A Unified Motion Planning Approach for Redundant and Non-Redundant Manipulators with Actuator Constraints

Ching Luan Chung

A DISSERTATION
SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN
MECHANICAL ENGINEERING

Department of Mechanical Engineering
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

December 1990

GRANT NAG3-811

FINAL REPORT

Abstract

The term trajectory-planning has been used to refer to the process of determining the time-history or "joint trajectory" of each joint variable corresponding to a specified trajectory of the end-effector of the manipulator. The trajectory-planning problem, in its original form, was solved as a purely kinematic problem. The drawback of this approach is that there is no guarantee that the actuators can deliver the effort necessary to track the planned trajectory. Furthermore, feedback-controller synthesis was addressed as a separate problem and without consideration of the actuator constraints. Later studies, which were concerned with the development of high-speed and high-precision manipulators did take actuator constraints into account but the control strategy used was primarily based on the classical open-loop optimal control approach. The performance of the robot manipulator resulting from the implementation of such an open-loop approach is extremely sensitive to uncertainty in the dynamic model and disturbances which may act on the manipulator. The addition of a feedback controller may not resolve this problem because the feedback control law is usually synthesized without taking the actuator constraints into account. To overcome these limitations, we have developed a motion planning approach which addresses the kinematics, dynamics and feedback-control of a manipulator in a unified-framework. Actuator constraints are taken into account explicitly and a-priori in the synthesis of the feedback control law. Therefore the result of applying the motion planning approach described in this thesis is not only the determination of the entire set of joint trajectories but also a complete specification of the feedback-control strategy which would yield these joint trajectories without violating actuator constraints.

The motion planning framework is developed in an optimization setting, which allows the analyst to (i) exploit any available freedom in the task specification of the manipulator, and (ii) exploit (kinematic) redundancy in the case of kinematically redundant manipulators. The effectiveness of the unified motion planning approach is demonstrated on two problems which are of practical interest in manipulator robotics. In the first problem feedback-controlled motions which minimize task time are planned for non-redundant manipulators. The second problem, which has useful applications in Space Robotics, addresses the use of kinematic redundancy in planning motions which minimize the magnitude of the reactions transmitted to the base of a manipulator.

Acknowledgement

I would like to thank my parents for their support and encouragement. I am grateful to my advisor, Professor Subhas Desa, for his advice and guidance during the course of my research.

I would like to thank the members of my thesis committee, Professor Dwight Baumann, Professor Tom Kurfess and Dr. Charles Lawrence, for providing constructive comments and suggestions related to the work.

I would like to acknowledge several useful discussions I had with my officemates Jin-Oh Kim, Harry Kim, and Dimitrios Apostolopoulos. Peggy Martin and Eunice Hench were a constant source of support and comfort during my stay at Carnegie Mellon.

Financial support from the NASA Lewis Research Center made this research possible.

Lastly, I would like to thank my friend, Monica Suk for her patience, understanding, and support.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

This thesis is about the development and application of a unified motion planning approach for robotic manipulators. In order to motivate the need for such a unified approach, it is useful to first briefly review existing motion-planning approaches. Traditionally, the term trajectory-planning has been used to refer to the process of determining the time-history or "joint trajectory" of each joint variable corresponding to a specified trajectory of the end-effector of the manipulator. The trajectory-planning problem, in its original form, was solved as a purely kinematic problem. The drawback of this approach is that there is no guarantee that the actuators can deliver the effort necessary to track the planned trajectory. Furthermore, feedback-controller synthesis was addressed as a separate problem and without consideration of the actuator constraints. Later studies, which were concerned with the development of high-speed and high-precision manipulators did take actuator constraints into account but the control strategy used was based on the classical open-loop optimal control approach or variations thereof. The performance of the robot manipulator resulting from the implementation of such an open-loop approach is extremely sensitive to uncertainty in the dynamic model and disturbances which may act on the manipulator. The addition of a feedback controller may not resolve this problem because the feedback control law is usually synthesized without taking the actuator constraints into account. To overcome these limitations, we have developed a motion planning approach which addresses the kinematics, dynamics and feedback-control of a manipulator in a unified-framework. Actuator constraints are taken into account explicitly and a-priori in the synthesis of the feedback control law.

Therefore the result of applying the motion planning approach described in this thesis is not only the determination of the entire set of joint trajectories but also a complete specification of the feedback-control strategy which would yield these joint trajectories without violating actuator constraints. Furthermore since the motion planning framework is developed in an optimization setting, one can plan motions and synthesize control laws which are optimal in some useful sense.

1.2 Contributions of the Research

The primary contributions of this research are the development, implementation and application of a unified motion planning approach for redundant and non-redundant manipulators. More specifically the contributions of this research are as follows:

1. The unified motion planning approach simultaneously plans the manipulator trajectory and synthesizes a feedback control law which does not violate actuator constraints.
2. Multi-criterion optimization is used as an integral part of the framework to plan trajectories which optimize dynamic performance.
3. The incorporation of optimization in the motion planning approach allows the analyst to (i) exploit any available freedom in the task specification of the manipulator, and (ii) exploit (kinematic) redundancy in the case of kinematically redundant manipulators.
4. The unified motion planning approach, by avoiding the drawbacks of most commonly used motion planning approaches, allows one to plan realizable motions in a relatively straightforward manner.

1.3 Overview of the Contents

The thesis is organized as follows. In Chapter 2 we state the goals of motion planning and conclude with a formal definition of motion planning. The unified motion planning approach, which underlies the present research, is developed in Chapter 3. Existing methods for trajectory planning and controller synthesis are first surveyed and then used to motivate the proposed motion planning approach. The basic building blocks in the unified approach are then described. The final section of Chapter 3 describes the proposed motion planning approach which integrates trajectory planning and feedback controller synthesis to plan feasible and optimal manipulator motions.

Chapter 4 deals with the application of the unified motion planning approach to non-redundant manipulators. After discussing the specifications of end-effector tasks and the

parameterization of end-effector trajectories, a procedure is developed for planning feasible and optimal motions for non-redundant manipulators. The effectiveness of the unified motion planning approach is then demonstrated by using it to plan a feedback-controlled motion which minimizes the task time.

Chapter 5 is the redundant manipulator counterpart of Chapter 4. Special attention is paid to the proper resolution and parametric representation of kinematic redundancy. As a demonstration of the usefulness of the unified motion planning approach we address the problem of minimizing the magnitude of the reactions transmitted to the base of a manipulator with one excess (or redundant) degree of freedom; the base-reaction minimization problem has useful applications to manipulators operating in "zero-gravity" environments in space.

An issue which must be confronted in using redundant manipulators is whether they are really effective in improving dynamic peformance. This issue is important in its own right and is studied in Chapter 6. In this chapter we also discuss certain issues which must be considered in the implementation of the unified motion planning approach. Finally, in Chapter 7 we summarize the work described in this thesis, draw some conclusions from the investigation and make some suggestions for future research.

Chapter 2

Definition of the Motion Planning Problem

2.1 Overview

In robotic manipulator work, *trajectory planning* refers to the process of obtaining the joint trajectories corresponding to a given task specification for the end-effector. We are interested in planning motions for high-performance manipulators, i.e. manipulators which must follow prescribed trajectories at very high speed and with very high accuracy. In order to ensure that the motion plan is robust, i.e. insensitive to uncertainties in the model and disturbances which might act on the manipulator, the manipulator must be feedback-controlled. Therefore in the present work we use the term motion planning to describe the planning of the optimal joint trajectories and also the determination of the optimal gains in a prescribed feedback control strategy. The underlying philosophy of the present work is that proper motion planning should simultaneously address both trajectory planning and controller synthesis.

In our motion planning problem, we make the assumption that the structure of the manipulator and the structure of the controller are known *a priori*. The controller parameter vector, denoted by $\boldsymbol{P}_c$, consists of all the unknown parameters or "gains" of the controller strategy. For the present, it is convenient to regard motion planning very simply as the determination of the combination of the optimal manipulator trajectory and the optimal controller parameter vector $\boldsymbol{P}_c$.

The rest of the chapter is devoted to a more detailed discussion of motion-planning culminating with a formal statement of motion- planning in Section 2.4.

2.2 Primary Goal of Motion Planning

Before we state the primary goal of motion planning, we will first need to define task specifications, end-effector trajectory, and joint trajectory.

Task specifications are high-level descriptions of the desired end-effector motion ($x_d(t)$). For example, if the task is for the end-effector of the manipulator to pick up an object at point A and place it at point B, then the task specifications would be the positions of points A and B in a coordinate system $\{U\}$ which is fixed to the base of the manipulator.

If x_1, x_2, and x_3 denote the Cartesian coordinates of the reference point on the end-effector in the coordinate system $\{U\}$ fixed to the base of the manipualtor, then

$$x=[x_1,x_2,x_3]^T. \tag{2.1}$$

is the vector which denotes the position of the end-effector and $\mathbf{x}(\cdot)$ is called the end-effector trajectory or task-space trajectory. For a planar manipulator, $x_3 = 0$.

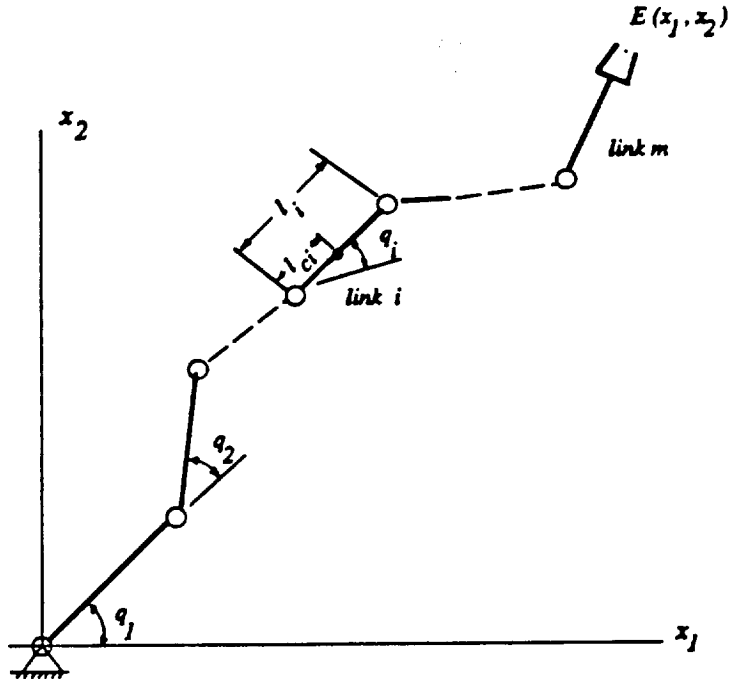


Fig. 2.1 A Planar m Degree-of-Freedom Manipulator

Consider an m degree-of-freedom manipulator with m revolute joints as shown in Fig. 2.1. Let q_i denote the joint variable at joint i. For an m degrees-of-freedom manipulator, the joint variable vector $q \in R^m$ can be defined as

$$q=[q_1,q_2,,\ldots,q_m]^T. \tag{2.2}$$

The time-history q_i, $(i=1,2,\ldots m)$, of each joint variable is called the joint trajectory. The vector of joint trajectories for all the joints is called the joint trajectory vector and is denoted by $q(\cdot)$. (In the sequel $q(\cdot)$ will be simply referred to as the joint trajectory rather than the joint trajectory vector.)

Before we define the primary goal of the manipulator, it is important that we identify the freedom provided by the task specifications and/or by the kinematic redundancy. In the following discussion we will illustrate the freedom that one can exploit in motion planning problems.

If the number of degrees of freedom (m) is greater than the minimum number of degrees of freedom (n) required to perform a task, then the manipulator is called a kinematically redundant manipulator with $\rho=m-n$ degrees of kinematic redundancy. When ρ is equal to zero, the manipulator is a non-redundant manipulator. The kinematic relationship between the end-effector position x and the joint variable vector q can be described by a nonlinear mapping called the forward kinematic mapping $\psi : R^m \rightarrow R^n$ which is expressed as follows:

$$\psi : (q_1,q_2,\ldots,q_m) \rightarrow (x_1,\ldots,x_n). \tag{2.3}$$

The "end-effector" velocity $\dot{x}$ is related to the joint velocity vector $\dot{q}$ by the following well known linear relationship [10]:

$$\dot{x}=J\dot{q}, \tag{2.4}$$

where $J=\frac{\partial \psi(q)}{\partial q} \in R^{n \times m}$ is the so-called *Manipulator Jacobian* matrix.

Using Eq. (2.4), we obtain the following linear relationship between the desired joint velocity $\dot{q}_d$ and the desired end-effector velocity $\dot{x}_d$:

$$\dot{x}_d = J\dot{q}_d. \tag{2.5}$$

For non-redundant manipulators, J is square and, in general, invertible and $\dot{q}_d$ can be uniquely obtained by premultiplying $\dot{x}_d$ by J^{-1}. Therefore, it is clear that the freedom that one can exploit in trajectory planning comes only from any available freedom in the task specification of the end-effector trajectory x_d.

However, in the case of a redundant manipulator, since $m>n$, there are in general an infinite number ($_{\infty}^{\rho}$) of joint velocity vectors $\dot{q}_d$ that satisfy Eq. (2.5). This implies that there are an infinite number of joint trajectories that can be used to achieve the task regardless of whether there is any freedom in the task specifications.

In reality, not all of the desired joint trajectories $q_d(\cdot)$ computed from Eq. (2.5) are realizable due to the fact that some of the trajectories would require torques which exceed the capabilities of the actuators. The purpose of motion planning is to determine an appropriate joint trajectory vector q_d and the paramters in the controller, such that, under the actuator constraints the primary goal of the task, which is stated below, is achieved.

*The **primary goal** of motion planning is to exploit the freedom in the end-effector trajectory, the joint-space trajectory, and the choice of the magnitudes of the controller parameters to plan a trajectory for which the associated actuator inputs do not exceed their bounds.*

2.3 Secondary Goal of Motion Planning

In addition to satisfying the primary goal of motion planning, we are also interested in exploiting the freedom in both the task specifications and the feedback control law to achieve an additional dynamic performance objective.

If I denotes the scalar which is a measure of dynamic performance, then I can be expressed in the well-known general form

$$I = \int_0^{t_f} h(q,\dot{q},\ddot{q},t)\,dt \; + \; \Phi(q(t_f),\dot{q}(t_f),\ddot{q}(t_f)). \tag{2.6}$$

where h and Φ denote functions defined by the analyst and t_f denotes the final time.

In certain applications where we are only concerned with the peak value of a function h, we can define I as the maximum value of a function $h(\cdot)$ over a time interval:

$$I = max \ \{h(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})\}. \tag{2.7}$$

As will be seen in the sequel, the general performance index defined in Eq. (2.6) can be used as a measure of dynamic performance in several problems of practical interest, for example, the base reaction minimization problem and the minimum-time problem.

The secondary goal of motion planning is to plan a trajectory which (also) minimizes the performance index (2.6) or (2.7).

2.4 Motion Planning

Having defined the primary goal and secondary goal of our motion planning approach, we are in a position to formally state the motion planning problem for a feedback-controlled manipulator:

Determine the desired joint trajectory $\mathbf{q}_d$ *and controller parameter vector* $\mathbf{P}_c$ *such that the actual end-effector trajectory* $\mathbf{x}(t)$

(i) satisfies the task specifications;

(ii) is robust (i.e. achieved by a feedback control strategy);

(iii) does not violate actuator constraints;

(iv) optimizes an additional measure of dynamic performance.

The first three motion requirements are related to the satisfaction of the primary goal of motion planning while the last motion requirement is used to achieve the secondary goal.

Having defined the motion-planning problem, the next step, of course, is to describe the approach that we have developed for obtaining a good motion plan. This is the subject of the next chapter.

Chapter 3

Unified Motion Planning Approach

3.1 Introduction

In this chapter, we will motivate the need for the unified motion planning approach and also give an overview of this approach. In Section 3.2, we point out the drawbacks of the conventional motion planning approaches and identify some of the research issues that we have to resolve. In Section 3.3, we survey relevant research in the areas of motion planning and comment on the strength and limitations of these works. Finally in Section 3.4 we describe the Unified Motion Planning Approach and the building blocks which are essential to its formulation.

3.2 Why do we need another motion planning approach?

There have been many studies addressing different aspects of motion planning and controller design for robotic manipulators. Some of these studies were devoted to solving purely kinematic problems, for example, determining inverse kinematic solutions for manipulators of different kinematic structures [1,7,10] or developing useful representations for joint-space trajectories such as the 4-3-4 trajectory and the five-cubic spline trajectory [10,15]. There are also numerous studies which address the dynamics of a manipulator, for example, Luh-Walker-Paul's algorithm [1] for computing the inverse dynamics of a manipulator and Hollerbach's recursive Lagrangian formulation of dynamic equations [1,15]. There are also numerous studies addressing the design of feedback controllers for manipulators [1,10,12,15].

Since there are so many studies in the area of motion planning, a natural question one might ask is why we need another motion planning approach.

To answer this question, we will first look at three classes of motion planning approaches proposed in the literature. The purpose of the following discussion is to present a general overview of the conventional approaches, to point out the drawback of these approaches and to thereby motivate some of the research issues addressed in this study. In Section 3.3, we will present a survey of the conventional approaches.

The first class of motion planning approaches is the *purely kinematic approach* [7,14,16,26,37]. The objective of these approaches is to determine a desired joint trajectory vector q_d for a given task specification. To guarantee that the planned motions do not require torques which exceed the bounds on actuator efforts some researchers have imposed kinematic constraints such as speed limits and acceleration limits on the allowable solutions. However, these kinematic constraints are not derived from the equations of motions of a manipulator and therefore the major drawback of this approach is that even though the kinematic constraints are satisfied, the trajectory may still require torques that are in excess of what the actuator can deliver.

To take actuator constraints into account, some researchers proposed the *constrained open-loop approach* for non-redundant manipulators [6,19,39]. In these approaches, the open-loop torque vector, u^*, which is required to achieve the desired end-effector trajectory is computed and used as the "control" input. The major drawback of this approach is that it is open-loop and therefore non-robust, i.e. the performance of the manipulator is sensitive to uncertainties in the model and disturbances.

To remedy the robustness problem in the above approach, an obvious solution is to simply add a feedback controller to the constrained open-loop approach to make it robust [12]. Based on this rationale, the controller effort vector u would consist of two parts: u^* and $\tilde{u}$. The first part of the controller effort, u^*, is computed based on the constrained open-loop approach. The second part of the controller effort, $\tilde{u}$ comes from the feedback control strategy which is usually designed without considering the actuator constraints. The major drawback of this approach is that it is unclear how one would distribute the actuator constraint vector u_{max} between u^* and $\tilde{u}$ such that $u = u^* + \tilde{u}$ is always less than the actuator constraint vector, u_{max}.

There is therefore a need for a motion planning approach which plans a feedback

controlled trajectory which does not violate the actuator constraints. The approach developed in this thesis uses a *feedback* control strategy which *a-priori* takes actuator constraints into account, thereby ensuring that the magnitudes of the actuator torques based on the control law do not exceed the (specified) bounds. In addition, our approach allows us to exploit any freedoms available in the manipulator task specifications and the freedoms available in the magnitudes of the controller parameters to simultaneously plan a trajectory and synthesize a feedback control law to optimize dynamic performance.

3.3 Survey of Related Research

Having addressed the need for a more complete motion planning approach, we will now survey some of the related research, specifically in the areas of trajectory planning, controller design and dynamic performance optimization of redundant and non-redundant manipulators. The purpose of this survey is to explore the strength and shortcomings of the existing studies and highlight some of the tools that we will use from these studies for the development of the unified motion planning approach.

3.3.1 Trajectory Planning of Manipulators

In this section we review studies that only address the kinematics of a manipulator[1]. A standard problem is the determination of the joint-trajectory for a completely or partially specified end-effector trajectory. We will discuss this problem separately for non-redundant and redundant manipulators.

3.3.1.1 Non-redundant Manipulators

Many researchers have investigated approaches for planning straight-line end-effector trajectories. Paul [31] proposed an approach that breaks the end-effector trajectory into a number of straight-line segments. The points where these segments meet are called the cartesian knot points. These cartesian knot points are then mapped into corresponding joint-space configurations. A quadratic polynomial is then used to connect these joint configurations to form a smooth joint trajectory.

Fu et al. [15] presented various joint-space trajectory interpolations such as the 4-3-4

[1]In Section 3.3.2, we will survey trajectory planning studies which consider the dynamics of a manipulator.

joint trajectory, 3-5-3 cubic joint trajectory and 5-cubic spline trajectory. These representations enable one to obtain smooth joint trajectories for pick and place operations.

Lin et al. [24] formulated an off-line approach for constructing a cubic-spline polynomial joint trajectory to fit selected cartesian knot points. They developed an algorithm that minimizes the total traveling time of the manipulator by varying the time intervals of the cubic polynomials. In their studies, they imposed kinematic constraints on the joint velocity and joint acceleration. One should note that in this study, the kinematic constraints are assigned arbitrarily instead of being derived from the equations of motion and the actuator constraints.

In all the above studies, there is no guarantee that the joint trajectory obtained from the trajectory planning algorithm can be executed successfully with or without feedback as there is no consideration of the dynamics of the manipulator and the actuator constraints in the analysis.

3.3.1.2 Redundant Manipulators

In the majority of the redundant manipulator studies, the problem of interest is to exploit the freedom available in the joint trajectory to achieve an additional task while the end-effector performs the primary task of tracking a prescribed end-effector trajectory.

An important research issue in the kinematics of redundant manipulators is referred to in the literature as redundancy resolution. Redundancy resolution refers to the process of selecting a joint-space solution from the ${}_{\infty}\rho$ possible jont-space solutions for a redundant manipulator with ρ degrees of redundancy.

To represent kinematic redundancy, there are two common approaches - the pseudo-inverse and the partitioned Jacobian - which are derived from the "velocity" relationship (Eq. (2.4)) which relates *joint-velocity* and *end-effector velocity*. The pseudo-inverse approach has been presented and discussed extensively in the literature. Therefore, for a more in-depth treatment of the subject of pseudo-inverse representation, we will refer the reader to [20] which provides a good review on this topic. The partitioned-Jacobian approach is relatively less popular and has only been used in recent years. In Section 5.2.1, we provide a detailed description of the partitioned Jacobian approach. Other

approaches for representing kinematic redundancy are given in [3,7,35,36,37,42]. However, these approaches are not general and therefore of limited usefulness.

Depending on the way in which kinematic redundancy is resolved, the resulting approaches for utilizing of the freedom in the joint trajectory are quite different. Therefore, we will divide the studies in redundant manipulators into three classes - pseudo-inverse based approaches, partitioned-Jacobian based approaches, and other types of representations.

A. Pseudo-inverse Approach

Liegeois [23] was one of the first researchers that studied the trajectory planning problem for redundant manipulators. He developed a formulation, based on the pseudo-inverse of the Jacobian, to avoid the joint limits. He proposed the following velocity equation that can be numerically integrated to obtain the joint trajectories q:

$$\dot{q}=J^{+}\dot{x}+\alpha\nabla H, \tag{3.1}$$

where J^{+} is the pseudo-inverse of the Jacobian and ∇H is the gradient of a smooth function $H(q)$ that characterizes a secondary goal such as joint limit avoidance. $H(q)$ takes the form of:

$$H(q)=\frac{1}{6}\sum_{i=1}^{6}\{\frac{q_i-a_i}{a_i-q_{iU}}\}^2, \quad (i=1,2,,\ldots,m) \tag{3.2}$$

where q_{iU} and q_{iL} are the upper and lower limits of q_i and a_i is given by the following equation:

$$a_i=(q_{iU}+q_{iL})/2, \quad (i=1,2,\ldots,m). \tag{3.3}$$

Yoshikawa [44] presented a similar but more general approach to avoid joint limits, avoid obstacles and increase the manipulatability[2] (sic) of a manipulator. He also experimentally verified his algorithm by implementing it on the 7 d.o.f. Ujibot to avoid

[2]The measure of manipulatability equals zero when a manipulator is at singular state and increases as the manipulator is moving away from the singular configurations [44].

an obstacle. However, the obstacle avoidance scheme would only work in an environment with one obstacle.

Maciejewski [26] proposed an approach for determining the joint trajectories that avoid moving obstacles. In his approach, if a particular link is close to an obstacle, the nullspace solution for the joint velocity equation is then utilized to move the link away from the obstacle based on a desired velocity vector which is normal to the obstacle.

B. Partitioned Jacobian Approach

In this section, we will discuss those trajectory planning approaches that use the partitioned Jacobian approach. These studies, like those discussed above, do not consider the actuator constraints nor the dynamics of a manipulator.

Fenton [14] first introduced the generalized inverse approach which partitions the Jacobian matrix into two submatrices - a non-redundant Jacobian matrix J_{nr} and a redundant Jacobian matrix J_r. The partitioning of the Jacobian matrix is made possible by observing the fact that some of the joint variables can be treated as independent free variables that can be utilized, for example, in optimizing dynamic performance. Chung et al. [8] applied this approach to minimize the magnitude of the reactions transmitted to the base of a manipulator (for more details, see Sections 3.3.2); to differentiate this representation from the pseudo-inverse representation[3], they renamed this approach the *partitioned-Jacobian representation* for kinematic redundancy. The approach developed by Ghosal [16] is essentialy the same as the partitioned Jacobian representation.

Other studies that utilize the partitioned-Jacobian representation are discussed in Section 3.3.2.

C. Other Techniques

In this section, we discuss approaches that are not based either on the pseudo-inverse matrix or the partitioned-Jacobian approaches.

Sciavicco et al. [37] presented an approach for the inverse kinematic problem of

---

[3]The pseudo-inverse is sometimes called the generalized inverse [20,33].

redundant manipulators with joint limits in a workspace containing obstacles. Treating the inverse kinematic problem as a closed-loop control problem, he was able to generate the desired joint trajectory for a prescribed end-effector trajectory. The controller gains of the "closed-loop algorithm" are computed from an appropriate Lyapunov function. One of the drawbacks of this approach is that the accuracy of the resulting open-loop end-effector trajectory x_d is dependent on the convergence rate of the Lyapunov function.

Seraji [36] developed an approach which augments the forward kinematics with some task-related kinematic functions. One such kinematic function specifies a desired arm posture which might be important when the motion is constrained due to workspace obstacles. The number of forward kinematics equations are augmented by such kinematic functions until the total number of equations is equal to the number of joint variables. The joint-space trajectories can then be obtained as in the non-redundant case. The drawback of this approach is that only a limited number of problems can be solved due to fact that the user-defined kinematic functions are functions of the manipulator configuration q only.

Besides the above approaches, many researchers have developed approaches based on various mathematical techniques such as dynamic programming [42] and graph-search techniques [3]. However, these approaches are not popular due to large memory requirements and intensive computational requirements.

3.3.2 Dynamic Performance Optimization

In this section, we review relevant research where trajectory planning is based on optimizing the dynamic performance of a manipulator. This class of problems differs from the trajectory planning problems that we discussed in Section 3.3.1 where the dynamics of the manipulator is not considered in the analysis.

We will first discuss a study in non-redundant manipulators. In trajectory planning for non-redundant manipulators, the only freedom one can exploit is in the end-effector trajectory. Schmitt el al. [38] developed a global approach that determines the optimal joint trajectory for an *unconstrained, open-loop* non-redundant manipulator. The dynamic performance considered in this problem is to minimize the energy consumed during the execution of a pick-and-place task. Instead of minimizing the energy directly,

they posed a minimization problem with a cost function which is a sum of the magnitudes of the joint torques. In their approach, the Raleigh-Ritz method is used to approximate the unknown optimal joint motions with a finite number of weighted shape functions. The necessary conditions for the optimal solutions are then obtained by equating the partial-derivative of the cost function with respect to each of the weights to zero. The necessary conditions are then described by a set of non-linear algebraic equations which can be solved by standard numerical techniques.

In redundant manipulator studies, there have been a few approaches proposed for exploiting the freedom in the joint trajectory to achieve optimal dynamic performance. Most of the studies in this area are essentially open-loop. Suh and Hollerbach [18] proposed a local pseudo-inverse based approach for minimizing the magnitudes of joint torques. The results of their study show that for most cases their approach produces acceptable solutions. However, for trajectories where the end-effector velocity is very high, the local approach produces very high torques. In their later study [40], they proposed a global approach that optimizes a cost function which is an integral of the magnitudes of the joint torques. They also compared the solutions of the local approach and global approach. In their findings, the global approach out-performed the local approach for all the test cases. But the drawback of the global approach is that the formulation is very complicated and requires the user to solve a two-point boundary value problem for a set of fourth order ordinary differential equations. In view of this difficulty, Hirose and Ma [17] proposed a local approach (based on the partitioned Jacobian representation) that places limits on the joint accelerations. For most test cases, this approach is able to overcome the high torque problem encountered by Hollerbach and Suh. However, the approach does not take actuator constraints into account.

The common theme of all the above redundant manipulator studies is to minimize the magnitudes of the joint torques. Several other studies were also conducted to exploit kinematic redundancy in minimizing the magnitude of the base reactions, an important issue in microgravity robotic operations. deSilva et al. [11] first developed a local approach to minimize a cost function which is a sum of the weighted magnitudes of the base force and the base moment. The kinematic redundancy is resolved using the partitioned Jacobian approach : the joint variables are divided into a vector of redundant

joint variables q_r and a vector of non-redundant joint variables q_{nr}. The joint trajectory is then broken into a number of segments. For each segment, each joint trajectory is approximated by a polynomial with unknown coefficients. These coefficients are then determined by optimizing a static cost function which is a measure of the magnitude of the base reactions at the end of each time segment. The shortcoming of this local approach is that it results in a base force which has relatively large peak magnitude. Quinn and Chen [32] used the local approach developed by deSilva to study manipulators with up to three degrees of kinematic redundancy. Their results showed that by incorporating more degrees of kinematic redundancy the base reactions can be further reduced. Chung and Desa [9] compared the above local approach with a global approach in which the redundant joint variables are approximated by three segments of third or fourth order polynomials. The results of their study showed that with a relatively small number of coefficients, the global approach can reduce the peak magnitude of the base force observed in the local approach, but is not effective in reducing the magnitude of the base moment. This is due to the fact that a relatively small number of coefficients are used to parameterize the overall behavior of the joint trajectory.

It is important to point out that all the above approaches do not take actuator constraints into account and the dynamic performance optimized is for a manipulator without feedback control.

3.3.3 Controller Synthesis

We will briefly survey some of the more commonly used techniques for controller synthesis, point out their drawbacks and finally present an overview of the control synthesis approach used in this thesis.

One common controller synthesis method which we will call the *local linearization control approach* [12] designs a closed-loop control law for a linearized dynamic system by using linear optimal control theory.

According to this scheme, the control input vector u to the manipulator can be separated into two parts: the nominal control input vector u^* and the (corrective) feedback control input vector $\tilde{u}$. The control input u is therefore given by

$$\boldsymbol{u} = \boldsymbol{u}^* + \tilde{\boldsymbol{u}}. \tag{3.4}$$

The nominal input vector $\boldsymbol{u}^*$ can be obtained from the nominal model of the manipulator (plant) and the nominal trajectory. There are several studies that address the problem of obtaining the optimal nominal input vector $\boldsymbol{u}^*$ for a manipulator based on a dynamic performance criteria. This class of problems, usually called the optimal control problem for a manipulator, is posed as an open-loop, constrained nonlinear optimization problem.

Bobrow et al. [6] studied the time-optimal control problem of a non-redundant manipulator with the end-effector trajectory specified in space but not in time. The path of the manipulator is first parameterized in terms of a variable called x (distance along the path). Then, by imposing the torque constraints, the upper limit and lower limit of the acceleration $\ddot{x}$ can be obtained as functions of x and $\dot{x}$. The optimal acceleration profile is obtained from the switching curves in the $\dot{x}$ - x phase plane. The optimal acceleration profile is the profile that produces the largest velocity profile possible in the x-$\dot{x}$ plane. Once the optimal acceleration profile is obtained, the actual torque profile for each joint can then be obtained. Independently, Shin and McKay [39] have also developed a similar algorithm based on the phase-plane concept to solve the minimum-time control problem for a non-redundant manipulator. The results obtained by Dubowsky and Shin specify the nominal torque vector $\boldsymbol{u}^*$ that can be used to perform a task in a minimum-time manner for a manipulator. Unfortunately, this approach is quite complicated and can only solve a small class of problems (such as the minimum-time problem).

The feedback control input $\tilde{\boldsymbol{u}}$ is given, for example, by a state-feedback control law:

$$\tilde{\boldsymbol{u}} = -\boldsymbol{K}\tilde{\boldsymbol{x}}, \tag{3.5}$$

where $\boldsymbol{K}$ is the gain matrix and $\tilde{\boldsymbol{x}}$ is the state variable vector of the linearized system which is obtained by linearizing the nonlinear system about the nominal trajectory [12]. The gain matrix $\boldsymbol{K}$ can be obtained by using classical linear optimal regulator (LQR) theory [22].

The shortcoming of the local linearization control scheme is that the actuator

constraints cannot be taken into account directly in the design of a closed-loop controller. Therefore, the control input vector $\boldsymbol{u}$ obtained by Eq. (3.4) may violate the actuator constraints. This means that the actuators would not be able to supply the control efforts which are necessary to obtain the desired performance; as a result the actual performance could be very unsatisfactory.

The other approach which has also attracted a lot of attention is *global feedback linearization* [10] which also partitions the control effort into two parts. One part of the control effort is used to "cancel" the nonlinear terms so that the system is "feedback linearized" and decoupled. The other part of the control effort is error driven which ensures tracking in the face of disturbances and modeling errors. The major drawback of this approach is that since actuator constraints are not explicitly taken into account in the synthesis of the control law, the control effort computed using the above scheme will in general violate the actuator constraints resulting in unsatisfactory performance.

Traditional optimal control theory [22] deals either with feedback control strategies for unconstrained linear systems, as in the classical Linear Optimal Regulator problem, or with "open-loop" control strategies for constrained nonlinear systems as in the classical minimum-time control problem. In this thesis we are interested in developing feedback control strategies for constrained nonlinear systems and therefore we use the extension of optimal control theory proposed by Beyers and Desa [5]. The first step in this approach is to define a feedback control law which explicitly and a priori takes actuator constraints into account. The second step is to obtain the combination of the optimal contoller and plant parameters that optimizes a user defined performance index through the use of optimization techniques. Beyers experimentally verified the above synthesis approach by comparing the simulation results with the actual experimental results for the control of a two-degree-of-freedom robot manipulator. The experimental results were within 10% of the simulation results.

Having surveyed the related literature in the area of motion planning for non-redundant and redundant manipulators, we are ready to present the basic building blocks and concepts of the unified motion planning approach.

3.4 Unified Motion Planning Approach

In the previous section, we have seen that many studies in the areas of the kinematics of non-redundant and redundant manipulators can provide us with useful tools for trajectory planning. However, using the purely kinematic tools developed in these studies will not guarantee a trajectory that can be executed by a manipulator without violating the actuator constraints. The studies in the dynamic performance optimization area provide us a model-based solution which is non-robust in the face of disturbances and modeling errors. Very few studies in feedback control system synthesis are for constrained, feedback-controlled, nonlinear systems. As we have mentioned earlier, both the local linearization control approach and the global feedback linearization approach have difficulty in ensuring that the control efforts do not violate the actuator constraints.

As we will demonstrate in Section 3.4.2, the unified motion planning approach proposed in this study would overcome the drawbacks that we have just mentioned. In addition, a major advantage of the unified motion planning approach is that it allows us to simultaneously design the optimal motion and the optimal control law in one single framework. In this section, the research issues which arise in motion planning are used to motivate the development of the various building blocks in the unified (motion planning) approach.

3.4.1 Building Blocks

The building blocks or elements of the motion-planning approach are as follows:

A. Parameterization of the desired end-effector trajectory

B. Parameterization of the desired joint trajectories

C. Closed-loop, constrained controller synthesis

D. Multi-criterion optimization technique.

A. Parameterization of Desired End-Effector Trajectory

Parameterization enables us to convert the dynamic optimization problem of minimizing the performance index Eqs. (2.6) or (2.7) into a static optimization problem. In order to parameterize the desired end-effector trajectory systematically, it is necessary to address the following issues:

1. The classification of end-effector trajectories for a general task specification.
2. A simple parameterization approach for different classes of end-effector trajectories.

In the literature, there is no good way of classifying the end-effector trajectory. In this study, we classify the end-effector trajectory x into three categories according to the freedom provided by the task specification. The reason for this classification is that it allows us to identify the freedom in the end-effector motion and it also facilitates the development of a simple parameterization scheme. Based on the classification of the end-effector trajectory, we can identify the independent or free variables that we can use for parameterization. From the task specifications, we can then determine the boundary conditions for the free variables. The idea of the parameterization scheme is to represent the free variables by simple functions which are a sum of weighted shape functions. Typical shape functions are $sin(t)$, $cos(t)$ or t^n, where t is a variable and n is any integer; the weights serve to parameterize the function. The end-effector trajectory parameter vector P_e is a vector of the parameters, or weights, which charaterize the functions used to represent the free variables. By varying P_e, one can describe a large class of end-effector trajectories that satisfy the task specifications.

B. Parameterization of the desired joint trajectories.

As we have mentioned earlier, there are two common approaches for representing the kinematic redundancy of a redundant manipulator, namely the pseudo-inverse approach and the partitioned Jacobian approach. However, in the literature, there are very few approaches that address the parameterization of the joint trajectory which is an important issue in motion planning for redundant manipulators. In this study, we have developed a simple parameterization approach for describing the infinite number of joint trajectories that achieve a specified or parameterized end-effector trajectory. The steps in the joint trajectory parameterization approach, discussed in detail in Section 5.3, can be summarized as follows:

1. Based on a kinematic redundancy resolution scheme such as the pseudo-inverse approach, identify the independent variables which can be used to characterize the freedom in the joint trajectory.

2. Then, determine the boundary conditions for these free variables that would satisfy the task specifications.
3. Parameterize each free variable using a function which satisfies the boundary conditions. The joint trajectory parameter vector P_j is the vector of the parameters or weights used to describe all the free variables. P_j can be varied to describe a large class of joint trajectories that accomplish the desired end-effector motion.

C. Closed-Loop, Constrained Controller Synthesis

The closed-loop controller synthesis approach proposed by Beyer and Desa [5] for constrained non-linear systems will be incorporated in the unified motion planning methodology. We extend their simultaneous plant-controller design concept to motion planning problems where the trajectory and the controller will be designed in a single framework. We call this concept *simultaneous trajectory-controller design.* The feedback control law is developed in two steps:

Step 1: Define an input u' in accordance with an appropriate feedback control strategy:

$$u' = G(y(t), r(t), P_c, t) \tag{3.6}$$

where

$G(\cdot)$ denotes the feedback control strategy selected by the analyst,

$y(t) = \begin{pmatrix} q \\ \dot{q} \end{pmatrix}$, is the state vector of the system in joint space,

$r(t) = \begin{pmatrix} q_d \\ \dot{q}_d \end{pmatrix}$, is the desired joint space trajectory vector, and

P_c is a vector of parameters or "gains" used to represent the controller.

Step 2: Since the magnitude of the actual control effort u cannot exceed the actuator constraints u_{max}, the closed-loop control strategy is modified as follows in order to satisfy the actuator constraints:

$$u(t) = \begin{cases} u' & \text{if } |u'| < u_{max} \\ u_{max}\ sgn(u') & \text{if } |u'| \geq u_{max} \end{cases} . \tag{3.7}$$

Note that the control law Eq. (3.7) is a feedback control strategy which generates a control input u that cannot exceed the actuator constraints u_{max}.

The unknown controller parameter (gain) vector P_c of the modified control strategy, the end-effector trajectory parameter vector P_e, and the joint trajectory parameter vector P_j are obtained simultaneously by posing and solving an appropriate multi-criterion optimization problem with actuator constraints. One can see that by simultaneously planning the optimal trajectory and determining the optimal control law, we are able to obtain (i) an optimal manipulator motion that, because it is feedback-controlled, is robust in the face of disturbances and modeling errors and (ii) an optimal control strategy for which the control efforts do not violate the actuator constraints. In other words, by using the unified motion planning approach, we can overcome the shortcomings of the conventional approaches.

D. Multi-Criterion Optimization Technique

Multi-criterion optimization plays an important role in our motion planning framework. Using an appropriate multi-criterion optimization technique, we are able to consider the trade-offs between tracking performance and any additional measure of dynamic performance such as the magnitude of the base reactions or total task time.

Before we introduce the technique that we use, a few words on the differences between a single-criterion optimization problem and a multi-criterion optimization problem are in order.

A multi-criterion optimization problem can be simply stated as follow:

$$\text{minimize } I(P),$$

subject to:

$$g(P) \geq 0$$
$$h(P) = 0.$$

where P is a vector which consists of the decision parameters (or parameters to be optimized) and I is a vector of peformance indexes. In the motion planning problem, the optimization parameter vector P consists of the parameters that represent the freedom both in the trajectory (P_e and P_j) and the controller (P_c).

In the multi-criterion optimization area, the word "minimize" is used in a different sense. In a single criterion optimization problem, an optimal solution gives us the minimum cost function and is, in general, unique. However, in multi-criterion optimization problems, the optimal solution is usually a set. Only in situations where the performance indexes are non-conflicting can one obtain a unique optimal solution. For example, for the two non-conflicting performance indexes I_1 and I_2 shown in Fig. 3.1, the hatched area represents all possible values of $I(P) = [I_1\ (P)\ I_2(P)]^T$. The solution corresponding to point P is clearly the unique minimum solution.

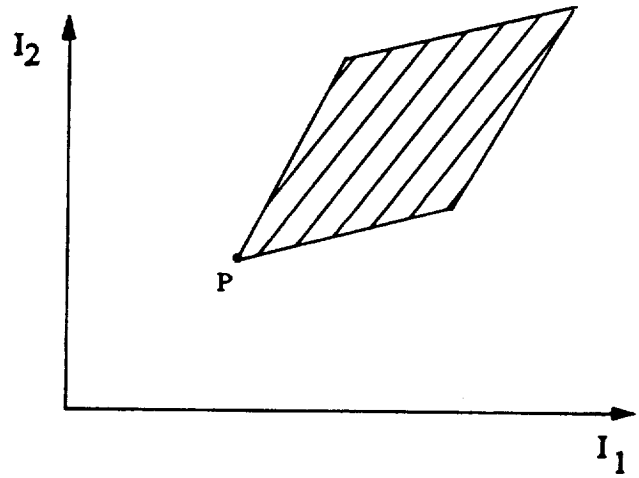


Fig. 3.1 Non-Conflicting Performance Indexes

For conflicting performance indexes, one has to choose an optimal solution from a set comprising an infinite number of optimal solutions. To illustrate this point, consider the space of the objective function vector $I(P) \in R^2$ shown in Fig. 3.2. The hatched area represents all the possible values of $I(P)$. One can see that the optimal solutions obviously lie on the heavy line AB. To describe all the possible optimal solutions as shown in Fig. 3.2, we will make use of the term *Pareto optimal* which is defined by Osyczka [29] :

An optimal solution P^* *is Pareto optimal if for every* $P \in P_{feasible}$, *there is at least one element of* $\mathbf{I}, I_i$ *such that*

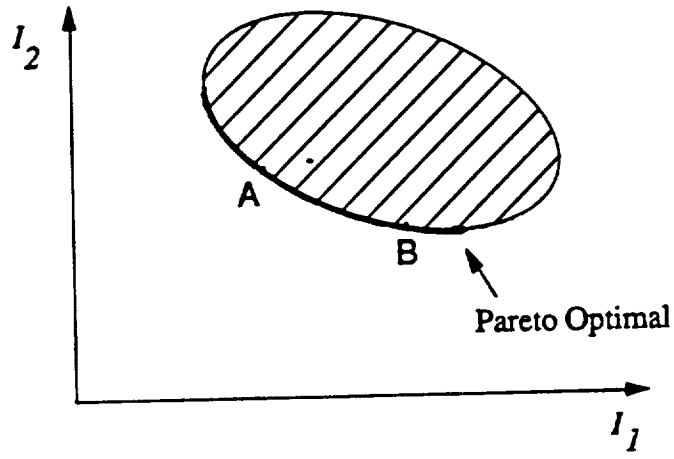


Fig. 3.2 An Illustration of Pareto Optimal Solutions.

$$I_i(P) > I_i(P^*). \tag{3.8}$$

where $P_{feasible}$ *denotes all the feasible parameter vectors.*

To illustrate this concept, we will compare two Pareto optimal solutions denoted by points *A* and *B* in Fig. 3.2. Let's first consider the optimal solution given by point *A*. Obviously, to decrease I_2, we can always pick the solution corresponding to point *B*. However, the solution given by point *B* will increase I_1. Similar arguments can also be applied to point *B*. One can see that all the points lying on the heavy line represent a set of optimal solutions called the *Pareto optimal* or *non-inferior solutions*. The term *non-inferior solution* reflects the fact that a Pareto optimal solution is one that cannot be improved without worsening at least one of the cost functions, I_1 or I_2.

In multi-criterion optimization, one always has to select a solution from the set of Pareto optimal solutions by carefully considering the trade-offs involved. There are various schemes that allow one to select an optimal solution from the set of Pareto-optimal solutions. Among some of the popular approaches are:

1. the trade-off method;
2. the weighting objective methods;

3. the goal programming method;
4. the min-max approach.

In our motion planning approach the performance index I_1 will be related to tracking performance while the performance index I_2 will be a measure of some additional dynamic performance. Since we have some knowledge of the maximum allowable magnitude of the tracking performance index I_1, the approach that we adopt in the unified motion planning problem is the trade-off approach which minimizes the performance index I_2 and treats the performance index I_1 as an inequality constraint; an example of the trade-off approach is given in Appendix A. Using the trade-off approach, we can then pose the following constrained optimization problem:

$$\begin{aligned} & \textit{minimize} \quad I_2 \\ \textit{subject to} \quad & \\ & I_1 \leq \gamma, \end{aligned} \tag{3.9}$$

where γ is the maximum allowable tracking error.

By solving the above constrained optimization problem, we can obtain the optimal solution, $\boldsymbol{P}^*$ for the parameter vector $\boldsymbol{P}$ which yields the optimal control strategy and the optimal joint trajectory.

3.4.2 Overview of the Unified Motion Planning Approach

In this section, we will give an overview of the unified motion planning approach. The motion planning approach is depicted in the block diagram shown in Fig. 3.3.

In this diagram, the inputs provided by the user are indicated by thick arrows. The inputs consists of the following:

1. the task specifications,
2. the maximum allowable tracking error γ,
3. an additional measure of dynamic performance,
4. initial guess $\boldsymbol{P}^0$ of the optimization parameter vector $\boldsymbol{P}$.

The unified motion planning approach can be divided into two key components, the *motion planning simulator* and the *optimization module*. We will first explain the basic operation of the motion planning simulator.

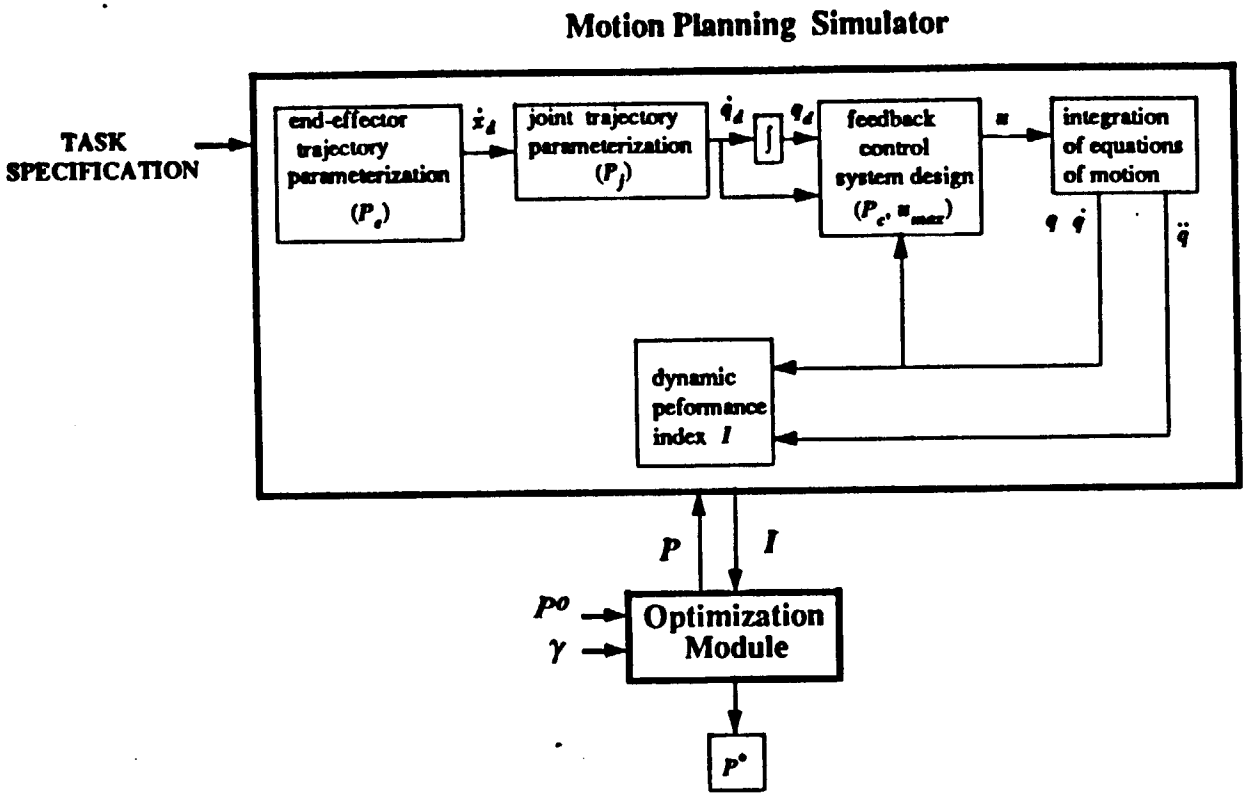


Fig. 3.3 A Block Diagram of the Unified Motion Planning Approach

Based on the task specifications, the end-effector trajectory parameterization block (see Fig. 3.3) characterizes any available freedom in the task specifications by polynomial functions with unknown coefficients. The elements of the end-effector trajectory vector P_e are the unknown coefficients of the polynomial functions. As we have discussed earlier these functions describe a desired end-effector trajectory x_d that satisfies the task specifications. From these polynomials, one can then determine the desired end-effector velocity $\dot{x}_d$ which is needed to compute the desired joint velocity $\dot{q}_d$.

The next block in the motion planning simulator is the joint trajectory parameterization block. As described in Section 3.4.1, one can obtain the desired joint velocity $\dot{q}_d$ from $\dot{x}_d$ using the velocity relationship, Eq. (2.5). For a non-redundant manipulator, one would obtain a unique desired joint velocity $\dot{q}_d$ corresponding to $\dot{x}_d$. In the case of a redundant manipulator, the desired joint velocity $\dot{q}_d$ is described by polynomial functions whose unknown coefficients are elements of the joint trajectory parameter P_j. One can integrate the desired joint velocity $\dot{q}_d$ to obtain the desired joint trajectory vector q_d.

The next module is the control system design module, characterized by the controller parameter vector P_c. The control strategy requires comparison of the actual joint trajectory q to the desired joint trajectory q_d and the actual joint velocity $\dot{q}$ to the desired joint velocity $\dot{q}_d$. Based on a control strategy specified by the analyst which is embedded in controller structure (3.7), the control system design module generates the actuator effort u which does not violate the actuator constraints u_{max}. The control effort u is the input to the equations of motion. The joint acceleration $\ddot{q}$ can then be obtained directly from the equations of motion. To obtain $\dot{q}$ and q, one can simply integrate $\ddot{q}$. Based on the actual joint trajectory $(q, \dot{q}, \ddot{q})$, one can then compute the performance index vector I (Eq. (2.6) or (2.7)).

As shown in Fig. 3.3, the outputs of the motion planning simulator are the values of $I(P)$ for a given optimization parameter vector P.

The second component of the unified motion planning approach is the optimization module which determines the optimal solution P^* for the parameter vector P. As shown in Fig. 3.3, the inputs to the optimization module are the initial guesses P^0 of P and the

maximum allowable tracking error γ. The optimization module contains a standard optimization technique for solving constrained, multi-criterion optimization problems based on the trade-off approach. The optimization module converts a multi-criterion objective optimization problem into a constrained single criterion optimization problem and determines a Pareto optimal solution P^* which satisfies the tracking performance ($I_1 \leq \gamma$) and simultaneously minimizes the additional dynamic performance index I_2. The output of the optimization module P^* contains the optimal desired motion characterized by P_e and P_j and the optimal control strategy gains which are elements of P_c.

To implement the results of the motion planning on the actual manipulator, one would generate the reference trajectory for a feedback control strategy from the optimal parameter vectors P_e^* and P_j^*. The optimal controller parameter vector P_c^* is then used in the feedback control strategy given by Eq. (3.7) to generate the control effort u to drive the actuators.

The software (programming) environment MATRIX$_x$™ was used to implement the unified motion planning approach for the following reasons:

1. It allows the user to perform simulations through convenient, user-defined, modular building blocks.
2. It contains a powerful optimization module.
3. The simulation module and the optimization can be linked to each other very easily, a requirement which is crucial to the realization of our motion planning approach.

In the next two chapters we will apply the unified motion planning approach to planning motions for both non-redundant and redundant manipulators.

Chapter 4

Motion Planning of Non-redundant Manipulators

4.1 Introduction

In this chapter, we will illustrate how one can use the motion planning approach to simultaneously plan the trajectory and design a feedback control law for a non-redundant manipulator such that actuator constraints are not violated. In Section 4.2, we introduce the concepts of a feasible motion plan and an optimal motion plan for a manipulator. In Section 4.3, we enumerate three types of task specifications and also present methods to parameterize the freedom available in each type of task specification. In Section 4.4, we discuss the application of our unified approach for planning motions for non-redundant manipulators. Finally, in Section 4.5 we illustrate the application of the motion planning approach to planning feedback-controlled minimum-time trajectories for a non-redundant 2 d.o.f. planar manipulator.

4.2 Feasible Motion Plan and Optimal Motion Plan

Using the motion planning approach proposed in this study, one can plan two types of motions - a feasible motion and the optimal motion. The purpose of this section is to formally define these motions.

In non-redundant manipulator motion planning, depending on the task specifications one may or may not be able to exploit the freedom in the end-effector trajectory. For those tasks that *do not* allow any freedom in the end-effector trajectory, it is important to find out whether the specified trajectory can be achieved by a control strategy under actuator constraints. If this is possible, then we say the trajectory is a *feasible trajectory* for the task specifications.

For those tasks that *do* allow freedom in the end-effector trajectory, in addition to determining whether the trajectory is feasible, we can also ask the question: *Can we find a feasible trajectory that improves the dynamic performance of a non-redundant manipulator?* The trajectory that satisfies the task specifications and also optimizes an additional measure of dynamic performance is called an *optimal trajectory* for the task. As an example, in applications such as arc-welding, a *feasible* trajectory would be an end-effector trajectory that can be tracked by a feedback-controlled manipulator without violating actuator constraints. If, in addition, we also want to execute the task in the shortest possible time, then an optimal trajectory would be a feasible trajectory that accomplishes the task in minimum task time.

As stated in Chapter 2, we require a motion plan which

(i) satisfies the task specifications;

(ii) is robust. (i.e. achieved by a feedback control strategy);

(iii) does not violate actuator constraints;

(iv) optimizes an additional measure of dynamic performance.

Let x_d and x be the desired end-effector trajectory vector and actual end-effector trajectory vector, respectively. We can then define the end-effector trajectory tracking error vector x_e by

$$x_e = x - x_d. \tag{4.1}$$

In order to meet the first three motion requirements we will define a performance index I_1

$$I_1 = max\,|x_e(t)|. \tag{4.2}$$

where $|x_e(t)|$ is the magnitude of the tracking error of the end-effector at time t. Let γ denote the maximum allowable tracking error for the task, i.e.

$$max\ |x_e(t)| \leq \gamma. \tag{4.3}$$

In order to meet the fourth motion requirement we will define an additional performance index I_2 which is a measure of the additional dynamic performance.

We now define two terms which play an important role in the development of our motion planning framework.

Feasible Motion: A manipulator motion that satisfies the first three requirements above and in addition satisfies Eq. (4.3) is called a *feasible motion.*

Optimal Motion: An optimal motion is a feasible motion that minimizes an additional dynamic performance index I_2.

4.3 Task Specifications

In pick-and-place processes, the end-effector is required to move an object from point *A* to point *B*. In this type of operation, we are only concerned with whether the end-effector reaches point *B* in a reasonable amount of time. Therefore, we have the freedom to select any end-effector trajectory between end-points that would give us good performance. However, in some other applications such as arc-welding, the end-effector has to track a prescribed end-effector trajectory which is either (1) only specified spatially or (2) completely specified. In the former case, since the trajectory is only specified spatially, we can utilize the freedom available in time to optimize a secondary dynamic performance criterion. In the latter case, the task specification of the end-effector trajectory does not provide any freedom for one to exploit.

In general, the task specifications of a manipulator can be classified into three categories:

- ***Type I Trajectory :*** End-effector trajectory specified in space and in time, i.e. $x(t)$ is known.
- ***Type II Trajectory :*** Trajectory specified in space but not in time.
- ***Type III Trajectory :*** Only the end points of the end-effector trajectory are specifed.

For tasks of type I, no freedom is available for the end-effector trajectory. The motion planning problem is to determine whether the specified trajectory is feasible or not. However, for tasks of types II and III, one can take advantage of the freedom available in the end-effector trajectory to determine either a feasible trajectory or an optimal trajectory.

4.3.1 Type II Specification

In a type II task specification, the end-effector is required to track a trajectory specified in space. The desired end-effector trajectory can be specified in space by

$$x_d = x_d(\alpha), \quad \alpha_{min} \le \alpha \le \alpha_{max} \tag{4.4}$$

where α is a variable which is a function of the time t. In some applications, it is convenient to choose α as the arc length of the curve that describes the trajectory.

The use of a parametric description to describe a curve in space is common in manufacturing and computer graphics [13]. In robotic applications, Bobrow et al. [6] used the above description to parameterize the path of the end-effector in their minimum-time studies.

By differentiating Eq. (4.4) with respect to time, one can obtain the desired velocity of the end-effector

$$\dot{x}_d = x_d'(\alpha)\,\dot{\alpha}, \tag{4.5}$$

where

$$x_d'(\alpha) = \frac{dx_d}{d\alpha}. \tag{4.6}$$

Let t_f be the variable that represents the total task time. Since the end-effector is at rest at the initial and at the final position, we impose the following boundary conditions on $x_d(t)$:

$$\dot{x}_d(0) = \dot{x}_d(t_f) = 0. \tag{4.7}$$

If we set $\dot{\alpha}(0) = \dot{\alpha}(t_f) = 0$, then from Eq. (4.5) we see that the boundary conditions given by Eq. (4.7) will be satisfied. By differentiating Eq. (4.7), we can obtain the acceleration of the end-effector trajectory as :

$$\ddot{x}_d = x_d''(\alpha)\,\dot{\alpha}^2 + x_d'(\alpha)\,\ddot{\alpha}, \tag{4.8}$$

where

$$x_d''(\alpha) = \frac{d^2 x_d}{d\alpha^2}. \tag{4.9}$$

In addition to the velocity boundary conditions, the acceleration of the end-effector must also be zero at $t=0$ and $t=t_f$, i.e.,

$$\ddot{x}_d(0)=\ddot{x}_d(t_f)=0. \tag{4.10}$$

If we let $\dot{\alpha}(0)=\dot{\alpha}(t_f)=0$ and $\ddot{\alpha}(0)=\ddot{\alpha}(t_f)=0$, then from Eq. (4.8) we see that the zero acceleration requirements at the end-points are satisfied. Hence, the appropriate boundary conditions that $\alpha(t)$ has to satisfy are

$$\begin{aligned}
&\alpha(0)=\alpha_{min}\\
&\alpha(t_f)=\alpha_{max}\\
&\dot{\alpha}(0)=\dot{\alpha}(t_f)=0\\
&\ddot{\alpha}(0)=\ddot{\alpha}(t_f)=0.
\end{aligned} \tag{4.11}$$

The variable $\alpha(t)$ can be parameterized in a simple fashion using a polynomial of order l as follows:

$$\alpha(t)=\sum_{i=0}^{l} a_i t^i, \tag{4.12}$$

where a_i $(i=0, \ldots, l)$ are the parameters or weights in the representation. In order to satisfy the six boundary conditions (4.11), the coefficients a_0, a_1, a_2, a_{l-2}, a_{l-1}, a_l must satisfy the following equations:

$$\begin{aligned}
&a_o=\alpha_{min}\\
&a_1=a_2=0\\
&a_{l-2}=\frac{(\alpha_{max}-\alpha_{min})(l^2-l)+\sum_{i=3}^{l-3}(2il-i-l^2+l-i^2)a_i t_f^i}{2t_f^{l-2}}\\
&a_{l-1}=\frac{-(\alpha_{max}-\alpha_{min})l(l-2)+\sum_{i=3}^{l-3}[l(l-2)+i(i+2-2l)]a_i t_f^i}{t_f^{l-1}}\\
&a_l=\frac{(\alpha_{max}-\alpha_{min})(l-1)(l-2)+\sum_{i=3}^{l-3}[-(l-1)(l-2)+2i(l-2)-i(i-1)]a_i t_f^i}{2t_f^l}.
\end{aligned} \tag{4.13}$$

The remaining variables a_3, a_4, . . . a_{l-3} are the free variables that one can use to describe a class of end-effector trajectories that satisfy the boundary conditions (4.11).

Let P_e denote the vector whose elements are the free variables a_3, ..., a_{l-3} and the total task time t_f, i.e.,

$$P_e = [a_3,\ldots,a_{l-3},t_f]^T. \tag{4.14}$$

4.3.2 Type III Specification

Since for a Type III task specification, only the end-points are specified, we have the freedom to pick any trajectory between the end-points. A spatial end-effector trajectory can be represented by three l^{th}-order polynomials in t:

$$\begin{aligned} x_{d1}(t) &= \sum_{i=0}^{l} a_{1i}t^i \\ x_{d2}(t) &= \sum_{i=0}^{l} a_{2i}t^i \\ x_{d3}(t) &= \sum_{i=0}^{l} a_{3i}t^i. \end{aligned} \tag{4.15}$$

The six boundary conditions for a Type III trajectory can be expressed as:

$$\begin{aligned} x_d(0) &= x_o \\ x_d(t_f) &= x_f \\ \dot{x}_d(0) &= \dot{x}_d(t_f) = 0 \\ \ddot{x}_d(0) &= \ddot{x}_d(t_f) = 0. \end{aligned} \tag{4.16}$$

The order of the polynomial given in Eq. (4.15) must be high enough to satisfy the boundary conditions given in Eq. (4.16) and also to provide enough freedom to represent a class of end-effector trajectories between the end-points. Since for each x_{di},$(i=1,2,3)$, there are six boundary conditions which must be satisfied, the order of the polynomial must be greater than five. If we allow k independent variables in the polynomial $x_{di}(t)$, then the order l of $x_{di}(t)$ is equal to $k+5$. If we choose a_{i3},a_{i4},a_{i5}, ...,$a_{i(l-3)}$, $(i=1,2,3)$ as the free variables for the polynomial, $x_{di}(t)$, $(i=1,2,3)$, we can then define the end-effector trajectory parameter vector, P_e as

$$P_e = [a_{13},a_{14},\ldots,a_{1(l-3)},\ldots,a_{33},a_{34},\ldots,a_{3(l-3)},t_f]^T. \tag{4.17}$$

The vector P_e, which has $3(l-5)$ elements, represents the freedom in the specification of the end-effector trajectory.

The boundary conditions imposed on $x_{di}(t)$ resemble those imposed on $\alpha(t)$ (see Eqs. (4.11) and (4.16)). Therefore a_{ij}, $(j=3,4,...,(l-3))$, $(i=1,2,3)$, can be readily computed by expressions similar to Eq. (4.13).

4.4 Procedure for Obtaining a Feasible Motion

In this section, we derive a procedure for obtaining a feasible trajectory for non-redundant manipulators.

Step 1: Define task specifications.

In a Type I specification, we command the end-effector to move from point x_o to point x_f along a prescribed trajectory. The task specifications can be simply stated as:

$$\begin{aligned} &x_d(0)=x_o \\ &x_d(t_f)=x_f \\ &x_d(t)=f(t),\ \ 0\le t\le t_f, \end{aligned} \tag{4.18}$$

where $f(t)\in R^3$ is a vector of prescribed functions which represent the end-effector motion in 3-space.

In a Type II specification, we command the end-effector to move from point x_0 to point x_f along an end-effector trajectory prescribed in space, $x(\alpha)$, within the maximum allowable task time, t_f'. Mathematically, we can express the task specifications by

$$\begin{aligned} &x_d=x_d(\alpha) \\ &x_d(0)=x_o \\ &x_d(t_f)=x_f \\ &\alpha(0)=\alpha_{min} \\ &\alpha(t_f)=\alpha_{max} \\ &\dot\alpha(t_0)=\dot\alpha(t_f)=0 \\ &\ddot\alpha(0)=\ddot\alpha(t_f)=0 \\ &t_f\le t_f'. \end{aligned} \tag{4.19}$$

In a Type III specification, the end-effector must move from point x_o to point x_f. Mathematically, we can express the task specifications by

$$
\begin{aligned}
&x_d(0)=x_o \\
&x_d(t_f)=x_f \\
&\dot{x}_d(0)=\dot{x}_d(t_f)=0 \\
&\ddot{x}_d(0)=\ddot{x}_d(t_f)=0 \\
&t_f \le t_f'.
\end{aligned}
\tag{4.20}
$$

Step 2: Parameterize the end-effector trajectory (for Type II and Type III task specifications only).

(a) *Type II Specification*

For a Type II trajectory, using the parameterized polynomial function developed in Section 4.3.1, α can be expressed by the following expression:

$$\alpha(t)=f_1(t,P_e), \tag{4.21}$$

where f_1 is a polynomial given by Eq. (4.12) and P_e is defined in Eq. (4.14).

Note : The vector P_e includes t_f as one of the parameters. This enables us to describe a class of trajectories that satisfy task specifications and whose task time t_f is less than or equal to the maximum allowable task time, t_f'.

(b)*Type III Specification*

For Type III trajectory, we can use Eq. (4.15) to obtain a parametric description for each $x_{di}(t)$, $i=1,2,3$. Using the parameterized polynomial functions developed in Section 4.3.2, we can express $x_{di}(t)$ by

$$x_d(t)=f_2(t,P_e), \tag{4.22}$$

where P_e is defined by Eq. (4.17) and the elements x_{di} $(i=1,2,3)$ of f_2 are given by Eq. (4.15).

Step 3: Determine the desired joint trajectories, q_d. Then, using the velocity

relationship stated in Eq. (2.4), we can obtain an expression for the derivative of the joint trajectory $\dot{q}_d$ by simply pre-multiplying the desired end-effector velocity vector by the inverse of the Jacobian matrix :

$$\dot{q}_d = J^{-1}\dot{x}_d. \tag{4.23}$$

The desired end-effector velocity vector $\dot{x}_d$ is first determined by differentiating the end-effector trajectory x_d obtained in Step 2. The expression for $\dot{q}_d(t)$ can be numerically integrated using standard routines such as the Runge-Kutta method or the Kutta-Merson method to obtain the desired joint trajectory vector $q_d(t)$. q_d and $\dot{q}_d$ will be used in the computation of the control inputs **u** (see Step 4).

Step 4: Obtain the state-space dynamic model of the manipulator.

The state vector y of a manipulator is defined as

$$y \equiv \begin{pmatrix} q \\ \dot{q} \end{pmatrix}. \tag{4.24}$$

The equations of motion of a manipulator [1] can be expressed as

$$u = M(q)\ddot{q} + V(q,\dot{q}) + G(q), \tag{4.25}$$

where

u is the vector of torques applied at the joints,

$M(\cdot)$ is the mass matrix of the manipulator,

$V(\cdot)$ is a vector of nonlinear terms in q and $\dot{q}$,

and $G(\cdot)$ is the vector of the terms contributed by gravitational forces acting on the manipulator.

The joint acceleration of the manipulator $\ddot{q}$ can then be obtained from the dynamic equations Eq. (4.25) as

$$\ddot{q} = M^{-1}(q)(u(t) - V(q,\dot{q}) - G(q)). \tag{4.26}$$

Defining matrices A and B as follows:

$$B=M^{-1}(q)$$
$$A=B\,(V(q,\dot{q})-G(q)), \tag{4.27}$$

we can express $\ddot{q}$ by

$$\ddot{q}=A(y)+B(y)\,u. \tag{4.28}$$

Using Eqs. (4.26)-(4.28), we can write the equations of motion of the dynamic system as a set of $2m$ ordinary first-order differential equations:

$$\dot{y}=C(y)+D(y)\,u(t), \tag{4.29}$$

where

$$C=\begin{pmatrix}\dot{q}\\ A\end{pmatrix}$$
$$D=\begin{pmatrix}O\\ B\end{pmatrix}. \tag{4.30}$$

The above state-space equations (4.29) can be integrated numerically to obtain the values of the state vector y. The values of y and $\dot{y}$ are useful in computing the performance index defined in Step 7 below.

Step 5: Formulate the appropriate control strategy

We want to formulate a closed-loop control strategy that yields a control effort vector u which does not violate the actuator constraints, u_{max}. We will develop the control structure[1] in two stages:

(*Stage 1*) Define an input u' based on some desired feedback control strategy:

$$u'=G(y(t),r(t),P_c,t) \tag{4.31}$$

where

$G(\cdot)$ denotes the control strategy,
$y(t)$ is the state vector,

[1]This stage was discussed in Section 3.4 of Chapter 3 and is repeated here for convenience

$r(t) = \begin{pmatrix} q_d \\ \dot{q}_d \end{pmatrix}$, is the desired joint space trajectory vector, and

P_c is a vector of parameters used to represent the controller.

(*Stage 2*) Since the magnitude of the actual input u cannot exceed the actuator constraint u_{max}, we modify the strategy defined in stage (1) to yield the following controller structure:

$$u(t) = \begin{cases} u' & \text{if } |u'| < u_{max} \\ u_{max}\ sgn(u') & \text{if } |u'| \geq u_{max} \end{cases} . \tag{4.32}$$

By using the controller structure defined in stage 2, we have a closed-loop controller that satisfies the motion planning requirements (ii) and (iii) as listed in Section 4.2.

Step 6: Identify the optimization parameters.

Using the parameterization expressions for the end-effector trajectory x, we can characterize the motions of a manipulator by a vector P, which is defined as follows:

$$P = \begin{pmatrix} P_e \\ P_c \end{pmatrix}, \tag{4.33}$$

where P_e is the task-space parameter vector defined in Step (2), and P_c is the controller parameter vector of the control strategy defined in Step (4). P is therefore the vector whose elements are the parameters which must be optimized. The elements of P will be determined by solving the tracking performance problem defined in Step (7). Note that for a Type I trajectory, P only consists of the elements in P_c.

Step 7: Determine the optimization parameter vector P by posing an appropriate optimization problem.

As we have mentioned earlier, the parameters in P are the optimization parameters that one can manipulate to obtain a feasible trajectory. To determine these parameters,

we must pose an appropriate optimization problem for the tracking performance index, I_1.

Before we formulate the optimization problem, we will first define a few key variables. Let q_o and q_f be, respectively, the joint-space configurations corresponding to the specified end-effector position, x_o and x_f. Then, the trajectory specifications in the joint space for the Type III specification are:

$$\begin{aligned} q_d(0) &= q_o \\ q_d(t_f) &= q_f \\ \dot{q}_d(0) &= \dot{q}_d(t_f) = 0. \end{aligned} \tag{4.34}$$

We can also define the joint-space error vector as:

$$y_e(t) = \begin{pmatrix} q_e(t) \\ \dot{q}_e(t) \end{pmatrix} = \begin{pmatrix} q_d(t) - q(t) \\ \dot{q}_d(t) - \dot{q}(t) \end{pmatrix}. \tag{4.35}$$

For Type III specification, to ensure that the end-effector is at the desired final position, x_f at $t=t_f$, we want $y_e(t_f)=0$. Thus, the most straight-forward performance index one can use is in the form:

$$I_1(P) = q_e(t_f)^T q_e(t_f) + w_1 \dot{q}_e(t_f)^T \dot{q}_e(t_f). \tag{4.36}$$

We can therefore pose the following optimization problem to obtain a feasible trajectory for a Type III trajectory:

$$min \;\; I_e(P) = q_e^T q_e + w_1 \dot{q}_e^T \dot{q}_e. \tag{4.37}$$

subject to the constraints (4.25), (4.32) and $t_f \leq t_f'$.

In above equation w_1 is the weighting factor for scaling and ensuring dimensional homogeneity of the two quadratic forms in I_1, t_f is a variable which defines the final time when the end-effector reaches $x_d = x_f$. If there is perfect tracking, then the actual position, $x(t_f)=x_f$, $\dot{x}(t_f)=0$ and therefore $y_e(t_f)=0$.

However, the performance index given in Eq. (4.37) does not ensure good tracking performance along the trajectory, a requirement which might be important in the general case. Therefore, a more suitable performance index (for type I, II and III trajectory) might be

$$I_1(P)=max|x_e|, \tag{4.38}$$

which minimizes the peak magnitude of the tracking error

With the latter performance index, we can pose the following unconstrained optimization problem to obtain the optimal P for a feasible motion:

$$min \qquad I_1(P)=max|x_e|, \tag{4.39}$$

subject to the constraints (4.25), (4.32) and $t_f \leq t_f'$.

The solution P^* to the optimization problem (4.37) or (4.39) yields the feasible trajectory and the gains for the feedback control strategy.

4.5 Procedure for Obtaining the Optimal Motion

In addition to realizing the primary goal of satisfying the tracking requirements, we are often interested in using any available freedom in the end-effector trajectory to improve some additional aspect of a manipulator's dynamic performance (the secondary goal). In such cases, we have to consider two performance indexes, one of which is a measure of the maximum tracking error and the other a measure of the additional dynamic performance of interest. Let I_1 and I_2 be the tracking performance index and the secondary dynamic performance index, respectively; I_1 is the tracking performance index defined in Eq. (4.38).

If $I(P) = [I_1\ I_2]^T$ denotes the performance index vector, then we can pose the following multi-criterion optimization problem for obtaining the optimal motion:

$$minimize \quad I(P). \tag{4.40}$$

subject to the constraints (4.25), (4.32) and $t_f \leq t_f'$.

The above optimization problem basically states that we want to minimize both I_1 and I_2 simultaneously. However, as pointed out in Section 3.4, only in situations where I_1 and I_2 are non-conflicting, can we attain the minimum of I_1 and I_2 simultaneously. For this situation, we can simply minimize I_1 or I_2 to obtain the minimum solution. In other cases, we have to consider the trade-off between conflicting performance indexes.

In a lot of motion planning problems we can only improve the additional performance index by sacrificing the tracking performance. Therefore, to minimize $I(P)$ effectively, we will use a multi-criterion optimization method called the trade-off method. In Fig. 4.1, a two-dimensional performance index space is shown. I_1 denotes the tracking performance index which has to be less than or equal to the maximum allowable tracking error, γ.

Let S_f be a set which consists of all the feasible motions, i.e. S_f is given by:

$$S_f = \{P \mid I_1(P) = \gamma' \leq \gamma\}. \tag{4.41}$$

The heavy line FO in Fig. 4.1 represents the set S_p of all the Pareto optimal solutions that meet the feasible motion requirement. Using the trade-off approach, described in Section 3.4, we can obtain an optimal motion from S_p by posing the following constrained single-criterion optimization problem:

$$\begin{aligned} &\min \ I_2(P) \\ &\text{subject to} \\ &I_1(P) \leq \gamma. \end{aligned} \tag{4.42}$$

The result of the above optimization problem is shown as point O in Fig. 4.1 which is the smallest value of performance index I_2 under the tracking performance constraint on I_1.

The procedure for obtaining an optimal trajectory can be summarized as:

1. Define the maximum allowable tracking error γ from the task specifications. Select an initial guess for the optimization parameter, P^0.
2. Obtain a feasible motion using the procedure described in Section 4.4. If a feasible motion is unattainable, modify the task specification or the actuator constraints in order to obtain a feasible motion.
3. With P^0 and γ, solve the nonlinear programming problem stated in Eq.

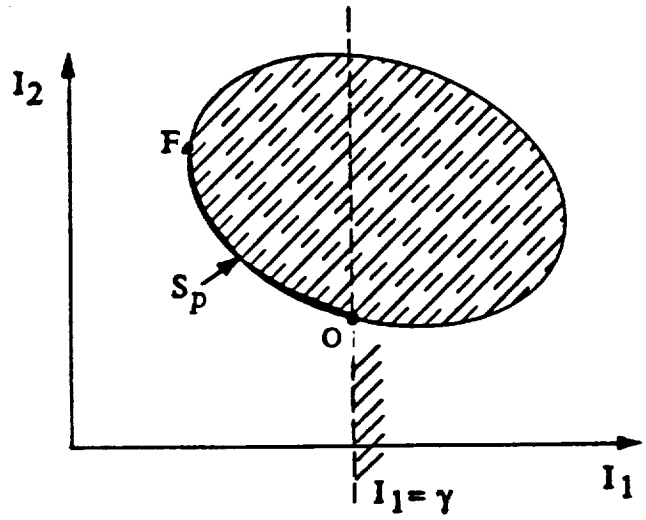


Fig. 4.1 Pareto Optimal Solutions Which Satisfy the Feasible Motion Requirement.

(4.42) and obtain the optimal solution P^* for the (optimization parameter vector) P. Once P^* is obtained, the optimal trajectory and associated feedback-control law can be readily determined.

4.6 Example: Feedback-Controlled Minimum-Time Motion Planning

In this section, we will use an example to demonstrate the process of obtaining the minimum time trajectory for a non-redundant feedback-controlled manipulators. In a lot of pick and place operations, a manipulator is required to pick up an object and place it down in the shortest possible amount of time. The problem that we wish to solve can be stated as follows:

Given the initial position and end position of the end-effector for a non-redundant manipulator with actuator constraints u_{max}, obtain an optimal trajectory $x(t)$ and the closed-loop control strategy that will achieve the task specifications in minimum task time.

The example that we will study is a two d.o.f. planar manipulator shown in Fig. 4.2. Let l_i and m_i denote the length and mass, respectively, of joint i. l_{ci} denotes the distance

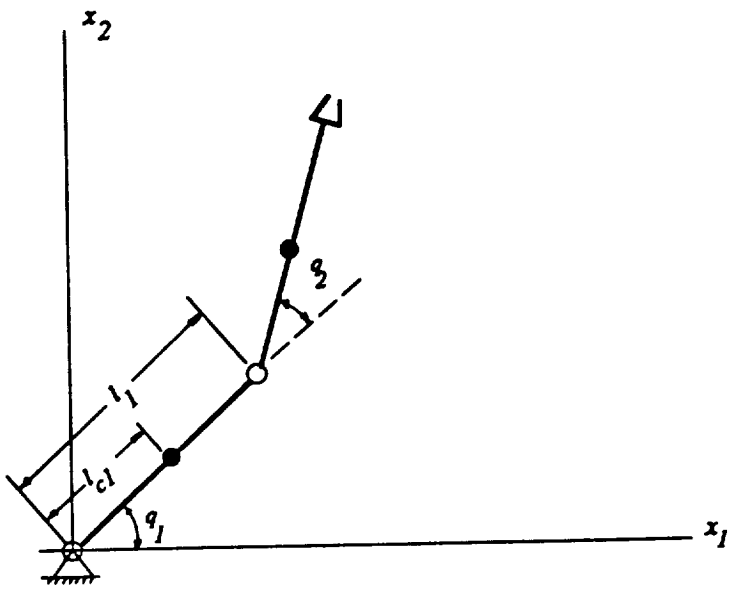


Fig. 4.2 A Two Degree-of-Freedom Planar Manipulator .

from the joint-axis of joint i to the center of mass of link i. The central moment of inertia for an axis perpendicular to the plane of motion of link i is given by I_i. The link dimensions and the mass properties are shown in Table 4.1. The equations of motion of this manipulator are given in [1]. The actuator at each joint can deliver a maximum torque of *0.1 N-m*. Therefore, the actuator constraints can be expressed by $u_{max} = (0.1\ 0.1)^T$.

In the following discussion, we will study three cases. The results of these three cases are tabulated and summarized in Table 4-2. In Case 1, we arbitrarily let task time $t_f = 2.0s$. The objective is to see if we can obtain a feasible trajectory that can be executed with task time t_f=2.0s. In Case 2, we solve the same problem as in Case 1, except the task time is increased to t_f=4.0s. The objective is to examine the effect of the task time on the tracking performance and required actuator torques. In Case 3, we find the optimal trajectory that can be executed in minimum-time.

	Link 1	Link 2
m_i [kg]	1.0	1.0
l_i [m]	0.5	0.5
l_{ci} [m]	0.25	0.25
I_{ci} [kg-m^2]	0.005	0.005

Table 4-1 Link Dimensions and Mass Properties

Case	Problem	Results	Conclusions
1	***Find a feasible trajectory*** • t_f=2.0s • $(u_{max})_1$=0.1N-m $(u_{max})_2$=0.1N-m	• Poor tracking performance • max $\|x_e\|$ = 0.0075m see Fig. 4.3 (a,b)	• Cannot find a feasible trajectory.
2	***Find a feasible trajectory*** • t_f=4.0 sec • $(u_{max})_1$=0.1N-m • $(u_{max})_2$=0.1N-m	• Tracking perf. is improved. • max $\|x_e\|$=0.0012m See Fig. 4.4 (a,b).	• By relaxing task time t_f, a feasible trajectory is found.
3	***Find the minimum-time trajectory*** • t_f< 4.0s.	• max $\|x_e(t)\|$=0.003m. See Fig. 4.5 (a,b).	• An optimal trajectory with t_f=3.0s is obtained.

Table 4-2 Summary of Results

4.6.1 Case 1 : Finding a Feasible Trajectory with t_f= 2.0s

For illustration purposes, we will solve the problem in accordance with the procedure outlined in Section 4.4 for a Type III specification.

Step 1: The task of the non-redundant manipulator is to move from point *A* (0.3536, 0.3536)(m) to point *B* (0.5656, 0.5656) (m) with the maximum tracking error less than γ which is specified as 1 % of total distance traversed. The task specifications (Type III) are given as follows:

$$
\begin{aligned}
&t_f = 2.0s \\
&\boldsymbol{x}(0) = (0.3536, 0.3536)^T \\
&\boldsymbol{x}(t_f) = (0.5656, 0.5656)^T \\
&\dot{\boldsymbol{x}}(0) = \dot{\boldsymbol{x}}(t_f) = \ddot{\boldsymbol{x}}(0) = \ddot{\boldsymbol{x}}(t_f) = 0 \\
&\gamma = 0.003m.
\end{aligned} \tag{4.43}
$$

Step 2: Parameterize the end-effector trajectory to satisfy the task specifications given in Step 1. In order to have enough parameters to represent a large class of end-effector trajectories, we allow two free variables in each of x_1 and x_2, i.e. the order of the polynomial used to represent x_1 and x_2 is equal to 7. We can then write x_1 and x_2 as

$$
\begin{aligned}
x_1(t) &= \sum_{i=0}^{i=7} a_{1i} t^i, \\
x_2(t) &= \sum_{i=0}^{i=7} a_{2i} t^i.
\end{aligned} \tag{4.44}
$$

From above equation, the end-effector trajectory parameter vector $\boldsymbol{P_e}$ is given by :

$$
\boldsymbol{P_e} = [a_{13} \;\; a_{14} \;\; a_{23} \;\; a_{24}]^T. \tag{4.45}
$$

Step 3: Obtain the desired joint trajectory using the joint velocity equation (Eq. (4.23)). The desired end-effector velocity vector can be obtained by differentiating Eq. (4.44).

Step 4: For simplicity we choose a PD controller for the control strategy $\boldsymbol{u}'$ in Eq. (4.31) with proportional gain vector, $\boldsymbol{k_p}$, and derivative gain vector, $\boldsymbol{k_d}$, as the unknown controller gain vectors.

For the PD controller, the controller parameter $\boldsymbol{P_c}$ can be written as

$$
\boldsymbol{P_c} = [k_{p1} \;\; k_{p2} \;\; k_{d1} \;\; k_{d2}]^T. \tag{4.46}
$$

The PD control strategy can be embedded in the control structure (4.32). Now, we have a PD control strategy that satisfies the actuator constraints with the control strategy parameters given by the elements of $\boldsymbol{P_c}$ in Eq. (4.46).

Note that in this example we have chosen a simple PD controller. For better

performance one might need to use a more complicated control strategy. Using the unified motion planning approach, one can test different control strategies by simply replacing u' in Eq. (4.32) by a particular control strategy of interest.

Step 5: Based on the dynamic equations of a 2 d.o.f. manipulator given in [1], obtain the state equations of the system using Eq. (4.29).

Step 6: The optimization parameter vector P, consisting of the end-effector trajectory parameter P_e and the controller parameter P_c, can be written as

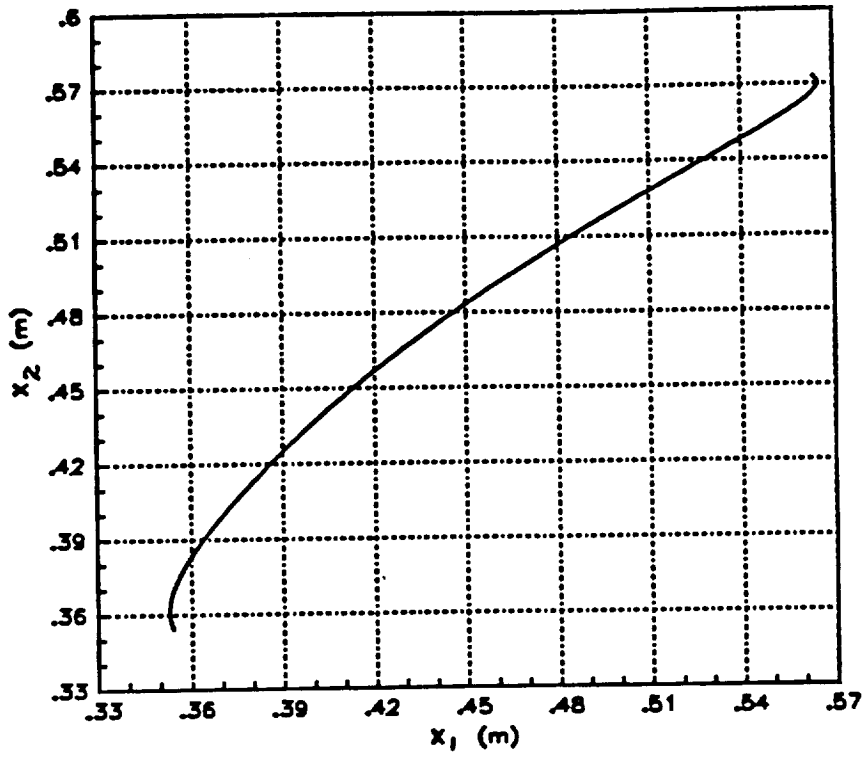
$$P=[a_{13}\ \ a_{14}\ \ a_{23}\ \ a_{24}\ \ k_{p1}\ \ k_{p2}\ \ k_{d1}\ \ k_{d2}]^T. \qquad (4.47)$$

Step 7: Determine P_c^* and P_e^*, the optimal solutions for the following tracking performance optimization problem:

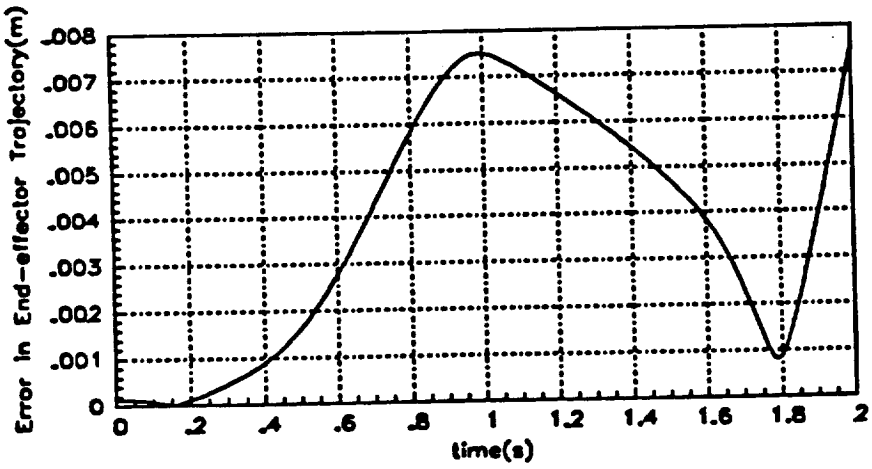
$$min\ \ I_1=max|\mathbf{x}_e(t)|. \qquad (4.48)$$

The initial guess for P is P^0 = $[0\ 0\ 0\ 0\ 10\ 10\ 10\ 10]^T$. After solving the above optimization problem, the optimal solution was given by $\mathbf{P}^*$ = $[-0.048\ -0.0241\ 0.0746\ 0.0339\ 9.9998\ 10\ 10.0002\ 10.0]^T$. Once P^* is obtained, we can then obtain the optimal trajectory and the optimal PD controller. The results of the simulation are shown in Figs. 4.3-4.6.

The actual end-effector trajectory is shown in Fig. 4.3 (a). In Figs. 4.3 (b), one can see that the trajectory has a maximum tracking error of 0.0075m which exceeds the tracking error specification of 0.003m. Therefore, no feasible end-effector trajectory can be obtained for a total task time of 2.0s. The joint trajectory of the manipulator is shown in Fig. 4.3. We can see that the trajectories are smooth and have zero velocity at the end of the task time. As shown in Fig. 4.3(d), the actuator of joint 1 is saturated from 0.25s to 1.0s and from 1.5s to 2.0s. The actuator of joint 2 is also saturated from 1.64s to 1.95s. The highly saturated torque profiles suggest that with the time constraint that we imposed on the problem, the tracking performance requirement cannot be satisfied.

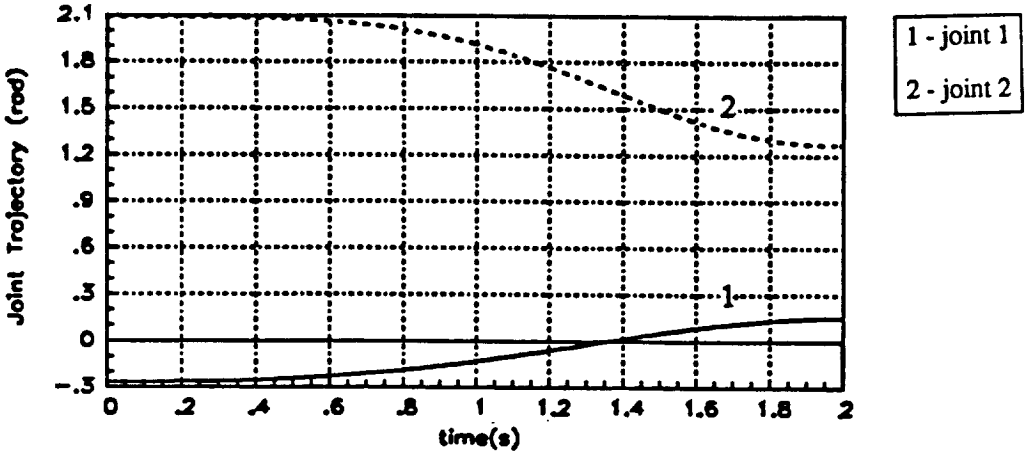


(a)

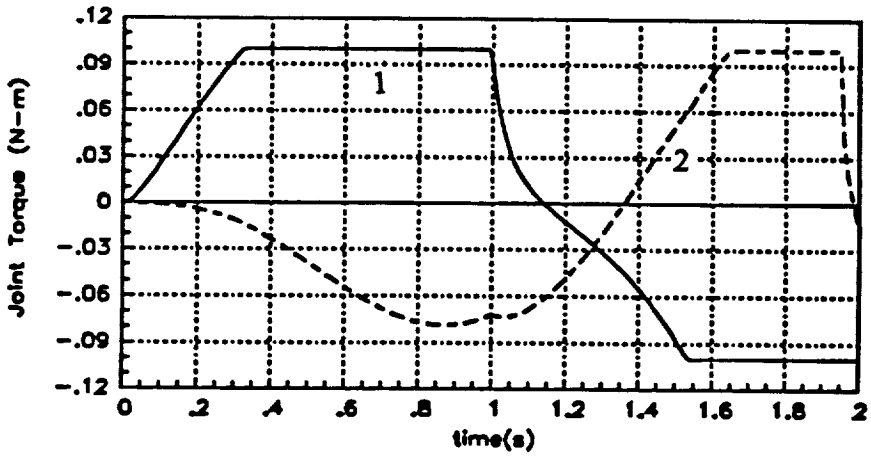


(b)

Fig. 4.3 Results of Case 1: (a) Actual End-effector Trajectory (b)Tracking Error



(c)



(d)

Fig. 4.3 (cont'd) (c) Joint Trajectory (d)Joint Torque

4.6.2 Case 2: Finding a Feasible Trajectory with t_f=4.0s.

In this case study, we relax the time constraint by letting t_f equal to 4.0s. Using the same procedure as outlined in Case 1 and the same initial guess P^o used for Case 1, we obtain the following optimal parameters, $\mathbf{P}^*$= [*-0.0126 -0.0097 0.0091 0.0039 10.00 10.00 10.00 10.00*]^T. From Figs. 4.4 (b), one can see that since the maximum tracking error is *0.0012m*, there is a considerable improvement in the tracking performance. Hence, we have obtained a feasible trajectory in *4.0*s. The actual end-effector trajectory is shown in Fig. 4.4 (a) and the corresponding joint trajectories are shown in Fig. 4.4 (c). In Fig. 4.4 (d), the joint torques for joint 1 and joint 2 are shown. The joint torques are not saturated and are well below the actuator constraints. From the torque profiles, we see that the actuators are obviously not utilized to their full capacity.

4.6.3 Case 3:Determine the Minimum-Time Trajectory

In this case study, the goal is to obtain the minimum time trajectory using the optimal motion procedure described in Section 4.5. Since we have obtained a feasible trajectory in Case 2, we can proceed directly to Step 3 of the optimal motion planning procedure.

Step 3: From the results of Cases 1 and 2, one can notice the optimal controller parameters are almost identical to the initial guesses. This fact was also observed for other set of initial guesses. Therefore, to reduce the computational load, we dropped[2] the controller parameters from the optimization parameters vector P. The optimization parameter vector P is reduced to P_e which is given by the following equation:

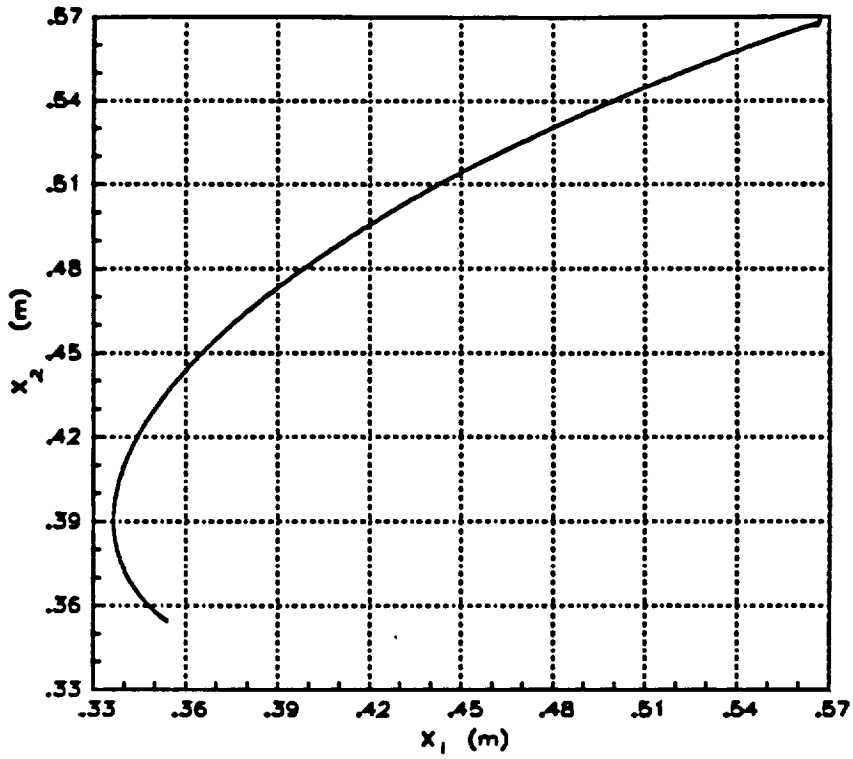
$$P_e = [a_{13} \;\; a_{14} \;\; a_{23} \;\; a_{24} \;\; t_f]^T. \tag{4.49}$$

Note that in above equation we include the total task time t_f as one of the optimization parameters.

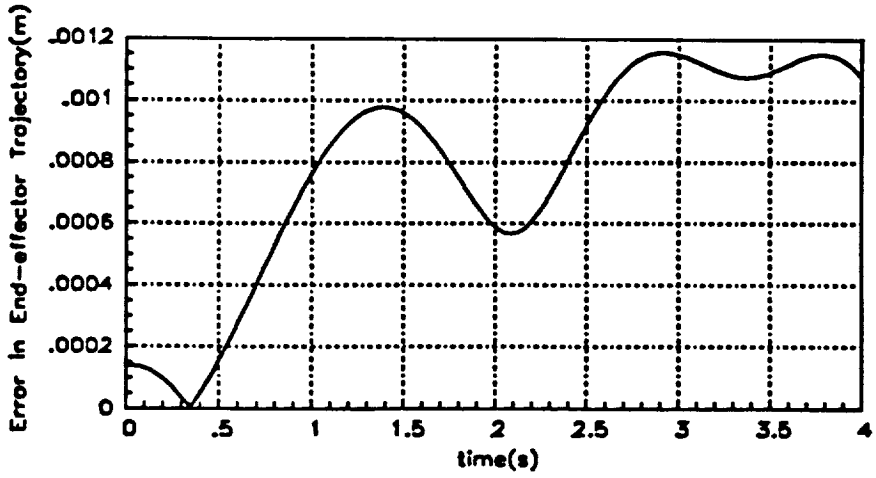
To obtain the minimum-time trajectory for the manipulator, we can pose the constrained optimization problem given in Eq. (4.42) with $I_2 = t_f$ and γ=0.003m:

---

[2]This practice is, of course, not valid in general.

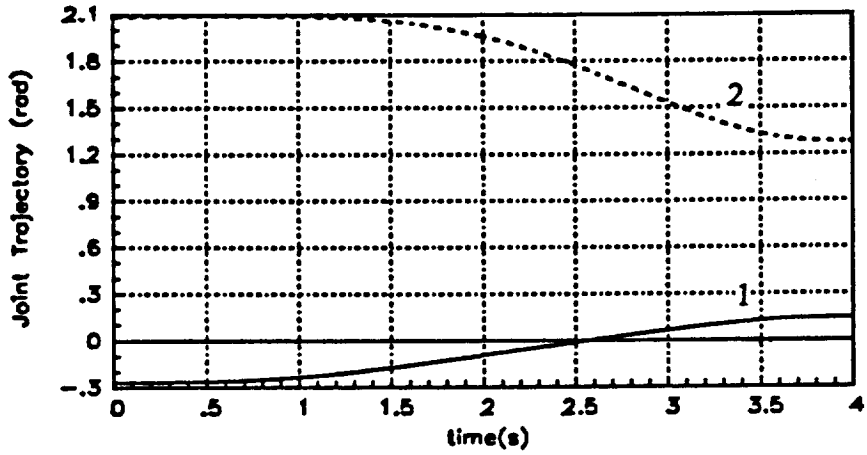


(a)

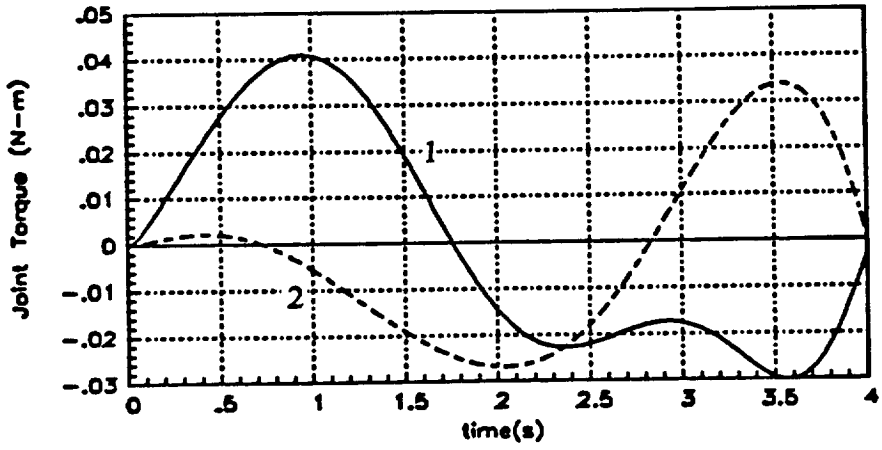


(b)

Fig. 4.4 Results of Case 2: (a) End-effector Trajectory (b) Tracking Error



(c)



(d)

Fig. 4.4(cont'd) (c) Joint Trajectory (d) Joint Torque

$$\begin{aligned} &\textit{minimize} \quad I_2 = t_f \\ \textit{subject to} \quad & \\ &I_1 \leq \gamma = 0.003m. \end{aligned} \tag{4.50}$$

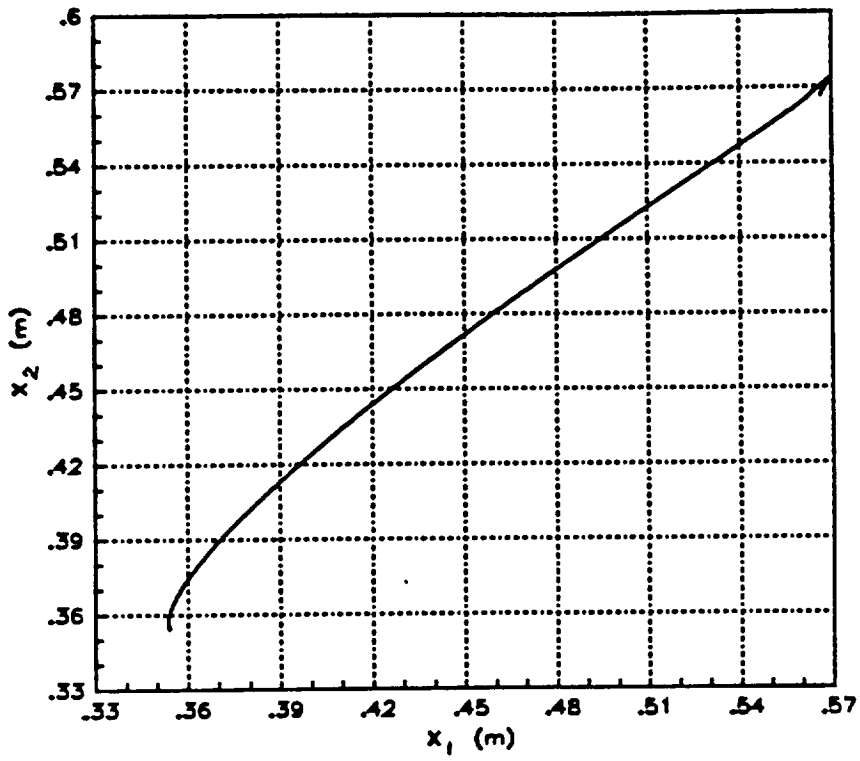
The optimal solution for the parameter vector $\boldsymbol{P}$ is given by $\boldsymbol{P}^* = \boldsymbol{P}_e = [-0.0078\ -0.0053\ 0.009\ 0.0044\ 3.0422]^T$. The optimal task time is equal to $3.0422s$. The simulation results for the minimum-time trajectory are shown in Fig. 4.5. The actual end-effector trajectory is shown in Fig. 4.5(a). We can see that there is a slight overshoot at the end of motion. From Fig. 4.5 (b), we can see that the maximum tracking error is 0.003m which occurs at the end-point of the trajectory and is equal to the maximum allowable tracking error.

The trade-off between I_1 and I_2 is shown in Fig. 4.6. From the trade-off curve, one can see the quantitative trade-off between the tracking performance I_1 and the task time I_2 for $\boldsymbol{u}_{max}$=[0.1 0.1 N-m]^T. As is to be expected, the task time can only be improved by sacrificing the tracking performance.

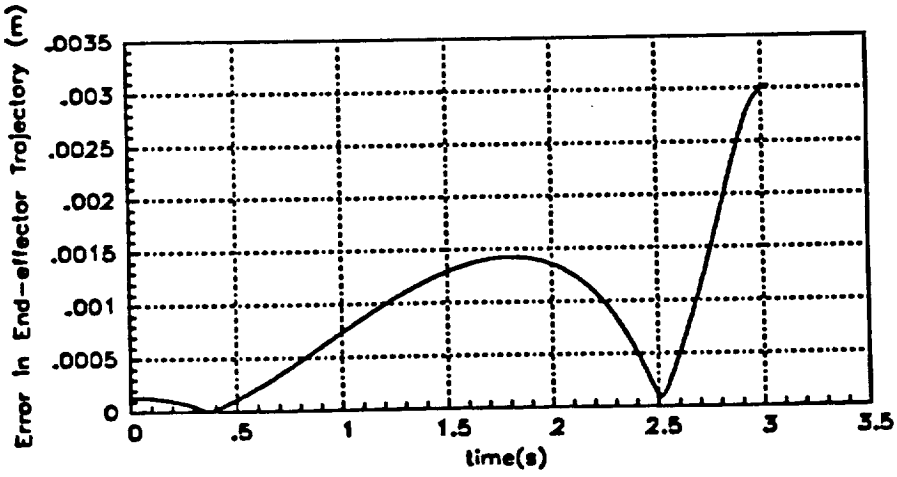
4.7 Summary

In this chapter, we have developed and applied the unified motion planning approach for planning the feasible and optimal motions for a non-redundant manipulator.

Using the (feedback-controlled) minimum-time trajectory example, we have shown how we can use the unified motion planning framework to pose and solve appropriate optimization problems for obtaining both the feasible and the optimal trajectory. The framework also provides a tool for the analyst to understand the trade-off between tracking performance and any other additional performance requirement. In the particular minimum time problem that we studied, one can understand the trade-off between tracking performance and the total task time for a constrained manipulator from the trade-off curve shown in Fig. 4.6. This type of quantitative understanding is necessary in a lot of robotic motion planning problems.

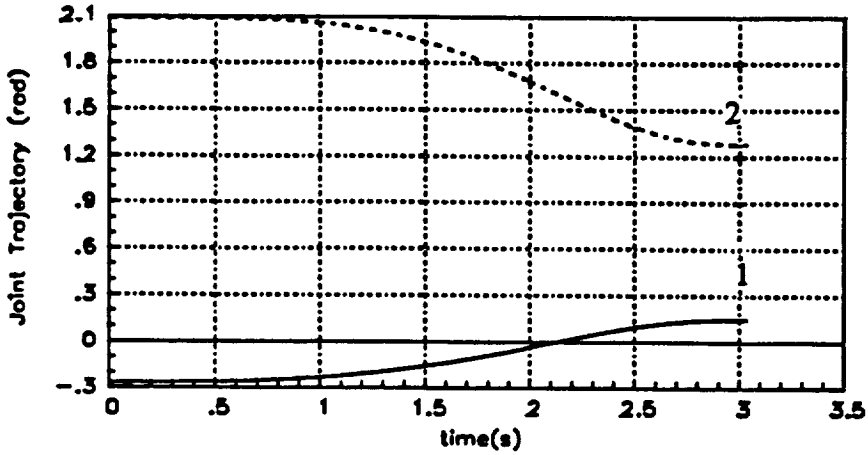


(a)

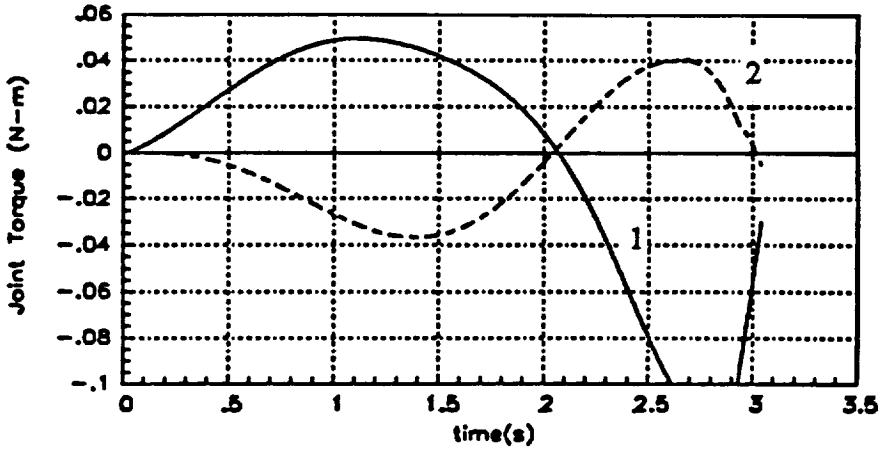


(b)

Fig. 4.5 Results of Case 3: (a) End-effector Trajectory (b) Tracking Error



(c)



(d)

Fig. 4.5(cont'd) (c) Joint Trajectory (d) Joint Torque

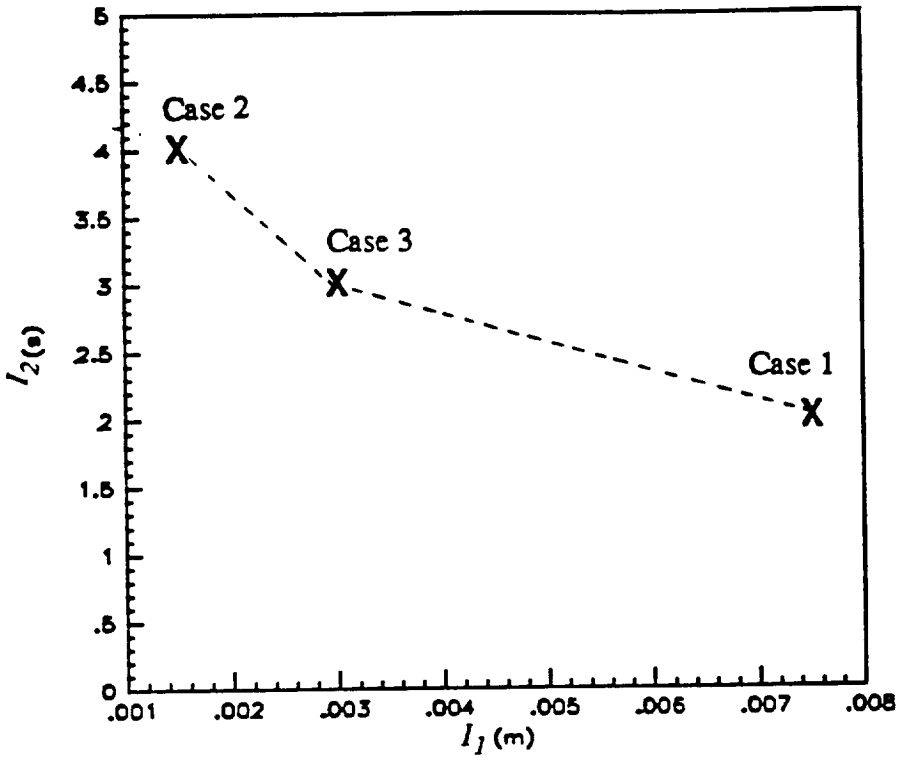


Fig. 4.6 Trade-Off Curve

Chapter 5

Motion Planning of Redundant Manipulators

5.1 Introduction

The ability of a redundant manipulator to accomplish an additional task has been the impetus for many redundant manipulator studies. In this chapter, we present a motion planning framework for a redundant manipulator which exploits the freedom in the joint-space trajectory and end-effector trajectory in order to optimize dynamic performance.

This chapter is organized as follows. In Section 5.2, we present and compare two different kinematic redundancy approaches - the partitioned Jacobian approach and the pseudo-inverse approach - and show why the latter approach is preferred. Then, in Section 5.3, we present a method of parameterizing the joint-space trajectory based on the pseudo-inverse approach. In Section 5.4, we will discuss the unified motion planning approach for redundant manipulators. In Section 5.5, we use the base reactions minimization problem to demonstrate the effectiveness of the unified approach in planning feedback-controlled feasible and optimal motions for a 3 d.o.f. manipulator with one degree of redundancy.

5.2 Redundancy Resolution Approaches

As seen in Chapter 2, a redundant manipulator with m degree-of-freedom (or m revolute joints) performing a task which requires n degrees of freedom, has $\rho=(m-n)$ degrees of redundancy. This means that for a given end-effector position there are an infinite number (${}_{\infty}\rho$) of joint-space solutions. Redundancy resolution refers to the process of selecting a joint-space solution from the ${}_{\infty}\rho$ possible joint-space solutions. Two of the most commonly used Redundancy Resolution Schemes - the Partitioned Jacobian

approach and the pseudo-inverse approach- will now be discussed.

5.2.1 Partitioned Jacobian Approach

The basic idea of the Partitioned Jacobian approach is to make use of the observation that in Eq. (2.4), if J is full rank, then $\rho=m-n$ of the joint variables, q_i, $(i=1,2,...,n)$, may be regarded as independent variables. These independent or free joint variables can be regarded as the elements of an ρ dimensional redundant joint velocity vector, $\dot{q}_r \in R^{m-n}$. The remaining m joint variables can be regarded as the elements of an m-dimensional non-redundant joint velocity vector, q_{nr}.

By dividing the joint variables into a non-redundant group and a redundant group, q can therefore be partitioned into:

$$q=\begin{pmatrix} q_{nr} \\ q_r \end{pmatrix}, \tag{5.1}$$

where $q_{nr} \in R^n$, is called the non-redundant joint vector and $q_r \in R^{m-n}$ is called the redundant joint vector.

We can then partition the Jacobian matrix J into the non-redundant Jacobian matrix (J_{nr}) and the redundant Jacobian matrix (J_r) such that the end-effector velocity $\dot{x}$ can be written in the following form:

$$\dot{x} = J_{nr}\ \dot{q}_{nr} + J_r\ \dot{q}_r \tag{5.2}$$

From the above equation, we can express the non-redundant joint velocity vector $\dot{q}_{nr}$ in terms of the redundant joint velocity vector $\dot{q}_r$:

$$\dot{q}_{nr} = J_{nr}^{-1}\ (\ \dot{x} - J_r\ \dot{q}_r\ ). \tag{5.3}$$

Eq. (5.3) is the basic redundancy resolution formulation for redundant manipulators and states that for a given end-effector velocity, the derivative of the redundant joint vector $\dot{q}_{nr}$ can be expressed as a unique function of the derivative of the redundant joint vector $\dot{q}_r$. Using the definition of q given in Eq. (5.1), the joint velocity, $\dot{q}$, can be written in a compact form as:

$$\dot{q}=\begin{pmatrix}\dot{q}_{nr}\\ \dot{q}_r\end{pmatrix}=\begin{pmatrix}J_{nr}^{-1}\dot{x}\\ 0\end{pmatrix}+\begin{pmatrix}-J_{nr}^{-1}J_r\\ I\end{pmatrix}\dot{q}_r. \tag{5.4}$$

In Eq. (5.4), the joint velocity vector is expressed as a function of the derivative of the redundant joint vector $\dot{q}_r$ and the end-effector velocity $\dot{x}$.

The expression that we have developed in Eq. (5.4) is based on the velocity relationship. An alternative form can be obtained by using the acceleration relationship. Taking the time-derivative of Eq. (5.2), the acceleration of the end-effector is given by

$$\ddot{x} = \dot{J}_{nr}\ \dot{q}_{nr} + \dot{J}_r\ \dot{q}_r + J_{nr}\ \ddot{q}_{nr} + J_r\ \ddot{q}_r. \tag{5.5}$$

From the above expression, we can then express the non-redundant joint acceleration vector $\ddot{q}_{nr}$ in terms of the redundant joint velocity $\dot{q}_r$ and redundant joint acceleration vector $\ddot{q}_r$:

$$\ddot{q}_{nr} = J_{nr}^{-1}\ (\ \ddot{x} - \dot{J}_r\ \dot{q}_r - \dot{J}_{nr}\ \dot{q}_{nr} - J_r\ \ddot{q}_r\ ). \tag{5.6}$$

The non-redundant joint acceleration vector $\ddot{q}_{nr}$ can also be written in the form

$$\ddot{q}_{nr}=J_{nr}^{-1}(\ddot{x}-\dot{J}\dot{q}-J_r\ddot{q}_r). \tag{5.7}$$

Using Eq. (5.1) and Eq. (5.6), the joint acceleration vector $\ddot{q}$ can be expressed in a compact form:

$$\ddot{q}=\begin{pmatrix}J_{nr}^{-1}[\ddot{x}-\dot{J}\dot{q}]\\ 0\end{pmatrix}+\begin{pmatrix}-J_{nr}^{-1}J_r\\ I\end{pmatrix}\ddot{q}_r \tag{5.8}$$

One can resolve redundancy using either Eq. (5.4) and (5.8). However, both Eq. (5.4) and Eq. (5.8) require that the matrix J_{nr} be invertible. Since J_{nr}^{-1} is not invertible for so-called singular configurations of a manipulator, one must identify all of these singular configurations in order to "avoid" them. Identifying all possible singular positions is extremely difficult if the manipulator has many degrees of redundancy and is even more so if the manipulator is spatial. Due to this shortcoming, even though the partitioned

Jacobian approach is conceptually very simple, it is not easy to implement. In the next section, we will discuss a more convenient approach, the pseudo-inverse approach.

5.2.2 Pseudo-Inverse Approach

One method [18,20,23,26,28] of determining the joint velocity vector $\dot{q}$, based on the pseudo-inverse matrix J^+ of the Jacobian matrix J (see Eq. (2.4)), is as follows:

$$\dot{q} = J^+ \dot{x} + (I - J^+ J)k \tag{5.9}$$

where $k \in R^m$ is an arbitrary vector, $I \in R^{m \times m}$, $J \in R^{n \times m}$ and $J^+ \in R^{m \times n}$.

Substitution of the expression for $\dot{q}$ given by Eq. (5.9) into Eq. (2.4) shows that the right-hand-side of Eq. (5.9) is indeed a solution of Eq. (2.4). (Remember that $JJ^+=I$). Let us interpret the physical meaning of the solution given by Eq. (5.9). The first term, on the right-hand-side of Eq. (5.9) gives the joint velocity component that would produce the desired velocity in the task space, i.e. $JJ^+ \dot{x} = I\dot{x} = \dot{x}$. In Eq. (5.9), $(I-J^+J)$ is a projection matrix that projects any vector $k \in R^m$ onto the nullspace $N(J)$ of the Jacobian matrix, i.e., $J(I-J^+J)k=0$. Since the projection matrix maps any arbitrary k onto the nullspace $N(J)$, we call the second term the nullspace solution for the joint velocity.

The dimension of the nullspace $N(J)$ is $m-r$ where r is the rank of J. In order for a manipulator to track a general curve in n space, J must be full rank, i.e. $r=n$. Therefore the dimension of the nullspace $N(J)$ is, in general, (m-n).

If we let the i^{th} column of $(I-J^+J)$ be ψ_i, $(i=1, 2, ..., m)$, Eq. (5.9) can be written as

$$\dot{q} = J^+ \dot{x} + \sum_{i=1}^{m} \psi_i k_i, \tag{5.10}$$

where k_i is the i^{th} element of k. Each vector ψ_i lies in the nullspace $N(J)$ of J.

Since there are m elements in k, one might conclude that there are m independent variables. However, since the dimension of the nullspace $N(J)$ is $m-n$, only $(m-n)$ of the column vectors, ψ_i, $(i=1,2, ..., n)$ are linearly independent. If ψ_i, $(i=1,2, ..., n)$ is ordered such that the first $(m-n)$ vectors, i.e., ψ_i, $(i=1, 2, ..., m-n)$, are linearly independent, then Eq. (5.10) can be written as

$$\dot{q}=J^{+}\dot{x}+\sum_{i=1}^{m-n}\psi_i s_i \tag{5.11}$$

where s_i are arbitrary coefficients. Note that in above equation there are only m-n independent variables s_i, $(i=1,2,...,m-n)$.

If we let Φ denote an mx$(m-n)$ matrix whose i^{th} column is ψ_i, $i=1,...,(m-n)$ and s denotes a column vector with elements s_i, $i=1,..,(m-n)$, we can write Eq. (5.11) as

$$\dot{q}=J^{+}\dot{x}+\Phi s \tag{5.12}$$

From Eq. (5.12), one can see that for a manipulator with one degree of kinematic redundancy, Φ consists of only one column vector ($\Phi = \psi_1$) and one coefficient s_1, due to the fact that the dimension of the nullspace $N(J)$ is equal to one. Therefore, to represent the basis for $N(J)$, one can arbitrarily choose a single column vector from the m columns ψ_i, $(i=1,2, .., m-n)$. We can also see that the scalar s_1 expresses the freedom in the joint trajectory (for the one degree of kinematic redundancy case).

In a lot of applications, the joint acceleration vector $\ddot{q}$ is often required in the computation of the performance index. The joint acceleration $\ddot{q}$, obtained by taking the time-derivative of $\dot{q}$ in Eq. (5.9), is given by the following expression:

$$\ddot{q}=J^{+}\ddot{x}+\dot{J}^{+}\dot{x}+\dot{W}k+W\dot{k}, \tag{5.13}$$

where W is $(I-J^{+}J)$. As shown in Appendix C for a 4 d.o.f. planar manipulator, the computation of $\ddot{q}$ using Eq. (5.13) can be rather laborious. The difficulty of computing $\ddot{q}$ arises from the fact that in order to compute the time-derivative of the pseudo-inverse J^{+}, ($J^{+} = J^{T}(JJ^{T})^{-1}$), we have to determine the time-derivative of $(J\,J^{T})^{-1}$ which is not a trivial task. To get around this problem, in our unified motion planning approach, the joint acceleration vector is computed from the equations of motion (Eq. (4.28)) instead.

In some studies [18,40], the redundancy is resolved at the acceleration level instead of at the velocity level. The advantage of this approach is that in applications where $\ddot{q}$ is needed in the computation of the performanc index, the joint acceleration vector $\ddot{q}$ can be

obtained directly through the use of the pseudo-inverse of the Jacobian matrix.

Taking the time-derivative of Eq. (2.4), the acceleration of the end-effector trajectory is obtained as:

$$\ddot{\boldsymbol{x}} = \boldsymbol{J}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}\dot{\boldsymbol{q}}. \tag{5.14}$$

From Eq. (5.14) the pseudo-inverse solution for $\ddot{\boldsymbol{q}}$ is then given by

$$\ddot{\boldsymbol{q}} = \boldsymbol{J}^{+}(\ddot{\boldsymbol{x}} - \boldsymbol{J}\dot{\boldsymbol{q}}) + (\boldsymbol{I} - \boldsymbol{J}^{+}\boldsymbol{J})\boldsymbol{k}. \tag{5.15}$$

In the above equation, the joint acceleration vector consists of two terms: the first term is used to maintain the desired acceleration of the end-effector trajectory; the second term is the nullspace solution for the joint acceleration that expresses the freedom arising from kinematic redundancy. It is worth noting that the same projection matrix used in Eq. (5.9) is used to project vector $\boldsymbol{k}$ to the nullspace of the Jacobian matrix. Therefore, the dimension of the null-space term (the second term in Eq. (5.15)) is still m-n. The joint acceleration $\ddot{\boldsymbol{q}}$ can be integrated numerically to obtain the joint velocity $\dot{\boldsymbol{q}}$ and the joint trajectory $\boldsymbol{q}$ once the initial joint variables and the initial joint velocities are known.

One disadvantage of resolving redundancy at the acceleration level is that in our joint trajectory parameterization scheme (to be discussed in Section 5.3), it is unclear how one would impose appropriate boundary conditions on the free variables $\boldsymbol{k}$ to achieve a desired end-effector motion. However, using the joint velocity equation (Eq. (5.9)) and the joint acceleration (Eq. (5.13)) derived from the velocity equation, we will be able to determine the appropriate boundary conditions on $\boldsymbol{k}$ very easily. In the next section, we will discuss the joint trajectory parameterization scheme based on the joint velocity expression (Eq. (5.9)).

5.3 Parameterization of Joint Trajectories

In the parameterization of the joint-space trajectories, the joint trajectories $\boldsymbol{q}(t)$, $(0 \le t \le t_f)$, are represented by polynomial functions with unknown coefficients. With these functions, we are able to describe a class of joint trajectories that satisfy the task specifications. Mathematically, we can then express the joint trajectory as

$$q(t)=h(t,P_j,x(t)),\quad 0\le t\le t_f, \tag{5.16}$$

where P_j is the joint trajectory parameter vector, x is a desired end-effector trajectory and t is the time.

In a Type I specification, $x(t)$ is a vector of prescribed functions, whereas in a Type II and Type III specification, $x(t)$ is a function of the end-effector trajectory parameter vector P_e. Hence, for both Type II and III trajectories, we can write:

$$x(t)=f(t,P_e). \tag{5.17}$$

In view of Eqs. (5.16) and (5.17), we observe that for a Type II and a Type III end-effector trajectory, $q(t)$ is a function of both the end-effector parameter vector P_e and the joint-space parameter vector P_j.

The first step of our joint trajectory parameterization scheme is to use the joint velocity equation given by Eq. (5.9) and parameterize k with some appropriate functions. To obtain a set of representation functions for k, we first need to determine the suitable boundary conditions for k. For redundant manipulators, ensuring that the end-effector trajectory satisfies the boundary conditions is not sufficient because a stationary end-effector does not necessarily imply that the links are at rest. Therefore, it is also necessary to ensure the joint trajectory q satisfies the zero boundary conditions, i.e. $\dot{q}(0) = \dot{q}(t_f) = \ddot{q}(0) = \ddot{q}(t_f) = 0$.

In view of Eq. (5.9), if

$$k(t)=0 \quad \text{at } t=0 \text{ and } t=t_f. \tag{5.18}$$

then $\dot{q}(0) = \dot{q}(t_f) = 0$ for $\dot{x}=0$.

Similarly, in view of Eqs. (5.13) and (5.18) and if in addition, we let

$$\dot{k}(t)=0 \quad \text{at } t=0 \text{ and } t=t_f, \tag{5.19}$$

then $\ddot{q}(0) = \ddot{q}(t_f) = 0$ for $\dot{x}=\ddot{x}=0$. Therefore, the boundary conditions for k can be expressed by Eqs. (5.18) and (5.19). To obtain a parameterized expression for k, we can represent $k_i(t)$, $(i=1,...,m)$, by the polynomial

$$k_i(t)=\sum_{j=0}^{l} a_{ij} t^j. \tag{5.20}$$

In order to satisfy the $4m$ boundary conditions (given by Eqs. (5.18) and (5.19)), the coefficients a_{i0}, a_{i1}, $a_{i(l-1)}$, and a_{il} $(i=1,2,...,m)$ must satisfy the following equations:

$$\begin{aligned} a_{io}&=a_{i1}=0 \\ a_{i(l-1)}&=\frac{\sum_{i=2}^{l-2}(i-l)a_i t_f^i}{t_f^{l-1}} \\ a_{il}&=\frac{\sum_{i=2}^{l-2}(l-1-i)a_i t_f^i}{t_f^l}. \end{aligned} \tag{5.21}$$

The remaining variables a_{i2}, a_{i3}, ..., $a_{i(l-2)}$, $(i=1,2,...,m)$ are the free variables one can use to represent a class of joint trajectories that satisfy the boundary conditions (Eqs. (5.18) and (5.19)). Let P_j denote the vector whose elements are the $m(l-3)$ free variables , i.e.

$$P_j=[a_{12}, a_{13}, ..., a_{1(l-2)}, ..., a_{l2}, a_{l3}, ..., a_{l(l-2)}]. \tag{5.22}$$

P_j represents the freedom in the joint trajectory q for a given end-effector trajectory x.

We can take further advantage of kinematic redundancy by treating some of the joint variables which specify the initial configuration of the manipulator as independent variables. If a manipulator has $(m-n)$ degrees of redundancy and we let σ_1, σ_2, ..., σ_{m-n} denote the independent initial configuration parameters then the vector P_j can be defined as follows:

$$P_j=[a_{12}, ..., a_{1(l-2)}, ..., a_{l2}, ..., a_{l(l-2)}, \sigma_1, ..., \sigma_{m-n}]. \tag{5.23}$$

To determine the dependent initial joint variables in terms of the $(m-n)$ independent initial joint variables for a given end-effector position one must use the so-called Inverse Kinematic Equations for the manipulator. These equations are given in Appendix D for a 3 d.o.f. and a 4 d.o.f. planar manipulator.

5.4 Motion Planning for Redundant Manipulators

In chapter 4, we have described a motion planning procedure for obtaining the feasible trajectory and optimal trajectory for non-redundant manipulators. The procedure described in Chapter 4 for obtaining the feasible trajectory and optimal trajectory for non-redundant manipulators can be easily extended to the motion planning problem of redundant manipulators.

In this section we describe how the procedure described in Section 4.4 for obtaining the feasible trajectory must be modified in order to handle redundant manipulators. The procedure for obtaining the optimal trajectory, given in Section 4.5, are directly applicable to redundant manipulators.

The first two steps for obtaining the feasible trajectory (in Section 4.4) deal with the parameterization of the freedom available in the end-effector trajectory. These steps are the same for the redundant case.

In Step 3, the pseudo-inverse approach is used for redundancy resolution. The joint velocity $\dot{q}_d$ can then be represented as

$$\dot{q}_d = J^+ \dot{x}_d + (I - J^+ J)k.$$

We represent k by means of the polynomial functions defined by Eq. (5.20) and Eq. (5.23).

For a manipulator with one degree of redundancy, the dimension of the nullspace $N(J)$ is equal to 1. We can then use following simpler equation for determining $\dot{q}_d$:

$$\dot{q}_d = J^+ \dot{x}_d + \phi_1 s_1, \tag{5.24}$$

where ϕ_1 is the first column of the projection matrix $(I\text{-}J^+J)$ and s_1 can be represented in a fashion similar to Eq. (5.20).

Steps 4 and 5 of the procedure, which deal with prescribing the state-space model of the manipulator and the appropriate control strategy are the same for the redundant case.

For the redundant case, the optimization parameter in Step 6 is defined as follows:

$$P=[P_e^T \ P_j^T \ P_c^T]^T. \tag{5.25}$$

where P_e is the end-effector trajectory parameter vector, P_j is the joint trajectory parameter vector, and P_c is the controller parameter vector.

Application of Step 7 of the procedure in Section 4.4 to the present problem would then yield the optimal value of P which in turn yields the feasible trajectory and the corresponding feedback control strategy.

5.5 Illustrative Example : Base Reaction Minimization

In section 5.4, we have discussed the unified motion planning approach for redundant manipulators. In this section, we will demonstrate the procedure for obtaining a motion plan that optimizes a specified dynamic performance. The problem that we address is the minimization of the magnitude of the reactions transmitted to the base of a manipulator used in space.

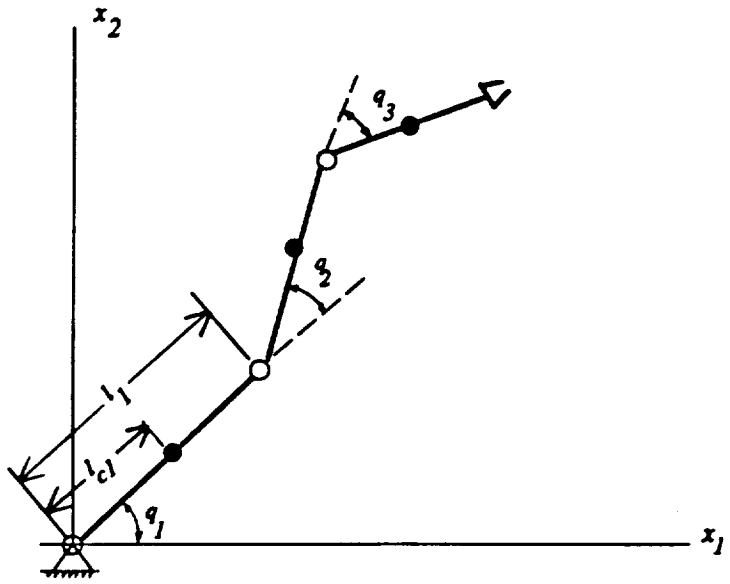


Fig. 5.1 A 3 d.o.f. Planar Manipulator

We propose to minimize the magnitude of the base reaction for the 3 d.o.f. manipulator shown in Fig. 5.1. The mass properties and link dimensions of the manipulator are given in Table 5-1.

	Link 1	Link 2	Link 3
m_i [kg]	1.0	0.5	0.5
l_i [m]	0.5	0.25	0.25
l_{ci} [m]	0.25	0.125	0.125
I_{ci} [kg-m^2]	0.005	0.0026	0.0026

Table 5-1 Mass Properties and Link Dimensions

The task of the 3 d.o.f. redundant manipulator is to track a completely specified straight-line trajectory described by a Type I specification using a PD controller. The end-effector trajectory $x(t)$ can be expressed as follows:

$$\mathbf{x}(t)=\alpha(t)\mathbf{e}, \quad 0\le t\le t_f,$$

where

$$\alpha(t)=b(t-\frac{t_f}{2\pi}sin(\frac{2\pi t}{t_f})),$$
$$b=\frac{\alpha(t_f)}{t_f}, \tag{5.26}$$

$\mathbf{e}$ is a unit vector parallel to the straight-line trajectory and t_f is the total time of the task. The speed $v(t)$ and the acceleration $a(t)$ of the end-effector trajectory are given by the following equations:

$$v(t)=b(1-cos(\frac{2\pi t}{t_f}))$$
$$a(t)=\frac{\pi b}{t_f}sin(\frac{2\pi t}{t_f}), \tag{5.27}$$

which ensure zero boundary conditions on the velocity and acceleration of the end-effector.

The initial position of the end effector is x_0=(0.35355, 0.35355)^T (m). The desired final position is x_f= (0.5664, 0.5664)^T(m). The following parameters are used for the trajectory:

$$\begin{aligned} t_f &= 2.0s \\ \alpha(t_f) &= 0.3m \\ b &= 0.15m/s. \end{aligned} \tag{5.28}$$

The tracking error of the end-effector trajectory cannot exceed the maximum allowable tracking error γ which is chosen to be 1% of the total distance traversed by the end-effector. Therefore, we can constrain tracking performance index I_1 by the following inequality:

$$I_1 \leq \gamma, \tag{5.29}$$

where γ=0.003m. The maximum torque available at each actuator is *0.15N-m.*

According to the optimal trajectory framework developed in Section 4.4, we will first determine whether a feasible trajectory can be obtained for the given task specifications. If the task specifications are not feasible, then we change the task specifications such that a feasible trajectory may be obtained. Once a feasible trajectory is achieved, then we can proceed to obtain an optimal motion plan by using the procedure described in Section 4.5.

Based on the above rationale, we will investigate the following two cases:

- Case 1. Finding a feedback-controlled feasible trajectory with u_{max} = [*0.15,0.15,0.15*]^T (N-m) which satisfies Eq. (5.29).
- Case 2. Obtaining a feedback- controlled optimal trajectory to minimize the peak magnitude of the base force.

The results of the above two cases are tabulated in Table 5-2.

5.5.1 Case Study 1: Obtaining a Feasible Trajectory

In this case study, we want to obtain a feasible trajectory under the task specifications given by (5.28), (5.29) and the specified actuator constraints.

Using the procedure outlined in Section 4.6.1, we can obtain the optimal control

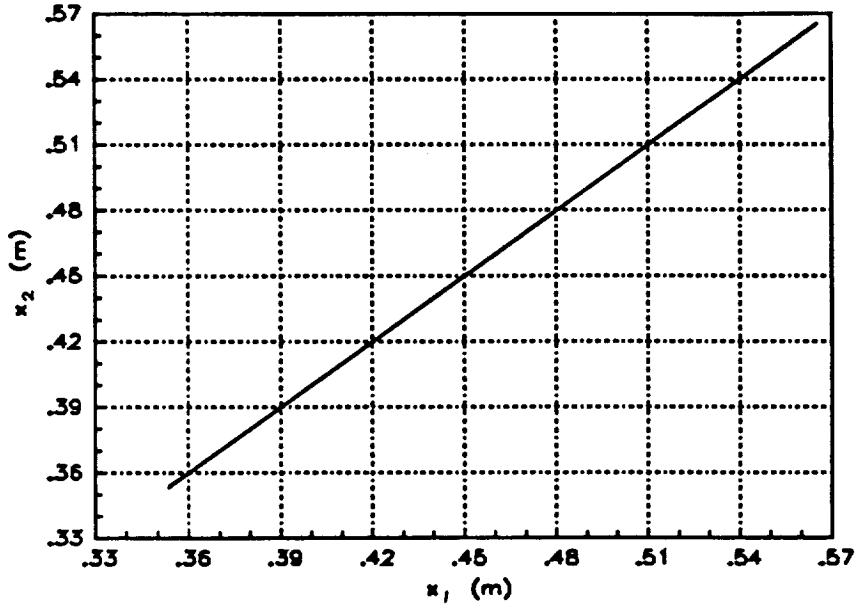
Case	Problem	Results	Summary
1	• ***Find a feasible trajectory:*** $\min I_1$ with $u_{max}=[0.15\ 0.15\ 0.15]^T$(N-m).	• Acceptable tracking performance is obtained. I_1=0.0006m (see Fig. 5.2 (a,b)) • Actuator torques do not violate the actuator constraints. • peak torque = 0.078N-m. (see Fig. 5.2 (d))	• A feasible trajectory was obtained.
2	• ***Obtain an optimal trajectory to minimize base force:*** $\min I_2 = I_f$ subject to $I_1 \leq \gamma$ γ=0.003m	• Peak tracking error=0.0017m. (see Fig. 5.3 (b)) • Optimal peak base force=0.29N (see Fig. 5.3 (c)) • actuator torques are below u_{max}.	• An optimal trajectory was obtained that minimizes the peak magnitude of base force by sacrificing the tracking performance.

Table 5-2 Summary of Results

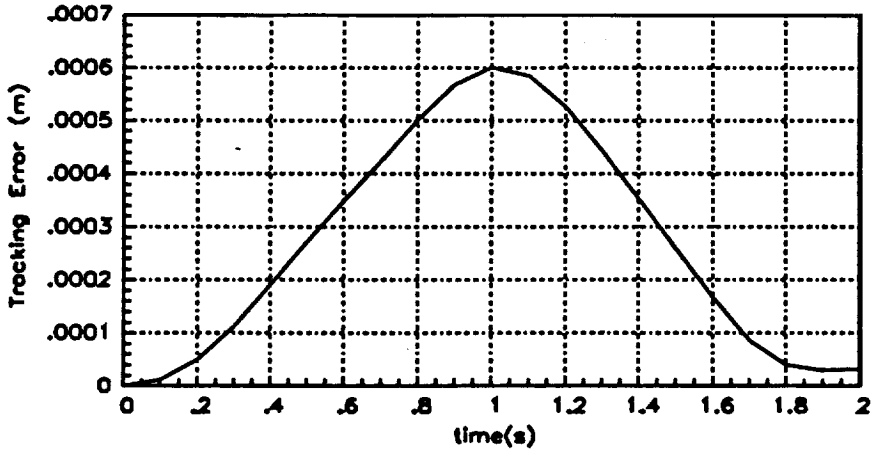
parameter vector P_c^* and optimal joint trajectory parameter vector P_j^* for the tracking performance optimization problem posed in Eq. (4.38) with $u_{max}=[0.15\ 0.15\ 0.15]^T$ (N-m). To ensure that a local minimum solution was not obtained, several initial values of P were used to obtain P^*.

The optimal solution P^* was given by: $P^* = [P_j^{*T}\ P_c^{*T}] = [-7.999\ 3.009\ 4.003\ 1.0312\ 1.0018\ 0.9985\ 1.0027\ 19.999\ 20.000\ 20.000]^T$. The results for this case study are shown in Figs. 5.2 (a-d).

The actual end-effector trajectory and the tracking errors are shown in Figs. 5.2 (a,b). From these results, we observe that the maximum tracking error (0.0006m) is less than the maximum allowable tracking error (0.003m). Therefore, the trajectory is acceptable and it is a feasible trajectory. Furthermore, as shown in Fig. 5.2(d), the joint torques are all below the actuator constraints. This shows that our motion planning framework is capable of finding a feasible trajectory to satisfy the tracking performance requirement and the actuator constraints. Also, in Fig. 5.2(c), the joint trajectories are plotted. Note that the angular displacement q_1 of the first joint variable is very small relative to the angular displacement of either the joint variable q_2 or q_3.

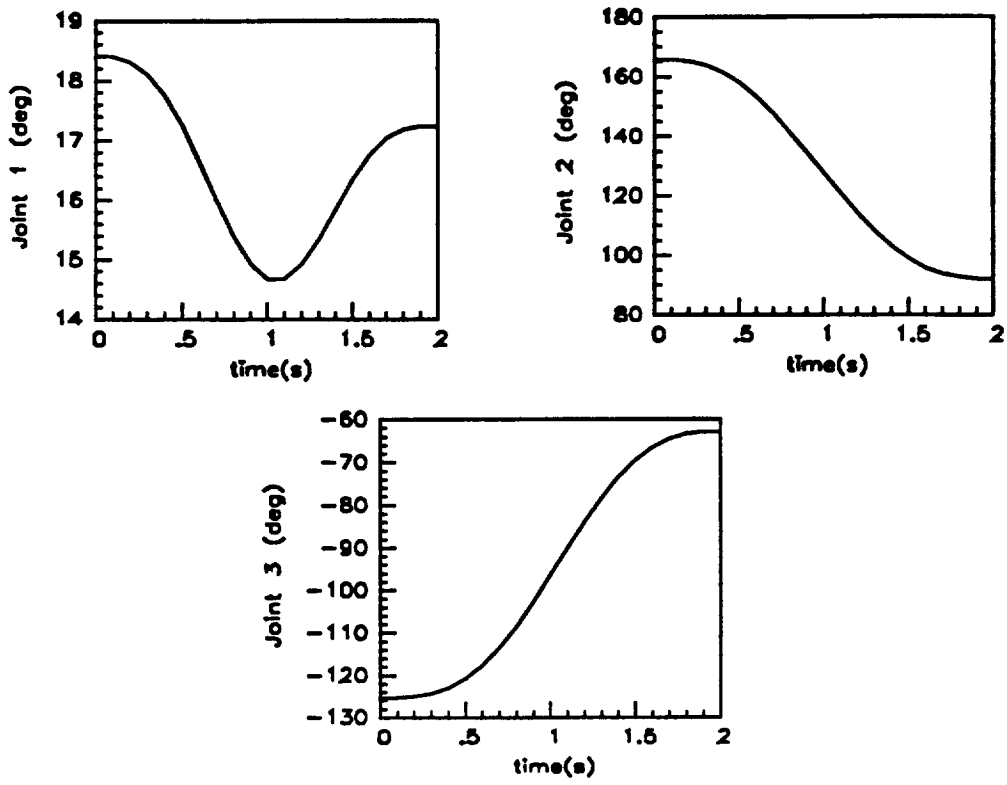


(a)

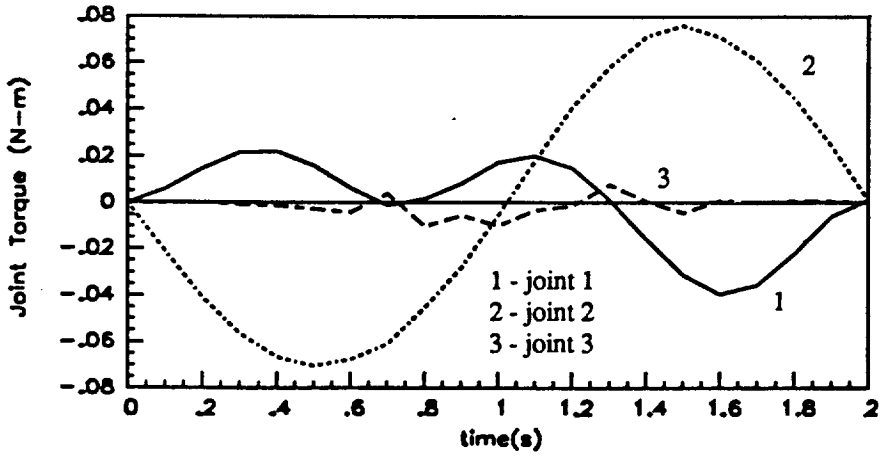


(b)

Fig. 5.2 Results of Case 1: (a) End-Effector Trajectory (b) Tracking Error



(c)



(d)

Fig.5.2(cont'd) Results of Case 1: (c)Joint Trajectory (d)Joint Torque

5.5.2 Case Study 2: Obtaining Minimum Base Reactions Trajectory

Recently, redundant manipulators such as the Oak-Ridge arm, which has seven degrees of freedom (one degree of kinematic redundancy) have been proposed to be used in space to provide dexterity and obstacle avoidance ability [21]. In using a manipulator in microgravity environments, one has to consider the problem of minimizing the magnitude of the force and the moment exerted by a manipulator on its base as it performs a task. One reason for minimizing, and if possible eliminating, the base reactions is that high base forces and moments could disturb other tasks or experiments in the vicinity of the manipulator. The problem that we are interested in differs from the study by Longman, [25], in which the base reactions are compensated (instead of being minimized) by modifying the desired joint-space trajectories using a special set of kinematics equations that account for base motion caused by the base reactions.

The task of a redundant manipulator used in space can be divided into a primary goal and a secondary goal. The primary goal of a manipulator is to accomplish the end-effector motion specified by the user. While the manipulator tracks the trajectory, base force and base moment denoted by f_0 and n_0 are transmitted to the base. Since there is freedom in the joint-space trajectory, we can exploit this freedom to accomplish the secondary goal - reducing the magnitudes of the base reactions. The dynamic equations which describe the forces and moments transmitted to the base of a manipulator are given in Appendix B.

In this case study, we will obtain an optimal motion plan that minimizes the magnitude of the base reactions while satisfying the tracking requirement.

To avoid undesired large magnitudes of base force and base moment, we will use the following performance index as a measure of the magnitudes of the base reactions:

$$I_2 = w_1 I_f + w_2 I_m, \tag{5.30}$$

where w_1 and w_2 are the weighting factors and I_f and I_m are defined as follows:

$$I_f = max \ (|f_0|)$$
$$I_m = max \ (|n_0|). \tag{5.31}$$

In the above equations, I_f and I_m are the peak magnitude of the base force f_0 and the peak magnitude of the base moment n_0, respectively. In this example we will study the problem of minimizing the peak magnitude of the base force. Therefore, the weighting factors in Eq. (5.30) are w_1=1.0 and w_2=0.0.

With the above performance index (Eq. (5.30)), one can then pose and solve the following optimization problem to obtain the optimal motion plan:

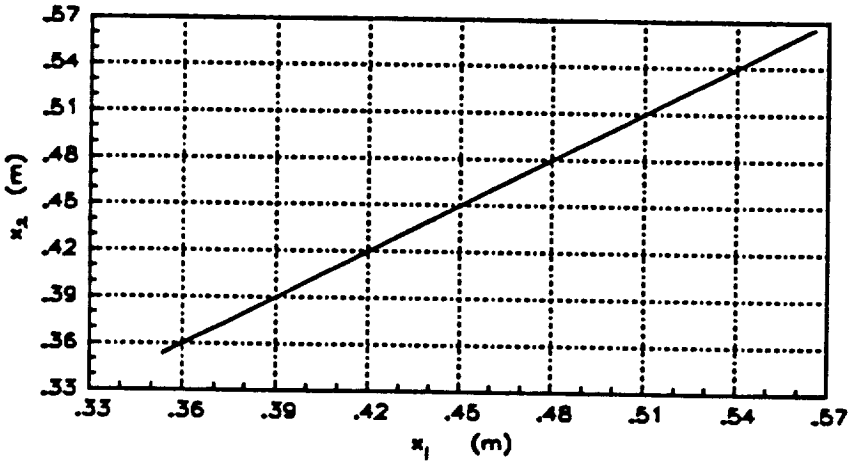
$$\begin{aligned} & min \;\; I_2(P)=I_f \\ \text{subject to:} & \\ & I_1(P)<\gamma=0.003m. \end{aligned} \tag{5.32}$$

The optimal solution P^* of the above optimization problem yields the optimal trajectory and corresponding feedback control strategy. The simulation results are shown in Fig. 5.3 (a-d).

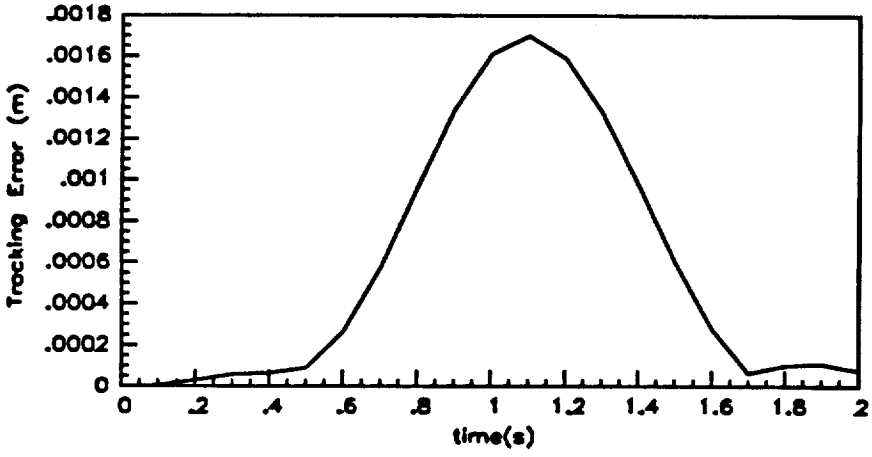
The actual end-effector trajectory and the magnitude of the tracking error are shown in Fig. 5.3 (a,b). From these results we observe that the maximum magnitude of the tracking error is *0.0018m* which is larger than that of the feasible trajectory (obtained in Case 1). The magnitude of the base force of the optimal trajectory and the feasible trajectory are compared in Fig. 5.3 (c). The dotted line is the magnitude of the base force of the feasible trajectory and the solid line is the magnitude of the base force of the optimal trajectory. One can see that the peak magnitude of the base force of the optimal trajectory is *30%* less than that of the feasible trajectory. Furthermore, as shown in Fig. 5.3 (d), the joint torques are all below the actuator constraints. The maximum magnitude of the actuator torque is *0.12N-m*.

5.6 Summary

In this chapter we demonstrated the application of the unified approach to planning motions for non-redundant manipulators. The Pseudo-Inverse Redundancy Resolution Approach was found to be particular well suited for our motion-planning framework. Finally we showed how the unified approach could be used to plan feedback-controlled motions which minimize the magnitude of the reactions transmitted to the base of a manipulator.

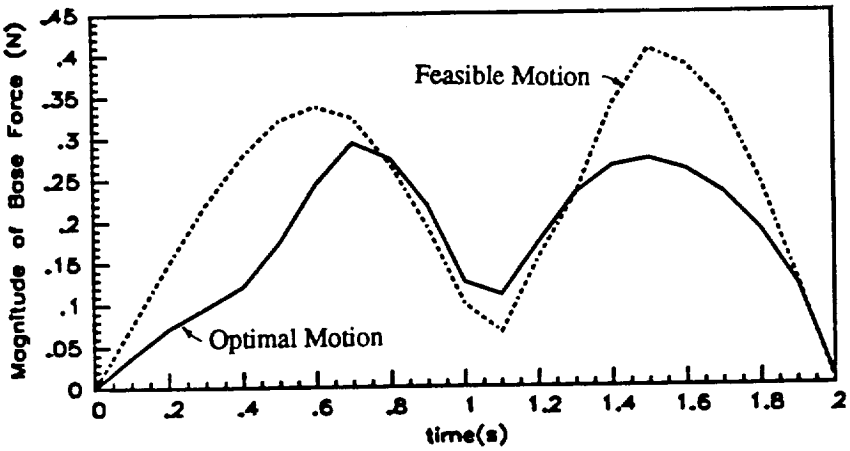


(a)

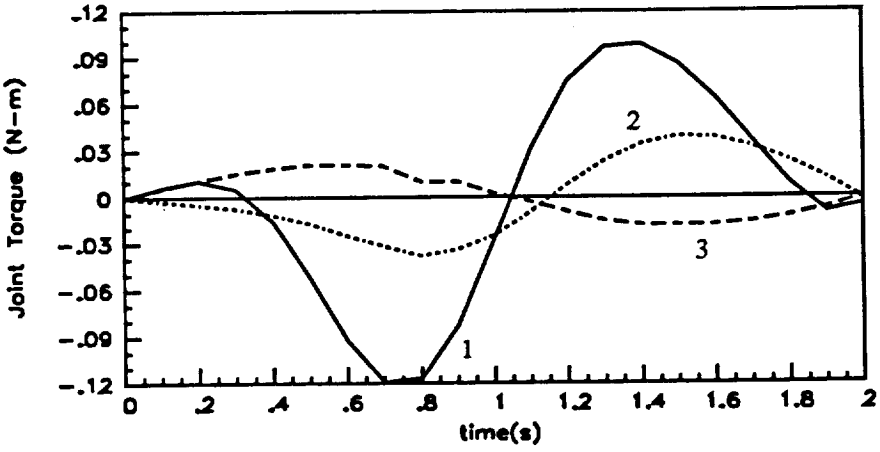


(b)

Fig. 5.3 Results of Case 2: (a) End-Effector Trajectory (b) Tracking Error



(c)



(d)

Fig.5.3(cont'd) Results of Case 2: (c) Base Force (d)Joint Torque

Chapter 6

Special Topics

6.1 Introduction

In this section, we will discuss some special topics related to motion planning of manipulators. In Section 6.2, we develop and apply a procedure for evaluating the effectiveness of kinematic redundancy. In Section 6.3, we examine some of the implementation issues which are important in motion planning such as sensitivity, local minima, and the number of parameters needed to parameterize a trajectory.

6.2 Evaluation of Effectiveness of Kinematic Redundancy

We first establish the need for defining suitable compatibility criteria for comparing the performance of alternative manipulator types. These compatibility criteria form the basis for a procedure which can be used to systematically evaluate the effectiveness of kinematic redundancy. The procedure is then applied to show that kinematic redundancy does in fact minimize base reactions.

6.2.1 Is kinematic redundancy useful in minimizing base reactions?

In the literature, many studies have been conducted to explore the utility of kinematic redundancy in various applications. In most of these studies, no attempts have been made to evaluate the effectiveness of using kinematic redundancy to improve the dynamic performance of a manipulator. In fact, most of these studies are conducted based on the following scenario: *For a given redundant manipulator, find the optimal joint trajectory that achieves the primary goal of tracking a specified end-effector trajectory and an additional secondary goal.*

A basic question which must be answered in using redundant manipulator is the following: Is the addition of redundant degrees-of-freedom to a kinematic structure really beneficial? As stated before, the advantage of using redundant manipulators lies in the fact that there are an infinite number of joint trajectories for a given end-effector trajectory. However, one cannot overlook the following trade-offs of adding more degrees of freedom. The complexity of the trajectory planning problem increases as the degree of redundancy increases. With additional hardware in the form of links, motors, and sensors, the overall system complexity and cost are increased. In view of these trade-offs, there is a need to justify the effectiveness of a redundant manipulator.

In the following example, we will illustrate some of the issues one has to consider when evaluating the performance of a redundant manipulator.

Assume that we are evaluating the design of a planar 3 d.o.f. manipulator with the following physical dimensions: $l_1=l_2=l_3=0.5m$ and $m_1=m_2=m_3=1.0kg$. The issue of interest is whether the use of a 3 d.o.f redundant manipulator (with one degree of redundancy) is more effective in reducing base reactions than a 2 d.o.f. non-redundant manipulator. Using the framework we developed in Chapters 4 and 5, we can obtain the optimal base reaction for the redundant manipulator. But, the following two equivalent questions would still remain unanswered: (1) Do the optimal base reactions tell us anything about the effectiveness of using kinematic redundancy in minimizing the magnitude of the base reactions? (2) If we add a degree of redundancy to a non-redundant manipulator, will it help reduce the magnitude of the base reactions?

To answer the questions just raised, one might try to compare the performance of kinematic structures with different degrees of freedom. For the case of the base reaction minimization problem that we discussed, we would compare a 2 d.o.f. non-redundant manipulator with a 3 d.o.f redundant manipulator both of which are shown in Fig. 6.1 (for the non-redundant manipulator, $l_1=l_2=0.5m$ and $m_1=m_2=1.0kg$).

Assume that both these manipulators must track a prescribed straight-line trajectory. The open-loop base reaction profiles are shown in Figs. 6.2 (a,b). From the non-redundant manipulator base reaction profile (Fig. 6.2), we can see that the peak force is $0.52N$ and the peak moment is at $0.13\ N\text{-}m$. Also, from the redundant manipulator base

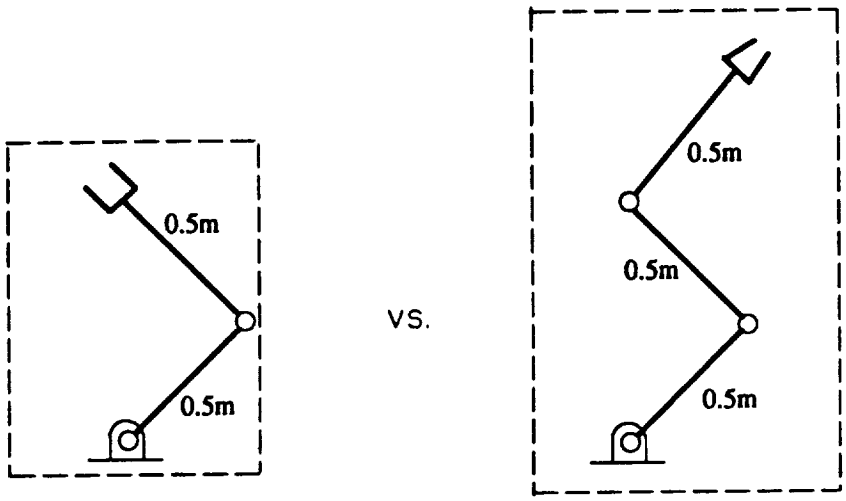
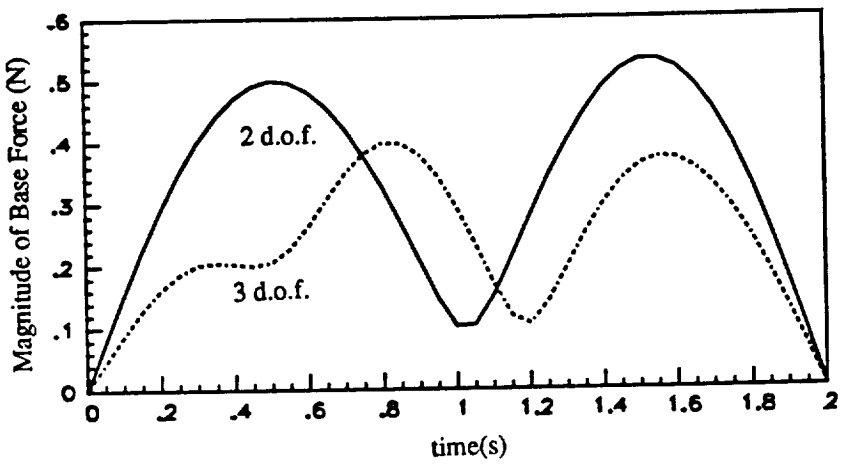


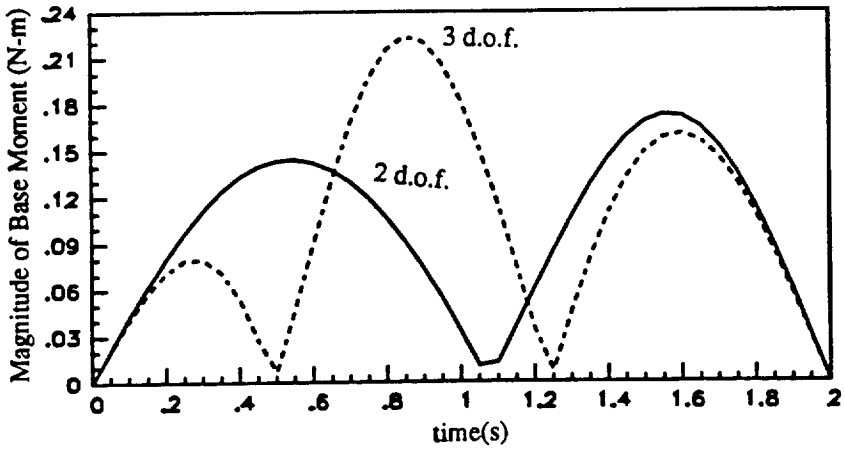
Fig 6.1 Comparison of Incompatible Non-redundant and Redundant Manipulators

reaction profile, we can observe that the peak base force is about *0.4 N* and the peak base moment is around *0.23 N-m*. From these profiles, one can see that the peak magnitude of the base moment of the redundant manipulator is *higher* than that of the non-redundant manipulator while the peak magnitude of the base force of the redundant manipulator is *lower* than that of the non-redundant manipulator. From these results, it is tempting to conclude that the manipulator with kinematic redundancy does not reduce the base moment. Furthermore, in view of the trade-offs associated with using redundant manipulators, one would probably conclude that the non-redundant manipulator should be used instead of the redundant manipulator.

The problem with the above comparison lies in the incompatibility of the kinematic structures that we chose to compare. For example, the redundant manipulator shown in Fig. 6.1 is much larger and heavier than its non-redundant counterpart. It is obvious that one can always choose a smaller and lighter non-redundant manipulator for comparison with a redundant manipulator and conclude wrongly that the use of the kinematic redundancy is not effective in reducing the base reactions. In order to draw accurate conclusions it is necessary to have a procedure for evaluating the effectiveness of using



(a)



(b)

Fig. 6.2 Comparison of Magnitudes of Base Reactions

kinematic redundancy to enhance the dynamic performance of a manipulator.

6.2.2 Compatibility Criteria for Performance Comparison

The two manipulators (kinematic structures) that we choose to compare can be denoted by K_i and K_j where the subscripts i and j refer to the number of degrees of freedom of the two manipulators. In general, i and j are either greater than or equal to the dimension of the task space, n.

We will first establish the compatibility criteria. There are three requirements that we are concerned with: (1) task compatibility; (2) geometric compatibility; (3) mass compatiblity. The task compatibility requirement demands that the two manipulators perform the same class of end-effector tasks. The geometrical compatibility guarantees that the sizes of the kinematic structure and the class of tasks that can be performed by these two manipulators are the same. The mass compatibility requirement simply ensures that the weights of the two manipulators are equal.

By *task compatibility*, we mean that the Task Specifications of the end-effector trajectory for the two manipulators are the same. For example, if the end-effector trajectory is of type II (x specified in space but not in time.), then the following equation has to be true:

$$[x(\alpha)]_{K_i} = [x(\alpha)]_{K_j}, \quad 0<\alpha<\alpha_{max}. \tag{6.1}$$

where α is the arc length of the curve that the end-effector must track. In the above equation, the function describing arc length for manipulator K_i, $(\alpha(t))_{K_i}$, is not necessarily the same as its counterpart for the manipulator K_j, $(\alpha(t))_{K_j}$. The above criterion enables us to describe a common task for both manipulators to perform.

To ensure *geometrical compatibility*, the basic requirement is that the workspaces of the two manipulators must be the same. If we let W_i denote the workspace of K_i, then we can define the elements of W_i as a set of points that are inside or on the boundary of W_i. Mathematically, W_i can be expressed as

$$W_i = \{x=(x_1,...x_n)^T \mid x=f(q), \ for \ \ q_{min}<q<q_{max}\}, \tag{6.2}$$

where $f(q)$ is the forward kinematic equations of K_i and q_{min} and q_{max} are vectors denoting joint limits of K_i. The ideal geometrical compatibility condition for the two manipulators is as follows:

If W_i equals W_j, then K_i is geometrically compatible with K_j.

With this compatibility condition, one can determine quite easily if two manipulators are geometrically compatible with each other. For example, as shown in Fig. 6.3, K_2 is obviously geometrically compatible to K_3 as W_2 (the set of points inside and on a circle of radius 1) is equal to W_3. From this example, one might be tempted to state that the condition of compatibility implies that the sum of the link lengths of K_2 must be equal to the sum of the link lengths of K_3. However, as shown in Fig. 6.4, we see that the latter condition is obviously not true. The manipulator shown in Fig. 6.4(a) is not geometrically compatible to the one shown in Fig. 6.4(b) even though the sum of the link lengths for both manipulators are the same. (The workspace W_3 has a "hole" in the center which is absent in W_2.) In general, for planar manipulators, it is possible to obtain two geometrically compatible manipulators by inspection. However, for spatial mechanisms, one may have to resort to computer-aided software to find two compatible kinematic structures. If one cannot find two perfectly compatible kinematic structures, then the next best would be for them to be almost compatible. In many problems, the two manipulators shown in Fig. 6.4. may be considered to be sufficiently compatible for all practical purposes if the radius of the "hole" in the center of W_3 is small compared to the outer radius of W_3.

The third compatiblity condition, mass compatibility, can be obtained by imposing constraints on the total masses of the two manipulators. This condition can be simply stated as

$$(\sum_{i=1}^{j} m_i)_{K_j} = (\sum_{i=1}^{l} m_i)_{K_l}. \tag{6.3}$$

In applications in space, this mass equality constraint is crucial as there is always a constraint on the total permissible weight of a payload.

Using the above compatibility conditions we can develop a general procedure for

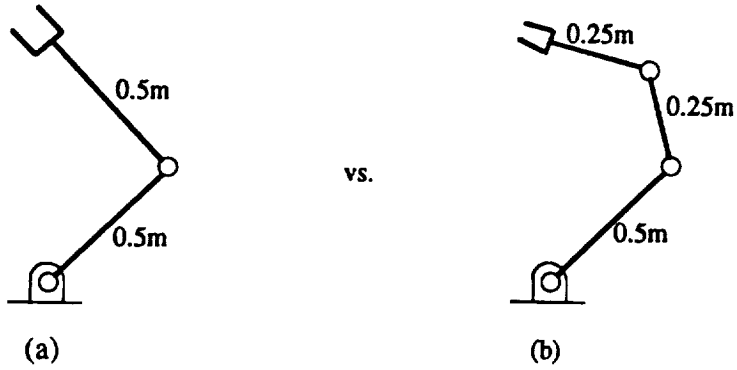


Fig. 6.3 Compatible Manipulators

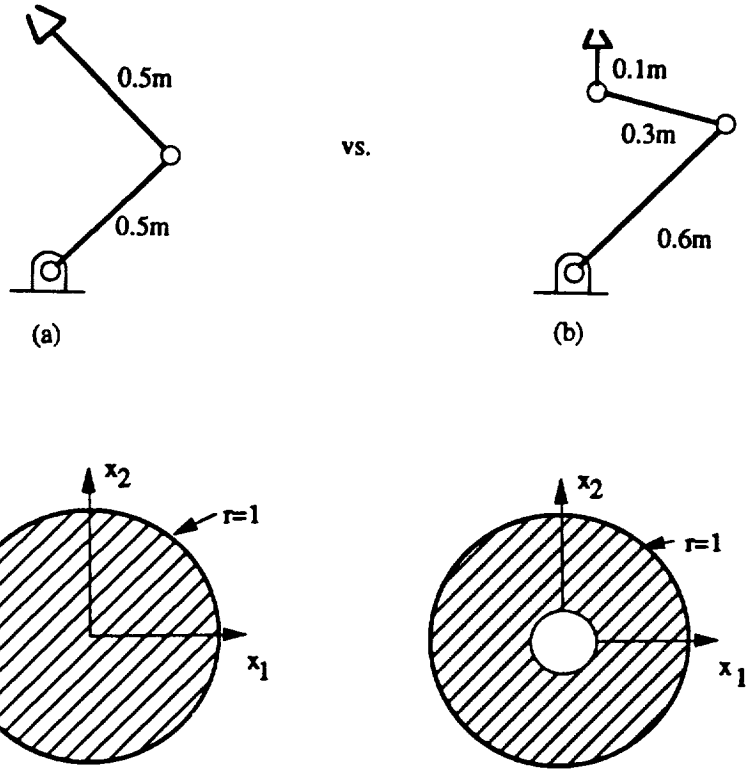


Fig. 6.4 Incompatible Manipulators

evaluating the use of kinematic redundancy in enhancing the dynamic performance. This procedure is developed with the following scenario in mind:

> An engineer has a manipulator design in mind. He would like to determine whether the addition of a single degree of freedom to his current design will improve the dynamic performance of the existing manipulator.

The procedure can be stated as follows:

1. Based on the current design (K_j), compute the workspace W_j and the total mass $(m_T)_j$ of the manipulator K_j.
2. Based on the workspace of K_j, obtain a set of link lengths of the manipulator K_{j+1} that would satisfy $W_j = W_{j+1}$.
3. From the mass compatibility conditions (Eq. (6.3)) obtain a set of masses $(m_i, i=1,..., j+1)$ for manipulator K_{j+1}.
4. Define the type of tasks for K_j and K_{j+1} based on the task compatibility criterion.
5. Compute the optimal joint trajectories for K_j and K_{j+1} to optimize some desired performance criterion (or criteria). Then, obtain the values of the performance indexes for K_j and K_{j+1}, respectively. Comparison of the performance indexes for K_j with those for K_{j+1} could then be used as a fair basis for evaluating the effectiveness of adding a degree of redundancy to K_j.

6.2.3 Illustrative Example

In the base reaction minimization problem studied in Chapter 5, we obtained the optimal joint trajectory for a redundant manipulator. In that study, we used the freedom in the joint motion to minimize the base reactions of a redundant manipulator. We did not really question the effectiveness of kinematic redundancy in minimizing the base reactions. In the following example, we will illustrate the use of the above procedure to systematically examine the issue of whether kinematic redundancy improves or degrades dynamic performance.

To examine this issue, we will first assume that we have an exisiting non-redundant manipulator to accomplish the prescribed task as shown in Fig. 6.6(a). The non-redundant manipulator K_2 is a 2 d.o.f manipulator with link length of $0.5m$ and mass of $0.5kg$ for each link. The center of mass for each link is assumed to be at the middle of the link. The moment of inertia for link i is assumed to be a function of the mass m_i and the

link length l_i of link i. We wish to examine the improvement one would get by having an additional degree of freedom. The following steps correspond to the procedure described in Section 6.2.2.

(1) Following the procedure outlined in Section 6.2.2, we will first compute the workspace, W_2. For a planar non-redundant manipulator with equal link lengths, the workspace is a complete circle with radius of *1.0m*. The total mass of the manipulator is *2.0 kg*.

(2) Next, from the workspace constraint, $W_2=W_3$, the total length of K_2 must be equal to *1.0m*:

$$\sum_{i=1}^{3} l_i = 1.0m. \tag{6.4}$$

For a planar 3 d.o.f. manipulator with a voidless workspace, the following conditions must be true :

$$l_1 \leq l_2+l_3. \tag{6.5}$$

To satisfy the above two equations, one can select a specific 3 d.o.f. redundant kinematic structure from an infinite number of possible kinematic structures. To simplify matter, we let $l_2=l_3=0.25m$; l_1 is then equal to *0.5m*. Now, we have computed the geometrical parameters for the 3 d.o.f. manipulator, K_3 which is compatible with K_2.

(3) We know the total mass is equal to 2.0kg. Assuming that the cross-sectional area and material of each link is the same and neglecting the actuator mass, the mass of each link is proportional to its length, $m_1=1.0kg$, $m_2=m_3=0.5kg$.

(4) The task of the end-effector of K_2 and K_3 is the same as the straight line task of the example in Chapter 5.

(5) Using the open-loop analysis, we can obtain the optimal base reactions profile of the 3 d.o.f. redundant manipulator (K_3) with the link lengths and link masses computed in Steps (2) and (3).

Repeating Steps (1-5), we can also find a compatible 4 d.o.f. planar manipulator with

2 degrees of kinematic redundancy. In Fig. 6.6(b,c), we show the base reaction profiles of the 2 d.o.f., 3 d.o.f., and 4 d.o.f. planar manipulators (as depicted in Fig. 6.6 (a)) performing the same straight-line task from point *A* to *B*. The peak magnitude of the base force of the 3 d.o.f. manipulator is *0.3N*, about *40%* less than that of the 2 d.o.f. manipulator. The peak magnitude of the base moment of the 3 d.o.f. manipulator is *0.12N-m*, about 23% less than that of the 2 d.o.f. manipulator. For the 4 d.o.f. planar manipulator, we see drastic reduction in the magnitudes of the base reactions. The peak magnitude of the base force for the 4 d.o.f. redundant manipulator is only 20% of the peak magnitude of the non-redundant manipulator while the peak magnitude of the base moment is 36% of the peak magnitude of the non-redundant manipulator. One can also note that the base reactions of the 3 d.o.f. and 4 d.o.f. manipulators at the beginning and the end portions of the trajectory are of the same magnitude. From *0.3s* to *1.9s*, the base reactions of the 4 d.o.f. planar manipulator is significantly smaller than that of the 3 d.o.f. manipulator, but more oscillatory. From these results, one can conclude that by increasing the degree of kinematic redundancy, the base reactions can be reduced quite drastically.

6.3 Some Implementation Issues in Motion Planning

In the implementation and application of the unified motion planning, the following questions naturally arise:

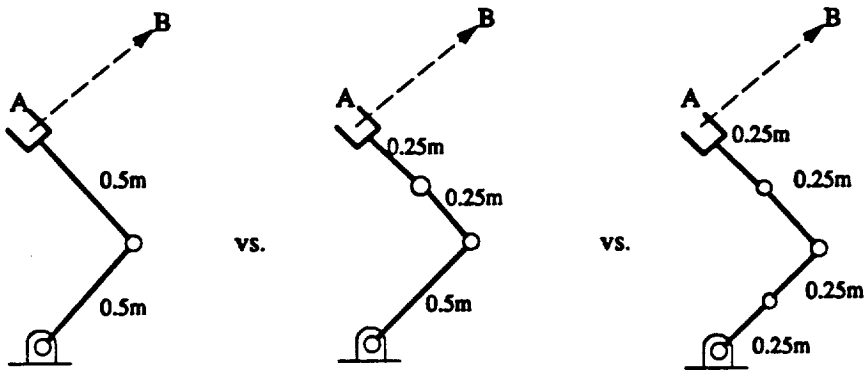
1. How many parameters should be used to represent the joint trajectories?
2. To what parameters is the performance index most sensitive?
3. What is the "landscape" of the optimization problem in the optimization parameter space?

These questions will be addressed in the following subsections.

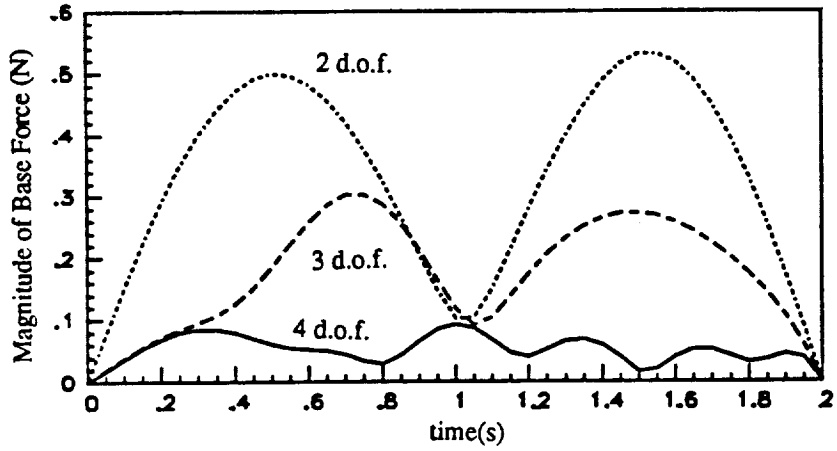
6.3.1 Appropriate Number of Parameters

In using the parameterization scheme developed in Chapters 4 and 5, one has to determine the number of parameters used in the parameterization of the independent variables. We will examine this important topic in the context of the base reaction minimization problem studied in Chapter 5.

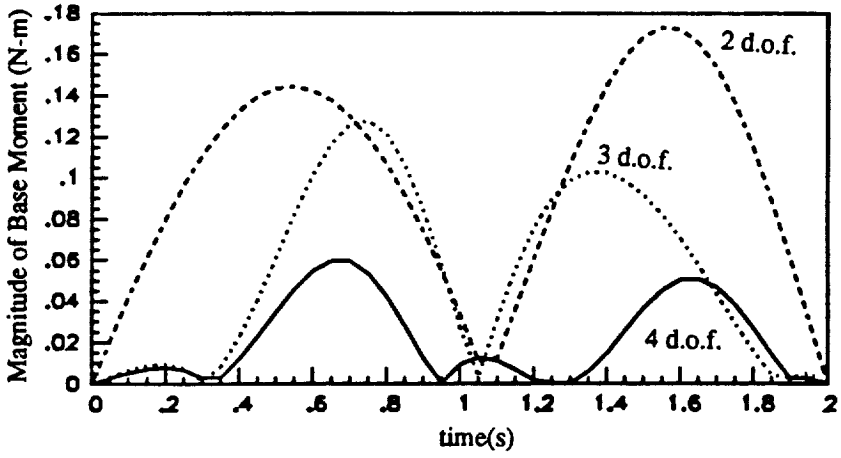
In the base reaction minimization problem, the optimal joint motion was obtained by



(a)



(b)



(c)

Fig. 6.5 Base Reactions of 2 d.o.f., 3 d.o.f., and 4 d.o.f Planar Manipulators

using three independent parameters to represent the joint trajectory. The logical question one can ask is the following: *What is the minimum number of parameters necessary to represent the optimal joint motion?* The answer to this problem is problem dependent. In order to gain some insight into the answer to this question, one must increase the number of parameters used in the representation scheme and determine whether the resulting solution is better.

To study this issue for the base reaction minimization problem, we compared the results of a three parameter representation scheme versus a six parameter scheme representation. The results of the two schemes shown in Figs. 6.6 (a,b) are almost identical which strongly indicates that three parameters are sufficient for obtaining an optimal solution (in this case).

6.3.2 Sensitivity

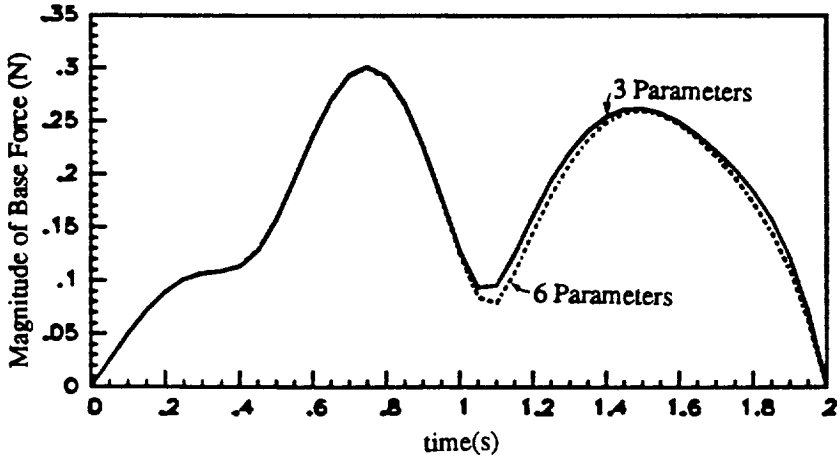
In this section, we examine the sensitivity of the base reactions with respect to the joint trajectory parameters (P_j) for the base reaction minimization problem (of the 3 d.o.f. planar manipulator) examined in Chapter 5.

The parameters of the joint trajectory are denoted by vector P_j which consists of the parameters used in the joint-space trajectory parameterization and the initial joint configuration parameter σ. In this example, we will use a polynomial of sixth order to represent the scalar s_1 of Eq. (5.24). The independent variables in the representation of s_1 are a_1, a_2, and a_3;also one is free to choose σ. Two possible candidates for σ are the orientation of the end-effector ($\sigma = q_1(0) + q_2(0) + q_3(0)$) and the joint variable of the first link ($\sigma = q_1(0)$). In our analysis, we choose the orientation of the end-effector, $\sigma = q_1(0) + q_2(0) + q_3(0)$, as an independent variable.

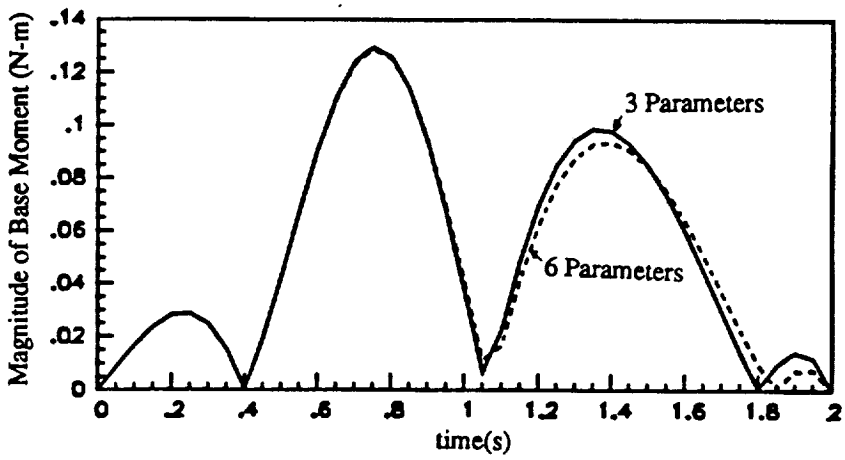
In vector form, we can express the parameter vector P_j as:

$$P_j=[a_1 \;\; a_2 \;\; a_3 \;\; \sigma]^T. \tag{6.6}$$

To determine the sensitivity of a performance index I with respect to the parameters, we perturb the optimization parameters and compute the sensitivity coefficients $\frac{\partial I}{\partial p_j^{(k)}}$ which can be numerically approximated by the following equation:



(a) Base Force



(b)Base Moment

Fig. 6.6 Comparison of 3 Parameter Representation Versus 6 Parameter Representation

C-2

$$\frac{\partial I}{\partial P_j^{(k)}} \approx \frac{I(P_j^{(1)},\ldots,P_j^{(k)}+\Delta P_j^k,..,P_j^{(l)}) - I(P_j^{(1)},\ldots,P_j^{(k)}..,P_j^{(l)})}{\Delta P_j^k}. \tag{6.7}$$

where $P_j^{(k)}$ denotes the k^{th} element of P_j and l represents the order of the polynomial used in representing s_1.

Using the above equation, we will examine the performance index I_2 that we used in the base reaction minimization problem of chapter 5. The sensitivity analysis is performed at the optimal solution P_j^* = [0.974 0.6632 0.3128 -1.0]. Using Eq. (6.7), we obtain the following sensitivities for I_2 with respect to the optimization parameters:

$$\frac{\partial I_2}{\partial a_1} = 0.02$$

$$\frac{\partial I_2}{\partial a_2} = 0.02$$

$$\frac{\partial I_2}{\partial a_3} = 0.01$$

$$\frac{\partial I_2}{\partial \sigma} = 0.39.$$

From the above sensitivities, we see that the initial joint configuration parameter σ is the most sensitive parameter. In general, it is useful to obtain sensitivities when solving an optimization problem since these sensitivities tell us about the behavior of the function in the vicinity of a solution and are also useful in identifying the critical parameters.

6.3.3 Landscape of the Optimization Problem

The purpose of this subsection is to give some insight into the "landscape" of the optimization problem. The term landscape refers to the graph of the performance index versus the most sensitive parameters in the optimization problem. We will use the base reaction minimization example to illustrate the process of determining the landscape of the problem.

To generate the landscape for the base reactions, we will investigate the following two cases:

1. *Minimizing peak base force using* $I_f = max\ |f_0|$.
2. *Minimizing peak base moment using* $I_m = max\ |n_0|$.

The profiles of the optimal peak base force and optimal peak base moment are plotted as functions of the orientation of the manipulator in Fig. 6.7. From this figure one can note that the minimization of the base force is somewhat correlated to the minimization of the base moment. The valleys of the locally optimal peak base force and the locally optimal peak base moment correspond to approximately the same orientation angles. This shows that when one minimizes peak base force, the peak base moment is also simultaneously minimized.

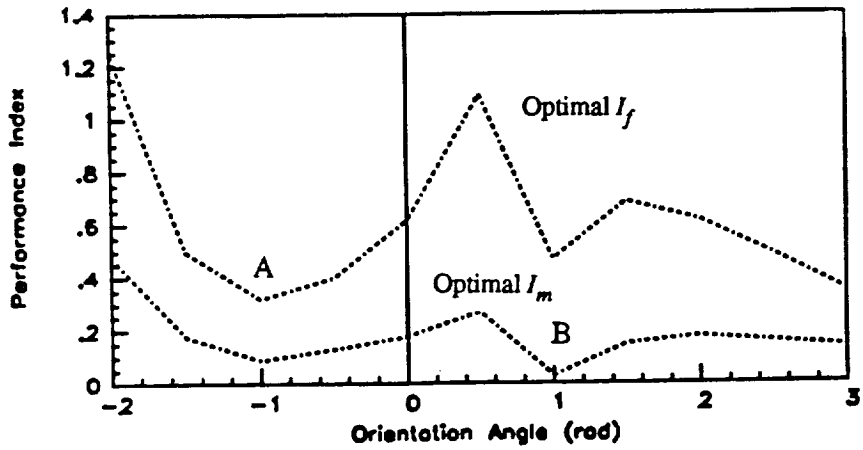


Fig. 6.7 Landscapes of Performance Indexes

Furthermore, from Fig. 6.7, we can investigate the trade-offs one has to consider when optimizing the base reactions. If one wants to minimize only the magnitude of the base moment we would want to obtain the solution corresponding to *B* in Fig. 6.7. For minimum value of peak base force, one would like to obtain the solution corresponding to point *A* in Fig. 6.7.

As mentioned earlier, one of the trade-offs in employing extra degrees of freedom is that the optimal trajectory may not be attainable. To illustrate this point, the optimal I_f is plotted as a function of the initial orientation angle σ of the end-effector and the initial

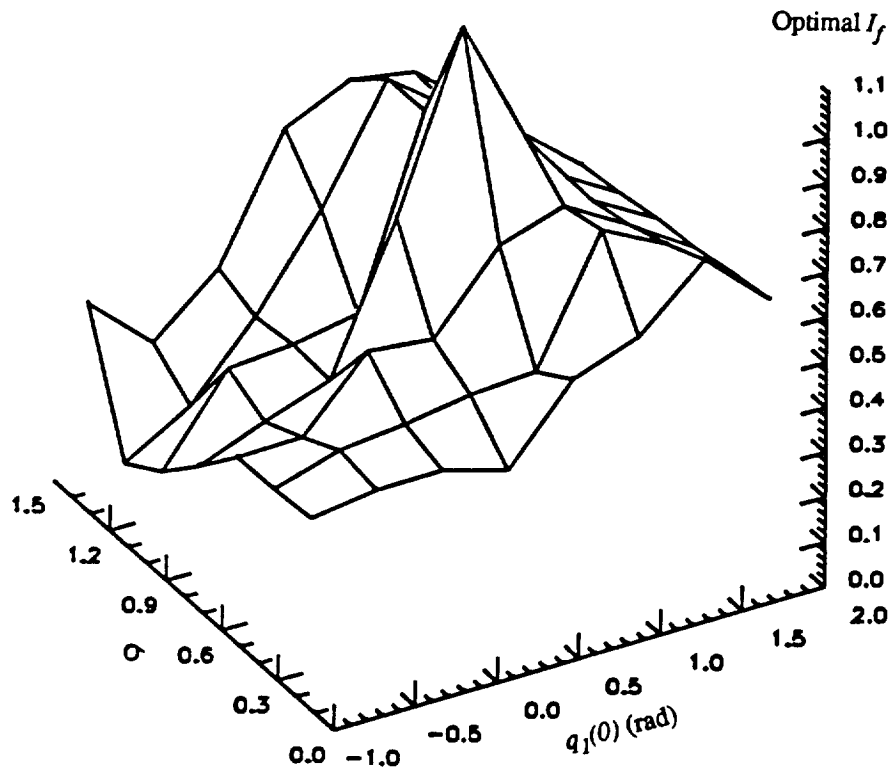


Fig. 6.8 Optimal I_f as a Function of Initial Joint Configuration Parameters

joint angle of link 1, $q_1(0)$ (see Fig. 6.8) for a 4 d.o.f. planar manipulator with two degrees of redundancy. The 3-D plot shows that there are numerous local minima. Therefore, more initial guesses should be used for obtaining the optimal trajectory for the globally minimum solution. We have examined the landscape of the optimization problem to gain some feel for the "goodness" of the optimal solution.

6.4 Summary

In this chapter, we have developed a procedure for evaluating the effectiveness of the use of kinematic redundancy in improving the dynamic performance of a manipulator. The framework enables one to choose two compatible manipulators for proper comparison. In the base reaction example, we have illustrated that without this framework, one may compare manipulators that are not compatible and draw incorrect conclusions regarding the effectiveness of the kinematic redundancy. We then showed that by increasing the degree of kinematic redundancy, the base reactions of a planar manipulator can be reduced quite drastically, but the trade-off is that the global optimal trajectory may be very difficult to obtain. In addition, we also examined the issues of how many parameters should be used in representing the independent variables in the optimization problem as well as the sensitivity of the performance index to these parameters. For the base reaction minimization problem of a 3 d.o.f. redundant manipulator, we found that the initial end-effector orientation σ is the most sensitive parameter.

Chapter 7

Conclusions and Future Work

7.1 Summary and Conclusions

We have developed a unified motion planning approach for robotic manipulators. As we have seen, this approach has the following features:

1. It simultaneously plans the trajectory for the manipulator and synthesizes a feedback control strategy which does not violate actuator constraints.
2. Both non-redundant and redundant manipulators are addressed in the same framework.
3. The approach is set in an optimization framework which allows the analyst to plan motions which optimize dynamic performance by exploiting any available freedom in the end-effector and/or joint trajectories.

In Chapter 4 we demonstrated the application of the unified approach to planning feedback-controlled minimum-time motions for a 2 d.o.f. non-redundant manipulator. We have also seen how the unified approach may be used as a tool for quantitatively studying the trade-off between tracking error and any other measure of dynamic performance. In Chapter 5 we showed how the unified approach could be used to plan motions which minimize the magnitude of the reactions transmitted to the base of the manipulator, a problem of considerable importance in Space Robotics.

We have thus clearly demonstrated the power of the unified motion planning approach in planning realizable motions for robotic manipulators.

One issue of great interest to a designer is the evaluation of the effectiveness of kinematic redundancy. With the systematic procedure described in Chapter 6 we are able

to compare different kinematic structures in a meaningful fashion. In the planar examples that we studied, we found that by increasing the degree of kinematic redundancy, we are able to reduce the base reactions quite drastically. We also pointed out one major disadvantage of increasing the number of excess degrees of freedom - the global optimal solution may be difficult to obtain due to the increase in the number of local minima.

There are some limitations on the proposed unified motion planning approach. The unified approach is optimization-based. Therefore, it also contains all the problems associated with the solution of non-convex non-linear programming problems. These problems include the choice of an appropriate optimization technique for the problem, local minima, the selection of the proper initial values for the parameters, and the choice of appropriate convergence criteria. Another limitation of this approach is that one has to choose a small enough integration time step for obtaining the correct simulation results; the choice of the integration time-step depends on the parameter vector $\boldsymbol{P}$. In this thesis, we use a variable Kutta-Merson algorithm of MATRIX_X^{TM} which chooses an appropriate integration time-step based on the local error tolerance criterion. However, using this algorithm can lead to very slow simulation and consequently, planning a trajectory using this approach for a complex manipulator can take up to 10 hours on a Micro-Vax running under the VMS4.7 Operating System.

7.2 Future Work

7.2.1 Orientation

The unified motion planning approach in its current form can only be used for manipulator tasks that do not impose any orientation requirement. To extend this approach for a general task, we can expand the Jacobian equation in the following way to map the joint velocity vector $\dot{q}$ to a generalized velocity $\dot{x}'$:

$$\dot{x}' = J'\dot{q}, \tag{7.1}$$

where $\dot{x}' = [\dot{x}\ \omega]^T$ and ω is the angular velocity of the end-effector in the base frame. With the above generalized velocity equation we can then address the orientation issue in a relatively straightforward fashion.

7.2.2 Kinematic Redundancy

The parameterization of the $\boldsymbol{k}$ vector is still not computationally efficient for manipulators with more than one degree of kinematic redundancy since we parameterize all the elements of $\boldsymbol{k}$. One possible improvement is to obtain the basis vectors that represent the nullspace of $\boldsymbol{J}$ and use Eq. (5.12) to reduce the number of parameterized variables from m to m-n. A simple procedure for obtaining a basis for the nullspace of a matrix is discussed in [41]. Using this procedure, one can then obtain the column vectors of the matrix Φ in Eq. (5.12). The coefficients s_i, corresponding to these column vectors, can then be parameterized using the approach developed in Section 5.3.

7.2.3 Kinematic Constraints

In our approach, kinematic constraints are not considered. Kinematic constraints such as joint limits and obtacle constraints present great challenges in motion planning problems. These constraints are very difficult to incorporate in the unified motion planning approach without increasing the complexity of the optimization problem. Most approaches proposed in the literature for handling kinematic constraints are not suitable for the unified motion planning framework. Therefore an important issue that needs to be addressed is, how the motion planning approach should be modified to include kinematic constraints.

With the incorporation of the above considerations, the unified motion planning approach would be an extremely powerful tool for planning motions for manipulators operating in complex environments.

Appendix A

Multi-Criterion Optimization Example

This multi-criterion optimization problem is taken from (Osyczka, 1984,pp. 31) [29].

$$\min \quad f_1(\mathbf{x}) = x_1^2 + x_2^2 + 12(x_1 + x_2)$$
$$f_2(\mathbf{x}) = -x_1 x_2$$

subject to

$$g_1(\mathbf{x}) = -0.5x_1^2 + 5x_1 - x_2 - 6 \geq 0$$
$$g_2(\mathbf{x}) = -x_1^2 + 6x_1 - x_2^2 + 14x_2 - 42 \geq 0$$
$$g_3(\mathbf{x}) = -x_1^2 + 16x_1 - x_2^2 + 6x_2 - 48 \geq 0$$

The feasible space of the optimization variables $\mathbf{x}_{\text{feasible}}$ is shown in Fig. A.1. The space of the performance indexes (f_1 and f_2) are shown in Fig. A.2. The optimal solution $\mathbf{x}^*$ is the straight-line $x_1 = x_2$. The Pareto optimal solutions are all the points of the lower boundary of the hatched area.

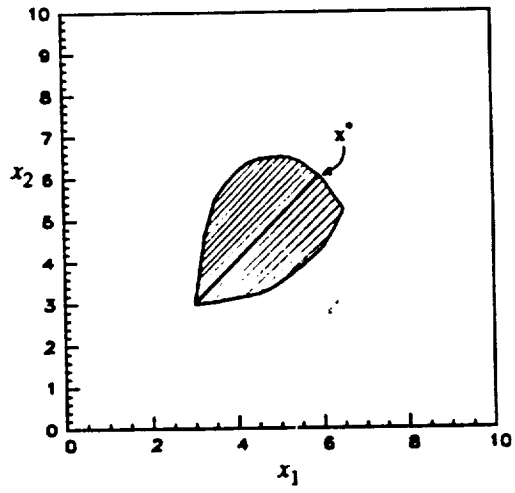


Fig. A.1: Feasible Space of **x**

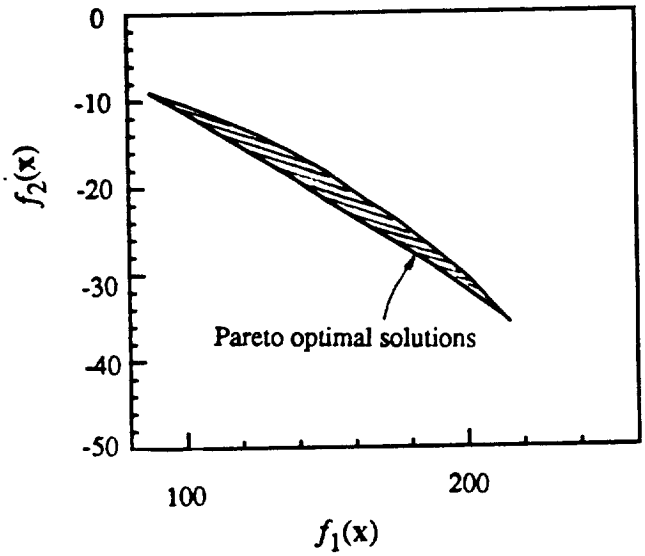


Fig. A.2: Space of Performance Indexes

Using the trade-off approach, the multi-criterion optimization problem is converted to a single criterion optimization problem.

$$\min \quad f_1(\mathbf{x}) = x_1^2 + x_2^2 + 12(x_1 + x_2)$$

subject to

$$g_1(\mathbf{x}) = -0.5x_1^2 + 5x_1 - x_2 - 6 \geq 0$$
$$g_2(\mathbf{x}) = -x_1^2 + 6x_1 - x_2^2 + 14x_2 - 42 \geq 0$$
$$g_3(\mathbf{x}) = -x_1^2 + 16x_1 - x_2^2 + 6x_2 - 48 \geq 0$$
$$f_2(\mathbf{x}) = -x_1 x_2 \leq \rho$$

The optimal solutions for different values of ρ are tabulated in Table A-1. The above problem was also solved using the weighting objective method which minimizes a weighted sum of f_1 and f_2 with weighting factors w_1 and w_2. For this example, the weighted objective method has difficulty obtaining the Pareto optimal solutions. The optimal solution which is obtained by the weighted objective method is always $\mathbf{x}^* = (3,3)$, regardless of what values of w_1 and w_2 were used.

ρ	x^*	$f_1(x)$	$f_2(x)$
0	(3,3)	90.0	-9
-10	(3.16,3.16)	95.89	-10
-20	(4.47,4.47)	147.335	-20
-30	(5.47,5.47)	191.45	-30

Table A-1 Results of the Trade-Off Approach

Appendix B

Base Reaction Equations for a 3 d.o.f. Planar Manipulator

In this section, we derive the dynamic equations for the base reactions transmitted by the manipulator while performing a task.

Considering a 3 d.o.f. planar manipulator, the magnitude of the base moment (n_o) is simply equal to the magnitude of the torque produced by the first joint.

It is easy to show that in microgravity condition, the force transmitted to the base is equal to the sum of the inertia forces for all the links. Let the mass of link i be m_i and the position of the center of mass of link i with respect to a reference frame in the base be $X_{c,i}$. For a manipulator of m degrees of freedom, the base force f_o is simply given by

$$f_o = \sum_{i=1}^{m} m_i \ddot{X}_{c,i} \qquad (B.1)$$

where $\ddot{X}_{c,i}$ is the acceleration of center of mass of link i.

Using the Jacobian relation, the velocity of the center of gravity of link i, $\dot{X}_{c,i}$, is given by

$$\dot{X}_{c,i} = J_{c,i}\dot{\mathbf{q}}. \qquad (B.2)$$

where $J_{c,i}$ is the Jacobian matrix that relates $\dot{X}_{c,i}$ and $\dot{q}$.

Taking the time-derivative of above equation, the acceleration of the center of mass of link i is simply given by

$$\ddot{X}_{c,i} = J_{c,i}\ddot{q} + \dot{J}_{c,i}\dot{q} \qquad (B.3)$$

Substituting Eq. (B.1) in Eq. (B.3), f_o can be expressed as

$$f_o = \sum_{i=1}^{m} m_i (J_{c,i}\ddot{q} + \dot{J}_{c,i}\dot{q}) \qquad (B.4)$$

Combining base force and base moment vectors, we can express F_b as

$$F_b=\begin{pmatrix} f_o \\ n_o \end{pmatrix}=A\ddot{q}+B(q,\dot{q}) \qquad (B.5)$$

where $A=\left(\sum_{1^{st}\ row\ of\ M} m_i J_{cj}\right)$, B is the collection of nonlinear terms in $\dot{q}$, and M is the mass matrix of a manipulator.

For a 3 d.o.f. planar manipulator, the base force vector has a x-direction component $(f_o)_x$ and a y-direction component $(f_o)_y$. The base moment vector has only a z-direction component, $(n_o)_z$. Hence, the base reactions vector is given by

$$F_b = [\,(f_o)_x\ (f_o)_y\ (n_o)_z]^T.$$

Let the elements of A be A_{ij}, (i=1,2,3) and (j=1,2,3). The expressions for A_{ij} are:

$$\begin{aligned}
A_{11}&=-l_{c3}m_3 s_{123}-l_2 m_3 s_{12}-l_{c2}m_2 s_{12}-l_1 m_2 s_1-l_{c1}m_1 s_1\\
A_{12}&=-l_{c3}m_3 s_{123}-l_2 m_3 s_{12}-l_{c2}m_2 s_{12}\\
A_{13}&=-l_{c3}m_3 s_{123}\\
A_{21}&=l_{c3}m_3 c_{123}+l_2 m_3 c_{12}+l_{c2}m_2 c_{12}+l_1 m_2 c_1+l_{c1}m_1 c_1\\
A_{22}&=l_{c3}m_3 c_{123}+l_2 m_3 c_{12}+l_{c2}m_2 c_{12}\\
A_{23}&=l_{c2}m_3 c_{123}\\
A_{31}&=l_1 l_{c3}m_3 c_{23}+2l_2 l_{c3}m_3 c_3+2l_1 l_2 m_3+2l_1 l_{c2}m_2 c_2+l_{c3}^2 m_3\\
&\quad +l_2^2 m_3+l_1^2 m_3+l_{c2}^2 m_2+l_1^2 m_2+l_{c1}^2 m_1+I_1+I_2+I_3\\
A_{32}&=l_1 l_{c3}m_3 c_{23}+2l_2 l_{c3}m_3 c_3+l_1 l_2 m_3 c_2+l_1 l_{c2}m_2 c_2+l_{c3}^2 m_3+l_2^2 m_3\\
&\quad +l_{c2}^2 m_2+I_2+I_3\\
A_{33}&=l_1 l_{c3}m_3 c_{23}+l_2 l_{23}m_3 c_{23}+l_{2c3}m_3 c_3+l_{c3}^2 m_3
\end{aligned} \qquad (B.6)$$

Let the elements of B be denoted by B_i and they are given by:

$$\begin{aligned}
B_1&=-[\dot{q}_3^2+2(\dot{q}_2+\dot{q}_1)\dot{q}_3+(\dot{q}_1+\dot{q}_2)^2]l_{c3}m_3 c_{123}-(l_2 m_3+\\
&\quad l_{c2}m_2)c_{12}(\dot{q}_1+\dot{q}_2)^2-(l_1 m_3+l_1 m_2+l_{c1}m_1)c_1\dot{q}_1^2.\\
B_2&=-[\dot{q}_3^2+(2\dot{q}_2+\dot{q}_1)\dot{q}_3+(\dot{q}_1+\dot{q}_2)^2]l_{c3}m_3 s_{123}-\\
&\quad (l_2 m_3+l_{c2}m_2)s_{12}(\dot{q}_1+\dot{q}_2)^2-[l_1 m_3+l_1 m_2+l_{c1}m_1)]s_1\dot{q}_1^2.
\end{aligned}$$

$$B_3=-[(\dot{q}_2+\dot{q}_3)^2+2\dot{q}_1\dot{q}_3+2\dot{q}_1\dot{q}_2]l_1l_{c3}m_3s_{23}-(2\dot{q}_2\dot{q}_3+2\dot{q}_1\dot{q}_3+\dot{q}_3{}^2)l_2l_{c3}m_3s_3-(\dot{q}^2+2\dot{q}_1\dot{q}_2)l_1l_2m_3s_3-(\dot{q}_2+2\dot{q}_1\dot{q}_2)l_1l_{c2}m_2s_2. \quad (B.7)$$

Appendix C

Joint Acceleration Equation for a 4 d.o.f. Planar Manipulator

The joint acceleration equation can be obtained by differentiating Eq. (5.9) with respect to time.

$$\ddot{q} = J^{+}\ddot{q} + \dot{J}^{+}\dot{x} + \dot{W}k + W\dot{k} \qquad (C.1)$$

where $W = I - J^{+}J$.

The derivative of $\dot{W}$ can be obtained as

$$\dot{W} = -\dot{J}^{+}J - J^{+}\dot{J} \qquad (C.2)$$

The Jacobian J is given by the following equations:

$$\begin{aligned}
J_{11} &= -l_1 s_1 - l_2 s_{12} - l_3 s_{123} - l_4 s_{1234} \\
J_{21} &= l_1 c_1 + l_2 c_{12} + l_3 c_{123} + l_4 c_{1234} \\
J_{12} &= -l_2 s_{12} - l_3 s_{123} - l_4 s_{1234} \\
J_{22} &= l_2 c_{12} + l_3 c_{123} + l_4 c_{1234} \\
J_{13} &= -l_3 s_{123} - l_4 s_{1234} \\
J_{23} &= l_3 c_{123} + l_4 c_{1234} \\
J_{14} &= -l_4 s_{1234} \\
J_{24} &= l_4 c_{1234}
\end{aligned} \qquad (C.3)$$

The derivative of the pseudo-inverse $\dot{J}^{+}$ is given by

$$\dot{J}^{+} = \dot{J}^{T}Q + J^{T}\dot{Q} \qquad (C.4)$$

where $Q = (J J^{T})^{-1}$.

The derivative of Q is given by the following equation:

$$\dot{Q}=\frac{1}{ad-bc}\begin{pmatrix}\dot{d} & -\dot{b}\\ -\dot{c} & \dot{a}\end{pmatrix}-\frac{\dot{a}d+a\dot{d}-\dot{b}c-b\dot{c}}{(ad-bc)^2}\begin{pmatrix}d & -b\\ -c & a\end{pmatrix}$$

where

$$\begin{aligned}a&=J_{11}^2+J_{12}^2+J_{13}^2+J_{14}^2\\ b&=J_{11}J_{12}+J_{12}J_{22}+J_{13}J_{23}+J_{14}J_{24}\\ c&=J_{21}J_{11}+J_{22}J_{12}+J_{23}J_{13}+J_{24}J_{14}\\ d&=J_{21}^2+J_{22}^2+J_{23}^2+J_{24}^2\end{aligned}\qquad (C.5)$$

The time derivatives of a, b, c, d can be expressed as follows:

$$\begin{aligned}\dot{a}&=2(J_{11}\dot{J}_{11}+J_{12}\dot{J}_{12}+J_{13}\dot{J}_{13}+J_{14}\dot{J}_{14})\\ \dot{b}&=\dot{J}_{11}J_{21}+J_{11}\dot{J}_{21}+J_{12}\dot{J}_{22}+\dot{J}_{12}J_{22}+\dot{J}_{13}J_{23}+J_{13}\dot{J}_{23}+J_{13}\dot{J}_{23}+\dot{J}_{14}J_{24}\\ &\quad +J_{14}\dot{J}_{24}\\ \dot{c}&=\dot{J}_{21}J_{11}+J_{21}\dot{J}_{11}+J_{22}\dot{J}_{12}+\dot{J}_{22}J_{12}+\dot{J}_{23}J_{13}+J_{23}\dot{J}_{13}+J_{23}\dot{J}_{13}+\dot{J}_{24}J_{14}+\\ &\quad J_{24}\dot{J}_{14}\\ \dot{d}&=2(J_{21}\dot{J}_{21}+J_{22}\dot{J}_{22}+J_{23}\dot{J}_{23}+J_{24}\dot{J}_{24})\end{aligned}\qquad (C.6)$$

The derivatives of the elements of the Jacobian matrix are:

$$\begin{aligned}\dot{J}_{11}&=-l_1c_1\dot{q}_1-l_2c_{12}\dot{q}_{12}-l_3c_{123}\dot{q}_{123}-l_4c_{1234}\dot{q}_{1234}\\ \dot{J}_{21}&=-l_1s_1\dot{q}_1-l_2s_{12}\dot{q}_{12}-l_3s_{123}\dot{q}_{123}-l_4s_{1234}\dot{q}_{1234}\\ \dot{J}_{12}&=-l_2c_{12}\dot{q}_{12}-l_3c_{123}\dot{q}_{123}-l_4c_{1234}\dot{q}_{1234}\\ \dot{J}_{22}&=-l_2s_{12}\dot{q}_{12}-l_3s_{123}\dot{q}_{123}-l_4s_{1234}\dot{q}_{1234}\\ \dot{J}_{13}&=-l_3c_{123}\dot{q}_{123}-l_4c_{1234}\dot{q}_{1234}\\ \dot{J}_{23}&=-l_3s_{123}\dot{q}_{123}-l_4s_{1234}\dot{q}_{1234}\\ \dot{J}_{14}&=-l_4c_{1234}\dot{q}_{1234}\\ \dot{J}_{24}&=-l_4s_{1234}\dot{q}_{1234}\end{aligned}\qquad (C.7)$$

Appendix D

Inverse Kinematic Equations for a 3 d.o.f and a 4 d.o.f. Planar Manipulator

(1) 3 d.o.f. planar manipulators:

In our inverse kinematic equations, $\sigma=q_1+q_2+q_3$, is chosen to be the independent variable. Using the following equations, the joint variables q_1, q_2, q_3 can be obtained.

$$c_2=\frac{(x_1-l_3 c\sigma)^2+(x_2-l_3 s\sigma)^2-l_1^2-l_2^2}{2l_1 l_2} \qquad (D.1)$$

$$s_2=\pm\, sqrt(1-c_2^2) \qquad (D.2)$$

$$q_2=\text{ATAN2}\,(s_2,c_2) \qquad (D.3)$$

$$k_1=l_1+l_2 c_2 \qquad (D.4)$$

$$k_2=l_2 s_2 \qquad (D.5)$$

$$k_3=x_1-l_3 c\sigma \qquad (D.6)$$

$$k_4=x_2-l_3 s\sigma \qquad (D.7)$$

$$q_1=\text{ATAN2}\,(k_4,k_3)-\text{ATAN2}\,(k_2,k_1) \qquad (D.8)$$

$$q_3=\sigma-q_1-q_2 \qquad (D.9)$$

(2) 4 d.o.f. planar manipulators:

In our inverse kinematic equations, $\sigma=q_1+q_2+q_3+q_4$, and q_1 are chosen as the independent variables. Using the following equations, the joint variables q_2, q_3, q_4 can be obtained.

$$\begin{aligned} x_1' &= x_1-l_1 c_1 \\ x_2' &= x_2-l_1 s_1 \end{aligned} \qquad (D.10)$$

$$c_3 = \frac{(x_1' - l_4 c\sigma)^2 + (x_2' - l_4 s\sigma)^2 - l_2^2 - l_3^2}{2 l_2 l_3} \qquad (D.11)$$

$$s_3 = \pm\, sqrt(1 - c_3^2) \qquad (D.12)$$

$$q_3 = \text{ATAN2}\,(s_3, c_3) \qquad (D.13)$$

$$k_1 = l_2 + l_3 c_3 \qquad (D.14)$$

$$k_2 = l_3 s_3 \qquad (D.15)$$

$$k_3 = x_1' - l_4 c\sigma \qquad (D.16)$$

$$k_4 = x_2' - l_4 s\sigma \qquad (D.17)$$

$$q_2' = \text{ATAN2}\,(k_4, k_3) - \text{ATAN2}\,(k_2, k_1) \qquad (D.18)$$

$$q_2 = q_2' - q_1$$
$$q_4 = \sigma - q_1 - q_2 - q_3 \qquad (D.19)$$

References

1. Asada, H., and Slotine, J.J. E., *Robot Analysis and Control*, John Wiley and Sons, New York, 1986.
2. Bazaraa, M. S., *Nonlinear Programming*, John Wiley & Sons, 1979.
3. Barraquand, J., Langlois, B. and Latombe, J.-C., "Robot Motion Planning with Many Degrees of Freedom and Dynamic Constraints," *5th Int. Sym. of Robotics Research*, August 1989, Tokyo, Japan.
4. Beltzer, A. I., *Variational and Finite Element Methods*, Springer-Verlag, New York, 1990.
5. Beyers, R., and Subhas, D., "An Optimization-Based Approach for Simultaneous Plant-Controller Redesign," *J. Mechanisms, Automation, and Transmission in Design.*
6. Bobrow, J. E., Dubowsky, S., Gibson, J. S., "Time-Optimal Control of Robotic Manipulators Along Specified Paths," *Int. J. of Robotics Research*, pp. 3-17, vol. 4, No. 3, Fall, 1985.
7. Chang, P. H., "A Closed-form Solution for Control of Manipulators with Kinematic Redundancy," *IEEE Int. Conf. on Robotics and Automation*, pp.9-14, San Francisco, 1986.
8. Chung, C. L., Desa, S., de Silva, C. W., "Base Reaction Optimization of Redundant Manipulators for Space Applications," Technical Report of the Robotics Institute, Carnegie Mellon University, 1988.
9. Chung, C. L., Desa, S., "A Global Approach for using Kinematic Redundancy to Minimize Base Reactions of Manipulators," ASME Design Automation Conference, Montreal, Canada, September 1989.
10. Craig, J. J., *Introduction to Robotics*, Addison-Wesley, 1986.
11. de Silva, C. W., Chung, C. L., and Lawrence, C., "Base Reaction Optimization of Robotics Manipulators for Space Applications," *Proc. Int. Symp. Robots*, Sydney, Australia, November 1988.
12. Desa, S., and Roth, B, " Synthesis of Control Systems for Manipulators Using Multivariable Robust Servomechanism Theory," *Int. J. of Robotic*, pp. 18-34, Vol. 4, No. 3, Fall 1985.
13. Faux, I. D., and Pratt, M. J., *Computational Geometry for Design and Manufacture*, Ellis Hortwood Ltd., 1979.

14. Fenton, R. G., Benhabib, B., and Goldenberg, A., "Optimal Point-to-point Motion of Robots with Redundant Degrees of Freedom," *ASME Journal of Engineering for Industry*, vol. 108, May 1986, pp.120-126.

15. Fu, K. S., Gonzalez, R. C., and C. G. S. Lee, *Robotics:Control, Vision, and Intelligence*, McGraw-Hill, 1987.

16. Ghosal, A., "A New Approach for Kinematic Resolution of Redundancy," *Int. Journal of Robotics Research*, vol. 7, No. 2, pp. 22-35,1988.

17. Hirose, S., Ma, S., "Redundancy Decomposition Control for Multi-Joint Manipulator," 1989, *IEEE Int. Conf. on Robotics and Automation.*

18. Hollerbach, J. M., Suh, K. C., "Redundancy Resolution of Manipulators Through Torque Optimization," 1985, *IEEE Int. Conf. on Robotics and Automation*, St. Louis, Missouri.

19. Kahn, M.e E., and Roth, B., "The Near-Minimum-time Control of Open-loop Articulated Kinematic Chains," 1971, *J. Dyn. Sys., Meas., Contr.* 93:164-172.

20. Klein, C. A., and Huang, C. H., " Review of Psuedo-inverse Control for Use With Kinematically Redundant Manipulators," *IEEE Trans. Sys. Man, Cyber.*, SMC-13 (2), pp. 245-250.

21. Lerner, E. J., "Robots on the Space Station," *Aerospace America*, June 1987, pp. 42-45.

22. Lewis, F. L., *Optimal Control*, John Wiley & Sons, 1986.

23. Liegeois, A., "Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms," 1977, *IEEE trans. System, Man, Cybernetics*, SMC-7, pp. 868-871.

24. Lin, C-S, Chang, P-R, and J. Y. S. Luh, "Formulation and Optimization of Cubic Polynomial Joint Trajectories for Industrial Robots," *IEEE Trans. on Automatic Control*, Vol. AC-28, no. 12, Dec, 1983.

25. Longman, R. W., Lindberg, R. E., and Zedd, M. F., "Satellite-Mounted Robot Manipulators - New Kinematics and Reaction Moment Compensation", *Int. J. of Robotics Research*, vol. 6, no. 3, Fall 1987.

26. Maciejewski, A. A. and Klein, C. A., "Obstacle Avoidance for Kinematically Redundant Manipulators in Dynamically Varying Environements," *International Journal of Robotics Research*, pp. 109-117, vol. 4, No. 3, Fall 1985.

27. Michelena, N. F., and Agogino, A. M., "Multiobjective Hydraulic Cylinder Design," J. of Mechanisms, Transmission, and Automation in Design, vol. 110, 1988.

28. Nakamura, Y., Hanafusa, H., "Optimal Redundancy Control of Robot Manipulators," *The Int. Journal of Robotics Research*, vol. 6 no. 1, spring 87, pp.32-42.

29. Osyczka, Andrezj, *Multicriterion Optimization in Engineering*, Ellis Horwood Limited, 1984.

30. Papalambros, P. Y., and Wilde, D. J., *Principles of Optimal Design*, Cambridge University Press, 1988.

31. Paul, R. P., "Manipulator Cartesian Path Control," *IEEE Trans. Systems, Man, Cybern.*, vol. SMC-9, no. 11, pp 702-711.

32. Quinn, R. D., Chen, J. L., and Lawrence, C., "Redundant Manipulators for Momentum Compensation in a Microgravity Environment," AIAA, 1988.

33. Rao, C. R., and Mitra, S. K. *Generalized inverse of matrices and its applications*, New York, Wiley, 1971.

34. Rogers, D. F., *Mathematical Elements for Computer Graphics*, McGraw-Hill, 1976.

35. Sahar, G., and Hollerbach, J. M., "Planning of Minimum-time Trajectories for Robot Arms," *Int. J. of Robotics Research*, vol. 5, No. 3, Fall 1986.

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

37. Sciavicco, L. and Siciliano, B., "A Dynamic Solution to the Inverse Kinematic Problem for Redundant Manipulators," *IEEE International Conference on Robotics and Automation*, pp. 1081-1087, Raleigh, N.C., 1987.

38. Schmitt, D., Soni, A. H., Srinivasan, V., and Naganathan, G., "Optimal Motion Programming of Robot Manipulators," *ASME Journal of Mechanisms, Transmissions, and Automation in Design*, vol 107, June 1985, pp. 239-244.

39. Shin, K. G., and McKay N. D., "Minimum-time Control of Robotic Manipulators with Geometric Path Constraints," *IEEE trans. on Automation and Control*, vol. AC-30, No. 6, JUne 1985.

40. Suh, K. C., Hollerbach, J. M., "Local Versus Global Optimization of Redundant Manipulators," 1987, *IEEE Int. Conf. on Robotics and Automation.*

41. Strang, G., *Linear Algebra and Its Applications*, Academic Press, 1980.

42. Thakor, N. V., and McNeela, M. A., " Application of a Dynamic Programming Algorithm for Trajectory Planning of Finger-Like Manipulators," *J. of Robotic Systems*, 4(3),341-354, 1987.

43. Tolstov, G. P., *Fourier Series*, Dover Publications, New York, 1962.

44. Yoshikawa, T., "Analysis and Control of Robot Manipulators with Redundancy," *Robotics Research: The First Symposium*, ed. M. Brady and R. Paul, Cambridge, Mass., MIT Press, pp. 735-747, 1984.