Globally Optimal Kinematic Control of Redundant Manipulators with Application to REDIESTRO

Ming Liang Dong

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Applied Science at

Concordia University

Montreal, Quebec, Canada

March 1994

ISBN 0-315-90845-9

ABSTRACT

Globally Optimal Kinematic Control of Redundant Manipulators

with Applications to REDIESTRO

Ming Liang Dong

This thesis presents a global optimization approach for resolving redundancy in kinematic control of redundant manipulators. The approach which is termed Globally Optimal Kinematic Control (GOKC), minimizes a performance criterion of integral type, and yields an optimal solution at the joint velocity level. The formulation of the performance criterions is such that the approach suppresses large joint velocities near singularities while reducing tracking errors, so that a singularity-robust control is obtained. Furthermore, the approach can be carried out in real-time for a class of desired end-effector trajectories. This is an important consideration in practical applications, and is also the main advantage of the approach over other global optimization methods. The approach is illustrated by extensive simulation studies for a variety of redundancy resolution goals for REDIESTRO, a kinematically isotropic redundant manipulator constructed in the Centre for Intelligent Machines at McGill University.

DEDICATION

To my dear mother, Mrs. Zhong Meiyin, with love and reverence.

ACKNOWLEDGEMENTS

I owe very special thanks to my supervisor, Dr. Rajnikant Patel, for guiding my understanding of robot manipulator control as well as for the direction and financial support provided to me during the course of this research.

I would also like to thank all my friends and fellow graduate students for the mental support and friendship. I wish to particularly thank my friends Zhengcheng, Rajeev, Farzam, Yonghai for their numerous suggestions and helps.

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION

1.1 INTRODUCTION

Over the last two decades, robots have played an increasingly important role especially in industry. Because of their stability and precision as well as their ability to function in an environment which is impossible or difficult for humans to work in, robots have found their way into almost all walks of life. The increasing demand and the potential of robots have thus inspired more and more research on the design and control of robot manipulators. An important aspect of this research has been the focus on nonredundant manipulators as opposed to redundant manipulators. A manipulator is said to be redundant if it has more degrees of freedom than are required to accomplish a particular task. Although the kinematic and dynamic control of nonredundant manipulators has attained a certain degree of maturity, some inherent shortcomings of this type of manipulators have hindered their applications in practice. On the other hand, due to its dexterity and versatility, the redundant manipulator makes it possible to implement a significantly larger number of practical tasks. Therefore, during the last ten years, research attention has turned to the area of redundant manipulators.

Focusing on redundant manipulators, this thesis discusses methods for solving the optimal redundancy resolution problem, and proposes a new global optimization algorithm. In the rest of this chapter, some background on redundant manipulators and redundancy resolution, the motivation for this research, and the structure and contents of the thesis, are described.

1.2 MANIPULATOR KINEMATICS

The position and orientation of a manipulator's end-effector can be defined by the 4×4 homogeneous transform matrix [1]

$$T = \left[\begin{array}{ccc|c} & R & & p \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \tag{1.1}$$

where $R = \{r_{ij}\}$ is the 3×3 end-effector rotation matrix, and $p = [x, y, z]^T$ is the 3×1 end-effector position vector, with respect to the base. One common representation of the end-effector orientation is the set of roll-pitch-yaw angles (α, β, γ). These three parameters are related to the elements r_{ij} of R as follows [2]:

$$\alpha = \mathrm{Atan2}(r_{32}, r_{33}) \tag{1.2}$$

$$\beta = \mathrm{Atan2}(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2}) \tag{1.3}$$

$$\gamma = \mathrm{Atan2}(r_{21}, r_{11}) \tag{1.4}$$

where Atan2 is the two-argument arc tangent function, and the pitch angle β is assumed to be within $(-90^o, 90^o)$. Therefore, the end-effector position and orientation can be described by a 6×1 vector $X = [x, y, z, \alpha, \beta, \gamma]^T$ in the manipulator's task space. Conventionally, the kinematic relation of the manipulator, which maps the manipulator's joint space into its work space, is defined by

$$X = f(\theta) \tag{1.5}$$

where θ is the $m \times 1$ joint configuration vector. It is easy to see that m should be equal to or greater than six for equation (1.5) to be always solvable, i.e., it is necessary for a manipulator to have six or more joints in order to achieve any arbitrary pose (position and orientation) in its workspace.

The linear velocity v and the angular velocity ω of the end-effector with respect to the base coordinate system can be computed using the vector cross-product form [3] [4]

$$v = \sum_{i=1}^{m} b_i \dot{\theta}_i = M_{vt} \dot{\theta} \tag{1.6}$$

$$\omega = \sum_{i=1}^{m} c_i \dot{\theta}_i = M_{vr} \dot{\theta} \tag{1.7}$$

where

$$b_i = \begin{cases} \hat{z}_i \times P_i & \text{when link } i \text{ is rotational} \\ \hat{z}_i & \text{when link } i \text{ is translational} \end{cases} \tag{1.8}$$

$$c_i = \begin{cases} \hat{z}_i & \text{when link } i \text{ is rotational} \\ 0 & \text{when link } i \text{ is translational} \end{cases} \tag{1.9}$$

where $\hat{z}_i$ is the unit vector along the z-axis of link frame $\{i\}$, and P_i is the position vector from the origin O_i of link frame $\{i\}$ to the origin of the end-effector frame $\{m\}$. We can combine v and ω into a 6×1 velocity vector of the end-effector to yield

$$V = \begin{bmatrix} \omega \\ v \end{bmatrix} = \begin{bmatrix} M_{vr} \\ M_{vt} \end{bmatrix} \dot{\theta} \tag{1.10}$$

In order to relate the joint velocities to the rate of change of the roll-pitch-yaw angles that represent the end-effector orientation, the rotational part in (1.10) should be modified to yield [5]

$$\frac{d}{dt}\begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} 0 & -\sin\gamma & \cos\gamma\cos\beta \\ 0 & \cos\gamma & \sin\gamma\cos\beta \\ 1 & 0 & -\sin\beta \end{bmatrix}^{-1} M_{vr}\dot{\theta} = LM_{vr}\dot{\theta} \tag{1.11}$$

From (1.10) and (1.11), we obtain the $6 \times m$ end-effector Jacobian matrix

$$J = \begin{bmatrix} LM_{vr} \\ M_{vt} \end{bmatrix} \tag{1.12}$$

which relates $\dot{X}$ and $\dot{\theta}$ as

$$\dot{X} = J\dot{\theta} \tag{1.13}$$

Equation (1.13) is called the Jacobian relation of the manipulator.

1.3 REDUNDANT MANIPULATORS AND REDUNDANCY RESOLUTION

It has been found that the configuration singular regions of robot manipulators near which a small translation or orientation in work space requires physically unrealizable joint speeds, significantly limit the work space of a robotic system. A successful means of overcoming this problem is to introduce some kinematic redundancy. In recent years, some researchers have solved a variety of tasks requiring sophisticated mechanical motion in an unpredictable and dynamically varying environment. These tasks also motivated the design of redundant robot manipulators. For example, while a nonredundant system is very limited in performing a task for external obstacle avoidance, a kinematically redundant system may successfully solve this task [6] [7]. In addition, redundant systems are capable of optimizing various performance criteria such as joint-limit avoidance [8], minimum joint motion [9], minimum joint velocity [10], minimum joint acceleration, minimum joint torque [11][12], minimum kinetic energy [13], minimum impact force etc.. Manipulator redundancy can also be used to solve problems such as singularity avoidance [14] and posture control [9]. Recently, it has been shown [15] that kinematic redundancy can be used to address the problem of joint flexibility in robot manipulators.

Consider the kinematic relation of the manipulator

$$X = f(\theta), \tag{1.14}$$

where $X \in \Re^n$ is the vector containing workspace variables that denotes the end-effector position and orientation, and $\theta \in \Re^m$ is the joint vector. Most conventional robot manipulators are kinematically nonredundant, i.e., $n = m$. If $n < m$, the manipulator is said to be kinematically redundant. As discussed above, any motion task for manipulators can be generally defined by six independent task variables, in which three variables denote the position of the end-effector and the others denote the orientation of the end-effector, it is usually assumed that a kinematically redundant manipulator should have more than six joints or six degrees of freedom (d.o.f.'s). However, in general, we can say that any robot manipulator is kinematically redundant with *degrees of redundancy* $r = m - n$, where n denotes the number of task variables, and m the number of joints or d.o.f.'s. For instance, a four-link manipulator has two degrees of redundancy when it performs a positioning task in a 2-dimensional plane. Note that, in redundant manipulators, there is usually a continuum of alternative configurations that yields the same end-effector position and orientation. Therefore, the control problem for redundant manipulators is to find a proper joint space trajectory from a possibly infinite number of joint space trajectories, which produces the desired end-effector motion, and at the same time meets the need for achieving some additional task(s).

Fundamental approaches for resolving redundancy can be classified into two categories: (i) *local optimization methods*, and (ii) *global optimization methods*. Most of the approaches developed to date are local optimization methods. They are based on local information, and give joint trajectories that depend only on the local behavior of the work space path. Accordingly, sensor-based real-time control can be implemented, and the formulation is relatively simple as compared to global optimization methods. Local optimization, however, has some inherent drawbacks. First, global optimality cannot be guaranteed. In general, the solutions are suboptimal compared to those obtained using global methods. Second, motion control under a local optimization algorithm may lead to some undesired effects. For example, problems of high joint speed may arise, and in some cases, a closed path in the manipulator's work space does not yield a closed path in its joint space, i.e. the motion is nonconservative. A more serious problem is that certain numerical and algorithmic instabilities may occur, e.g., see [16][17][18].

In contrast, global optimization methods compute the joint trajectories from complete information of the workspace path, so that the two drawbacks of the local optimization can be overcome. However, global optimal control methods reported to date are limited to off-line approaches, and do not allow real-time path corrections based on sensor measurements. Besides, global optimization usually requires expensive computations, which also rules out real-time control in practice.

1.4 MOTIVATION FOR THIS THESIS

The motivation for the research presented here comes from the intent in redundant manipulators for practical applications, especially in space robotics. The aim of the research is to examine the nature of local and global optimization methods for redundancy resolution, and to show the advantages and shortcomings of the two types of methods. In recent years, a significant amount of literature has appeared concerning control of kinematically redundant manipulators. The following are the main characteristics of the strategies developed so far. First, the common thread in all these studies is that redundancy should be resolved in such a way that the manipulator optimizes some performance criteria while carrying out its given task. Second, while many researchers have proposed redundancy resolution methods involving optimized solutions to the inverse kinematics problem, there are very few techniques that use global optimization. Finally, most of the global methods can only be implemented in off-line strategies. This thesis makes a contribution to the area of redundancy resolution by developing novel techniques based on global optimization, while attempting to achieve efficient computational solutions for real-time kinematic control of redundant manipulators. This thesis also includes the results of several computer simulations that illustrate the techniques for obtaining globally optimal redundancy resolution for an actual 7 d.o.f.'s robot manipulator.

1.5 OUTLINE OF THE THESIS

The thesis is organized as follows:

In Chapter 2, a review of local and global optimizing algorithms for redundancy resolution is given. Most of the local optimization methods utilize a formulation of inverse kinematics at the velocity level. Recall the Jacobian relation of the manipulator

$$\dot{X} = J\dot{\theta}, \tag{1.15}$$

where $J \in \Re^{n \times m}$ is the Jacobian matrix corresponding to $f(\theta)$. In the case of redundancy $(m > n)$, equation (1.15) is underdetermined. Section 2.1 introduces three major local methods for solving the underdetermined equation (1.15) while minimizing a given performance criterion. These methods resolve the redundancy by utilizing: (1) a particular solution of (1.15), which can be defined by means of a generalized inverse of the Jacobian matrix; (2) the general solution of (1.15), which can be represented as the sum of a particular solution of (1.15) and the general solution of the homogeneous equation $J\dot{\theta} = 0$; and (3) a nonredundant system, in which the dimension of the workspace is extended by incorporating $m - n$ additional constraints. Two global optimization methods are presented in Section 2.2. One of them resolves the redundancy at the joint acceleration level based on the calculus of variations. The other gives the optimal solution at the velocity level based on Pontryagin's maximum principle, in which the solution is obtained by solving a two-point boundary value problem. The results of these two methods show the main difficulties in real-time global optimization. These problems, along with other unsolved problems in optimal redundancy resolution, are discussed in Section 2.3.

In Chapter 3, a new global optimization scheme for redundancy resolution called *globally optimal kinematic control* (GOKC) is presented. The real-time computation of this method is an important advantage over other global methods. Section 3.1 describes the formulation of the GOKC problem in the framework of using redundancy to minimize the weighted kinetic energy. The GOKC control law is derived, and properties of the optimal solution are investigated. Section 3.2 deals with the problem of real-time GOKC. A recursive algorithm for real-time optimal tracking control is derived. This real-time algorithm can be used for tasks in which the desired end-effector trajectories end with zero velocity. The stability of the real-time GOKC is proved in Section 3.3. Finally, Section 3.4 proposes several variations in using GOKC for redundancy resolution.

In Chapter 4, an isotropic redundant manipulator is considered. The kinematic control problem is solved using GOKC. Control simulations for different redundancy resolution goals are carried out. Section 4.1 introduces the 7 d.o.f.'s isotropic redundant manipulator REDIESTRO. The rest of this chapter presents five control simulations for resolving redundancy to optimize certain performance criteria. They are: (1) minimum norm of joint velocities, which minimizes $\dot{\theta}^T\dot{\theta}$ globally; (2) optimal joint motion, in which the user can restrict the motion of any joint at the expense of the motions of the other joints; (3) joint limit avoidance, in which a new method is proposed to achieve the goal; (4) obstacle avoidance; (5) posture control, which utilizes redundancy to adjust the arm configuration. Finally, Chapter 5 presents some concluding remarks and suggests some future work in the framework of GOKC.

1.6 REFERENCES

[1] R. C. Paul, *Modeling Trajectory Calculation and Servoing of a Computer Control Arm*, Stanford Artificial Intelligence Lab., Stanford Univ., Stanford, CA., Memo 177, Sept. 1972.

[2] J. J. Craig, *Robotics - Mechanics and Control*, Reading, MA: Addison Wesley, 1989, pp. 45-48.

[3] R. C Paul, J. Y. S. Luh, et al., *Advanced industrial robot control systems*, Rep. Period

July 1, 1977 to Jan. 1, 1978, Purdue University, IN. TR-EE 78, May 1978.

[4] J. Y. S. Luh, M. W. Walker, and R. C. Paul, "On-line Computational schemes of mechanical manipulators", ASME, *J. Dyn Syst. Meas., Contr.*, Vol. 102, June 1980.

[5] H. Seraji M. K. Long, and T. S. Lee, "Motion Control of 7-DOF Arms: The Configuration Control Approach", *IEEE Trans Rob. and Automat.*, Vol. 9, No. 2, 1993, pp. 125-139.

[6] A. A. Maciejewski and C. A. Klein, "Obstacle Avoidance for Kinematically Redundant Manipulators in Dynamically Varying Environments", *Int J. Rob. Res.*, Vol. 4, No. 3, Fall 1985.

[7] R. Colbaugh et al, "Obstacle Avoidance for Redundant Robots Using Configuration Control", *J. Rob. Syst.*, Vol. 6, No. 6, 1989, pp. 721-744.

[8] H. Seraji, "Improved Configuration Control for Redundant Robots", *J. Rob. Syst.*, 7(6), 1990, pp. 897-928.

[9] H. Seraji, M. Long, and T. Lee, "Configuration Control of 7-DOF Arms", *Proc. IEEE Int. Conf. on Robotics and Automation*, Sacramento, California-April 1991.

[10] H. Seraji, "Task-Based Configuration Control of Redundant Manipulators", *J. Rob. Syst.*, 9(3), 1992, pp. 411-451.

[11] A. Nedungadi, "Torque Optimization of Redundant manipulators", *Proc. 17th ISIR*, Chicago, Illinois, 1987, pp. 1016-1021.

[12] H. Seraji, "New Goals for redundancy Resolution Using Configuration Control", *J. Rob. Syst.*, 1990.

[13] O.Khatib, "Dynamic Control of Manipulators in Operational space", *Proc. of the 6th World Congress on Theory of Machines and Mechanisms*, New Delhi, India, 1983, pp. 1128-1131.

[14] R. V. Mayorga, A. K. C. Wong, and S. K. Ma, "An Efficient Local Approach for the Path Generation of Robot Manipulators", *J. Rob. Syst.*, Vol. 7.1, Feb. 1990.

[15] Z.C.Lin and R.V.Patel, "Cartesian Control of Redundant Flexible-Joint

Manipulators", *Proc. IEEE Conf. on Dec. & Contr.*, San Antonio, TX, 1993.

[16] K. Kazerounian and A. Nedungadi, "Redundancy Resolution of Robotics manipulators at the acceleration level", *Proc. 7th World Congress on The Theory of Machines and Mechanisms*, Seville, Spain, Sept. 1987.

[17] K. Kazerounian and A. Nedungadi, "An Alternative Method for Minimization of Driving Forces in Redundant Manipulators", *IEEE Int. Conf. on Robotics and Automation*, Raleigh, N.C., 1987.

[18] J. Hollerbach and K. Suh, "Redundancy Resolution of Manipulators through Torque Optimization", *IEEE Int. Conf. on Robotics and Automation*, St. Louis, MO, March 1985.

CHAPTER 2 OPTIMIZATION SCHEMES FOR REDUNDANCY RESOLUTION

Since a redundant manipulator has more degree-of-freedoms than are necessary to fulfill its primary motion task, we may utilize these "redundant" degrees-of-freedom to optimize some performance criteria while the end-effector tracks a desired trajectory. Optimization-based redundancy resolution usually involves one of two types of performance criteria: the instantaneous performance criterion, and the integral performance criterion. The instantaneous criterion containing the local path information leads to local optimal redundancy resolution that optimizes the performance criterion at each step in time. Generally, the local optimal resolution is not unique so that a closed trajectory in the work space may not yield a closed trajectory in the joint space, i.e. the motion is nonconservative. This problem is further discussed in section 2.3.3. In contrast, the integral performance criterion containing the entire information of the basic task yields globally optimal redundancy resolution provided that the initial joint configuration is given. This resolution is unique and therefore leads to conservative motion.

Many schemes for optimal redundancy resolution have been reported to date, and most of them use local optimization methods. Very few global optimization algorithms have been proposed, and the problem of real-time globally optimal redundancy resolution is as yet unsolved.

This chapter surveys both local and global optimization based kinematic control schemes for redundant manipulators. The advantages and drawbacks of each scheme are also discussed. This survey provides the motivation for the idea of a new global optimization based scheme which can be implemented in real-time. The detailed description of this new kinematic controller is presented in Chapter 3.

2.1 LOCAL OPTIMIZATION FOR REDUNDANCY RESOLUTION

Most local optimization methods for redundancy resolution use repeated solutions of the differential relations between joint space coordinates and work space coordinates. That is, if the transformation from joint space to work space is defined as

$$X = f(\theta) \tag{2.1}$$

where X is a $n \times 1$ vector denoting the position and orientation of the end-effector in work space with respect to the base-frame coordinates, and θ is a $m \times 1$ joint space vector, $n < m$, and the differential relation for equation (2.1) is given by

$$\dot{X} = J(\theta)\dot{\theta} \tag{2.2}$$

where $J(\theta)$ is the Jacobian matrix of $f(\theta)$. Then a solution $\dot{\theta}$ of equation (2.2) is obtained at discrete time steps during the motion of the end-effector. As equation (2.2) is solved repeatedly in real-time, $\dot{X}$ can be modified according to sensor measurements or interactive human commands. Several methods have been proposed to solve this equation for redundant manipulators, each differing in the manner in which a unique $\dot{\theta}$ is chosen among many solutions to this equation. The most common approaches for local optimal redundancy resolution are outlined below.

2.1.1 MINIMUM NORM CRITERION

Whitney was the first researcher to discuss redundant manipulators, and to suggest resolving redundancy by using a performance criterion [1]. That is, the manipulator's motion must satisfy some performance criterion while executing a given task. This performance criterion is usually a quadratic optimality index, for example,

$$\min_{\theta} \; L = \frac{1}{2}\dot{\theta}^T A \dot{\theta} \tag{2.3}$$

where A is a positive definite weighting matrix. The problem for resolving redundancy then becomes a constrained optimization problem, i.e. minimize L in (2.3) subject to equation (2.2). The solution for this problem can be obtained using the method of Lagrange multipliers. Adjoining equation (2.3) and (2.2) using Lagrange multipliers, we get

$$L' = \frac{1}{2}\dot{\theta}^T A \dot{\theta} + \lambda (\dot{X} - J\dot{\theta}) \tag{2.4}$$

where λ a $1 \times n$ Lagrange multiplier vector. The necessary conditions for minimizing L' has been found as

$$\frac{\partial L'}{\partial \lambda} = \dot{X} - J\dot{\theta} = 0 \tag{2.5}$$

and

$$\frac{\partial L'}{\partial \dot{\theta}} = \dot{\theta}^T A - \lambda J = 0. \tag{2.6}$$

Note that equation (2.5) is the same as equation (2.2). Solving equation (2.6) for $\dot{\theta}$, we get

$$\dot{\theta}^T = \lambda J A^{-1}. \tag{2.7}$$

Assuming that J has full rank, by multiplying J^T in the both sides of equation (2.7) and then using equation (2.5) in the resultant equation, λ is obtained as

$$\lambda = \dot{X}^T (J A^{-1} J^T)^{-1}. \tag{2.8}$$

From equation (2.7) and (2.8), the optimal $\dot{\theta}$ is then given by

$$\dot{\theta}^T = \dot{X}^T (J A^{-1} J^T)^{-1} J A^{-1}. \tag{2.9}$$

Note that this is a particular solution of equation (2.2), and the solution optimizes weighted kinetic energy function L.

The weighting matrix A in (2.3) can be chosen so as to meet different performance criteria. The simplest way is to choose A as an $m \times m$ unit matrix. The optimality index then becomes $\frac{1}{2}\dot{\theta}^T\dot{\theta}$ which is proportional to the norm of the joint velocities. The optimal solution minimizing the norm of the joint velocities is then obtained directly from (2.9) as

$$\dot{\theta} = J^T (J J^T)^{-1} \dot{X}. \tag{2.10}$$

Next, we use the concept of a generalized inverse [2], denoted by P^*, of an $n \times m$ matrix P, the matrix P^* is characterized by the following equation

$$P P^* P = P \tag{2.11}$$

and in the case of $rank(P) = n\ (<m)$,

$$PP^{*} = I_n \tag{2.12}$$

where I_n is the $n \times n$ unit matrix. Moreover, the Moore-Penrose generalized inverse or pseudoinverse [2], which we shall denote by $P^{\dagger}$, in addition to equation (2.11), satisfies

$$P^{\dagger} P P^{\dagger} = P^{\dagger}, \tag{2.13}$$

$$(P^{\dagger} P)^{T} = P^{\dagger} P, \tag{2.14}$$

and

$$(P P^{\dagger})^{T} = P P^{\dagger}. \tag{2.15}$$

When $rank(P) = n < m$, the pseudoinverse is given by

$$P^{\dagger} = P^{T} (P P^{T})^{-1}. \tag{2.16}$$

Thus, equation (2.10) can be written in the well-known form

$$\dot{\theta} = J^{\dagger} \dot{X}. \tag{2.17}$$

Another choice of the weighting matrix A may be the inertia matrix of the manipulator. This leads to a solution which gives the minimum kinetic energy of the manipulator [3].

We can also choose A so as to emphasize the role of some components of X and deemphasize others, e.g., by heavily penalizing the motions of the former relative to the latter. To synthesize the required A for this purpose, we begin with the optimality criterion

$$\min_{\dot{X}} V = \frac{1}{2}\dot{X}^T B \dot{X} \tag{2.18}$$

where B is chosen to be positive definite and provide relative emphasis, e.g., we can change the end-effector's orientation while keep its location fixed. Substituting equation (2.2) into equation (2.18) gives

$$\min_{\dot{\theta}} V = \frac{1}{2}\dot{\theta}^T J^T B J \dot{\theta} \tag{2.19}$$

Comparing equation (2.19) and (2.3), we can define A as

$$A = J^T B J. \tag{2.20}$$

Although the methods that use the minimum norm criteria mentioned above are appealing because of their generality, they have some drawbacks. First, these methods have the general undesirable property that repetitive end-effector motion does not necessarily yield repetitive joint motion [4]. However, there are some exceptions, e.g. [5], where repetitive joint motion is achieved. Worse, although the motion is pointwises optimal, the manipulator configuration can blunder into a region near a singular point, where the minimum norm is unacceptably large [6]. Due to these drawbacks, the control methods based on equation (2.9) are rarely used in practice.

2.1.2 GRADIENT MINIMIZATION OF POTENTIAL FIELDS

The main idea in developing the general solution of equation (2.2), originally due to Liegeois [7], derives intuitively from the fact that it is possible to add to the solution of (2.17) any vector consistent with the constraints. This is asserted by the following theorem.

Theorem 2.1.

The set of linear consistent equations such as (2.2) with $rank(J) = n < m$, admits a solution as

$$\dot{\theta} = J_1^* \dot{X} + (I_m - J_2^* J) Z \tag{2.21}$$

where J_1^* and J_2^* are the generalized inverses of J, I_m is the $m \times m$ identity matrix, and Z is an arbitrary $m \times 1$ vector.

The proof of this theorem is straightforward. From equation (2.21), (2.11) and (2.12), we have

$$J\dot{\theta} = JJ_1^* \dot{X} + (J - JJ_2^* J) Z = JJ_1^* \dot{X} = \dot{X} \tag{2.22}$$

The matrix $(I_m - J_2^* J)$ in equation (2.21) is a projection operator, which projects the arbitrary vector Z onto the null space of J along the range-space of $J_2^* J$.

If we set

$$Z = k \nabla H(\theta) \tag{2.23}$$

where k is a real scalar, and $\nabla H(\theta)$ is the gradient of a function $H(\theta)$ with respect to θ, and if the projection operator $(I_m - J_2^* J)$ is chosen properly, then the component $(I_m - J_2^* J) Z$ serves to decrease $H(\theta)$ when $k < 0$, and to increase $H(\theta)$ when $k > 0$. A common choice for the operator is to implement the orthogonal projection, i.e.

$$(J_2^* J)^2 = J_2^* J \tag{2.24}$$

and

$$(J_2^* J)^T = J_2^* J. \tag{2.25}$$

It can be easily shown that equation (2.24) and equation (2.25) is satisfied if we set J_2^* to be the pseudoinverse of J, i.e.,

$$J_2^* = J^\dagger = J^T (J J^T)^{-1} \tag{2.26}$$

For simplicity, usually we set $J_1^* = J_2^* = J^\dagger$. Thus, the general solution (2.21) for equation (2.2) becomes

$$\dot{\theta} = J^\dagger \dot{X} + k (I_m - J^\dagger J) \nabla H(\theta). \tag{2.27}$$

The joint-velocity vector in (2.27) consists of two components $\dot{\theta}_p$ and $\dot{\theta}_g$, where $\dot{\theta}_p = J^\dagger \dot{X}$ stands for the particular solution of equation (2.2) which causes the desired motion of the end-effector, while $\dot{\theta}_g = k (I_m - J^\dagger J) \nabla H(\theta)$ stands for the general solution of the homogeneous equation $J\dot{\theta} = 0$. The term $\dot{\theta}_g$ drives $H(\theta)$ towards either a minimum or a maximum, depending on the sign of k, but does not affect the end-effector motion. The joint motion due to $\dot{\theta}_g$ is therefore called the "self-motion" of the manipulator.

One may chose $H(\theta)$ according to certain performance criteria so that equation (2.27) can be used to resolve redundancy. If one wants to reduce the value of $H(\theta)$, k should be a negative scalar in equation (2.27), and for increasing the value of $H(\theta)$, it should be a positive scalar.

The term $H(\theta)$ is called the *objective function* or *potential function*. In the literatures, the proposed potential functions include the so-called manipulability index [8], the inverse of the square of distances to obstacles [9] or limit stops [9] etc.. It can be shown that the joint trajectories which avoid singularities can also be generated by an appropriate choice of $\nabla H(\theta)$ in equation (2.27). However, it is difficult to construct a potential function that guarantees that singularities will indeed be avoided at all times without introducing large gradients that in themselves induce large joint velocities. This issue has not been thoroughly studied, and a comprehensive analysis is complicated by the fact that the trajectories of joints depend on the speed of the path in the work space.

2.1.3 EXTENDED JACOBIAN TECHNIQUE

The original work on the extended Jacobian technique is due to Baillieul [10]. As mentioned in the preceding sections, there are many possible joint configurations θ corresponding to a given end-effector configuration. One way to impose practical limits on the number of choices of θ for a given X is to insist that θ optimizes some potential function $H(\theta)$. For the purpose of optimizing $H(\theta)$ subject to equation (2.1), the following theorem is very useful.

<u>Theorem 2.2</u>

Let X be a given configuration of the end-effector, and let $\theta = \theta_o$ be a joint configuration setting at which $H(\theta)$ is extremized subject to the constraint $X = f(\theta)$. If

N_e is an $m \times (m-n)$ matrix belonging to the null-space of J, i.e. $J \cdot N_e = 0$, then the equation

$$N_e^T(\theta_o) \cdot \nabla_\theta H(\theta_o) = 0 \tag{2.28}$$

must hold.

The proof of this theorem can be found in [10].

Now, once an objective function $H(\theta)$ has been selected, we define a function

$$G(\theta) = N_e^T(\theta) \cdot \nabla_\theta H(\theta). \tag{2.29}$$

If a linkage is positioned with its end-effector at X so that $H(\theta)$ is extremized, then the equation

$$\begin{bmatrix} f(\theta) \\ G(\theta) \end{bmatrix} = \begin{bmatrix} X \\ 0 \end{bmatrix} \tag{2.30}$$

is satisfied. If the end-effector tracks a trajectory $X(t)$ along which the corresponding joint configuration $\theta(t)$ extremizes the objective function $H(\theta)$ at each point, then we have

$$\begin{bmatrix} f(\theta(t)) \\ G(\theta(t)) \end{bmatrix} = \begin{bmatrix} X(t) \\ 0 \end{bmatrix}. \tag{2.31}$$

Equation (2.29) consists of $(m-n)$ scalar equations, and the kinematic relation (2.1) has n scalar equations. If these equations are independent, then equation (2.31) has m independent nonlinear equations which now fully specify the m unknown joint angles θ.

Note that equation (2.31) has to be solved numerically.

If $G(\theta)$ is differentiable with respect to θ, an initial solution for (2.31) can be propagated along a path by solving the differential equation

$$J_{EX}\dot{\theta} = \begin{bmatrix} \dot{X} \\ 0 \end{bmatrix}, \tag{2.32}$$

where

$$J_{EX} = \begin{bmatrix} J(\theta) \\ \dfrac{\partial G}{\partial \theta} \end{bmatrix} \tag{2.33}$$

is an $m \times m$ square matrix called *the extended Jacobian matrix*. Provided that the extended Jacobian matrix is non-singular along the entire path, we may solve the inverse kinematics problem as

$$\dot{\theta}(t) = J_{EX}^{-1} \begin{bmatrix} \dot{X}(t) \\ 0 \end{bmatrix}. \tag{2.34}$$

Since equation (2.34) gives a unique solution for $\dot{\theta}$, conservative motion is obtained, i.e. a closed end-effector trajectory generates a closed joint space trajectory.

When extended Jacobian technique is implemented in real-time, it is very important to find an efficient method for computing N_e in equation (2.29). For this purpose, Chang [11] proposed following

$$N_e = \begin{bmatrix} J_n^{-1} \cdot J_{m-n} \\ -I_{m-n} \end{bmatrix}, \tag{2.35}$$

where J_n is an $n \times n$ matrix which consists of n linearly independent columns of J, J_{m-n} is an $n \times (m-n)$ matrix which consists of the remaining $m-n$ columns of J, and I_{m-n} is the $(m-n)$ order identity matrix. Without loss of generality we may assume that

$$J = \begin{bmatrix} J_n & | & J_{m-n} \end{bmatrix}. \tag{2.36}$$

It is easy to see that the N_e as defined in (2.35) belongs to the null-space of J because $J \cdot N_e = J_{m-n} - J_{m-n} = 0$. However, this approach may not be numerically reliable, especially if J is ill-conditioned (e.g. in the neighborhood of a singular configuration). A numerically more reliable approach would be a QR decomposition of J^T or a singular value decomposition, e.g. see [12].

Equation (2.31) provides a fixed transformation from work space to joint space, which directly solves the inverse kinematics problem for redundant manipulators. The accuracy achieved in the work space with this method is better than that with the pseudoinverse method. The fact that the approach generates conservative motions is another advantage.

2.2 GLOBAL OPTIMIZATION FOR REDUNDANCY RESOLUTION

In contrast to local optimization methods, global optimization methods compute a joint trajectory from a complete description of the task in the work space. Therefore, all the global optimality criteria are in the form of integrals. As mentioned previously, there are only a few papers on global optimal redundancy resolution to date in the robotics literatures. Some of them are aimed at optimizing particular performance criteria such as completion time or average kinetic energy [13] [14] [16], while others deal with general cases in which performance criteria can be any arbitrary objective functions. These methods usually produce joint trajectories with acceptable joint velocities, but most of them are limited to off-line planning, and do not allow for real-time correction based on sensor measurements. Two global optimal methods outlined in this section show the major features of the existing work on this topic.

2.2.1 GLOBALLY OPTIMAL REDUNDANCY RESOLUTION BASED ON LOCAL OPTIMIZATION

This method, proposed by Kazerounian & Wang [17], exposes an interesting relationship between joint velocities and joint accelerations as redundancy is resolved globally by minimizing the norm of the joint velocities, and locally by minimizing the norm of the joint accelerations.

By differentiating the Jacobian relation (2.2) with respect to time, we can get the acceleration relation

$$J\ddot{\theta} = \ddot{X} - \dot{J}\dot{\theta}. \tag{2.37}$$

Using the same method as described in Section 2.1.1, the optimal solution for joint accelerations $\ddot{\theta}$, which instantaneously minimizes the performance criterion $\frac{1}{2}\ddot{\theta}^T\ddot{\theta}$, can been found as

$$\ddot{\theta} = J^{\dagger}\,(\ddot{X} - \dot{J}\dot{\theta})\,. \tag{2.38}$$

The study on the global effects of the local optimal solution $\ddot{\theta}$ in equation (2.38) involves by examining the solution to the following global optimization problem:

$$\text{Minimize} \quad L = \int_{t_0}^{t_f} (\dot{\theta}^T \dot{\theta})\, dt \tag{2.39}$$

$$\text{Subject to} \quad f(\theta) - X = 0. \tag{2.40}$$

This optimization problem can be solved by applying the general theory of algebraic Constraints in calculus of variations equation [15].

Define the augmented objective function

$$L^* = \int_{t_0}^{t_f} g(\theta, \dot{\theta}, t)\, dt, \tag{2.41}$$

where

$$g(\theta, \dot{\theta}, t) = \dot{\theta}^T \dot{\theta} + \lambda (f(\theta) - X) \tag{2.42}$$

The λ in equation (2.42) is the Lagrangian multiplier vector, then the extremal of L^* is governed by

$$\frac{\partial g}{\partial \theta} - \frac{d}{dt}\left(\frac{\partial g}{\partial \dot{\theta}}\right) = 0 \tag{2.43}$$

and

$$-\frac{\partial g}{\partial \lambda} = X - f(\theta) = 0. \tag{2.44}$$

Equation (2.43) is the Euler-Lagrange equation corresponding to variable θ, and equation (2.44), which is the Euler -Lagrange equation corresponding to the variable λ, recover the constraint condition(2.39). In equation (2.43),

$$\frac{\partial g}{\partial \theta} = \left(\frac{\partial f}{\partial \theta}\right)^T \lambda^T = J^T \lambda^T \qquad (2.45)$$

and

$$\frac{d}{dt}\left(\frac{\partial g}{\partial \dot{\theta}}\right) = 2\ddot{\theta}. \qquad (2.46)$$

Substituting equation (2.45) and (2.46) into (2.43) yields

$$J^T \lambda^T - 2\ddot{\theta} = 0. \qquad (2.47)$$

Therefore

$$\ddot{\theta} = 0.5 J^T \lambda^T. \qquad (2.48)$$

Equation (2.48) and (2.37) together construct a system of $m+n$ differential equations in $m+n$ unknown variables ($\ddot{\theta}$ and λ). Substituting (2.48) into (2.37) gives

$$0.5 J J^T \lambda^T = \ddot{X} - \dot{J}\dot{\theta}. \qquad (2.49)$$

For a full rank J, λ^T can be solved as

$$\lambda^T = 2\,(J J^T)^{-1}\,(\ddot{X} - \dot{J}\dot{\theta}). \qquad (2.50)$$

then we can obtain the optimal solution $\ddot{\theta}$ by substituting (2.50) into (2.48) as

$$\ddot{\theta} = J^T (JJ^T)^{-1} (\ddot{X} - \dot{J}\dot{\theta}) = J^{\dagger} (\ddot{X} - \dot{J}\dot{\theta}) , \tag{2.51}$$

which is the same as equation (2.38). Therefore, *the local minimization of the joint accelerations will result in the global minimization of the joint velocities.*

In order to uniquely specify the optimal solution $\ddot{\theta}$, we must consider the boundary conditions for θ and $\dot{\theta}$ in addition to the necessary condition given by equation (2.51). In general, the only constraints imposed on the initial or final joint configurations and velocities are that at times t_0 and t_f they should satisfy the kinematic relation and Jacobian relation. That is

$$f(\theta(t)) = X(t) \qquad t = t_0, t_f \tag{2.52}$$

and

$$\dot{X}(t) = J\dot{\theta}(t) \qquad t = t_0, t_f \tag{2.53}$$

In such cases, which are referred to as "natural" boundary conditions, we seek the least-squares solution over all joint velocities which satisfy equation (2.53), i.e.

$$\dot{\theta}(t) = J^{\dagger} \dot{X}(t) \qquad t = t_0, t_f \tag{2.54}$$

The necessary condition equation (2.51) and the boundary conditions (2.52) and (2.54) together define a system of second-order equations. The solution to these equations gives the minimum value of the objective function in (2.39) for a specified trajectory.

For a given trajectory in the work space, the boundary conditions $\dot{\theta}(t_0)$ and $\dot{\theta}(t_f)$

are specified by equation (2.54). Here, the problem is to find the initial joint values such that the joint velocities reach specified $\dot{\theta}(t_f)$ at time t_f. To solve this problem, the initial value adjusting method [24] is used, which can be briefly described as follows. First, the initial value $\theta(t_0)$ which is governed by $X(t_0) = f(\theta(t_0))$ is estimated. Second, the differential equation (2.51) is solved based on the estimated initial values. Then, depending on the difference between the specified $\dot{\theta}(t_f)$ and its computed value, the estimated initial value is modified. These processes are repeated until the boundary values converge to the given values at both endpoints t_0 and t_f. Obviously, these processes require extensive computation and can only be carried out in off-line control.

If the joint values and velocities are specified at time t_0, equation (2.51) gives the evolution of the optimal solution satisfying the initial conditions. However, the boundary condition of joint velocities at time t_f may not satisfied, the value of the objective function of equation (2.39) will in general be larger than the one in the case of "natural" boundary conditions.

2.2.2 GLOBAL OPTIMIZATION FOR GENERAL OBJECTIVE FUNCTIONS

In this section, the optimal problem of redundancy resolution is formulated as:

$$\text{Minimize} \quad V = \int_{t_0}^{t_f} [p(\theta, t) + \dot{\theta}^T \dot{\theta}]\, dt \tag{2.55}$$

$$\text{subject to} \qquad X = f(\theta)\,. \tag{2.56}$$

where $p(\theta, t)$ is an objective function defined such that V represents a performance index of the integral type which evaluates the performance of redundancy utilization. By differentiating (2.56), we obtain the Jacobian relation

$$\dot{X} = J(\theta)\,\dot{\theta} \tag{2.57}$$

Suppose that a $\dot{\theta}$ satisfying equation (2.57) exists; then, from Theorem 2.1 (see section 2.1.2), equation (2.57) can be replaced by

$$\dot{\theta} = J^{\dagger}\dot{X} + (I - J^{\dagger}J)\,Z. \tag{2.58}$$

where $J^{\dagger} \in \Re^{m \times n}$ is the pseudoinverse of J and $Z \in \Re^{m}$ is an arbitrary vector. The second term on the right-hand side represents redundancy. By replacing (2.56) with (2.58), the optimal control problem of redundancy resolution is represented by the problem

$$\text{Minimize} \qquad V = \int_{t_0}^{t_f} [\,p(\theta, t) + \dot{\theta}^T\dot{\theta}\,]\,dt \tag{2.59}$$

$$\text{subject to} \qquad \dot{\theta} = J^{\dagger}\dot{X} + (I - J^{\dagger}J)\,Z \equiv g(\theta, Z, t)\,. \tag{2.60}$$

Although equation (2.60) represents the kinematic relations between θ and X, it can be regarded as a system equation of a time-varying, non-linear, dynamical system with θ as the state vector and Z as the input vector. Now, regarding equation (2.59) and (2.60) as an optimal control problem for a dynamical system, we apply Pontryagin's maximum principle [18] to the problem.

The Hamiltonian for this problem with a fixed initial value $\theta_0 = \theta(t_0)$ and free final value $\theta(t_f)$ is as follows:

$$H(\lambda, \theta, t, Z) = -p(\theta, t) - g^T g + \lambda^T g. \tag{2.61}$$

where $\lambda \in \Re^m$ is an adjoint vector. Equation (2.61) can be rewritten as

$$H(\lambda, \theta, t, Z) = -(g - 0.5\lambda)^T (g - 0.5\lambda) + 0.25\lambda^T\lambda - p. \tag{2.62}$$

If we choose a $Z(t)$ such that the Hamiltonian is maximized at every moment t, the optimal joint trajectory $\theta(t)$ is obtained by solving the following differential equations:

$$\dot{\theta} = \left[\frac{\partial H}{\partial \lambda}\right]^T \tag{2.63}$$

$$\dot{\lambda} = -\left[\frac{\partial H}{\partial \theta}\right]^T. \tag{2.64}$$

Such a $Z(t)$ can be obtained from equation (2.60) and (2.62). In equation (2.62), because $Z(t)$ is included only in $g(\theta, Z, t)$, the $Z(t)$ that minimizes $\|g - 0.5\lambda\|$ will maximizes H. In other word, such a $Z(t)$ should satisfy $g(\theta, Z, t) = 0.5\lambda$. Thus equation (2.60) yields

$$Z = (I - J^{\dagger}J)^{\dagger} (-J^{\dagger}\dot{X} + 0.5\lambda). \tag{2.65}$$

Using the following property of pseudoinverses:

$$(I - P^{\dagger} P)^{\dagger} = (I - P^{\dagger} P) \tag{2.66}$$

and

$$(I - P^{\dagger} P) P^{\dagger} = 0, \tag{2.67}$$

equation (2.65) yields

$$Z = 0.5\,(I - J^{\dagger} J)\,\lambda. \tag{2.68}$$

From equation (2.61), (2.63), (2.64) and (2.68), it is found that the optimal trajectory $\theta(t)$ is governed by the following differential equations:

$$\dot{\theta} = J^{\dagger} \dot{X} + 0.5\,(I - J^{\dagger} J)\,\lambda \equiv g \tag{2.69}$$

and

$$\dot{\lambda} = \left(\frac{\partial g}{\partial \theta}\right)^{T} (2g - \lambda) + \left(\frac{\partial p}{\partial \theta}\right)^{T}. \tag{2.70}$$

The boundary conditions which are necessary for solving equation (2.69) and (2.70) are the given initial joint configuration

$$\theta(t_0) = \theta_0 \tag{2.71}$$

and the boundary condition derived from the condition of a free $\theta(t_f)$ [18]

$$\lambda(t_f) = 0. \tag{2.72}$$

This is a two-point boundary value problem and a detailed numerical example can be found in [19].

2.3 SOME UNSOLVED PROBLEMS

In this section, three major difficulties in solving the optimal redundancy resolution problem are discussed.

2.3.1 ALGORITHMIC SINGULARITIES

In the case of local optimization for redundancy resolution, a suitable objective function $H(\theta)$ can be chosen such that optimizing $H(\theta)$ yields the desired performance in addition to achieving the desired end-effector trajectory. When $H(\theta)$ is chosen, the optimal joint velocity can be calculated using equation (2.27), i.e.,

$$\dot{\theta} = J^{\dagger}\dot{X} + (I - J^{\dagger}J)\nabla_{\theta}H(\theta).$$

However, in some joint configurations, the gradient $\nabla_{\theta}H(\theta)$ may be very large, so that an unacceptably large joint velocity will be induced. Such a joint configuration is said to result from an *algorithmic singularity*. A similar problem also exists in the extended Jacobian technique. Algorithmic singularities may be introduced in the extended Jacobian matrix J_{EX} due to the submatrix $\frac{\partial G}{\partial \theta}$ (see equation (2.33)). These algorithmic singularities occur when either $\frac{\partial G}{\partial \theta}$ is rank-deficient, or some rows of J and $\frac{\partial G}{\partial \theta}$ are linearly dependent. By a judicious choice of the objective function $H(\theta)$, some algorithmic singularities may be avoided [20] [21], but a systematic approach is not yet available for totally avoiding algorithmic singularities.

2.3.2 REAL-TIME GLOBAL OPTIMIZATION

To date all reported globally optimal redundancy resolution can only be carried out off-line. Two major difficulties arise when one tries to implement these algorithms in real-time. First, most global optimization algorithms give the optimal solution at joint acceleration level, e.g., the algorithm in Section 2.2.1. Therefore boundary conditions have to be imposed on both the joint configuration θ and the joint velocity $\dot{\theta}$ at time t_0 and t_f. Usually, the initial value adjusting method is used to solve this problem. This requires estimation and adjustment many times to find an initial joint value $\theta(t_0)$ such that the joint velocities reach a specified boundary condition $\dot{\theta}(t_f)$. Obviously, such an algorithm cannot in general be carried out in real-time. Second, several global optimization algorithms are based on Pontryagin's minimum principle. All these algorithms lead to the two-point boundary value problem as shown in Section 2.2.2. There are many methods for solving the two-point boundary value problems, but none of them yields an algorithm which can be implemented in real-time.

2.3.3 CONSERVATIVE MOTION

If a closed trajectory of an end-effector in its work space yields a closed trajectory in the manipulator's joint space, we call the motion *conservative*. In practice, many manipulator tasks are cyclic closed-trajectory motions in which conservative motion is desired. However, kinematic control under a local optimization algorithm cannot guarantee conservative motion. This problem was further discussed in [22] [23]. It is shown that if the relation between the work space $X \in \Re^n$ and the joint space $\theta \in \Re^m$ is defined as $X = f(\theta)$, where $f(\cdot): \Re^m \rightarrow \Re^n$ is a continuous function, then there are some mechanisms (for example, a two-d.o.f. pointing mechanism and a three-d.o.f. orienting wrist) where no continuous closed-form inverse kinematic function exists over the whole work space. In the global optimization approach, no closed-form inverse kinematic functions are necessary. However, for some desired trajectories in the work space, even the global optimization algorithm may lead to nonconservative motions. This problem has yet to be investigated thoroughly.

2.4 REFERENCES

[1] D. E. Whitney, "Resolved Motion Rate Control of Manipulators and Human Prostheses", *IEEE Trans. on Man-Machine Systems*, Vol. MMS-10, No. 2, 1969.

[2] D. N. Nenchev, "Redundancy Resolution through Local Optimization: A Review", *J. Rob. Syst.*, Vol. 6, pp. 769 - 798, 1989.

[3] S. Gupta & J. Y. S. Luh, "Closed-Loop Control of Manipulators with Redundant Joints Using the Hamilton-Jacobi-Bellman Equation", *Proc. IEEE Int. Conf. on Robotics and Automation*, Sacramento, Califonia, 1991.

[4] C. A. Klein & C-H. Huang, "Review of Pseudoinverse Control for Use with Kinematically redundant manipulators", *IEEE Trans. on Sys., Man, Cybern.*, vol. SMC-13, pp.245-250, 1983.

[5] J. Angeles & S. Mathur, "Resolved-Rate Control of Redundant Manipulators with Elimination of Non-Conservative Effects", *Proc. 5th Intl. Symposium of Robotics Research*, Tokyo, Japan, 1989, pp. 33-40.

[6] J. M. Hollerbach & K. C. Suh, "Redundancy Resolution of Manipulators through Torque Optimization", *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 1016-1021, 1985.

[7] A. Liegeois, "Automatic Supervisory Control of the Configuration and Behavior of Mutibody Mechanisms", *IEEE Trans. on Sys., Man, and Cybern*, Vol. SMC-7, pp. 868-871, 1977.

[8] T. Yoshikawa, "Manipulability of Robotic Mechanisms", *Robotics Research -The 2nd Intl. Symposium*, eds. H. Hanafusa and H. Inoue, Cambridge, Mass.: MIT Press,

pp. 439-446, 1985.

[9] O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots", *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 500-505, 1985.

[10] J. Baillieul, "Kinematic Programming Alternatives for Redundant Manipulators", *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 722-728, 1985.

[11] P. H. Chang, "A Closed-Form Solution for Inverse Kinematics of Robot Manipulators with Redundancy", *IEEE J. Rob. & Automat.*, Vol. RA-3, No. 5, October, 1987.

[12] G.H.Golub and C.F.Van Loan, "Matrix Computations", Johns Hopkins University Press, Baltimore, MD,1983.

[13] Y. Nakamura & H. Hanafusa, "Task Priority Based Redundancy Control of Robot Manipulators", *Robotics Research* --- The *2nd Intl. Symposium*, eds. H. Hanafusa and H. Inoue, Cambrige, Mass.: MIT Press, pp. 155-162, 1985.

[14] D. W. Johnson & E. G. Gillert, "Minimum Time Path Planning in the Presence of Obstacles", *Proc. 24th IEEE Conf. on Decision and Control*. Vol. 3, pp. 1748-1753, 1985.

[15] J.W.Cdaggs, *Calculus of Variations*, George Allen & Unwin Ltd, 1973.

[16] J. E. Bobrow et al, "On the Optimal Control of Robotic Manipulators with Actuator Constraints", *Proc. Automatic Control Conf.*, Vol. 3, pp. 782-787, 1983.

[17] K. Kazerounian & Z. Wang, "Global versus Local Optimization in Redundancy Resolution of Robotic Manipulators", *Int. J. Rob. Res.*, Vol, 7, No. 5, October 1988.

[18] L. S. Pontryagin et al, The *Mathematical Theory of Optimal Processes*, New York: Wiley. (trans. from Russian by K. N. Trirogoff; ed. L. W. Neustadt), 1962.

[19] Y. Nakamura & H. Hanafusa, "Optimal Redundancy Control of Robot Manipulators", *Int. J. Rob. Res.*, Vol. 6, No.1, pp. 32-42, 1987.

[20] H. Seraji, "Configuration Control of Redundant Manipulators: Theory and Implementation", *IEEE Trans.On Robotics and Automation*, Vol. 5, No. 4, pp. 472-

490, 1989.

[21] J. Burdick & H. Seraji, "Characterization and Control of Self-Motions in Redundant Manipulators", *Proc. NASA Conf. on Space Telerobotics* Pasadena, CA, 1989.

[22] D.R. Baker & C.W. Wampler, "Some Facts Concerning the Inverse Kinematics of Redundant Manipulators", *Proc. IEEE Int. Conf. on Robotics and Automation*, Raleigh, North Carolina, 1987, pp 604-609.

[23] C.W. Wampler, "Inverse Kinematic Functions for Redundant Manipulators", *Proc. IEEE Int. Conf. on Robotics and Automation*, Raleigh, North Carolina, 1987, pp 610-617.

[24] T. Ojika & Y. Kasue, "Initial-Value Adjusting Method for The Solution of Nonlinear Multipoint Boundary-Value Problems", *J. Math. Anal. Appl.*, 69:359-371, 1979.

CHAPTER 3 GLOBALLY OPTIMAL KINEMATIC CONTROL (GOKC) OF REDUNDANT MANIPULATROS

In this chapter, a new global optimization algorithm for redundancy resolution, namely *globally optimal kinematic control* (GOKC), is proposed. The algorithm gives the optimal solution at the joint velocity level rather than at the joint acceleration level. The proposed approach provides an automatic means of balancing the trade-off of the task error and the joint velocities, thereby yielding a singularity-robust implementation of optimal kinematic control. Another advantage of the GOKC is that, for a class of trajectories in the work space of a redundant manipulator, GOKC can be carried out in real-time and yields conservative motions. The stability of the GOKC is guaranteed, and the tracking error in the work space can be made arbitrarily small.

3.1 GLOBALLY OPTIMAL KINEMATIC CONTROLLER DESIGN

3.1.1 THE FORMULATION OF THE GOKC PROBLEM

The Jacobian relation of robot manipulators may be described as a *linear* time-varying system. This property makes it easier to use the Jacobian relation as a constraint in optimization-based redundancy resolution, rather than to use the *nonlinear* kinematic relation. The system can be written in the state-space form

$$\begin{aligned} \dot{X} &= 0X + J\dot{\theta} \\ Y &= X \end{aligned} \tag{3.1}$$

where X is the $n \times 1$ state vector of Cartesian positions, $\dot{\theta}$ is the $m \times 1$ input vector of joint velocities, and Y is the output of the system. Obviously, this system is controllable and observable if the Jacobian matrix is of full-rank.

In order to control the system (3.1) such that its state X tracks a desired trajectory X_d while using minimal control energy over the entire path, we choose the following integral performance criterion with quadratic cost terms

$$V = \int_{t_0}^{t_f} [\dot{\theta}^T R \dot{\theta} + (X - X_d)^T Q (X - X_d)]\, dt \qquad (3.2)$$

where Q is an $n \times n$ nonnegative definite matrix, and R is an $m \times m$ positive definite matrix. Q and R can be either constant or time-varying matrices. The quadratic nature of the cost term ensures that the optimal control law will be linear while the constraints of the matrices Q and R guarantee that the control law leads to a stable control.

Now, for redundant manipulators, the problem of achieving a desired trajectory with minimal weighted kinetic energy can be expressed as the following constrained optimal control problem:

$$\text{Minimize} \quad V = \int_{t_0}^{t_f} [\dot{\theta}^T R \dot{\theta} + (X - X_d)^T Q (X - X_d)]\, dt \qquad (3.3)$$

$$\text{Subject to} \qquad -\dot{X} + 0X + J\dot{\theta} = 0\ .$$

We can chose R to be a diagonal matrix such that its elements reflect the relative cost, in terms of energy, of using each joint. R can also be chosen to match with certain performance criterion. For example, minimization of the true kinetic energy requires that one minimizes the cost term $\dot{\theta}^T M \dot{\theta}$, where M is the inertia matrix of the manipulator, so that we set $R = M$. A similar argument can be applied to the matrix Q. It may be that the orientation error of the end-effector is not critical but high accuracy in position is required in a welding task. In such a case, we can choose a diagonal Q whose elements corresponding to the position variables are much larger than those corresponding to the orientation variables.

We can also adopt the extended Jacobian technique to meet the need for different redundancy resolution goals. When the Jacobian matrix J is extended to an $m \times m$ square matrix, Q should be an $m \times m$ matrix too. Usually, .he objective (potential) function has a value with different units from that of the Cartesian positions, and so the elements of Q corresponding to the objective function should be adjusted appropriately.

Since the constraint in (3.3) is based on the manipulator's kinematic information, and the solution for this problem is globally optimal, we refer to the problem as *globally optimal kinematic control* (GOKC). Note that the optimal solution for this problem is at the joint velocity level. The boundary conditions are therefore only imposed on joint values.

3.1.2 BRIEF REVIEW OF LINEAR OPTIMAL CONTROL

In this section, we give a brief outline of the basic theory of the linear optimal control.

In linear optimal control, the plant that is controlled is assumed to be linear, and the controller is determined such that a quadratic performance index is minimized. The index is quadratic in the control and state/error variables as shown in equation (3.3). Methods that achieve linear optimal control are termed Linear-Quadratic (LQ) methods. Such methods have been used successfully for two types of control problems: *the optimal regulator problem* and *the optimal tracking problem.*

Consider the system

$$\dot{x}(t) = F(t)x(t) + G(t)u(t) \tag{3.4}$$
$$y(t) = H(t)x(t)$$

where $F(t)$, $G(t)$ and $H(t)$ are matrix functions of time with continuous entries. The optimal regulator problem is to minimize the following quadratic performance index

$$V = x^T(t_f)Ax(t_f) + \int_{t_0}^{t_f} (u^T R(t)u + x^T Q(t)x)\,dt \tag{3.5}$$

with equation (3.4) representing a dynamic constraint. In the optimal tracking problem, we seek to minimize the quadratic performance index

$$V = \int_{t_0}^{t_f} [u^T R(t)u + (y_d - y)^T Q(t)(y_d - y)]\,dt \tag{3.6}$$

subject to (3.4), where y_d is the desired system output. If $Q(t)$ and $R(t)$ have continuous entries, are symmetric, and nonnegative and positive definite, respectively, and if A is a nonnegative definite symmetric matrix, we can obtain linear control laws as solutions to both the regulator and the tracking problems.

To solve these two problems, the use of the Minimum Principle of Pontryagin [1]- [4] seems particularly well suited. Consider the optimal regulator problem (3.5). Define the Hamiltonian using the costate vector λ as

$$H(x, u, t, \lambda) = (u^T R(t)u + x^T Q(t)x) + \lambda^T [F(t)x(t) + G(t)u(t)] \tag{3.7}$$

and its minimum as

$$H^* (x, t, \lambda) = \min_{u(x, t, \lambda)} H(x, u, t, \lambda) \tag{3.8}$$

If the minimum exists, and at the minimum, $\frac{\partial H}{\partial u} = 0$, then the equations

$$\dot{x} = \frac{\partial}{\partial \lambda} H^*, \qquad x(0) \text{ given} \tag{3.9}$$

$$\dot{\lambda} = -\frac{\partial}{\partial x} H^*, \qquad \lambda(t_f) = \frac{\partial}{\partial x} [x^T (t_f) A x(t_f)] \Big|_{x(t_f)} \tag{3.10}$$

are satisfied along the optimal trajectory. Note that equations (3.9) and (3.10) are coupled ordinary differential equations with two-point boundary conditions. This problem is called a *two-point boundary-values problem*. An approach [1] for solving this problem leads to the following optimal solution of the problem:

$$u^* (t) = -R^{-1} G^T P x \tag{3.11}$$

where P is a symmetric positive-definite matrix, and is the solution of the following deferential Riccati equation:

$$-\dot{P} = PF + F^T P - PGR^{-1} G^T P + Q, \qquad P(t_f) = A \tag{3.12}$$

A similar approach can be used for solving the optimal tracking problem.

3.1.3 DERIVATION OF THE OPTIMAL CONTROL LAW

Now, we go back to the problem (3.3), which is an optimal tracking problem, and apply the above approach. We define the Hamiltonian with costate vector λ as

$$H = \frac{1}{2}E^T Q E + \frac{1}{2}\dot{\theta}^T R \dot{\theta} + \dot{\theta}^T J^T \lambda \tag{3.13}$$

where $E = X - X_d$ denotes the tracking error. According to Pontryagin's Minimum Principle, we have

$$\frac{\partial H}{\partial \dot{\theta}} = R\dot{\theta} + J^T \lambda = 0 \tag{3.14}$$

$$\dot{\lambda}(t) = -\frac{\partial H}{\partial E} = -QX(t) + QX_d(t) \tag{3.15}$$

$$\lambda(t_f) = 0 \tag{3.16}$$

Equation (3.14) is the necessary condition for minimum H, that is

$$\dot{\theta} = -R^{-1} J^T \lambda \tag{3.17}$$

Substituting (3.17) into the constraint of the problem (3.3) gives

$$\dot{X} = -JR^{-1} J^T \lambda \tag{3.18}$$

Then, adjoining (3.18) and (3.15), we get a linear time-varying system with $2n$ differential equations as

$$\begin{bmatrix}\dot{X}\\ \dot{\lambda}\end{bmatrix} = \begin{bmatrix}0 & -JR^{-1}J^T\\ -Q & 0\end{bmatrix}\begin{bmatrix}X\\ \lambda\end{bmatrix} + \begin{bmatrix}0\\ QX_d\end{bmatrix}. \tag{3.19}$$

Among the $2n$ boundary conditions, which are needed for solving (3.19), there are n initial states $X(t_0)$ and n final costates $\lambda(t_f)$. The initial states $X(t_0)$ are known, and the final costates $\lambda(t_f)$ are given by (3.16). The solution to (3.19) can be expressed as

$$\begin{bmatrix}X(t)\\ \lambda(t)\end{bmatrix} = \Phi(t,t_0)\left\{\begin{bmatrix}X(t_0)\\ \lambda(t_0)\end{bmatrix} + \int_{t_0}^{t}\Phi^{-1}(\tau,t_0)\begin{bmatrix}0\\ QX_d(\tau)\end{bmatrix}d\tau\right\} \tag{3.20}$$

where $\Phi(t, t_0)$ is the $2n \times 2n$ state transition matrix of the system (3.19). Therefore, at the terminal time $t = t_f$ we have

$$\begin{bmatrix}X(t_f)\\ \lambda(t_f)\end{bmatrix} = \Phi(t_f,t)\left\{\begin{bmatrix}X(t)\\ \lambda(t)\end{bmatrix} + \int_{t}^{t_f}\Phi^{-1}(\tau,t)\begin{bmatrix}0\\ QX_d(\tau)\end{bmatrix}d\tau\right\}. \tag{3.21}$$

If we divide the state transition matrix Φ into four $n \times n$ sub-matrices as

$$\Phi = \begin{bmatrix}\phi_{11} & \phi_{12}\\ \phi_{21} & \phi_{22}\end{bmatrix}, \tag{3.22}$$

then equation (3.21) can be rewritten as

$$X(t_f) = \phi_{11}(t_f, t)X(t) + \phi_{12}(t_f, t)\lambda(t) + \Omega_1(t_f, t) \quad (3.23)$$

and

$$\lambda(t_f) = \phi_{21}(t_f, t)X(t) + \phi_{22}(t_f, t)\lambda(t) + \Omega_2(t_f, t) \quad (3.24)$$

where

$$\Omega_1(t_f, t) = \left[\phi_{11}(t_f, t)\ \phi_{12}(t_f, t)\right]\int_t^{t_f}\Phi^{-1}(\tau, t)\begin{bmatrix}0\\ QX_d(\tau)\end{bmatrix}d\tau , \quad (3.25)$$

and

$$\Omega_2(t_f, t) = \left[\phi_{21}(t_f, t)\ \phi_{22}(t_f, t)\right]\int_t^{t_f}\Phi^{-1}(\tau, t)\begin{bmatrix}0\\ QX_d(\tau)\end{bmatrix}d\tau . \quad (3.26)$$

From (3.23), (3.24) and (3.16), the costate λ can be solve as

$$\lambda(t) = P(t)X(t) + b(t) \quad (3.27)$$

where $P(t)$ is an $n \times n$ matrix defined as

$$P(t) = [\phi_{22}(t_f, t) - \phi_{12}(t_f, t)]^{-1}[\phi_{11}(t_f, t) - \phi_{21}(t_f, t)] \quad (3.28)$$

and $b(t)$ is the $n \times 1$ column vector

$$b(t) = X(t_f) + \Omega_1(t_f, t) - \Omega_2(t_f, t) . \quad (3.29)$$

Equation (3.27) shows the linear relation between the costate and the state, and that $P(t)$

and $b(t)$ depend on the end time t_f but are independent of the initial state $X(t_0)$. One may note that solving for $P(t)$ and $b(t)$ requires calculating inverse matrices of order $n \times n$ and $2n \times 2n$, i.e., it is computationally expensive. Therefore, we try to find some properties of $P(t)$ and $b(t)$ by which calculation of the inverse matrices can be avoided. To do so, we differentiate both sides of (3.27) to get

$$\dot{\lambda}(t) = \dot{P}(t) X(t) + P(t) \dot{X}(t) + \dot{b}(t) . \tag{3.30}$$

By substituting (3.27) into (3.18), we have

$$\dot{X}(t) = -JR^{-1}J^T P(t) X(t) - JR^{-1}J^T b(t) \tag{3.31}$$

Then, substituting (3.31) back into (3.30) yields

$$\dot{\lambda} = (\dot{P} - PJR^{-1}J^T P) X - PJR^{-1}J^T b + \dot{b} . \tag{3.32}$$

Next, we recall equation (3.15):

$$\dot{\lambda} = -QX + QX_d . \tag{3.33}$$

Equation (3.32) and (3.33) will hold for all $X(t)$, $t \in [t_0, t_f]$, if the following equations are satisfied

$$\dot{P} = PJR^{-1}J^T P - Q \tag{3.34}$$

and

$$\dot{b} = PJR^{-1}J^T b + QX_d . \tag{3.35}$$

From (3.16) and (3.27), at the end time t_f we have

$$\lambda(t_f) = P(t_f)X(t_f) - b(t_f) = 0 . \tag{3.36}$$

Since (3.36) should be satisfied for any $X(t_f)$, we can conclude that the boundary conditions for (3.34) and (3.35) are

$$P(t_f) = 0 \text{ , and } b(t_f) = 0 . \tag{3.37}$$

Now, from (3.17), (3.27), (3.34), (3.35) and (3.37), the optimal control law for optimal tracking problem is given by

$$\dot{\theta}(t) = -R^{-1}(t)J^T(\theta(t))\,[P(t)X(t) + b(t)] \tag{3.38}$$

where $P(t)$ is an $n \times n$ symmetric positive definite matrix given by the solution of the following differential Riccati equation

$$\dot{P}(t) = P(t)J(\theta(t))R^{-1}(t)J^T(\theta(t))P(t) - Q(t) \tag{3.39}$$

with boundary condition $P(t_f) = 0$, and $b(t)$ is an $n \times 1$ column vector which can be obtained by solving the following differential equation

$$\dot{b}(t) = P(t)J(\theta(t))R^{-1}(t)J^T(\theta(t))b(t) + Q(t)X_d(t) \tag{3.40}$$

with the boundary condition $b(t_f) = 0$.

Since the boundary conditions are given only at the terminal time t_f, equation (3.39) and (3.40) have to be integrated backward in time. Obviously, it is difficult to implement this in real-time. Besides, $J(\theta(t_f))$ is needed for solving (3.39) and (3.40), and $\theta(t_f)$ should satisfy the kinematic constraint

$$X_d(t_f) = f(\theta(t_f)). \tag{3.41}$$

Usually, it is difficult to solve for $\theta(t_f)$ from (3.41). These two difficulties can be circumvented, and the details will be discussed in Section 3.2.

Fig. 3.1 shows the closed-loop kinematic control system using the GOKC control law of equation (3.38). The signal $b(t)$ can be regarded as the reference input which is driven by the desired trajectory $X_d(t)$.

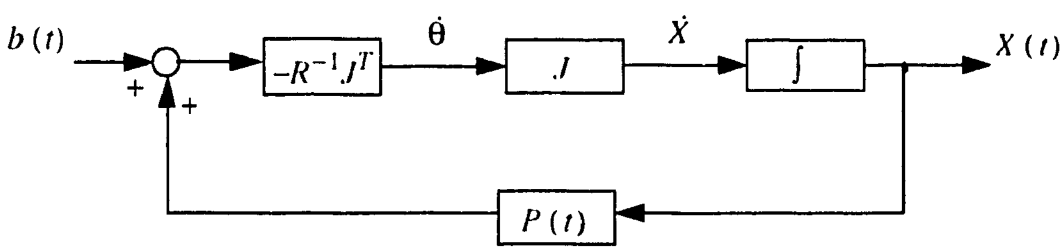

Fig. 3.1 Block Diagram of the GOKC Control System

3.1.4 EXISTENCE AND UNIQUENESS OF THE OPTIMAL SOLUTION

To show that the GOKC is a global optimization scheme, we should not only show that the optimal solution exists, but also prove the uniqueness of the solution.

Lemma 3.1 The optimal control given by equation (3.38) produces a minimum of V in equation (3.3).

Proof As has been shown in Section 3.1.2, the optimal control law satisfies the necessary condition for minimizing V in equation (3.3). The sufficient condition is that the Hessian matrix of the Hamiltonian function in (3.13) is non-negative definite. This matrix $\aleph$ is defined as

$$\aleph = \begin{bmatrix} \frac{\partial^2 H}{\partial E^2} & \frac{\partial^2 H}{\partial E \partial \dot{\theta}} \\ \frac{\partial^2 H}{\partial \dot{\theta} \partial E} & \frac{\partial^2 H}{\partial \dot{\theta}^2} \end{bmatrix} . \tag{3.42}$$

From equation (3.13), we have

$$\frac{\partial^2 H}{\partial E^2} = Q , \quad \frac{\partial^2 H}{\partial \dot{\theta}^2} = R , \quad \frac{\partial^2 H}{\partial E \partial \dot{\theta}} = \frac{\partial^2 H}{\partial \dot{\theta} \partial E} = 0 , \tag{3.43}$$

so that the Hessian matrix is $\begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix}$. Since Q and R are chosen to be non-negative and positive definite respectively, this matrix is non-negative, and the sufficient condition is satisfied.

Lemma 3.2 If an optimal solution exists to problem in (3.3), then this solution is unique and given by equation (3.38).

Proof Consider equation (3.31):

$$\dot{X}(t) = -JR^{-1}J^T P(t) X(t) - JR^{-1}J^T b(t)$$

The $n \times n$ matrix $P(t)$ is uniquely determined by (3.39), and the $n \times 1$ column vector $b(t)$ is uniquely determined by (3.40). Therefore (3.31) has a unique solution $X(t)$. The optimal control law is given by

$$\dot{\theta}(t) = -R^{-1}(t)J^T(\theta(t))\,[P(t)X(t) + b(t)]\,, \qquad (3.44)$$

since $P(t)$, $b(t)$ and $X(t)$ are unique, the optimal solution $\dot{\theta}(t)$ must be unique.

3.2 REAL-TIME GOKC

For high tracking accuracy and flexibility of operation, it is necessary to have real-time kinematic control. To meet this requirement, the following method is proposed.

When the GOKC is implemented on a digital computer, a discrete optimal control law is needed. The control signal $\dot{\theta}(t)$ can be computed at each time step. That is, the matrices R^{-1}, J, P, and the vector b are computed at each step. Thus, these parameters are assumed to remain constant during each sampling interval. The discrete optimal control law can then be expressed as

$$\dot{\theta}(t_f - k\Delta t) = -R^{-1}J^T(\theta(t_f - k\Delta t))\,[P(t_f - k\Delta t)X(t_f - k\Delta t) + b(t_f - k\Delta t)] \quad (3.45)$$

where Δt is the sampling period, and $k = 0, 1, 2, \ldots, \frac{(t_f - t_0)}{\Delta t}$. The joint configuration $\theta(t)$, matrix $P(t)$, and vector $b(t)$, which are needed in equation (3.45), can be calculated by using first-order estimation based on first derivative information, i.e.,

$$\theta(t_f-(k+1)\Delta t) = \theta(t_f-k\Delta t) - \Delta t\dot{\theta}(t_f-k\Delta t), \tag{3.46}$$

$$P(t_f-(k+1)\Delta t) = P(t_f-k\Delta t) - \Delta t\dot{P}(t_f-k\Delta t), \tag{3.47}$$

and

$$b(t_f-(k+1)\Delta t) = b(t_f-k\Delta t) - \Delta t\dot{b}(t_f-k\Delta t), \tag{3.48}$$

where

$$\dot{P}(t_f-k\Delta t) = P(t_f-k\Delta t)J(\theta(t_f-k\Delta t))R^{-1} J^T(\theta(t_f-k\Delta t))P(t_f-k\Delta t) - Q \tag{3.49}$$

$$\dot{b}(t_f-k\Delta t) = P(t_f-k\Delta t)J(\theta(t_f-k\Delta t))R^{-1} J^T(\theta(t_f-k\Delta t))b(t_f-k\Delta t) - QX_d(t_f-k\Delta t). \tag{3.50}$$

Now, we describe a method to implement real-time GOKC based on the discrete control law (3.45). Assuming that the desired end-effector trajectory is symmetric about $t = \frac{t_f+t_0}{2}$ in both position and orientation with continuous first derivative, as shown in Figure 3.2 for one element of $X_d(t)$. For this kind of trajectories, we have $\theta(t_f) = \theta(t_0)$ and the following result holds.

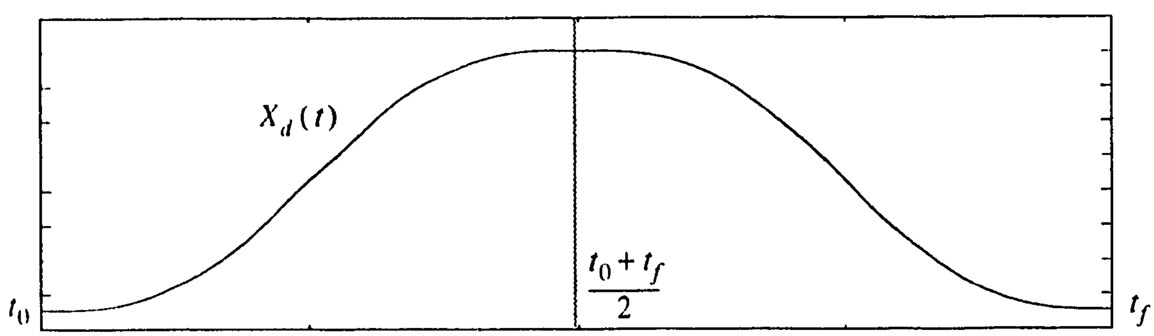

Fig. 3.2 Symmetric End-Effector Trajectory

Lemma 3.3

When the end-effector traces a trajectory that has continuous first derivative and is symmetric about $t = \frac{t_f + t_0}{2}$, using the GOKC, the control sequence $\dot{\theta}(t_f - k\Delta t)$, $k = 0, 1, 2, \ldots, (t_f - t_0)/(\Delta t)$, has symmetric properties and satisfies the following equations:

$$\begin{aligned}
\dot{\theta}(t_0) &= -\dot{\theta}(t_f) \\
\dot{\theta}(t_0 + \Delta t) &= -\dot{\theta}(t_f - \Delta t) \\
\dot{\theta}(t_0 + 2\Delta t) &= -\dot{\theta}(t_f - 2\Delta t) \\
&\vdots \\
\dot{\theta}\left(\frac{t_f + t_0}{2} - \Delta t\right) &= -\dot{\theta}\left(\frac{t_f + t_0}{2} + \Delta t\right)
\end{aligned} \tag{3.51}$$

Proof Consider the symmetric end-effector trajectory shown in Figure 3.2, the trajectory from t_0 to $t = \frac{t_f + t_0}{2}$ is the same as that from t_f to $t = \frac{t_f + t_0}{2}$ proceeding in the opposite direction. So that if there exists an optimal solution for the control sequence

$$\dot{\theta}(t_0),\ \dot{\theta}(t_0 + \Delta t),\ \dot{\theta}(t_0 + 2\Delta t),\ \ldots,\ \dot{\theta}(t_f - 2\Delta t),\ \dot{\theta}(t_f - \Delta t),\ \dot{\theta}(t_f),$$

then the sequence

$$-\dot{\theta}(t_f),\ -\dot{\theta}(t_f - \Delta t),\ -\dot{\theta}(t_f - 2\Delta t),\ \ldots,\ -\dot{\theta}(t_0 + 2\Delta t),\ -\dot{\theta}(t_0 + \Delta t),\ -\dot{\theta}(t_0)$$

is also an optimal solution. Since the optimal solution is unique, the equations shown in (3.51) must hold.

For a given initial joint configuration $\theta(t_0)$, conservative motion can be deduced from equation (3.51), i.e.

$$\theta(t_0) = \theta(t_f), \theta(t_0+\Delta t) = \theta(t_f-\Delta t), \dots, \theta\left(\frac{t_f+t_0}{2}-\Delta t\right) = \theta\left(\frac{t_f+t_0}{2}+\Delta t\right) \quad (3.52)$$

Equation (3.51) and (3.52) show an important property of the proposed GOKC scheme. This property makes real-time kinematic control possible for a large class of trajectories. If the desired trajectory $X_d(t)$ is defined in the time interval $\left[t_0, \frac{t_f+t_0}{2}\right]$ with continuous first derivative, and if its first derivative is equal to zero at the terminal time $t = \frac{t_f+t_0}{2}$, i.e., $\dot{X}_d\left(\frac{t_f+t_0}{2}\right) = 0$, then by extending its mirror image in the right side of the vertical axis at $t = \frac{t_f+t_0}{2}$, we have

$$X_d(t_0+t) = X_d(t_f-t), \quad 0 \le t \le \frac{t_f-t_0}{2}. \quad (3.53)$$

Using equation (3.53) in equation (3.48), and noting that $\theta(t_f) = \theta(t_0)$, we can get the kinematic control sequence by using (3.45) recursively from t_f to $\frac{t_f+t_0}{2}$ as

$$\dot{\theta}(t_f), \dot{\theta}(t_f-\Delta t), \dot{\theta}(t_f-2\Delta t), \dots, \dot{\theta}\left(\frac{t_f+t_0}{2}+\Delta t\right), \dot{\theta}\left(\frac{t_f+t_0}{2}\right). \quad (3.54)$$

Since we have already shown that this sequence is the same as the following sequence

$$-\dot{\theta}(t_0),\ -\dot{\theta}(t_0+\Delta t),\ -\dot{\theta}(t_0+2\Delta t),\ \ldots,\ -\dot{\theta}\left(\frac{t_f+t_0}{2}-\Delta t\right),\ \dot{\theta}\left(\frac{t_f+t_0}{2}\right), \tag{3.55}$$

it means that we can obtain the kinematic control sequence recursively from t_0 to $\frac{t_f+t_0}{2}$, i.e., the real-time GOKC is realized.

For illustration and verification of the above mentioned method, a simulation has been carried out for a 3 degrees of freedom (d.o.f's) planar manipulator. The link lengths of this manipulator are $l_1 = l_2 = l_3 = 1$ meter, and all joint offsets and link twists are zero. The end-effector is required to trace a straight line from point a to point b in the 2-D plane, and then go back from point b to point a along the same path. The desired trajectory of the end-effector is given by

$$X_d(t) = \begin{bmatrix} x_0 - 2\left(2t - \frac{1}{2\pi}\sin(2\pi t)\right) \\ y_0 + 0.2\left(2t - \frac{1}{2\pi}\sin(2\pi t)\right) \end{bmatrix},\quad 0 \le t \le 0.5, \tag{3.56}$$

and

$$X_d(t) = \begin{bmatrix} x_0 - 2 + 2\left(2t - \frac{1}{2\pi}\sin(2\pi t)\right) \\ y_0 + 0.2 - 0.2\left(2t - \frac{1}{2\pi}\sin(2\pi t)\right) \end{bmatrix},\quad 0.5 < t \le 1. \tag{3.57}$$

where x_0 and y_0 are the values of point a in Cartesian space. With the sampling period $\Delta t = 0.001s$, the matrices Q and R are selected as

$$Q = \begin{bmatrix} 10000 & 0 \\ 0 & 10000 \end{bmatrix}, \qquad R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.58}$$

The simulation results are shown in Figure 3.3. The exact symmetry of the joint trajectories, $\theta(t)$, yields conservative motion, just as predicated by equation (3.52). Therefore, if the desired trajectory of the end effector is only defined for the period $0 \le t \le 0.5$ as in equation (3.56), then by extending its mirror trajectory (equation (3.57)) and calculating the joint velocity, $\dot{\theta}(t)$, from $t_f = 1$ to $\frac{t_f}{2} = 0.5$, the GOKC can also be achieved in real-time since we have the relations shown in equation (3.51) and (3.52).

To verify the effect of the GOKC, we have compared the performance index ($V = \sum_k \dot{\theta}^T(t_0 + k\Delta t)\,\dot{\theta}(t_0 + k\Delta t)\,\Delta t$) of the GOKC with that of K.&W algorithm, which is described in Section 2.2 as a global optimal approach. The GOKC has $V = 3.2508$ which is almost the same as $V = 3.2600$ for the K.&W. algorithm.

Note that if the desired trajectory of the end-effector has nonzero end velocities, i.e. $\dot{X}_d(t_f) \neq 0$, this method may not work. In such a case, after extending its mirror trajectory, the first derivative of the whole trajectory defined for the period $t_0 \le t \le (2t_f - t_0)$ is discontinuous in the moment t_f, so that equation (3.51) may not hold, i.e., we cannot realize the real-time GOKC.

3.3 STABILITY OF THE GOKC SCHEME

In this section, we shall be concerned with the stability of the closed-loop system formed when the GOKC control law in (3.44) is implemented. There are actually two key issues in the stability analysis of this "control" system. The first is to ensure the

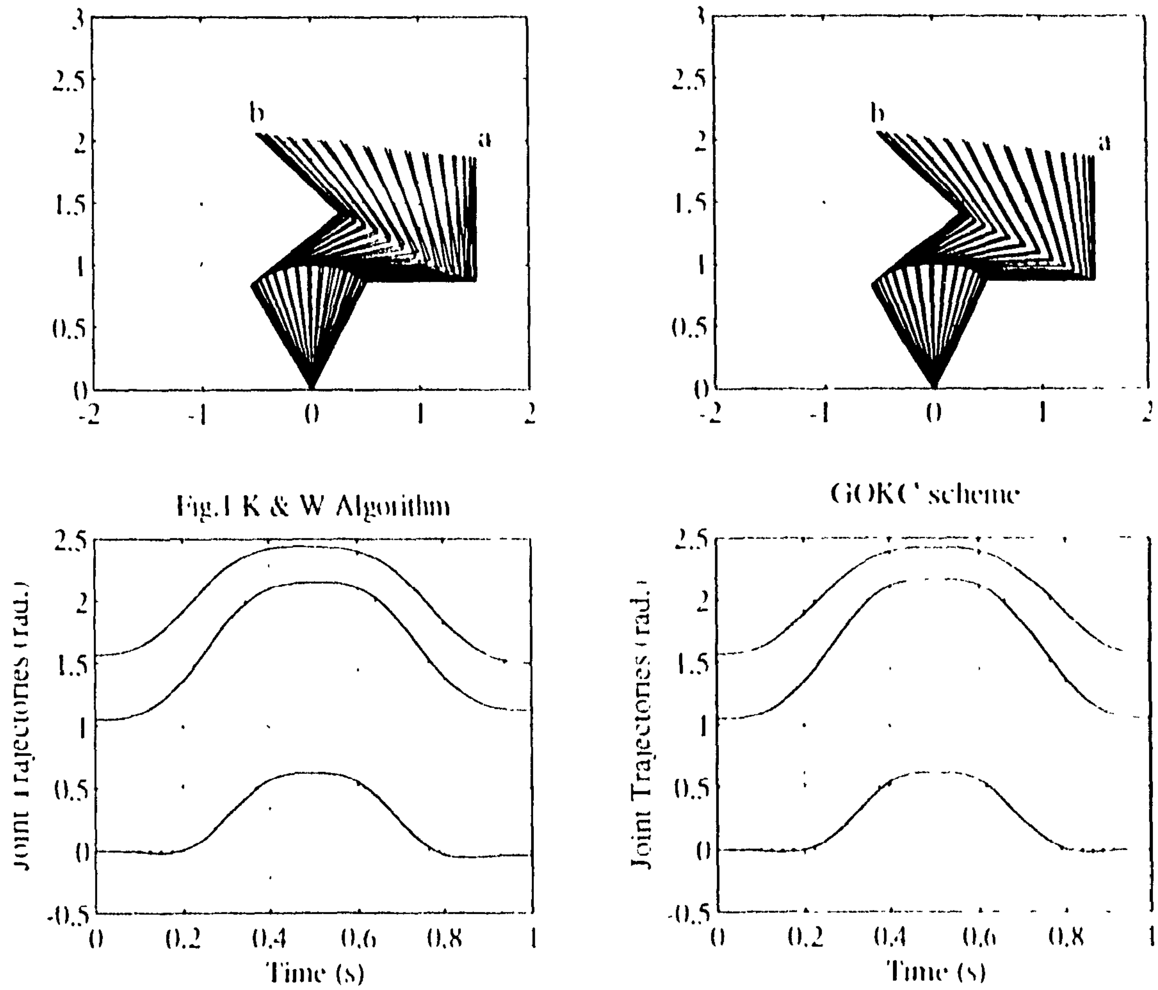

Fig. 3.3 Simulation Results of GOKC and K & W Algorithm

boundedness of the matrix $P(t)$, the vector $b(t)$ and the control signal $\dot{\theta}(t)$. The second is to ensure exponential stability since one of our major concerns is the tracking error.

Recall the open-loop system, i.e. the Jacobian relations of the manipulator

$$\dot{X} = AX + J\dot{\theta}, \quad A = 0 \tag{3.59}$$

and the optimal control law (3.44)

$$\dot{\theta} = -R^{-1}J^{T}(PX + b).$$

It clearly makes sense to restrict attention to the boundedness of J, Q, R, R^{-1}, X_d. Then with $[A, J]$ uniformly controllable, $[A, D]$ uniformly observable, where D is any matrix such that $DD^{T} = Q$, and $\dot{X}_d$ continuous, it can be shown [6] [7] that the GOKC achieves the desired boundedness and exponential stability. In (3.59), since $A = 0$, the controllability and the observability will be satisfied if Jacobian matrix J and matrix Q have full-rank. In this case, the stability of the GOKC is guaranteed.

3.4 GOKC FOR GENERAL REDUNDANCY RESOLUTION GOALS

In this section, we discuss how the GOKC scheme can be applied to achieve redundancy resolution goals other than that of minimum weighted kinetic energy ($\int \dot{\theta}^{T} R\dot{\theta}dt$). Actually, there are two ways to resolve redundancy to meet the needs of different performance criteria. One way is simply to replace the matrix Q or R with appropriate matrices. Another is to introduce the extended Jacobian technique in the GOKC. For some goals which cannot be described with an optimal performance criterion such as posture control, a generalized extended Jacobian technique called *configuration* *control* [11] can be used. In this section, three examples are provided. Each example is typical of its kind.

(1). True Kinetic Energy [8]

A manipulator's redundancy may be utilized to improve the dynamic performance of the system rather than the kinematic performance. The closed form representation of the dynamic equations of a manipulator is

$$\tau = M\ddot{\theta} + C(\theta, \dot{\theta}) + G(\theta) \tag{3.60}$$

where $M \in \Re^{m \times m}$ is the inertia matrix of the system, $\tau \in \Re^{m}$ represents the actuator torques applied at the joints, $\ddot{\theta}$ represents joint accelerations, $C(\theta, \dot{\theta}) \in \Re^{m}$ denotes torques caused by Coriolis and centrifugal effects, and $G(\theta) \in \Re^{m}$ represents gravitational effects. The dynamic control of robot manipulators involves generating joint torques τ, i.e., driving the joints with torques τ, such that the end-effector follows a desired trajectory closely. It is expected that a good control law would require a minimum amount of the joint torques or minimum true kinetic energy. The true kinetic energy of a manipulator system is defined as

$$T = \frac{1}{2}\dot{\theta}^T M\dot{\theta}. \tag{3.61}$$

Therefore an integral performance criterion which minimizes the true kinetic energy over the time duration of interest while the end-effector traces the desired trajectory can be chosen as

$$\text{Minimize } V = \int_{t_0}^{t_f} [\dot{\theta}^T M \dot{\theta} + (X - X_d)^T Q (X - X_d)] \, dt \tag{3.62}$$

Obviously, equation (3.3) is the same as equation (3.3) provided $M = R$. Note that M is a symmetric positive definite matrix, so that we can apply the GOKC to achieve the minimization of true kinetic energy. However, M is a function of θ, i.e., it is time-varying and must be calculated at each step of time. A simple simulation is given below for illustration. The three-link planar manipulator shown in Figure 3.3 is used in the simulation. The manipulator dynamic parameters are link masses $m_1 = m_2 = m_3 = 10.0$kg, joint viscous friction coefficients $v = v = v = 40.0$Nt.m./rad.s^{-1}; the link inertias are modeled by thin uniform rods. The inertia matrix is then given by $M = [m_{ij}]$, where

$$m_{11} = 73.33 + 50\cos\theta_2 + 30\cos\theta_3 + 30\cos(\theta_2 + \theta_3)$$

$$m_{12} = m_{21} = 40 + 25\cos\theta_2 + 15\cos(\theta_2 + \theta_3) + 30\cos\theta_3$$

$$m_{13} = m_{31} = 16.67 + 15\cos\theta_3$$

$$m_{22} = 40 + 30\cos\theta_3$$

$$m_{23} = m_{32} = 16.67 + 15\cos\theta_3$$

$$m_{33} = 16.67$$

The desired end-effector trajectory is chosen as same as equations (3.56) and (3.57). Let $Q = diag\{1000, 1000\}$, R=M, and $\Delta t = 1.0$ms, the GOKC yields a solution which minimizes the true kinetic energy. Figure 3.4 shows the profile of $\dot{\theta}^T M \dot{\theta}$. The performance index $\sum \dot{\theta}^T M \dot{\theta} \Delta t = 1259.1$, as compared with that of 1251.3 in the case of $R = I_3$.

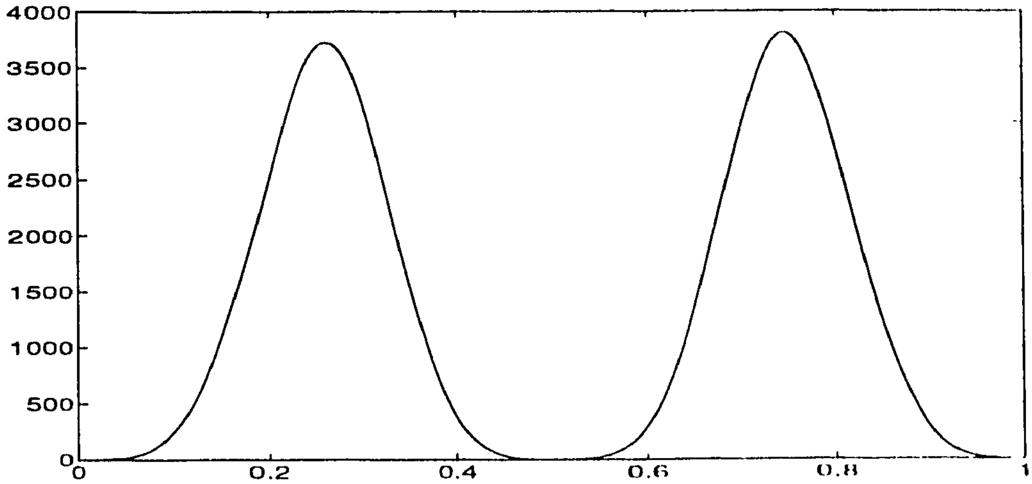

Fig. 3.4 The profile of true kinetic energy in GOKC with $R=M$

(2). Gravitational Torques [9]

In this case, redundancy is utilized to minimize the effect of gravity loading on the joints. This yields the optimal configuration for which the payload capacity of the manipulator is maximized. It also enables the user to optimally preconfigure the redundant manipulator before picking up a payload.

The joint torque due tc gravity loading is represented by $G(\theta)$ in equation (3.60), and is configuration dependent. Let us define the scalar weighted gravity loading objective function as

$$L = G^T(\theta) W G(\theta) \tag{3.63}$$

where the $m \times m$ constant diagonal matrix W represents the weighting factors assigned by the user to the joints. To optimize L subject to the end-effector constraints $X = f(\theta)$, we require (from Section 2.1.3)

$$N_e^T(\theta)\frac{\partial L}{\partial \theta} = 0 \tag{3.64}$$

where $N_e(\theta)$ is the $m \times (m-n)$ matrix whose columns span the null space of the Jacobian matrix J, i.e., $JN_e = 0$. The optimality condition (3.64) can be treated as a set of additional kinematic constraint functions by defining $\Phi = h(\theta) = N_e^T(\theta)\frac{\partial L}{\partial \theta}$, and their desired trajectory as $\Phi_d(t) = 0$. The augmented kinematic constraints can be expressed as

$$Y = \begin{bmatrix} X \\ \Phi \end{bmatrix} = \begin{bmatrix} f(\theta) \\ h(\theta) \end{bmatrix} \tag{3.65}$$

and their desired trajectories are

$$Y_d(t) = \begin{bmatrix} X_d(t) \\ \Phi_d(t) \end{bmatrix}. \tag{3.66}$$

The extended Jacobian matrix is

$$J_{EX} = \begin{bmatrix} J(\theta) \\ \frac{\partial}{\partial \theta}h(\theta) \end{bmatrix}. \tag{3.67}$$

Finally, by usin J_{EX}, Y and $Y_d(t)$ in the GOKC law in place of J, X and $X_d(t)$ respectively, we can attempt to minimize L while ensuring that the end-effector tracks the desired path. Note that the weighting matrix Q should be set to be an $m \times m$ matrix since

J_{EX} is an $m \times m$ matrix in this case. A detailed simulation study using the same technique can be found in Section 4.3.

(3). Posture Control [10]

The presence of redundant degrees-of-freedom in a manipulator structure results in an infinite number of distinct arm configurations with the same end-effector position and orientation. This leads to a phenomenon known as "self-motion", which is a continuous motion of the joints that keeps the end-effector motionless. In posture control, we wish to utilize "self-motion" in order to adjust the arm configuration to certain desired postures. For this purpose, we define $m-n$ additional kinematic constraints $\Phi(\theta) = [\phi_1, \phi_2, \ldots, \phi_{m-n}]$, which specify the desired posture. The desired trajectories $\Phi_d(t)$ can be determined such that the manipulator reaches its desired posture smoothly. Once the $\Phi(\theta)$ and $\Phi_d(t)$ have been defined, we can use the generalized extended Jacobian technique of configuration control [9] to implement posture control by means of the GOKC framework. The only difference from the previous case (gravitational torques) is that $\Phi_d(t)$ can be arbitrary functions because $\Phi(\theta)$ does not come from an optimal performance criterion.

For the sake of illustration, let us consider the three-link planar manipulator shown in Figure 3.5. This manipulator is redundant with the degree of redundancy equal to one. Suppose that we want to control the elbow position A or the shoulder angle θ_1 in addition to the end-effector position, the additional kinematic constraint $\phi(\theta)$ can be defined in a number of ways, e.g.,

$\phi(\theta) = l_1 \sin\theta_1$ elbow vertical position

$\phi(\theta) = l_1 \cos\theta_1$ elbow horizontal position

$\phi(\theta) = \theta_1$ shoulder angle

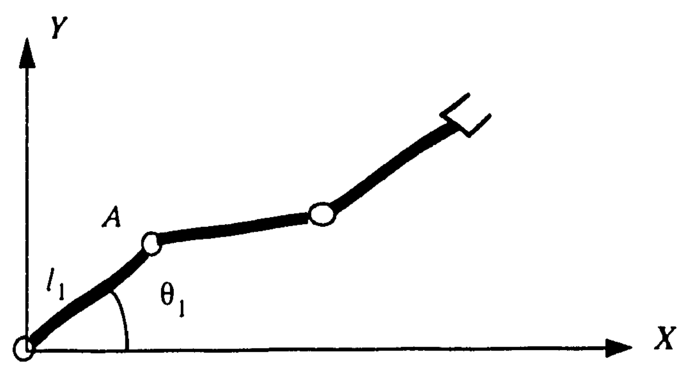

Fig. 3.5 Planar Three-link Manipulator

To adjust the arm configuration from its initial posture $\phi(t_0)$ to its desired final posture $\phi(t_f)$, we should define a smooth trajectory $\phi_d(t)$ with zero velocity at the final point, i.e. $\dot{\phi}_d(t_f) = 0$. For example, we can choose a cycloidal trajectory

$$\phi_d(t) = \phi_d(t_0) + \frac{\phi_d(t_f) - \phi_d(t_0)}{2\pi}\left[\frac{2\pi t}{T} - \sin\frac{2\pi t}{T}\right] \tag{3.68}$$

where $T = t_f - t_0$ is the specified time duration.

The augmented kinematic functions Y, its desired trajectory $Y_d(t)$ and the extended Jacobian matrix J_{EX} can be obtained by using the same procedure as that in the extended Jacobian technique, that is

$$Y = \begin{bmatrix} X \\ \Phi \end{bmatrix}, \qquad Y_d = \begin{bmatrix} X_d(t) \\ \Phi_d(t) \end{bmatrix}, \qquad J_{EX} = \begin{bmatrix} \frac{\partial}{\partial\theta} f(\theta) \\ \frac{\partial}{\partial\theta}\Phi(\theta) \end{bmatrix}. \tag{3.69}$$

By using Y, Y_d and J_{EX} to replace X, X_d and J in the GOKC scheme, the desired posture control can be achieved. A simulation based on a seven-joint redundant manipulator is shown in Section 4.6.

3.5 GENERAL REMARK ON GOKC

As mentioned in Chapter 2. in conventional control schemes for redundant manipulators, there are usually two parts to the control law. One determines the motion of the end-effector, and the other contributes to the "self-motion", e.g. by optimizing some performance criterion. But, when the GOKC is used to control redundant manipulators, we cannot explicitly separate the control effect into such two parts.

The GOKC can be used to control non-redundant manipulators as well as redundant ones. In the case of non-redundant manipulators, we have to make a compromise between the tracking error of the end-effector and the upper bound of the joint velocities. We can select the matrices Q and R according to the task priority and the limits on the joint velocities. For the redundant manipulators, there are two cases in the GOKC which are worth mentioning. Some redundancy resolution goals such as minimum joint velocities and minimum kinetic energy can be achieved by using the GOKC directly. In these applications, the tracking error E and the performance measure are two distinct terms in the objective function V. Since there exist some degrees of redundancy, the GOKC may uniquely determine a sequence of the joint configurations which minimizes the tracking error as well as the performance measure. In other goals such as posture control and minimum gravitational torques, the tracking error and the performance measure are in the same term that forms an augmented tracking error

$$E_{Ex} = Y_d - Y = \begin{bmatrix} X_d - X \\ \Phi_d - \Phi \end{bmatrix} . \tag{3.70}$$

This corresponds to a non-redundant system if $Y \in \Re^m$. The GOKC can achieve arbitrarily small augmented error E_{EX} provided the matrix Q is sufficiently large. On the other hand, we can restrict the joint velocities to lie within given limits by increasing the magnitude of the diagonal elements of R. However, this will be at the expense of the augmented error. Therefore, a trade-off should be made in selecting Q and R. The restricted joint velocities yield a singularity-robust solution, and give an important advantage which may overcome the difficulty of algorithmic aingularities. This advantage makes it easier to achieve various redundancy resolution goals using GOKC.

3.6 REFERENCES

[1] B.D.O. Anderson & J.B. Moore, *Optimal Control*, Prentice-Hall, Englewood Cliffs, NJ, 1990.

[2] F.L. Lewis, *Optimal Control*, New York: John Wiley and Sons, 1986.

[3] A.P. Sage, *Optimal Systems Control*, Prentice-Hall, Englewood Cliffs, N.J., 1968.

[4] L.S. Pontryagin, V.G. Boltyanskii, R.V. Gamkrelidze, and E.F. Mischenko, *The Mathematical Theory of Optimal Processes*, K.N. Trirogoff (transl.), L.W. Neustadted. New York: Interscience Publishers, 1962.

[5] D.E. Kirk, *Optimal Control Theory: An Introduction*, Pretice-Hall, 1970.

[6] R.E. Kalman, "Contributions to The Theory of Optimal Control", *Bol. Soc. Matem. Mex.*, 1960, pp. 102-119.

[7] B.D.O. Anderson & J.B. Moore, "Detectability and Stabilizability of Time-Varying Discrete-Time Linear Systems", *SIAM J. of Control and Optimization*, Vol. 18, No. 1, 1981, pp. 20-32.

[8] S. Gupta and J.Y.S. Luh, "Closed-Loop Control of Manipulators with Redundant Joints using Hamilton-Jacobi-Bellman Equation", *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Sacramento, Califonia, April, 1991.

[9] H. Seraji, "Task-Based Configuration Control of Redundant Manipulators", *J. Rob.*

Syst., Vol. 9, No. 3, pp. 411-451, 1992.

[10] H. Seraji, M. Long and T. Lee, "Configuration Control of 7 DOF Arms", *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Sacramento, California, April, 1991.

[11] H. Seraji, "Configuration Control of Redundant Manipulators: Theory and Implementation", *IEEE Trans. on Robotics and Automation*, Vol. 5, No. 4, August 1989.

CHAPTER 4 OPTIMAL KINEMATIC CONTROL OF AN ISOTROPIC REDUNDANT MANIPULATOR

In this chapter, five studies of the optimal kinematic control for redundancy resolution are presented. The globally optimal kinematic control (GOKC) schemes of the previous chapter are applied to an *isotropic* redundant manipulator with seven revolute joints. This manipulator, named REDIESTRO, has been designed and constructed at the McGill Centre for Intelligent Machines. This robot is now fully operational and experimental results for the verification of optimal kinematic control are also presented in this chapter. All the desired trajectories are chosen to have zero velocity at t_f, so that the real-time GOKC scheme is used in this chapter.

4.1 THE ISOTROPIC REDUNDANT MANIPULATOR

A manipulator is termed isotropic if the condition number of its Jacobian matrix can reach the minimum value (=1) at certain kinematically achievable configurations [2]. These configurations are called *isotropic configurations*. The main advantage of an isotropic manipulator arises from the fact that the condition number of a manipulator's Jacobian is a measure of the sensitivity of the solution of the inverse kinematics problem for the manipulator [1], i.e., if we consider the system $\dot{X} = J\dot{\theta}$, for a given $\Delta\dot{X}$, the smaller the condition number of J, the smaller will be the magnitude of $\Delta\dot{\theta}$. Therefore, more accurate kinematic control can be expected for isotropic manipulators. An isotropic manipulator has been designed at the McGill University Centre for Intelligent Machines. This is a seven-axes, revolute joint isotropic redundant manipulator which has been named REDIESTRO (REDundant, Isotropically Enhanced, Seven-Turning-pair RObot) [3] [4]. The Hartenberg-Denavit parameters of this manipulator are given in the following table:

Table 1: Hartenberg-Denavit Parameters of REDIESTRO

Joint No.	a (mm)	b (mm)	α (degree)	θ (degree)
1	0.0	0.0	-58.3127	0.0
2	231.1273	-22.9113	-20.0289	-11.0101
3	0.0	36.9275	105.2568	91.9445
4	398.8379	0.0	60.9094	113.9273
5	0.0	-471.5880	59.8823	-2.2616
6	135.5890	578.2057	-75.4715	150.2461
7	234.4458	-145.0499	0.0	63.7636

In table 1, a is the link length, b is the offset distance, α is the link twist angle, and θ is the joint angle. From these four parameters, the 4×4 homogeneous transformation, T_i^{i-1}, which relates the position and orientation in coordinate system i to those in coordinate system $i-1$, may be computed using the expression [5] :

$$T_i^{i-1} = \begin{bmatrix} c\theta_i & -s\theta c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & b_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.1}$$

where $c\theta_i$ and $s\theta_i$ indicate the cosine and sine respectively of θ_i. The transformation between various link coordinate systems may be obtained through multiplication of the intermediate transformation matrices.

The joint angles listed in Table 1 denote one isotropic configuration. The condition number of the Jacobian matrix J is equal to one when the manipulator is in this configuration. Fig 4.1 is the assembly drawing of REDIESTRO in its isotropic configuration.

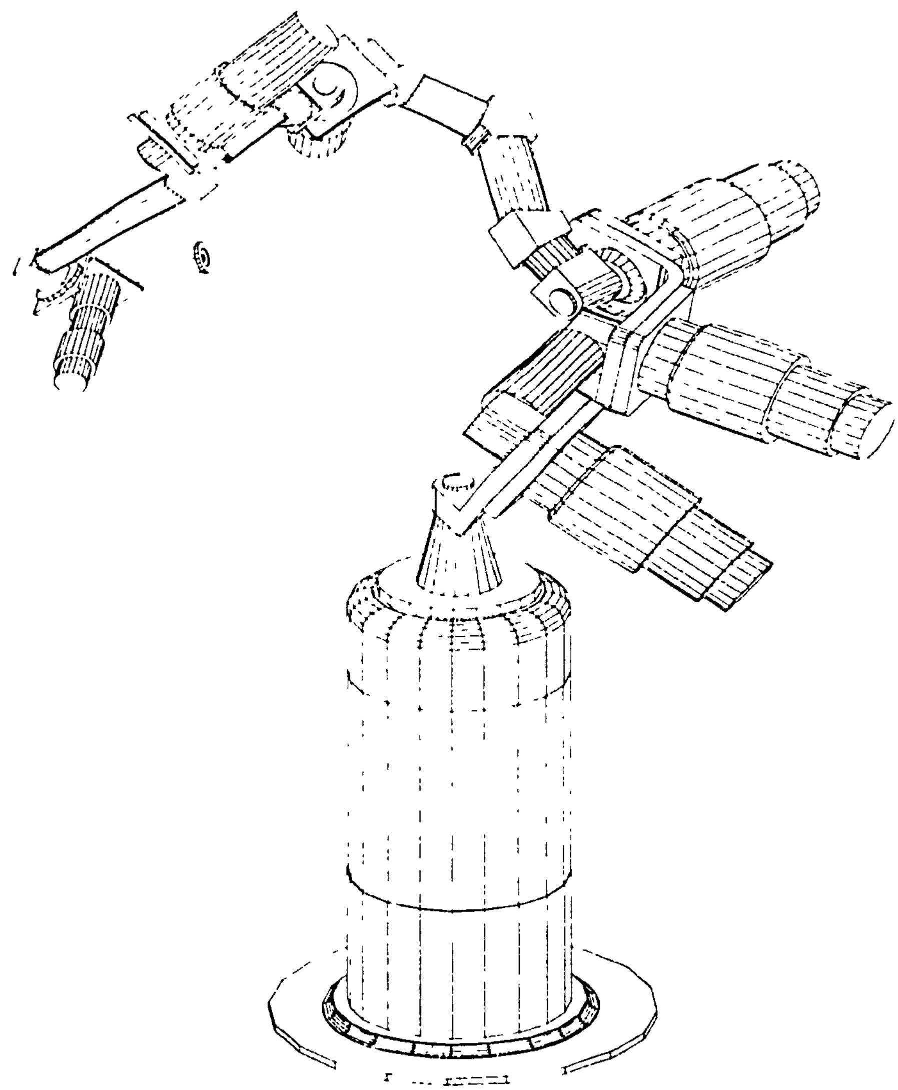

Fig. 4.1 Isotropic Manipulator REDIESTRO

4.2 MINIMUM JOINT VELOCITIES

The original globally optimal kinematic control problem is to minimize a weighted norm of joint velocities $\dot{\theta}^T R \dot{\theta}$ globally while the end-effector tracks the desired trajectory. When this control scheme is applied to REDIESTRO in simulation studies, a local optimization algorithm is used for comparison. This local optimization algorithm is based on equation (2.16) in Chapter 2, i.e.,

$$\dot{\theta} = J^{\dagger} \dot{X}. \tag{4.2}$$

The position error *E(t)* is given by

$$E(t) = X_d(t) - X(t). \tag{4.3}$$

We assume that *E(t)* satisfies the equation

$$\dot{E}(t) + KE(t) = 0 \tag{4.4}$$

where $\dot{E}(t)$ is the velocity error, $K = k_p I$, with k_p a positive constant. Then $E(t) \to 0$ as $t \to \infty$. Using equation (4.3) and (4.4) in equation (4.2) yields

$$\dot{\theta} = J^{\dagger} (\dot{X}_d + KE). \tag{4.5}$$

In the present simulation, the locally optimal kinematic control (LOKC) of equation (4.5) is compared with the GOKC. The desired trajectory of the end-effector was chosen as

$$X_d(t) = \begin{bmatrix} \Omega_\gamma(t) \\ \Omega_\beta(t) \\ \Omega_\alpha(t) \\ P_x(t) \\ P_y(t) \\ P_z(t) \end{bmatrix} = \begin{bmatrix} \Omega_\gamma(0) \\ \Omega_\beta(0) \\ \Omega_\alpha(0) \\ P_x(0) + 1.5\,(t - \frac{1}{2\pi}\sin 2\pi t) \\ P_y(0) + 1.5\,(t - \frac{1}{2\pi}\sin 2\pi t) \\ P_z(0) + 1.5\,(t - \frac{1}{2\pi}\sin 2\pi t) \end{bmatrix}, \qquad 0 \le t \le 1. \tag{4.6}$$

where Ω and P indicate the orientation and the position of the end-effector respectively, the subscripts x, y, z stand for the X, Y, Z axis of the base frame coordinate, and the subscripts γ, β, α stand for the orientations about X, Y, Z axis. This trajectory has $\dot{X}_d(t_f) =$, so that we can use real-time GOKC to trace this trajectory while minimizing the weighted norm of the joint velocities at the same time. In the simulation study, the (6×6) matrix Q is chosen as a diagonal matrix, $Q = diag\{10^4 \dots 10^4\}$, while the (7×7) matrix R is selected as $R = I_7$, and the sampling period $\Delta t = 0.001$ s. The simulation results in Figure 4.2 show that the maximum relative tracking error is less than 1%. Figure 4. 3 gives the corresponding profiles under LOKC and GOKC. As can be seen, the GOKC performs optimization better than the local method does.

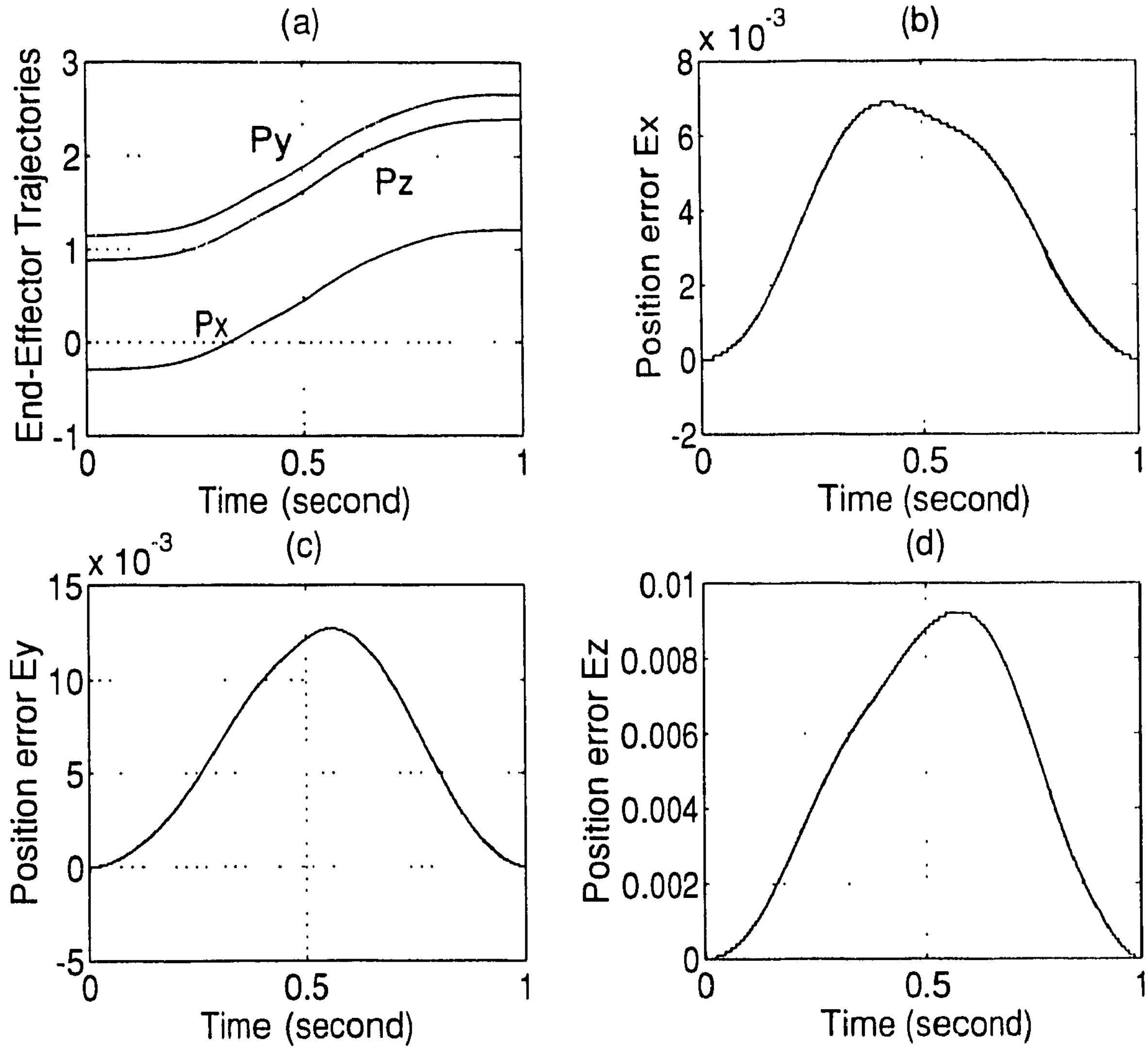

Fig. 4.2 Trajectories of End-Effector and Tracking Error in Minimum Joint Velocity Control using GOKC

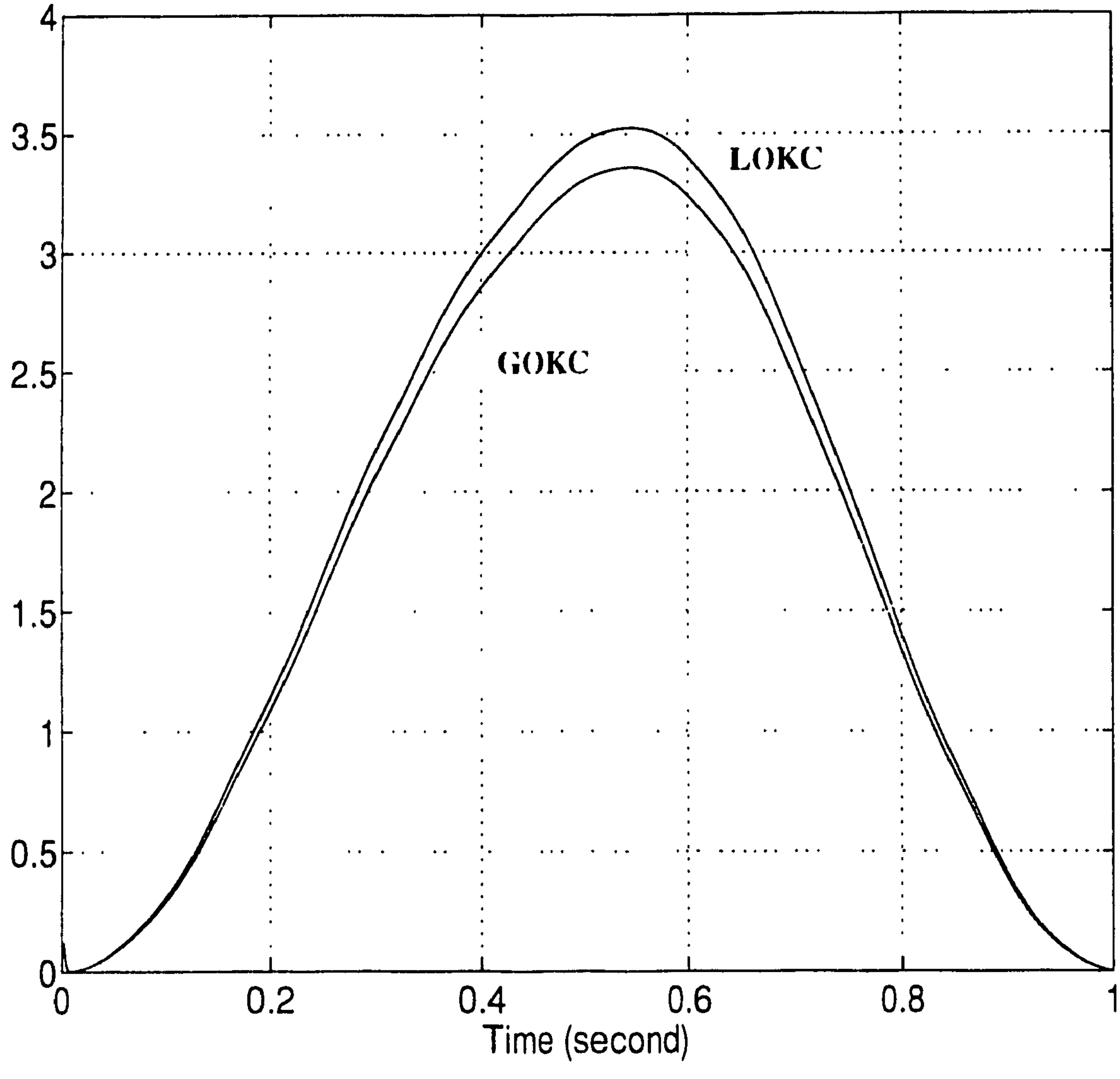

Fig. 4.3 Profiles of the 2-Norm of the joint velocities obtained using GOKC and LOKC

4.3 MINIMUM JOINT MOTIONS

In this section, the extended Jacobian technique is applied to the optimization task within the GOKC framework. The optimization task is to distribute the joint motions in such a way that a weighted sum of the joint motions is kept at a minimum. For this purpose, the optimization objective function is chosen as follows:

$$G(\theta(t)) = \frac{1}{2}\sum_{i=1}^{7} k_i [\theta_i(t) - \theta_i(t_0)]^2, \qquad t_0 \le t \le t_f, \tag{4.7}$$

where k_i is the scalar weighting for the motion of joint i. $\theta_i(t)$ and $\theta_i(t_0)$ denote the current joint angle and the initial joint angle respectively. By choosing appropriate numerical values for k_i, we can resolve the end-effector motion among the joints such that the joints with larger k_i move less than those with smaller k_i.

In order to minimize the objective function $G(\theta)$ subject to the end-effector constraint $\dot{X} = J\dot{\theta}$, we can apply the results obtained in section 2.1.2 to get the necessary condition for the constrained optimization problem as

$$(I - J^{\dagger}J)\frac{\partial G}{\partial \theta} = 0 \tag{4.8}$$

where $J^{\dagger}$ is the pseudoinverse of J. The $m \times m$ matrix $(I - J^{\dagger}J)$ is of rank $r\ (= m - n)$ and therefore equation (4.8) reduces to

$$N_e\frac{\partial G}{\partial \theta} = 0 \tag{4.9}$$

where N_e is a $r \times m$ matrix formed from r linearly independent rows of $(I - J^{\dagger}J)$. The rows of N_e span the r-dimensional null-space of the Jacobian J, since $J(I - J^{\dagger}J) = 0$ and $(I - J^{\dagger}J)$ is symmetric. Equation (4.9) implies that the projection of the gradient of the objective function $G(\theta)$ onto the null-space of the Jacobian matrix J must be zero. Now, we define an additional kinematic function as

$$\phi(\theta) = N_e \frac{\partial G}{\partial \theta} = N_e \begin{bmatrix} k_1 (\theta_1 (t) - \theta_1 (t_0)) \\ k_2 (\theta_2 (t) - \theta_2 (t_0)) \\ \dots \\ k_7 (\theta_7 (t) - \theta_7 (t_0)) \end{bmatrix} \tag{4.10}$$

and the desired trajectory as $\phi_d(t) = 0$ to represent equation (4.9). Then we construct an augmented configuration vector Y as

$$Y = \begin{bmatrix} f(\theta) \\ \phi(\theta) \end{bmatrix} = \begin{bmatrix} X \\ N_e \frac{\partial G}{\partial \theta} \end{bmatrix}. \tag{4.11}$$

The desired trajectory $Y_d(t)$ is defined by

$$Y_d(t) = \begin{bmatrix} X_d(t) \\ \phi_d(t) \end{bmatrix} = \begin{bmatrix} X_d(t) \\ 0 \end{bmatrix}. \tag{4.12}$$

Thus the extended Jacobian matrix J_{EX} is now

$$J_{EX} = \begin{bmatrix} J \\ \frac{\partial}{\partial\theta}(N_e \frac{\partial G}{\partial\theta}) \end{bmatrix}. \quad (4.13)$$

The GOKC can now be applied directly to ensure that $Y(t)$ tracks the desired trajectory $Y_d(t)$ by replacing X, X_d, J with Y, Y_d and J_{EX} respectively in the control law. Thus, the GOKC minimizes the error E (= $Y_d - Y$) and the joint velocity globally. In the case of the degree of redundancy $r = 1$, there is no enough degree of redundancy to meet the needs of the minimum joint motion and the minimum norm of joint velocities. By choosing proper values of matrices Q and R, the GOKC automatically balance the trade-off of these two requirements. Therefore, a singularity-robust redundancy resolution is obtained at the expense of tracking accuracy. Note that the additional task is given by an instantaneous performance index (4.7), the GOKC in this case yields a solution which locally optimiz the joint motions and globally minimize the joint velocities. Because the resolution is singularity-robust, the algorithmic singularities, which may be introduced into J_{EX} due to the submatrix $\frac{\partial}{\partial\theta}(N_e \frac{\partial G}{\partial\theta})$, can be avoided. Fig. 4.4 shows the simulation results for REDIESTRO where the degree of redundancy $r = 1$. The desired end-effector trajectory is the same as in equation (4.6). Plot (d) gives the joint 2 trajectories for two different values of k_2. It can be see that larger value of k_2 constrains the motion of the joint more than the smaller value.

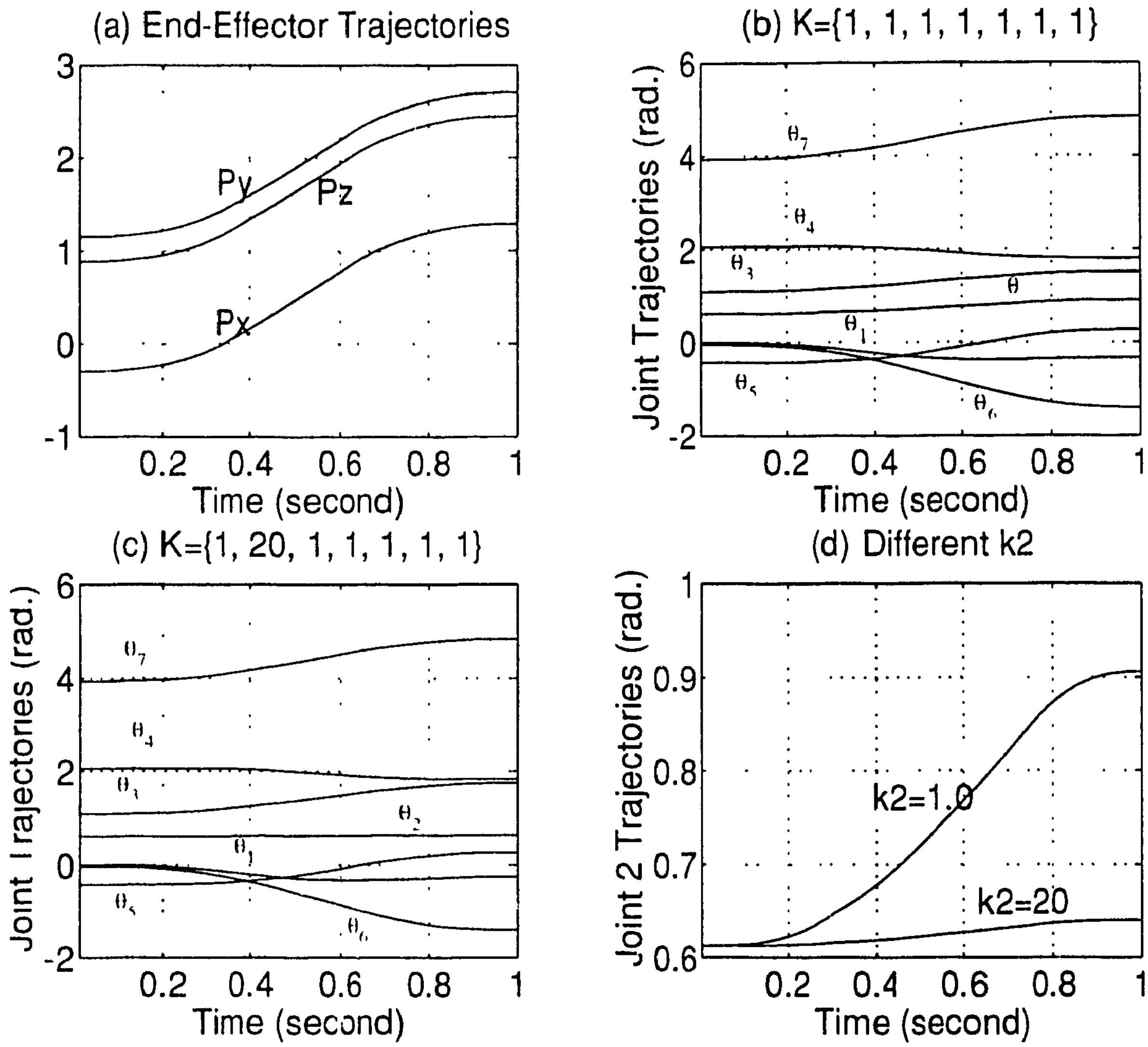

Fig. 4.4 Minimum Joint Motion Control using GOKC

4.4 JOINT LIMITS AVOIDANCE

When a robot manipulator carries out some motion task, its links may come into collision with each other if one or more joints exceed certain limits. The limit for each joint is usually a function of all joint angles and is thus very difficult to determine. A practical way to detect whether a joint reaches its limit is to use proximity (distance) sensors. For non-redundant manipulators, the only way to avoid link collisions when some joint limits are exceeded is to stop the current task immediately. This is one of the major limitations of the application of non-redundant manipulators. But, in redundant manipulators, the redundant d.o.f.'s may be used to avoid joint limits while the manipulator continues its basic motion task. Several methods have been reported on this topic [6]. In this section, a new method based on optimal joint motion is proposed to implement joint limit avoidance.

We assume that we have information on each joint limit, or that we have clear sensor signals when a certain joint is close to its limit. The principle of the proposed method is to constrain the motion of the joint that reaches close to its limit. When no joint limits are being exceeded, the redundant d.o.f.'s are used to achieve the minimum joint motion, as described in Section 4.3, (with all $k_i = 1$). If a certain joint, say joint i, moves close to its limit, a larger k_i is assigned to restrict the motion of joint i. In this way the joint limit avoidance is implemented. Due to the global optimization, if k_i is large enough, the GOKC will not only restrict the motion of joint i, but may also drive this joint back toward its initial value. This property allows the possibility of avoiding multiple joint limits if these limits do not occur simultaneously. For example, when joint i reaches the neighborhood of its limit, a large k_i in the minimum joint motion control moves it away from the limit and toward its initial position. When joint i is sufficiently far from the limit, we set k_i back to its normal value, (all k are normally set to the same value). If joint j moves close to its limit later during the operation of the task, we repeat the above procedure for joint j.

In the following simulation, we assume that θ_3 is constrained by

$$0 \leq \theta_3 \leq 1.2 \text{ rad.} \tag{4.14}$$

The end-effector is required to track the trajectory as expressed by equation (4.6). The normal values of k_i are chosen as $k_i = 1.0$, $i = 1, 2, ..., 7$. When θ_3 reaches 1.2 rad., k_3 is set to 3. But, a sudden change of k_3 will cause "chattering" in the robot system. To avoid this a (digital) filter is necessary in practice. To incorporate the inequality constraint smoothly, it is necessary to introduce a "buffer" region around the joint limit. In the present case, this buffer region is defined as $\theta_3 \subset [1.19, 1.20]$. Thus the inequality constraint is active when $\theta_3 \geq 1.19$. The weighting factor k_3 was made dependent on θ_3 as follows:

$$k_3 = \begin{cases} 1 & \theta_3 < 1.19 \\ 1 + 2\sin\frac{\pi}{2}\left(\frac{\theta_3 - 1.19}{1.2 - 1.19}\right) & 1.19 \leq \theta_3 \leq 1.2 \\ 3 & 1.20 < \theta_3 \end{cases} \tag{4.15}$$

In order to restrict the relative tracking error within 2%, the 7×7 matrices Q and R are chosen as $Q = diag\{\ 10^4 \ldots 10^4 \quad 100\ \}$ and $R = I_7$. The sampling period is $\Delta t = 1$ ms. The performance of the GOKC with joint limit avoidance is shown in Figure 4.5. If k_3 increases gradually up to 20 when θ_3 reaches its limit, the trajectory of θ_3 moves away from the limit and tends toward its initial value as shown in Figure 4.6. This property gives the possibility of avoiding other joint limits in successive motion tasks.

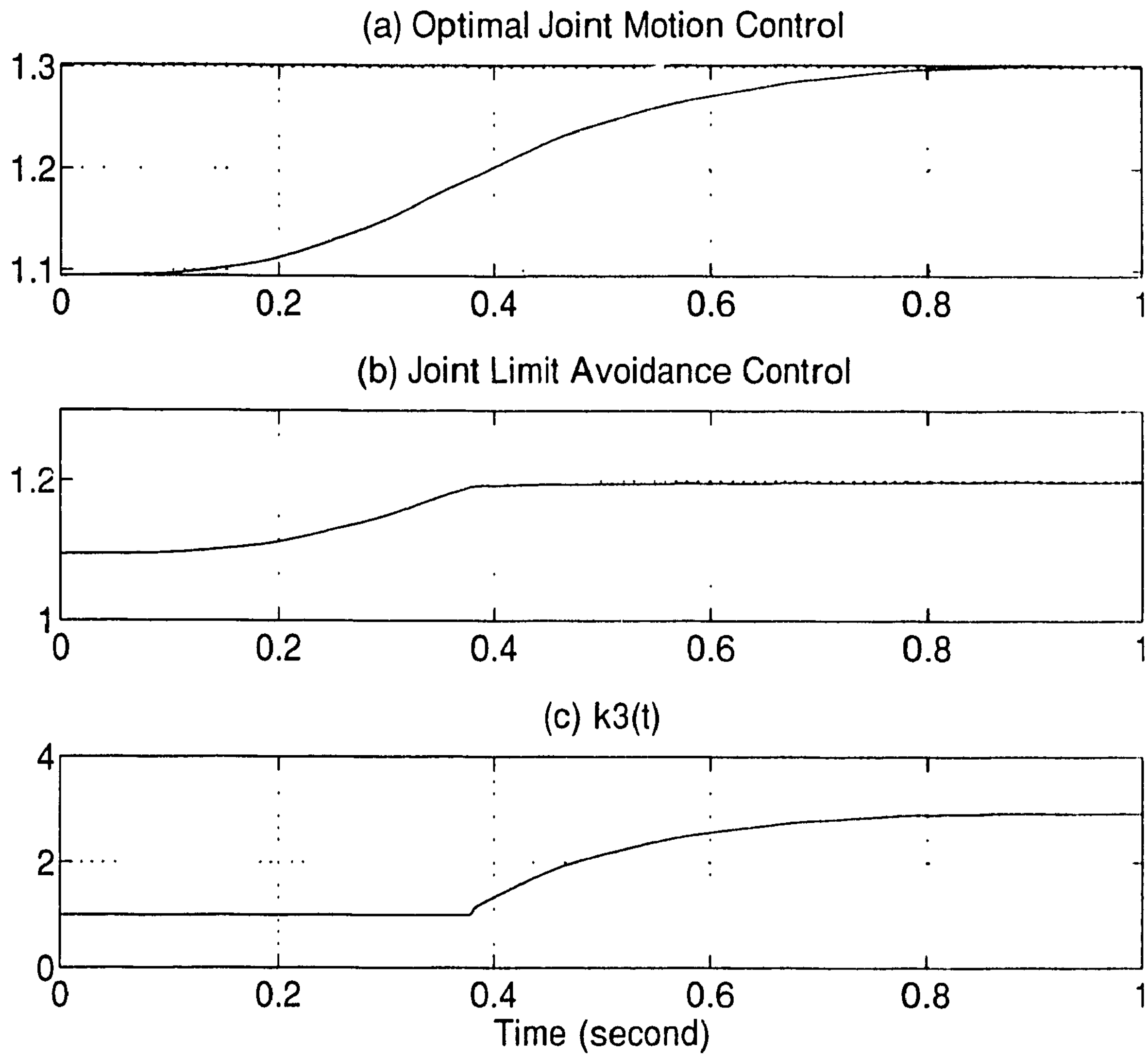

Fig. 4.5 Joint Limit Avoidance Using GOKC, (a) Trajectory of θ_3 with no limits (b) Trajectory of θ_3 with limits, (c) Change of k_3 when joint 3 nears its limit

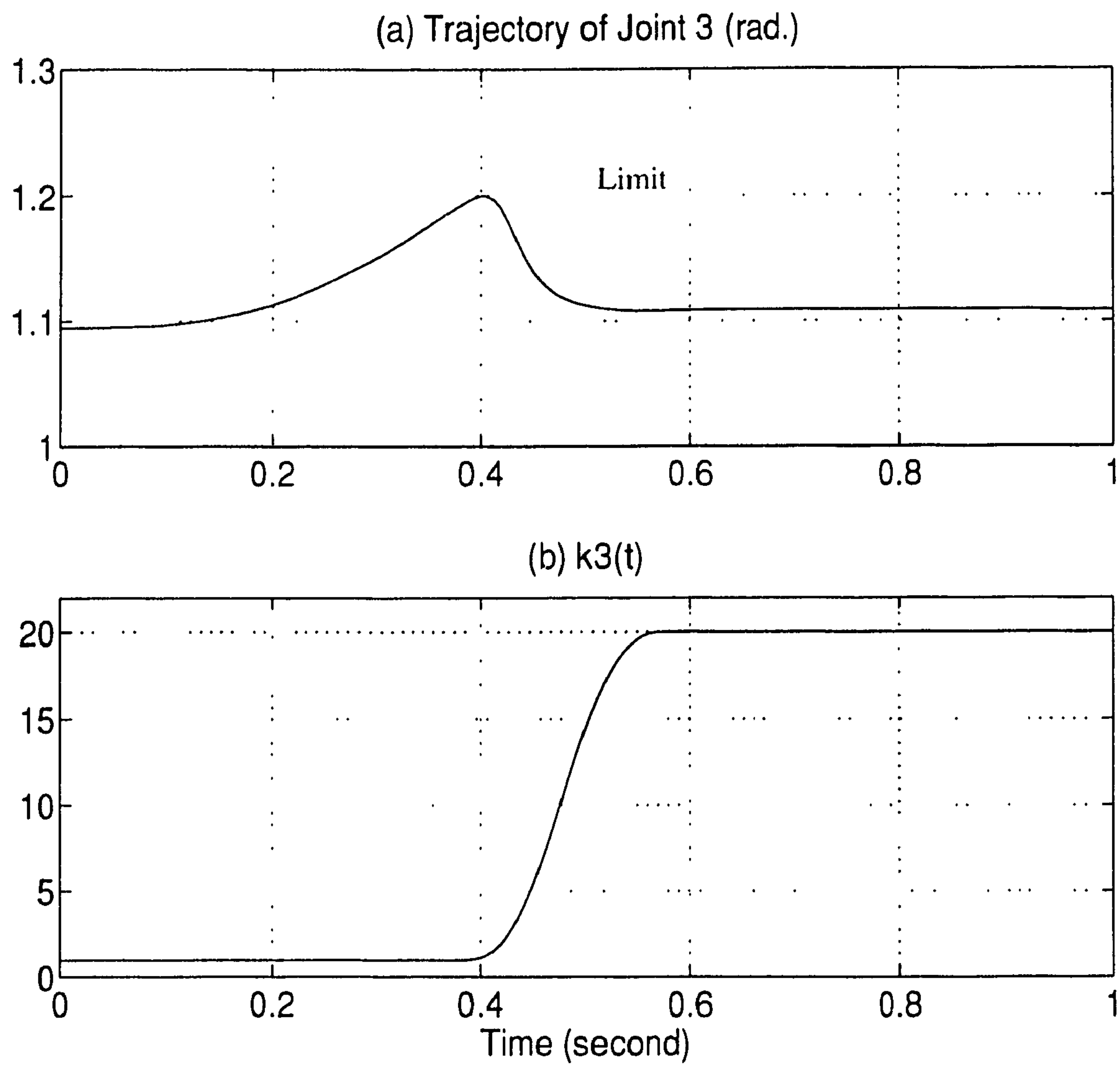

Fig. 4.6 Joint Limit Avoidance Using GOKC, (a) Trajectory of θ_3

(b) Change of k_3 when joint 3 nears its limit

4.5 OBSTACLE AVOIDANCE

The optimization control problem in this section is to demand the end-effector to track some desired trajectory while simultaneously ensuring that none of the manipulator links collide with workspace obstacles. In formulating this problem, we can follow the approach proposed by R.Colbaugh et al [7], which is based on the configuration control technique. In this formulation, all workspace obstacles are enclosed in convex volumes, and each volume defines a "space of influence" (SOI). The SOI's are assumed to be spheres in three-dimensional work space, but extension to other geometrical shapes is straightforward (see [7]). If any point on the manipulator enters the SOI of an obstacle, the manipulator redundancy is used to inhibit the motion of that point in the direction toward the obstacle. To implement this, we first need to find the "*body critical point*" (the point on the manipulator currently at minimum distance from the obstacle) and then determine whether this point is within the obstacle's SOI. To find the body critical point, one may proceed by locating all of the *link critical points* (the points on each link closest to the obstacle) and then pick the point which is closest to the obstacle as the body critical point. Figure 4.7 shows the relationship of link i, an obstacle and its associated SOI in the workspace.

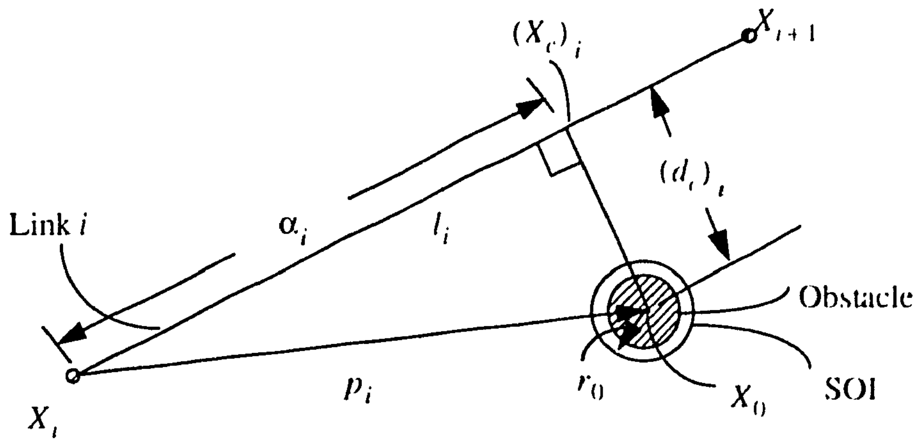

Fig. 4. 7. Relation of Link i and Obstacle

In Figure 4.7, l_i is the length of link i, $X_0 \in \Re^3$ defines the position of the SOI center in the base frame coordinate, r_0 is the radius of the SOI chosen to allow some "buffer" between the SOI and the obstacle, $X_i \in \Re^3$ is the location of joint i relative to the base frame, α_i is the distance measured along link i from joint i to the link critical point $(X_c)_i$, $p_i = X_o - (X_c)_i$ is the 3×1 vector pointing from joint i to the center of the obstacle. By using these definitions in Figure 4.7 and defining the 3×1 unit direction vector $e_i = [X_{i+1} - X_i]/l_i$, the following recursive algorithm for locating all link critical points is derived:

$$\alpha_i = \begin{cases} 0 & \alpha_i \leq 0 \\ e_i^T p_i = \frac{1}{l_i}[X_{i+1} - X_i]^T [X_0 - X_i] & 0 < \alpha_i < l_i \\ l_i & \alpha \geq l_i \end{cases} \tag{4.16}$$

$$(X_c)_i = X_i + \alpha_i e_i. \tag{4.17}$$

The distance from the link i critical point to the center of the obstacle is then

$$(d_c)_i = \| (X_c)_i - X_0 \| . \tag{4.18}$$

Defining the critical distance $d_c = min\{ (d_c)_i \}$, the corresponding link critical point as the body critical point X_c, and noting that the critical distance is joint configuration dependent, we can express the criterion for obstacle avoidance as the following inequality constraint:

$$d_c - r_0 \equiv g(\theta) \geq 0. \tag{4.19}$$

Once the obstacle avoidance problem is formulated as the kinematic constraint equation (4.19), two possible cases need to be considered: (1) $g(\theta) > 0$; in this case, equation (4.19) is satisfied and the entire manipulator is outside the SOI. Therefore, the manipulator redundancy can be used to achieve other additional tasks. (2) $g(\theta) \leq 0$; in this case, the manipulator is inside the SOI of the obstacle. Here, the redundancy is used to achieve obstacle avoidance by inhibiting the motion of the body critical point in the direction toward the obstacle. To this end, the inequality constraint (4.19) is replaced by the equality constraint

$$g(\theta) = 0, \tag{4.20}$$

so that the extended Jacobian technique can be used to resolve the manipulator's redundancy. The end-effector configuration vector X is augmented to obtain Y as

$$Y = \begin{bmatrix} X \\ g \end{bmatrix}. \tag{4.21}$$

The desired trajectory of Y is

$$Y_d(t) = \begin{bmatrix} X_d(t) \\ 0 \end{bmatrix}, \tag{4.22}$$

The extended Jacobian J_{EX} is obtained as

$$J_{EX} = \begin{bmatrix} J \\ \frac{\partial g}{\partial \theta} \end{bmatrix} = \begin{bmatrix} J \\ \frac{\partial d_c}{\partial \theta} \end{bmatrix}, \tag{4.23}$$

where $\frac{\partial d_c}{\partial \theta}$ may be computed through direct differentiation of the elements in equation (4.18) as

$$\frac{\partial d_c}{\partial \theta} = \frac{1}{d_c}[X_c - X_0]^T \frac{\partial X_c}{\partial \theta} = \frac{1}{d_c}[X_c - X_0]^T J_{X_c} \tag{4.24}$$

where J_{X_c} is recognized as the Jacobian of the critical point X_c.

Now, we can work in GOKC framework using the results in equations (4.21) - (4.23). By the same approach as that shown in the previous sections, obstacle avoidance can be implemented by replacing the end-effector configuration vector X, the desired end-effector trajectory $X_d(t)$, and the Jacobian matrix J, with the augmented configuration vector Y, the desired trajectory Y_d, and the extended Jacobian matrix J_{EX}, respectively in the GOKC.

In the following simulation, for the ease of computation, the link critical points are assumed to be in the middle of each link. The simulation task is to avoid a moving obstacle while the end-effector remains motionless. The trajectory of the centre of the moving obstacle is given by

$$X_o(t) = \begin{bmatrix} 0.24 \\ 0.6 + 0.8t \\ 0.8 \end{bmatrix} \qquad 0 \le t \le 1. \tag{4.25}$$

The radius of the SOI is chosen to be 0.25m while the radius of the obstacle is assumed

0.2m. The obstacle would collide with link 4 if obstacle avoidance control was not employed. To avoid chattering at the SOI boundary, a simple digital filter is implemented in this simulation, the 7×7 matrix Q is chosen as $Q = diag\{10^3, 10^3, \ldots, Q_c\}$, where Q_c is a scalar weighting corresponding to the obstacle avoidance task, and is set to be

$$Q_c = \begin{cases} 0 & d_c > r_0 \\ 8000\left(\dfrac{d_c - O_r}{r_0 - O_r} - \dfrac{1}{2\pi}\sin\left(2\pi\dfrac{d_c - O_r}{r_0 - O_r}\right)\right) & O_r < d_c \le r_0 \end{cases} \tag{4.26}$$

where d_c is the critical distance, i.e., the distance between the body critical point and the center of the obstacle, r_0 is the radius of the SOI, and O_r is the radius of the obstacle. When $d_c > r_0$, Q_c is set to zero, thus the manipulator redundancy is utilized to achieve the minimum norm of joint velocities. If $d_c < r_0$, the redundancy is used to solve the obstacle avoidance problem. The other parameters in GOKC are chosen as $R = I_7$, and sampling period $\Delta t = 0.001s$. Figure 4.8 shows that $d_c > O_r = 2.0$ is satisfied. It also shows that when $d_c < r_0$ the GOKC drives the joints to avoid collision with the obstacle.

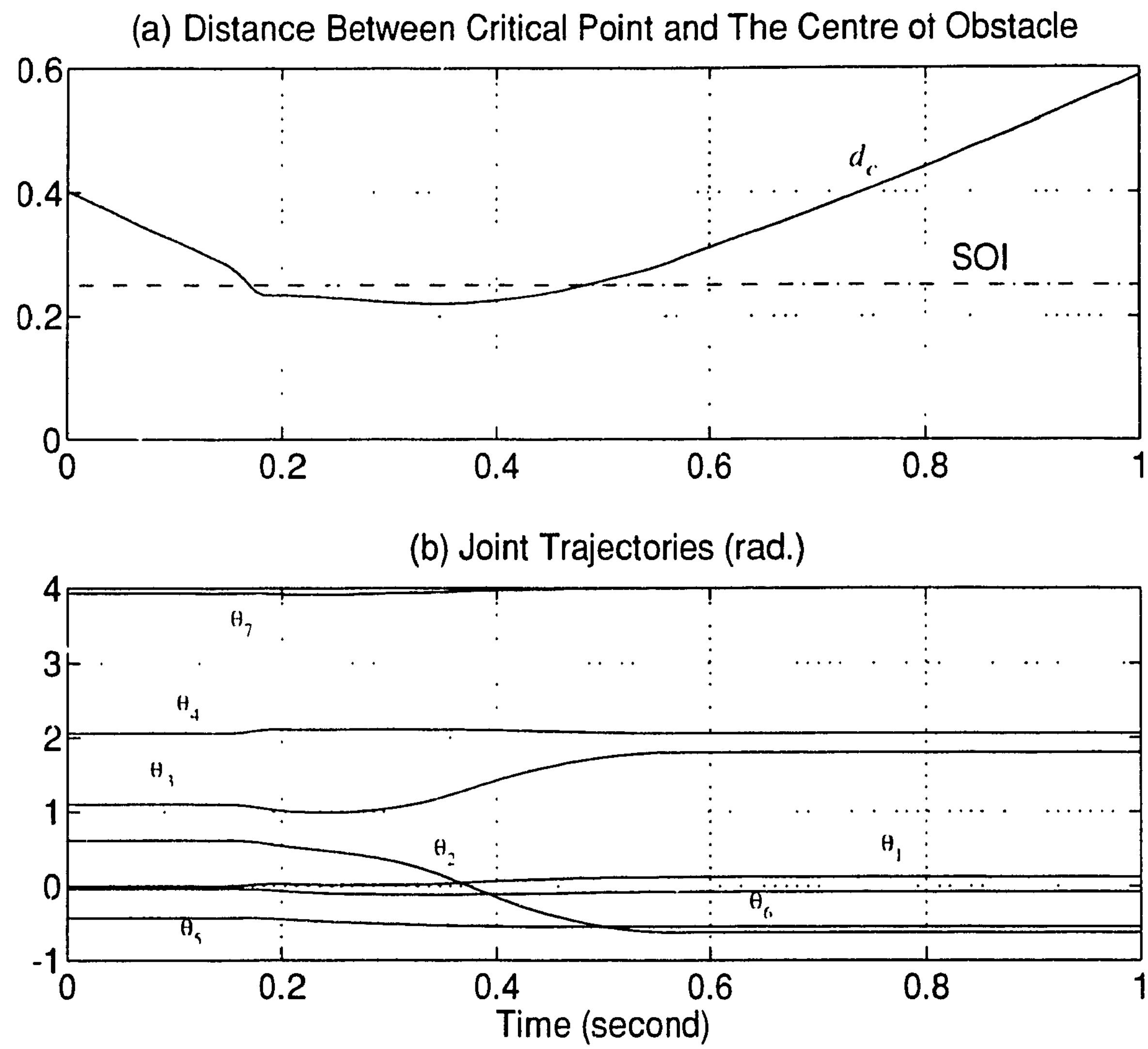

Fig. 4.8 Obstacle Avoidance Control using GOKC

4.6 POSTURE CONTROL

For a manipulator with r redundant joints, a phenomenon known as "self-motion" exists, which is a continuous internal motion of the manipulator joints that keeps the end-effector motionless. The problem of posture control is to use self-motion for the purpose of adjusting a manipulator's configuration while the end-effector is tracking a desired path. The main idea of the configuration control based method for posture control has already been introduced in Section 3.4.

In the following simulation study, the posture control task is to move joint 2 of REDIESTRO from its initial value $\theta_2(t_0)$ to the desired value $\theta_2(t_0) + \frac{\pi}{2}$, while the end-effector stays motionless. For this purpose, the additional kinematic constraint can be defined as

$$\phi(\theta) = \theta_2 , \tag{4.27}$$

and its desired trajectory is chosen as

$$\phi_d(t) = \theta_2(0) + \frac{\pi}{2}\left(t - \frac{1}{2\pi}\sin 2\pi t\right), \qquad 0 \le t \le 1 \tag{4.28}$$

The end-effector configuration variable X is augmented to Y as

$$Y = \begin{bmatrix} X \\ \phi \end{bmatrix}, \tag{4.29}$$

and the desired trajectory of Y is specified as

$$Y_d(t) = \begin{bmatrix} X_d(t) \\ \phi_d(t) \end{bmatrix} = \begin{bmatrix} X(0) \\ \phi_d(t) \end{bmatrix}. \tag{4.30}$$

The extended Jacobian matrix is given by

$$J_{EX} = \begin{bmatrix} J \\ \frac{\partial \phi}{\partial \theta} \end{bmatrix}, \tag{4.31}$$

where

$$\frac{\partial \phi}{\partial \theta} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{4.32}$$

By using Y, $Y_d(t)$ and J_{EX} in the GOKC, the posture control is achieved. The parameters of the GOKC are chosen as: $Q = diag\{10^4 \ldots 10^4\}$, $R = I_7$ and $\Delta t = 1$ ms. The simulation results show that the relative tracking error is less than 0.5%. Figure 4.9 gives the position trajectories of the end-effector and the position tracking errors and Figure 4.10 shows the trajectory of joint 2. It is clear that the posture control sub-task has been accomplished.

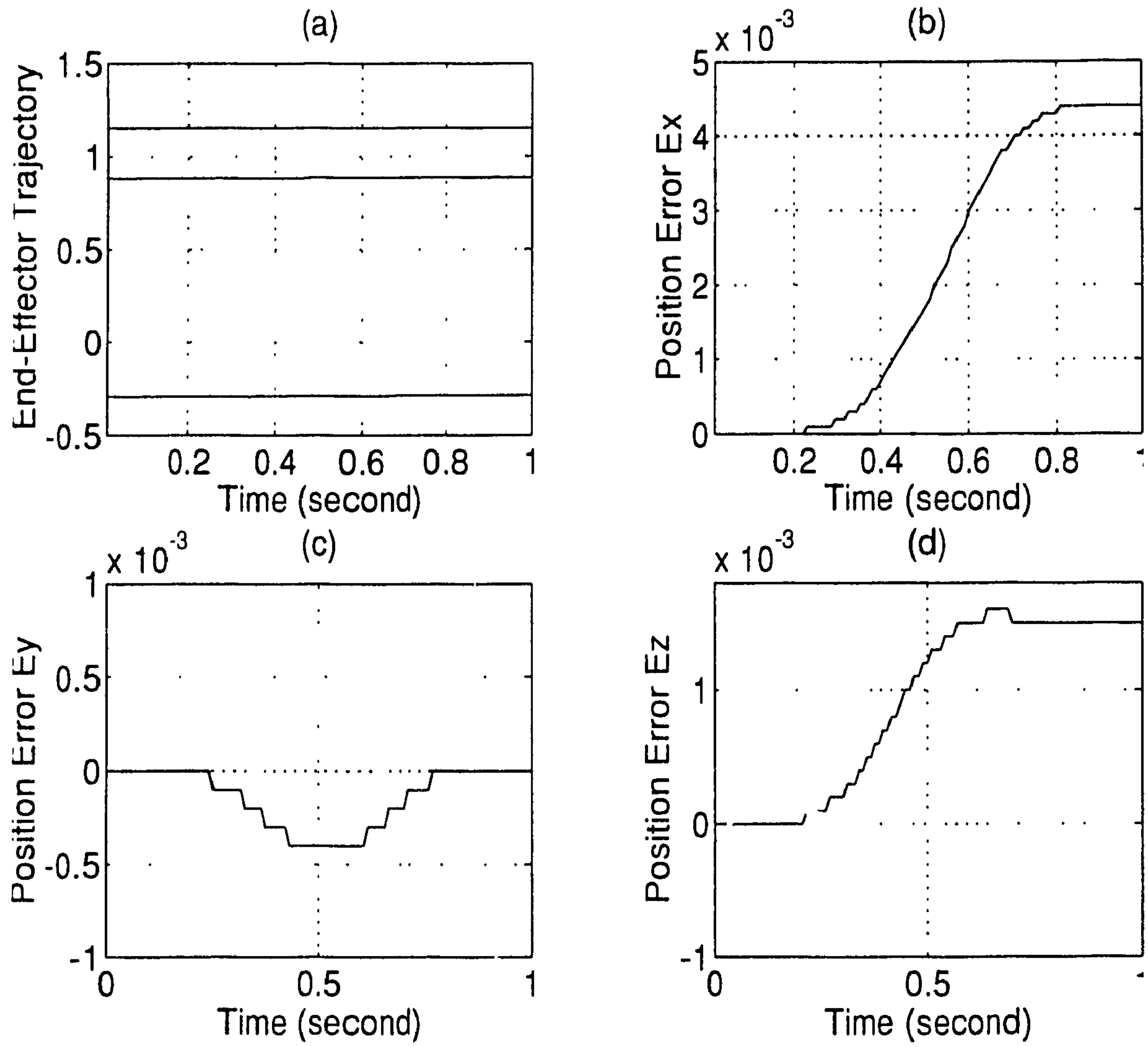

Fig. 4.9 Simulation Results for Posture Control using GOKC

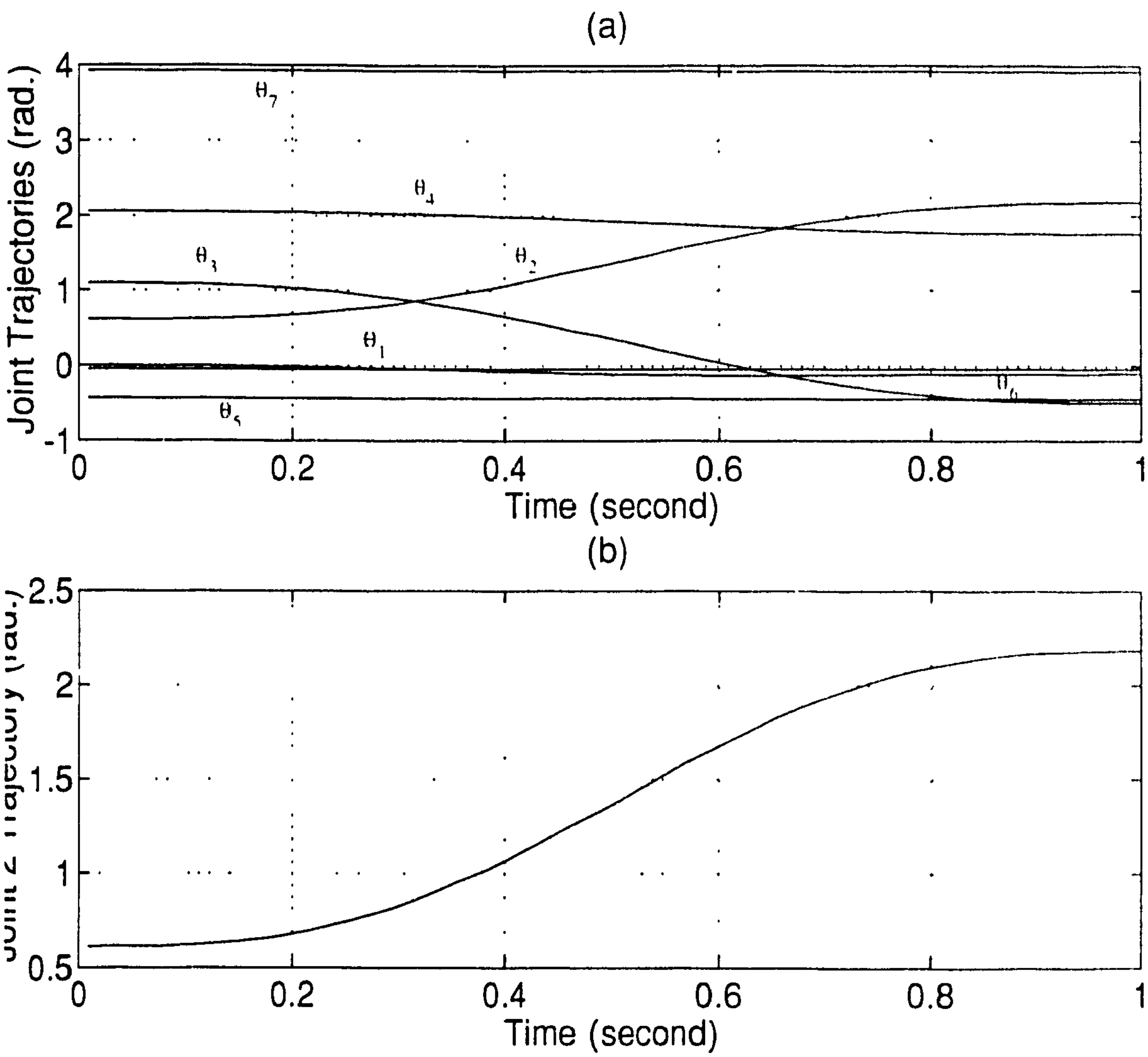

Fig. 4.10 Joint Trajectories in Posture Control using GOKC

4.7 REFERENCES

[1] P. M. Frank, *Introduction to System Sensitivity Theory*, Academic Press, New York, 1978.

[2] I.K. Salisbury and J.J. Craig, "Articulated Hands: Force and Kinematic Issues", *Int. J. Rob. Res.*, 1982, Vol. 1, No. 1. pp. 4 - 17.

[3] J.Angeles, "The Design of Isotropic Manipulator Architectures in the Presence of Redundancies", *Int. J. Rob. Res.*, 1992, Vol. 11, No. 3, pp 196-201.

[4] J. Angeles, F. Ranjbaran, and R. V. Patel, "On the Design of the Kinematic Structure of Seven-Axes Redundant Manipulators for Maximum Conditioning", *Proc. IEEE Int. Conf. Robotics and Automation*, 1992, pp. 494 -499.

[5] R.P. Paul, *Robot Manipulators: Mathematics, Programming, and Control*, MIT Press, Cambridge, MA,1981.

[6] H. Seraji and R. Colbaugh, "Improved Configuration Control for Redundant Robots", *J. Rob. Syst.*, vol.7, No.6, 1990, pp. 897-928.

[7] R. Colbaugh, H. Seraji, and K. Glass, "Obstacle Avoidance for Redundant Robots Using Configuratuion Control", *J. Rob. Syst.*, 1989, Vol. 6, No. 6, pp. 721-744.

CHAPTER 5 CONCLUSIONS AND FURTHER WORK

5.1 CONCLUSIONS

The main contribution of this thesis is a new approach, called globally optimal kinematic control (GOKC), for optimal redundancy resolution. The approach minimizes an integral quadratic cost function $V = \int_{t_0}^{t_f} [E^T Q E + \dot{\theta}^T R \dot{\theta}]\, dt$, where E is the tracking error vector and $\dot{\theta}$ is the joint velocity vector, such that the GOKC strategy achieves a trade-off between the tracking error E and magnitude of the joint velocities. The following are the main advantages of the approach:

(1) It is a global optimization method, which ensures conservative motion.

(2) It yields a singularity-robust implementation of optimal control, which can be used for both redundant as well as nonredundant manipulators.

(3) It does not require computation of the generalized inverse of the Jacobian matrix which is computationally expensive.

(4) It can be implemented in real time for a class of trajectories which occur in industrial application, e.g., for pick-and-place operations.

On the other hand, the GOKC scheme has disadvantage that it may cause the tracking errors in some situations, even when the manipulator is far from any singular configuration. In other words, the weighted kinetic energy $\dot{\theta}^T R \dot{\theta}$ can guarantee bounded joint velocities and smooth transitions through singular configurations, but the tracking performance at well-conditioned configurations may be unnecessarily compromised. This problem may be alleviated by using a configuration-dependent weighting matrix $R(\theta)$ instead of the constant matrix R. The matrix $R(\theta)$ should have a large value in the neighborhood of singularities, and a small value away from the singularities.

5.2 FURTHER WORK

There are some aspects of the GOKC which could be investigated further. The most important ones are as follows:

As indicated in the preceding discussion, a constant positive definite weighting matrix R can give rise to tracking errors in some situation, e.g. in singularity avoidance. One possible way to overcome this difficulty is to choose R as a function of the manipulability index such as $\sqrt{det[JJ^T]}$ where J is the Jacobian matrix. Another choice could be as a function of the condition number of the Jacobian matrix. Note that it is tempting to choose R as a function of the joint velocity $\dot{\theta}$, with R increasing with increasing $\dot{\theta}$. But this approach is not useful since the solution to the optimal problem (equation (3.3)) is obtained assuming that R is independent of $\dot{\theta}$.

The GOKC scheme ensures exponential stability of the redundant manipulator, so that when the final desired configuration $X_d(t_f)$ is constant, as is common, the steady-state tracking error vanishes, i.e. $X \rightarrow X_d$ as $t \rightarrow \infty$. But there is no a analytical expression or simple way to control the transient tracking error $E(t)$. Also, the relationship between the tracking error and the parameters of the GOKC scheme is not straightforward. Some existing results from optimal control theory may be help in this respect.

At present, the GOKC cannot be implemented in real-time for a general class of trajectories. Future work should be directed at enlarging the class of trajectories for which the GOKC strategy can be applied at real-time or, as a first step, in obtaining a sub-optimal (approximate) scheme that can be implemented in real-time.