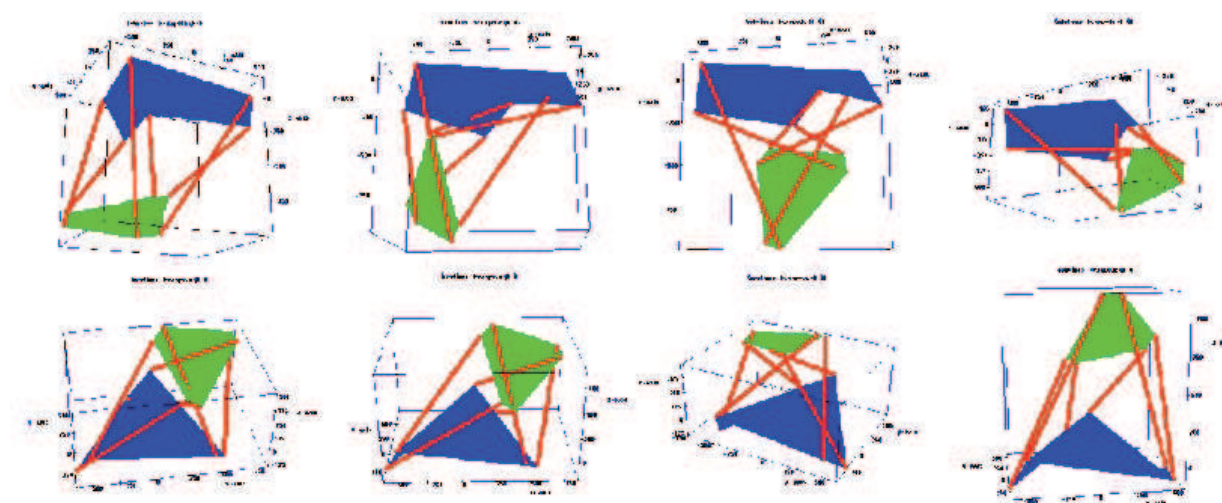


Fig. 12. Posture analysis for the general **6-6** manipulator

Definition 6.1 Given a manipulator configuration where the fixed base, the mobile platform and kinematics chain lengths are specified, for a set of active joint positions, determine all the possible geometric assemblies for the selected manipulator.

For the selected example, eight real solutions were obtained which geometrically represent the only possible assembly modes. These modes are then drafted using the *XMuPAD* environment, fig. 12. The position based models lead to root results which are directly usable to draw the effective postures. This exemplifies that posture analysis is feasible for any manipulator which can be modelled as a general 6-6 hexapod.

7. Conclusion

In this chapter, one complete exact method to solve the parallel manipulator **FKP** has been explained. The method was applied to the 6-6 general parallel manipulator, which is recognized to yield the most difficult problem, and also to various other manipulators such as the **SSM**, **TSSM**, **MSSM**, **Hexa** and **Hexaglide**.

Moreover, the modeling of the **FKP** was investigated. Six displacement based models and two position-based models were derived for the 6-6 general parallel manipulator.

One complete algebraic method to solve the Forward Kinematics Problem was applied. Although many methods can find solutions to some of these **FKP** systems, the proposed algebraic exact method insures the exactness of the real solution results, since it is based on one *Gröbner Basis*, which completely describes the ideal related to the original system. From this basis, one computes the *Rational Univariate Representation* including one univariate equation for root isolation. The selected algorithms always succeeded to solve any parallel robot **FKP** in all tested non-singular instances.

The selected manipulator was a typical 6-6 hexapod known as the Gough platform in a calibrated configuration, measured on a real parallel robot prototype constructed from a theoretically singularity free **SSM** design.

The 8 polynomial formulations were implemented and compared. We identified three models that allowed for computation termination, out of which two were retained since their computations occur with acceptable performances: a displacement based formulation with the rotation matrix *Gröbner Basis* with end-effector position and rotation matrix parameters as variables and a formulation with three points on a platform.

Solving typical posture examples, the *Rational Univariate Representation* comprised a univariate equation of degree 40 and 8 to 12 real solutions were computed depending on the position in the workspace. The total computation averaged 130 seconds on a relatively old computer. On a faster computer, the response time falls to less than one second. Hence, this method can be suitable for small-scale trajectory pursuit applications.

This result is very important since any *Gröbner Basis* completely describes the mathematical object related to the original system. From this basis, one can try to build an exact equivalent system with the original one including one univariate equation. Up to author's knowledge, this is actually the only known method that is certified to establish a truly equivalent system that preserves the properties.

Further testing led to favor the first formulation, since it yields slightly faster computations and it gives directly the end-effector position. The quaternion-based models can lead to difficulties in the case of simpler configurations and thus longer computation times, since it is doubling the solution number, which is explained by the *Rational Univariate Representation*

equations having a degree twice as large as the others. Moreover, one final advantage is that the displacement-based equations can be applied on any manipulator mobile platform.

8. Acknowledgment

I would like to thank my wife Clotilde for the time spent on rewriting and correcting the book chapter in Word.

9. References

- Alonso, M.-E.; Becker, E.; Roy M.F. & Woermann T. (1996) Multiplicities and idempotents for zerodimensional systems. In *Algorithms in Algebraic Geometry and Applications*, Vol. 143, Progress in Mathematics, pages 1--20.
- Buchberger, B. & Loos, R. (1982) Algebraic Simplification. In *Computer Algebra-Symbolic and Algebraic Computation*. SpringerVerlag, Vienna.
- Buchberger B. (1985) Gröbner bases: An Algorithmic Method in Polynomial Ideal Theory. In *Multidimensional Systems Theory - Progress, Directions and Open Problems in Multidimensional Systems*, N.K. Bose (e.d.) Reidel Publishing Company, Dordrecht, pp.184-232.
- Bruyninckx, H. & DeSchutter, J. (1996) A class of fully parallel manipulators with closed-form forward position kinematics. In *Advances in Robot Kinematics*, pages 411--420.
- Cox, D.; Little, J. & O'Shea D. (1992) *Ideals, varieties, and algorithms an introduction to computational algebraic geometry and commutative algebra*. Undergraduate texts in mathematics. SpringerVerlag, New York.
- Dedieu, J.P. & Norton, G.H. (1990) Stewart varieties: a direct algebraic method for stewart platforms. In *Proceedings of SigSam*, volume 244, pages 42--59.
- P. Dietmaier. (1998) The Stewart-Gough platform of general geometry can have 40 real postures. In *Advances in Robot Kinematics*, pages 7--16.
- Dieudonné, E.; Parrish, R. & Bardusch, R. (1972) An actuator extension transformation for a motion simulator and an inverse transformation applying Newton-Raphson's method. Technical report D7067, NASA, Washington.
- Didrit, O.; Petitot, M. & Walter, E. (1998) Guaranteed solution of direct kinematics problems for general configurations of parallel manipulators. *IEEE Transactions on Robotics and Automation*, Vol. 14, No. 2, pages 259--265.
- Egner, S. (1996) Semi-numerical solution to 6/6-stewart-platform kinematics based on symmetry. In *Applicable Algebra in Engineering, Communication and Computing*, Vol. 7, No. 6, pages 449--468.
- Faugère, J.-C.; Gianni, P.; Lazard, D. & Mora, T. (1991) Efficient computation of zero-dimensional Gröbner basis by change of ordering. *Journal of Symbolic Computation*, Vol. 16, No. 4, pages 329--344.
- Faugère, J.C. & D. Lazard. (1995) The combinatorial classes of parallel manipulators. *Mechanism and Machine Theory*, Vol. 30, No. 6, pages 765--776.
- Faugère, J.C. (1999) A new efficient algorithm for computing Gröbner bases (f4). *J. of Pure and Applied Algebra*, Vol. 139, No. 13, pages 61--88.
- Fischer, P.J. & Daniel, R.W. (1992) Real time kinematics for a 6 dof telerobotic joystick. In *Proceedings of RoManSy 9*, Udine, pages 292--300.

- Geddes, K.; Czapor, S. & Labahn, G. (1994) *Algorithms for computer algebra*. Kluwer Academic Publishers, Nonwell.
- Gosselin, C. & Angeles, J. (1988) The optimum kinematic design of a planar three dof parallel manipulator. *J. of Mechanisms, Transmissions and Automation in Design*, Vol. 110, pages 35--41.
- Gosselin, C.; Sefrioui, J. & Richard, M.J. (1994) On the direct kinematics of spherical three dof parallel manipulators with coplanar platform. *J. of Mechanical Design*, Vol. 116, pages 587--593, June 1994.
- Griffis, M. & Duffy, J. (1989) A forward displacement analysis of a class of Stewart platform. *J. of Robotic Systems*, Vol. 6, No. 6, pages 703--720.
- Hebsacker, M. (1998) Parallel werkzeugmaschinenkinematik. In *Proceedings of IPK 98, Internationales ParallelkinematikKolloquium*, Zürich, pages 21--32.
- Hunt, K.H. (1983) Structural kinematics of inparallelactuated robotarms. *J. of Mechanisms, Transmissions and Automation in Design*, Vol. 105, pages 705--712.
- Husty, M. (1996) An algorithm for solving the direct kinematic of Stewart-Gough type platforms. *J. of Mechanism and Machine Theory*, Vol. 31, No. 4, pages 365--379, 1996.
- Innocenti, C. & ParentiCastelli, V. (1990) Direct position analysis of the Stewart platform mechanism. *Mechanism and Machine Theory*, Vol. 25, No. 6, pages 611-- 621.
- Kohli, D.; Dhingra, A. & Xu, Y.X. (1992) Direct kinematics of general Stewart platforms. In *Proceedings of ASME Conference on Robotics, Spatial Mechanisms and Mechanical Systems*, Vol. 45, pages 107--112.
- Lazard, D. (1992) Solving zerodimensional algebraic systems. *J. of Symbolic Computation*, Vol. 13, pages 117--131.
- Lazard, D. (1992) Stewart platforms and Gröbner basis. In *Proceedings of Advances in Robotics Kinematics*, pages 136--142, Ferrare, September 1992.
- Lazard, D. (1993) On the representation of rigidbody motions and its application to generalized platform manipulators. *J. of Computational Kinematics*, Vol. 1, No. 1, pages 175--182.
- Merlet, J.-P. (1987) Parallel manipulators, part1: Theory; design, kinematics, dynamics and control. Technical report 646, INRIA, SophiaAntipolis.
- Merlet, J.-P. (1994) Parallel manipulators: state of the art and perspectives. *J. of Advanced Robotics*, Vol. 8, No. 6, pages 589--596, 1994.
- Merlet, J.-P. (1997) *Les Robots parallèles*. Série Robotique. Hermès, Paris, second edition, traité des nouvelles technologies edition, 1997.
- Merlet, J.-P. (2004) Solving the forward kinematics of a Gough type parallel manipulator with interval analysis. *The International Journal of Robotics Research*, Vol. 23, No. 3, pages 221--235.
- Mourrain, B. (1993) The 40 generic positions of a parallel robot. In *proceedings of ISSAC'93*, Kiev, pages 173--182.
- Mourrain, B. (1993) About the rational map associated to a parallel robot. Technical report 2141, INRIA, SophiaAntipolis, November 1993.
- Murray, P.; et al. (1997) A planar quaternion approach to the kinematics synthesis of a parallel manipulator. *Robotica*, Vol. 15, pages 360--365.
- Nanua, P.; Waldron, K. T & Murthy, V. (1990) Direct kinematic solution of a Stewart platform. In *IEEE transactions on Robotics and Automation*, Vol. 6, No.4, pages 438-444.

- ParentiCastelli, V. & Innocenti, C. (1990) Forward displacement analysis of parallel mechanisms: closedform solution of PRR3s and PPR3s structures. In *Proceedings of the ASME 21th Biennial Mechanisms Conf.*, Chicago, pages 263--269.
- Patel, A. & Ehmann, K. (1997) Volumetric error analysis of a Stewart platform based machine tool. In *Annals of the CIRP*, Vol. 46, pages 287--290.
- Petuya, V.; Alonso, A.; Altazurra, O. & Hernandez, A. (2005) Resolution of the direct position problem of the parallel kinematic platforms using the geometric iterative method. In *EEE Intern. Conf. on Robotics and Automation*, Barcelona, pages 3255--3260.
- Pierrot, F.; Dauchez, F. & Fournier, A. (1991) Hexa: a fast six dof fully parallel robot. In *Proceedings of the ICAR Conference*, Pisa, pages 1159--1163.
- Primrose, E.J.F. & Freudenstein, F. (1969) Spatial motions. part 1: Point paths of mechanisms with four or fewer links. *ASME J. of engineering for industry*, Vol. 91, No. 1, pages 103-- 114.
- Raghavan, M. (1993) The stewart platform of general geometry has 40 configurations. *ASME Trans. of Mech. Design*, Vol. 115, No. 2, pages 277--282.
- Raghavan, M. & Roth, B. (1995) Solving polynomial systems for the kinematic analysis and synthesis of mechanisms and robot manipulators. *Transactions of the ASME*, Vol. 117, pages 71--79.
- Rolland, L. (2003) Outils algébriques pour la résolution de problèmes géométriques et l'analyse de trajectoire de robots parallèles prévus pour des applications à haute cadence et grande précision. PhD thesis, Université Henri Poincaré, Nancy 1, December 2003.
- Rolland, L. (2005) Certified solving of the forward kinematics problem with an exact method for the general parallel manipulator. *Advanced Robotics*, Vol. 19, No. 9, pages 995--1025.
- Rolland, L. (2006) Synthesis on the forward kinematics problem algebraic modeling for the planar parallel manipulator. Displacement-based equation systems. *Advanced Robotics*, Vol. 20, No. 9, pages 1035--1065.
- Rolland, L. (2007) Synthesis on the forward kinematics problem algebraic modeling for the spatial parallel manipulator. Displacement-based equation systems. *Advanced Robotics*, Vol. 21, No. 9, 32 pages 1071--1092.
- Ronga, F. & Vust, T. (1992) Stewart platforms without computer ? In *Proc. of the Intern. Conf. of real, analytic and algebraic Geometry*, Trento, pages 197--212.
- Rouillier, F. (1999) Solving zerodimensional systems through the rational univariate representation. *Journal of Applicable Algebra in Engineering, Communication and Computing*, Vol. 9, NO. 5, pages 433--461.
- Rouillier, F. & Zimmermann, P. (2001) Efficient isolation of a polynomial real roots. Technical report RR4113, INRIA.
- Sreenivasan, S.V. & Nanua, P. (1992) Solution of the direct position kinematics problem of the general stewart platform using advanced polynomial continuation. In *22nd Biennial Mechanisms Conf.*, Scottsdale, pages 99--106.
- Sreenivasan, S.V.; Waldron, K.J. & Nanua, P. (1994) Direct displacement analysis of a 6-6 stewart platform. *Mechanism and Machine Theory*, Vol. 29, No. 6, pages 855--864.

- Sugimoto, K. (1987) Kinematic and dynamic analysis of parallel manipulators by means of motor algebra. *J. of Mechanisms, Transmissions and Automation in Design*, Vol. 109: pages 3--7, 1987.
- Tsai, L.W. & Morgan, A.P. (1984) Solving the kinematics of the most general 6 and 5 dof manipulators by continuation methods. *ASME J. of Mechanisms, Transmissions and Automation in Design*, Vol. 107, pages 189--200.
- Vischer, P. (1996) Improving the accuracy of parallel robots. PhD thesis, Ecole Polytechnique Fédérale de Lausanne.
- Wampler, C.W. (1996) Forward displacement analysis of general six-in-parallel SPS (Stewart) platform manipulators using soma coordinates. *Mechanism and Machine Theory*, Vol. 31, NO. 3, pages 33--337.

IntechOpen



Parallel Manipulators, towards New Applications

Edited by Huapeng Wu

ISBN 978-3-902613-40-0

Hard cover, 506 pages

Publisher I-Tech Education and Publishing

Published online 01, April, 2008

Published in print edition April, 2008

In recent years, parallel kinematics mechanisms have attracted a lot of attention from the academic and industrial communities due to potential applications not only as robot manipulators but also as machine tools. Generally, the criteria used to compare the performance of traditional serial robots and parallel robots are the workspace, the ratio between the payload and the robot mass, accuracy, and dynamic behaviour. In addition to the reduced coupling effect between joints, parallel robots bring the benefits of much higher payload-robot mass ratios, superior accuracy and greater stiffness; qualities which lead to better dynamic performance. The main drawback with parallel robots is the relatively small workspace. A great deal of research on parallel robots has been carried out worldwide, and a large number of parallel mechanism systems have been built for various applications, such as remote handling, machine tools, medical robots, simulators, micro-robots, and humanoid robots. This book opens a window to exceptional research and development work on parallel mechanisms contributed by authors from around the world. Through this window the reader can get a good view of current parallel robot research and applications.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Luc Rolland (2008). Certified Solving and Synthesis on Modeling of the Kinematics. Problems of Gough-Type Parallel Manipulators with an Exact Algebraic Method, *Parallel Manipulators, towards New Applications*, Huapeng Wu (Ed.), ISBN: 978-3-902613-40-0, InTech, Available from:
http://www.intechopen.com/books/parallel_manipulators_towards_new_applications/certified_solving_and_synthesis_on_modeling_of_the_kinematics__problems_of_gough-type_parallel_manip

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2008 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen