Contributions to On–Line Robot Kinematics

by

Murat Tandirci

B. Eng. (McGill University), 1987

Department of Mechanical Engineering

McGill University, Montreal

Canada

A thesis submitted to the Faculty of Graduate Studies and Research

in partial fulfillment of the requirements for the degree of

Masters of Engineering

November 26, 1991

© Murat Tandirci

Abstract

This thesis presents various results associated with forward and inverse kinematics of serial six-axis robotic manipulators in continuous path applications. The methods and algorithms thus derived are implemented on Unix-based workstations in the *C* language, in a package called *CINVERSE.*

The orientation expression of the end–effector is described using natural, linear and quadratic invariants, the latter better known as Euler parameters. The linear and quadratic invariants can be derived by either multiplying the rotation matrices or by vector operations which make use of invariant compositions.

Since the solution of inverse kinematics methods involve numerical iterations, an initial guess is first derived, based on the kinematic architecture of the robot, this guess being denoted as the *home configuration* of the robot. The home configuration is defined, in turn, as that entailing the Jacobian with the minimum condition number. Since both positioning and orientation tasks are considered, we begin by defining a *characteristic length* of the robot that allows us to render the translational kinematics equations in nondimensional form, thereby deriving a dimensionally homogeneous Jacobian. Next, a *characteristic point* is defined as a point of the end–effector that minimizes the said condition number.

Kinematic equations of rotation are derived using linear invariants, quadratic invariants and natural invariants. Although linear invariants allow a straightforward derivation of partial derivatives of the nonlinear kinematic equations with respect to the joint coordinates, they are prone to algebraic singularities. The numerical conditioning and operation counts of the three methods are analyzed and comparisons are made. A detailed study of spherical-wrist conditioning is included.

A complete inverse kinematics solution involves derivation of joint coordinates, joint rates and joint accelerations. In the presence of kinematic singularities, the joint rates and joint accelerations are derived by introducing a scheme that ensures that the manipulator will not jump to another branch while tracking a smooth trajectory.

Résumé

Cette thèse presente plusieurs resultats reliés à la cinématique directe et inverse des manipulateurs robotiques à six degres de liberté pour des applications en trajectoire continue. Les méthodes et algorithmes ainsi obtenus sont implantés sous-jacents sont implantés en langage *C* sur des stations de travail a système UNIX, dans un programme qui s'appelle *CINVERSE*.

L'orientation de l'organe terminal est décrite par des invariants naturels, linéaires et quadratiques, ces derniers étant mieux connus sous l'appellation de paramètres d'Euler. Les invariants linéaires et quadratiques peuvent être calculés soit par multiplication des matrices de rotation, soit par des opérations vectorielles utilisant les règles de compositions des invariants.

Etant donné que les méthodes de résolution de la cinematique inverse nécessitent des itérations numériques, on propose tout d'abord un estimé initial, basé sur l'architecture cinematique du robot. On appelle cet estimé la *configuration de départ*. Le nombre de condition de la matrice jacobienne de la configuration de départ se doit d'être un minimum. Comme on considère autant la position que l'orientation, on definit tout d'abord une *longueur caractéristique* pour le robot, ce qui nous permet de mettre les équations cinématiques de translation sous forme non-dimensionelle et d'en dériver une matrice jacobienne dimensionnellement homogène. Ensuite, on définit le *point caracteristique* comme un position de l'organe terminal minimisant ledit nombre de condition.

Les équations cinématiques de rotation sont obtenus en utilisant les méthodes d'invariants linéaires, quadratiques et naturelles. Bien que les invariants linéaires permettent une obtention aisée des dérivées partielles des équations cinématiques non-lineaires par rapport au coordonnées articulaires, ils souffrent des singularités algébriques. On analyse le numbre de condition numérique et le nombre d'opérations des trois methodes, et on les compare. Une étude très détaille du nombre de condition des poignets sphériques est également inclue.

Une solution complète de la cinématique inverse demande l'obtention des coordonnées, des vitesses et des accélérations articulaires. En présence de singularités cinématiques, les vitesses et les accélérations articulaires sont obtenues en introduisant une methode qui garan-

tit que le manipulateur est confiné à une seule branche de solutions, ce qui assure une trajectoire souple.

Acknowledgements

I am indebted to my research supervisor, Professor Jorge Angeles for his support and guidance. I would like to thank to all my colleagues for sharing their valuable knowledge with me. Special thanks are due to John Darcovich for his assistance in proofreading the thesis and also for his assistance in many computer-related issues. Many thanks to Stéphane Aubry for translating the abstract into French.

The research work reported here was made possible under NSERC (Natural Sciences and Engineering Research Council of Canada) Research Grants A4532, STR0100971 and EQP00-92729. The partial support of IRIS (Institute for Robotics and Intelligent Systems) under the C-3 Project, is highly acknowledged.

Contents

List of Tables

Chapter 1

Introduction

1.1 Definitions and Terminology

1.1.1 Robot Kinematics

Broadly speaking, kinematics is the geometrical study of the motion of bodies. This thesis is confined to the kinematics of open-chain, or serial-type manipulators that do not necessarily admit a closed-form solution. Furthermore, the manipulators studied consist of revolute and prismatic joints, enabling, respectively, rotation and translation about and along a specified axis.

The position and orientation of the end-effector (EE), are together referred to as the *pose* of a serial manipulator. In order to perform an arbitrary operation in 3D space, the robot must have six degrees of freedom. Three of these are needed for positioning the EE at a specified location with respect to the fixed frame, and another three are needed to orient it with respect to the same frame. *Wrist-partitioned manipulators* consist of a spherical wrist, which accounts for orienting the EE, and an arm accounting for positioning the EE.

Having six joints does not always guarantee six degrees of freedom. When two of the axes of the arm joints are collinear, Fig. 1.1(a), or when the three wrist axes are coplanar, Fig. 1.1(b), the robot will lose one of its degrees of freedom. These configurations are referred to as *kinematic singularities*, and they can happen naturally during the motion of the robot.

In order to study the position and orientation of the links with respect to each other, the

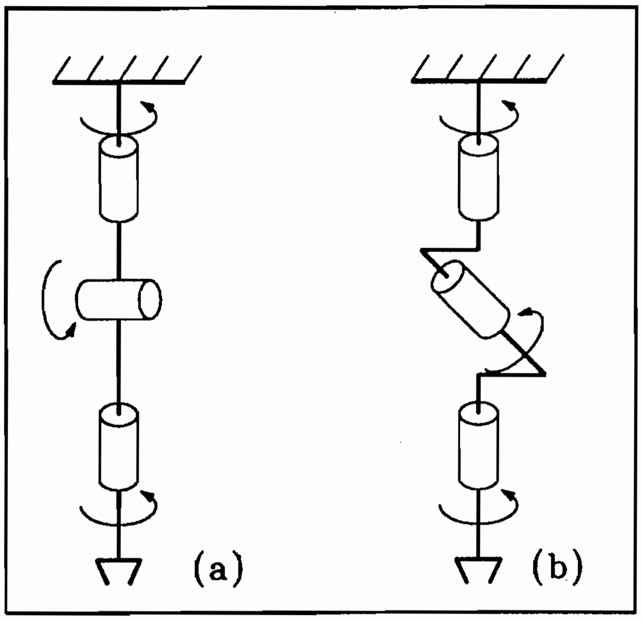


Figure 1.1: a) Two collinear axes b) Three coplanar axes

well-known Hartenberg-Denavit, (HD) parameters are used (Hartenberg and Denavit, 1964). In order to use this convention, we attach an orthogonal frame to each link as follows: The Z-axis of frame i (Z_i) is defined along the axis of rotation in the case of a revolute joint, or the axis of translation in the case of a prismatic joint. The origin of frame i is defined at the intersection of the Z_i and the common normal to Z_i and Z_{i-1}. The said common normal is referred to as the X-axis of frame i (X_i). After determining the orthogonal frames, the four HD parameters are specified, namely, α_i, a_i, b_i and θ_i. The HD parameters are best understood with the help of Fig. 1.2. For two arbitrary joints i and $i+1$, we have,

- α_i: angle from axis Z_i to axis Z_{i+1}, measured about axis X_{i+1}, using the right-hand rule.
- a_i: distance between axes Z_i and Z_{i+1}.
- b_i: signed distance between the X_i and X_{i+1} axes; it is positive if the intersection of axes Z_i and X_{i+1} lies on the positive side of axis Z_i; and is negative otherwise.
- θ_i: angle from axes X_i to X_{i+1} measured about axis Z_i using the right-hand rule.

For a revolute joint, parameters α_i, a_i and b_i are determined by design, i.e., they describe

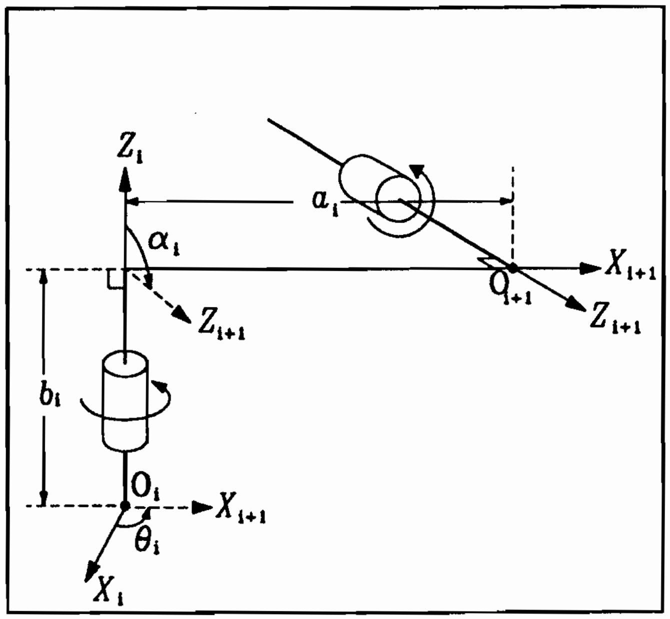


Figure 1.2: Hartenberg-Denavit parameters

the architecture of the robot, whereas, θ_i describes motion and is thus termed as the *joint-variable.* On the other hand, for a prismatic joint, α_i, a_i and θ_i are the design parameters, and b_i is the joint-variable.

Because of the freedom in choosing the direction of the axes Z_i and X_i, the numerical values of the HD parameters α_i, b_i and θ_i are not unique. A complete analysis of the effect of choosing opposite axes on these parameters is done in Appendix A.

A complete *inverse kinematics problem* (IKP) involves displacement, velocity and acceleration inverse kinematics. In *displacement inverse kinematics* (DIK) the position and orientation of the end-effector is given, and the joint-variables, $\{\theta_k\}_1^6$, attaining these specifications, are calculated. In *velocity inverse kinematics* (VIK), the time rate of change of the translational and angular velocity of the EE are given, and the joint-rates, $\{\dot{\theta}_k\}_1^6$, attaining these specifications, are calculated. Finally, in *acceleration inverse kinematics* (AIK), the translational and angular accelerations of the EE are given, and the joint-accelerations, $\{\ddot{\theta}_k\}_1^6$, attaining these specifications, are calculated.

spatial mechanisms consisting of seven links and seven revolutes. For the purposes of inverse kinematics, a closed-loop seven link manipulator can be considered equivalent to an open-loop six link manipulator (Angeles and Cyril, 1986). Thus, the above formulation is applicable to the analysis of general six-revolute manipulators.

The degree of the polynomial was reduced to 48 by Albala and Angeles (1979). Next the said degree was reduced to 32 almost simultaneously by Albala (1982) and Duffy and Crane (1980), by expressing the polynomial in the form of a 16×16 determinant equated to zero, all of whose entries are quadratic in the tangent of one-half a joint angle. Recently, Tsai and Morgan (1985) formulated the IKP in the form of four bilinear equations in the sines and cosines of four joint angles that were constrained by four quadratic equations. This method yielded up to sixteen real solutions, the authors thus conjecturing that the IKP can have at most 16 real solutions. Then, it was proven that a given six-axis manipulator would admit a maximum of 16 solutions (Primose, 1986), thus confirming that additional solutions suggested by previous methods were spurious.

Finally, Lee and Liang (1988) showed that a 16-degree univariate polynomial can be derived, without proposing a method for evaluating the polynomial coefficients. Raghavan and Roth (1990) provided an algorithm for the computation of the coefficients of the 16th order polynomial in the tangent of one-half of the joint angle, θ_3. Next, Lee (1990) used the same procedure but employed a different elimination technique deriving the coefficients of a 16th order polynomial in the tangent of one-half of the joint angle, θ_1.

A joint-variable elimination technique was recently advanced with the help of powerful symbolic manipulation software such as MATHEMATICA and MACSYMA (Chang, 1991; Williams, 1989). The use of such software allowed the non-linear system of displacement equations to be reduced to two equations in two variables. These equations yield corresponding contours in the space of two variables, their intersections thus producing the desired solutions. Moreover, Lloyd and Hayward (1988) used MACSYMA to find solution equations for simple manipulators by making use of known decouplings in the manipulator kinematics.

The second methodology is based on iterative procedures. Goldenberg, Benhabib and Fenton (1985) derived a system of six nonlinear equations for arbitrary, n-degree of freedom manipulators, and suggested a modified Newton-Raphson method for its solution. The six nonlinear equations constituted a six-dimensional vector containing a combination of orientation and translational entries. The gradient of the six dimensional vector function with respect to the joint angles is calculated, and a first-order approximation of the Taylor series expansion of the vector function is used. A new incremental vector of joint angles is obtained as a solution of the linear system derived from the aforementioned first-order approximation and used in the next iteration. The procedure is repeated until a norm of the vector function becomes smaller than a prescribed tolerance.

Angeles (1985) proposed, in turn, a method that formulates the IKP as an overdetermined system of seven equations with six unknowns, the system being solved using the Newton-Gauss method (Dahlquist and Björk, 1974).

Kazerounian (1987) also proposed an iterative algorithm based on the modified Newton-Raphson approach. The objective of the algorithm is to minimize a scalar function of deviation which is constructed by the sum of the squares of the entries of the difference matrix of orientation between the prescribed and current manipulator configurations. In addition, the sum of squares of the entries of the difference vector of position is also included in the scalar function. The gradient of the scalar function with respect to incremental joint angles are calculated, and a new vector of joint angles is sought that yields a smaller objective function. The procedure is repeated until the objective function becomes smaller than a prescribed tolerance.

Podhorodeski (1989) analyzed the inverse velocity problem with the help of screw theory. The screw quantities are sequentially calculated from their relations to associated wrenches, which in turn are related to force, length and time expressions. The joint rates are then resolved from the screw quantities through reciprocal products. The displacement inverse kinematics was formulated by defining a pose error, and the incremental joint angles were calculated through reciprocal products with wrench expressions, using a Newton-type procedure.

Lenarčič and Košutnik (1989) proposed a few methods to compute approximate solutions for the kinematic inversion of six-axis manipulators with arbitrary architectures. The authors separated the problem into two, in which approximate solutions were found by first solving for the first three joint coordinates by considering the position equations only. Next, the remaining joint coordinates were approximated by considering the orientation equations. The authors concluded that the approximation errors do not cause a significant problem in a number of applications, such as arc-welding.

Very recently, Wampler (1991) used unit vectors parallel to joint axes as design variables, and found a 12×12 sparse Jacobian matrix. The method is applicable to general 6-axis manipulators provided *twist angles* α_i ($i = 1, \ldots, 6$) are non-zero. Otherwise, common normal vectors $\mathbf{x}_i$ should be used as design variables (Wampler and Morgan, 1989). Compared to the dense 6×6 Jacobian, the proposed algorithm required comparable computation time if α_i, for $i = 1, \ldots, 6$, were non-zero. In the most general case, twice as many operations were required.

In this thesis, methods based on numerical procedures are studied. Below a brief comparison among the numerical and algebraic methods is made.

In path tracking, iterative procedures are attractive because the current solution is close to the previous one. Thus, the previous solution can be used as an initial guess, thereby allowing a quick convergence in a few iterations. Furthermore, iterative procedures converge to only one solution, and in path tracking only one solution is needed.

The disadvantage of the numerical procedures is that, for the first point on a given curve, an initial guess is not known. Moreover, this solution may heavily depend on the initial guess, so that, with different initial guesses, up to sixteen different solutions may be obtained. Moreover, since the numerical procedures converge to only one solution, no information is provided about the remaining solutions. For manipulator architectures allowing a closed-form solution, it is preferable to use a symbolic approach, since solutions can be obtained faster (Eppinger and Kreuzer, 1990) and, unlike numerical procedures, the solutions calculated symbolically are obtained in a predetermined amount of time.

1.2.2 Previous Work Related to Conditioning of Robotic Manipulators

As an attempt to measure the kinematic performance of manipulators, the *service angle* was defined as the range of joint angles allowing the EE to reach a specified point in space

(Vinogradev et al., 1971). Kumar and Waldron (1980) analyzed a subregion of the workspace in which the EE could attain any orientation. Yang and Lai (1985), on the other hand, investigated the properties of *service regions* that were defined in terms of service angles.

Salisbury and Craig (1982) introduced first the condition number of the Jacobian matrix as a measure of the kinematic performance of a manipulator. Next, manipulability was defined as a measure of kinematic performance by Yoshikawa (1983). Yoshikawa used the determinant of the product of the Jacobian with its transpose, that amounts to the absolute value of that determinant, as a manipulability measure, while Klein and Blaho (1987) related kinematic performance to the minimum singular value of the Jacobian.

As discussed in (Li, 1990), the condition number of the Jacobian, unlike the determinant, is point dependent, i.e., the condition number of the Jacobian depends on the point of the EE on which the Jacobian definition is based henceforth termed the *operation point.* On the contrary, the Jacobian determinant is independent of that point. However, from our experience we know that manipulability depends on the choice of the operation point. In fact, a common manipulator task is handwriting, in which the point of interest is the tip of the pencil. Our handwriting is highly influenced on the position of the pencil tip with respect to the finger tips.

In this thesis, we base our measure of kinematic performance on the condition number of the Jacobian, which allows us to define the characteristic point and the characteristic length of the manipulator.

1.3 Motivation

Inverse kinematics calculations are needed in order to determine the motion at the actuators of the robot, so that the EE will follow a prescribed trajectory with a prescribed velocity and acceleration. As discussed above, when the manipulator contains three intersecting axes, a closed form solution can be obtained. However, a solution method has to be implemented that will work in the most general case. The motion task can be specified to the IKP program either off-line or on-line, as desired. At the early stages of robotics technology, robots were only used in applications where the task was predetermined and repetitive, so that off-line IKP calculations were sufficient. However, as robotics technology advances, robots are required to understand their environment, and engage in tasks which depend on changes in a time-varying environment. Thus, the need for quick, on-line, IKP calculations becomes evident.

When solving the inverse kinematics of a robot iteratively, as proposed in (Angeles, 1985; Goldenberg, Benhabib and Fenton, 1985; Takano, 1985; Tsai and Morgan, 1985), a fast convergence to a solution is required for robot simulation and control in real time. For example, in an interactive computer animation program, the EE of the robot can be made to follow a specified trajectory using a graphical input, and the motion resulting from the change of joint-variables can be observed immediately. Moreover, quick inverse kinematics results are needed in telerobotics applications, where the operator may describe ongoing tasks based on the actual surroundings. Similarly, robots equipped with vision systems determine their tasks based on the information they gather about their environment; in such cases, quick and accurate inverse kinematics results are needed to perform the upcoming tasks.

Furthermore, when a numerical procedure is used to solve the inverse kinematics problem, an initial guess is needed. In displacement inverse kinematics, it is desired that the Jacobian be well conditioned, particularly upon starting the procedure. This motivates us to find the home configuration of the manipulator to be used as an initial guess in the numerical procedure. However, the Jacobian matrix contains entries with dimensions of length associated with translation, along with non-dimensional entries related to orientation. This motivates us to define the *characteristic length*, by which the translational entries are divided, thereby rendering the Jacobian dimensionally homogeneous while minimizing its condition number. Finally, the robot task planner has a certain freedom in the choice of the location of the end point anywhere in the EE of the manipulator. The location of this point influences the said condition number, which motivates us to determine the optimum point, that we term the *characteristic point* P_C, to minimize the condition number.

1.4 Thesis Contributions

The thesis contributions can be summarized as follows: First, two rotation representations are analyzed with operation counts, and the best method is found. Next, three numerical methods are outlined to solve the DIK, and comparisons are made on the basis of operation counts, convergence rate and conditioning of the Jacobian. Furthermore, velocity and acceleration inverse kinematics are reformulated to produce a faster solution. Next, the home configuration of general 6-axis manipulators, bearing the best conditioned Jacobian matrix is derived. In doing so, a characteristic length and a characteristic EE point are defined and determined. Finally, the conditioning of spherical wrists is analyzed, and it is shown that, for a given condition number, the orientation of the EE can be calculated using only two rotation expressions. By keeping the intermediate revolute fixed, a surface of constant condition number is then obtained.

All the above research contributions have been implemented in a C-language package, CINVERSE. The package is divided into three modules, namely, *TRAJ_PLAN*, *HOME_CONF* and *INV_KIN*, allowing for trajectory planning, home-configuration and inverse kinematics calculations, respectively.

1.5 Thesis Outline

Below we give an account of the overall thesis. Chapter 2 explains the formulation of forward and inverse kinematics. Efficient methods to represent a rotation are discussed, and comparisons are made on the basis of operation counts. Further in the chapter, the formulation of displacement, velocity and acceleration inverse kinematics is discussed, and a singularity treatment algorithm is presented. The displacement inverse kinematics problem is analyzed with three methods, and comparisons are made on the basis of operation count, CPU times, conditioning and observed convergence rate.

In Chapter 3, the determination of the home configuration is discussed. In addition, the characteristic length and the characteristic point of six-revolute manipulators are derived. The same methods are applied to three-axis planar and spherical manipulators. For the latter case, the isoconditioning surface is obtained and example loci are displayed.

In Chapter 4, applications of the CINVERSE package are discussed. A trajectory tracking example is taken, in which the data needed for displacement, velocity and acceleration inverse kinematics is derived with the TRAJ_PLAN module. Furthermore, four existing industrial robots are analyzed, and their home configuration, characteristic length and characteristic point are found using the HOME_CONF module. Finally, using an initial guess that is obtained from the HOME_CONF module, actual joint variables, joint rates and joint accelerations are calculated and plotted.

In Chapter 5, the author discusses the thesis conclusions, and suggests areas of improvements and possible future contributions to this field of study.

Chapter 2

Computational Analysis of Forward and Inverse Kinematics

2.1 Forward Kinematics

Forward Kinematics refers to the calculation of the orientation and position of the EE, given the joint angles. The well-known Hartenberg and Denavit notation (Hartenberg and Denavit, 1964) is used throughout.

For a general six-axis manipulator, the orientation of the EE in the base frame is expressed as the product of six rotation matrices, namely,

$$\mathbf{Q} = \mathbf{Q}_1\mathbf{Q}_2\mathbf{Q}_3\mathbf{Q}_4\mathbf{Q}_5\mathbf{Q}_6$$

where $\mathbf{Q}_i \equiv [\mathbf{Q}_i]_i$ denotes a rotation matrix expressing the orientation of the $(i+1)$st frame with respect to the ith frame, in ith-frame coordinates. Because the product matrix is orthogonal, only three out of its nine entries are independent. Different methods can be employed to express the rotation representation, as discussed in (Funda and Paul, 1989). Here, we use the *invariants* of the rotation matrix to express the orientation equations (Angeles, 1988). These quantities are preferred over Euler angles because they are invariant under a change of coordinate frame, and hence, less prone to singularities.

The EE position is readily derived, as indicated below, when the rotation matrices expressing the orientation of the EE in all frames are available:

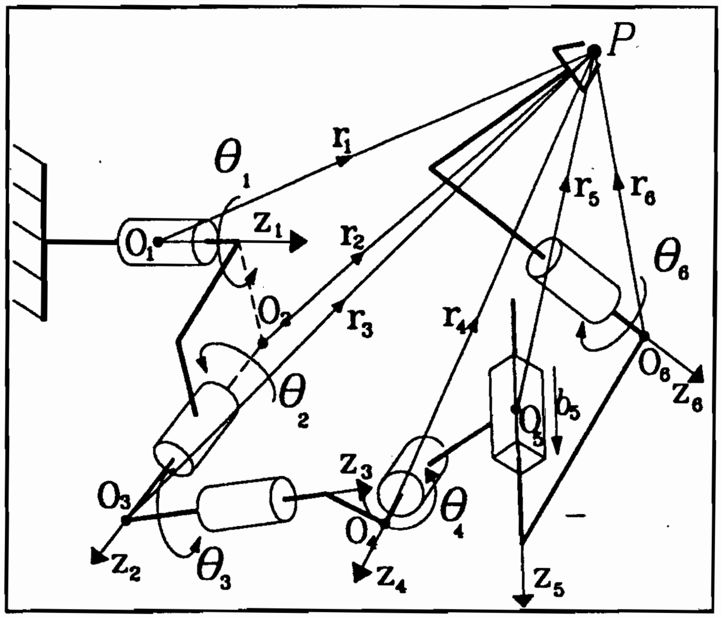


Figure 2.1: General 6-axis manipulator

$\mathbf{r}_6 \leftarrow \mathbf{a}_6$

```
For   i=5 to 1  do
```

$$\mathbf{r}_i \leftarrow \mathbf{a}_i + \mathbf{Q}_i\mathbf{r}_{i+1}$$

```
enddo
```

where $\mathbf{a}_i$ is the displacement vector from O_i to O_{i+1}, expressed in the ith frame, whereas $\mathbf{r}_i$ is the displacement vector from O_i to point P of the EE, expressed in the ith frame, $\mathbf{r}_1$ thus describing the desired EE position in the base frame (Fig. 2.1).

2.1.1 Linear Invariants

The *linear invariants* are linearly related to the the rotation matrix through vector and trace expressions, denoted as vect($\mathbf{Q}$) and tr($\mathbf{Q}$), respectively (Angeles, 1988),

$$\text{vect}(\mathbf{Q}) = \frac{1}{2}\begin{bmatrix} q_{32} - q_{23} \\ q_{13} - q_{31} \\ q_{21} - q_{12} \end{bmatrix}, \quad \text{tr}(\mathbf{Q}) = q_{11} + q_{22} + q_{33}$$

where q_{ij} represents the (i,j)th entry of $\mathbf{Q}$. The linear invariants, denoted as $\mathbf{q}$ and q_0, are now introduced as follows (Angeles, 1988):

$$\mathbf{q} \equiv \text{vect}(\mathbf{Q}) = \sin\phi\mathbf{u}, \qquad q_0 \equiv \frac{\text{tr}(\mathbf{Q}) - 1}{2} = \frac{q_{11} + q_{22} + q_{33} - 1}{2} = \cos\phi$$

where $\mathbf{u}$ is the unit vector parallel to the axis of rotation, and ϕ is the angle of rotation.

The above representation is a consequence of Euler's Theorem (Euler, 1775/1776a&b), which states that any rigid body motion leaving one point of the body fixed is equivalent to a single rotation, through an angle ϕ about an axis $\mathbf{u}$. In the formulation of displacement inverse kinematics (DIK), three quantities of the vector expression $\mathbf{q}$ and the scalar q_0 will be required to be equivalent to their prescribed counterparts. The scalar q_0 is not an independent quantity, but is introduced because of the ambiguity in the derivation of ϕ from $\sin\phi$. The dependence between the invariants can be expressed as follows:

$$\|\text{vect}(\mathbf{Q})\|^2 + \left(\frac{\text{tr}(\mathbf{Q}) - 1}{2}\right)^2 = 1$$

The two linear invariants were derived above from the the rotation matrix. The reverse can also be done, i.e., the matrix $\mathbf{Q}$ can be computed from the linear invariants, namely (Angeles, 1988),

$$\mathbf{Q} \equiv q_0\mathbf{1} + \frac{1}{1+q_0}\mathbf{q}\otimes\mathbf{q} + \mathbf{1}\times\mathbf{q} \tag{2.1}$$

The above relation is invalid when the angle of rotation is π, since $\mathbf{q}$ vanishes when $\sin\phi = 0$. However, $\mathbf{Q}$ can alternatively be obtained from the relation below, using the *natural invariants* of $\mathbf{Q}$ (Angeles, 1989):

$$\mathbf{Q} \equiv \mathbf{u}\otimes\mathbf{u} + \cos\phi(\mathbf{1} - \mathbf{u}\otimes\mathbf{u}) + \sin\phi\mathbf{1}\times\mathbf{q}$$

which becomes

$$\mathbf{Q} = -\mathbf{1} + 2\mathbf{u}\otimes\mathbf{u}$$

when $\phi = \pi$.

2.1.2 Quadratic Invariants

Quadratic invariants, better known as *Euler parameters*, are computed from the vector and trace of orthogonal square root of $\mathbf{Q}$, represented as $\sqrt{\mathbf{Q}}$. The said invariants are denoted below as $\hat{\mathbf{q}}$ and $\hat{q}_0$, namely,

$$\hat{\mathbf{q}} \equiv \text{vect}(\sqrt{\mathbf{Q}}) = \frac{1}{2}\begin{bmatrix}\hat{q}_{32} - \hat{q}_{23}\\ \hat{q}_{13} - \hat{q}_{31}\\ \hat{q}_{21} - \hat{q}_{12}\end{bmatrix} \qquad \hat{q}_0 \equiv \frac{\text{tr}(\sqrt{\mathbf{Q}}) - 1}{2} = \frac{\hat{q}_{11} + \hat{q}_{22} + \hat{q}_{33} - 1}{2}$$

where $\hat{q}_{ij}$ represents the (i,j)th entry of $\sqrt{\mathbf{Q}}$. Furthermore, the quadratic invariants can be expressed as,

$$\hat{\mathbf{q}} \equiv \sin(\frac{\phi}{2})\mathbf{u} \quad \hat{q}_0 \equiv \cos(\frac{\phi}{2})$$

Moreover, similar to eq.(2.1), the rotation matrix $\sqrt{\mathbf{Q}}$ can be computed from the quadratic invariants as (Angeles, 1988),

$$\sqrt{\mathbf{Q}} \equiv \hat{q}_0\mathbf{1} + \frac{1}{1+\hat{q}_0}\hat{\mathbf{q}} \otimes \hat{\mathbf{q}} + \mathbf{1} \times \hat{\mathbf{q}}$$

Unlike eq.(2.1), the above relation is always valid, since $\hat{\mathbf{q}}$ and $(1+\hat{q}_0)$ never vanish.

Derivation of the Quadratic Invariants from the Given Rotation Matrix

Because $\sqrt{\mathbf{Q}}$ is not readily available, alternate methods of computation of the quadratic invariants from the given rotation matrix $\mathbf{Q}$ are investigated.

First, the quadratic invariants can be computed from the linear invariants (Angeles, 1988) as

$$\hat{q}_0 = \sqrt{\frac{1+q_0}{2}} \qquad \hat{\mathbf{q}} = \mathbf{q}\frac{\sqrt{2(1+q_0)}}{2(1+q_0)} = \frac{\hat{q}_0}{(1+q_0)}\mathbf{q} \tag{2.2}$$

Using this approach, the computation of $\mathbf{q}$ and q_0 requires $4M$ and $6A$, where M and A denote multiplications and additions, respectively. Next, the computational cost of deriving the quadratic invariants from the linear invariants is $1S+\ 1D+4M+1A$, giving a total of $1S+1D+8M+7A$, where S and D denote square root and division operations. These results are shown in Appendix C.

Secondly, the quadratic invariants can be derived directly from the entries of the given rotation matrix, namely,

$$\begin{aligned} \hat{q}_0 &= \pm\frac{1}{2}\sqrt{1+q_{11}+q_{22}+q_{33}} \\ \hat{\mathbf{q}} &\equiv \begin{bmatrix} \hat{q}_1 \\ \hat{q}_2 \\ \hat{q}_3 \end{bmatrix} = \frac{1}{4\hat{q}_0}\begin{bmatrix} q_{32}-q_{23} \\ q_{13}-q_{31} \\ q_{21}-q_{12} \end{bmatrix} \end{aligned}$$

The above derivation of $\hat{\mathbf{q}}$ is prone to calculation errors if $\hat{q}_0$ is close to 0, which happens when the angle of rotation approaches π. If $\hat{q}_0 \approx 0$, an alternate method can be used to compute $\hat{\mathbf{q}}$. To begin with, using eqs.(2.2), $\mathbf{q}$ can be expressed in terms of $\hat{\mathbf{q}}$ as follows.

$$q_0 = 2\hat{q}_0^2 - 1 \qquad \mathbf{q} = 2\hat{q}_0\hat{\mathbf{q}} \tag{2.3}$$

Next, eqs.(2.3) are replaced into eq.(2.1) to yield an expression for $\mathbf{Q}$ in terms of the quadratic invariants:

$$\mathbf{Q} = (2\hat{q}_0^2 - 1)\mathbf{1} + 2\hat{\mathbf{q}} \otimes \hat{\mathbf{q}} + 2\hat{q}_0(\mathbf{1} \times \hat{\mathbf{q}}) \tag{2.4}$$

Now, from the diagonal entries of both sides of eq.(2.4), we derive:

$$\hat{q}_i = \pm\sqrt{\frac{q_{ii}+1}{2} - q_0^2} \qquad i = 1,2,3$$

In the above derivations, the signs of $\hat{q}_i$ are unknown. However, the entry with the largest absolute value can be found, and the remaining ones can be computed using the following equations, that are derived from the off-diagonal entries of eq.(2.4):

$$4\hat{q}_1\hat{q}_2 = q_{21} + q_{12}$$

$$4\hat{q}_1\hat{q}_3 = q_{31} + q_{13}$$

$$4\hat{q}_2\hat{q}_3 = q_{32} + q_{23}$$

In general, then, the following algorithm can be used to compute the quadratic invariants from the entries of the rotation matrix:

$\hat{q}_0 \leftarrow +\frac{1}{2}\sqrt{1 + q_{11} + q_{22} + q_{33}}$

`if` $|\hat{q}_0| > \epsilon$

$$\begin{bmatrix} \hat{q}_1 \\ \hat{q}_2 \\ \hat{q}_3 \end{bmatrix} \leftarrow \frac{1}{4\hat{q}_0} \begin{bmatrix} q_{32} - q_{23} \\ q_{13} - q_{31} \\ q_{21} - q_{12} \end{bmatrix}$$

`else`

`find maximum` $|q_{ii}|$

$\hat{q}_i \leftarrow \sqrt{\frac{q_{ii}+1}{2} - q_0^2}$

$j \leftarrow (i+1)(mod\ \ 3)$

$\hat{q}_j \leftarrow \frac{1}{4\hat{q}_i}(q_{ji} + q_{ij})$

$k \leftarrow (j+1)(mod\ \ 3)$

$\hat{q}_k \leftarrow \frac{1}{4\hat{q}_i}(q_{ki} + q_{ik})$

linear invariants of the product **AB** are denoted by $\mathbf{q}$ and q_0. Below we include expressions for $\mathbf{q}$ and q_0 in terms of $\mathbf{q}^A$, q_0^A, $\mathbf{q}^B$ and q_0^B, namely,

$$\mathbf{q} = \frac{\mathbf{n}}{2D} \qquad q_0 = \frac{N - D}{2D} \tag{2.5}$$

where

$$D \equiv (1 + q_0^A)(1 + q_0^B) \tag{2.6}$$

$$N \equiv (1 + q_0^A)(1 + q_0^B)(q_0^A + q_0^B + q_0^A q_0^B) + (\mathbf{q}^A \cdot \mathbf{q}^B)(\mathbf{q}^A \cdot \mathbf{q}^B - 2D) \tag{2.7}$$

$$\mathbf{n} \equiv (D - \mathbf{q}^A \cdot \mathbf{q}^B)[(1 + q_0^B)\mathbf{q}^A + (1 + {}_0^A)\mathbf{q}^B + \mathbf{q}^A \times \mathbf{q}^B] \tag{2.8}$$

The reader is referred to Appendix D for the derivation of computational costs with this method.

Quadratic Invariants

Method 1: Matrix Multiplications. As discussed above, the cost of computing five matrix products is $117M$ and $60A$. Using the expressions below, the calculation of the quadratic invariants from the product matrix requires $1S$+$1D$+$5M$+$6A$

$$\hat{q}_0 \equiv \frac{\sqrt{\mathrm{tr}(\mathbf{Q}) + 1}}{2} \qquad \hat{\mathbf{q}} \equiv \frac{1}{2\hat{q}_0}\mathrm{vect}(\mathbf{Q})$$

Method 2: Vector Composition of Linear Invariants. The method of calculation of the linear invariants using eqs.(2.5) can be extended to quadratic invariants. When eqs.(2.5) are substituted into eqs.(2.2), the relations shown below are found (Tandirci, 1991):

$$\hat{q}_0 = \frac{1}{2}\sqrt{1 + \frac{N}{D}} \qquad \hat{\mathbf{q}} = \frac{1}{2}\frac{\sqrt{D(N + D)}}{D(N + D)}\mathbf{n} = \frac{\hat{q}_0}{N + D}\mathbf{n}$$

where D, N and $\mathbf{n}$ were defined in eqs.(2.6–2.8). The computational costs involved are included in Appendix E.

Method 3: Vector Composition of Quadratic Invariants. These relations were found by Rodrigues (1840). They can be verified from the relations obtained in eqs.(2.2) and eqs.(2.5), namely,

$$\begin{aligned} \hat{q}_0 &= \hat{q}_0^A \hat{q}_0^B - \hat{\mathbf{q}}^{\mathbf{A}} \cdot \hat{\mathbf{q}}^{\mathbf{B}} \\ \hat{\mathbf{q}} &= \hat{q}_0(\hat{q}_0^B \hat{\mathbf{q}}^{\mathbf{A}} + \hat{q}_0^A \hat{\mathbf{q}}^{\mathbf{B}} + \hat{\mathbf{q}}^A \times \hat{\mathbf{q}}^{\mathbf{B}}) \end{aligned}$$

$$\min_{\boldsymbol{\theta}} ||\mathbf{f}(\boldsymbol{\theta})||^2 \tag{2.9}$$

without constraints, where

$$\mathbf{f}(\boldsymbol{\theta}) \equiv \begin{bmatrix} 2[\mathrm{vect}(\mathbf{Q}) - \mathrm{vect}(\mathbf{Q}_g)] \\ \mathrm{tr}(\mathbf{Q}) - \mathrm{tr}(\mathbf{Q}_g) \\ \mathbf{p} - \mathbf{p}_g \end{bmatrix}$$

The first four components of vector $\mathbf{f}$ are nonlinearly dependent; in the absence of singularities, we have six independent equations. The factor two multiplying the first three entries is used to eliminate divisions by two, thus saving time in computations.

In order to solve problem (2.9), the gradient of $\mathbf{f}(\boldsymbol{\theta})$ with respect to the joint variables has to be calculated. As derived in (Angeles, 1985), for a six-axis manipulator, this gradient can be expressed as

$$\mathbf{J}(\boldsymbol{\theta}) \equiv \frac{\partial \mathbf{f}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \mathbf{HK}$$

where the 7×6 matrix $\mathbf{H}$ arises from the formulation of the orientation equations, while $\mathbf{K}$, a 6×6 matrix, is commonly known as the Jacobian matrix (Whitney, 1972), taking on the form

$$\mathbf{K} \equiv \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \dots & \mathbf{e}_6 \\ \mathbf{e}_1 \times \mathbf{r}_1 & \mathbf{e}_2 \times \mathbf{r}_2 & \dots & \mathbf{e}_6 \times \mathbf{r}_6 \end{bmatrix} \tag{2.10}$$

As derived in (Angeles, 1985), the $\mathbf{H}$ matrix takes on the form,

$$\mathbf{H} \equiv \begin{bmatrix} \mathbf{1}\mathrm{tr}(\mathbf{Q}) - \mathbf{Q} & \mathbf{0}_3 \\ -2\mathrm{vect}(\mathbf{Q})^T & \mathbf{0}^T \\ \mathbf{0}_3 & \mathbf{1}_3 \end{bmatrix}$$

where $\{\mathbf{e}_i\}_1^6$ are the unit vectors parallel to the axes of the joints, and $\{\mathbf{r}_i\}_1^6$ are the vectors directed from point O_i to point P of the EE as shown in Fig. 2.1. If the ith joint is prismatic, then $\mathbf{k}_i$, the ith column of $\mathbf{K}$ matrix becomes,

$$\mathbf{k}_i = \begin{bmatrix} \mathbf{0} \\ \mathbf{e}_i \end{bmatrix}$$

From its series expansion, a first-order approximation of function $\mathbf{f}(\boldsymbol{\theta})$ evaluated at the current value of $\boldsymbol{\theta}$ allows the computation of $\Delta\boldsymbol{\theta}$ from

$$\mathbf{J}\Delta\boldsymbol{\theta} = -\mathbf{f}$$

Depending on the chosen initial guess, the above method yields different solutions upon convergence. As proven by Lee and Liang (1988), up to 16 solutions are to be expected. The solution obtained will be a local minimum of the problem in eq.(2.9) that verifies the normality condition,

$$\mathbf{J}^T\mathbf{f} = \mathbf{0}$$

2.2.2 Quadratic Invariants (Euler Parameters)

With this rotation representation, the formulation of the problem is similar to the one with the linear invariants. Instead of the linear invariants, the quadratic invariants are required to match their prescribed counterparts. Thus, the problem is

$$\min_{\boldsymbol{\theta}} ||\mathbf{f}(\boldsymbol{\theta})||^2$$

without constraints, where

$$\mathbf{f}(\boldsymbol{\theta}) \equiv \begin{bmatrix} 2[\text{vect}(\sqrt{\mathbf{Q}}) - \text{vect}(\sqrt{\mathbf{Q}_g})] \\ \text{tr}(\sqrt{\mathbf{Q}}) - \text{tr}(\sqrt{\mathbf{Q}_g}) \\ \mathbf{p} - \mathbf{p}_g \end{bmatrix}$$

As shown in (Angeles, 1991), the gradient of $\mathbf{f}(\boldsymbol{\theta})$ is now calculated by making use of the following relations:

$$\frac{\partial\sqrt{\mathbf{Q}}}{\partial\theta_i} = \mathbf{1} \times \mathbf{e}_i\sqrt{\mathbf{Q}}$$

Hence,

$$\frac{\partial\text{vect}(\sqrt{\mathbf{Q}})}{\partial\theta_i} = \text{vect}(\mathbf{1} \times \mathbf{e}_i\sqrt{\mathbf{Q}}) = \frac{1}{2}[\text{tr}(\sqrt{\mathbf{Q}})\mathbf{1} - \sqrt{\mathbf{Q}}]\mathbf{e}_i$$

$$\frac{\partial\text{tr}(\sqrt{\mathbf{Q}})}{\partial\theta_i} = \text{tr}(\mathbf{1} \times \mathbf{e}_i\sqrt{\mathbf{Q}}) = -2[\text{vect}(\sqrt{\mathbf{Q}})]^T\mathbf{e}_i$$

Therefore, the Jacobian matrix can now be factored as,

$$\mathbf{J}' \equiv \mathbf{H}'\mathbf{K}$$

where

$$\mathbf{H}' \equiv \begin{bmatrix} \mathbf{1}\text{tr}(\sqrt{\mathbf{Q}}) - \sqrt{\mathbf{Q}} & \mathbf{0}_3 \\ -2\text{vect}(\sqrt{\mathbf{Q}})^T & \mathbf{0}^T \\ \mathbf{0}_3 & \mathbf{1}_3 \end{bmatrix}$$

and $\mathbf{K}$ is the same velocity Jacobian defined before. The solution is obtained using the Newton-Gauss procedure.

2.2.3 Natural Invariants

This method is different from the above two methods because it does not need an auxiliary matrix, such as $\mathbf{H}$ or $\mathbf{H}'$, in its Jacobian. Instead, the Jacobian used is the velocity Jacobian $\mathbf{K}$ defined in eq.(2.10). The objective here is twofold. First, we wish to minimize the difference between the current and the prescribed rotation matrices, $\mathbf{Q}$ and $\mathbf{Q}_g$. Secondly, we wish to minimize the difference between the current and prescribed position vectors of the EE, which is denoted below as $\Delta\mathbf{p}$. The velocity Jacobian relates the incremental joint angles to the vector of the difference in poses (Hiller and Woernle, 1989; Paul, 1981), namely

$$\mathbf{K}(\boldsymbol{\theta})\Delta\boldsymbol{\theta} = \begin{bmatrix} \sin\Delta\phi\mathbf{u} \\ \Delta\mathbf{p} \end{bmatrix} \tag{2.11}$$

where the unit vector $\mathbf{u}$ and the scalar $\Delta\phi$ express the difference between the current and prescribed orientations, $\mathbf{Q}$ and $\mathbf{Q}_g$. Thus, frame $\mathcal{C}$ is carried into $\mathcal{G}$ by a rotation about an axis parallel to the unit vector $\mathbf{u}$ through an angle $\Delta\phi$. The vector $\mathbf{u}$ and the angle $\Delta\phi$ are termed the ***natural invariants*** of the rotation involved in (Angeles, 1988). Let us denote the prescribed pose as the $\mathcal{G}$-frame, the actual or current configuration as the $\mathcal{C}$-frame, and the base frame as the $\mathcal{F}$-frame, as shown in Fig. 2.2. The rotation carrying $\mathcal{C}$ into $\mathcal{G}$, here denoted as $\Delta\mathbf{R}$, is first calculated in the $\mathcal{C}$ frame and then transformed into base coordinates using an orthogonal transformation carrying $\mathcal{F}$ into $\mathcal{C}$, and denoted as $\mathbf{Q}$, namely,

$$\mathbf{Q}\Delta\mathbf{R} = \mathbf{Q}_g$$

Hence,

$$[\Delta\mathbf{R}]_{\mathcal{C}} = \mathbf{Q}^T\mathbf{Q}_g$$

i.e., $\Delta\mathbf{R}$ in the $\mathcal{F}$ frame is given as

$$[\Delta\mathbf{R}]_{\mathcal{F}} = \mathbf{Q}\,[\Delta\mathbf{R}]_{\mathcal{C}}\,\mathbf{Q}^T = \mathbf{Q}_g\mathbf{Q}^T$$

The product $\sin\Delta\phi\mathbf{u}$ is then readily calculated from $\Delta\mathbf{R}$ as

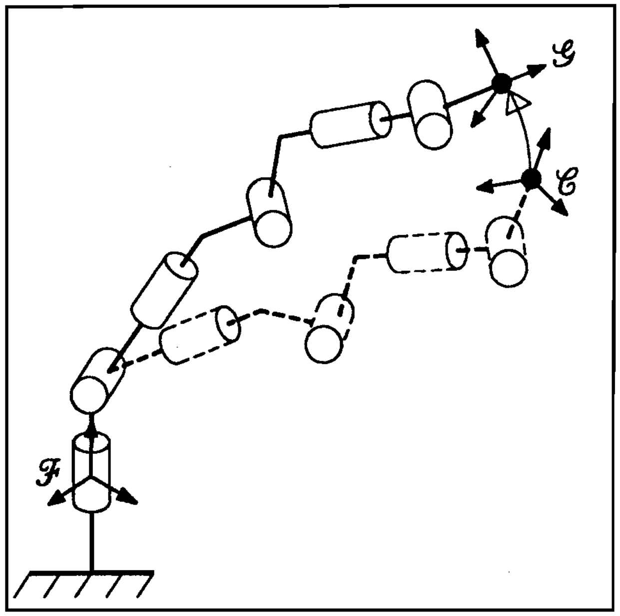


Figure 2.2: General 6-axis manipulator at the current $\mathcal{C}$ and prescribed $\mathcal{G}$ configurations

$$\sin \Delta\phi \mathbf{u} = \mathrm{vect}(\Delta \mathbf{R})$$

Once the right-hand side of the algebraic system of eq.(2.11) is determined, the solution $\Delta\boldsymbol{\theta}$ can be obtained using the LU-decomposition (Golub and Van Loan, 1983). The new vector of joint variables is then obtained as,

$$\boldsymbol{\theta}^1 = \boldsymbol{\theta}^0 + \Delta\boldsymbol{\theta}$$

and, at the next iteration, $\boldsymbol{\theta}^1$ is used to compute the said right-hand side vector, as well as the Jacobian $\mathbf{K}$. The procedure continues until $\Delta\boldsymbol{\theta}$ becomes smaller than a specified tolerance.

2.2.4 Comparisons Among Displacement Inverse Kinematics Methods

When computing inverse kinematics solutions, it is necessary to converge to a solution quickly. Thus, the comparisons between three methods are made on the following bases:

- speed in calculation of the function $\mathbf{f}(\boldsymbol{\theta})$ and of its Jacobian matrix;

Table 2.2: Computational cost of the vector function

	Linear Inv	Quadratic Inv	Natural Inv
operations	$117M + 72A$	$1S{+}1D{+}123M{+}74A$	$144M{+}78A$

• condition number of the Jacobian, indicating the numerical conditioning of the problem formulation;

• overall performance of the numerical procedure based on the number of iterations needed to converge both close to a solution and away from it.

Time Complexity in the Formulation of the Objective Function

The formulation of the positioning equations is identical in all methods compared. The position vector $\mathbf{r}_1$ is available as the velocity Jacobian is calculated. Thus, only 3 subtractions are needed for the difference vector between the prescribed and current positions.

From the final rotation matrix product, which is calculated in Appendix B with $117M$ and $60A$, the vector $2\mathbf{q}$ and the trace can be extracted in $5A$. Furthermore, $7A$ are needed to find the difference with the prescribed pose expressions. Thus computation of $\mathbf{f}(\boldsymbol{\theta})$ requires $117M$ and $72A$ in the case of linear invariants.

As shown in Section 2.1.3, the quadratic invariants $2\hat{\mathbf{q}}$ and $\hat{q}_0$ can be computed with $1S + 1D + 122M + 66A$, while the trace can be computed from $\hat{q}_0$ in $1M{+}1A$. In the case of quadratic invariants, the computation of $\mathbf{f}(\boldsymbol{\theta})$ requires $1S + 1D + 123M + 74A$.

In the case of the method based on the natural invariants, we have first the relation

$$[\Delta\mathbf{R}]_{\mathcal{F}} = \mathbf{Q}_g\mathbf{Q}^T$$

which requires $24M$ and $12A$, using a matrix-product method similar to the one outlined in Appendix B. Furthermore, vect($\Delta\mathbf{R}$) requires $3M$ and $3A$, and thus, the derivation of the left-hand-side vector requires $144M$ and $78A$. Table 2.2 shows the operations needed to calculate the orientation equations with all three methods.

Time Complexity in the Formulation of the Linear Algebraic System

The velocity Jacobian is needed in all three formulations. The computational cost of the velocity Jacobian is calculated in Appendix F as $12T$, $81M$ and $49A$, where T denotes trigonometric operations such as sine and cosine. The current position vector $\mathbf{r}_1$ is readily available from the computation of the said Jacobian. The computation of the Jacobians of the methods based on the invariant vectors requires derivations of the auxiliary matrices and their products with the velocity Jacobian. With linear invariants, the first three rows of the velocity Jacobian are multiplied by a 4×3 matrix $\mathbf{L}$ defined as,

$$\mathbf{L} = \begin{bmatrix} \mathbf{1}\mathrm{tr}(\mathbf{Q}) - \mathbf{Q} \\ -2\mathrm{vect}(\mathbf{Q})^T \end{bmatrix}$$

Moreover, the Jacobian expression can be written as,

$$\mathbf{J} = \begin{bmatrix} \mathbf{LA} \\ \mathbf{B} \end{bmatrix} = \begin{bmatrix} [\mathbf{1}\mathrm{tr}(\mathbf{Q}) - \mathbf{Q}]\mathbf{A} \\ -2[\mathrm{vect}(\mathbf{Q})]^T\mathbf{A} \\ \mathbf{B} \end{bmatrix}$$

It is noted that $\mathbf{Q}$, $\mathrm{tr}(\mathbf{Q})$ and $2\mathrm{q}$ are available from the derivation of the function $\mathbf{f}(\boldsymbol{\theta})$. Hence, the construction of $\mathbf{L}$ takes $3A$ for the upper three rows and no operations for the fourth row. The product of $\mathbf{L}$ with $\mathbf{A}$ requires in turn $72M$ and $48A$. Thus, an additional $72M$ and $51A$ are needed for $\mathbf{J}$, once $\mathbf{K}$ is derived.

In the case of the quadratic invariants, $\mathrm{tr}(\sqrt{\mathbf{Q}})$, $2\hat{\mathrm{q}}$ and $\hat{q}_0$ are available from the derivation of $\mathbf{f}(\boldsymbol{\theta})$, and $\sqrt{\mathbf{Q}}$ can be computed from the quadratic invariants $\hat{\mathrm{q}}$ and $\hat{q}_0$ in $1D+12M+10A$, as shown in Appendix G, whereas $\hat{\mathrm{q}}$ is derived from $2\hat{\mathrm{q}}$ in $3M$. Furthermore, $\mathbf{J}'$ is calculated similar to the above case in $72M$ and $51A$.

On the other hand, using the natural invariants, the Jacobian is $\mathbf{K}$ itself, and no additional operations are needed. The number of operations and the observed CPU times needed to compute $\mathbf{J}$ and $\mathbf{f}$ are reported in Table 2.3.

From Table 2.3, it is apparent that all three methods are comparable for the evaluation of $\mathbf{J}(\boldsymbol{\theta})$ and $\mathbf{f}(\boldsymbol{\theta})$. However, there is one more consideration: In the case of natural invariants, the Jacobian is of 6×6, and the LU-decomposition is used to solve the system of equations. The other two methods involve a 7×6 Jacobian matrix, and use Householder Reflections (Golub and Van Loan, 1983). Because the array size is smaller and because the

Table 2.3: Computation of Jacobian **J** and vector function **f**

Method	Operation Count	CPU Times (μsec)
Linear Invariants	$12T$ $270M$ $172A$	325.0
Quadratic Invariants	$1S$ $12T$ $2D$ $291M$ $184A$	345.0
Natural Invariants	$12T$ $225M$ $127A$	310.0

LU-decomposition is computationally less expensive than Householder reflections (Press et al., 1988), in comparison with the other two, the third method is expected to be even less time consuming. In Table 2.4, the total CPU times observed for the derivation of the same solution are reported. Table 2.4 shows that, as expected, the third method requires the least

Table 2.4: Overall CPU times per iteration

	Linear Inv	Quadratic Inv	Natural Inv
CPU Times (μsec)	835.	850.	747.5

amount of time per iteration loop in the numerical procedure. However, a difference of less than ten percent between the three methods may be considered insignificant.

Comparison of the Condition Numbers of the Jacobians

The major disadvantage of the linear invariants is that the Jacobian becomes singular when the angle of rotation ϕ is π. This type of singularity is known as *formulation* or *algebraic singularity*, since it arises only because of the way the problem is formulated. The first three rows of matrix $\mathbf{L}$ defined above will be linearly dependent for $\phi = \pi$, and $\phi = \pm\frac{\pi}{2}$. However, the overall matrix $\mathbf{L}$ is of full rank for $\phi = \pm\frac{\pi}{2}$, as shown in (Angeles, 1991). If $\mathbf{L}$ is rank-deficient, then the factor matrix $\mathbf{H}$ will also be rank-deficient. The product of a rank-deficient matrix with any other matrix is also rank-deficient. Thus, $\mathbf{J}$ will also be rank-deficient, and a solution cannot be obtained with the underlying numerical procedure.

At $\phi = \pi$, the matrix $\mathbf{M}$, defined as

$$\mathbf{M} \equiv \mathbf{1}\mathrm{tr}(\mathbf{Q}) - \mathbf{Q}$$

becomes

$$\mathbf{M}(\pi) = -2\mathbf{u} \otimes \mathbf{u}$$

which is a rank-one matrix, and $\mathrm{vect}(\mathbf{Q})$ vanishes by virtue of its symmetry. As shown in (Angeles, 1991), the condition number of $\mathbf{L}$ takes on the form,

$$\kappa(\mathbf{L}) = \frac{2}{1 + \cos\phi}$$

from which the singularity at $\phi = \pi$ can be verified.

Unlike the linear invariants, in the case of quadratic invariants, the matrix $\mathbf{M}'$ defined below

$$\mathbf{M}' \equiv \mathbf{1}\mathrm{tr}(\sqrt{\mathbf{Q}}) - \sqrt{\mathbf{Q}}$$

remains of full rank for all possible rotations ϕ (Angeles, 1991). Therefore, the matrix $\mathbf{L}'$, defined as

$$\mathbf{L}' = \begin{bmatrix} \mathbf{1}\mathrm{tr}(\sqrt{\mathbf{Q}}) - \sqrt{\mathbf{Q}} \\ -2\mathrm{vect}(\sqrt{\mathbf{Q}})^T \end{bmatrix}$$

is always of full rank and the quadratic invariants do not lead to formulation singularities.

In the case of the natural invariants, formulation singularities do not exist, since the Jacobian $\mathbf{K}$ does not appear multiplied by any other matrix.

Table 2.5: Range of κ_{max} values

index on the horizontal axis	range of κ_{max}
1	$\kappa_{max} \leq 10$
2	$10 < \kappa_{max} \leq 20$
3	$20 < \kappa_{max} \leq 100$
4	$100 < \kappa_{max} \leq 1000$
5	$\kappa_{max} > 1000$

It has been proven that there is an upper-bound for the condition number of the product of two matrices (Stewart, 1973), namely

$$\kappa(\mathbf{HK}) \leq \kappa(\mathbf{H})\kappa(\mathbf{K})$$

however, nothing can be said for the lower bound, except for the fact that $\kappa(\mathbf{HK})$ is greater or equal to unity. Therefore, no comparison can be made between the Jacobian used with the natural invariants and that of the other invariants.

In order to assess the conditioning of the numerical schemes for DIK associated with each of the three rotation representations, experiments were done using closed path tracking applications with 100 data points. The said points are obtained on the intersection of two cylinders. Furthermore, Frenet-Serret frames are used (Angeles, Rojas and López-Cajún, 1988) to specify the orientation of the EE. Histograms of the maximum condition numbers encountered along the above paths are shown in Fig. 2.3. Here, the vertical axis indicates the number of occurrences of the condition number falling in the range of values indicated on the horizontal axis. The ranges of condition-number values are shown in the Table 2.5.

From the histograms in Fig. 2.3, it is observed that the quadratic invariants allow more occurrences of lower condition numbers in the 100 points traced, whereas the linear invariants lead to high condition numbers in a number of data points. Moreover, in this example, the natural invariants never lead to condition numbers higher than 100.

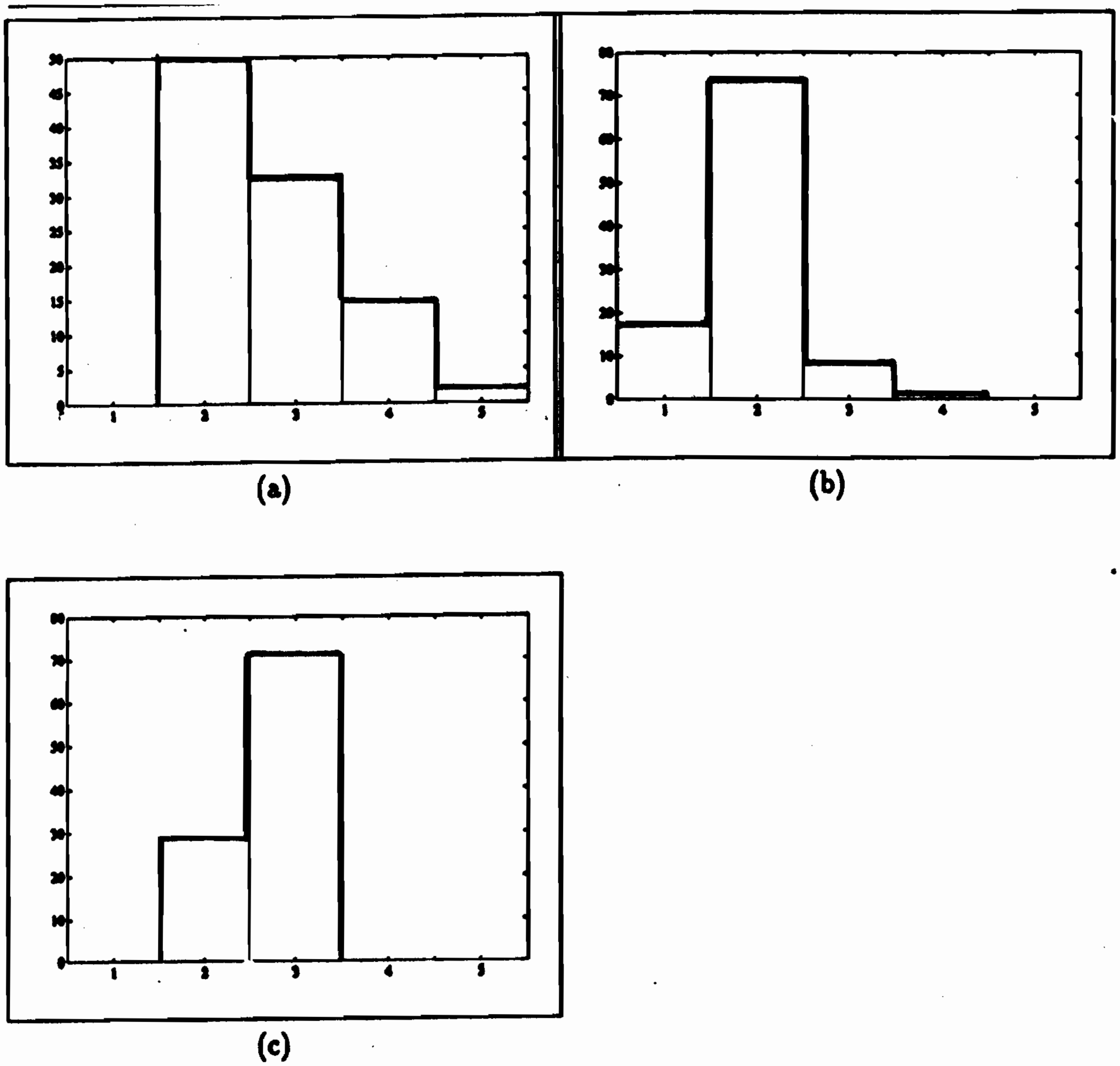


Figure 2.3: Histogram of condition-number frequencies a) Linear invariants b) Quadratic invariants c) Natural invariants

Comparison on the Basis of Convergence Speed

Since all three methods are based on approximations, convergence properties cannot be predicted theoretically. The first two methods rely on the Taylor expansion of the function $\mathbf{f}(\boldsymbol{\theta})$ and employ the Newton-Gauss method. The third formulation is derived from the joint-rate relations.

Convergence in the Neighbourhood of a Solution: Experiments were made in order to investigate the convergence properties of the algorithms studied. The foregoing set of data points are approached in the neighbourhood of the solution, such that $\|\mathbf{f}\| < 1.0$ at the initial guess. The convergence speeds obtained are summarized with the histograms shown in Fig. 2.4. Furthermore, total times spent to traverse the above three paths are measured on an *IRIS 4D/210VGX*. These are reported in Table 2.6. In the histograms, the vertical

Table 2.6: Convergence speed in the vicinity of the solution

	Linear Inv	Quadratic Inv	Natural Invariants
Total CPU (msec)	340.	2500.	290.

axis indicates the number of occurrences of the number of iterations falling in the ranges of values indicated on the horizontal axis. The range of numbers of iterations are shown in Table 2.7. From the histograms, it is observed that the natural invariants always converge in

Table 2.7: Range of iteration numbers

index on the horizontal axis	range of iteration no.s: n
1	$n \leq 5$
2	$5 < n \leq 10$
3	$10 < n \leq 20$
4	$20 < n \leq 40$
5	$n > 100$

less than 5 iterations. Similarly, the linear invariants also converge very quickly, whereas the

Convergence Away From the Solution: The above observations are valid if the initial guess is in the vicinity of the solution. Such observations are also made when the initial guess lies far away from the prescribed set of data points. Five arbitrary solution points are selected, and two quantities are monitored: *i)* n, the number of iterations, and *ii)* κ_{max}, the maximum condition number encountered. The results are shown in Table 2.8, which shows a clear correlation between κ_{max} and n.

Table 2.8: Convergence speed and conditioning away from the solution

	Linear Inv		Quadratic Inv		Natural Invariants	
Test	n	κ_{max}	n	κ_{max}	n	κ_{max}
1	7	26.	21	14.	7	39.
2	5	9.	19	8.	5	14.
3	19	60316.	28	3407.	26	607.
4	5	10.	21	10.	5	15.
5	4	9.	20	8.	4	14.

2.3 Velocity Inverse Kinematics

In velocity inverse kinematics (VIK), the aim is to calculate the joint-rates so as to match the angular and translational velocity of the EE to their prescribed counterparts. The angular velocities of a six-axis manipulator are expressed as follows (Angeles, 1989):

$$\begin{aligned} \omega_1 &= \dot{\theta}_1 \mathbf{e}_1 \\ \omega_2 &= \dot{\theta}_1 \mathbf{e}_1 + \dot{\theta}_2 \mathbf{e}_2 \\ &\cdot \\ \omega_6 &= \dot{\theta}_1 \mathbf{e}_1 + \dot{\theta}_2 \mathbf{e}_2 + \ldots + \dot{\theta}_6 \mathbf{e}_6 \end{aligned} \tag{2.12}$$

The angular velocity of the EE can be expressed as

$$\omega \equiv \omega_6 = \mathbf{A}\dot{\boldsymbol{\theta}} \tag{2.13}$$

where $\mathbf{A}$ is the 3×6 matrix displayed below:

$$\mathbf{A} \equiv [\mathbf{e}_1 \quad \mathbf{e}_2 \quad \mathbf{e}_3 \quad \mathbf{e}_4 \quad \mathbf{e}_5 \quad \mathbf{e}_6]$$

Furthermore, for the position vector of point P of the EE, we have,

$$\mathbf{p} = \mathbf{a}_1 + \mathbf{a}_2 + \ldots + \mathbf{a}_6$$

where $\{\mathbf{a}_i\}_1^6$ denotes the set of vectors directed from O_i to O_{i+1}. Upon differentiation of both sides of the above equation with respect to time, the velocity of the EE is written as

$$\dot{\mathbf{p}} = \dot{\mathbf{a}}_1 + \dot{\mathbf{a}}_2 + \ldots + \dot{\mathbf{a}}_6 \tag{2.14}$$

where

$$\dot{\mathbf{a}}_i = \boldsymbol{\omega}_i \times \mathbf{a}_i \tag{2.15}$$

Upon substitution of eqs.(2.12) and (2.15) into (2.14), we have

$$\dot{\mathbf{p}} = \dot{\theta}_1 \mathbf{e}_1 \times \mathbf{r}_1 + \dot{\theta}_2 \mathbf{e}_2 \times \mathbf{r}_2 + \ldots + \dot{\theta}_6 \mathbf{e}_6 \times \mathbf{r}_6 \tag{2.16}$$

where $\mathbf{r}_i$ is the vector directed from point O_i of the ith axis to point P of the EE, i.e.,

$$\mathbf{r}_i = \mathbf{a}_i + \mathbf{a}_{i+1} + \ldots + \mathbf{a}_6$$

Therefore, eq.(2.16) is written in the form

$$\dot{\mathbf{p}} \equiv \dot{\mathbf{r}}_1 = \mathbf{B}\dot{\boldsymbol{\theta}} \tag{2.17}$$

where the 3×6 matrix $\mathbf{B}$ takes on the form:

$$\mathbf{B} \equiv [\mathbf{e}_1 \times \mathbf{r}_1 \quad \mathbf{e}_2 \times \mathbf{r}_2 \quad \mathbf{e}_3 \times \mathbf{r}_3 \quad \mathbf{e}_4 \times \mathbf{r}_4 \quad \mathbf{e}_5 \times \mathbf{r}_5 \quad \mathbf{e}_6 \times \mathbf{r}_6]$$

From eqs.(2.13) and (2.17), we construct the velocity Jacobian matrix and then solve for the joint rates from the linear system of equations thus derived, namely,

$$\mathbf{K}\dot{\boldsymbol{\theta}} \equiv \mathbf{t} \tag{2.18}$$

where $\mathbf{t}$ is defined as the twist vector, and is expressed as

$$\mathbf{t} \equiv \begin{bmatrix} \boldsymbol{\omega} \\ \dot{\mathbf{p}} \end{bmatrix}$$

where $\boldsymbol{\omega}$ and $\dot{\mathbf{p}}$ denote angular and translational velocities of the operation point P of the end-effector.

2.3.1 Treatment of Singularities

In continuous-path applications, the EE is to follow a set of prescribed positions and orientations in 3D space. Let points $i-1$, i and $i+1$ be three consecutive points in the six-dimensional configuration space, for the EE to follow, with corresponding configurations having the joint variables $\boldsymbol{\theta}^{-1}$, $\boldsymbol{\theta}^{0}$ and $\boldsymbol{\theta}^{+1}$. Further, let us assume that the manipulator has a kinematic singularity at point i, while being full ranked at two other points. The DIK algorithm applied at point $i-1$ yields the solution $\boldsymbol{\theta}^{0}$. With the above assumption, $\mathbf{K}(\boldsymbol{\theta}^{0})$ is singular, and the joint-rate vector $\dot{\boldsymbol{\theta}}^{0}$ cannot be computed uniquely from the linearized equations (2.18), namely, from

$$\mathbf{K}(\boldsymbol{\theta}^{0})\dot{\boldsymbol{\theta}}^{0} = \begin{bmatrix} \boldsymbol{\omega}^{i} \\ \dot{\mathbf{p}}^{i} \end{bmatrix}$$

Since $\mathbf{K}$ is rank-deficient, the above underdetermined system of equations lead to infinitely many solutions. Commonly, in such cases, minimum-norm algorithms are used to determine $\dot{\boldsymbol{\theta}}^{0}$. However, the minimum-norm solution thus obtained is likely to be away from the previous solution, $\dot{\boldsymbol{\theta}}^{-1}$, thus resulting in an infinite joint acceleration between points $i-1$ and i.

In order to avoid a discontinuity in the joint-rate vectors, an objective function is introduced to select a solution at the singularity $\dot{\boldsymbol{\theta}}^{0}$ that will minimize the distance to the solution just before the singularity, $\dot{\boldsymbol{\theta}}^{-1}$. The problem is now formulated as in (Angeles, Anderson, Cyril and Chen, 1988). Let the objective function be

$$z(\boldsymbol{\theta}^{0}) = \frac{1}{2}\boldsymbol{\psi}^{T}\mathbf{W}\boldsymbol{\psi}$$

where

$$\boldsymbol{\psi} \equiv (\dot{\boldsymbol{\theta}}^{0} - \dot{\boldsymbol{\theta}}^{-1})$$

The problem is defined as

$$\min_{\boldsymbol{\theta}} z(\boldsymbol{\theta}^{0})$$

subject to

$$\mathbf{K}_{r}(\boldsymbol{\theta}^{0})\boldsymbol{\psi} = \mathbf{t}_{r}{}^{i} - \mathbf{K}_{r}(\boldsymbol{\theta}^{0})\dot{\boldsymbol{\theta}}^{-1}$$

where $\mathbf{W}$ is an $n \times n$ dimensional positive-definite matrix, accounting for nondimensionalizing and scaling, and $\mathbf{K}_{r}$ is a reduced $r \times n$-dimensional velocity Jacobian of rank r, and $\mathbf{t}_{r}$ is a reduced r-dimensional vector of rank r.

The above problem is solved using the orthogonal decomposition algorithm, whose details are outlined in (Angeles, Anderson and Gosselin, 1987).

2.4 Acceleration Inverse Kinematics

The acceleration inverse kinematics problem is formulated by taking the time derivative of both sides of eqs.(2.18), which yields

$$\mathbf{K}\ddot{\boldsymbol{\theta}} \equiv \dot{\mathbf{t}} - \dot{\mathbf{K}}\dot{\boldsymbol{\theta}}$$

where $\dot{\mathbf{t}} \equiv [\dot{\boldsymbol{\omega}}, \ddot{\mathbf{p}}]^T$ denotes the prescribed values of angular and translational accelerations. The time derivative of $\mathbf{K}$, $\dot{\mathbf{K}}$ is computed as follows:

$$\dot{\mathbf{K}} = \begin{bmatrix} \dot{\mathbf{A}} \\ \dot{\mathbf{B}} \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{e}}_1 & \dot{\mathbf{e}}_2 & \cdots & \dot{\mathbf{e}}_5 & \dot{\mathbf{e}}_6 \\ \dot{\mathbf{e}}_1 \times \dot{\mathbf{r}}_1 & \dot{\mathbf{e}}_2 \times \dot{\mathbf{r}}_2 & \cdots & \dot{\mathbf{e}}_5 \times \dot{\mathbf{r}}_5 & \dot{\mathbf{e}}_6 \times \dot{\mathbf{r}}_6 \end{bmatrix}$$

From (Angeles, 1989), we have,

$$\begin{aligned} \dot{\mathbf{e}}_i &= \boldsymbol{\omega}_i \times \mathbf{e}_i, \quad \text{for } i = 2, 3.., 6 \\ \dot{\mathbf{e}}_1 &= 0 \end{aligned}$$

We also have, from the same reference,

$$\dot{\mathbf{r}}_6 = \boldsymbol{\omega}_6 \times \mathbf{a}_6$$

and

$$\dot{\mathbf{r}}_i = \boldsymbol{\omega}_i \times \mathbf{a}_i + \mathbf{Q}_i \mathbf{r}_{i+1}, \quad \text{for} \quad i = 5, 4.., 1$$

so that

$$\dot{\mathbf{K}} = \begin{bmatrix} \boldsymbol{\omega}_1 \times \mathbf{e}_1 & \boldsymbol{\omega}_2 \times \mathbf{e}_2 & \cdots & \boldsymbol{\omega}_6 \times \mathbf{e}_6 \\ \mathbf{e}_1 \times \dot{\mathbf{r}}_1 & (\boldsymbol{\omega}_1 \times \mathbf{e}_2) \times \mathbf{r}_2 + \mathbf{e}_2 \times \dot{\mathbf{r}}_2 & \cdots & (\boldsymbol{\omega}_5 \times \mathbf{e}_6) \times \mathbf{r}_6 + \mathbf{e}_6 \times \dot{\mathbf{r}}_6 \end{bmatrix}$$

When kinematic singularities exist on the path, we have

$$\mathbf{K}(\boldsymbol{\theta}^0)\ddot{\boldsymbol{\theta}}^0 = \dot{\mathbf{t}}^i - \dot{\mathbf{K}}(\boldsymbol{\theta}^0)\dot{\boldsymbol{\theta}}^0$$

where $\dot{\boldsymbol{\theta}}^0$ is obtained in VIK, and $\ddot{\boldsymbol{\theta}}^0$ is computed using the procedure similar to the one outlined in the singularity-handling algorithm.

Given that $\mathbf{K}(\boldsymbol{\theta}^0)$ is singular, the DIK problem at point i is solved using a truncated Taylor series approximation with the help of the joint rates and joint accelerations that are obtained previously (Angeles, Anderson, Cyril and Chen, 1988), i.e.,

$$\boldsymbol{\theta}^1 = \boldsymbol{\theta}^0 + \dot{\boldsymbol{\theta}}^0 \Delta t + \frac{1}{2}\ddot{\boldsymbol{\theta}}^0 \Delta t^2$$

thereby completing all the expressions needed for the IKP.

Chapter 3

Conditioning Analysis of Robotic Manipulators

3.1 Condition Number Minimization

The condition number of a matrix is a measure of the error amplification upon solving the linear algebraic system associated with that matrix. For example, when relating the joint-rate vector $\dot{\boldsymbol{\theta}}$ with the 6-dimensional twist vector $\mathbf{t}$, the linear system given by eq.(2.18) arises. If $\mathbf{K}$ is ill-conditioned, small perturbations in the data, i.e., in the entries of $\mathbf{K}$ and $\mathbf{t}$, may cause large variations in the solution, $\dot{\boldsymbol{\theta}}$. In the worst case, when $\mathbf{K}$ is singular, the system does not admit a solution.

Introduced in this chapter is the characteristic length L of a manipulator. This length is used to achieve dimensional homogeneity in the kinematics equations given by eq.(2.18). For instance, the displacement terms involve units of length, while the orientation terms are dimensionless. In order to render the Jacobian dimensionally homogeneous, the last three rows containing components with dimensions of length are divided by the characteristic length, thus obtaining

$$\mathbf{K} \equiv \begin{bmatrix} \mathbf{A} \\ \frac{1}{L}\mathbf{B} \end{bmatrix} = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \cdots & \mathbf{e}_5 & \mathbf{e}_6 \\ \frac{1}{L}\mathbf{e}_1 \times \mathbf{r}_1 & \frac{1}{L}\mathbf{e}_2 \times \mathbf{r}_2 & \cdots & \frac{1}{L}\mathbf{e}_5 \times \mathbf{r}_5 & \frac{1}{L}\mathbf{e}_6 \times \mathbf{r}_6 \end{bmatrix}$$

Dimensional homogeneity is needed to avoid potentially large numerical differences between the orientation and translational entries of the Jacobian, thus improving the numerical stability of the linear system of equations that are solved at every iteration of the Newton-type procedure. We will discuss below how L is determined for a given manipulator.

Furthermore, the characteristic point, P_C of a general six-axis manipulator is introduced. In most cases, the end point P of the EE is assumed to lie on the axis of the last joint. It will be shown that, if P is chosen conveniently, a Jacobian with a minimum condition number can be achieved.

The procedure to minimize the condition number and at the same time, to determine the characteristic point and characteristic length, is based on the fact that, for isotropic Jacobians $\mathbf{K}_I$, the product $\mathbf{K}_I\mathbf{K}_I{}^T$ is proportional to the identity matrix $\mathbf{1}$, i.e.,

$$\mathbf{K}_I\mathbf{K}_I{}^T = \sigma^2\mathbf{1}$$

where σ is a real number. Isotropic Jacobians have a condition number of unity, which is the minimum that the condition number can attain. The procedure that we will use here is that introduced in (Angeles and López-Cajún, 1988) which is based on a least-square approach.

3.2 Examples

We illustrate the foregoing concepts with a few examples of manipulators of various types in this section.

3.2.1 Planar 3-Axis Manipulators

In this subsection, two special cases are studied. In the first case, the link lengths and joint angles will be found so that isotropy can be achieved, while, in the second case, all link lengths are assumed to be equal. With the latter assumption isotropy cannot be obtained, but the manipulator configuration of minimum condition number will be determined, along with its characteristic length.

The dimensionless form of the Jacobian of planar 3-axis manipulators can be written as

$$\mathbf{K}_p \equiv \begin{bmatrix} 1 & 1 & 1 \\ \frac{1}{L}\mathbf{E}\mathbf{r}_1 & \frac{1}{L}\mathbf{E}\mathbf{r}_2 & \frac{1}{L}\mathbf{E}\mathbf{r}_3 \end{bmatrix}$$

where $\mathbf{E}$ is the 2×2 orthogonal matrix defined as

$$\mathbf{E} \equiv \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

Therefore,

$$\mathbf{K}_p\mathbf{K}_p{}^T = \begin{bmatrix} 3 & \frac{1}{L}(\sum_1^3 \mathbf{r}_k)^T\mathbf{E}^T \\ \frac{1}{L}\mathbf{E}\sum_1^3 \mathbf{r}_k & \frac{1}{L^2}\mathbf{E}(\sum_1^3 \mathbf{r}_k\mathbf{r}_k{}^T)\mathbf{E}^T \end{bmatrix}$$

From the above definitions it is evident that, in order for $\mathbf{K}_p\mathbf{K}_p{}^T$ to be proportional to the identity matrix, we must have,

$$\sigma^2 \equiv 3 \quad \text{and} \quad \sum_1^3 \mathbf{r}_k \equiv \mathbf{0}$$

which means that the proportionality constant is 3, and the EE must be the centroid of the position vectors $\{\mathbf{r}_i\}_1^3$. Furthermore, the lower-right block of $\mathbf{K}_p\mathbf{K}_p^T$ should be 3 times the 2×2 identity matrix $\mathbf{1}_2$, i.e.,

$$\frac{1}{L^2}\mathbf{E}(\sum_1^3 \mathbf{r}_k\mathbf{r}_k{}^T)\mathbf{E}^T = (3)\mathbf{1}_2 \tag{3.1}$$

which is attained if $\{\mathbf{r}_k\}_1^3$ are the position vectors of the vertices of an equilateral triangle and have the same magnitude. The link lengths of this manipulator are $a_1 = a_2 = a, a_3 = \sqrt{3}a/3$, in the configurations shown in Fig. 3.1, i.e.,

$$\theta_2 = 120^\circ, \quad \theta_3 = 150^\circ$$

and,

$$\theta_2 = -120^\circ, \quad \theta_3 = -150^\circ$$

Moreover, θ_1 can be assigned arbitrarily, since it does not affect the condition number, as it amounts to a rigid-body rotation of the overall manipulator.

The characteristic length is obtained readily from eq.(3.1) and the above conditions, namely, as

$$\frac{a}{L} = \sqrt{6} \quad \text{or} \quad L = \frac{\sqrt{6}}{6}a$$

Now, the configurations of minimum condition number of a 3-axis planar manipulator with all link lengths equal are determined. The product $\mathbf{K}_p\mathbf{K}_p^T$ is written as

$$\mathbf{K}_p\mathbf{K}_p{}^T = \begin{bmatrix} 3 & f_1 & f_2 \\ f_1 & f_3 & f_4 \\ f_2 & f_4 & f_5 \end{bmatrix}$$

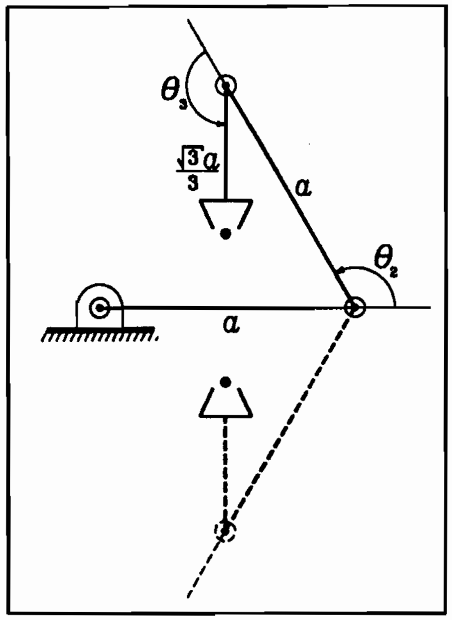


Figure 3.1: Isotropic configurations of 3-axis planar manipulator

which, in this case, cannot be rendered proportional to the identity matrix. We will try then to approximate the isotropy conditions with a least-square error. These conditions lead to five equations, namely,

$$
\begin{aligned}
f_1 &= 2s_2 + 3s_{23} = 0 \\
f_2 &= 2c_2 + 3c_{23} + 1 = 0 \\
f_3 &= (s_2 + s_{23})[1 + 2(c_2 + c_{23}) + c_{23}s_{23}] = 0 \\
f_4 &= \lambda^2[2(s_2 + s_{23})^2 + s_{23}^2] - 3 = 0 \\
f_5 &= \lambda^2[(1 + c_2 + c_{23})^2 + (c_2 + c_{23})^2 + c_{23}^2] - 3 = 0
\end{aligned}
$$

Because this design is not isotropic, all five equations of the above system cannot be satisfied simultaneously, and hence, we have an overdetermined nonlinear system of five equations with three unknowns, namely θ_2, θ_3 and $\lambda \equiv 1/L$. The Newton-Gauss procedure is used to derive a least-square approximation to the system and the solution is obtained in a few iterations. Two symmetric solutions are found, namely, $\theta_2 = 81.035$, $\theta_3 = 158.512$ and $\theta_2 = -81.035$, $\theta_3 = 201.488$. The minimum condition number derived is $\kappa_{min} = 2.3$, whereas the characteristic length is derived as $0.51258577a$. The solutions are shown in Fig. 3.2.

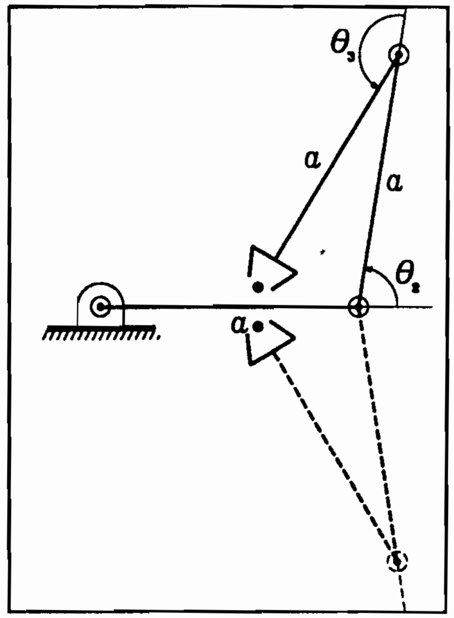


Figure 3.2: Best configurations of 3-axis planar manipulator with identical link lengths

3.2.2 General Six-Axis Manipulators

Here we find the configurations of minimum condition number of general six-axis manipulators. We wish to obtain a configuration such that the symmetric matrix $\mathbf{KK}^T$ will be as close to a multiple of the 6×6 identity matrix as possible. Thus, we will minimize the Frobenius norm of the matrix difference $\mathbf{M}$, defined as,

$$\mathbf{M} \equiv \mathbf{KK}^T - \sigma^2\mathbf{1} \tag{3.2}$$

where

$$\mathbf{K} = \begin{bmatrix} \mathbf{A} \\ \frac{1}{L}\mathbf{B} \end{bmatrix}$$

with $\mathbf{A}$ and $\mathbf{B}$ defined as the 3×6 subblocks given below:

$$\mathbf{A} \equiv [\mathbf{e}_1 \quad \mathbf{e}_2 \quad \dots \quad \mathbf{e}_6] \quad \mathbf{B} \equiv [\mathbf{e}_1 \times \mathbf{r}_1 \quad \mathbf{e}_2 \times \mathbf{r}_2 \quad \dots \quad \mathbf{e}_6 \times \mathbf{r}_6]$$

Matrix M can then be written as

$$\mathbf{M} = \begin{bmatrix} \mathbf{AA}^T & \frac{1}{L}\mathbf{AB}^T \\ \frac{1}{L}\mathbf{BA}^T & \frac{1}{L^2}\mathbf{BB}^T \end{bmatrix} - \begin{bmatrix} \sigma^2\mathbf{1}_3 & \mathbf{0} \\ \mathbf{0} & \sigma^2\mathbf{1}_3 \end{bmatrix}$$

where $\mathbf{1}_3$ is the 3×3 identity matrix, so that isotropy is achieved if $\mathbf{M}$ is rendered zero. Its upper-left block vanishes if

$$\mathbf{A}\mathbf{A}^T = \sum_1^6 \mathbf{e}_k \mathbf{e}_k^T = \sigma^2 \mathbf{1}_3$$

Taking the trace of both sides of the above equation, and noting that

$$\mathrm{tr}(\sum_1^6 \mathbf{e}_k \mathbf{e}_k^T) = 6 = \sigma^2 3$$

we readily derive

$$\sigma^2 = 2$$

Unlike the previous derivations, here it will be very cumbersome to expand the entries of $\mathbf{A}$ and $\mathbf{B}$ explicitly and replace them into eq.(3.2). Instead, we use a numerical procedure to minimize the Frobenius norm of matrix $\mathbf{M}$. Since $\mathbf{M}$ is a 6×6 symmetric matrix, we will be interested only in its 21 entries displayed below:

$$\mathbf{M} = \begin{bmatrix} f_1 & f_2 & f_3 & f_4 & f_5 & f_6 \\ & f_7 & f_8 & f_9 & f_{10} & f_{11} \\ & & f_{12} & f_{13} & f_{14} & f_{15} \\ & \mathrm{sym} & & f_{16} & f_{17} & f_{18} \\ & & & & f_{19} & f_{20} \\ & & & & & f_{21} \end{bmatrix}$$

Thus, the objective function to minimize is the norm of vector $\mathbf{f}$, whose components are $f_1, \ldots, f_{21}$. In order to render the Jacobian non-dimensional, we include the characteristic length L as a further design variable. Because the reciprocal of L yields a simpler partial derivative of $\mathbf{K}$, $\lambda \equiv 1/L$ is chosen as a design variable instead. Furthermore, we introduce the HD parameters of the last link, a_6 and b_6, as additional design variables to determine the position of point P_C. It is noted that α_6 does not affect the condition number of the Jacobian, and hence, it is not considered as a design variable.

In the most general case, then, we have 21 nonlinear equations and 8 unknowns, the objective being to minimize the Euclidean norm of vector $\mathbf{f}$, and thus finding a configuration which yields a velocity Jacobian of minimum condition number. This problem is solved by resorting to the Newton-Gauss procedure, which requires the gradient of $\mathbf{f}$ with respect to the design variables. The design variables of the problem are contained in vector $\mathbf{x}$, defined as

$$\mathbf{x} = [\theta_2 \;\; \theta_3 \;\; \theta_4 \;\; \theta_5 \;\; \theta_6 \;\; \lambda \;\; a_6 \;\; b_6]^T \tag{3.3}$$

The aforementioned gradient is thus the 21×8 matrix $\mathbf{F}$ defined below:

$$\mathbf{F} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} = [\frac{\partial \mathbf{f}}{\partial \theta_2} \;\; \frac{\partial \mathbf{f}}{\partial \theta_3} \;\; \frac{\partial \mathbf{f}}{\partial \theta_4} \;\; \frac{\partial \mathbf{f}}{\partial \theta_5} \;\; \frac{\partial \mathbf{f}}{\partial \theta_6} \;\; \frac{\partial \mathbf{f}}{\partial \lambda} \;\; \frac{\partial \mathbf{f}}{\partial a_6} \;\; \frac{\partial \mathbf{f}}{\partial b_6}]$$

From the above discussion it is apparent that, for each column of $\mathbf{F}$, we need to calculate

$$\frac{\partial \mathbf{M}}{\partial x_i} = \mathbf{K}\frac{\partial \mathbf{K}^T}{\partial x_i} + \frac{\partial \mathbf{K}}{\partial x_i}\mathbf{K}^T = \mathbf{P}_i + \mathbf{P}_i{}^T$$

In order to compute the gradient of the Jacobian $\mathbf{K}$ with respect to the joint angles, we use the basic relations (Angeles and López-Cajún, 1988),

$$\frac{\partial \mathbf{e}_j}{\partial \theta_i} = \begin{cases} \mathbf{e}_i \times \mathbf{e}_j, & \text{if } i > j; \\ 0, & \text{otherwise.} \end{cases}$$

and,

$$\frac{\partial \mathbf{r}_j}{\partial \theta_i} = \begin{cases} \mathbf{e}_i \times \mathbf{r}_j & \text{if } i > j; \\ \mathbf{e}_j \times \mathbf{r}_j & \text{otherwise.} \end{cases}$$

Thus, we have

$$\frac{\partial \mathbf{K}}{\partial \theta_i} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \cdots & \mathbf{g}_j & \cdots & \mathbf{g}_6 \\ \frac{1}{L}\mathbf{e}_1 \times \mathbf{v}_1 & \frac{1}{L}\mathbf{e}_1 \times \mathbf{v}_1 & \cdots & \frac{1}{L}\mathbf{c}_j & \cdots & \frac{1}{L}\mathbf{c}_6 \end{bmatrix}$$

where

$$\begin{aligned} \mathbf{g}_j &= \mathbf{e}_i \times \mathbf{e}_j, \qquad j = i+1, \ldots 6 \\ \mathbf{h}_j &= \mathbf{e}_i \times \mathbf{r}_j \\ \mathbf{c}_j &= \mathbf{g}_j \times \mathbf{r}_j + \mathbf{e}_j \times \mathbf{h}_j \\ \mathbf{v}_j &= \mathbf{e}_i \times \mathbf{r}_i \end{aligned}$$

As mentioned above, the partial derivative of $\mathbf{K}$ with respect to λ is very simple to express, namely,

$$\frac{\partial \mathbf{K}}{\partial \lambda} = \begin{bmatrix} \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{e}_1 \times \mathbf{r}_1 & \cdots & \mathbf{e}_6 \times \mathbf{r}_6 \end{bmatrix}$$

The gradient with respect to a_6 and b_6 is derived next. Because a change in a_6 and b_6 does not cause any change in the direction of joint axes, we have,

$$\frac{\partial \mathbf{e}_i}{\partial a_6} = \frac{\partial \mathbf{e}_i}{\partial b_6} = 0 \qquad i = 1, \ldots, 6$$

On the other hand, the displacement vectors all include vector $\mathbf{a}_6$, namely,

$$\begin{aligned} \mathbf{r}_i &= \mathbf{a}_i + \ldots + \mathbf{a}_6 \qquad i = 1, \ldots, 5 \\ \mathbf{r}_6 &= \mathbf{a}_6 \end{aligned}$$

where $\mathbf{a}_6$, in frame-6 coordinates, is given by

$$[\mathbf{a}_6]_6 = \begin{bmatrix} a_6 \cos(\theta_6) \\ a_6 \sin(\theta_6) \\ b_6 \end{bmatrix}$$

and none of $\mathbf{a}_1, \ldots, \mathbf{a}_5$ is a function of a_6 and b_6. We have then,

$$\frac{\partial [\mathbf{r}_i]_1}{\partial a_6} = \mathbf{Q}_1 \ldots \mathbf{Q}_5 \frac{\partial [\mathbf{a}_6]_6}{\partial a_6}, \quad \frac{\partial [\mathbf{r}_i]_1}{\partial b_6} = \mathbf{Q}_1 \ldots \mathbf{Q}_5 \frac{\partial [\mathbf{a}_6]_6}{\partial b_6}, \qquad i = 1, \ldots, 6$$

where

$$\frac{\partial [\mathbf{a}_6]_6}{\partial a_6} = \begin{bmatrix} \cos(\theta_6) \\ \sin(\theta_6) \\ 0 \end{bmatrix}, \qquad \frac{\partial [\mathbf{a}_6]_6}{\partial b_6} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \qquad i = 1, \ldots, 6$$

which then yield the expressions below:

$$\frac{\partial \mathbf{K}}{\partial a_6} = \begin{bmatrix} 0 & \ldots & 0 \\ \mathbf{e}_1 \times \frac{\partial \mathbf{r}_1}{\partial a_6} & \ldots & \mathbf{e}_6 \times \frac{\partial \mathbf{r}_6}{\partial a_6} \end{bmatrix}, \quad \frac{\partial \mathbf{K}}{\partial b_6} = \begin{bmatrix} 0 & \ldots & 0 \\ \mathbf{e}_1 \times \frac{\partial \mathbf{r}_1}{\partial b_6} & \ldots & \mathbf{e}_6 \times \frac{\partial \mathbf{r}_6}{\partial b_6} \end{bmatrix}$$

thereby completing all the derivatives needed to obtain the gradient $\mathbf{F}$.

3.3 Spherical Wrists

A spherical wrist contains three revolutes whose axes intersect at one point. In this section, the condition number of the wrist will be calculated in terms of design variables α_1, α_2 and the joint variable θ_2, as shown in Fig. 3.3. The said variables are defined in accordance with the HD notation. The first and last joint angles, namely, θ_1 and θ_3, as well as the

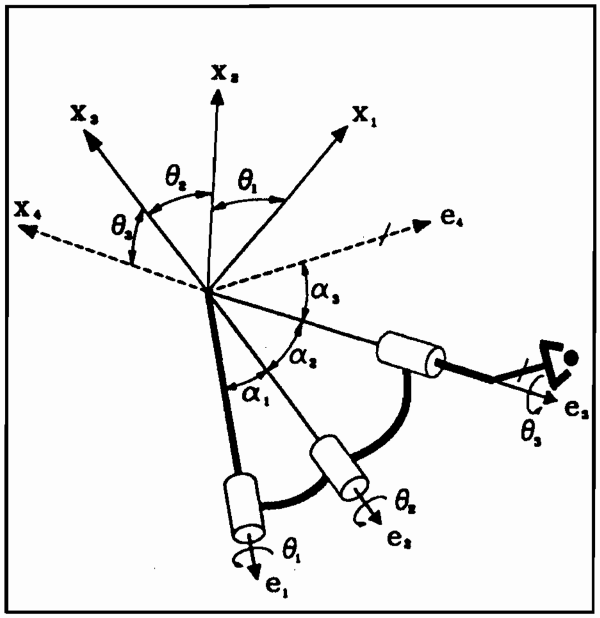


Figure 3.3: A spherical wrist with associated HD parameters

remaining HD parameter, α_3, produce a rigid-body rotation of the overall manipulator, thus not affecting the condition number of the orientation Jacobian that is shown below

$$\mathbf{K}_w = [\,\mathbf{e}_1 \quad \mathbf{e}_2 \quad \mathbf{e}_3\,]$$

We now express the Jacobian in the first frame, namely

$$\mathbf{K}_w = \begin{bmatrix} 0 & \sin\theta_1 \sin\alpha_1 & \cos\theta_1 \sin\theta_2 \sin\alpha_2 + \sin\theta_1 \cos\theta_2 \cos\alpha_1 \sin\alpha_2 + \sin\theta_1 \sin\alpha_1 \cos\alpha_2 \\ 0 & -\cos\theta_1 \sin\alpha_1 & \sin\theta_1 \sin\theta_2 \sin\alpha_2 - \cos\theta_1 \cos\theta_2 \cos\alpha_1 \sin\alpha_2 - \cos\theta_1 \sin\alpha_1 \cos\alpha_2 \\ 1 & \cos\alpha_1 & -\cos\theta_2 \sin\alpha_1 \sin\alpha_2 + \cos\alpha_1 \cos\alpha_2 \end{bmatrix} \tag{3.4}$$

Thus, we have

$$\mathbf{K}_w{}^T \mathbf{K}_w = \begin{bmatrix} 1 & \cos\alpha_1 & p \\ \cos\alpha_1 & 1 & \cos\alpha_2 \\ p & \cos\alpha_2 & 1 \end{bmatrix}$$

where,

$$p \equiv \cos\alpha_1 \cos\alpha_2 - \sin\alpha_1 \sin\alpha_2 \cos\theta_2$$

We derive the inverse of the above matrix and use eq.(1.1) to calculate the reciprocal of κ, denoted as k, namely,

$$k^2 = \frac{3(1 - \cos^2\alpha_1 - \cos^2\alpha_2 + 2p\cos\alpha_1\cos\alpha_2 - p^2)}{1 + \sin^2\alpha_1 + \sin^2\alpha_2 - p^2} \tag{3.5}$$

The variable k is used here, since its range is limited to the interval between zero and unity. We now derive from eq.(3.5) a quadratic equation in θ_2, for a given condition number, i.e.,

$$a\cos^2\theta_2 - 2b\cos\theta_2 + c = 0 \tag{3.6}$$

with the definitions given below:

$$\begin{aligned}
a &= (k^2-3)\sin^2\alpha_1\sin^2\alpha_2 \\
b &= \cos\alpha_1\cos\alpha_2\sin\alpha_1\sin\alpha_2 k^2 \\
c &= 3(1-\cos^2\alpha_1-\cos^2\alpha_2+\cos^2\alpha_1\cos^2\alpha_2) - k^2(1+\sin^2\alpha_2+\sin^2\alpha_1-\cos^2\alpha_1\cos^2\alpha_2)
\end{aligned}$$

In order to derive θ_2 uniquely from eq. (3.6), we use the half-angle relation

$$\cos(\theta_2) = \frac{1-t^2}{1+t^2}, \qquad t \equiv \tan\frac{\theta_2}{2}$$

thereby obtaining

$$(a+2b+c)t^4 - 2(a-c)t^2 + a - 2b + c = 0 \tag{3.7}$$

so that,

$$t^2 = \frac{a-c\pm\sqrt{b^2-ac}}{a+2b+c}$$

From the latter expression for t^2, we can derive either zero, two or four real values of θ_2, for a given k. It is evident that, for some k, real solutions may not be possible.

As shown in (Angeles and Rojas, 1987), an isotropic wrist has $\alpha_1 = \alpha_2 = 90°$, i.e., the axes of the neighbouring revolutes are perpendicular to each other, thus constituting an *orthogonal wrist*. Furthermore, in the same reference, it was also found that $|\theta_2| = 90°$ leads to isotropy, a result which can be verified with eq.(3.7).

3.3.1 Isoconditioning Loci of Spherical Wrists

As discussed above, once the intermediate revolute is locked, θ_2 is given a prescribed value, and the condition number of the wrist will remain constant for any value of θ_1 and θ_3. The set of configurations attained by the EE, for a constant condition number, defines a manifold in the space of either linear or quadratic invariants, that is referred to as the *isoconditioning*

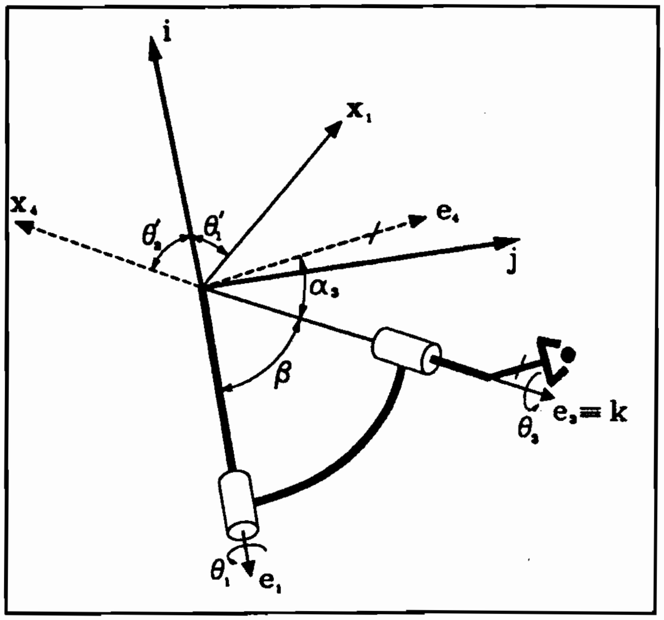


Figure 3.4: The spherical wrist with two joint angles

locus. Because the overall rotation under these conditions does not depend on θ_2, the second frame is removed and the third frame is redefined as $\{\mathbf{i}, \mathbf{j}, \mathbf{k}\}$, as shown in Fig. 3.4. Moreover, the HD parameters associated with the introduced frame are defined as (β, θ_1') and (α_3, θ_3'). Consistent with the HD convention, the introduced frame is defined such that, $\mathbf{k} \equiv \mathbf{e}_3$, and $\mathbf{i}$ is perpendicular to the plane of $\mathbf{e}_1$ and $\mathbf{e}_3$. Thus

$$\mathbf{i} \equiv \frac{\mathbf{e}_1 \times \mathbf{e}_3}{\sin\beta}, \quad \mathbf{j} \equiv \mathbf{e}_3 \times \mathbf{i}$$

where β is the angle between $\mathbf{e}_1$ and $\mathbf{e}_3$, and $\mathbf{j}$ is perpendicular to $\mathbf{i}$ and $\mathbf{k}$. Noting that α_3 is the angle between $\mathbf{e}_3$ and $\mathbf{e}_4$, as it was before the introduction of the intermediate frame, the aim now is to compute β, θ_1' and θ_3' in terms of $\alpha_1, \alpha_2, \alpha_3, \theta_1, \theta_2$ and θ_3. Now, β can be readily computed from the cosine and sine expressions below,

$$\cos\beta = \mathbf{e}_1 \cdot \mathbf{e}_3 = p$$

where $\mathbf{e}_1$ and $\mathbf{e}_3$ are readily available from eq.(3.4). Similarly

$$\sin\beta = \|\mathbf{e}_1 \times \mathbf{e}_3\| = \sqrt{\sin^2\alpha_2 \sin^2\theta_2 + (\sin\alpha_1 \cos\alpha_2 + \cos\alpha_1 \sin\alpha_2 \cos\theta_2)^2} > 0$$

The computation of θ_1' and θ_3' is shown in Appendix H. The expressions derived are reproduced below for quick reference.

$$
\begin{aligned}
\cos\theta_1' &= \frac{1}{\sin\beta}\left[\frac{\cos\theta_1}{\sin\alpha_1}(\cos\alpha_2 - \cos\alpha_1\cos\beta) - \sin\theta_1\sin\theta_2\sin\alpha_2\right] \\
\sin\theta_1' &= \frac{1}{\sin\theta_1}\left[-\cos\theta_1\cos\theta_1' + \frac{\cos\alpha_2 - \cos\alpha_1\cos\beta}{\sin\alpha_1\sin\beta}\right] \\
\cos\theta_3' &= \frac{1}{\sin\beta}\left[\frac{\cos\theta_3}{\sin\alpha_2}(\cos\alpha_1 - \cos\alpha_2\cos\beta) - \sin\theta_2\sin\theta_3\sin\alpha_1\right] \\
\sin\theta_3' &= \frac{1}{\sin\theta_3}\left[-\cos\theta_3\cos\theta_3' + \frac{\cos\alpha_1 - \cos\alpha_2\cos\beta}{\sin\alpha_2\sin\beta}\right]
\end{aligned}
$$

Having determined the above angles, the wrist rotation can be expressed using a single composition of individual linear or quadratic invariants. The linear invariants of each rotation, denoted with superscripts A and B, are expressed as

$$
\mathbf{q}^A = \frac{1}{2}\begin{bmatrix} \sin\beta(1+\cos\theta_1') \\ \sin\theta_1'\sin\beta \\ \sin\theta_1'(1+\cos\beta) \end{bmatrix} \qquad q_0^A = \frac{1}{2}(\cos\theta_1' + \cos\beta + \cos\theta_1'\cos\beta - 1)
$$

$$
\mathbf{q}^B = \frac{1}{2}\begin{bmatrix} \sin\alpha_3(1+\cos\theta_3') \\ \sin\theta_3'\sin\alpha_3 \\ \sin\theta_3'(1+\cos\alpha_3) \end{bmatrix} \qquad q_0^B = \frac{1}{2}(\cos\theta_3' + \cos\alpha_3 + \cos\theta_3'\cos\alpha_3 - 1)
$$

The quadratic invariants are derived from the linear invariants as

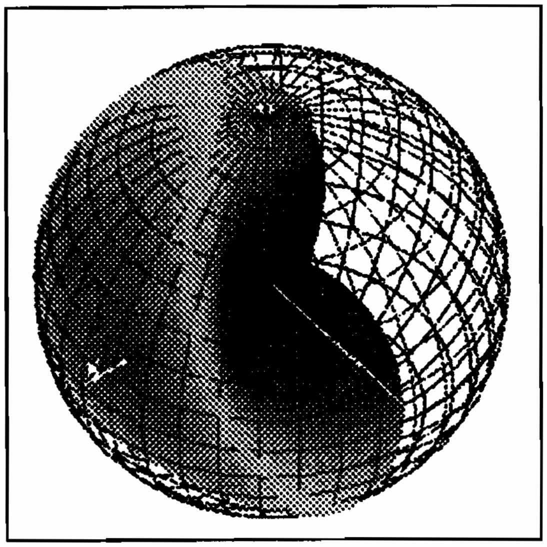
$$
\hat{q}_0^A = \sqrt{\frac{1+q_0^A}{2}} \qquad \hat{\mathbf{q}}^A = \mathbf{q}^A\frac{\sqrt{2(1+{q^A}_0)}}{2(1+{q^A}_0)}
$$

$$
\hat{q}_0^B = \sqrt{\frac{1+q_0^B}{2}} \qquad \hat{\mathbf{q}}^B = \mathbf{q}^B\frac{\sqrt{2(1+{q^B}_0)}}{2(1+{q^B}_0)}
$$

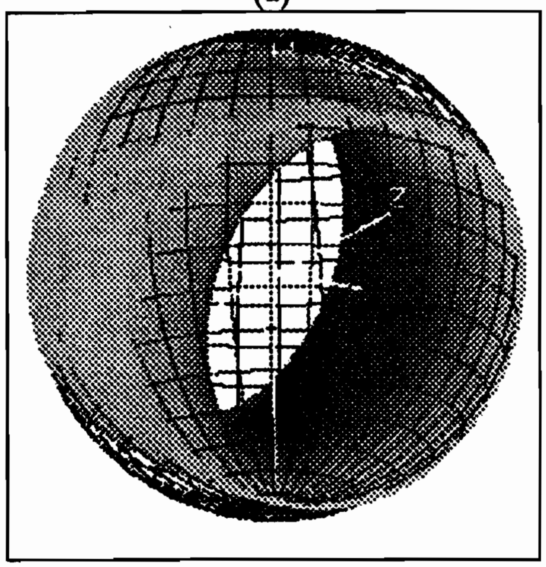
Finally, the invariants of the product are derived from the above relations using the vector composition techniques outlined in section 2.1.3.

Graphical Display of the Isoconditioning Loci

As an example, we display the isoconditioning loci of a three-roll spherical wrist, for which $\alpha_1 = \alpha_2 = 120°$. This wrist was found to have a minimum condition number of 1.197175, which corresponds to a value of $k = 0.8353$ (Angeles and López-Cajún, 1988) with the intermediate angle locked to $\theta_2 = 95.652$. Such loci corresponds to configurations lying farthest away from singularities. Figs. 3.5a-b show these loci in the space of linear and quadratic invariants, respectively.



(a)



(b)

Figure 3.5: Isoconditioning Loci of the three-roll wrist a) $\mathbf{q}$ space b) $\hat{\mathbf{q}}$ space

Chapter 4

Robotics Applications of CINVERSE

4.1 Off-Line Robot Kinematics

4.1.1 Trajectory Planning: TRAJ_PLAN Module

Industrial robots are most often used in continuous-path applications such as arc-welding, cutting and materials handling. In these applications, the position and orientation of the EE is required to undergo a gradual change between adjacent points on the traced curve. The smoothness of the orientation is ensured by expressing the orientation with *Frenet-Serret frames* at each point. These frames are composed of three orthonormal vectors representing the unit tangent, normal and binormal vectors along the curve. The said orientation is represented by a rotation matrix of the form

$$\mathbf{Q}_g = [\mathbf{e}_b \quad \mathbf{e}_n \quad \mathbf{e}_t]$$

where b, n, t stand for binormal, normal and tangent directions to the chosen curve, respectively. The application of CINVERSE to off-line trajectory planning is illustrated with an example below.

The example chosen is about arc welding along a curve defined by the intersection of two cylinders as shown in Fig. 4.1.

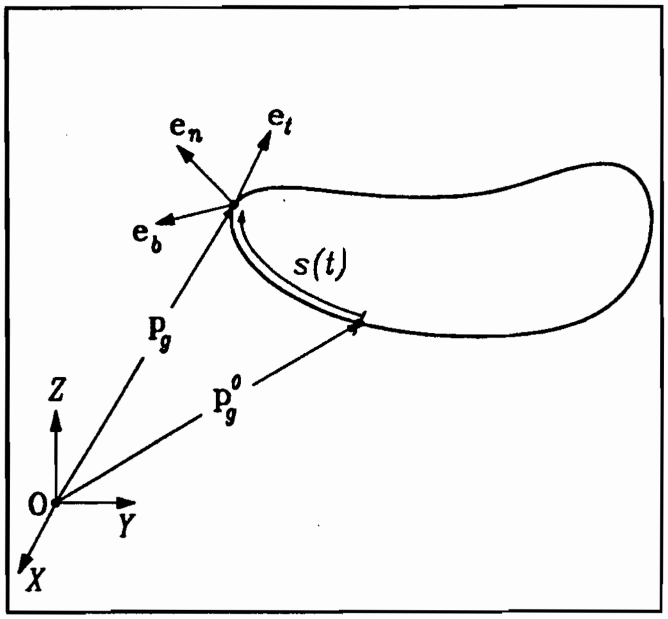


Figure 4.1: Intersection of two cylinders

Calculation of the Prescribed Data for DIK

The intersection of two cylinders is computed and parametrized as shown in Appendix I. R_1 and R_2 denote the radii of the cylinders used, and γ is the angle between the axes of intersection of the two cylinders, whereas (X_o, Y_o, Z_o) denote the Cartesian coordinates of the point of intersection.

In our example, the cylinders have radii of 0.4 and 0.5 meters, and their axes intersect at 60 degrees. Moreover, the point of intersection is located at $(0.5, 0.7, 0.2)$ meters.

The position vector of a point in the curve is thus expressed as

$$\mathbf{p}_g = \mathbf{p}_g = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} R_1 \cos\beta + X_o \\ R_1 \sin\beta + Y_o \\ \frac{R_1 \cos\gamma \sin\beta + \sqrt{R_2^2 - (R_1 \cos\beta)^2}}{\sin\gamma} + Z_o \end{bmatrix} \tag{4.1}$$

Here, β is a parameter that is chosen to be a smooth function of time, t, having the following form (Fig. (4.2)

$$\beta(t) = \frac{2\pi}{10}t - \sin(\frac{2\pi}{10}t)$$

As t is incremented by 0.1 seconds from 0. to 10. seconds, 100 sample points of the curve are obtained.

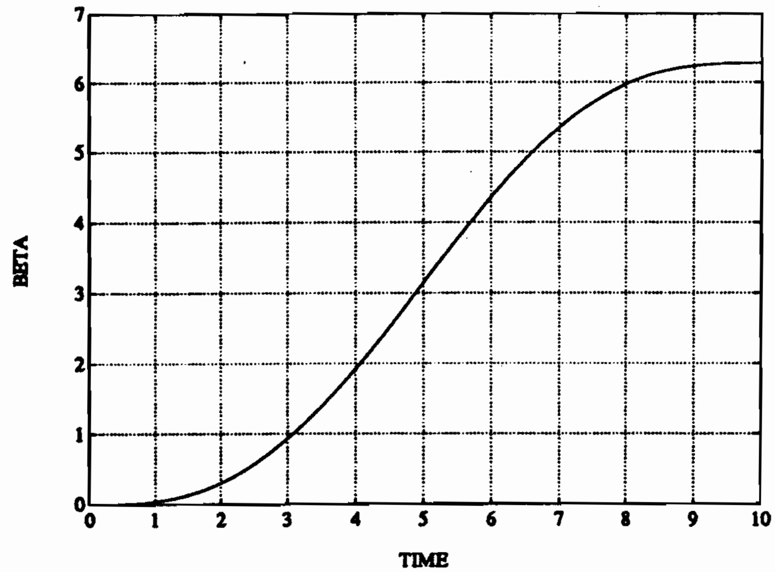


Figure 4.2: Parameter β as a function of time

The above expression constitutes the prescribed position vector of a point on the curve that is chosen. The orientation matrix $\mathbf{Q}_g$ is derived next. First, we have (Angeles, Rojas and López-Cajún, 1988)

$$\begin{aligned} \mathbf{e}_t &= \frac{d\mathbf{p}_g}{ds} \\ \mathbf{e}_b &= \frac{\frac{d\mathbf{p}_g}{ds} \times \frac{d^2\mathbf{p}_g}{ds^2}}{\|\frac{d\mathbf{p}_g}{ds} \times \frac{d^2\mathbf{p}_g}{ds^2}\|} \\ \mathbf{e}_n &= \mathbf{e}_t \times \mathbf{e}_b \end{aligned}$$

where $s = s(t)$ is a measure of the arc length on the intersection curve. Moreover, $\mathrm{d}\mathbf{p}_g/\mathrm{d}s$ and $\mathrm{d}^2\mathbf{p}_g/\mathrm{d}s^2$ require computation of $\mathrm{d}\mathbf{p}_g/\mathrm{d}\beta$ and $\mathrm{d}^2\mathbf{p}_g\mathrm{d}\beta^2$ which are readily available from eq.(4.1).

Calculation of the Prescribed Data for VIK

For VIK, the angular velocity and the rate of change of the position vector of the EE are required.

Now, the time rate of change of $\mathbf{p}_g$ is calculated readily as

$$\dot{\mathbf{p}}_g = \frac{d\mathbf{p}_g}{dt} = \frac{d\beta}{dt}\frac{d\mathbf{p}_g}{d\beta}$$

The derivation of the angular velocity requires the computation of the *Darboux* vector, $\boldsymbol{\delta}$, (Angeles, Rojas and López-Cajún, 1988), i.e.,

$$
\begin{aligned}
\boldsymbol{\omega}_g &= \dot{s}\boldsymbol{\delta} \\
\boldsymbol{\delta} &= \tau \mathbf{e}_t + \kappa \mathbf{e}_s
\end{aligned}
$$

Here, $\dot{s}$ is the speed of the EE along the intersection curve, calculated as

$$\dot{s} = \frac{ds}{d\beta}\frac{d\beta}{dt}$$

where

$$\frac{ds}{d\beta} = \|\frac{d\mathbf{p}_g}{d\beta}\|$$

Furthermore, τ and κ are the *torsion* and the *curvature* of the intersection curve, and are functions of higher derivatives of $d\mathbf{p}_g/ds$, as shown in the same reference.

Calculation of the Prescribed Data for AIK

For the solution of AIK, we require the second derivative of $\mathbf{p}_g$ with respect to time, as well as the time derivative of the angular velocity. These expressions are derived from,

$$
\begin{aligned}
\ddot{\mathbf{p}}_g &= \frac{d^2\mathbf{p}_g}{dt^2} = \frac{d^2\beta}{dt^2}\frac{d\mathbf{p}_g}{dt} + \left(\frac{d\beta}{dt}\right)^2 \frac{d^2\mathbf{p}_g}{d\beta^2} \\
\dot{\boldsymbol{\omega}}_g &= \ddot{s}\boldsymbol{\delta} + \dot{s}\dot{\boldsymbol{\delta}}
\end{aligned}
$$

where $\ddot{s}$ is calculated as

$$\ddot{s} = \frac{d^2 s}{d\beta^2}\frac{d\beta}{dt} + \frac{ds}{d\beta}\frac{d^2\beta}{dt^2}$$

The time derivative, $\dot{\boldsymbol{\delta}}$ is derived as,

$$\dot{\boldsymbol{\delta}} = \dot{\tau}\mathbf{e}_t + \tau\dot{\mathbf{e}}_t + \dot{\kappa}\mathbf{e}_b + \kappa\dot{\mathbf{e}}_b$$

where expressions for $\dot{\tau}$ and $\dot{\kappa}$ involve higher derivatives of $d\mathbf{p}_g/ds$ as shown in (Angeles, Rojas and López-Cajún, 1988). Moreover $\dot{\mathbf{e}}_t$ and $\dot{\mathbf{e}}_b$ are derived as

$$
\begin{aligned}
\dot{\mathbf{e}}_t &= s\kappa\mathbf{e}_n \\
\dot{\mathbf{e}}_b &= -\dot{s}\tau\mathbf{e}_n
\end{aligned}
$$

Table 4.1: Home configuration of four industrial robots

Design Variable	$YaskawaAid810$	$Puma560$	$FanucArcMate$	$AseaIrb6/2$
θ_2 (deg)	78.96	74.10	97.70	103.82
θ_3 (deg)	-27.53	-201.19	-46.53	-150.70
θ_4 (deg)	-40.5	-136.49	26.37	-37.56
θ_5 (deg)	116.45	-113.19	-72.36	-114.85
θ_6 (deg)	-6.53	166.07	135.76	-13.79
L (mm)	423.522	226.389	296.837	298.665
a_6 (mm)	382.959	175.166	223.585	282.728
b_6 (mm)	-397.864	214.312	274.221	-275.054
κ	1.692666	1.665548	1.591313	1.767348

The TRAJ_PLAN module is applicable to any trajectory that can be parametrized, provided that up to fourth order derivatives of the position vector with respect to that parameter exist. As output, the TRAJ_PLAN module calculates $\mathbf{p}_g, \mathbf{Q}_g, \dot{\mathbf{p}}_g, \boldsymbol{\omega}_g, \ddot{\mathbf{p}}_g$ and $\dot{\boldsymbol{\omega}}_g$ and stores them in ASCII and Binary files. When the trajectory is the intersection of two cylinders, the data input required are i) the radii of the cylinders, ii) the angle of rotation between cylinders, γ, and iii) the offset of the intersection of the cylinders.

4.1.2 Home Configuration: HOME_CONF Module

Four industrial robots are chosen as examples, namely, *Yaskawa Aid 810, Puma 560, Fanuc Arc Mate* and *Asea Irb 6/2*. More examples can be found from (Cugy, 1983). Among the above robots, *Puma 560* and *Asea Irb 6/2* are wrist-partitioned. The HD parameters of the robots are shown in Appendix J. As shown in the said Appendix, all architectures contain identical values of $|\alpha_i|$, for $(i = 1, \ldots 6)$. However, the distances and offsets between axes, namely, a_i and b_i, for $(i = 1, \ldots 6)$, are not identical.

The numerical results for the above-mentioned robots are summarized in Table 4.1. It is noted from this table that, although none of these robots is isotropic, if their Jacobian matrices are defined at their characteristic points, the minimum condition numbers thus obtained are fairly close to unity.

Table 4.2: Home configurations disregarding the characteristic points

Design Variable	$YaskawaAid810$	$Puma560$	$FanucArcMate$	$AseaIrb6/2$
θ_2 (deg)	89.03	-83.65	121.75	88.29
θ_3 (deg)	-134.07	41.25	-58.73	-44.86
θ_4 (deg)	-182.48	-37.81	15.53	0.0
θ_5 (deg)	-48.47	199.38	-37.07	-56.09
L (mm)	653.594	294.117	366.300	458.715
κ	3.73	4.68	3.95	4.02

For comparison purposes, we include in Table 4.2 the home configurations and characteristic lengths of the same four robots, as provided by the manufacturer, i.e., disregarding their characteristic points.

The home configuration and the characteristic point of these robots are illustrated with the help of figures generated on an *IRIS 4D/210VGX* in Fig. 4.3. Moreover, Fig. 4.4 shows the home configurations of the same robots, as provided by the manufacturers.

The input needed for the HOME_CONF module are, i) a user-supplied initial guess of design parameters as indicated with eq.(3.3), and ii) the HD parameters of the manipulator. The output produced is the said design vector.

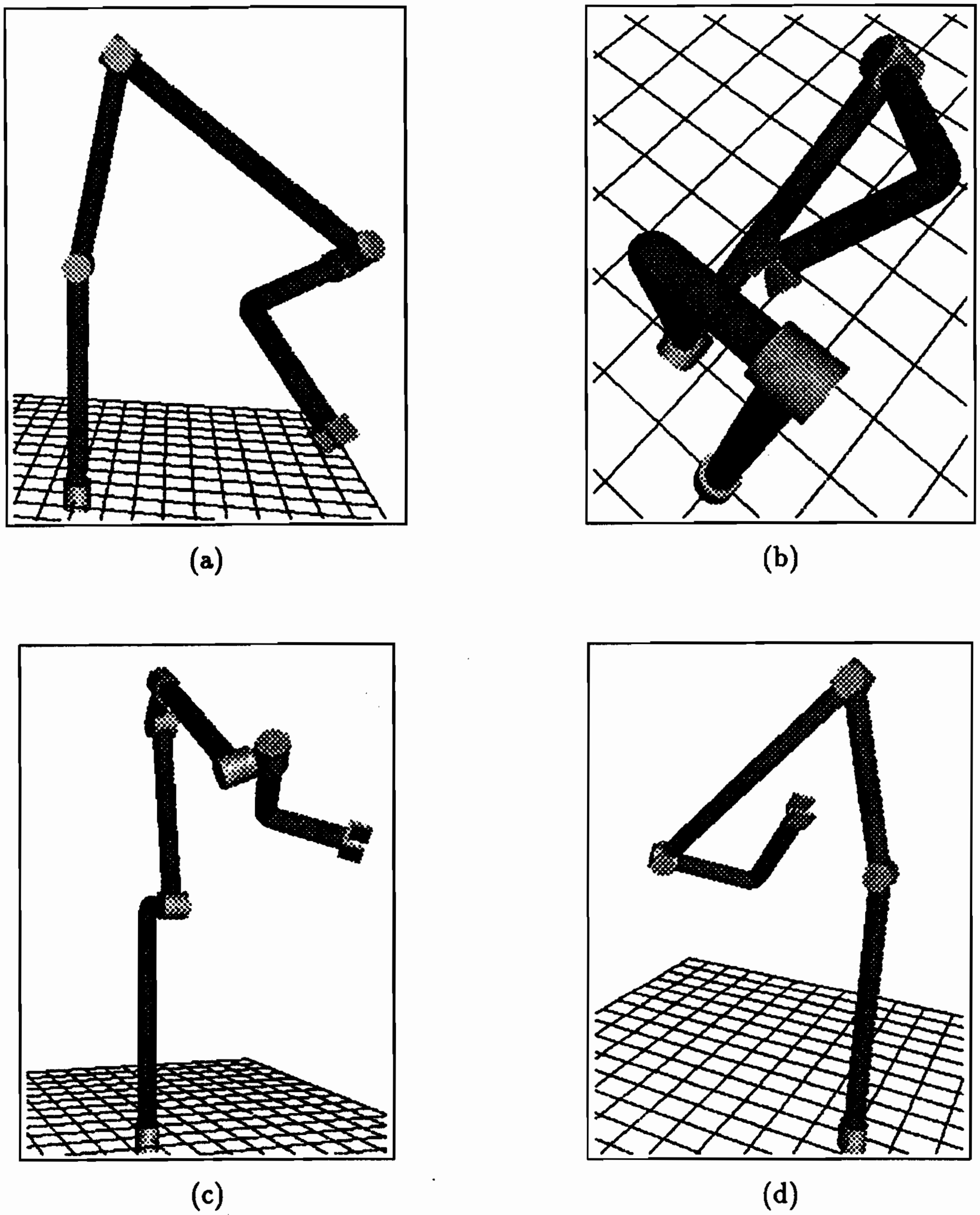


Figure 4.3: Various robots at their home configurations a) Yaskawa Aid 810 b) Puma 560 c) Fanuc Arc Mate d) Asea Irb 6/2

4.2 On-Line Robot Kinematics

4.2.1 Inverse Kinematics: INV_KIN Module

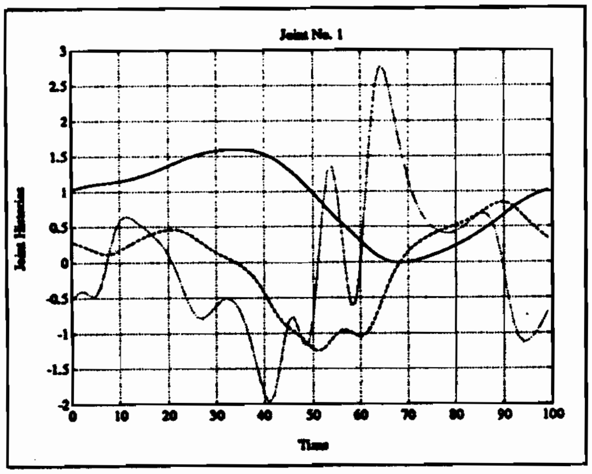
First, the data $\mathbf{p}_g$, $\dot{\mathbf{p}}_g$, $\ddot{\mathbf{p}}_g$ are non-dimensionalized through division by the characteristic length, L. Similarly, the translational terms of the HD parameters, namely, a_i and b_i, for $i = 1, \ldots 6$, are also divided by L. Moreover, the values for the the HD parameters a_6 and b_6 are assigned in accordance with the calculation of the characteristic point P_C.

For the first point on the traced curve, the home configuration obtained is used as an initial guess. For subsequent points, the solution obtained from the previous data point is used as an initial guess so that a quicker convergence can be reached and the likelihood of branch switching is reduced. The method based on the natural invariants is used in DIK not only because of its faster convergence speed, but also in order to avoid formulation singularities.

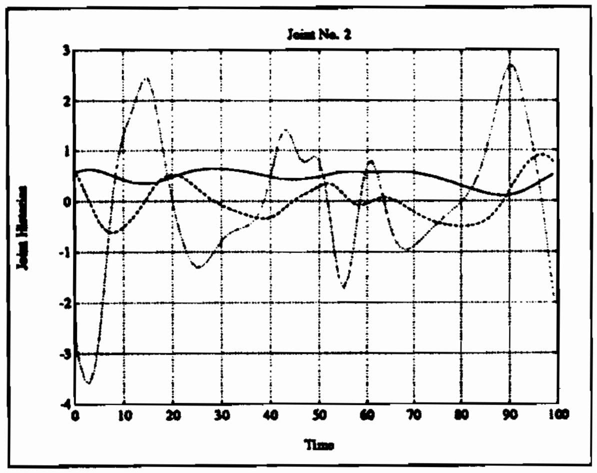
Furthermore, the joint rates for each point are determined by making use of the joint angles just derived. If the joint angles lead to a singular velocity Jacobian, the singularity handling algorithm is applied to prevent branch switching. Otherwise, a linear algebraic set of equations is solved for the joint rates. Moreover, the joint accelerations are determined by making use of the joint angles and joint rates obtained for the point in consideration. In the case of a singularity, the joint accelerations are determined using a similar numerical procedure.

Below, we display the IKP solutions obtained for all joint variables of the *Yaskawa Aid 810* robot, using the example path that was discussed earlier in the Chapter. As can be verified from Fig. 4.5a-f, in all cases, the joint rate $\dot{\theta}$ (dashed line) vanishes when the joint angle θ (solid line) is a minimum or a maximum. Similarly, $\ddot{\theta}$ (dotted line) vanishes when $\dot{\theta}$ is a minimum or a maximum.

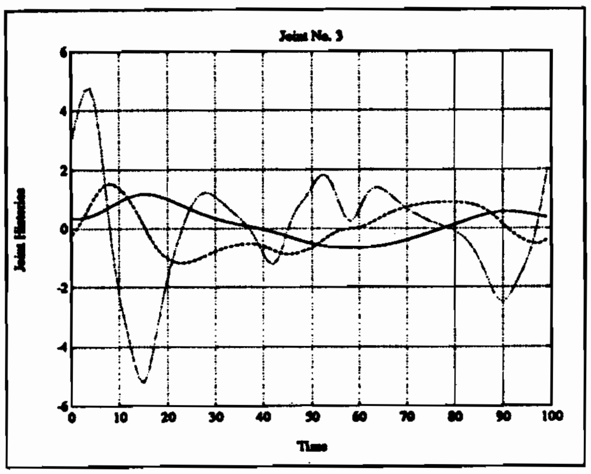
The input required for the INV_KIN module are, i) all the outputs produced from the TRAJ_PLAN and HOME_CONF modules, and ii) HD parameters of the manipulator. The output produced are joint angles, rates and accelerations.



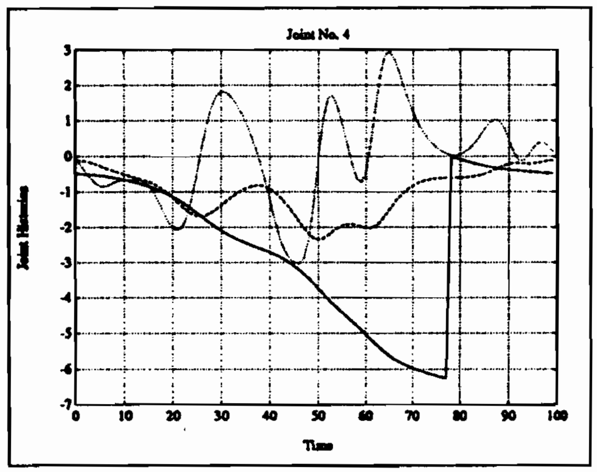
(a)



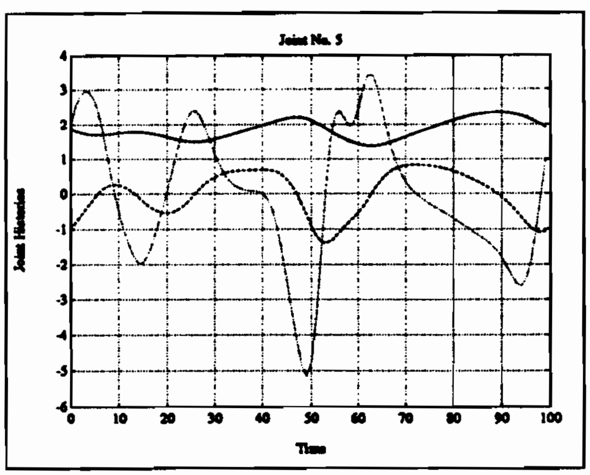
(b)



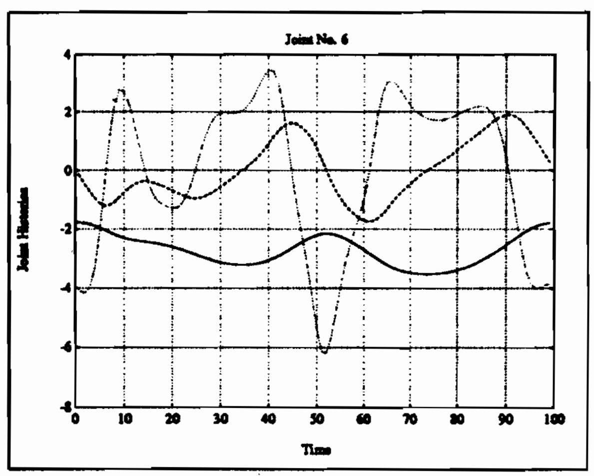
(c)



(d)



(e)



(f)

Figure 4.5: IKP solutions with *Yaskawa Aid 810*

Chapter 5

Conclusions

5.1 Concluding remarks

The forward kinematics problem has been implemented using two different methods, as described below. The first method, based on the composition of invariants, allows the implementation of rotation calculations with four scalar quantities. Thus, only vector operations are required with this method to derive the orientation expressions. The second method, which is based on matrix products, requires the use of 3×3 matrices. Although the use of invariant quantities is more elegant, it leads to more expensive algorithms, as shown in the operation count analyses. Because Newton-type numerical procedures are used to implement displacement inverse kinematics, it is always preferable to minimize the overhead in orientation calculations, since at each iteration of the numerical procedure, the invariants have to be recalculated. Because of the significant time advantage, matrix calculations are recommended over invariant compositions.

The displacement inverse kinematics problem was solved here using three different sets of invariants. Although linear invariants are computationally inexpensive, they have the disadvantage of causing ill-conditioning in the Jacobian when the angle of rotation is close to π. On the other hand, quadratic invariants are well defined for all rotation angles and do not admit formulation singularities, but their use leads to a much slower convergence rate than the use of linear invariants. The third set of invariants studied here, which we call the natural invariants, consists of the unit vector parallel to the axis of rotation and the angle of rotation. Using natural invariants, the associated Jacobian takes on a much simpler form than in the previous cases, thus reducing the overhead in the set-up time of the linear algebraic system. Furthermore, because its Jacobian is square (6×6), the method based on the natural invariants allows us to use a faster solution technique, such as LU-decomposition. Moreover, the said Jacobian is also employed in the calculation of joint rates and joint accelerations Thus, the complete inverse kinematics problem can be implemented using only one Jacobian. One more advantage of the method based on natural invariants is that it does not admit formulation singularities, the stability of the numerical procedure thus being dependent only on the configuration of the manipulator. Because of both time and numerical stability advantages, the natural invariants are preferred in the implementation of inverse kinematics.

The condition number of the Jacobian matrix of serial manipulators has been used as a measure of kinematic and numerical performance. A least-squares algorithm has been presented to minimize the condition number of a square matrix.

The condition number of a general six-axis manipulator has been optimized using the above-mentioned algorithm. The optimization is enhanced by defining a constant length that is used to render the Jacobian matrix dimensionally homogeneous. It is also shown that, by using the Hartenberg-Denavit parameters of the last frame as additional design variables, an optimum location of the end-effector can be derived that allows the condition number of the said Jacobian to approach unity. The algorithm was used to derive the best conditioned configurations of several common robots.

The conditioning of spherical wrists is also analyzed. When the condition number of a spherical wrist is kept fixed to a single value, the intermediate joint is locked, the other two thus producing a set of motions of the end-effector that span an isoconditioning surface lying inside a unit sphere. The said surface is derived in Appendix H, and example loci are displayed in Chapter 3.

5.2 Suggestions for Further Research

Further improvements in this research area should address the handling of kinematic singularities during the numerical iterations of the displacement inverse kinematics. Although any of the above-mentioned three inverse kinematics methods would come out of the singularity due to the robustness of the Newton-type methods used, a branch switching is very likely to occur in the presence of a kinematic singularity. Definitely, a branch switching has to be avoided in on–line applications, for it leads to jump discontinuities in velocities and to infinite discontinuities in accelerations.

Furthermore, when the numerical procedure converges to a solution at a singularity, the singularity-handling algorithm presented here uses a series approximation to calculate the joint angles at the next point. Since joint angles are calculated using an approximation, the error induced will be reflected to the joint rates and joint accelerations in the neighbourhood of the said singularity, thus causing sometimes noticeable deviations in joint histories. In order to avoid the deviations, the approximations obtained should be refined by introducing another solution technique that does not require the gradient of the vector function used.

References

Albala, H., 1976, *Displacement Analysis of the N-Bar, Single-Loop, Spatial Linkage*, Ph. D. Thesis, Technion-Israel Institute of Technology, Haifa.

Albala, H., 1982, "Displacement analysis of the n-bar, single-loop, spatial linkage, part 1: underlying mathematics and useful tables, part 2: basic displacement equations in matrix and algebraic form", *ASME Journal of Mechanism Design*, Vol. 104, No. 2, pp. 504-525.

Albala, H., and Angeles, J., 1979, "Numerical solution to the input-output displacement equation of the general 7R spatial mechanism", *Proceedings of Fifth World Congress on Theory of Machines and Mechanisms*, pp. 1008-1011.

Angeles, J., 1989, *Robotic Mechanical Systems, An Introduction*, Lecture Notes, Dept. of Mechanical Engineering, McGill University, Montreal.

Angeles, J., 1988, *Rational Kinematics*, Springer-Verlag, New York.

Angeles, J., 1985, "On the numerical solution of the inverse kinematics problem", *The Int. J. Robotics Res.*, Vol. 2, pp. 21-36.

Angeles, J., 1991, "Die theoretischen Grundlagen zur Behandlung algebraischer Singularitäten der kinematischen Koordinatetenumkehr in der Robotertechnik", *Mechanism and Machine Theory*, Vol. 26, No. 3, pp. 315-322.

Angeles, J., Anderson, K., Cyril, X., and Chen, B., 1988, "The kinematic inversion of robot manipulators in the presence of singularities", *Journal of Dynamic Systems, Measurement, and Control*, Vol. 110, pp. 246-254.

Angeles, J., Anderson, K. and Gosselin, C., 1987, "An orthogonal-decomposition algorithm for constrained least-square optimization", *Proc. 13th ASME Design Automation Conference*, Vol. 2, pp. 215-220.

Angeles, J. and Cyril, X., 1986,"Kinematic analysis of linkages using a robotics package", *Proceedings of 8th Symposium on the Engineering Applications of Mechanics*, pp. 107-114.

Angeles, J. and López-Cajún, C., 1988, "The dexterity index of serial-type robotic manipulators", *Proc. 20th Biennial Mechanisms Conference*, Sept. 25-28, Kissimmee, FL: pp. 79-84.

Angeles, J. and Rojas, A., 1987, "Manipulator inverse kinematics via condition-number minimization and continuation" *Int. J. of Robotics and Automation*, Vol. 2, No. 2, pp. 61-69.

Angeles, J., Rojas, A. and López-Cajún, C., 1988, "Trajectory planning in robotics continuous-path applications", *IEEE J. of Robotics and Automation*, Vol. 4, No. 4, pp. 380-385.

Angeles, J. and Tandirci, M., 1991, "The linear invariants of the composition of two rotations", *CANCAM, 13th Canadian Congress of Applied Mechanics*, Winnipeg, Vol. 2, pp. 656-657.

Quadratic Invariants of the Composition of Two Rotations", *Department of Mechanical Engineering and Research Centre for Intelligent Machines Technical Report*, McGill University, Montreal.

Chang, C., 1991, *The Graphical Determination of All Real Solutions of Nonlinear Kinematics Problems*, Honours Thesis, McGill University, Dept. of Mechanical Engineering, Montreal.

Cugy, A., 1983, *Industrial Robot Specifications*, Kogan Page Ltd., London.

Dahlquist, G. and Björck, Å., 1974, *Numerical Methods*, Prentice-Hall, Inc., Englewood Cliffs.

Duffy, J. and Crane, C., 1980, "A displacement analysis of the general spatial 7-link, 7R mechanism", *Mechanism and Machine Theory*, Vol. 15, pp. 153-169.

Eppinger, M. and Kreuzer, E., 1990, "Evaluation of methods for solving the inverse kinematics of manipulators", *Meerestechnik II-Strukturmechanik Technical Report*, Technische Universität Hamburg, Hamburg.

Euler, L., 1775a, "Formulae generales pro transitione quacunque corporum rigidorum", *Novi Commentari Acad. Imp. Petrop.*, Vol. 20, pp. 189-207.

Euler, L., 1775b, "Nova methodus motum corporum rigidorum determinandi", *Novi Commentari Acad. Imp. Petrop.*, Vol. 20, pp. 208-238.

Funda, J. and Paul, R. P., 1989, "A computational analysis of screw transformations in robotics", *IEEE Transactions on Robotics and Automation*, Vol. 6, No. 3, pp. 348-356.

Goldenberg, A. A., Benhabib, B. and Fenton, R. G., 1985, "A complete generalized solutions to the inverse kinematics of robots", *IEEE J. of Robotics and Automation*, Vol. RA-1, pp. 14-20.

Golub, G. H. and Van Loan, C. F., 1983, *Matrix Computations* The John Hopkins University Press, Baltimore.

Hartenberg, R. S. and Denavit, J., 1964, *Kinematic Synthesis of Linkages*, McGraw-Hill Book Co., New York.

Hiller, M. and Woernle, C., 1989, *Grundlagen und Anwendungen der Robotik*, Lecture Notes, Pfalzakademie, Lambrecht.

Kazerounian, K., 1987, "On the numerical inverse kinematics of robotic manipulators", *ASME J. of Mechanisms, Transmissions, and Automation in Design*, Vol. 109, pp. 8-13.

Klein, C. and Blaho, B., 1987, "Dexterity measures for the design and control of kinematically redundant manipulators", *Int. J. of Robotics Research*, Vol. 6, No. 2, pp 72-83.

Kumar, A. and Waldron, K. J., 1981, "The workspace of a mechanical manipulator", *ASME J. of Mechanism Design*, Vol. 103, No. 3, pp. 665-672.

Lee, H.-Y. and Liang, C.-G., 1988, "Displacement analysis of the general spatial 7-link 7R mechanism", *Mechanism and Machine Theory*, Vol. 23, No. 3, pp. 219-226.

Lee, H.-Y, 1990, "Ein Verfahren zur vollständigen Lösung der Rückwärtstransformation für Industrieroboter mit allgemeiner Geometrie", Ph. D. Thesis, Gesamthochschule, Duisburg, Germany.

Li, Z., 1990, "Geometrical considerations of robot kinematics", *Int. J. of Robotics and Automation*, Vol. 5, No. 3, pp. 139-145.

Paul, Richard P., 1981, *Robot Manipulators: Mathematics, Programming, and Control*, MIT Press, Cambridge.

Pieper, D.L., 1968, *The Kinematics of Manipulators Under Computer Control*, Ph. D. Thesis, Stanford University, Stanford.

Podhorodeski, R. P., 1989, *New Approaches For the Solution of Inverse Instantaneous Kinematic Problems and of Contact Forces in Multiple Contact Grasping*, Ph. D. Thesis, University of Toronto, Toronto.

Press W. H., Flannery, B. P., Teukolsky S. A. and Vetterling W. T., 1988, *Numerical Recipes in C*, Cambridge University Press, Cambridge.

Primose, E. J. F., 1986, "On the input-output equation of the general 7R-mechanism", *Mechanism and Machine Theory*, Vol. 21, No. 6, pp. 509-510.

Raghavan, M. and Roth, B., 1990, "Kinematic Analysis of the 6R Manipulator of General Geometry", *Proc. of 5th Int. Symp. on Rob. Res.*, edited by H. Miura and S. Arimoto, MIT Press Cambridge, pp. 263-269.

Rodrigues, O., 1840, "Des lois géométriques qui régissent les déplacements d'un système solide dans l'espace, et la variation des coordonnées provenant de ces déplacements considérés indépendamment des causes qui peuvent les produire." *J. de Mathématiques Pures et Aplliquées*, Vol. 5, pp. 380-440.

Salisbury, J. K. and Craig, J. J., 1982, "Articulated hands: force control and kinematic issues", *Int. J. of Robotics Res.*, Vol. 1, No. 1, pp. 4-17.

Stewart, G. W., 1973, *Introduction to Matrix Computations*, Academic Press, London.

Takano, M., 1985, "A new effective solution to inverse kinematics problem of a robot with any type of configuration", *Journal of the Faculty of Engineering*, The University of Tokyo, Vol. B, No. 2, pp. 107-135.

Tsai, L. W. and Morgan, A. P., 1985, "Solving the kinematics of the most general six and five-degree-of-freedom manipulators by continuation methods", *ASME J. of Mechanisms, Transmissions, and Automation in Design*, Vol. 107, No. 2, pp. 189-200.

Vinogradov, I. B., Kobrinski, A. E., Stepanenko Y. E., and Tives L. T., 1971, "Details of kinematics of manipulators with the method of volumes", (in Russian), *Mekhanika Mashin.*, No. 27-28, pp. 5-16.

Wampler, C. W., 1991, "A new jacobian formulation for general six-revolute manipulators", *Int. Conference on Robotics and Automation*, Vol. 2, pp. 1046-1051.

Wampler, C. W. and Morgan, A. P., 1989, "Solving 6R inverse position problem using a generic-case solution methodology", *Mechanisms and Machine Theory*, Vol. 26, No. 1, pp. 91-106.

Williams, O. R., 1989, *Kinematics and Design of Robotic Manipulators with Complex Architectures*, M. Eng. Thesis, McGill University, Dept. of Mechanical Engineering, Montreal.

Whitney, D. E., 1972, "The mathematics of coordinated control of prosthetic arms and manipulators", *Journal of Dynamic Systems, Measurement, and Control*, Vol. 94, No. 14, pp. 303-309.

Appendix A

Equivalent HD Representations

Once the origin of the frames (O_i) are determined for each link, one has the freedom of choosing the direction of the axis of rotation Z_i, as well as of choosing the direction of the axis X_i. Among four HD parameters, only a_i will remain unchanged if different directions are chosen for the above axes. The effect of choosing an opposite direction of the said axes on the remaining three HD parameters is analyzed below. For a six-axis manipulator, assuming the directions of X_1 and X_7 are fixed,

i) If X_i, for $i = 2, \ldots 6$, is chosen in the opposite direction, then α_{i-1}, θ_{i-1}, and θ_i are affected as follows,

$$\begin{aligned} \alpha_{i-1} &\rightarrow -\alpha_{i-1} \\ \theta_{i-1} &\rightarrow \theta_{i-1} + \pi \\ \theta_i &\rightarrow \theta_i + \pi \end{aligned}$$

i) If Z_i, for $i = 2, \ldots 6$, is chosen in the opposite direction, then $\alpha_{i-1}, \alpha_i, \theta_i$, and b_i are affected as follows,

$$\begin{aligned} \alpha_{i-1} &\rightarrow \alpha_{i-1} + \pi \\ \alpha_i &\rightarrow \alpha_i + \pi \\ \theta_i &\rightarrow -\theta_i \\ b_i &\rightarrow -b_i \end{aligned}$$

Appendix B

Computational Analysis of Matrix Products

B.1 Multiplication of Two Rotation Matrices

Given the rotation matrices $\mathbf{Q}_1, \mathbf{Q}_2, \ldots, \mathbf{Q}_6$ for a six-axis manipulator, the computational cost of the first product of those matrices is calculated below, an asterix indicating a non-zero entry, while M and A denote multiplications and additions, respectively,

$$\begin{bmatrix} * & * & * \\ * & * & * \\ 0 & * & * \end{bmatrix} \begin{bmatrix} * & * & * \\ * & * & * \\ 0 & * & * \end{bmatrix} \Rightarrow \begin{bmatrix} 2M+1A & 3M+2A & 3M+2A \\ 2M+1A & 3M+2A & 3M+2A \\ 1M & 2M+1A & 2M+1A \end{bmatrix} \Rightarrow 21M + 12A$$

B.2 Derivation of the Remaining Four Products

Once the first product is derived, the remaining ones can be derived by making use of the properties of the rotation matrices, as outlined in (Angeles, 1989),

$\mathbf{P}_1 \leftarrow \mathbf{Q}_1\mathbf{Q}_2$

For i=3,n do

$\mathbf{P}_{i-1} \leftarrow \mathbf{P}_{i-2}\mathbf{Q}_i$

enddo

$\mathbf{Q} \leftarrow \mathbf{P}_{n-1}$

The product $\mathbf{P}_{i-2}\mathbf{Q}_i$ is computed as follows: Let $\mathbf{P}_{i-2}$ and $\mathbf{P}_{i-1}$ be denoted as,

$$\mathbf{P}_{i-2} \equiv \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad \mathbf{P}_{i-1} \equiv \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix}$$

We first compute u, v, and w as

$$\begin{aligned} u &= a_{11} \sin\theta_i - a_{12} \cos\theta_i \\ v &= a_{21} \sin\theta_i - a_{22} \cos\theta_i \\ w &= a_{31} \sin\theta_i - a_{32} \cos\theta_i \end{aligned}$$

The product $\mathbf{P}_i$ can now be computed as,

$$\begin{aligned} b_{11} &= a_{11} \cos\theta_i + a_{12} \sin\theta_i \\ b_{12} &= -u \cos\alpha_i + a_{13} \sin\alpha_i \\ b_{13} &= u \sin\alpha_i + a_{13} \cos\alpha_i \\ b_{21} &= a_{21} \cos\theta_i + a_{22} \sin\theta_i \\ b_{22} &= -v \cos\alpha_i + a_{23} \sin\alpha_i \\ b_{23} &= v \sin\alpha_i + a_{23} \cos\alpha_i \\ b_{31} &= a_{31} \cos\theta_i + a_{32} \sin\theta_i \\ b_{32} &= -w \cos\alpha_i + a_{33} \sin\alpha_i \\ b_{33} &= w \sin\alpha_i + a_{33} \cos\alpha_i \end{aligned}$$

The derivation of u, v and w requires $6M$ and $3A$, whereas each b_{ij} requires $2M$ and $1A$. Therefore, the above product requires $24M$ and $12A$. Thus, the remaining four products require $96M$ and $48A$. Hence, the computation of $\mathbf{Q}$ requires

$$117M \text{ and } 60A$$

Appendix C

Derivation of Quadratic Invariants from the Rotation Matrix

First, the quadratic invariants of a given rotation matrix **Q** can be computed from its linear invariants, namely,

$$\hat{q}_0 = \sqrt{\frac{1+q_0}{2}} \qquad \hat{\mathbf{q}} = \frac{\hat{q}_0}{(1+q_0)}\mathbf{q}$$

where $\hat{q}_0$ requires $1S + 1M + 1A$, with S denoting square-root operation. Furthermore, $\hat{\mathbf{q}}$ requires $1D$ and $3M$, with D denoting divisions. Thus, the cost of deriving the quadratic invariants from linear invariants is $1S + 1D + 4M + 1A$. Moreover, the cost of computing the linear invariants is $4M$ and $6A$. Thus, the total cost of deriving the quadratic invariants is

$$1S + 1D + 8M + 7A$$

Secondly, the quadratic invariants of a given rotation matrix **Q** are computed directly from the entries of the matrix using the following algorithm:

$\hat{q}_0 \leftarrow +\frac{1}{2}\sqrt{1 + q_{11} + q_{22} + q_{33}}$ $1S + 1M + 3A$

if $|\hat{q}_0| > \epsilon$

$$\begin{bmatrix} \hat{q}_1 \\ \hat{q}_2 \\ \hat{q}_3 \end{bmatrix} \leftarrow \frac{1}{4\hat{q}_0} \begin{bmatrix} q_{32} - q_{23} \\ q_{13} - q_{31} \\ q_{21} - q_{12} \end{bmatrix} \qquad 1D + 4M + 3A$$

else

find maximum $|q_{ii}|$

$\hat{q}_i \leftarrow \sqrt{\frac{q_{ii}+1}{2} - q_0^2}$ $1S + 2M + 1A$

$\hat{q}_j \leftarrow \frac{1}{4\hat{q}_i}(q_{ji} + q_{ij})$ $1D + 2M + 1A$

$\hat{q}_k \leftarrow \frac{1}{4\hat{q}_i}(q_{ki} + q_{ik})$ $1M + 1A$

The above algorithm requires $1S + 1D + 5M + 6A$ when $\hat{q}_0 \neq 0$, otherwise it requires $2S + 1D + 6M + 6A$.

Appendix D

Linear Invariants of Matrix Products

D.1 Method 1: Matrix Multiplications

A single product of two rotation matrices requires $21M$ and 12A. The derivation of the linear invariants from this product requires $4M$ and 6A, hence the total number of operations is

$$25M \text{ and } 18\text{A}$$

Moreover, as shown above the product of 6 rotation matrices is computed in $117M$ and 60A. The derivation of the linear invariants from this matrix requires $4M$ and 6A, hence the total number of operations is

$$121M \text{ and } 66\text{A}$$

D.2 Method 2: Vector Compositions

Using vector calculations, the computational cost in the derivation of the linear invariants of the first product $\mathbf{Q}_1\mathbf{Q}_2$ is determined below in terms of the number of divisions, multiplications and additions required.

$$q_0^{(1)} = \frac{\mathrm{tr}(\mathbf{Q}_1) - 1}{2} \qquad 1M + 3A$$

$$q_0^{(2)} = \frac{\mathrm{tr}(\mathbf{Q}_2) - 1}{2} \qquad 1M + 3A$$

$$\mathbf{q}^{(1)} = \mathrm{vect}(\mathbf{Q}_1) \qquad 3M + 2A$$

$$
\begin{array}{rl}
\mathbf{q}^{(2)} = \mathrm{vect}(\mathbf{Q}_2) & 3M + 2A \\
(1 + q_0^{(2)})\mathbf{q}^{(1)} & 3M + 1A \\
(1 + q_0^{(1)})\mathbf{q}^{(2)} & 3M + 1A \\
\mathbf{q}^{(1)} \times \mathbf{q}^{(2)} & 6M + 3A \\
D \equiv (1 + q_0^{(1)})(1 + q_0^{(2)}) & 1M \\
D - \mathbf{q}^{(1)} \cdot \mathbf{q}^{(2)} & 3M + 3A \\
\dfrac{1}{2D} & 1D + 1M \\
\mathbf{n} \equiv (D - \mathbf{q}^{(1)} \cdot \mathbf{q}^{(2)})[(1 + q_0^{(2)})\mathbf{q}^{(1)} + (1 + q_0^{(1)})\mathbf{q}^{(2)} + \mathbf{q}^{(1)} \times \mathbf{q}^{(2)}] & 3M + 6A
\end{array}
$$

$$
\mathbf{q} \equiv \frac{\mathbf{n}}{2D}
$$

Thus, calculating $\mathbf{q}$ takes 1 more multiplication for a total of 1 division, 29 multiplications and 24 additions. To find the trace, one further needs to compute N

$$
\begin{array}{rl}
(q_0^{(1)} + q_0^{(2)} + q_0^{(1)} q_0^{(2)}) & 1M + 2A \\
(\mathbf{q}^{(1)} \cdot \mathbf{q}^{(2)})(\mathbf{q}^{(1)} \cdot \mathbf{q}^{(2)} - 2D) & 1M + 1A \\
D(q_0^{(1)} + q_0^{(2)} + q_0^{(1)} q_0^{(2)}) & 1M
\end{array}
$$

$$
N = D(q_0^{(1)} + q_0^{(2)} + q_0^{(1)} q_0^{(2)}) + (\mathbf{q}^{(1)} \cdot \mathbf{q}^{(2)})(\mathbf{q}^{(1)} \cdot \mathbf{q}^{(2)} - 2D)
$$

Thus, N takes 3 more multiplications and 4 more additions after the derivation of $\mathbf{q}$. Finally,

$$
q_0 = \frac{N - D}{2D}
$$

Thus, q_0 will take 1 more multiplication and 1 more addition for a total of $1D + 33M + 29A$ for the computation of the linear invariants of the first product with the proposed method.

To compute the linear invariants of the EE orientation matrix, four more products are required. For subsequent products $\mathbf{P}_i\mathbf{Q}_{i+2}, i = 1,..,4$, the linear invariants of the first rotation matrix in the product is known from the previous step, thus saving $4M + 5A$. Therefore the total cost with the proposed algorithm will be:

$$
(1D{+}33M{+}29A) + 4[(1D{+}33M{+}29A){-}(4M{+}5A)] = 5D{+}149M{+}125A
$$

Appendix E

Quadratic Invariants of Matrix Products

E.1 Method 1: Matrix Multiplications

For the first matrix product, $21M$ and $12A$ are needed. Furthermore, the quadratic invariants are extracted from the derived matrix in $1S + 1D + 5M + 6A$, thus total cost for the first product is

$$1S + 1D + 26M + 18A$$

If the quadratic invariants of the EE is required, we have $117M$ and $60A$ from the matrix multiplications, hence a total of

$$1S + 1D + 122M + 66A$$

are needed.

E.2 Method 2: Quadratic Invariants from the Composition of Linear Invariants

The proposed method of vector calculations is extended to derive the quadratic invariants, namely,

$$\hat{q}_0 = \frac{1}{2}\sqrt{\frac{D+N}{D}} \quad \hat{\mathbf{q}} = \frac{\hat{q}_0}{N+D}\mathbf{n}$$

The computation of $\hat{q}_0$ requires $1S + 1D + 1M + 1A$, and that of $\hat{\mathbf{q}}$ requires an additional $1D + 3M$. It is also recalled that the derivation of $\mathbf{n}, N$ and D requires $1D + 31M + 28A$, which is $2M$ and $1A$ less than the derivation of $\mathbf{q}$ and q_0. Thus, the total cost for deriving the quadratic invariants for the first product is

$$1S + 3D + 35M + 29A$$

Furthermore, the derivation of $\mathbf{n}$, N and D for the EE orientation matrix requires $5D + 147M + 124A$, which is again $2M$ and $1A$ less than the number of operations required by the linear invariants of the final product. Thus, the total cost for deriving the quadratic invariants of the final product is

$$1S + 7D + 151M + 125A$$

E.3 Method 3: Vector Compositions of Quadratic Invariants

The quadratic invariants of the end product of two rotation matrices is derived from the quadratic invariants of the individual matrices, namely

$$\begin{aligned} \hat{q}_0 &= \hat{q}_0^{(1)}\hat{q}_0^{(2)} - \hat{\mathbf{q}}^{(1)} \cdot \hat{\mathbf{q}}^{(2)} \\ \hat{\mathbf{q}} &= \hat{q}_0(\hat{q}_0^{(2)}\hat{\mathbf{q}}^{(1)} + \hat{q}_0^{(1)}\hat{\mathbf{q}}^{(2)} + \hat{\mathbf{q}}^{(1)} \times \hat{\mathbf{q}}^{(2)}) \end{aligned}$$

The derivation of $\hat{q}_0$ requires $4M$ and $3A$, while $\hat{\mathbf{q}}$ requires $15M$ and $9A$. Furthermore, the derivation of each set of quadratic invariants requires $1S + 1D + 5M + 6A$. Thus, the total cost for the first product is

$$2S + 2D + 29M + 24A$$

For five products we have

$$(2S + 2D + 29M + 24A) + 4[(1S + 1D + 5M + 6A) + (19M + 12A)]$$

which gives a total of $6S + 6D + 125M + 96A$.

Appendix F

Computational Cost of the Velocity Jacobian

Assuming that the product matrices expressing the orientation of the EE in each frame are known, we proceed to calculate the velocity Jacobian defined in eq.(2.10).

Since the product of a rotation matrix with the vector $[0,0,1]^T$ amounts to the third column of that matrix, this product does not require any operation, and the unit vectors $\mathbf{e}_i$, for $i = 1,\dots,6$, are calculated at no cost. The computation of $\mathbf{r}_i$, for $i = 1,\dots,6$, is discussed next. Since we have

$$\mathbf{r}_6 \leftarrow \mathbf{a}_6$$

```
For   i=5 to 1   do
```

$$\mathbf{r}_i \leftarrow \mathbf{a}_i + \mathbf{Q}_i\mathbf{r}_{i+1}$$

```
enddo
```

we first need to calculate $\mathbf{a}_i$ for $i = 1,\dots,6$, defined as,

$$\mathbf{a}_i \equiv \begin{bmatrix} \cos\theta_i a_i \\ \sin\theta_i a_i \\ b_i \end{bmatrix}$$

each of which requires $(2T{+}2M)$, for a total of $12T$ and $12M$. Next, $\mathbf{r}_6$ does not require any operation but $\mathbf{r}_i \leftarrow \mathbf{a}_i + \mathbf{Q}_i\mathbf{r}_{i+1}$ requires $(8M{+}7A)$ each, for a total of $40M$ and $35A$ for the remaining five vectors. Thus, the calculation of $\mathbf{r}_i$, for $i = 1,\dots,6$, requires $12T$, $52M$ and $35A$.

To calculate $[\mathbf{e}_i \times \mathbf{r}_i]_1$, for $i = 1, \ldots, 6$, we proceed as follows. Since we have

$$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} = \begin{bmatrix} -r_y \\ r_x \\ 0 \end{bmatrix}$$

the cross products do not require any operation. However, to compute $[\mathbf{e}_2 \times \mathbf{r}_2]_1$ we have,

$$[\mathbf{e}_2 \times \mathbf{r}_2]_1 = \mathbf{Q}_1[\mathbf{e}_2 \times \mathbf{r}_2]_2 \rightarrow \begin{bmatrix} * & * & * \\ * & * & * \\ 0 & * & * \end{bmatrix} \begin{bmatrix} -r_y \\ r_x \\ 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 2M + 1A \\ 2M + 1A \\ 1M \end{bmatrix}$$

and for $[\mathbf{e}_i \times \mathbf{r}_i]_1$, for $i = 3, \ldots, 6$, we have,

$$\begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix} \begin{bmatrix} -r_y \\ r_x \\ 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 2M + 1A \\ 2M + 1A \\ 2M + 1A \end{bmatrix}$$

totaling $(5M+2A) + 4[(6M+3A)] = 29M+14A$. Therefore, for the computation of the velocity Jacobian we add the cost of $\mathbf{r}_i$, for $i = 1, \ldots, 6$, and that of $[\mathbf{e}_i \times \mathbf{r}_i]_1$, for $i = 1, \ldots, 6$, yielding a total of

$$12T + 81M + 49A$$

Appendix G

Rotation Matrix from the Quadratic Invariants

The rotation matrix $\sqrt{\mathbf{Q}}$ is computed from the quadratic invariants as

$$\sqrt{\mathbf{Q}} \equiv \hat{q}_0\mathbf{1} + \frac{1}{1+\hat{q}_0}\hat{\mathbf{q}} \otimes \hat{\mathbf{q}} + \mathbf{1} \times \hat{\mathbf{q}}$$

The individual entries of the above matrix is expressed with the aid of the auxiliary variables c, u, v, w, x, y, z as

$$\begin{aligned}
c &= \frac{1}{1+\hat{q}_0} && 1D+1A \\
u &= c\hat{q}_1\hat{q}_2 && 2M \\
v &= c\hat{q}_1\hat{q}_3 && 2M \\
w &= c\hat{q}_2\hat{q}_3 && 2M \\
x &= c\hat{q}_1\hat{q}_1 && 2M \\
y &= c\hat{q}_2\hat{q}_2 && 2M \\
z &= c\hat{q}_3\hat{q}_3 && 2M
\end{aligned}$$

so that

$$\begin{aligned}
\hat{q}_{11} &= \hat{q}_0 + x && 1A \\
\hat{q}_{12} &= u - \hat{q}_3 && 1A
\end{aligned}$$

$$\begin{aligned}
\hat{q}_{13} &= v + \hat{q}_2 && 1A \\
\hat{q}_{21} &= u + \hat{q}_3 && 1A \\
\hat{q}_{22} &= \hat{q}_0 + y && 1A \\
\hat{q}_{23} &= w - \hat{q}_1 && 1A \\
\hat{q}_{31} &= u - \hat{q}_2 && 1A \\
\hat{q}_{32} &= w + \hat{q}_1 && 1A \\
\hat{q}_{33} &= \hat{q}_0 + z && 1A
\end{aligned}$$

The total cost for the above calculations is thus $1D + 12M + 10A$.

Appendix H

Derivation of the Isoconditioning Locus

H.1 Representation of a Spherical Wrist

Following the HD notation, a spherical wrist is analyzed using three pairs of HD parameters, namely (α_i, θ_i), for $i = 1, 2, 3$, where

α_i : angle from $\mathbf{e}_i$ to $\mathbf{e}_{i+1}$ measured about $\mathbf{x}_{i+1}$

θ_i : angle from $\mathbf{x}_i$ to $\mathbf{x}_{i+1}$ measured about $\mathbf{e}_i$

and,

$\mathbf{e}_i$ is the unit vector parallel to the axis of the ith joint

$\mathbf{x}_i$ is the axis perpendicular to both $\mathbf{e}_{i-1}$ and $\mathbf{e}_i$

The three HD frames with axes $(\mathbf{x}_i, \mathbf{y}_i, \mathbf{e}_i)$, for $i = 1, 2, 3$, give rise to the following relations (Fig. 3.3):

$$\cos\alpha_1 = \mathbf{e}_1 \cdot \mathbf{e}_2 \tag{H.1}$$

$$\cos\alpha_2 = \mathbf{e}_2 \cdot \mathbf{e}_3 \tag{H.2}$$

$$\cos\alpha_3 = \mathbf{e}_3 \cdot \mathbf{e}_4 \tag{H.3}$$

$$\cos\theta_1 = \mathbf{x}_1 \cdot \mathbf{x}_2 \tag{H.4}$$

$$\cos\theta_2 = \mathbf{x}_2 \cdot \mathbf{x}_3 \tag{H.5}$$

$$
\begin{aligned}
\cos\theta_3 &= \mathbf{x}_3 \cdot \mathbf{x}_4 && \text{(H.6)}\\
\mathbf{x}_2 &= \frac{\mathbf{e}_1 \times \mathbf{e}_2}{\sin\alpha_1} && \text{(H.7)}\\
\mathbf{x}_3 &= \frac{\mathbf{e}_2 \times \mathbf{e}_3}{\sin\alpha_2} && \text{(H.8)}\\
\mathbf{x}_4 &= \frac{\mathbf{e}_3 \times \mathbf{e}_4}{\sin\alpha_3} && \text{(H.9)}\\
\mathbf{e}_1 &= \frac{\mathbf{x}_1 \times \mathbf{x}_2}{\sin\theta_1} && \text{(H.10)}\\
\mathbf{e}_2 &= \frac{\mathbf{x}_2 \times \mathbf{x}_3}{\sin\theta_2} && \text{(H.11)}\\
\mathbf{e}_3 &= \frac{\mathbf{x}_3 \times \mathbf{x}_4}{\sin\theta_3} && \text{(H.12)}
\end{aligned}
$$

H.2 The Computation of Two Rotations

When the second revolute of a 3-axis spherical wrist is locked, the wrist is considered as a two-revolute manipulator (Fig. 3.4). Here, the first and last frames, namely $(\mathbf{x}_1, \mathbf{y}_1, \mathbf{e}_1)$ and $(\mathbf{x}_4, \mathbf{y}_4, \mathbf{e}_4)$ are unchanged, the second frame $(\mathbf{x}_2, \mathbf{y}_2, \mathbf{e}_2)$ is removed, and finally, in the third frame, $\mathbf{x}_3$ and $\mathbf{y}_3$ are replaced with $\mathbf{i}$ and $\mathbf{j}$, respectively.

The two pairs of HD parameters needed now are (β, θ_1') and (α_3, θ_3'). In order to derive θ_1' and θ_3', as a function of $(\alpha_i,\ \theta_i)$, for $i = 1, 2, 3$, we consider the following relations

$$
\begin{aligned}
\cos\beta &= \mathbf{e}_1 \cdot \mathbf{e}_3 && \text{(H.13)}\\
\cos\theta_1' &= \mathbf{x}_1 \cdot \mathbf{i} && \text{(H.14)}\\
\cos\theta_3' &= \mathbf{i} \cdot \mathbf{x}_4 && \text{(H.15)}\\
\mathbf{i} &= \frac{\mathbf{e}_1 \times \mathbf{e}_3}{\sin\beta} && \text{(H.16)}\\
\mathbf{e}_1 &= \frac{\mathbf{x}_1 \times \mathbf{i}}{\sin\theta_1'} && \text{(H.17)}\\
\mathbf{e}_3 &= \frac{\mathbf{i} \times \mathbf{x}_4}{\sin\theta_3'} && \text{(H.18)}
\end{aligned}
$$

Furthermore, the following vector identities are used

$$
(\mathbf{a} \times \mathbf{b}) \cdot (\mathbf{c} \times \mathbf{d}) = (\mathbf{a} \cdot \mathbf{c})(\mathbf{b} \cdot \mathbf{d}) - (\mathbf{a} \cdot \mathbf{d})(\mathbf{b} \cdot \mathbf{c}) \qquad \text{(H.19)}
$$

$$(\mathbf{a} \times \mathbf{b}) \times (\mathbf{c} \times \mathbf{d}) = \mathbf{b}(\mathbf{a} \cdot (\mathbf{c} \times \mathbf{d})) - \mathbf{a}(\mathbf{b} \cdot (\mathbf{c} \times \mathbf{d})) \tag{H.20}$$

H.2.1 Derivation of θ_1'

Combining eqs.(H.16) and (H.17) one has:

$$\mathbf{e}_1 = \frac{\mathbf{x}_1 \times (\mathbf{e}_1 \times \mathbf{e}_3)}{\sin\theta_1' \sin\beta} = \mathbf{e}_1(\mathbf{x}_1 \cdot \mathbf{e}_3) - \mathbf{e}_3(\mathbf{x}_1 \cdot \mathbf{e}_1)$$

thus

$$\mathbf{x}_1 \cdot \mathbf{e}_3 = \sin\theta_1' \sin\beta \tag{H.21}$$

Furthermore, applying the dot product on both sides of eq.(H.7) with $\mathbf{e}_3$ gives

$$\mathbf{x}_2 \cdot \mathbf{e}_3 = \frac{(\mathbf{e}_1 \times \mathbf{e}_2) \cdot \mathbf{e}_3}{\sin\alpha_1} = \frac{(\mathbf{e}_3 \times \mathbf{e}_1) \cdot \mathbf{e}_2}{\sin\alpha_1} = \frac{(\mathbf{e}_2 \times \mathbf{e}_3) \cdot \mathbf{e}_1}{\sin\alpha_1}$$

From eqs.(H.8) and (H.16) we have

$$\mathbf{x}_2 \cdot \mathbf{e}_3 = \frac{\mathbf{x}_3 \cdot \mathbf{e}_1}{\sin\alpha_1} \sin\alpha_2 = -\frac{\mathbf{i} \cdot \mathbf{e}_2}{\sin\alpha_1} \sin\beta \tag{H.22}$$

To find an expression for $\mathbf{i} \cdot \mathbf{e}_2$, one resorts to eqs.(H.11), (H.8) and (H.7).

$$\mathbf{e}_2 = \frac{\mathbf{x}_2 \times (\mathbf{e}_2 \times \mathbf{e}_3)}{\sin\theta_2 \sin\alpha_2} = \frac{(\mathbf{e}_1 \times \mathbf{e}_2) \times (\mathbf{e}_2 \times \mathbf{e}_3)}{\sin\theta_2 \sin\alpha_2 \sin\alpha_1}$$

which can be reduced, namely,

$$\mathbf{e}_2 = \frac{\mathbf{e}_2[\mathbf{e}_1 \cdot (\mathbf{e}_2 \times \mathbf{e}_3)]}{\sin\theta_2 \sin\alpha_2 \sin\alpha_1} = \mathbf{e}_2 \frac{[(\mathbf{e}_3 \times \mathbf{e}_1) \cdot \mathbf{e}_2]}{\sin\theta_2 \sin\alpha_2 \sin\alpha_1} = \mathbf{e}_2 \frac{-\sin\beta}{\sin\theta_2 \sin\alpha_2 \sin\alpha_1} \mathbf{i} \cdot \mathbf{e}_2$$

Thus

$$\mathbf{i} \cdot \mathbf{e}_2 = -\frac{\sin\theta_2 \sin\alpha_2 \sin\alpha_1}{\sin\beta} \tag{H.23}$$

Combining eqs.(H.22) and (H.23) one has

$$\mathbf{x}_2 \cdot \mathbf{x}_3 = \sin\theta_2 \sin\alpha_2 \tag{H.24}$$

$$\mathbf{x}_3 \cdot \mathbf{e}_1 = -\sin\theta_2 \sin\alpha_1 \tag{H.25}$$

Moreover, from the definition of the frame $(\mathbf{i}, \mathbf{j}, \mathbf{k})$, one has the following relation:

$$\mathbf{e}_1 = \sin\beta \mathbf{j} + \cos\beta \mathbf{k} = \sin\beta \mathbf{e}_3 \times \mathbf{i} + \cos\beta \mathbf{e}_3$$

Taking the dot product of the foregoing expression with $\mathbf{e}_2$ on both sides gives

$$\mathbf{e}_1 \cdot \mathbf{e}_2 = \sin\beta \mathbf{e}_3 \times \mathbf{i} \cdot \mathbf{e}_2 + \cos\beta \mathbf{e}_3 \cdot \mathbf{e}_2 = \sin\beta \mathbf{e}_2 \times \mathbf{e}_3 \cdot \mathbf{i} + \cos\beta\cos\alpha_2$$

thus

$$\mathbf{x}_3 \cdot \mathbf{i} = \frac{\cos\alpha_1 - \cos\beta\cos\alpha_2}{\sin\beta\sin\alpha_2} \tag{H.26}$$

On the other hand, the cross product of both sides of eq.(H.10) with $\mathbf{x}_1$ gives

$$\mathbf{e}_1 \times \mathbf{x}_1 = \frac{(\mathbf{x}_1 \times \mathbf{x}_2) \times \mathbf{x}_1}{\sin\theta_1} = \frac{\mathbf{x}_2(\mathbf{x}_1 \cdot \mathbf{x}_1) - \mathbf{x}_1(\mathbf{x}_2 \cdot \mathbf{x}_1)}{\sin\theta_1} = \frac{\mathbf{x}_2 - \mathbf{x}_1\cos\theta_1}{\sin\theta_1}$$

which gives the relation

$$\mathbf{x}_2 = \frac{1}{\cos\theta_1}(\mathbf{x}_2 - \sin\theta_1 \mathbf{e}_1 \times \mathbf{x}_1) \tag{H.27}$$

Similarly, taking the cross product of both sides of (H.10) with $\mathbf{x}_2$ gives

$$\mathbf{e}_1 \times \mathbf{x}_2 = \frac{(\mathbf{x}_1 \times \mathbf{x}_2) \times \mathbf{x}_2}{\sin\theta_1} = \frac{\mathbf{x}_2(\mathbf{x}_1 \cdot \mathbf{x}_2) - \mathbf{x}_1(\mathbf{x}_2 \cdot \mathbf{x}_2)}{\sin\theta_1} = \frac{\mathbf{x}_2\cos\theta_1 - \mathbf{x}_1}{\sin\theta_1}$$

thus

$$\mathbf{x}_1 = (\mathbf{x}_2\cos\theta_1 - \sin\theta_1 \mathbf{e}_1 \times \mathbf{x}_2) \tag{H.28}$$

One can compute $\cos\theta_1'$ by taking the dot product of both sides of eq. (H.28) with $\mathbf{i}$

$$\mathbf{x}_1 \cdot \mathbf{i} = (\mathbf{x}_2\cos\theta_1 - \sin\theta_1 \mathbf{e}_1 \times \mathbf{x}_2) \cdot \frac{\mathbf{e}_1 \times \mathbf{e}_3}{\sin\beta}$$

which further reduces to

$$\mathbf{x}_1 \cdot \mathbf{i} = \frac{1}{\sin\beta}(\cos\theta_1 \frac{\mathbf{e}_1 \times \mathbf{e}_2}{\sin\alpha_1} \cdot \mathbf{e}_1 \times \mathbf{e}_3 - \sin\theta_1(\mathbf{e}_1 \times \mathbf{x}_2) \cdot (\mathbf{e}_1 \times \mathbf{e}_3)) =$$

$$\frac{1}{\sin\beta}\{\frac{\cos\theta_1}{\sin\alpha_1}(\cos\alpha_2 - \cos\alpha_1\cos\beta) - \sin\theta_1[(\mathbf{e}_1 \cdot \mathbf{e}_1)(\mathbf{x}_2 \cdot \mathbf{e}_3) - (\mathbf{e}_1 \cdot \mathbf{e}_3)(\mathbf{e}_1 \cdot \mathbf{x}_2)]\}$$

Therefore ,

$$\cos\theta_1' = \frac{1}{\sin\beta}[\frac{\cos\theta_1}{\sin\alpha_1}(\cos\alpha_2 - \cos\alpha_1\cos\beta) - \sin\theta_1\sin\theta_2\sin\alpha_2] \tag{H.29}$$

An expression for $\cos\theta_1'$ can also be derived by taking the dot product of both sides of eq.(H.27) with $\mathbf{i}$

$$\begin{aligned}\cos\theta_1' &= \mathbf{x}_1 \cdot \mathbf{i} = \frac{1}{\cos\theta_1}(\mathbf{x}_2 - \sin\theta_1 \mathbf{e}_1 \times \mathbf{x}_1) \cdot \frac{\mathbf{e}_1 \times \mathbf{e}_3}{\sin\beta} \\ &= \frac{1}{\cos\theta_1\sin\beta}[\mathbf{x}_2 \cdot \mathbf{e}_1 \times \mathbf{e}_3 - \sin\theta_1(\mathbf{e}_1 \times \mathbf{x}_1) \cdot (\mathbf{e}_1 \times \mathbf{e}_3)]\end{aligned}$$

which reduces to

$$\frac{1}{\cos\theta_1 \sin\beta}\left(\frac{\cos\alpha_2 - \cos\alpha_1\cos\beta}{\sin\alpha_1} - \sin\theta_1\sin\theta_1'\sin\beta\right)$$

We now solve for $\sin\theta_1'$, namely,

$$\sin\theta_1' = \frac{-\cos\theta_1'\cos\theta_1}{\sin\theta_1} + \frac{(\cos\alpha_2 - \cos\alpha_1\cos\beta)}{\sin\theta_1\sin\alpha_1\sin\beta} \tag{H.30}$$

Thus, θ_1' is uniquely determined with the expressions (H.29) and (H.30).

H.2.2 Derivation of θ_3'

To begin with, we equate eqs.(H.12) and (H.18), and take the cross product with $\mathbf{x}_4$ on each side of the equation.

$$\mathbf{e}_3 = \frac{\mathbf{i}\times\mathbf{x}_4}{\sin\theta_3'} = \frac{\mathbf{x}_3\times\mathbf{x}_4}{\sin\theta_3}$$

Hence,

$$\frac{(\mathbf{i}\times\mathbf{x}_4)\times\mathbf{x}_4}{\sin\theta_3'} = \frac{(\mathbf{x}_3\times\mathbf{x}_4)\times\mathbf{x}_4}{\sin\theta_3}$$

which reduces to

$$\frac{\mathbf{x}_4\cos\theta_3' - \mathbf{i}}{\sin\theta_3'} = \frac{\mathbf{x}_4\cos\theta_3 - \mathbf{x}_3}{\sin\theta_3} \tag{H.31}$$

Taking the dot product of both sides of eq.(H.31) with $\mathbf{e}_1$ yields

$$\frac{\mathbf{x}_4\cdot\mathbf{e}_1\cos\theta_3' - \mathbf{i}\cdot\mathbf{e}_1}{\sin\theta_3'} = \frac{\mathbf{x}_4\cdot\mathbf{e}_1\cos\theta_3 - \mathbf{x}_3\cdot\mathbf{e}_1}{\sin\theta_3}$$

where $\mathbf{x}_3\cdot\mathbf{e}_1$ is known from eq.(H.25). Now, solving for $\mathbf{x}_4\cdot\mathbf{e}_1$ yields

$$\mathbf{x}_4\cdot\mathbf{e}_1 = \frac{\sin\alpha_1\frac{\sin\theta_2}{\sin\theta_3}}{\frac{\cos\theta_3'}{\sin\theta_3'} - \frac{\cos\theta_3}{\sin\theta_3}} \tag{H.32}$$

An expression for $\mathbf{x}_4\cdot\mathbf{e}_1$ is also derived when both sides of eq.(H.18) are crossed with $\mathbf{e}_1$, namely,

$$\mathbf{e}_3\times\mathbf{e}_1 = \frac{(\mathbf{i}\times\mathbf{x}_4)\times\mathbf{e}_1}{\sin\theta_3')}$$

$$-\mathbf{i}\sin\beta\sin\theta_3' = \mathbf{x}_4(\mathbf{i}\cdot\mathbf{e}_1) - \mathbf{i}\mathbf{x}_4\cdot\mathbf{e}_1$$

Thus,

$$\mathbf{x}_4\cdot\mathbf{e}_1 = \sin\beta\sin\theta_3' \tag{H.33}$$

We now solve for $\sin\theta_3'$ from eqs.(H.32) and (H.33), namely,

$$\sin\theta_3' = \frac{\sin\theta_3}{\cos\theta_3}\cos\theta_3' + \frac{\sin\alpha_1\sin\theta_2}{\cos\theta_3\sin\beta} \tag{H.34}$$

Taking the dot product of both sides of eq.(H.31) with the right-hand side of eq.(H.31) gives the following expression:

$$\frac{(\mathbf{x}_4\cos\theta_3' - \mathbf{i})\cdot(\mathbf{x}_4\cos\theta_3 - \mathbf{x}_3)}{\sin\theta_3'\sin\theta_3} = 1$$

or

$$\cos\theta_3\cos\theta_3' - \cos\theta_3'\mathbf{x}_4\cdot\mathbf{x}_3 - \mathbf{i}\cdot\mathbf{x}_4\cos\theta_3 + \mathbf{i}\cdot\mathbf{x}_3 = \sin\theta_3'\sin\theta_3$$

An expression for $\mathbf{i}\cdot\mathbf{x}_3$ is known from eq.(H.26); solving for $\sin\theta_3'$ from the above equation yields

$$\sin\theta_3' = \frac{1}{\sin\theta_3}\left(-\cos\theta_3\cos\theta_3' + \frac{\cos\alpha_1 - \cos\beta\cos\alpha_2}{\sin\beta\sin\alpha_2}\right) \tag{H.35}$$

Finally, from eqs.(H.33) and (H.34),

$$\cos\theta_3' = \frac{1}{\sin\beta}\left[\frac{\cos\theta_3(\cos\alpha_1 - \cos\beta\cos\alpha_2)}{\sin\alpha_2} - \sin\theta_2\sin\theta_3\sin\alpha_1\right] \tag{H.36}$$

Thus, θ_3' is uniquely determined from the expressions (H.35) and (H.36), thereby completing all the calculations necessary for the computation of two rotations.

Appendix I

Parametrizing the Intersection of Two cylinders

Let us choose two coordinate frames (X, Y, Z) and (X', Y', Z') such that X and X' are coincident, and that Z and Z' intersect at an angle γ, measured about the X axis. Furthermore, let us attach a cylinder to each frame such that the axis of the cylinder is coincident with the Z-axis of the associated frame. The equations of the two cylinders can be expressed as,

$$x^2 + y^2 = R_1^2 \qquad -h < z < h \tag{I.1}$$

$$(x')^2 + (y')^2 = R_2^2 \qquad -h < z' < h \tag{I.2}$$

Moreover, (X', Y', Z') is expressed in the (X, Y, Z) frame as follows,

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \mathbf{Q}_x \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x \\ \cos\gamma y - \sin\gamma z \\ \sin\gamma y + \cos\gamma z \end{bmatrix}$$

Substituting the expressions for x' and y' into eq.(I.2) yields an expression for the intersection of the two cylinders, from which an expression for z^2 is readily derived as

$$z^2 = \frac{\cos\gamma y \pm \sqrt{R_2^2 - x^2}}{\sin\gamma}$$

which allows two solutions, thus leading to two trajectories. Any one of the trajectories can be chosen. We choose the positive square roots. Now, x and y are parametrized with the usual trigonometric expressions,

$$\begin{aligned} x &= R_1 \cos\beta \\ y &= R_1 \sin\beta \end{aligned}$$

thus allowing an expression of a point on the intersection curve parametrically as,

$$\mathbf{p}_g = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} R_1 \cos\beta \\ R_1 \sin\beta \\ \frac{R_1 \cos\gamma \sin\beta + \sqrt{R_2^2 - (R_1 \cos\beta)^2}}{\sin\gamma} \end{bmatrix}$$

Appendix J

HD Parameters of Example Robots

Below, the H-D parameters of *Yaskawa Aid 810, Puma 560, Fanuc Arc Mate* and Asea IRB 6/2 are included. Since all the above robots have only revolute joints, the joint variables $\{\theta_k\}_1^6$ can attain arbitrary values and are not shown below. Similarly, the values for α_6 and a_6 depend on the definition of the frame attached to the EE, and are neither shown in the tables below.

Table J.1: Hartenberg Denavit Parameters of Yaskawa Aid 810

index	α (rad)	a (mm)	b (mm)
1	$\frac{\pi}{2}$	0.	785.
2	0.	670.	0.
3	$\frac{\pi}{2}$	0.	0.
4	$\frac{\pi}{2}$	0.	950.
5	$\frac{\pi}{2}$	0.	90.
6	*	*	128.

Table J.2: Hartenberg Denavit Parameters of Puma 560

index	α (rad)	a (mm)	b (mm)
1	$\frac{-\pi}{2}$	0.	400.
2	0.	432.	149.
3	$\frac{-\pi}{2}$	20.	0.
4	$\frac{-\pi}{2}$	0.	432.
5	$\frac{\pi}{2}$	0.	0.
6	*	*	56.

Table J.3: Hartenberg Denavit Parameters of Fanuc Arc Mate

index	α (rad)	a (mm)	b (mm)
1	$\frac{\pi}{2}$	200.	810.
2	0.	600.	0.
3	$\frac{\pi}{2}$	130.	30.
4	$\frac{\pi}{2}$	0.	550.
5	$\frac{\pi}{2}$	0.	100.
6	*	*	100.

Table J.4: Hartenberg Denavit Parameters of Asea IRB 6/2

index	α (rad)	a (mm)	b (mm)
1	$\frac{\pi}{2}$	0.	700.
2	0.	450.	0.
3	$\frac{\pi}{2}$	0.	0.
4	$\frac{\pi}{2}$	0.	670.
5	$\frac{\pi}{2}$	0.	100.
6	*	*	95.