On the Cartesian Control of Orientation and Force for Robotic Manipulators

Murphy, Steve

1988

Document Version:
Publisher's PDF, also known as Version of record

Link to publication

Citation for published version (APA):
Murphy, S. (1988). *On the Cartesian Control of Orientation and Force for Robotic Manipulators*. (Technical Reports TFRT-7396). Department of Automatic Control, Lund Institute of Technology (LTH).

Total number of authors:
1

CODEN: LUTFD2/(TFRT-7396)/1-019/(1988)

On the Cartesian Control of Orientation and Force for Robotic Manipulators

Steve Murphy

Department of Automatic Control
Lund Institute of Technology
August 1988

Department of Automatic Control
Lund Institute of Technology
P.O. Box 118
S-221 00 Lund Sweden

Document name
Report

Date of issue
August 1988

Document Number
CODEN:LUTFD2/(TFRT-7396)/1-019/(1988)

Supervisor

Sponsoring organisation
The American-Scandinavian Foundation, New York

Author(s)
Steve Murphy

Title and subtitle
On the Cartesian Control of Orientation and Force for Robotic Manipulators

Abstract

The problem of controlling the orientation of a manipulator end-effector when the dynamic equations of the manipulator are expressed directly in Cartesian space is examined. The model of manipulator orientation is developed through a common Cartesian control scheme and the basic nonlinearities of Cartesian orientation are shown. Three methods of regulating orientation are investigated, and the results show the conflict between performance and calculation complexity. None of the rotational regulators performs adequately over the entire space of orientations. The control of orientation provides information into the development and interpretation of the manipulator Jacobian and the impact on force control in Cartesian space. The work shows the need for an orientation regulator that does not artificially limit the range of manipulator orientations and has a reasonable calculation cost.

Key words
robotics, Cartesian control, Jacobian, compliance

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title

ISBN

Language
English

Number of pages
19

Security classification

Recipient's notes

1. Introduction

One of the goals for robotic development has been the control of the motion of a robot end-effector in common Cartesian space. One approach to meeting this goal is the control of the manipulator directly in Cartesian space rather than control of individual joints with a Cartesian to joint translation. In the direct control approach, the equations of motion that describe the robot manipulator are written in terms of the Cartesian coordinates, and a control system is designed to track a desired Cartesian motion. This approach has also been used when designing controllers to handle both forces and positions in the Cartesian space (Hayati, 1986; Khatib, 1985; Luh et. al., 1980; Raibert and Craig, 1981).

In the work by Luh, Walker, Paul (Luh et. al., 1980), and Khatib (Khatib, 1985) the equations of motion of the robot manipulator were linearized through the use of nonlinear feedback. For the control of end-effector position in Cartesian space, a simple PD controller was designed around the linearized system of equations. The same approach is also applied to the control of end-effector orientation but the problem is complicated due to the fundamental problem of representing rotation in Cartesian space. Luh, Walker, Paul, and Khatib used different methods for the representation and control of orientation. The choice of rotation representation is important in that it will have an impact on the design and performance of a Cartesian orientation regulator.

Important work towards this problem has been accomplished by Kreutz and Wen (Kreutz and Wen, 1988). Using a Lyapunov approach, they were able to place the problem in a theoretical framework and to develop a globally stable controller for the control of orientation. They also showed a stability analysis of the controller by Luh, Walker, and Paul. Some computational aspects where investigated, but did not deal directly with Jacobian calculations and force compliance.

It is the purpose of this report to examine the methods for representing orientation in Cartesian space, and investigate their effect upon the control of Cartesian orientation. Three regulators for the control of orientation are presented and simulated. The first regulator is based upon the work by Luh, Walker, and Paul (Luh et. al., 1980) and uses the complete rotation matrix to describe the orientation. The second regulator uses a linearized model of rotation based upon Z-Y-X Euler angles while the third uses a form of feedback linearization with Euler angles.

The use of the Euler angle feedback linearization leads to the redefinition of the manipulator Jacobian. This redefined Jacobian is equivalent to the Jacobian used in much of the robotics literature (Tarn, 1986b,1987; Hayati, 1986; Khatib, 1985,1986). A few of the limitations to this Jacobian are examined. Finally, it is shown how the use of different Jacobians and forms of orientation representation have an impact upon the performance of a Cartesian controller when utilizing force control.

The report is organized as follows: In Chapter 2 a direct Cartesian controller is presented and the problem of controlling the orientation is discussed. Three methods of controlling the orientation are presented and simulated. The results lead to the development of another form of the manipulator Jacobian which in turn provides an alternate description of the Cartesian controller, this is presented in Chapter 3. Included in Chapter 3 is a discussion of the limitations of this Jacobian and how it relates to compliance and force control. Conclusions are presented in Chapter 4.

2. Orientation Control

The problem of orientation control arises through the Cartesian linearization of the robot manipulator. Luh, Walker, Paul, (Luh et. al., 1980), use feedback linearization to write the equations of motion of the robot directly in Cartesian space. The result of the feedback is a system where the input signal directly affects the linear and angular accelerations. The linear motion is easily handled using a PD regulator; the object is then to design a regulator for the resulting system of angular accelerations.

The regulator design problem is complicated by the physics of rotation. In linear motion, the linear velocity may be integrated to find the position. However, in rotation, it is not meaningful to integrate the angular velocities (Goldstein 1980). The exception is when rotation occurs in a plane, whereupon the rotational angle is equivalent to the integral of the angular velocity. In general, the relationship between angular velocity and rotational position is nonlinear and depends upon the representation of rotation. Through different representations of rotational position, the model of rotation may be transformed for use in designing regulators, planning rotational motions, and complying to external forces.

The first section of this chapter presents the background on the Cartesian linearization of the equations of motion of the robotic manipulator. This introduces the rotational model and the following section describes some of the representations of orientation available in the robotics literature. The next section, 2.3, describes the details of the three rotational regulators that were investigated: how they represent rotation, and how they were simulated. In Section 2.4 the results of the simulation are presented along with a discussion of the limitations of each regulator. Section 2.5 summarizes the chapter.

2.1 Background to Cartesian Control

The Cartesian control scheme that is presented in this section is derived from the work of (Luh et. al., 1980; Khatib, 1985; Hayati, 1986) and is used for presenting the problem source. It is also used as a case example in later chapters as some of the results are applied to position and force control. A six degree of freedom manipulator is linearized in the Cartesian space using the Jacobian as defined in (Paul, 1980; Craig, 1986). The linearization gives a system in which the input signal affects the linear and angular accelerations of the manipulator end effector.

The dynamic model for a 6 degree of freedom robotic manipulator may be written as

$$\tau = D(q)\ddot{q} + fg(q,\dot{q}) \qquad (2.1).$$

Where τ is a 6×1 vector of joint torques/forces, q is a vector of joint angles/positions, $D(q)$ is the inertia matrix, and $fg(q,\dot{q})$ represents the combined terms for gravity, Coriolis, centrifugal, and friction moments/forces.

The manipulator Jacobian is defined to give the motion of the manipulator end effector in Cartesian coordinates for incremental motions of the the manipulator, or $\dot{x}_c = J(q)\dot{q}$. The Jacobian may be defined to give incremental motions in the base frame, end effector frame, or any other frame for the robot manipulator. In the following presentation, the Jacobian will be defined in the base coordinate system and will give the motion of the end effector in base coordinates. Accordingly, $\dot{x}_c = [\dot{x} \quad \dot{y} \quad \dot{z} \quad \omega_x \quad \omega_y \quad \omega_z]^T$, and $\dot{q} = [\dot{q}_1 \quad \dot{q}_2 \quad \dot{q}_3 \quad \dot{q}_4 \quad \dot{q}_5 \quad \dot{q}_6]^T$, where $\dot{x}$, $\dot{y}$, $\dot{z}$ are the components of the linear velocity vector, and ω_x, ω_y, ω_z are the components of the angular velocity vector written in the base coordinate frame, and $\dot{q}_1, \ldots \dot{q}_6$ are the joint angular/linear velocities. Note that x_c is not defined, as there is no meaning to the integral of the angular velocity components.

Differentiating the Jacobian equation gives: $\ddot{x}_c = J\ddot{q} + \dot{J}\dot{q}$. If the input joint torque/force to the robotic manipulator is calculated as:

$$\tau = D(q)J^{-1}(u - \dot{J}\dot{q}) + fg(q, \dot{q}) \tag{2.2}$$

then the feedback linearized system becomes

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_x \\ u_y \\ u_z \end{bmatrix}. \tag{2.3}$$

Assuming that an accurate model of the manipulator is available.

It is useful to split the problem into control of end-effector position and control of end-effector orientation. For the control of x, y, and z positions, the problem is relatively uncomplicated. Measurements of Cartesian position and linear velocity are available through the forward kinematics and through the Jacobian. A PD regulator is often proposed for such a double integrator (Hayati, 1986; Khatib, 1985; Luh et. al., 1980). For the control of orientation, it is first necessary to specify which representation will be used to describe the rotational position.

2.2 Representation and Measurement of Orientation

From the forward kinematics one can extract the rotation matrix

$${}^0_6R(q) = \begin{bmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{bmatrix} \tag{2.4}$$

specifying the orientation of the end-effector in base coordinates. The rotation matrix uniquely describes any orientation in Cartesian space. The last column of the matrix is a combination of the two first columns. The remaining elements of the matrix must also satisfy the three constraint equations:

$$\begin{aligned} n_x^2 + n_y^2 + n_z^2 &= 1 \\ o_x^2 + o_y^2 + o_z^2 &= 1 \\ n_x o_x + n_y o_y + n_z o_z &= 0 \end{aligned} \tag{2.5}$$

Obviously, due to the three constraint equations, three of the above elements of the rotation matrix may be eliminated leaving three independent variables.

Any such elimination would require dividing by one of the elements $n_x, \ldots o_z$. As these elements have a range from -1 to 1, there would exist some rotational position where the model would be undefined. This occurs with all of the three variable representations considered below.

As rotation in Cartesian space has only 3 degrees of freedom, and the Cartesian linearization provides only three control signals, representations using only the three independent variables of the rotation matrix are desirable. Some of the common representations are: Z-Y-X Euler angles, Z-Y-Z Euler angles, Equivalent angle-axis, and Roll-Pitch-Yaw angles (Craig, 1986; Paul 1981). Values for these representations may be extracted from elements of the rotation matrix. While there are other representations available, (Goldstein, 1980) these are some of the most commonly used in robotics, and are the only ones considered at this time.

For a given orientation, only three variables are necessary to describe that orientation. However, the set of three variables is not unique. All of the above representations will have at least two sets of variables describing the same orientation. Additionally, there exists orientations for which there are an infinite number of sets of three variables.

With the Equivalent angle-axis, and the Z-Y-Z Euler angle representations infinite sets of Euler angles for a given orientation occurs at zero rotation from the initial reference orientation. With Z-Y-X Euler angles and Roll-Pitch-Yaw angles this occurs at 180° rotation about the Y, or Pitch axis. Because it was considered to be important to regulate about zero orientation, the Z-Y-Z Euler angles and the Equivalent angle-axis representations were not appropriate for investigation into orientation control. Roll-Pitch-Yaw angles and Z-Y-X Euler angles have the same rotation matrix representation and Z-Y-X Euler angles may easily be interpreted as Roll-Pitch-Yaw angles (Craig, 1986). Thus Z-Y-X Euler angles were chosen as the three variable representation to be used in this work.

The result is two representations that were considered to be feasible for investigation into orientation control: the rotation matrix, and Z-Y-X Euler angles. In order to design regulators for the control of orientation is it necessary to develop the dynamic model of rotation using these representations.

Dynamics using the Rotation Matrix

Equation 2.3 gives the results of the Cartesian linearization of the robot equations of motion where direct control over the angular accelerations is possible. The angular acceleration vector may be integrated to find the instantaneous angular velocity. Each column of the rotation matrix represents a the coordinates of point in space rotating with angular velocity ω. The velocity of each point is the cross product of the angular velocity vector and the point's coordinate vector. Thus, the rate of change of one of the columns of the rotation matrix equals $\omega \times n$, where n is a column of the rotation matrix. Writing the cross product in matrix form the complete dynamic model of rotation becomes:

$$\begin{aligned} \dot{\omega}_x &= u_x \\ \dot{\omega}_y &= u_y \\ \dot{\omega}_z &= u_z \end{aligned} \tag{2.6}$$

$$
\begin{bmatrix} \dot{n}_x \\ \dot{n}_y \\ \dot{n}_z \end{bmatrix} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix}
$$
$$
\begin{bmatrix} \dot{o}_x \\ \dot{o}_y \\ \dot{o}_z \end{bmatrix} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \begin{bmatrix} o_x \\ o_y \\ o_z \end{bmatrix} \tag{2.7}
$$

The system has 3 control signals, u_x, u_y, u_z, and six outputs, $n_x, \ldots, o_z$ along with the 3 constraints of 2.5. A regulator built upon the above model would require the setpoint as a desired rotation matrix and require measurements of the rotation matrix and angular velocity. The elements of the rotation matrix are available from the forward kinematics of the manipulator, and the angular velocity may be calculated using the joint velocities and the manipulator Jacobian.

Dynamics using Euler Angles

From the rotation matrix, one can extract the Z-Y-X Euler angles, α, β, γ (Craig, 1986). It is important to note that when $\beta = \pm\frac{\pi}{2}$ only the sum of α and γ may be determined from the rotation matrix. The condition $\beta = \pm\frac{\pi}{2}$ occurs whenever $[\,n_x \quad n_y \quad n_z\,]^T = [\,0 \quad 0 \quad \pm 1\,]^T$.

To complete the transformation of the system to Euler angles, it is necessary to derive the Euler angle velocities from the Euler angles and angular velocities. This may be done through the derivation of the equations given in (Craig, 1986) or in a simple manner as described by (Goldstein, 1980). The results of such a transformation are as follows:

$$
\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} 0 & -s_\alpha & c_\alpha c_\beta \\ 0 & c_\alpha & s_\alpha c_\beta \\ 1 & 0 & -s_\beta \end{bmatrix} \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} \tag{2.8}
$$

where $s_\alpha = sin(\alpha)$, etc. Let E be the vector of Euler angles, $E = [\,\alpha \quad \beta \quad \gamma\,]^T$, ω the vector of angular velocities, and define the matrix as B. The velocity transformation becomes: $\omega = B\dot{E}$.

When the inverse of B exists:

$$
\begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} = \begin{bmatrix} c_\alpha s_\beta / c_\beta & s_\alpha s_\beta / c_\beta & 1 \\ -s_\alpha & c_\alpha & 0 \\ c_\alpha / c_\beta & s_\alpha / c_\beta & 0 \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad \text{or,} \quad \dot{E} = B^{-1}\omega. \tag{2.9}
$$

Differentiating, and substituting $u = \dot{\omega}$,

$$
\ddot{E} = B^{-1}u + \dot{B}^{-1}\omega \tag{2.10}
$$

and substituting for ω gives:

$$
\ddot{E} = B^{-1}u + \dot{B}^{-1}B\dot{E}. \tag{2.11}
$$

The rotational system is thus completely described in terms of Euler angles.

The matrix B does not exist whenever $\beta = \pm\frac{\pi}{2}$ and no model is available at that point. This is the point where an infinite number of Euler angle sets describe the orientation. It is an unavoidable consequence due to the transformation of the rotational system to Z-Y-X Euler angles. The same

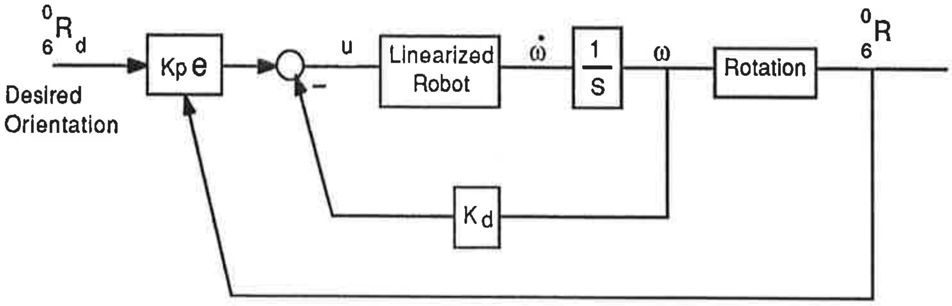

Figure 2.1 Block Diagram of Rotational Regulator by Luh, Walker, and Paul.

difficulties would occur with Z-Y-Z Euler angles, and Equivalent angle-axis representations except at zero rotation instead of at $\beta = \pm\frac{\pi}{2}$.

The model in Equation 2.11 is highly nonlinear but with three control signals u_x, u_y, u_z and three outputs α, β, γ. The measurement of the Euler angles must come through calculations based upon elements of the rotation matrix. Euler angle velocity can be calculated through derivation of the Euler angles or from Equation 2.8. The angular velocity in 2.8 is calculated using the manipulator Jacobian and joint velocities.

2.3 Control of Orientation

Rotation Matrix Regulator

Luh, Walker, and Paul, (Luh et. al., 1980) designed a tracking controller based upon the rotation matrix dynamic model of rotation, Equations 2.6–2.7. The controller uses angular velocity for derivative control and calculates a position error from the elements of the rotation matrix.

The rotational error is calculated:

$$e = \frac{1}{2}(n \times n_d + o \times o_d + a \times a_d) \tag{2.12}$$

where $n \times n_d$ is the vector cross product of the vectors n and n_d. For a regulator, the control input is calculated from:

$$u = K_p e - K_v \omega \tag{2.13}$$

and $u = [\, u_x \quad u_y \quad u_z \,]^T$, $\omega = [\, \omega_x \quad \omega_y \quad \omega_z \,]^T$. A block diagram of the system is shown in Figure 2.1. The controller was proven to be stable for sufficiently small rotational errors (Luh et. al., 1980) and further investigated by Kreutz and Wen, (Kreutz and Wen, 1988). Through simulations, the capability of this controller to regulate orientation is compared to two regulators based upon the Euler angle dynamic model.

Linearized Euler Model Regulator

The system described by Equation 2.11 is highly nonlinear. A direct approach to the design of a regulator for such a system is to design around a linearized model. Linearizing the system of 2.11 around zero gives:

$$\begin{bmatrix} \ddot{\alpha} \\ \ddot{\beta} \\ \ddot{\gamma} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} \tag{2.14}$$

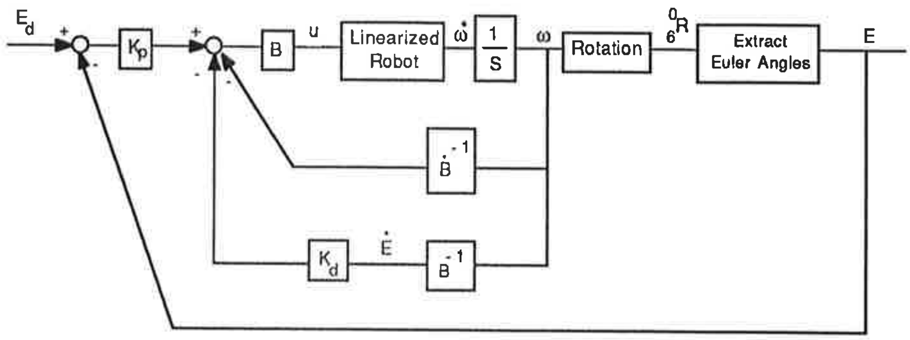

Figure 2.2 Block Diagram of Euler Angle Feedback Linearized Regulator

At this stage the design of a regulator is quite straightforward and a simple PD regulator was used. The control signal was calculated from $u = -K_pE - K_d\dot{E}$, and K_p, K_d were chosen to be 100 and 20, respectively.

Feedback Linearized Euler Regulator

The form of the nonlinear system in Equation 2.10 suggests that a feedback linearization, similar to the one used in the Cartesian linearization, might provide a reasonable solution to the regulator problem. If the input u, is defined to be

$$u = B(u_e - \dot{B}^{-1}\omega) \tag{2.15}$$

where $u_e = [\, u_\alpha \quad u_\beta \quad u_\gamma \,]^T$. The resultant system of 2.10 becomes

$$\begin{bmatrix} \ddot{\alpha} \\ \ddot{\beta} \\ \ddot{\gamma} \end{bmatrix} = \begin{bmatrix} u_\alpha \\ u_\beta \\ u_\gamma \end{bmatrix} \tag{2.16}$$

The system has now become three decoupled double integrators, and a simple regulator may be designed. A complete block diagram showing the major calculation elements used in the simulation is shown in Figure 2.2.

The regulator for the decoupled system of Equation 2.16 was chosen to be PD, similar to the previous two. The input u_e was calculated to be: $u_e = K_p(E_d - E) - K_d\dot{E}$. E_d is the desired end effector rotation specified in Euler angles.

2.4 Simulation

The rotational system and the above regulators were simulated using Simnon (Åström, 1982). Each regulator was started with some initial rotation and was expected to drive the system to zero rotation. The goal was to obtain information on the stability and performance of the regulators and their comparison.

Rotation Matrix Regulator

The results of the simulation show that this regulator was stable over the entire range of rotations but had varying settling times for different initial orientations. In the linear region of small rotations about zero, the settling times could be chosen from the feedback gains. For large rotations the settling times would increase and could become infinity at certain orientations. These

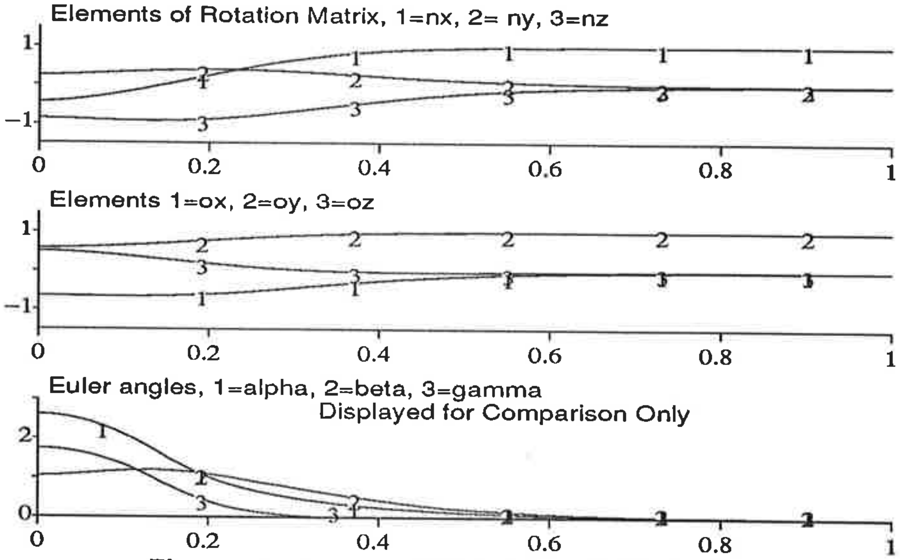

Figure 2.3 Response of the Rotation Matrix Regulator

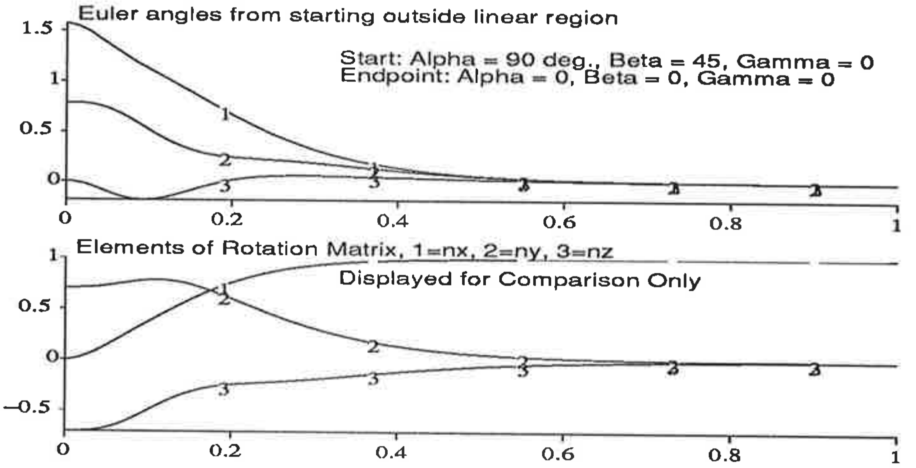

Figure 2.4 Response of PD Regulator using Euler Model Linearized about 0

orientations exist due to the use of the cross product in calculating the position error: anti-parallel vectors also have a zero cross product. An example is a 180° rotation about any axis. It was also possible for the cross product elements to cancel in a series of points about the 180° rotation. The general behavior of the regulator when not close to one of these points is shown in Figure 2.3. Overall, the stability, and simplicity of this regulator was in sharp contrast to the first Euler angle based regulator that follows.

Linearized Euler Model Regulator

The regulator based upon the linearized model produced the expected behavior for such a linearization. In small regions about the zero Euler angle point, the system was stable with the expected settling times. Outside this region, the nonlinear cross coupling between the Euler angles influenced the performance and settling times in addition to stability. However, the closed loop system was stable for much larger initial values of the Euler angles than might be expected. An approximate range of Euler angles for which the system was stable was found to be $-\frac{\pi}{2} < \alpha < \frac{\pi}{2}$, $-\frac{\pi}{2} < \beta < \frac{\pi}{2}$, $-\pi < \gamma < \pi$. Figure 2.4 shows an example of this behavior. Because of the calculation methods for E and $\dot{E}$, the regulator had extreme difficulties whenever β passed through $\pm\frac{\pi}{2}$.

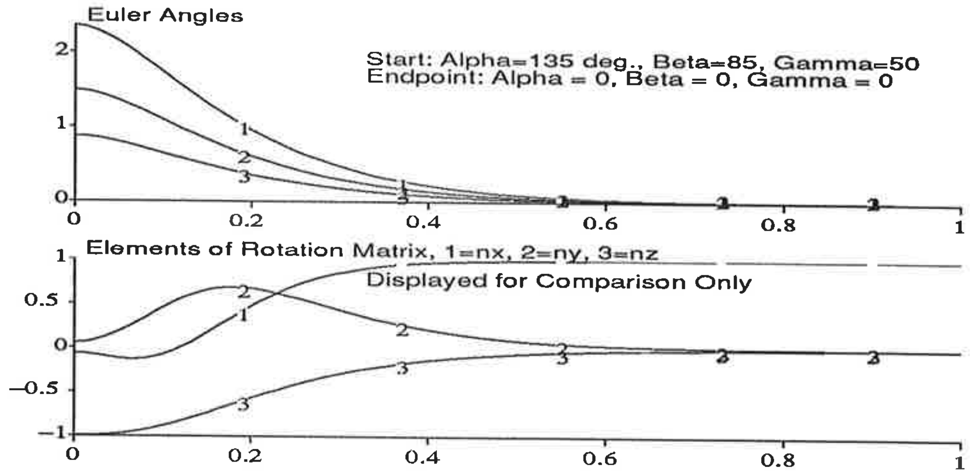

Figure 2.5 Response of the Feedback Linearized Regulator

On basis of the simulation results, this regulator was not sufficient to drive the end effector rotation to any other coordinate other than $[0 \quad 0 \quad 0]^T$. This problem may be avoided by defining the desired rotation as the zero point and calculating the Euler angles referenced to the desired point instead of in terms of the base coordinate system. This approach might be useful if the regulator was to be used as a tracking controller.

Feedback Linearized Euler Regulator

The feedback linearized regulator was simulated on the system given by equations 2.6–2.7 using the values of 100, 20 for K_p, and K_d. The results of the simulation validated the feedback linearization approach. The regulator produced a linear response for the entire range of Euler angles excluding the boundary points: $\alpha = \pm\pi$, $\beta = \pm\frac{\pi}{2}$, $\gamma = \pm\pi$. An example of the behavior of the regulator is given in Figure 2.5. The point $\beta = \pm\frac{\pi}{2}$ created numerical problems due to divide by zero in elements of the B matrix. Passing through the points $\alpha = \pm\pi$, and $\gamma = \pm\pi$ caused problems due to discontinuities in the values produced by the *atan2* function. The function *atan2* is used to calculate the Euler angles from the elements of the rotation matrix (Craig, 1986). There are no problems with model accuracy because the matrices B and $\dot{B}$ are all calculated within the regulator. In practical situations, the active range of the feedback linear regulator would exclude a small region of points around the boundary points. This would assure that the regulator could not be driven through any of the boundary points and avoid possible erratic behavior due to numerical instabilities in the calculation of the regulator.

2.5 Summary

This chapter has dealt with the problems involved with the control of rotation in Cartesian space. A straightforward Cartesian control system has been presented as a motivation for the problem and the source of the rotational system. The rotational system was shown to be nonlinear due to the representation of rotation in 3 dimensional space. Three forms of regulators for the rotational system where investigated: a regulator by Luh, Walker, and Paul, based upon elements of the rotation matrix, a regulator using Euler angles linearized about zero rotation, and a nonlinear regulator using feedback linearization of the Euler angles.

The results of the simulations may be compared with respect to two goals: performance, and ease of calculation. The rotational regulator by Luh, Walker, and Paul showed reasonable performance over the entire range of rotations but with uncertain settling times and points at which the orientation error would be erroneously zero. The regulator was remarkably simple to calculate with no numeric difficulties.

The regulator based upon the linearization of the Euler model proved to be satisfactory only in a linear region around zero rotation. While this may be acceptable for a tracking controller with small errors, the calculations involved are more involved than for the previous regulator.

Using the feedback linearized regulator dramatically improved the performance of the system over the largest range of initial orientations. However, the regulator did have difficulty on the boundary of rotation where the model of rotation failed to be valid and numeric difficulties would cause undesired behavior. This regulator had the highest calculation requirements.

Overall, no regulator performed adequately in terms of both calculation cost and performance. The difficulties to producing a reasonable regulator for orientation lie in the representation of rotation and the design of controllers for nonlinear models. It is possible that through different representations or different controllers, better results may be achieved (Kreutz and Wen, 1988). But cost of calculation and performance are not the only considerations for rotational control. The choice of representation will also have an impact on the response of a compliant control system. This topic is developed as part of the following chapter.

3. Jacobian and Force Control

The problem of controlling orientation in the previous chapter arose from the Cartesian linearization of the robot's dynamic equations. The nonlinear equations of the robot were linearized through a calculated nonlinear feedback. The same technique was applied to design the feedback linearized Euler regulator for rotation control. This chapter presents the combination of the Euler angle feedback linearization with the Cartesian linearization of the robot. The result is a different Jacobian, the Euler Jacobian, which translates incremental motions of the manipulator joints into incremental linear x, y, z motions of the end effector and incremental Euler angle motions of the end effector orientation. This Jacobian has been found previously in the literature but without comment on its limitations. Section 1 discusses the Jacobian and its relation to the literature.

The Euler Jacobian can be used to greatly simplify the Cartesian feedback linearization of the robot manipulator. The results of such a linearization are presented in Section 3.2. The problems with such a linearization are noticeable primarily in the increase in the required calculations. An additional difficulty arises in the meaning of compliant motion for a position/force controller built upon the Euler Jacobian. This is discussed in Section 3.3.

3.1 Euler Jacobian

The term Euler Jacobian will be defined as the Jacobian matrix that provides the relation between incremental joint motions and incremental Cartesian motions; where Cartesian motions are described by linear position changes and Euler angle changes.

From the work by (Paul, 1980), and (Craig, 1986), the common Jacobian provides the relation:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = J(q) \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \\ \dot{q}_5 \\ \dot{q}_6 \end{bmatrix} . \tag{3.1}$$

From the previous chapter, the matrix B^{-1} was used to transform components of the angular velocity vector to Euler angle velocities,

$$\begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} = B^{-1} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} . \tag{3.2}$$

The matrix B^{-1} is a function of α, β, γ, and is undefined at the points $\beta = \pm\frac{\pi}{2}$.

Then,

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & B^{-1} \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & B^{-1} \end{bmatrix} J(q) \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \\ \dot{q}_5 \\ \dot{q}_6 \end{bmatrix}. \tag{3.3}$$

The matrix $J_E = \begin{bmatrix} I & 0 \\ 0 & B^{-1} \end{bmatrix} J(q)$ is the Euler Jacobian. As a combination of both the previous Jacobian matrix and the B^{-1} matrix, it has the property of being zero at points in the workspace as well as being undefined, or having elements values approaching infinity, at other points in the workspace.

Relation to the Literature

It is quite common in the robotics literature to define a 6×1 vector representing Cartesian position. The first three elements contain the position of the end effector while the last three elements contain the orientation, represented as a set of three Euler angles. The complete Cartesian vector may be specified as x_e. A nonlinear function is assumed available which gives the Cartesian position x_e as a function of the joint angles.

$$x_e = \Theta(q) \tag{3.4}$$

The function is differentiated with respect to the joint angles and the resultant matrix is the manipulator Jacobian with respect to Euler angles.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} = \frac{\partial \Theta}{\partial q} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \\ \dot{q}_5 \\ \dot{q}_6 \end{bmatrix} \qquad \text{or,} \qquad \dot{x}_e = J_E \dot{q} \tag{3.5}$$

A small sample of references that use this method to derive the Euler Jacobian are: Hayati, 1986; Khatib, 1985; Kosuge, 1987; Tarn, 1986b, 1987; Uchiyama, 1987; Yoshikawa, 1986; Raibert and Craig, 1981; Unseren, 1987.

This Jacobian has the same properties as that presented in the previous section including the property that it becomes undefined at certain orientations. It is important to note that for whatever form of Euler angles applied, there will be some rotation which is poorly defined. When these Euler angles are combined with the Jacobian, the result is an Euler Jacobian with elements that will have infinite value for specific manipulator orientations. In the implementation of robot controllers that use such a Jacobian, other methods must be used in the vicinity of these orientations.

3.2 Application to the Cartesian Controller

Using the Euler Jacobian, J_E, the Cartesian controller from section 2.2 may be redefined (Khatib, 1985). Differentiating equation 3.5 provides the relation

$\ddot{x}_e = J_E\ddot{q} + \dot{J}_E\dot{q}$. Following the substitution into equation 2.1 gives the model of the manipulator in Cartesian-Euler space,

$$\ddot{x}_e = J_E D^{-1}(\tau - fg) + \dot{J}_E\dot{q} \tag{3.6}$$

If the torque to each joint is calculated as:

$$\tau = D(q)J_E^{-1}(q)(u - \dot{J}_E(q)\dot{q}) + fg(q, \dot{q}) \tag{3.7}$$

then the linearized system becomes:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \ddot{\alpha} \\ \ddot{\beta} \\ \ddot{\gamma} \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_\alpha \\ u_\beta \\ u_\gamma \end{bmatrix} \tag{3.8}$$

Assuming that an accurate model of the manipulator is calculated.

The result is truly a system of decoupled double integrators. There is no problem with the integration of $\dot{\alpha}$ to α as a measure of the rotation. The Euler angles and Euler angle velocities may be used in the design of a regulator or tracker for the double integrator. However, this linearization fails to exist for certain orientations mentioned above and it hides the expense of calculating the Euler angles.

3.3 Implications for Position/Force Control

Following the work presented by Raibert and Craig (Raibert and Craig, 1981), a position/force controller may be built on top of the Cartesian controller of Sections 2.1 or 3.2. This involves the definition of a selection matrix whose diagonal elements indicate whether a given Cartesian direction is to be position controlled or force controlled. While the complete details of such a controller will not be repeated here, some comments on the impact of the Euler Jacobian will be discussed.

A position/force controller requires some device for measurement of the forces and moments exerted on the end effector of the manipulator. A common such device is a 6 DOF force wrist which measures the forces along the x, y, z directions and moments around the x, y, z axes. Inclusion of the force measurements into the controller of Section 2.1 is straightforward and involves the comparison with a desired force signal for the force regulator and the multiplication by the selection matrix before being converted to joint torques by the Jacobian transpose. The resulting position/force controller for Section 2.1 will control position and rotation of the end effector as well as forces along the x, y, z directions and moments around the x, y, z axes.

A position/force controller for the Cartesian model from Section 3.2 requires a few more calculations due to the Euler Jacobian. Such a position/force controller will control forces along the x, y, z directions and *generalized forces* along the Euler angles. Thus, all measurements from the force wrist must first be converted to the coordinate system of the controller. The conversion requires the B matrix and is presented below.

From Equation 3.3,

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & B^{-1} \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad \text{or,} \quad \dot{x}_e = \begin{bmatrix} I & 0 \\ 0 & B^{-1} \end{bmatrix} \dot{x}_c \tag{3.9}$$

The work done by an applied external force through an incremental motion must be the same in both systems of measurement. Thus,

$$\delta W = \delta x_e^T F_e = \delta x_c^T F_c \tag{3.10}$$

where F_c is the 6×1 force/moment vector in the Cartesian space containing forces along the x, y, z directions, moments around the x, y, z axes, and F_e is the force vector in Euler space, that is, forces along the x, y, z directions and generalized forces along the Euler angles.

Applying the relation in Equation 3.9 gives:

$$\begin{bmatrix} I & 0 \\ 0 & B^{-1} \end{bmatrix}^T F_e = F_c \tag{3.11}$$

Since the matrix B always exists,

$$F_e = \begin{bmatrix} I & 0 \\ 0 & B \end{bmatrix}^T F_c \tag{3.12}$$

Forces measured by the force wrist may always be converted to forces usable by the Euler Jacobian controller for every manipulator orientation.

Compliance

While force measurements may be converted to the Euler coordinates, there still exists the problem of compliance with a controller that is built using the Euler Jacobian. Through the selection matrix one may choose which axes are to be position controlled or force controlled. With the Euler angle controller, rotational compliance will be selected along the Euler angles. When only one rotational axis is in compliance, the effect is the same as compliance in the normal Cartesian space because, with Z-Y-X Euler angles, single Euler angle rotation corresponds to rotation about one of the x, y, z axes. However, when two rotational axes are to comply to external forces, Euler angle rotations and rotations about two of the x, y, z axes do not correspond. The performance of complex compliant robot motions will depend upon which form of Jacobian is used in the controller.

In the discussions of the previous sections, it has been assumed that Z-Y-X Euler angles where being used. For the most part, the results remain independent of the form of Euler angles, with the exception of the problem of compliance. Clearly compliance will be a problem if, for example, Z-Y-Z Euler angles are used and rotational compliance is desired about the x axis.

3.4 Summary

There are many types of Jacobians present in the robotics literature, and this chapter has dealt with the details involved with two: the common Jacobian, and the Euler Jacobian. Two methods for generating the Euler Jacobian were presented; one using methods from Chapter 2, and one from the definitions in the literature. The Euler Jacobian was used to generate another form of the Cartesian controller for a robot manipulator. The advantages to the generated Cartesian controller are the much simpler linearization and subsequent ability to prove stability. The disadvantages are the increased complexity of required calculations, the inability to control on the boundary points of the rotation, and the difficulties with compliance. While these problems are rarely mentioned in the literature they are extremely important to the implementation and understanding of Cartesian control.

4. Conclusion

The goal of this work has been to present some of the difficulties involved in the control of manipulator orientation in Cartesian control. This work is important not only for Cartesian control and implementation, but also for understanding the impact that Cartesian control decisions will have on force control and compliance.

A straightforward manipulator Cartesian control scheme was presented to introduce the source of the orientation control problem. The problem is complicated by the requirements both to represent orientation and control orientation. Different forms of representation have different advantages for the purpose of control.

The dynamic models of orientation for two of the most common orientation representations where presented and three regulators based upon these representations were investigated. The three regulators were evaluated on the basis of calculation cost and performance. Reasonable performance was achieved with two of the regulators but no solution proved optimal in both complexity and performance. The results compliment those achieved by Kreutz and Wen, (Kreutz and Wen, 1988).

In Chapter 3 the representation of rotation with Euler angles was included in the manipulator Jacobian to produce the Euler Jacobian. This Jacobian is commonly used in the literature without statement to the fact that it must exclude certain orientations and has a higher calculation cost. Force compliance using the Euler Jacobian produces compliance around the Euler angles and not necessarily about the Cartesian axes.

In summary, the problem of controlling manipulator orientation in Cartesian space has not been satisfactorily solved. Representations of orientation must also take into account the requirements for control. Additionally, the choice of representation for orientation has consequences for the performance of force compliance, path and trajectory planning, and the interpretation of sensor data.

Areas of further work include the development of a better rotation matrix based regulator and the examination of other forms of orientation representation.

References

ÅSTRÖM, K.J. (1982): "A Simnon Tutorial," Report CODEN: LUTFD2/ TFRT-3168, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

CRAIG, J.J. (1986): *Introduction to Robotics, Mechanics & Control*, Addison-Wesley Publishing Company, Reading, Mass.

GOLDSTEIN, H. (1980): *Classical Mechanics*, Addison-Wesley Publishing Company, Reading, Mass.

HAYATI, S. (1986): "Hybrid Position/Force Control of Multi-Arm Cooperating Robots," *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 82–89, April.

KHATIB, O. (1985): "The Operational Space Formulation in Robot Manipulator Control," *Proceedings of the 15th ISIR*, pp. 165–172, September.

KHATIB, O. and J. BURDICK (1986): "Motion and Force Control of Robot Manipulators," *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1381–1386.

KOSUGE, K., K. FURUTA, and T. YOKOYAMA (1987): "Virtual Internal Model Following Control of Robot Arms," *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1549–1554.

KREUTZ, K., and J.T. WEN (1988): "Attitude Control of an Object Commonly Held by Multiple Robot Arms: A Lyapunov Approach," *Proceedings of the 1988 American Control Conference*, pp. 1790–1794, June.

LUH, J.Y.S., M.W. WALKER, and R.P.C PAUL (1980): "Resolved - Acceleration Control of Mechanical Manipulators," *IEEE Transactions on Automatic Control*, pp. 468–474, June.

MASON, M.T. (1981): "Compliance and Control for Computer Controlled Manipulators," *IEEE Transactions on Systems, Man, and Cybernetics*, June.

PAUL, R.P. (1981): *Robot Manipulators: Mathematics, Programming, and Control*, The MIT Press, Cambridge, Mass..

RAIBERT, M.H. and J.J. CRAIG (1981): "Hybrid Position/Force Control of Manipulators," *Journal of Dynamic Systems, Measurement and Control*, **Vol. 102**, June.

TARN, T.J., A.K. BEJCZY, and X. YUN (1986a): "Coordinated Control of Two Robot Arms," *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1193–1202, April.

TARN, T.J., A.K. BEJCZY, and X. YUN (1986b): "Dynamic Coordination of Two Arms," *Proceedings of the 25th Conference on Decision and Control*, December.

TARN, T.J., A.K. BEJCZY, and X. YUN (1987): "Design of Dynamic Control of Two Cooperating Robot Arms: Closed Chain Formulation," *Proceedings* *of the IEEE International Conference on Robotics and Automation*, pp. 7–13.

UCHIYAMA, M., N. IWASAWA, and K. HAKOMORI (1987): "Hybrid Position/Force Control for Coordination of a Two-Arm Robot," *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1242–1247.

UNSEREN, M.A., and A.J. KOIVO (1987): "Kinematic Relations and Dynamic Modeling for Two Cooperating Manipulators in Assembly," *Proceedings of the 1987 International Conference on Systems, Man, and Cybernetics*, October.

YOSHIKAWA, T. (1986): "Dynamic Hybrid Position/Force Control of Robotic Manipulators," *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1393–1398, April.