CARNEGIE MELLON UNIVERSITY

DYNAMICS AND CONTROL OF' UNDERACTUATED MANIPULATORS

A DISSERTATION
SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
for the degree of
DOCTOR OF PHILOSOPHY
in
ELECTRICAL AND COMPUTER ENGINEERING

by

MARCEL BERGERMAN

Pittsburgh, Pennsylvania
December, 1996

To Clara and Rafael, for all the love.

To my parents, for my education.

To Yangsheng Xu, for the guidance 'and friendship.

Acknowledgments

I wish to express my sincere gratitude to Yangsheng Xu, for his technical as well as personal support; he is to me a role model, both as a researcher and as a human being.

I wish to thank also the members of my committee: Yasuyoshi Yokokohji, for bringing new ideas into this project; Takeo Kanade, for his support and constructive input; and Bruce Krogh and Howie Choset, for their support and positive feedback.

I wish to thank Christopher Lee for the brainstorming sessions we held together. Chris Lee also wrote most of the control software for this work, and served as my programming consultant whenever I ran into compilation problems; and H. Ben Brown, for designing and building the underactuated manipulator with which I validated the theory developed in this dissertation.

I further wish to thank Paulo Sérgio Pereira da Silva and Chris Paredis, for their contributions to this dissertation in the form of comments and feedback; Elaine Lawrence and Lynn Philibin, for their help in dealing with all the course paperwork, and for their friendship, in particular, their acting as my substitute moms; Kate Fissell, James Moody, and Bill Ross, for their excellent support with all matters related to computer usage; Debbie Scappatura and Ava Cruse, for providing courteous and professional secretarial support during my stay at CMU; Marie Elm, for kindly proofreading this dissertation and correcting all the little grammatical mistakes I made; and Fabio Gagliardi Cozman, for helping me to generate a video tape of the experimental results obtained in this dissertation.

Finally, I wish to acknowledge the financial support of the Brazilian National Council for Research and Development, CNPq. I hope I can give back to my country as generously as they paid for my studies.

Abstract

The class of underactuated systems is composed of a variety of mechanical as well as biological systems. The term underactuation refers to the fact that not all joints or degrees-of-freedom (DOFs) of the system are equipped with actuators, or are directly controllable. The DOFs of an underactuated system are dynamically coupled, and their dynamic equations are nonlinear and are constrained by nonholonomic differential equations. In this dissertation, it is our goal to study these fundamental characteristics toward the aim of developing modeling and control methods for underactuated systems, with focus on the class of serial chain underactuated manipulators with joints connected by rigid links.

Although an underactuated manipulator can be structurally identical to a fully actuated one, their dexterities differ because of the presence of unactuated joints. We quantify the dexterity of underactuated manipulators, and propose an optimization index to determine the positions of the unactuated joints which maximize the dexterity. Also differently from a conventional manipulator, the unactuated joints of an underactuated manipulator can be driven only indirectly through their coupling with the actuated ones. We present the dynamic characteristics of underactuated manipulators, and propose the coupling index concept to measure the dynamic coupling between the joints. The coupling index is utilized for the analysis, design, and optimal control of underactuated manipulators.

Parameter uncertainty and external disturbances interfere with the control of manipulators. We propose a robust control method to control all joints of an underactuated manipulator to an equilibrium point. We investigate the controllability of the system, and demonstrate that controllability and the coupling index are related concepts. We also propose the optimal control sequence of the unactuated joints to account for underactuated manipulators with more unactuated than actuated joints. In practice, manipulators operate in workspaces populated with obstacles. We propose a high-level motion planning method that generates collision-free trajectories to be followed by the manipulator. The methods proposed are implemented on a planar manipulator designed and built at the Space Robotics Laboratory at Carnegie Mellon University.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The class of underactuated systems is composed of a variety of mechanical as well as biological systems. The term *underactuation* refers to the fact that not all joints or degrees-of-freedom (DOFs) of the system are equipped with actuators, or are directly controllable. A classical example of an underactuated system is the movie actor Bruce Lee, who is able to project the tip of a 3-link nunchuck at any point in space. When one considers his arm as a set of links connected by controllable joints, and the nunchuck as another set of links connected by unactuated joints, the system formed by his arm and the nunchuck is underactuated. The presence of unactuated DOFs makes underactuated systems fundamentally different from conventional, fully actuated ones. It is our goal in this dissertation to model such kinds of systems and to propose control and planning methods that allow them to be utilized in practice.

A system becomes underactuated in three possible ways. First, a system may become underactuated when one or more of its actuators fail to work properly. A typical example is a conventional robot manipulator with failed joints. Usually, these joints are locked and the manipulator resumes its task, whenever possible. Control of this type of underactuated system is important from a fault-tolerance point-of-view, for sometimes it is necessary that the robot complete its task before repairs can be performed (e.g., when the robot is mounted on a space structure and has to return to its home position for stowage, or when the robot is manipulating nuclear waste and must store it before humans can safely approach the robot).

A system may also become underactuated when it is specifically designed to be so, or when the operator deliberately keeps one or more of the actuators available inoperative during a task. One example is a hyper-redundant snake-like robot, which, because of its high dexterity, is capable of moving in highly cluttered environments. Equipping all joints of such robot with actuators makes it heavier and more bulky than is sometimes necessary. By reducing the number of actuators without reducing the number of joints and by devising the proper control system, one obtains a mechanism whose energy consumption is smaller, yet whose dexterity is maintained. A biological system which can be considered underactuated in this sense is the human body. When gymnasts perform acrobatic maneuvers on a high bar, they are able to rotate about their wrists by actuating the muscles on their hips and knees. The wrist is therefore a joint kept unactuated, whose displacement can be controlled by the actuation of other joints. Normal walking is yet another example of how humans are able to control several DOFs with few actuators. There is evidence that, in this situation, the knee is powered by the muscles at the hip, and its muscles are utilized mainly for stopping the knee at the end of the swing [10].

Finally, underactuation may be an inherent property of the system. A typical example is a free-floating space satellite equipped with a conventional robot manipulator. While the DOFs of the manipulator can be controlled by its own joint motors, the position and orientation of the satellite in inertial space cannot. The satellite can be represented by the end-effector of a virtual unactuated spherical joint [23]; therefore, the overall satellite and manipulator system is underactuated. A robot manipulator with flexible links is another mechanical system inherently underactuated. While its joints can be controlled directly by the actuators located there, the internal DOFs representing the flexibility of the links cannot. It has been shown that such systems can be modeled as combinations of conventional and unactuated joints, therefore, as underactuated systems [54].

From the discussion above, it is clear that underactuated systems are abundant both as mechanical as well as biological "machines." Despite the large physical differences between the existing underactuated systems with unactuated joints, they share several common characteristics:

(1) *dynamic coupling*: the DOFs of an underactuated system are usually coupled by dynamic forces. For example, the joints of a robot manipulator are coupled due to the off-diagonal terms in their inertia matrix [22]; a robot manipulator mounted on a satellite is dynamically coupled to its free-floating base [51]; and the human lower leg is coupled to the thigh during walking [18]. Dynamic coupling is usually regarded as an undesirable characteristic of regular robotic mechanisms, for it complicates the control of the DOFs [22]. Control of underactuated systems, on the other hand, requires the existence of dynamic coupling, because control of the unactuated DOFs can be performed only indirectly through the application of forces at the actuated ones;

(2) *controllability of the unactuated DOFs and sequential control*: all DOFs of an underactuated system can usually be controlled, although not concurrently, directly or indirectly by the actuated DOFs. For example, unactuated joints of a robot manipulator can be controlled to a fixed position via the application of torques at the actuated ones [34]; and the attitude of a satellite can be controlled by motions of its on-board manipulator [29]. Consequently, it is usually possible to control all DOFs to an equilibrium position by the use of a well-designed control sequence. By control sequence, we mean the sequential control of some DOFs, followed by the control of some others, and so on, until all DOFs have been controlled. When it is possible to prevent some or all of the DOFs from moving, sequential control is facilitated, because the DOFs kept stationary do not interfere with the ones being controlled. When this is not possible, sometimes it is still possible to control all DOFs by forcing previously controlled ones to follow closed paths that bring them back to their position. Sequential control has been successfully applied to automobiles [31], robot manipulators with unactuated joints [1], [30], [47], [56] and space manipulators [29];

(3) *nonholonomic constraints:* there usually exist nonholonomic constraints associated with an underactuated system's dynamic equations. The term *holonomous*, used to describe a system whose generalized coordinates do not depend on their own time derivatives, was coined by Hertz in 1894 [42]. These constraints arise naturally from the lack of actuation of some of the DOFs. For example, free-floating space satellites must obey the law of conservation of angular momentum (when no external momentum is applied to the system) [8]; automobiles cannot usually drive sideways, a restriction that can be expressed as a nonholonomic constraint [20]; and manipulators with unactuated joints cannot have all of their joint accelerations controlled concurrently, another restriction which can be expressed as a nonholonomic constraint [34]. These constraints usually do not allow one to apply classical control techniques to the control of underactuated systems; on the other hand, it is because of the nonholonononiic constraints that one is able to control the unactuated DOFs of underactuated systems such as automobiles [20] and space manipulators [29];

(4) *nonlinearity and dependency on kinematic and dynamic parameters*: the dynamic equations of underactuated systems are in general nonlinear, and their kinematic and dynamic models may depend not only on their structure but also on their inertial characteristics. For example, the generalized Jacobian matrix of a space manipulator [49] depends on the value of the links' length as well as on their mass and inertia. Consequently. parameter uncertainty is more critical for this class of systems than for fully actuated ones. It is the controller designer's responsibility to guarantee robustness against uncertainties by proper utilization of robust control laws.

In this dissertation, it is our goal to study these fundamental characteristics toward the aim of developing modeling and control methods for underactuated systems. We focus on the class of serial chain underactuated manipulators with joints connected by rigid links. We assume that all joints are equipped with position sensors, and that the unactuated joints are equipped with on-off brakes. The use of brakes facilitates the control of the underactuated manipulator by allowing one to freeze some or all of the unactuated DOFs while controlling others. In other words, we take advantage of mechanical hardware to achieve precise control of an underactuated system.

We study the kinematic and dynamic modeling of underactuated manipulators with unactuated joints, including their dexterity and dynamic coupling between the actuated and unactuated joints; study control methods to robustly drive all joints of an underactuated manipulator to an equilibrium position, or a set of joints or Cartesian coordinates to follow a desired trajectory; and study motion planning methods to drive an underactuated manipulator to a desired configuration among obstacles in its workspace.

1.2 Literature review

The first topic that we studied in this dissertation related to the modeling and control of underactuated manipulators is the dexterity of these mechanisms. Pradeep et al. [39] analyzed qualitatively, but not quantitatively, the capabilities of three commercial manipulators with a failed joint. Roberts [40] and Roberts and Maciejewsky [41] studied the manipulability [53] and worst-case dexterity [26] of redundant manipulators with any number of failed joints. Their objective was to determine optimal manipulator configurations that would keep the manipulability or worst-case dexterity above a prescribed minimum value were a joint (or joints) to fail. No work has yet considered quantifying the dexterity of underactuated manipulators in a general sense, nor how to maximize it by changing the positions of the unactuated joints,

The modeling of underactuated manipulators for control purposes, and the analysis of the resulting dynamic equations, have been presented in different forms by several authors. Arai and Tachi [1] derived an openloop relationship between the forces applied at the actuated joints and the accelerations of the unactuated ones. With this relationship they obtained a sufficient condition for controllability of the unactuated joints around equilibrium points, but not over the entire state space of the unactuated joints' positions and velocities. Jain and Rodriguez extended their coordinate free spatial modeling method [16] to underactuated manipulators [17]. Their analysis include the definition of an underactuated manipulator's generalized Jacobian, which relates Cartesian space accelerations to accelerations of the actuated joints; and the disturbance Jacobian, which relates accelerations of the unactuated to those of the actuated joints. The disturbance Jacobian was not quantitatively analyzed to yield a dynamic coupling measure for underactuated manipulators. Finally, Mukherjee and Chen [30] modeled a satellite-mounted underactuated manipulator utilizing Hamiltonian dynamics, which are useful because the Hamiltonian of a dynamic system can be used as a basis for candidates to Lyapunov functions.

It is in the area of control of underactuated manipulators that the majority of past research concentrates. Past work can be broadly classified in two areas: (1) control of a subset of the joints to equilibrium points or all joints to an equilibrium manifold, and (2) control of all joints to an equilibrium point. In the first category, Berkemeier and Fearing [5] proposed using the Acrobot, a 2-link underactuated manipulator, as a hopping machine. Hauser and Murray [13] proposed **a** method to control the Acrobot in the neighborhood of the manifold of inverted equilibrium positions. Meldrum et al. [28] proposed a control method to make an underactuated manipulator follow a trajectory in Cartesian space (based on Jain and Rodriguez's modeling of underactuated manipulators [17]). Mukherjee and Chen [30] proposed controlling the actuated joints of a space manipulator while guaranteeing the stability of the unactuated joints and of the free-floating base. Nakakuki et al. [32] showed how a 3-link mechanism with two actuators and no fixed base is able to stand up, fall, and climb stairs. Oriolo and Nakamura's pioneer work [34] demonstrated that the nonholonomic constraints inherent to an underactuated manipulator generally involve joint accelerations in addition to joint velocities. They also addressed the control of the unactuated joints to an equilibrium position. Papadopoulos and Dubowsky [36] proposed to control the failed joint of a space manipulator to an equilibrium point while guaranteeing that the active joints and the base would come to **a** rest at the end of the motion. Takashima [48] demonstrated how a gymnast robot can perform acrobatic stunts. Perhaps most interestingly of all, Saito et al. [43] built a brachiation-type robot that is able to swing along horizontal parallel bars.

The second category of research on control of underactuated manipulators (control of all joints to an equilibrium point) can be subdivided into two groups, namely, those which assume that the unactuated joints are completely free of actuation, and those which assume they are equipped with brakes. Early work on control of underactuated manipulators with completely free unactuated joints focussed mainly on the control of the double pendulum and its variants [4], [45], [52]. More recently, focus has shifted to the control of all joints of a manipulator with any number of links to an arbitrary equilibrium position. Research on this area, though, is still in its infancy; it has thus far been demonstrated practically on a 2-link horizontally mounted arm by Suzuki et al. [47] and in simulation on a 3-link arm with two actuators by Arai [3]. Chung et al. [6] proposed and built special nonholonomic gears that allow one to drive all joints of a manipulator equipped with such gears to an equilibrium point.

Research on manipulators with unactuated joints equipped with brakes has been receiving a great deal of attention lately. Arai and Tachi [1] were the first to propose a 2-phase method to control all joints of an underactuated manipulator to an equilibrium position. In the first phase, the unactuated joints are controlled via their dynamic coupling with the actuated ones to a desired joint position, and are subsequently locked. In the second phase, the actuated joints are controlled as if the manipulator were a fully actuated one. Because this method can be used to position the manipulator's end-effector at a desired Cartesian position at the end of the two control phases, the authors named it point-to-point control, or PTP. A shortcoming of this method is that it can handle only manipulators with more actuated than unactuatedjoints. Arai et al. [2] later extended the method to make the end-effector follow a trajectory in Cartesian space, using the PTP control method to accelerate the end-effector in the direction of the trajectory. Yu et al. [56] proposed the PTP control of a 2-link manipulator with one unactuated joint via the first integral approach; this method, however, is based on the integral of the nonholonomic constraints and cannot be applied to general underactuated manipulators. Finally, Mukherjee and Chen [30] proposed PTP motion of a space mounted underactuated manipulator with the 2-phase control method of [1]; they did not, however, address the problem of controlling the satellite's orientation while driving the manipulator.

To this date there are no published works on the motion planning of underactuated manipulators. A large number of authors have proposed methods to plan the collision-free motion of velocity constrained nonholonomic systems among obstacles [9], [20], [31], [35], [44], [50]. These results, however, cannot be directly extended to the motion planning of underactuated manipulators because these systems' nonholonomic constraints involve joint accelerations in addition to joint velocities.

Despite all the work on control of underactuated manipulators, some topics remain to be addressed: the robust control of all joints to an equilibrium point in the presence of modeling error and external disturbances; the control of manipulators with more unactuated than actuated joints; and a method to plan collision-free trajectories for underactuated manipulators operating in a workspace populated with obstacles.

1.3 Overview

In this dissertation we discuss important problems related to the modeling and control of underactuated manipulators that have not been studied in any other work so far. Although an underactuated manipulator can be structurally identical to a fully actuated one, their dexterities differ because of the presence of unactuated joints. Furthermore, the dexterity of an underactuated manipulator varies as do the positions of the unactuated joints. We quantify the dexterity of underactuated manipulators, performing an analysis that is valid for any serial link manipulator, and for any dexterity measure. Because dexterity is a kinematic property of robot manipulators, we assume that the unactuated joints are locked in an arbitrary position within their joint limits. Noting that such locking position can be modified using the control methods we propose in this work (or one of the control methods described in Section 1.2), we propose an optimization index to determine the locking positions of the unactuated joints which maximize the dexterity of an underactuated manipulator. With this approach we intend to demonstrate that it is possible to operate a manipulator with less actuators than joints while maintaining a minimal amount of dexterity throughout the task execution.

To be able to control all the joints of an underactuated manipulator, one must derive its dynamic model and analyze the dynamic coupling between the actuated and the unactuated joints. This is important because the unactuated joints can be driven only indirectly, through their coupling with the actuated ones. We present the joint and Cartesian space models that will be utilized for model-based control, following closely the treatment of Arai and Tachi [1]. Based on the dynamic equations we propose the coupling index concept, to measure the dynamic coupling between the joints. The coupling index is used for analysis and design of underactuated manipulators, and for the optimal control of the unactuated joints.

Parameter uncertainty and external disturbances often interfere with the control of dynamic systems, and underactuated manipulators are not the exception. We propose a 2-phase robust controller to control all joints of an underactuated manipulator to an equilibrium point, assuming that the manipulator operates in a workspace free of obstacles. We investigate the controllability of the system, and demonstrate that controllability and the coupling index are related concepts. When the underactuated manipulator has more unactuated than actuated joints, the 2-phase control method is insufficient to allow one to control all joints to an equilibrium point. We propose an optimal control sequence of the unactuated joints to account for this case.

In practice, manipulators operate in workspaces populated with obstacles. For this reason, we propose a high-level motion planning method that generates collision-free trajectories to be followed by the manipulator. The method is valid for manipulators operating on gravity-free environments, or when the unactuated joints are not acted upon by gravitational forces.

1.4 Noinenclature

Consider an n-link, fixed-base, open chain, serial robot manipulator with rigid links. The joints and links are numbered from 1 to n, with joint 1 and link 1 being the closest to the base. We use q to represent the robot's $n \times 1$ joint vector, and τ to represent its $n \times 1$ torque vector. The dynamic parameters of link i, namely, its mass and inertia, are represented by m_i and I_i, respectively. The kinematic parameters length and location of the center of mass are represented by l_i and l_{c_i}, respectively.

Assume that $n_a < n$ joints of the manipulator are equipped with actuators, and that the remaining $n_p = n - n_a$ ones are equipped with on-off brakes. Let $I_a = \{a_1, \ldots, a_{n_a}\}$ be the set of indices corresponding to the location of the actuators among all n joints, and similarly let $I_p = \{p_1, \ldots, p_{n_p}\}$ be the set of indices corresponding to the location of the brakes. We refer to the joints equipped with actuators as the *active joints,* and to the ones equipped with brakes as the *passive joints*. To distinguish between the active and passive joints we rewrite the joint vector q as:

$$q = \begin{bmatrix} q_a \\ q_p \end{bmatrix} \tag{1.1}$$

where q_a is the $n_a \times 1$ vector representing the position of the active joints, and q_p is the $n_p \times 1$ vector representing the position of the passive ones. To further distinguish among the passive joints that are locked or unlocked at a certain instant, we partition the vector q_p as:

$$q_p = \begin{bmatrix} q_u \\ q_l \end{bmatrix} \tag{1.2}$$

We also partition the set of indices I_p into two sets I_u and I_l respectively corresponding to the location of the unlocked and the locked passive joints. While the sets I_a and I_p do not change for a given underactuated manipulator structure, the sets I_u and I_l are modified every time a passive joint is locked or unlocked. At any given instant, the relations $I_u \cap I_l = \emptyset$ and $I_u \cup I_l = I_p$ must be satisfied. We graphically represent the active, locked passive, and unlocked passive joints with the color convention of Figure 1.1.

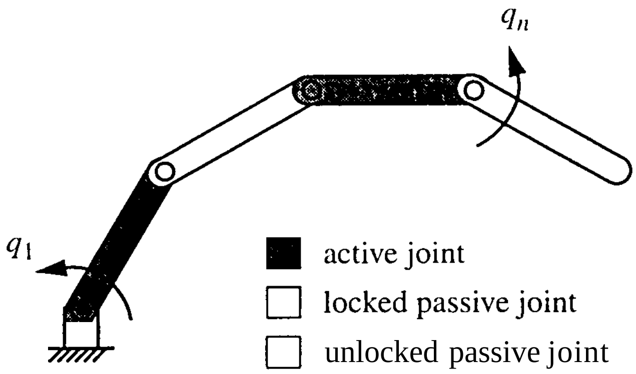


Figure 1.1: Graphical representation of an underactuated manipulator and its active, unlocked passive, and locked passive joints.

1.5 List of Symbols

$[x]$	dimension of the vector x
$\sigma_i(A)$	i-th singular value of the matrix $\boldsymbol{A}$
$A^{\#}$	pseudo-inverse of the matrix $\boldsymbol{A}$
$(A)_i$	i-th column of matrix $\boldsymbol{A}$
$s_{xy...}$	$\sin(q_x + q_y + \dots)$
$c_{xy...}$	$\cos(q_x + q_y + \dots)$
m_i	mass of the i-th link
I_i	inertia of the i-th link
l_i	length of the i-th link
l_{ci}	distance between the i-th joint and the center of mass of the i-th link
J	Jacobian matrix
K	pseudo-inverse of the Jacobian matrix
M	inertia matrix
W	inverse of the inertia matrix
C	matrix of Coriolis and centrifugal forces
G	vector of gravitational forces
F	vector of frictional forces
n	number of joints

n_a	number of active joints
n_p	number of passive joints
n_u	number of unlocked passive joints
n_l	number of locked passive joints
q	joint vector
q_a	active joint vector
q_p	passive joint vector
q_u	unlocked passive joint vector
q_l	locked passive joint vector
I_a	set of indices corresponding to the location of the active joints
I_p	set of indices corresponding to the location of the passive joints
I_u	set of indices corresponding to the location of the unlocked passive joints
I_l	set of indices corresponding to the location of the locked passive joints
x	vector of Cartesian coordinates
x_a	vector of active Cartesian coordinates
x_p	vector of passive Cartesian coordinates
V	volume of the manipulator's workspace
w	manipulability index
x^d	desired (set-point) value of the vector x
$\tilde{x}$	time-varying error between the desired and the current value of the vector x
s	sliding surface
Γ	sliding surface's time constant
P	variable structure controller's gain
ε	variable structure controller's boundary layer width

Chapter 2

Dexterity of Underactuated Manipulators

In this chapter, we quantify the kinematic dexterity of an underactuated manipulator with an arbitrary number of passive joints, and compare it to the dexterity of a redundant fully actuated manipulator. We then propose an optimization index that allows us to determine the positions at which the passive joints should be locked for the sake of maximizing the underactuated manipulator's dexterity. The optimization is performed with respect to the locking positions of the passive joints, for which analytical methods or an exhaustive search quickly yield the global optimal solution. We consider here three important dexterity measures: the volume of the reachable workspace, the capability of reaching a set of points in Cartesian space, and the nianipulability index [53].

2.1 Dexterity analysis and optimization

Although the kinematic structure of an underactuated manipulator is identical to that of a similar fully actuated one, in general their dexterities are different. This is due to the presence of locked passive joints, which may reduce the dexterity. Study of the dexterity of underactuated manipulators is important to determine whether such manipulators possess sufficient dexterity to perform their assigned tasks despite the presence of passive joints in their mechanisms. Early work by Pradeep et al. [39] analyzed the capabilities of three different commercial manipulators with a locked failed joint. The analysis is qualitative in nature and is not valid for a general manipulator. More recently, Roberts [40] and Roberts and Maciejewski [41] discussed the decrease in a manipulator's kinematic manipulability index and worst-case dexterity due to the failure and locking of one or more of its joints. In their work the authors were concerned with finding pre-failure optimal manipulator configurations that guarantee that the post-failure manipulability is maximized. The difference between their work and the analysis in this chapter is that we are concerned with measuring dexterity as a general quantity, and in determining the positions of the passive joints which maximize the dexterity.

Let $D(q)$ represent the dexterity measure of an n-link redundant fully actuated manipulator locally evaluated at the configuration q. Suppose now that joints $p_1, \ldots, p_{n_l}$ in the set I_p are passive joints locked at an arbitrary position within their joint limits. We represent by $D_{I_p}(q)$ the dexterity of the corresponding underactuated manipulator at the configuration q. The *relative dexterity loss* is defined as:

$$\tilde{D}_{I_p}(q) = 1 - \frac{D_{I_p}(q)}{D(q)} \tag{2.1}$$

The relative dexterity loss measures the decrease in dexterity of the fully actuated manipulator when one or more of its joints fail or are intentionally designed as passive joints. This index allows one to verify whether the underactuated manipulator's dexterity is sufficiently large for completion of the manipulator's assigned task.

Ideally, one would wish that the relative dexterity loss be as small as possible, as this would imply that the presence of passive joints has little affect on the manipulator's capabilities. One might be interested in determining the positions at which the passive joints should be locked for the sake of minimizing the relative dexterity loss. This problem can be cast as an optimization problem, with the optimization index given by the value of the relative dexterity loss:

$$q_p^* = \arg\min[J_{D_{I_p}}(q)] = \arg\min[\tilde{D}_{I_p}(q)] \tag{2.2}$$

Global minimization of the optimization index $J_{D_{I_p}}(q)$ with respect to the positions of the passive joints yields the global minimum value of the underactuated manipulator's relative dexterity loss. At times, the relative dexterity loss may depend on the positions of the active joints as well as on the positions of the passive ones. Consequently, minimization of $J_{D_{I_p}}$ with respect to q_p only is an ill-defined problem. One possible solution is to average the optimization function with respect to the positions of the active joints, and then minimize the resulting averaged optimization function:

$$\bar{J}_{D_{I_p}}(q_p) = \frac{\int J_{D_{I_p}}(q)dq_a}{\int dq_a} \tag{2.3}$$

2.2 Workspace

Being able to reach a large number of points in Cartesian space is usually a design criterion and a desirable characteristic of a robot manipulator. When the manipulator *has* both active and passive joints, it is important to guarantee that the number of reachable points (which compose the reachable workspace) is as large as possible. Denote by V the volume or area of a fully actuated manipulator's workspace. This quantity does not depend on the local configuration of the manipulator and therefore is not a function of q. Denote by V_{I_p} the volume or area of the corresponding underactuated manipulator. This quantity depends only on the positions where the passive joints are locked, and is therefore a function of q_p. The *relative workspace loss* is defined as:

$$\tilde{V}_{I_p}(q_p) = 1 - \frac{V_{I_p}(q_p)}{V} \tag{2.4}$$

Clearly, $V \neq 0$ and the relative workspace loss varies between 0 and 1. If it is equal to zero, the presence of passive joints does not incur reduction of the manipulator's workspace; the closer the loss gets to 1, the smaller the volume (or area) of the underactuated manipulator's workspace. The relative workspace loss has been mentioned in the literature [39], but there has not been a quantitative analysis of it so far.

To compute the angle at which joints q_p should be locked for relative workspace loss minimization we minimize the following optimization function:

$$J_{V_{l_p}}(q_p) = 1 - \frac{V_{l_p}(q_p)}{V} \tag{2.5}$$

Example 2.1 Consider a 3-link planar rotary manipulator with link lengths $l_i = 0.3$ m, $i = 1, 2, 3$, as shown in Figure 2.1. We assume without loss of generality that **all** joint limits are equal to $\pm\pi$ rd; different joint limits can be easily considered by simply limiting the search for the global minimum of J_V to the angles between the new joint limits. The manipulator's original workspace has an area equal to:

$$V = \pi(l_1 + l_2 + l_3)^2 = 0.81\pi \text{ m}^2 \tag{2.6}$$

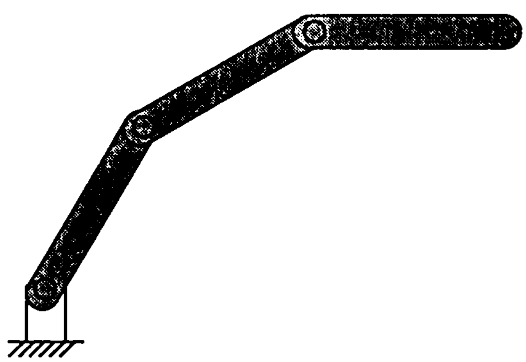


Figure 2.1: Three-link planar rotary manipulator.

If the manipulator is equipped with one passive joint at joint p, which is kept locked during, e.g., manipulation tasks, the manipulator is reduced in practice to a 2-link mechanism with a reduced workspace. The underactuated manipulator workspace's is an annulus with inner and outer radii $R_i(q_p)$ and $R_o(q_p)$, respectively. Its area is equal to:

$$V_p(q_p) = \pi[R_o^2(q_p) - R_i^2(q_p)] \tag{2.7}$$

Table 2.1 presents the expressions for R_i, R_o and V_p for $p = 1, 2, 3$. The quantity l_{12} is the distance between the first and third joints and l_{23} is the distance between the second joint and the end-effector:

$$\begin{aligned} l_{12} &= \sqrt{l_1^2 + l_2^2 + 2l_1 l_2 \cos(q_2)} \\ l_{23} &= \sqrt{l_2^2 + l_3^2 + 2l_2 l_3 \cos(q_3)} \end{aligned} \tag{2.8}$$

Table 2.1: Boundary radii and workspace areas for all possible locations of one passive joint on a 3-link planar manipulator.

passive joint	$R_i(q_p)$	$R_o(q_p)$	$V_p(q_p)$
1	$\lvert l_2 - l_3 \rvert$	$l_2 + l_3$	$4\pi l_2 l_3$
2	$\lvert l_{12} - l_3 \rvert$	$l_{12} + l_3$	$4\pi l_{12} l_3$
3	$\lvert l_1 - l_{23} \rvert$	$l_1 + l_{23}$	$4\pi l_1 l_{23}$

Figure 2.2 presents the value of the relative workspace loss for all possible locations of the passive joint, as the passive joint's angle varies from $-\pi$ to π rd. Figure 2.3 presents both the original workspace V (a circle with radius 3 m) and the workspace V_2 when joint 2 is passive and is locked at a few selected angles. We conclude from these figures that, whenever joints 2 or 3 are passive, they must be locked at 0 rd if the underactuated manipulator's workspace is to be maximized. One can also conclude from Figure 2.2 that the closer to π rd joints 2 or 3 are locked, the smaller the workspace, which vanishes at that angle (see also the lower right graph in Figure 2.3, which shows that, when q_2 is locked at π rd, the workspace is a circumference with zero area). On the other hand, when joint 1 is the passive one, the workspace area is constant, independent of the locking angle. In this case one might choose to lock the first joint at the angle q_1 that maximizes another dexterity measure.

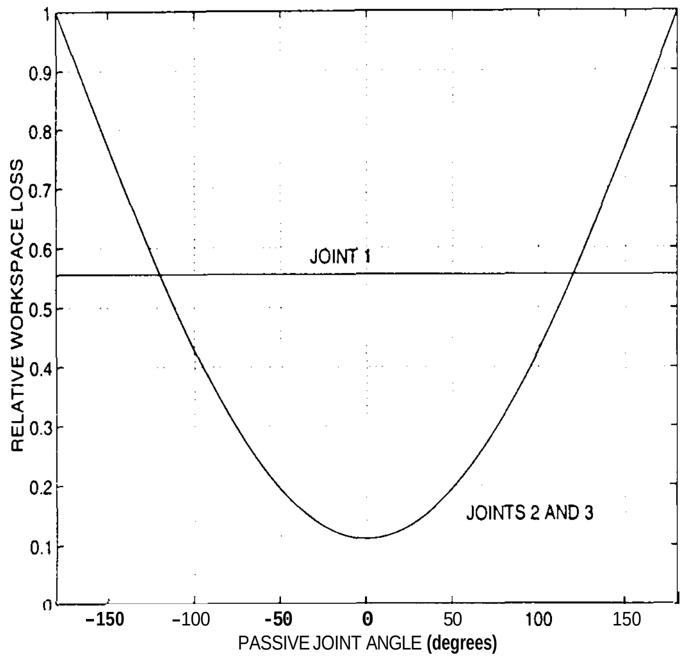
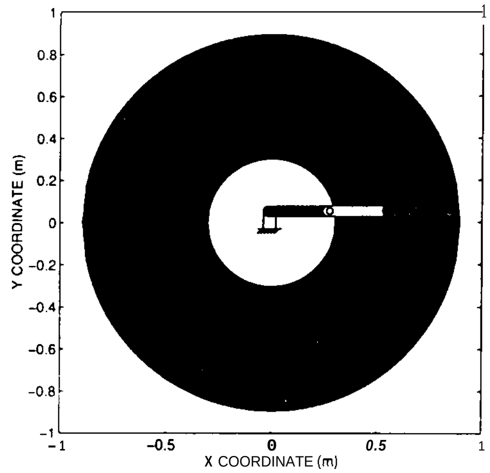


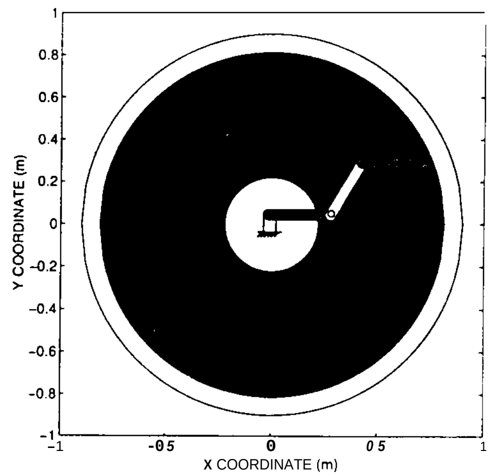
Figure 2.2: Relative workspace loss for all possible locations of a 3-link planar manipulator's passive joint.

2.3 Reachability

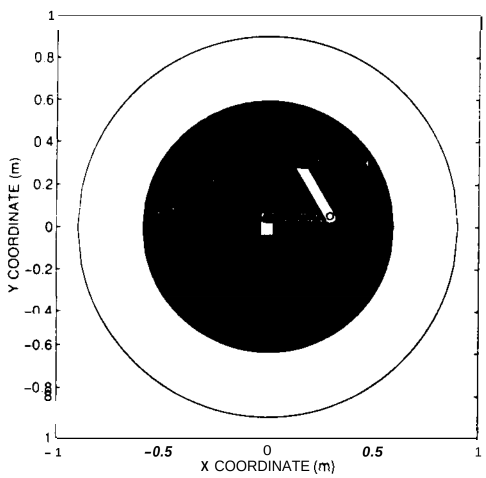
Another important design criterion of a robot manipulator is its capability to reach a number of pre-defined points in Cartesian space for manipulation tasks. This property is not guaranteed to exist when the manipulator contains a passive joint. even if the passive joint is locked at the angle which maximizes the manipulator's workspace volume. To understand this, consider again Figure 2.3; if joint 2 is locked at $q_2 = 0$ rd, then points close to the robot's base cannot be reached. On the other hand, locking joint **2** at $q_2 = 2\pi/3$ rd does allow the end-effector to reach those points. We propose to measure the *reachability* of the underactuated manipulator by computing the sum of the individual distances between all points that must be reached by the manipulator's end-effector and the respective closest point in the workspace boundary:



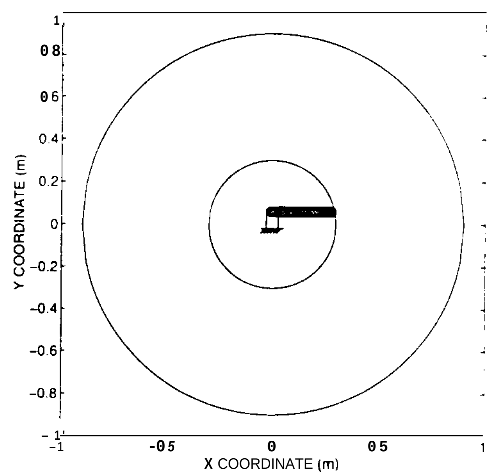
(a) $q_2 = 0$ rd



(b) $q_2 = \pi/3$ rd



(c) $q_2 = 2\pi/3$ rd



(d) $q_2 = \pi$ rd

Figure 2.3: Workspace of a 3-link planar manipulator with joint 2 passive. The outermost circle represents the boundary of the corresponding fully actuated manipulator's workspace.

(a) The workspace has maximum area when joint 2 is locked at $q_2 = 0$ rd;

(b), (c) as the locking angle increases, the area, as well as the location of the workspace, vary;

(d) if joint 2 is locked **at** $q_2 = \pi$ rd the workspace reduces to a circumference with zero area.

$$d_{I_p}(q_p) = \sum_{i=1}^{P} \|X_i - B_i(q_p)\| \tag{2.9}$$

In Equation (2.9), P is the number of Cartesian points to be reached, X_i are the coordinates of the i-th point, and $B_i(q_p)$ is the point in the workspace boundary closest to X_i. The coordinates of B_i depend on the positions where the passive joints are locked, because the location of the workspace in inertial space varies as q_p varies. Since workspace boundaries are directly related to kinematic singularities, a large reachability implies that the pre-defined Cartesian points will, with a greater probability, be reached at nonsingular configurations of the manipulator.

Usually more than one locking angle will allow the end-effector to reach a set of pre-defined points in Cartesian space. To compute the optimal one, we propose to minimize the *relative reachability loss* defined as:

$$\tilde{d}_{I_p}(q_p) = 1 - \frac{d_{I_p}(q_p)}{d} = 1 - \frac{\sum_{i=1}^{P} \|X_i - B_i(q_p)\|}{\sum_{i=1}^{P} \|X_i - B_i\|} \tag{2.10}$$

The denominator, d, in the righthand side of (2.10) is the reachability of the corresponding fully actuated manipulator. The quantity d is independent of the local configuration of the manipulator and, therefore, is constant. Clearly, if $d = 0$, then all pre-defined Cartesian points are situated at the boundary of the workspace. Consequently, if the manipulator is equipped with passive joints, then either $d_{I_r} = 0$, or one or more of the Cartesian points will not be reachable by the end-effector. We must then consider only the case $d \neq 0$.

The relative reachability loss indicates, on a 0 to 1 scale, how much closer to a workspace boundary a Cartesian point becomes when passive joints are present in the manipulator. It is important to mention that *the minimization of* $\tilde{d}_{I_p}(q_p)$ *is carried only over the joint angles* q_p *for which all points* X_i *are actually reachable*. If any of the X_i is not reachable for all possible q_p, then the underactuated manipulator cannot complete its task because of the presence of the passive joints, and is defined as non-fault tolerant [38]. Minimization of $\tilde{d}_{I_p}(q_p)$ effectively leads one to choose the optimal workspace whose boundaries are as far as possible from the desired Cartesian points.

Example 2.2 Consider the 3-link manipulator in Figure 2.1, programmed to reach the Cartesian point X = (0.3, 0.3). Suppose that joint 1 is passive. **As** its locking angle varies, so does the location of the reachable workspace. Figure **2.4** shows the value of $\tilde{d}_1(q_1)$ as a function of q_1. **As** expected, this is a continuous function, which attains the value 1 at the angles for which the Cartesian point is exactly at the boundary of the workspace. It can be seen that the Cartesian point (0.3, 0.3) cannot be reached whenever joint 1 is locked outside the range [-65°,155°]. Inside this range, the Cartesian point is optimally reached (i.e., it is farthest from the workspace boundary) when $q_1^* = \pi / 4$ rd.

Figitre 2.5 presents results for the set of Cartesian points $X = \{(-0.3,0.3), (-0.3,0.6), (-0.6,0.3), (-0.6,0.6)\}$. The optimal locking angle is now $q_1^* = 3\pi / 4$ rd. Figure 2.6 presents the optimal workspace that enables the manipulator to reach all four desired Cartesian points.

One will note that the curve in Figure **2.4** is continuous, while that in Figure 2.5 is not. This is due to the fact that, when several Cartesian points have to be reached, one cannot expect that in general all of them will be located at the workspace boundary for sonic position of the passive joint (and consequently yield $\tilde{d}_1 = 0$).

2.4 Manipulability

The manipulability index introduced by Yoshikawa [53] is a measure of the dexterity of a robot manipulator, which geometrically defines how far from a kinematic singularity a manipulator is. The manipulability index has been used for years as a tool for posture optimization, niotion planning, and singularity avoidance of manipulators. One of the most commonly used manipulability indices is defined as:

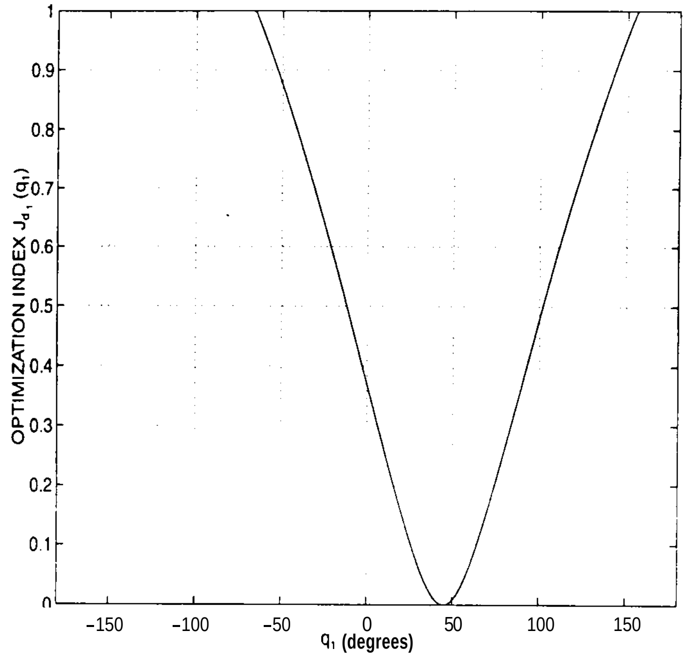


Figure 2.4: Relative reachability loss of the 3-link planar manipulator evaluated for the Cartesian point $X = (0.3, 0.3)$.

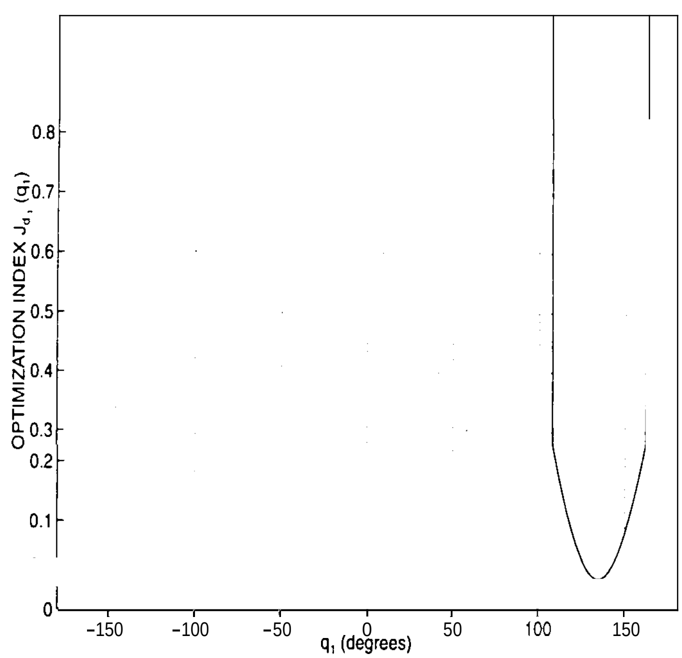


Figure 2.5: Relative reachability loss of the 3-link planar manipulator evaluated for the set of Cartesian points $X = \{(-0.3, 0.3), (-0.3, 0.6), (-0.6, 0.3), (-0.6, 0.6)\}$.

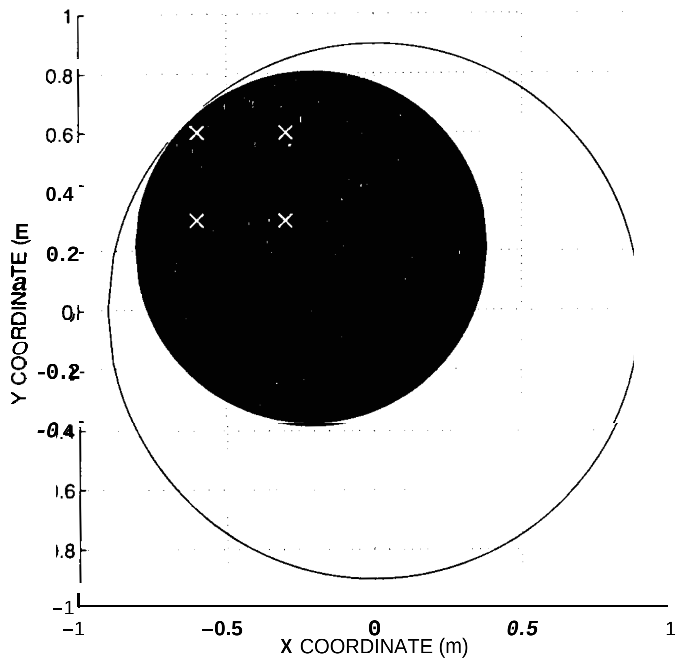


Figure 2.6: Optimal workspace for the set of Cartesian points $X = \{(-0.3, 0.3), (-0.3, 0.6), (-0.6, 0.3), (-0.6, 0.6)\}$.

$$w(q) = \sqrt{\det[J(q)J(q)^T]} \tag{2.11}$$

where $J(q)$ is the position-dependent manipulator's Jacobian, which transforms velocities in Cartesian space to joint velocities according to the well-known linear equation $\dot{x} = J(q)\dot{q}$.

Let j_i represent the i-th column of J. When the manipulator is equipped with a set of locked passive joints, located at joints $I_p = \{p_1, \ldots, p_{n_p}\}$, its new Jacobian is obtained by deleting from J all columns corresponding to the set of passive joints I_p:

$$J_{I_p} = \left[j_1 \cdots j_{p_1-1}\ j_{p_1+1} \cdots j_{p_n-1} \cdots {}_{+1} \cdots \right] \tag{2.12}$$

The manipulability of the resulting underactuated manipulator is computed as:

$$w_{I_p}(q) = \sqrt{\det[J_{I_p}(q)J_{I_p}(q)^T]} \tag{2.13}$$

The ***relative manipulability loss*** is defined as the complement in $[0, 1]$ of the relative manipulability index [41]:

$$\tilde{w}_{I_p}(q) = 1 - \frac{w_{I_p}(q)}{w(q)} \tag{2.14}$$

The relative manipulability loss indicates, on a 0 to 1 scale, how much closer to a singularity a manipulator at configuration q becomes when some of its joints are passive. If the manipulability index is zero, we can consider instead the constrained manipulability index [41], which is equal to the product of the non-zero singular values of J. In this case the relative manipulability loss will indicate how much further the dexterity of the manipulator decreases when passive joints are present.

To compute the joint angles q_p which minimize $\tilde{w}_{I_p}$, and therefore minimize the manipulability loss, we propose to minimize the optimization function:

$$J_{w_{I_p}}(q) = 1 - \frac{w_{I_p}(q)}{w(q)} \tag{2.15}$$

Usually the Jacobian, and therefore the manipulability, depends on the values of **all** joint angles. Consequently, minimization of J_w with respect to q_p only is an ill-defined problem. When this is the case we choose to minimize the averaged optimization index as given by Equation (2.3).

Example 2.3 Consider again the 3-link manipulator in Figure 2.1. Its Jacobian is equal to:

$$J = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} - l_3 s_{123} & -l_2 s_{12} - l_3 s_{123} & -l_3 s_{123} \\ l_1 c_1 + l_2 c_{12} + l_3 c_{123} & l_2 c_{12} + l_3 c_{123} & l_3 c_{123} \end{bmatrix} \tag{2.16}$$

and its manipulability index is given by:

$$w = \sqrt{(l_2 l_3 s_3)^2 + (l_2 l_3 s_3 + l_1 l_3 s_{23})^2 + (l_1 l_2 s_2 + l_1 l_3 s_{23})^2} \tag{2.17}$$

Assume that one of the joints is passive; according to the location of this passive joint, the Jacobian reduces to:

$$J_1 = \begin{bmatrix} -l_2 s_{12} - l_3 s_{123} & -l_3 s_{123} \\ l_2 c_{12} + l_3 c_{123} & l_3 c_{123} \end{bmatrix} \tag{2.18}$$

$$J_2 = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} - l_3 s_{123} & -l_3 s_{123} \\ l_1 c_1 + l_2 c_{12} + l_3 c_{123} & l_3 c_{123} \end{bmatrix} \tag{2.19}$$

$$J_3 = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} - l_3 s_{123} & -l_2 s_{12} - l_3 s_{123} \\ l_1 c_1 + l_2 c_{12} + l_3 c_{123} & l_2 c_{12} + l_3 c_{123} \end{bmatrix} \tag{2.20}$$

Accordingly, the manipulability indices are given by:

$$w_1 = |l_2 l_3 s_3| \tag{2.21}$$

$$w_2 = |l_2 l_3 s_3 + l_1 l_3 s_{23}| \tag{2.22}$$

$$w_3 = |l_1 l_2 s_2 + l_1 l_3 s_{23}| \tag{2.23}$$

It is interesting to note the geometric significance of the manipulability index of underactuated manipulators. For example, w_1 is numerically equal to twice the area formed by links 2 and 3 (see Figure **2.7).** When these links are aligned, the area is zero, and so is the manipulability. Because this triangle's area is not a function of q_1, if joint 1 is passive it can be locked arbitrarily within its allowable range for the sake of minimization of the relative manipulability loss. **As** for w_2, it is equal to twice the sum of the area of the triangle formed by links 2 and 3 and that of an imaginary triangle formed by links 1 and 3, if link 3 were located at the end of link 1. **An** analogous interpretation is valid for w_3.

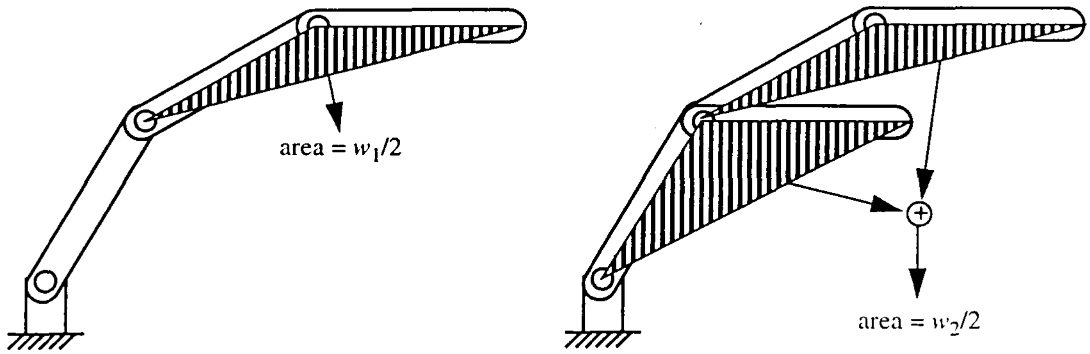


Figure 2.7: Geometric interpretation of the manipulability of underactuated manipulators.

The manipulability when joint 2 is passive is a function of both q_2 and q_3. To find the minimum of $J_{w_2}(q_2)$ we use Equation (2.3):

$$q_2^* = \arg\min[\bar{J}_{w_2}(q_2)] = \arg\min\left[\frac{\int_{-\pi}^{\pi} J_{w_2}(q_2, q_3)dq_3}{\int_{-\pi}^{\pi} dq_3}\right] \tag{2.24}$$

Figure 2.8 presents the value of $\bar{J}_{w_2}(q_2)$ as a function of q_2, as well as the value of $\bar{J}_{w_3}(q_3)$ as a function of q_3. Clearly, the relative manipulability loss is minimized when joint 2 is locked at $q_2 = 0$. **An** analogous result is obtained when one computes the q_3 that minimizes J_{w_3}; namely, $q_3^* = 0$. These results indicate that, for the sake of maximizing the manipulability of a 3-link planar manipulator with one passive joint located at either joints 2 or 3, one must lock the passive joint at its maximum extended position.

Although we presented only examples of underactuated manipulators with one passive joint, the method proposed accommodates the presence of several passive joints. When optimizing the workspace or the reachability of iinderactuated manipulators, one has to search for the global minimum of the optimization functions through the n_p-dimensional **space** consisting of all passive joints' positions. When optimizing the manipulability, one

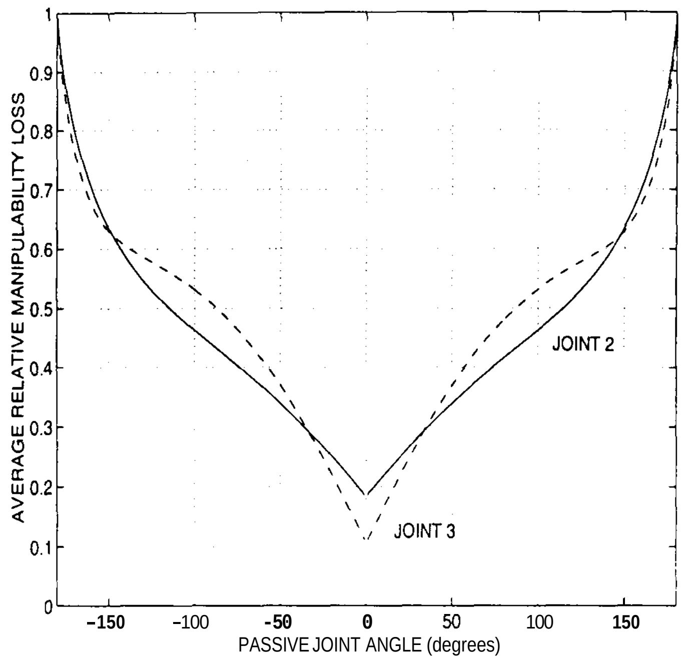


Figure 2.8: Average relative manipulability loss of a 3-link planar manipulator when joint 2 or joint 3 are passive.

first has to delete from the Jacobian matrix all columns corresponding to all passive joints, and then proceed to search for the global minimum on an n_p-dimensional space. Of course, as more passive joints are added to the manipulator, analytical and exhaustive search solutions become more and more difficult to obtain because of the larger dimension of the search space. If that is the case, one may resort to numerical optimization techniques to compute the optimal locking angle of the passive joints.

Chapter 3

Dynamic Modeling of Underactuated Manipulators

In the previous chapter, we studied the dexterity properties of underactuated manipulators. In this chapter, we will model their dynamic characteristics. Based on the dynamic models, we define a coupling index to measure the dynamic coupling between the active and the passive joints. The coupling index characterizes an important property of underactuated manipulators, and will be used for analysis and design of underactuated manipulators, as well as controllability analysis and control.

3.1 Joint space modeling

Underactuated manipulators differ from fully actuated ones in that the former are equipped with a number of actuators that is always smaller than the number of degrees-of-freedom (DOFs). Therefore, not all DOFs can be actively controlled concurrently. For example, with a 3-link planar manipulator equipped with two actuators, one can control concurrently two of the manipulator's joints, but not all of them. To control all joints of an underactuated manipulator, one must exchange the joints being controlled. This principle was first proved by Arai and Tachi [1] using linearized dynamics arguments, and is the basis for the joint and Cartesian spaces modeling that we will present in the sequel. Because at most n_a generalized coordinates (e.g., joint angles or Cartesian variables) can be controlled at any given instant, the vector of generalized coordinates is divided into two subsets, respectively representing the *active generalized coordinates* (i.e., the ones being controlled at any given instant) and the *passive generalized coordinates* (i.e., the remaining ones). A second-order differential equation relates the accelerations of the active and the passive generalized coordinates, and will be made explicit as we develop the dynamic models.

Consider an n-link, open chain, serial, underactuated robot manipulator with rigid links. Let q represent its joint vector and τ represent its torque vector. The dynamic equations of tlie manipulator are found in closed-form via the classical Lagrangian approach as [7]:

$$\tau = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + F(q, \dot{q}) \tag{3.1}$$

In Equation (3.1), M is the $n \times n$ symmetric, positive-definite inertia matrix, C is the $n \times n$ matrix of Coriolis and centrifugal terms, G is the $n \times 1$ vector of gravitational torques, and F is the $n \times 1$ vector of frictional torques. The inertia matrix and the gravitational torques are functions only of the manipulator's joint positions, while the matrix of Coriolis and centrifugal terms and the frictional torques are functions of the joint positions and velocities. For convenience, we combine tlie vectors in the right-hand side of (3.1), except for $M\ddot{q}$, in the vector of noninertial torques b:

$$\tau = M(q)\ddot{q} + b(q, \dot{q}) \tag{3.2}$$

To distinguish among the motions and torques of the active and passive joints, we partition Equation (3.2) as:

$$\begin{bmatrix} \tau_a \\ \tau_p \end{bmatrix} = \begin{bmatrix} M_{aa} & M_{ap} \\ M_{pa} & M_{pp} \end{bmatrix} \begin{bmatrix} \ddot{q}_a \\ \ddot{q}_p \end{bmatrix} + \begin{bmatrix} b_a \\ b_p \end{bmatrix} \tag{3.3}$$

where the subscripts a and p denote quantities related to the active and passive joints, respectively.

The sole difference between this manipulator and a structurally identical, fully actuated one is that the elements of τ_p, corresponding to the passive joints' torques, cannot be directly controlled. In fact, these elements can assume only the values 0 or $\tau_{l,i}$, where $\tau_{l\,i}$ is the i-th passive joint's nominal brake torque, assumed to be exactly the torque necessary to keep passive joint i locked despite the motion of all other joints. To further distinguish among the passive joints that are locked or unlocked at a certain instant, we partition Equation **(3.3)** as:

$$\begin{bmatrix} \tau_a \\ \tau_u \\ \tau_l \end{bmatrix} = \begin{bmatrix} M_{aa} & M_{au} & M_{al} \\ M_{ua} & M_{uu} & M_{ul} \\ M_{la} & M_{lu} & M_{ll} \end{bmatrix} \begin{bmatrix} \ddot{q}_a \\ \ddot{q}_u \\ \ddot{q}_l \end{bmatrix} + \begin{bmatrix} b_a \\ b_u \\ b_l \end{bmatrix} \tag{3.4}$$

where $\tau_u = 0$ because no torque is applied at the unlocked passive joints, and $\ddot{q}_l = 0$ because the locked passive joints do not move when their brakes are engaged. In practice, the torque at the locked joints τ_l counteracts the inertial and noninertial torques on the right-hand side of the third line of the dynamic equation **(3.4).** Therefore, that line can be eliminated from the equation, and we will not carry it over in subsequent paragraphs.

For reasons that will become obvious in Chapter 4. the dimension of the vector of unlocked passive joints is limited to the number of actuators, i.e., $[q_u] = n_u \le n_a$. Briefly, the reason is that one can control at most only n_a joints at a time; therefore, no more than n_a passive joints should be unlocked for control purposes at any given instant.

An important equation to be considered is the second line of **(3.4):**

$$M_{ua}\ddot{q}_a + M_{uu}\ddot{q}_u + b_u = 0 \tag{3.5}$$

This second-order differential equation represents the constraints imposed on the system by the unlocked passive joints. It is a second-order nonholonomic constraint which, in general, cannot be integrated to yield a relationship between the velocities or positions of the active and passive joints [34].

Recall that, at most, n_a joints can be controlled at any given instant. These joints can be either the active ones, a subset of n_a passive ones (when $n_p \geq n_a$), or a combination of some active and some passive joints. For control purposes, it is helpful to rewrite the dynamic equation (3.4) in a form where one can easily distinguish which joints are being controlled from the remaining ones.

We will refer to the joints being controlled at a given instant as the *controlled joints* and group them in the $n_a \times 1$ vector q_c. The positions of the joints in q_c are controlled at the expense of not controlling directly the positions of all other joints. We group the remaining joints in the vector q_r . We *do not include* the locked passive joints q_l in the vector q_r because these do not contribute to the motion of the system. Therefore, the dimension of q_r is, at most, equal to n_p. The dynamic equation of the underactuated manipulator can be rewritten in terms of q_c and q_r as:

$$\begin{bmatrix} \tau_a \\ 0 \end{bmatrix} = \begin{bmatrix} M_{ac} & M_{ar} \\ M_{uc} & M_{ur} \end{bmatrix} \begin{bmatrix} \ddot{q}_c \\ \ddot{q}_r \end{bmatrix} + \begin{bmatrix} b_a \\ b_u \end{bmatrix} \tag{3.6}$$

and the nonholonomic constraints (3.5) as:

$$M_{uc}\ddot{q}_c + M_{ur}\ddot{q}_r + b_u = 0 \tag{3.7}$$

3.2 Properties of the dynamic equations

The dynamic equations of an underactuated manipulator possess important properties which are summarized in this section.

Property 1: The inertia matrix M is symmetric and positive-definite.

Proof: The inertia matrix, M, is obtained from that of a fully actuated manipulator, M_0, by the rearrangement of its rows and columns. Initially, suppose that only one row/column exchange is performed, and that these are row/column i and j. Mathematically, this corresponds to:

$$M = T_{i,j} M_0 T_{i,j}^{-1} \tag{3.8}$$

where $T_{i,j}$ is the following $n \times n$ elementary matrix (the elements not shown are zero):

$$T_{i,j} = \begin{array}{cc}
\left[\begin{array}{cccccccc} 1&&&&&&&\\ &1&&&&&&\\ &&0&&&1&&\\ &&&1&&&&\\ &&&&1&&&\\ &&1&&&0&&\\ &&&&&&1&\\ &&&&&&&1 \end{array}\right] & \begin{array}{l} \\ \\ \longleftarrow \text{row } i\\ \\ \\ \longleftarrow \text{row } j\\ \\ \\ \end{array} \\
\begin{array}{cccccccc} &&\uparrow&&&\uparrow&&\\ &&\text{column } i&&&\text{column } j&& \end{array} &
\end{array} \tag{3.9}$$

The matrix $T_{i,j}$ is elementary, symmetric, and positive definite; consequently, M is similar to M_0. Now it is a well-known fact that the inertia matrix of a fully actuated manipulator is positive definite [22]; consequently, M is also positive definite. This proves that positive definiteness is invariant with respect to an exchange of rows and columns i and j. Additionally,

$$M^T = (T_{i,j} M_0 T_{i,j}^{-1})^T = M \tag{3.10}$$

which shows that symmetry is also preserved when one row/column exchange is performed. Naturally, if another exchange of, say, rows and columns k and l is performed, the same proofs can be used to guarantee that the resulting matrix is still positive definite and symmetric. In conclusion, no matter how many row/column exchanges are performed in M_0 for obtaining M, the latter will be always positive definite and symmetric. ∎

Property 2: The $n \times n$ matrix $N(q, \dot{q}) = \dot{M}(q) - 2C(q, \dot{q})$ is skew-symmetric.

Proof: The matrix C is defined as [22]:

$$C(k, j) = \sum_{i=1}^{n} \frac{1}{2}\left(\frac{\partial M(k, j)}{\partial q(i)} + \frac{\partial M(k, i)}{\partial q(j)} - \frac{\partial M(i, j)}{\partial q(k)}\right)\dot{q}(i) \tag{3.11}$$

Each element of the derivative of the inertia matrix is given by:

$$\dot{M}(k, j) = \sum_{i=1}^{n} \frac{\partial M(k, j)}{\partial q(i)}\dot{q}(i) \tag{3.12}$$

The elements of N can be computed using (3.11) and (3.12):

$$\begin{aligned} N(k, j) &= \dot{M}(k, j) - 2C(k, j) = \\ &= \sum_{i=1}^{n} \left\{\frac{\partial M(k, j)}{\partial q(i)} - \left[\frac{\partial M(k, j)}{\partial q(i)} + \frac{\partial M(k, i)}{\partial q(j)} - \frac{\partial M(i, j)}{\partial q(k)}\right]\right\}\dot{q}(i) = \\ &= \sum_{i=1}^{n} \left(\frac{\partial M(i, j)}{\partial q(k)} - \frac{\partial M(k, i)}{\partial q(j)}\right)\dot{q}(i) \end{aligned} \tag{3.13}$$

Recalling that the matrix M is symmetric, it is straightforward to deduce that:

$$N(k, j) = -N(j, k) \tag{3.14}$$

which proofs that N is skew-symmetric. Mathematically, this is equivalent to:

$$x^T N x = 0, \forall x \in \Re^n \tag{3.15}$$

Property 3: The time derivative of the manipulator's kinetic energy K does not depend on the centrifugal and Coriolis torques:

$$\dot{K} = 4_{,}^{T}[\tau_a - G_a] - \dot{q}_u^T G_u \tag{3.16}$$

Proof: The kinetic energy's time derivative can be computed as:

$$\begin{aligned}
\dot{K} &= \frac{d}{dt}\left(\frac{1}{2}\dot{q}^T M \dot{q}\right) = \\
&= \boldsymbol{q}^T M \ddot{q} + \frac{1}{2}\dot{q}^T \dot{M} \dot{q} = \\
&= \dot{q}^T\left[\tau - G - C\dot{q} + \frac{1}{2}\dot{M}\dot{q}\right] = \\
&= \dot{q}^T\left[\tau - G + \frac{1}{2}(\dot{M} - 2C)\dot{q}\right] = \\
&= \dot{q}^T[\tau - G] + \frac{1}{2}\dot{q}^T N \dot{q} \\
&= \dot{q}^T[\tau - G]
\end{aligned} \tag{3.17}$$

The torque vector τ and the joint vector q can be decomposed into three parts, corresponding to the active, unlocked passive, and locked passive joints. The time clerivative of the kinetic energy can then be expressed as:

$$\begin{aligned}
\mathrm{k} &= \dot{q}^T[\tau - G] = \\
&= \begin{bmatrix} \dot{q}_a^T & \dot{q}_u^T & \dot{q}_l^T \end{bmatrix} \begin{bmatrix} \tau_a - G_a \\ \tau_u - G_u \\ \tau_l - G_l \end{bmatrix} = \\
&= \dot{q}_a^T[\tau_a - G_a] - \dot{q}_u^T G_u
\end{aligned} \tag{3.18}$$

where the facts $\dot{q}_l = 0$ and $\tau_u = 0$ were used. ■

Property *1* allows one to develop model-based controllers via feedback linearization, as will be shown in Chapter **4.** Property 2 clarifies the relationship between the inertial torques and the Coriolis and centrifugal ones, stating that the latter are dependent on the elements of the inertia matrix. Property 3 states that the fictitious forces $N(q, \dot{q})\dot{q}$ do no work, and that the only power inputs to the underactuated manipulator are the torques generated at the active joints and the gravitational forces acting on the active and unlocked passive joints.

3.3 Dynamic coupling

The positions of an underactuated manipulator's passive joints cannot be directly controlled because these joints are not equipped with actuators. The passive joints, however, are dynamically coupled to the active joints because of the presence of non-zero off-diagonal elements in the inertia matrix. To be able to efficiently utilize underactuated manipulators for manipulation operations such as object placement or inspection, one must quantify the dynamic coupling to measure and to control the motion of the passive joints. In this section, we present an approach to determine the acceleration of the unlocked passive joints due to the accelerations or torques of the active ones.

3.3.1 Acceleration coupling index

Recall Equation (3.5):

$$M_{ua}\ddot{q}_a + M_{uu}\ddot{q}_u + b_u = 0 \tag{3.19}$$

Because M is positive definite, so is M_{uu}. We can then write:

$$\ddot{q}_u = -M_{uu}^{-1}M_{ua}\ddot{q}_a - M_{uu}^{-1}b_u \tag{3.20}$$

The second term on the righthand side of (3.20) is a function only of q and $\dot{q}$, and as such is completely determined once measurements of these variables are available. We focus on the acceleration relationship between the active and the passive joints, and rewrite equation (3.20) as:

$$\bar{\ddot{q}}_u = -M_{uu}^{-1}M_{ua}\ddot{q}_a \equiv G_{ua}\ddot{q}_a \tag{3.21}$$

where

$$\ddot{\bar{q}}_u = \ddot{q}_u + M_{uu}^{-1} b_u \tag{3.22}$$

The acceleration $\ddot{\bar{q}}_u$ can be viewed as a virtual acceleration of the unlocked passive joints, generated by the acceleration of the active ones, and by the nonlinear torques due to velocity effects.

Equation (3.21) is important in the understanding of how an underactuated manipulator works. Torques can be applied only at the active joints. These torques produce the accelerations $\ddot{q}_a$, which indirectly produce the accelerations $\ddot{q}_u$ at the unlocked passive joints. The unlocked passive joints' accelerations can be controlled only if tlie $n_u \times n_a$ *acceleration gain matrix* G_{ua} possesses a structure that allows the active joints' accelerations to be transmitted reasonably "well" (in a sense to be defined later) to tlie passive joints. Thus, the study of this matrix is of fundamental importance for the design and control of underactuated manipulators.

Jain and Rodriguez [17] called the acceleration gain matrix the *disturbance Jacobian*. In their work, they did not consider utilizing the disturbance Jacobian for analysis and design of underactuated manipulators. Their intent was simply to quantify the motion of the passive joints via (3.21) when the active ones are being controlled.

Example 3.1: Consider a 2-link underactuated manipulator as shown in Figure 3.1. Joint 1 rotates around the Z axis, while joint 2 rotates around an axis perpendicular to the first joint's axis. The inertia matrix M for this nianipulator is:

$$M = \begin{bmatrix} m_2 l_{c_2}^2 \sin(q_2)^2 + I_1 + I_2 & 0 \\ 0 & m_2 l_{c_2}^2 + I_2 \end{bmatrix} \tag{3.23}$$

Assume that joint 1 is active and joint 2 is passive; then $M_{ua} = 0$, $M_{uu} = m_2 l_{c_2}^2 + I_2$, and $G_{ua} = 0$. This indicates that it is not possible to control $\ddot{q}_2$ via its coupling with $\ddot{q}_1$. Thus, this underactuated system would not be useful for practical purposes. Were joint 1 passive and joint 2 active, the result would be the same, as M_{ua} would still be zero. We conclude that this manipulator's structure decouples the accelerations of the active and passive joints, independently of the location of the actuator and brake. This *does not imply* that the joints do not have any coupling at all. In fact, the second term in the right-hand side of (3.20) is generally non-zero, and the passive joint will be driven by those torques.

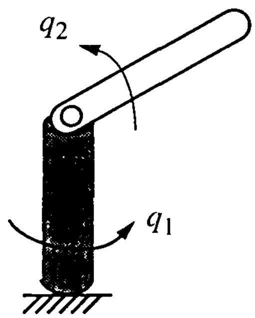


Figure 3.1: Two-link underactuated manipulator; joint 1 is active and joint 2 is passive.

Consider now the 2-link planar manipulator shown in Figure 3.2. Its inertia matrix is given by:

$$M = \begin{bmatrix} m_1 l_{c_1}^2 + m_2[l_1^2 + l_{c_2}^2 + 2l_1 l_{c_2} c_2] + I_1 + I_2 & m_2[l_{c_2}^2 + l_1 l_{c_2} c_2] + I_2 \\ m_2[l_{c_2}^2 + l_1 l_{c_2} c_2] + I_2 & m_2 l_{c_2}^2 + I_2 \end{bmatrix} \quad (3.24)$$

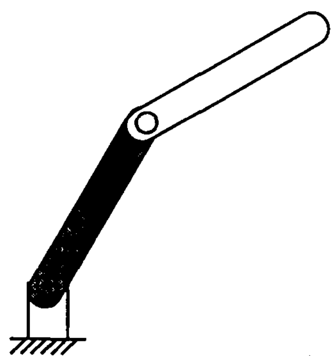


Figure 3.2: Two-link planar underactuated manipulator;
joint 1 is active and joint 2 is passive.

When joint 1 is active and joint *2* is passive, we have:

$$M_{ua} = m_2[l_{c_2}^2 + l_1 l_{c_2} c_2] + I_2, \qquad M_{uu} = m_2 l_{c_2}^2 + I_2 \tag{3.25}$$

$$G_{ua} = -\left[1 + \frac{m_2 l_1 l_{c_2} c_2}{m_2 l_{c_2}^2 + I_2}\right] \tag{3.26}$$

This manipulator's structure does not decouple the accelerations of the active and passive joints as in the previous case (except at the configurations where the term inside the brackets is zero). ∎

To quantify the coupling between the active and the unlocked passive joints, it is natural to think of the singular values of G_{ua}, which quantify the degree to which it can be inverted [21], [51], and thus its capacity to "transmit" the accelerations from the active to the passive joints. We define the *acceleration coupling* ***index*** of the underactuated manipulator as:

$$\rho_\alpha(q) = \prod_{i=1}^{n_u} \sigma_i(G_{ua}) \tag{3.27}$$

The acceleration coupling index provides a *local* measure of how well active joint accelerations are transmitted to the unlocked passive joints, because the elements of G_{ua} are functions of the manipulator's current position q. When the manipulator has both rotary and prismatic joints, one should first normalize the elements of the gain matrix G_{ua} in order to compute a coupling index with physically meaningful units.

The acceleration coupling index can be defined between a subset of the active and the unlocked passive joints. In this case, one must keep only the rows of G_{ua} corresponding to the active joints in a given subset **of** I_a, and the columns corresponding to the passive joints in a given subset of I_u. The new singular values must be recomputed and the coupling index can be computed as in (3.27), with the upper limit in the product replaced by the appropriate number. For example, the acceleration coupling index between active joint a_i and passive joint p_j is equal *to* the absolute value of $G_{ua}(i,j)$.

The acceleration coupling index can also be defined over the entire (or part of the) manipulator's workspace [11]:

$$\rho_\alpha^g = \frac{\int_\Theta \rho_\alpha^2 d\Theta}{\int_\Theta d\Theta} \tag{3.25}$$

In (3.28) the squared value of ρ_α is utilized to facilitate the analytical computation of the global coupling index, because the singular values of G,,, are equal to the square root of the eigenvalues of $G_{ua}^T G_{ua}$. The global acceleration coupling index provides a measure of how well the active joint accelerations are transmitted to the unlocked passive joints *on the average* over the joint space region **O.**

The acceleration coupling index is a useful mathematical tool for the analysis and design of underactuated manipulators, as shown in the next few examples.

Example 3.2: Consider a 3-link planar nianipulator with rotary joints, obtained by adding one link to the 2-link manipulator in Figure 3.2. Its inertia matrix is given by:

$$M = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} \tag{3.29}$$

where:

$$
\begin{aligned}
M_{11} &= m_1 l_{c_1}^2 + m_2(l_1^2 + l_{c_2}^2 + 2l_1 l_{c_2} c_2) + m_3(l_1^2 + l_2^2 + l_{c_3}^2 + 2l_1 l_2 c_2 + 2l_1 l_{c_3} c_{23} + 2l_2 l_{c_3} c_3) + I_1 + I_2 + I_3 \\
M_{12} &= M_{21} = m_2(l_{c_2}^2 + l_1 l_{c_2} c_2) + m_3(l_2^2 + l_{c_3}^2 + l_1 l_2 c_2 + l_1 l_{c_3} c_{23} + 2l_2 l_{c_3} c_3) + I_2 + I_3 \\
M_{13} &= M_{31} = m_3(l_{c_3}^2 + l_1 l_{c_3} c_{23} + l_2 l_{c_3} c_3) + I_3 \\
M_{22} &= m_2 l_{c_2}^2 + m_3(l_2^2 + l_{c_3}^2 + 2l_2 l_{c_3} c_3) + I_2 + I_3 \\
M_{23} &= M_{32} = m_3(l_{c_3}^2 + l_2 l_{c_3} c_3) + I_3 \\
M_{33} &= m_3 l_{c_3}^2 + I_3
\end{aligned}
\tag{3.30}
$$

The kinematic and dynamic parameters adopted for the manipulator's links are given in Table 3.1.

Table 3.1: Kinematic and dynamic parameters of a 3-link planar underactuated manipulator.

link	m_i (Kg)	I_i (Kg m^2)	l_i (m)	l_{ci} (m)
1	1.0	0.1	0.3	0.15
2	1.0	0.1	0.3	0.15
3	1.0	0.1	0.3	0.15

Assume that $n_a = 2$, i.e., there are two actuators to be placed either on joints 1 and 2, 1 and 3, or 2 and 3. Figure 3.3 shows the value of $\rho_{\alpha, i}$, $i = 1, 2, 3$ (with the indices 1, 2, and 3 used to differentiate each case) as a function of both q_2 and q_3. **A** careful consideration of these figures shows that, for most values of the joint angles, $\rho_{\alpha\ 1}$ is the greatest index of all three. This can be verified by the values in Table 3.2. In none of the cases does $\rho_{\alpha, i}$ become zero. This indicates that, for this manipulator, acceleration is always transmitted directly from the active to the passive joints everywhere inside the manipulator's workspace. Additionally, the choice of joint 3 as the passive joint increases the average acceleration coupling index and, therefore, the acceleration transmission, important for control purposes.

Table 3.2: Maximum, minimum, average and standard deviation values attained by the acceleration coupling index.

i	I_a	I_p	$\max(\rho_{\alpha,i})$	$\min(\rho_{\alpha,i})$	$\mathrm{avg}(\rho_{\alpha,i})$	$\mathrm{std}(\rho_{\alpha,i})$
1	{1,2}	{3}	2.2088	0.6860	1.4137	0.4060
2	{1,3}	{2}	1.4776	0.6807	1.0581	0.2740
3	{2,3}	{1}	0.7454	0.4619	0.5682	0.0712

The decision to place the two available actuators at joints 1 and 2 can be more easily reached with the computation of the global coupling index. In the following calculations, we considered that the joints can rotate freely around their respective axis from $-\pi$ to π rd. If there are physical joint limits, these can be taken into account in the calculation of the integrals in (3.28). For cases 1, 2, and 3 we have the results shown in Table 3.3. We conclude that case 1 is the one which provides greater dynamic coupling between the active and the passive joints in a global sense. ■

Table 3.3: Global acceleration coupling index in Example 3.2.

i	I_a	I_p	$\rho^g_{\alpha,i}$
1	{1,2}	{3}	2.2106
2	{1,3}	{2}	1.2200
3	{2,3}	{1}	0.3300

Example 3.3: The coupling index is useful for the purpose of designing the links of an underactuated mechanism. The 2-link planar manipulator of Figure 3.2, with joint 1 active and joint 2 passive, has the following parameters: $m_2 = 1$ Kg, $I_2 = 0.1$ Kg m^2, $l_1 = 1$ m. Suppose we want to determine l_{c_2} to maximize the global coupling index ρ^g_α. The global coupling index varies with respect to the value of l_{c_2} according to:

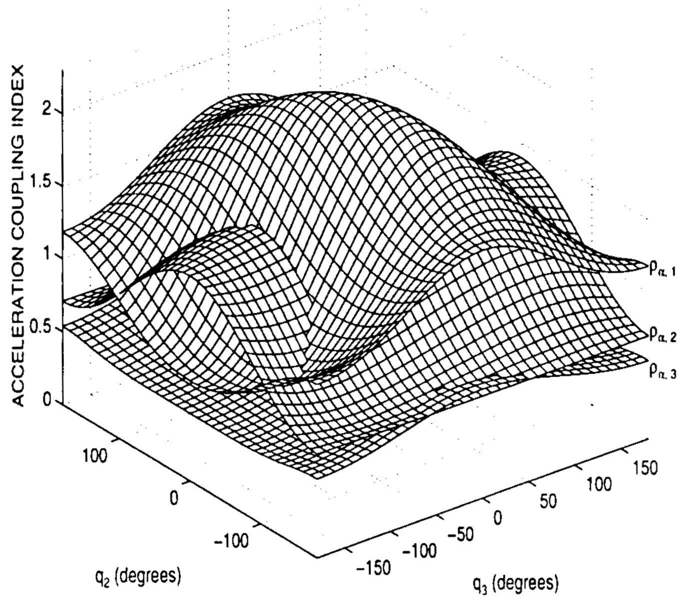


Figure 3.3: Acceleration coupling index for a 3-link planar underactuated manipulator with two actuators for the three possible locations of the actuators.

$$\rho_\alpha^g = \frac{1 + 70l_{c_2}^2 + 100l_{c_2}^4}{1 + 20l_{c_2}^2 + 100l_{c_2}^4} \tag{3.31}$$

Figure 3.4 presents the global coupling index as a function of l_{c_2}. The index attains the maximum value $\rho_\alpha^g = 2.250$ when $l_{c_2} = 0.316\text{m}$. This example shows how the global coupling index can be used for design issues other than actuator placement. ■

3.3.2 Torque to acceleration coupling

In some cases, one may be interested in the relationship between passive joints accelerations and physical variables other than active joints accelerations. For example, energy minimization requires study of the relationship to the torques at the actuators. Since acceleration in a complex nonlinear mechanism such as a manipulator is not necessarily proportional to torque (and hence to energy consumption), we cannot claim that a large

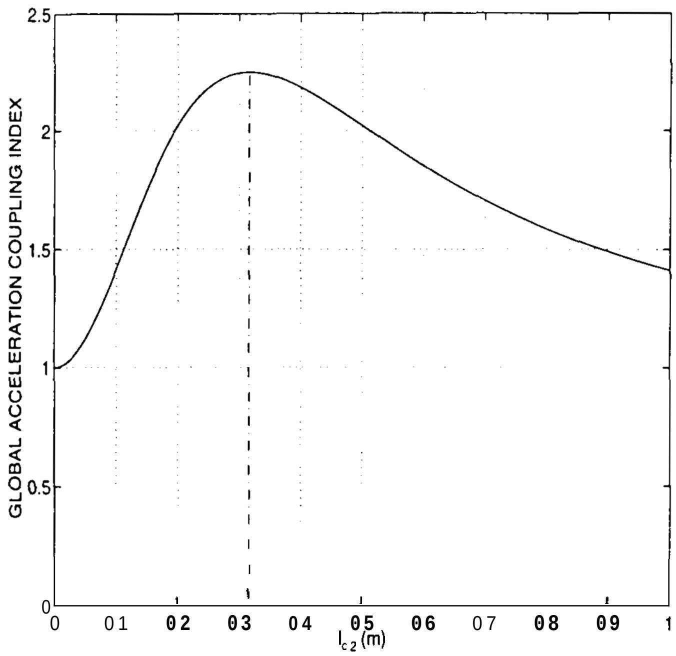


Figure 3.4: Global acceleration coupling index as a function of the location of the center of mass of a 2-link underactuated manipulator.

acceleration coupling index as defined previously implies low energy spending. To support this claim in subsequent chapters, we define an alternative coupling index.

Factoring $\ddot{q}_a$ in the first line of **(3.4)** we obtain:

$$\ddot{q}_a = M_{aa}^{-1}(\tau_a - M_{au}\ddot{q}_u - b_a) \tag{3.32}$$

Substituting this expression on the second line of **(3.4),** the following relationship between the acceleration of the passive joints and the torques applied at the active ones is obtained:

$$\ddot{q}_u = -W_{uu}M_{ua}M_{aa}^{-1}\tau_a + W_{uu}(M_{ua}M_{aa}^{-1}b_a - b_u) \tag{3.33}$$

where the $n_p \times n_p$ matrix W_{uu} is the inverse of the Schur complement of M_{aa} in M [19]:

$$W_{uu} = (M_{uu} - M_{ua}M_{aa}^{-1}M_{au})^{-1} \tag{3.34}$$

The matrix W_{uu} is positive definite, because it is equal to the lower diagonal block of the inverse of the inertia matrix.

We focus on the relationship between the accelerations of the passive joints and the active joints' torques, and rewrite equation (3.33) as:

$$\ddot{\bar{q}}_u = -W_{uu}M_{ua}M_{aa}^{-1}\tau_a \equiv W_{ua}\tau_a \tag{3.35}$$

where

$$\ddot{\bar{q}}_u = \ddot{q}_u - W_{uu}(M_{ua}M_{aa}^{-1}b_a - b_u) \tag{3.36}$$

As before, the vector $\ddot{\bar{q}}_u$ can be considered as a virtual acceleration of the unlocked passive joints, generated by the active torques and the nonlinear torques due to velocity and gravitational effects. We define the following *torque to acceleration coupling index* (henceforth referred to as torque coupling index for simplicity's sake):

$$\rho_\tau = \prod_{i=1}^{n_u} \sigma_i(W_{ua}) \tag{3.37}$$

Here, too, the elements of the matrix W_{ua} must be normalized if the manipulator has both rotary and prismatic joints.

The torque coupling index can also be computed for a subset of the active and unlocked passive joints, and over part or the entire workspace of the manipulator:

$$\rho_\tau^g = \frac{\int_\Theta \rho_\tau^2 d\Theta}{\int_\Theta d\Theta} \tag{3.38}$$

The torque coupling index will be utilized in Chapter 4 as an optimization index for the optimal control sequence of underactuated manipulators equipped with more passive joints than actuators.

3.4 Cartesian space modeling

The n_a generalized coordinates that can be controlled with an n_a-actuator underactuated manipulator do not need to be a subset of the joint vector. In this section, we cast the dynamic model in Cartesian space form. The Cartesian variables (position and orientation) are divided, as before, into two subsets, namely, the *active Cartesian variables* and the *passive Cartesian variables*. In the next chapter, a robust controller will be proposed for the control of the active Cartesian variables.

To obtain the dynamic model of an underactuated manipulator in Cartesian space, we start with the joint space model and use kinematic relationships to map joint space quantities into Cartesian space quantities. From (3.4) we can write:

$$\begin{bmatrix} \tau_a \\ 0 \end{bmatrix} = \begin{bmatrix} M_a \\ M_u \end{bmatrix} \ddot{q} + \begin{bmatrix} b_a \\ b_u \end{bmatrix} \tag{3.39}$$

Since control is to be performed in Cartesian space, and the torques commanding the manipulator are applied at the joints, a forward kinematic analysis must be performed in order to map the motion in joint space to that in Cartesian space. This mapping can be described by the position-dependent Jacobian matrix, that transforms velocities in joint space, $\dot{q}$, into velocities in Cartesian space, x :

$$X = J(q)\dot{q} \tag{3.40}$$

The Cartesian space variables are represented by the $n \times 1$ vector X. Computing the time derivative of (3.40), an expression for the acceleration of the joint angles is obtained as:

$$\ddot{q} = K\ddot{x} - K\dot{J}\dot{q} \tag{3.41}$$

where $K(q)$ represents the pseudo-inverse of the Jacobian matrix.

Since the underactuated manipulator has only n_a actuators, it is possible to control the positions of at most n_a Cartesian variables at a time. These variables will be called the *active Cartesian variables,* and represented by x_a. The other n_p ones cannot be controlled, and will be called the *passive Cartesian variables,* represented by x_p. The vector x can be partitioned as $x = [x_a^T \;\; x_p^T]^T$. Similarly, the matrix K can be partitioned as $K = [K_a \;\; K_p]$. Equation (3.41) can be rewritten as:

$$\ddot{q} = K_a\ddot{x}_a + K_p\ddot{x}_p - K\dot{J}\dot{q} \tag{3.42}$$

Substitution of (3.42) into (3.39) provides the relationship between Cartesian accelerations and joint torques:

$$\begin{bmatrix} \tau_a \\ 0 \end{bmatrix} = \begin{bmatrix} M_aK_a & M_aK_p \\ M_uK_a & M_uK_p \end{bmatrix} \begin{bmatrix} \ddot{x}_a \\ \ddot{x}_p \end{bmatrix} - \begin{bmatrix} M_a \\ M_u \end{bmatrix} K\dot{J}\dot{q} + \begin{bmatrix} b_a \\ b_u \end{bmatrix} \tag{3.43}$$

Equation (3.43) is the basis for the derivation of the *actuability index*, which measures the transmission between the active torques and the accelerations of the active Cartesian variables [21]. The actuability index is similar in nature to the coupling index, and can be used for the analysis and design of underactuated manipulators. We refer the reader to [21] for further details.

Chapter 4

Control of Underactuated Manipulators

In the previous chapter, we investigated the dynamic modeling of underactuated manipulators. In this chapter, we utilize the dynamic model developed to propose a robust model-based controller for underactuated manipulators. We start by discussing the concepts of feedback linearization and controllability, and then present a controller able to control all joints of the manipulator to an equilibrium point while operating in joint space. When the number of passive joints is greater than the number of active ones, a high-level optimization method is invoked to determine the optimal control sequence of the passive joints. We then present a controller able to control the active Cartesian variables to an equilibrium point or to follow a time-varying trajectory while operating in Cartesian space.

4.1 Joint space control

The following is the problem we study in the first part of this chapter: Given an underactuated manipulator with n_a actuators and n_p brakes, and initial and final desired configurations denoted by $q(t_0)$ and q^d respectively, drive all the joints so that after some time $T > 0, q(T) = q^d$.

This *poirzt-to-point* (PTP) control problem was first posed by Arai and Tachi [1]. The term point-to-point refers to the fact that, by controlling the positions of all joints to a desired joint vector, one is able to position the manipulator's end-effector at a desired point in Cartesian space. Arai and Tachi demonstrated that, at every instant, the positions of no more than n_a joints of an underactuated manipulator can be controlled concurrently. Their solution to the control problem above is valid for the case $n_a \geq n_p$, and consists of driving the joints of the manipulator in two distinct phases (see Figure **4.1).** First, the passive joints are driven to their set-points via their dynamic coupling with the active ones. Each passive joint is locked as it reaches its set-point. After all passive joints reach the desired positions and are locked, the second phase takes place, namely, control of the active joints to their set-points. Because the passive joints are locked during the second phase, the manipulator is as easily controlled as if it were fully actuated. Arai and Tachi addressed the controllability of the passive joints when driven by the active ones, and were able to obtain a condition sufficient for controllability around equilibrium points.

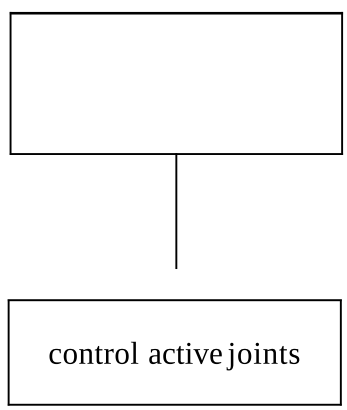


Figure 4.1: Flow diagram of the method proposed in [13 to control all joints of an underactuated manipulator with more active than passive joints.

Other researchers followed up on Arai's work and proposed similar control methods [30], [36]. These works, including Arai's, have the following limitations: (i) controllability of the passive joints over the entire state space is not addressed; (ii) the outer-loop controller utilized is a PID controller, which has been experimentally verified, on the underactuated manipulator described in Section 4.8, not to be robust against modeling errors or external disturbances; and (iii) the control methods do not handle manipulators with more passive joints than actuators.

The purpose of this chapter is to present our proposed solutions to the limitations described above. We begin with a description of the dynamic equations of the underactuated manipulator in terms of the joints that are being controlled at a given instant and their open-loop relationship to the torques applied at the actuators.

4.2 Feedback linearization

In Chapter 3, we presented the dynamic equation of the underactuated manipulator written in terms of the controlled joints q_c and the remaining joints q_r as:

$$\begin{bmatrix} \tau_a \\ 0 \end{bmatrix} = \begin{bmatrix} M_{ac} & M_{ar} \\ M_{uc} & M_{ur} \end{bmatrix} \begin{bmatrix} \ddot{q}_c \\ \ddot{q}_r \end{bmatrix} + \begin{bmatrix} b_a \\ b_u \end{bmatrix} \tag{4.1}$$

There are three possible ways of forming the $n_a \times 1$ vector q_c, each one leading to a different control strategy for the underactuated manipulator. First, q_c may contain only active joints. When this is the case, we choose to keep all other joints (i.e., the passive joints) locked in place. This choice allows us to control the active joints as if the manipulator were fully actuated. Second, q_c may contain only passive joints (which is only possible when the number of passive joints is at least equal to n_a). In this case, we choose to keep all other passive joints, if any, locked while q_c is controlled. This choice guarantees that the passive joints not being controlled do not move and do not introduce dynamic disturbances on the joints being controlled. Third, the vector q_c may contain both active and passive joints. In this case, too, we choose to keep all other passive joints, if any, locked, to guarantee that they do not move.

We will refer to the first case as control strategy ***A***, because only active joints are being controlled. In this case, $\ddot{q}_c = \ddot{q}_a$, the dimension of q_r is zero, and the dynamic equation (4.1) reduces to:

$$\tau_a = M_{aa}\ddot{q}_a + b_a \tag{4.2}$$

We will refer to the motion of the underactuated manipulator in joint space when control strategy ***A*** is executed as an ***A*** motion. During an ***A*** motion, the nonholonomic constraints (3.5) disappear because the dimension of q_u is zero.

We will refer to the second case as control strategy *P*, because only passive joints are being controlled. In this case, $\ddot{q}_c = \ddot{q}_u$, $\ddot{q}_r = \ddot{q}_a$, and the dynamic equation (4.1) can be written as:

$$\begin{bmatrix} \tau_a \\ 0 \end{bmatrix} = \begin{bmatrix} M_{au} & M_{aa} \\ M_{uu} & M_{ua} \end{bmatrix} \begin{bmatrix} \ddot{q}_u \\ \ddot{q}_a \end{bmatrix} + \begin{bmatrix} b_a \\ b_u \end{bmatrix} \tag{4.3}$$

When control strategy *P* is utilized, we say that the manipulator performs a *P* motion. During a *P* motion the nonholonomic constraints (3.5) dictate the behavior of the motion of the active joints.

Factoring out $\ddot{q}_a$ in the second line of (4.3) and substituting the result back on its first line, we obtain the open-loop relationship between $\ddot{q}_u$ and τ_a:

$$\tau_a = (M_{au} - M_{aa}M_{ua}^{-1}M_{uu})\ddot{q}_u - M_{aa}M_{ua}^{-1}b_u + b_a \tag{4.4}$$

Equation (4.4) requires the inversion of the $n_a \times n_a$ matrix M_{ua}, an issue that we will discuss in a subsequent section.

Finally, we will refer to the third case as an ***AP*** control strategy, because both active and passive joints are being controlled. The vector q_c will contain all elements from q_u and some elements from q_a, while q_r will contain the remaining elements from q_a not in q_c. An open-loop relationship similar to (4.4) can be obtained in this case as:

$$\tau_a = (M_{ac} - M_{ar} M_{ur}^{-1} M_{uc}) \ddot{q}_c - M_{ar} M_{ur}^{-1} b_u + b_a \tag{4.5}$$

Motions resulting from the execution of the ***AP*** control strategy will be referred to as ***AP*** motions. During ***AP*** motions, the nonholonomic constraints (3.5) take on the form (3.7).

With this categorization of the possible control strategies for underactuated manipulators, the sequential control method of Figure 4.1 can be described as follows: first, a *P* or ***AP*** motion is executed so that all passive joints (and perhaps some active joints, when $n_a > n_p$) are controlled to their desired set-points. Then an **A** motion is executed to bring the remaining active joints to their desired position. In other words, underactuated manipulators with more active than passive joints can be controlled to an equilibrium point in two phases, ***P-A*** or ***AP-A***, provided, of course, that the passive joints are controllable via their dynamic coupling with the active ones. (This issue will be investigated in the next section.)

All three control strategies above lead to open-loop relationships between $\ddot{q}_c$ and τ_a of the form:

$$\tau_a = \overline{M}_{ac} \ddot{q}_c + \bar{b}_a \tag{4.6}$$

Table 4.1 summarizes the expressions of $\overline{M}_{ac}$ and $\bar{b}_a$ for the three control strategies.

The resemblance of dynamic equation (4.6) with that of a fully actuated manipulator led Arai and Tachi [1] to choose a feedback linearization controller to control the joints in q_c. Feedback linearization controllers have been extensively used in the control of robot manipulators [22]. The method consists of defining an auxiliary input u, with

$$\tau_a = \overline{M}_{ac} u + \bar{b}_a \tag{4.7}$$

Table 4.1: Relationship between the acceleration of the controlled joints and the active torques for all possible control strategies.

Control stmtegy	$\overline{M}_{ac}$	$\bar{b}_a$
A	M_{aa}	b_a
P	$M_{au} - M_{aa}M_{ua}^{-1}M_{uu}$	$-M_{aa}M_{ua}^{-1}b_u + b_a$
AP	$M_{ac} - M_{ar}M_{ur}^{-1}M_{uc}$	$-M_{ar}M_{ur}^{-1}b_u + b_a$

such that, when $\overline{M}_{ac}$ is invertible,

$$\ddot{q}_c = u \tag{4.8}$$

The effect of the feedback linearization controller (4.7) is to decouple and linearize the nonlinear system (4.6). In the absence of modeling errors and external disturbances, substitution of u in (4.7) by **a** PID-like controller of the form

$$u = \ddot{q}_c^d + K_d(\dot{q}_c^d - \dot{q}_c) + K_p(q_c^d - q_c) + K_i\int_{t_0}^{t}(q_c^d - q_c)dt \tag{4.9}$$

guarantees that q_c will follow a desired trajectory $q_c^d(t)$. **A** block diagram of the feedback linearization technique is presented in Figure 4.2.

In practice, however, modeling errors and external disturbances are common. We present in the sequence a robust controller to force the controlled joints to follow $q_c^d(t)$ despite the presence of modeling errors or disturbances. (Another possible approach to cope with parameter uncertainties would be via the use of adaptive control techniques, as done by Gu and Xu [12].) Before presenting the proposed controller, we study the controllability of the nonlinear system (4.6) and obtain a sufficient condition under which q_c is controllable.

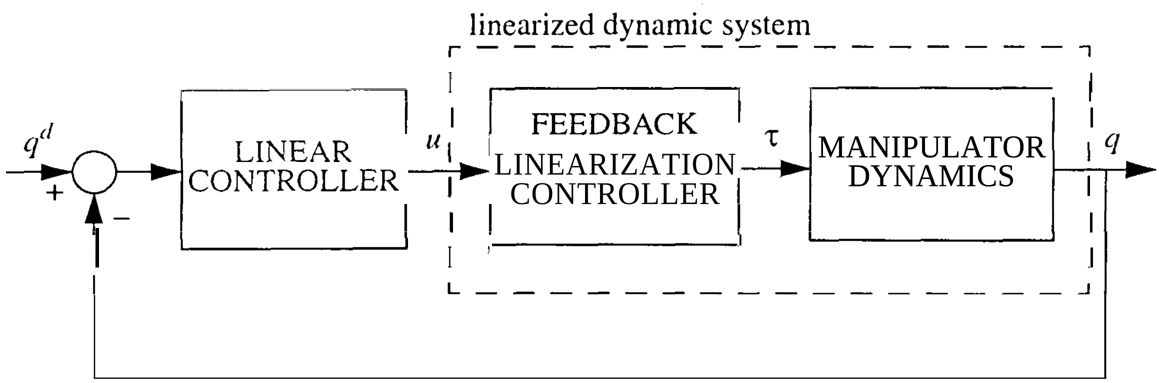


Figure 4.2: Block diagram of the feedback linearization controller plus linear controller proposed by Arai and Tachi [1].

4.3 Controllability

In this section we study the controllability of the joints in the vector q_c. By controllability we understand the possibility to transfer the joints in q_c from any initial position to any final position in finite time through the application of forces at the active joints.

Our controllability study will be divided into three parts, corresponding to the three control strategies, ***A***, *P*, and ***AP***, that we defined earlier. We assume that the dynamic model of the underactuated manipulator is known precisely, and that no external disturbances act on the system. Appendix A presents the differential geometry definitions and results that we will utilize in this section.

When control strategy ***A*** is utilized, the manipulator behaves as a fully actuated one, for which controllability has been shown in the past [22]. It rests then to consider control strategies *P* and ***AP***. We start with control strategy *P*, and our objective is to find a condition that guarantees that q_u is controllable via application of torques at the active joints. The dynamic relationship between $\ddot{q}_u$ and τ_a can be found by factoring out $\ddot{q}_a$ in the first line of (4.3) and substituting the result into its second line:

$$\ddot{q}_u = W_{ua}\tau_a - W_{ua}b_a - W_{uu}b_u \tag{4.10}$$

where

$$W_{uu} = (M_{uu} - M_{ua}M_{aa}^{-1}M_{au})^{-1} \tag{4.11}$$

and

$$W_{ua} = -W_{uu}M_{ua}M_{aa}^{-1} \tag{4.12}$$

Because the manipulator's inertia matrix is positive definite, the matrix M_{aa} is positive definite. The matrix W_{uu} too is positive definite, because it is equal to the lower diagonal block of the inverse of the inertia matrix. Consequently ([19], pp. 55):

$$\text{rank}(W_{ua}) = \text{rank}(M_{ua}) \tag{4.13}$$

To study the controllability of (4.10) we rewrite it in state-space form. Let $x_1 = q_u$, $x_2 = \dot{q}_u$ and the state x be given by

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} q_u \\ \dot{q}_u \end{bmatrix} \tag{4.14}$$

Equation (4.10) can be rewritten as:

$$\dot{x} = f(x) + g(x)\tau_a \tag{4.15}$$

where:

$$f(x) = \begin{bmatrix} x_2 \\ -W_{ua}b_a - W_{uu}b_u \end{bmatrix} = \begin{bmatrix} x_2 \\ \alpha_u(x_1, x_2, q_a, \dot{q}_a) \end{bmatrix} \tag{4.16}$$

$$g(x) = \begin{bmatrix} 0_{n_a \times n_a} \\ W_{ua}(x_1, q_a) \end{bmatrix} \tag{4.17}$$

The control vector fields g_i correspond to the columns of the matrix g:

$$g_i = \begin{bmatrix} 0_{n_a \times 1} \\ (W_{ua})_i \end{bmatrix}, \quad i = 1, \ldots, n_a \tag{4.18}$$

where $(W_{ua})_i$ represents the i-th column of W_{ua}. Each g_i is a function of x_1 but not of x_2, and the Lie bracket of any two g_i is equal to the null vector:

$$[g_i, g_j] = 0, \ 1 \le i, j \le na \tag{4.19}$$

Theorem 4.1: The nonlinear system (4.15) has vector relative degree at the state x if M_{ua} is a full rank matrix at x.

Proof: Since we are interested in controlling the joints grouped in q_u, we naturally choose the output function

$$y = h(x) = x_1 \tag{4.20}$$

Compute the following derivatives of the output function:

$$L_f^0 h = h = x_1, \quad L_{g_j} L_f^0 h = \frac{\partial x_1}{\partial x} g_j = 0, \quad 1 \le j \le n_a \tag{4.21}$$

$$L_f^1 h = \frac{\partial h}{\partial x} f = x_2, \quad L_{g_j} L_f^1 h = \frac{\partial x_2}{\partial x} g_j = (W_{ua})_j, \quad 1 \le j \le n_a \tag{4.22}$$

From Definition **A.7** in Appendix A, it follows that the system (4.15) has ***a*** vector relative degree $\boldsymbol{r} = \{2, \ldots, 2\}$. ■

Theorem 4.2: The nonlinear system (4.15) is controllable around the state $\boldsymbol{x}$ if M_{ua} is a full rank matrix at **x.**

Proof: Assume that $\mathrm{rank}(M_{ua}) = n_a$ at some state x. From Theorem 4.1, we know that the system has a vector relative degree at x, and from (4.13), it follows that $\mathrm{rank}(W_{ua}) = n_a$ at x. Consequently, from (4.17) the matrix $g(x)$ has rank n_a at x. Compute

$$\Delta_0 = \mathrm{span}\{g_i\} = \mathrm{span}\left\{\begin{bmatrix} 0 \\ (W_{ua})_1 \end{bmatrix}, \ldots, \begin{bmatrix} 0 \\ (W_{ua})_{n_a} \end{bmatrix}\right\} \tag{4.23}$$

Because W_{ua} is full rank, it follows that Δ_0 is regular at x; furthermore, (4.19) implies that Δ_0 is involutive. Compute now

$$\begin{aligned} \Delta_1 &= \mathrm{span}\{g_j, [g_i, g_j], [f, g_j], \ 1 \le i, j \le n_a\} = \\ &= \mathrm{span}\left\{\begin{bmatrix} 0 \\ (W_{ua})_1 \end{bmatrix}, \ldots, \begin{bmatrix} 0 \\ (W_{ua})_{n_a} \end{bmatrix}, \begin{bmatrix} -(W_{ua})_1 \\ A_1 \end{bmatrix}, \ldots, \begin{bmatrix} -(W_{ua})_{n_a} \\ A_{n_a} \end{bmatrix}\right\} \end{aligned} \tag{4.24}$$

where we used (4.19) to eliminate the ternis $[g_i, g_j]$ and

$$A_i = \frac{\partial (W_{ua})_i}{\partial x_1} x_2 - \frac{\partial \alpha_u}{\partial x_2} (W_{ua})_i \tag{4.25}$$

Once again, the full rank of W_{ua} implies that $\mathbf{A}_1$ is regular at **x**, and, more importantly, that the dimension of $\mathbf{A}_1$ is equal to $2n_a$. Consequently, the conditions of Theorem A.1 in Appendix **A** are fulfilled and the nonlinear system (4.15) can be linearized via an appropriate change of coordinates and feedback control law.

To find the equivalent linear version of (4.15), we choose the (trivial) change of coordinates $z = \mathbf{x}$ and the feedback

$$\tau_a = -W_{ua}^{-1}\alpha_u + W_{ua}^{-1}u \tag{4.26}$$

The resulting linear system is $z = Az + Bu$ with

$$A = \begin{bmatrix} 0_{n_a \times n_a} & I_{n_a \times n_a} \\ 0_{n_a \times n_a} & 0_{n_a \times n_a} \end{bmatrix} \tag{4.27}$$

and

$$B = \begin{bmatrix} 0_{n_a \times n_a} \\ I_{n_a \times n_a} \end{bmatrix} \tag{4.28}$$

which is controllable at x, since the rank of its controllability matrix is $2n_a$. Consequently, the nonlinear system (4.15) is also controllable at **x**. ■

We now turn our attention to the controllability of q_c under control strategy *AP*. The dynamic relationship between $\ddot{q}_c$ and τ_a can be found by factoring out $\ddot{q}_r$ in the first line of (4.1) and substituting the result into the second line:

$$\ddot{q}_c = W_{ur}\tau_a - W_{ur}b_a - W_{uc}b_u \tag{4.99}$$

where

$$W_{uc} = (M_{uc} - M_{ur}M_{ar}^{\#}M_{ac})^{\#} \tag{4.30}$$

and

$$W_{ur} = -W_{uc}M_{ur}M_{ar}^{\#} \tag{4.31}$$

We may now state the following theorem:

Theorem 4.3: The nonlinear system (4.29) is controllable around the state x if M_{ur} is a full rank matrix at **x.**

Proof: The proof is identical to the one presented for control strategy P and will be omitted for brevity. ■

Arai and Tachi [1] demonstrated that the nonlinear system (4.15) is locally controllable around equilibrium points whenever M_{ua} is a full rank matrix. It is important to note that our result extends theirs, for we showed that this is also a sufficient condition for local controllability around any point in the entire state space of the controlled joints' positions and velocities. This result is important. for example. in showing that these joints can be controlled not only to equilibrium positions but also to follow arbitrary trajectories.

4.4 Controllability and dynamic coupling

The controllability condition studied above allows one to determine whether a particular configuration of the manipulator will render the controlled joints uncontrollable. It does not provide, however, a measure of controllability at every configuration of the manipulator, a feature which is useful for control and planning purposes. In this section, we demonstrate that the controlled joints are controllable whenever the torque coupling index is non-zero.

We start with control strategy P, and recall the torque coupling index definition (3.37), with the upper limit of the summation replaced by n_a to reflect the fact that n_a passive joints are being controlled:

$$\rho_\tau(q) = \prod_{i=1}^{n_a} \sigma_i(W_{ua}) \tag{4.32}$$

The definition of the coupling index implies:

$$\text{rank}(W_{ua}) = n_a \quad \Leftrightarrow \quad \rho_\tau \neq 0 \tag{4.33}$$

Because $\text{rank}(W_{ua}) = \text{rank}(M_{ua})$, we can also write:

$$\text{rank}(M_{ua}) = n_a \quad \Leftrightarrow \quad \rho_\tau \neq 0 \tag{4.34}$$

Combining (4.34) and the result of Theorem 4.2 we obtain:

$$\rho_\tau \neq 0 \quad \Rightarrow \quad \text{system (4.15) is controllable} \tag{4.35}$$

A similar result can be obtained for control strategy ***AP***. Equation (4.35) allows one to utilize the torque coupling index as a measure of the controllability of the controlled joints at any configuration of the manipulator.

4.5 Robust joint control

In this section, we present a feedback control law that guarantees asymptotic convergence of the controlled joints to their set-points. The control law consists of an inner loop feedback linearization controller as proposed in [1], and an outer loop robust controller, as shown in Figure 4.3.

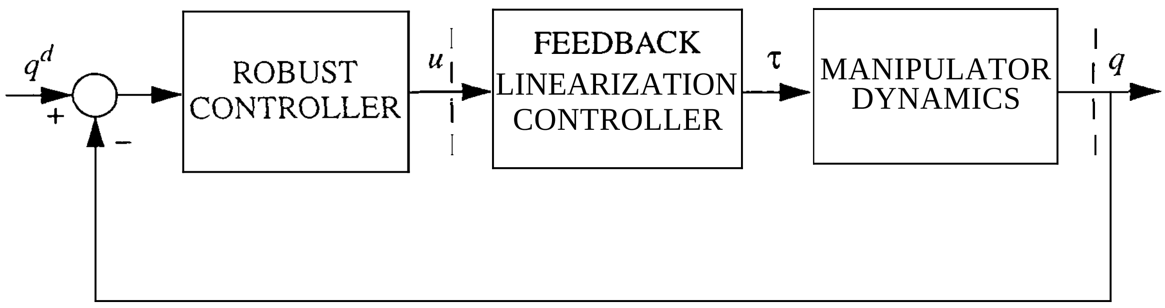


Figure 4.3: Block diagram of the feedback linearization controller plus robust controller.

The controller chosen for this work is the variable structure controller (VSC) [14]. The VSC has been successfully utilized for more than two decades for the control of nonlinear systems. Its most valuable characteristic is its robustness to model uncertainty and to external disturbances. The VSC achieves such robustness by forcing the system's state trajectory to follow a pre-defined *sliding surface* in the state space. The resulting state trajectory is known as the *sliding mode*. The state trajectory from the initial state to the sliding surface is known as the *reaching phase* (see Figure 4.4). Because the description of the sliding surface is independent of the system's kinematic and dynamic parameters, errors in these quantities do not affect the behavior of the system while in the sliding mode. It is the controller designer's responsibility to guarantee that the sliding mode begins after a finite time since the beginning of the state's motion. Let $s(x) = 0$ describe the sliding surface as a function of the system's state x. Then the sliding surface is guaranteed to be reached in finite time if [14]

$$s_i \dot{s}_i < 0 \tag{4.36}$$

for all elements of s.

Recall Equation (4.7), which defines the feedback linearization controller:

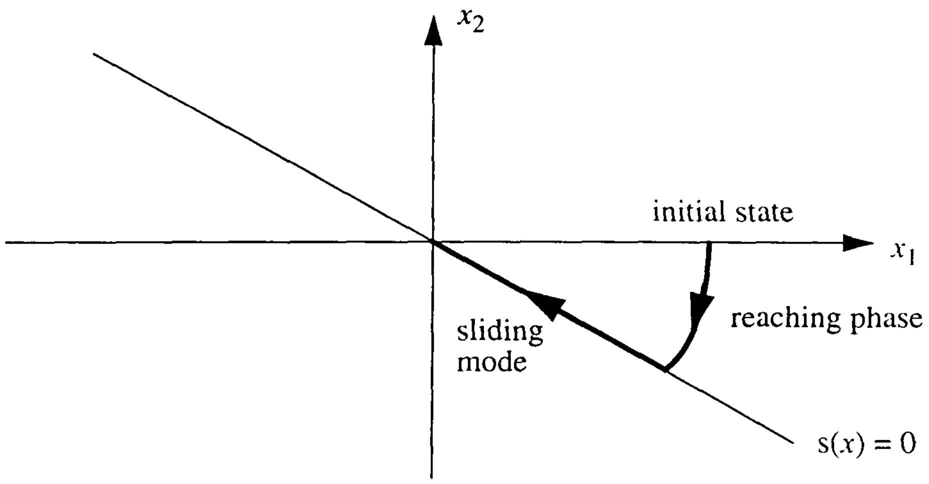


Figure 4.4: Graphical representation of the reaching phase and the sliding mode for a dynamic system whose state-space is 2-dimensional.

$$\tau_a = \overline{M}_{ac} u + \bar{b}_a \tag{4.37}$$

The matrix $\overline{M}_{ac}$ depends on which control strategy is being used, as shown in Table 4.1. In Table 4.2, we summarize the conditions under which $\overline{M}_{ac}$ is invertible for each of the control strategies ***A***, *P*, and ***AP***.

Table 4.2: Condition for existence of $\overline{M}_{ac}^{-1}$ according to the control strategy utilized.

Control strategy	$\overline{M}_{ac}$	*Existence of* $\overline{M}_{ac}^{-1}$ *guaranteed when*
A	M_{aa}	always
P	$M_{au} - M_{aa} M_{ua}^{-1} M_{uu}$	M_{ua} is invertible
AP	$M_{ac} - M_{ar} M_{ur}^{-1} M_{uc}$	M_{ur} is invertible

We assume that sufficient dynamic coupling exists between the active and the controlled joints everywhere inside the manipulator's workspace, regardless of the control strategy utilized. Consequently, controllability is guaranteed by the result of Section **4.4** and $\bar{M}_{ac}^{-1}$ is invertible. The auxiliary input u is chosen as:

$$u = \Gamma_c \dot{\tilde{q}}_c + \ddot{q}_c^d + P_c \operatorname{sgn}(s_c) \tag{4.38}$$

where $\ddot{q}_c^d$ is the desired acceleration of the controlled joints, and Γ_c and P_c are $n_a \times n_a$ diagonal gain matrices with positive elements.

Define the following sliding surface:

$$s_c = \Gamma_c \tilde{q}_c + \dot{\tilde{q}}_c \tag{4.39}$$

Using **(4.8)** and **(4.38)** one can verify that:

$$s_{c,i}\dot{s}_{c,i} = -P_{c_{i,i}}|s_{c,i}| < 0 \tag{4.40}$$

where $s_{c,i}$ is the i-th element of s_c, and $P_{c_{i,i}}$ is the i-th diagonal element of P_c. Condition **(4.40)** guarantees that the state trajectory reaches the surface $s_c = 0$ in a finite time. Once the sliding surface is reached, the dynamic behavior of $(\tilde{q}_c, \dot{\tilde{q}}_c)$ is dictated by $s_c = \Gamma_c \tilde{q}_c + \dot{\tilde{q}}_c = 0$, independently of modeling errors or dynamic disturbances. Because this represents a first-order linear system, an appropriate choice of Γ_c guarantees exponential convergence of $(\tilde{q}_c, \dot{\tilde{q}}_c)$ to (0,0) or, equivalently, of q_c to the desired q_c^d.

The use of the sign function in **(4.38)** introduces chattering in the state trajectory. To eliminate this chattering and avoid the excitation of unmodelled high-frequency dynamic components, we utilize instead the saturation function defined as

$$\operatorname{sat}(x) = \begin{cases} \operatorname{sgn}(x), & x > \varepsilon \\ x/\varepsilon, & x \le \varepsilon \end{cases} \tag{4.41}$$

where ε is the width of the so-called *boundary layer*. The value of ε must be chosen by the controller designer as a compromise between steady state accuracy and the amount of chattering acceptable.

The **VSC** guarantees convergence of all controlled joints in q_c to their set-points. (Because the sliding surface considered also takes into account the error in the controlled joints' velocities, the VSC controller allows q_c to also follow a time-varying trajectory). By exchanging the elements of q_c, or by switching between control strategies, we can control other joints as well, as long as the rank condition of Theorem 4.2 (or 4.3) is always met.

Example 4.1: Consider the 2-link underactuated manipulator shown in Figure 3.2, with joint 1 active and joint 2 passive. For its kinematic and dynamic parameters, we adopt the values shown in Table 4.3.

Table 4.3: Kinematic and dynamic parameters of a 2-link underactuated manipulator.

link	m_i (Kg)	I_i $(Kg\,m^2)$	l_i (m)	l_{ci} (m)
1	**0.994**	0.0270	0.262	0.108
2	**0.932**	0.0322	**0.464**	0.237

The objective in this example is to drive the manipulator from $q(t_0) = [0^o, 0^o]$ to $q^d = [90^o, 90^o]$. According to the theory presented in this and the preceding sections. this objective is attainable by executing control strategy P followed by strategy A. Table 4.4 presents the **VSC** controller parameters adopted during each control strategy.

Control strategy	Γ (s^{-1})	P (s^{-1})	ε (rd s^{-1})
P	10	20	0.2
A	10	20	0.2

the joint angles history, the torque applied at the joints, the configuration space trajectory (i.e., the trajectory in the plane $q_1 \times q_2$), and the phase plane trajectory. The 'X' marks on the last two graphs indicate the initial and final positions of the manipulator. The phase plane plot is, perhaps, the most interesting of all, for it clearly shows the reaching phase and the sliding mode of both joint angles, as in Figure 4.4. The joint angles graph clearly shows that joint 1 drives joint 2 to its set-point, which is reached at $t = 1.43$ s. At that instant, joint 2 is locked, and joint 1 converges quickly to its own set-point, which is reached at $t = 3.22$ s. The steady-state error is equal to [0.0013°, 0.0200°].

To verify the robustness of the VSC with respect to parameter uncertainty, we adopted two sets of kinematic and dynamic parameters, one for the controller and one to represent the manipulator. The manipulator parameters were kept at their nominal values, given in Table 4.3. The controller parameters were chosen as a multiple δ of the nominal ones. The new joint angles history and phase plane trajectory are shown in Figure 4.6, both for $\delta = 2$ and $\delta = 0.5$. One can clearly see that, despite the large parameter uncertainty, the performances are very similar to those presented in Figure 4.5. Table 4.5 summarizes the results of these three simulated motions, showing that the time instants when both joints reach their set-points and the final steady-state error are not adversely affected by parameter uncertainty.

Table 4.5: Comparison of the performance of the VSC under different parameter uncertainty conditions.

Joint	*Time to reach set-point* (s)			*Steady-state error* (°)		
	$\delta = 1$	$\delta = 2$	$\delta = 0.5$	$\delta = 1$	$\delta = 2$	$\delta = 0.5$
1	3.22	3.14	3.10	0.0013	0.0015	0.0005
2	1.43	1.32	1.59	0.0200	0.0151	0.0200

Assume now that the manipulator is placed vertically, under the influence of the gravitational acceleration of 9.8 m/s^2. Figure 4.7 presents the simulated motion of the manipulator starting from two different initial conditions, namely, from the stable equilibrium point $q(t_0) = [-90°, 0°]$, and from the unstable one, $q(t_0) = [90°, 0°]$. The set-

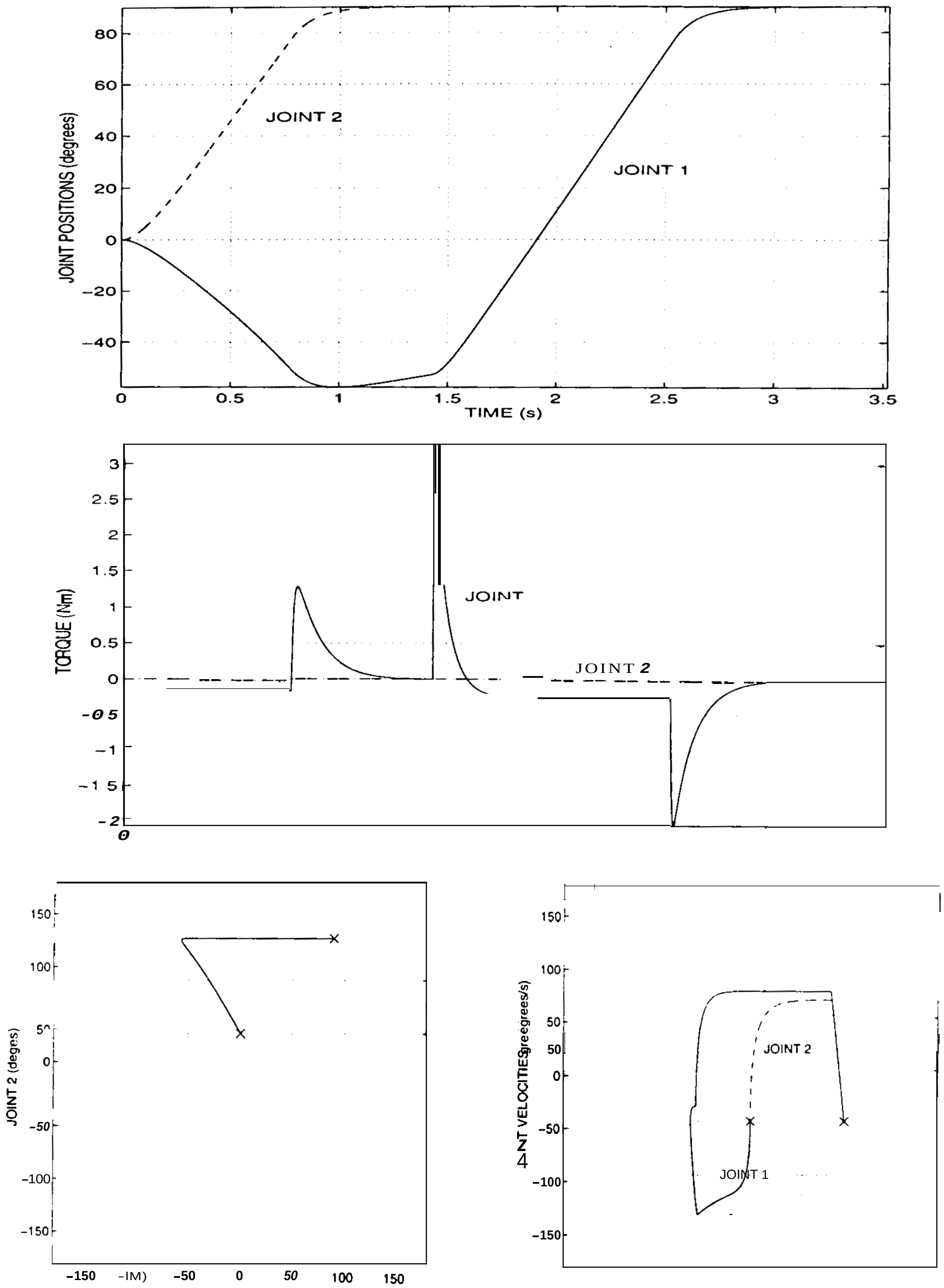


Figure 4.5: Simulated control of a 2-link underactuated manipulator. Top: joint angles history; middle: torque history; bottom left: configuration space trajectory; bottom right: phase plane trajectory.

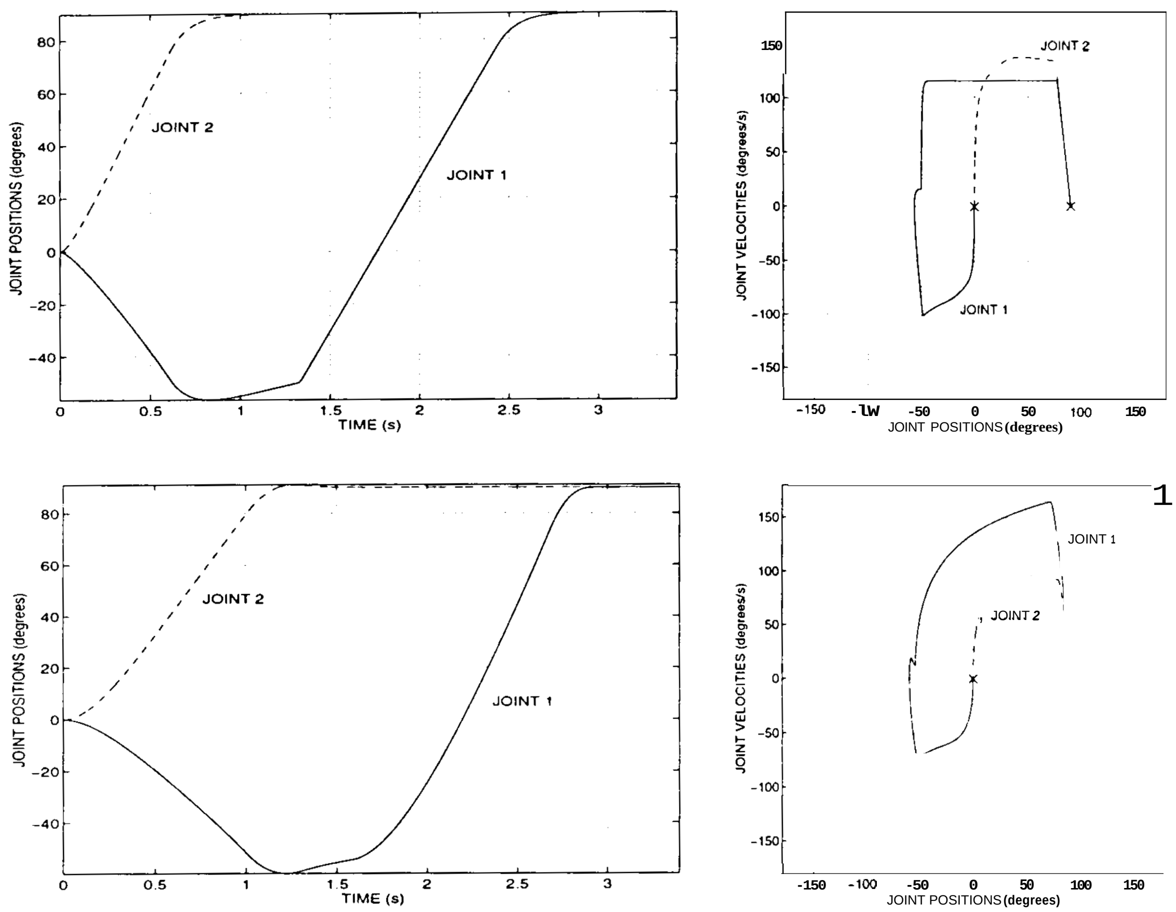


Figure 4.6: Simulated control of a 2-link underactuated manipulator with parameter uncertainty. Top: joint angles and phase plane trajectory for $6 = 2$; bottom: joint angles and phase plane trajectory for $6 = 0.5$.

point in both cases is $q^d = [90^o, 90^o]$. Table 4.6 presents the new control gains adopted, and Table 4.7 summarizes the results obtained. One can see that, even under the influence of gravity, the VSC is able to control the underactuated manipulator to a desired configuration.

■

Example 4.2: Consider now **a** 3-link planar underactuated manipulator with rotary joints, operating on a horizontal plane, with $I_a = \{1,3\}$, $I_p = \{2\}$ and parameters as in Table 4.8.

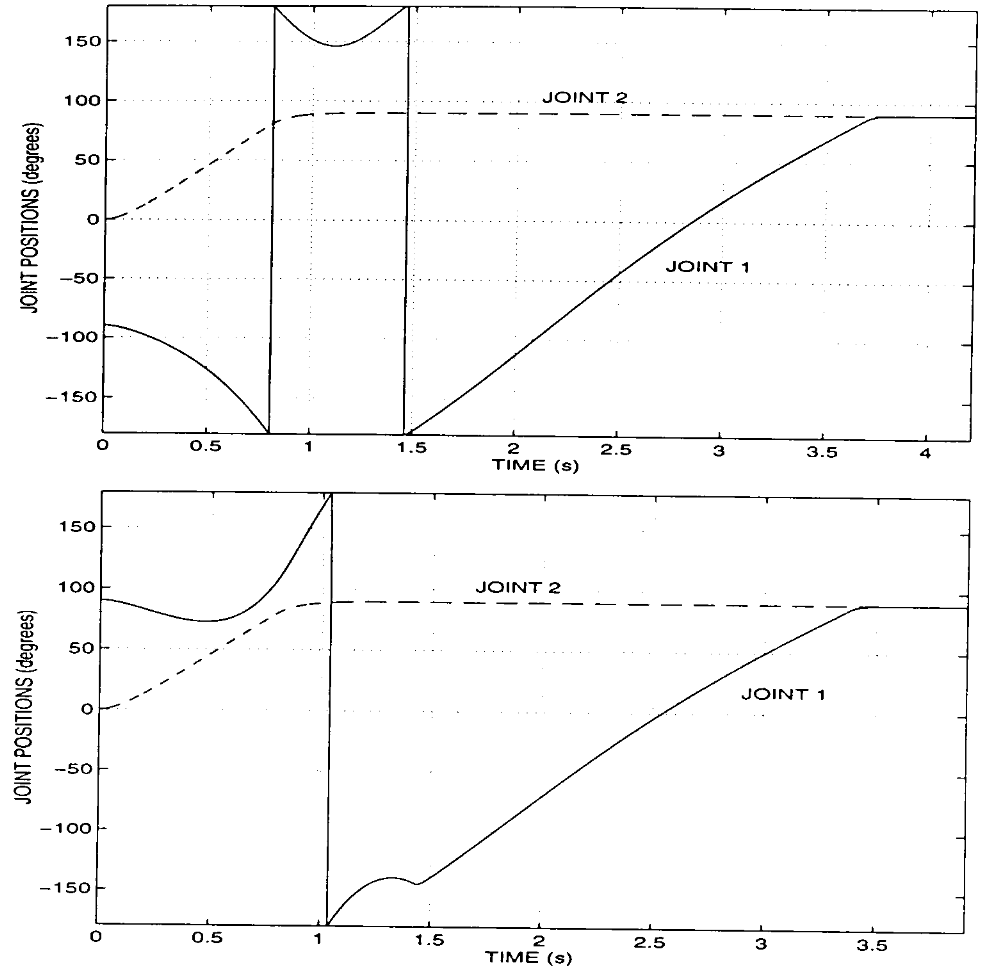


Figure 4.7: Simulated control of a 2-link underactuated manipulator on a vertical plane.
Top: joint angles for $q(t_0) = [-90^{\circ}, 0^{\circ}]$;
bottom: joint angles for $q(t_0) = [90^{\circ}, 0^{\circ}]$.

Control strategy	Γ (s^{-1})	P (s^{-1})	**E** (rd s^{-1})
P	10	20	0.2
A	30	60	0.2

Table 4.7: Performance of the VSC under influence of gravitational acceleration.

Initial configuration (°)	*Joint*	*Time to reach set-point* (s)	*Steady-state error* (°)
			[illegible]
	2	1.43	0.0200
90		3.61	0.0736
		1.43	0.0200

Table 4.5: Kinematic and dynamic parameters of a 3-link underactuated manipulator.

link	m_i (Kg)	I_i $(Kg\ m^2)$	l_i (m)	l_{ci} (m)
1	0.994	0.0270	0.262	0.108
2	0.564	0.0167	0.208	0.150
3	0.368	0.0155	0.256	0.161

strategy ***A*.** During the execution of control strategy ***AP*,** we chose to control joints 1 and 2. Table 4.9 presents the VSC controller parameters adopted during each control strategy.

Figure 4.8 presents the joint angles history, again obtained with the simulation environment, for $6 = 1$ and $6 = 2$. One can see that during the execution of control strategy *AP*, joints 1 and 2 converge to the their set-points driven by joints 1 and 3. When the passive joint (joint 2) reaches its set-point, it is locked and control strategy ***A*** starts. Table 4.10 summarizes the results, again showing that the VSC is able to maintain the manipulator's performance despite large parameter uncertainty. ■

4.6 Optimal control sequence of the passive joints

When the underactuated manipulator has more passive joints than actuators, not all passive joints' positions can be controlled concurrently to a desired value. It is possible, however, to control a subset of (at most) n_a passive joints at a time, while keeping the other ones locked. After the first subset is controlled, so can be a second, third, etc., until all passive

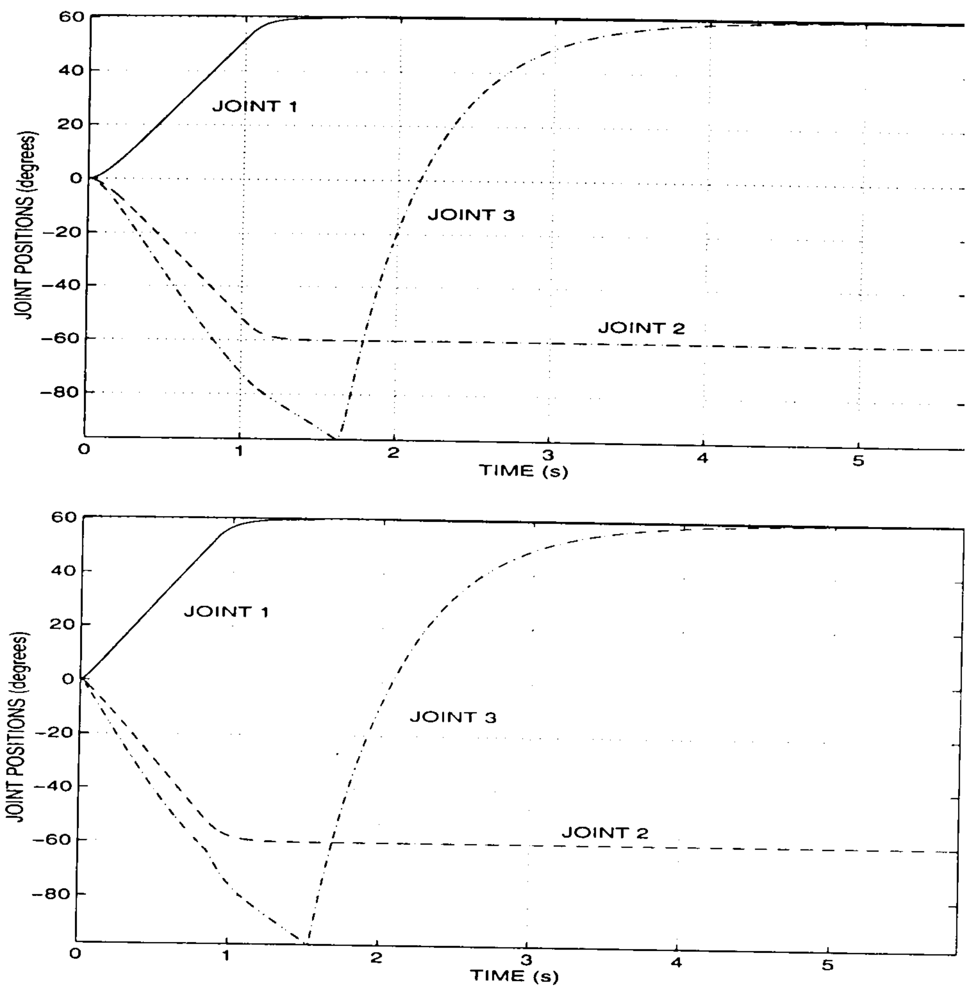


Figure 4.8: Simulated control of a 3-link underactuated manipulator. Top: joint angles for $\delta = 1$; bottom: joint angles for $\delta = 2$.

Control strategy	Γ (s^{-1})	P (s^{-1})	E (rd s^{-1})
AP	$\begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$	$\begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$	$\begin{bmatrix} 0.2 & 0 \\ 0 & 0.2 \end{bmatrix}$
A	$\begin{bmatrix} 10 & 0 \\ 0 & 2 \end{bmatrix}$	$\begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$	$\begin{bmatrix} 0.2 & 0 \\ 0 & 0.2 \end{bmatrix}$

Table 4.10: Comparison of the performance of the VSC under different parameter uncertainty conditions.

Joint	*Time to reach set-point* (s)		*Steady-state error* (°)	
	$\delta = 1$	$6 = 2$	$\delta = 1$	$6 = 2$
1	2.09	2.44	0.0001	0.0000
2	1.63	1.53	–0.0199	–0.0408
3	5.36	5.54	0.0553	0.0575

joints reach their set-points. If one then proceeds to control the active joints, then it is possible to drive all of the manipulator's joints to a desired equilibrium point. In terms of the control strategies we defined previously, this control sequence amounts to executing a series a *P* motions, followed by one (or none) *AP* motion, followed by a final *A* motion.

The niethodology just described is not fully specified until one decides which passive joints to control first, second, etc. Because it is possible to group the passive joints in many ways, a high-level controller must be devised to assign the order in which the passive joints will be controlled. This is done off-line before control is executed. We present in this section a method to optimally group the passive joints with the objective of maximizing the dynamic coupling between the active and passive joints.

We denote by p the number of groups of passive joints formed by grouping the n_I passive joints in groups of n_a joints each (note that the last group may contain fewer than n_a passive joints). The flow diagram of the proposed control method is shown in Figure 4.9. From it one can see that, in $p+1$ control phases, the manipulator reaches its final desired configuration.

Consider first the case $n_a = 1$, $n_p = n - 1$. To facilitate the comprehension of the following paragraphs, consider Figure 4.10, where a 4-link planar manipulator with one actuator is to be controlled from $q(t_0) = [0^\circ, 0^\circ, 0^\circ, 0^\circ]$ to $q^d = [90^\circ, 90^\circ, 90^\circ, 90^\circ]$. Starting from the top, that figure shows all possible ways of attaining the desired objective. For example, if one follows the left-most path, then the joints are controlled **in** this order: {4, 3,

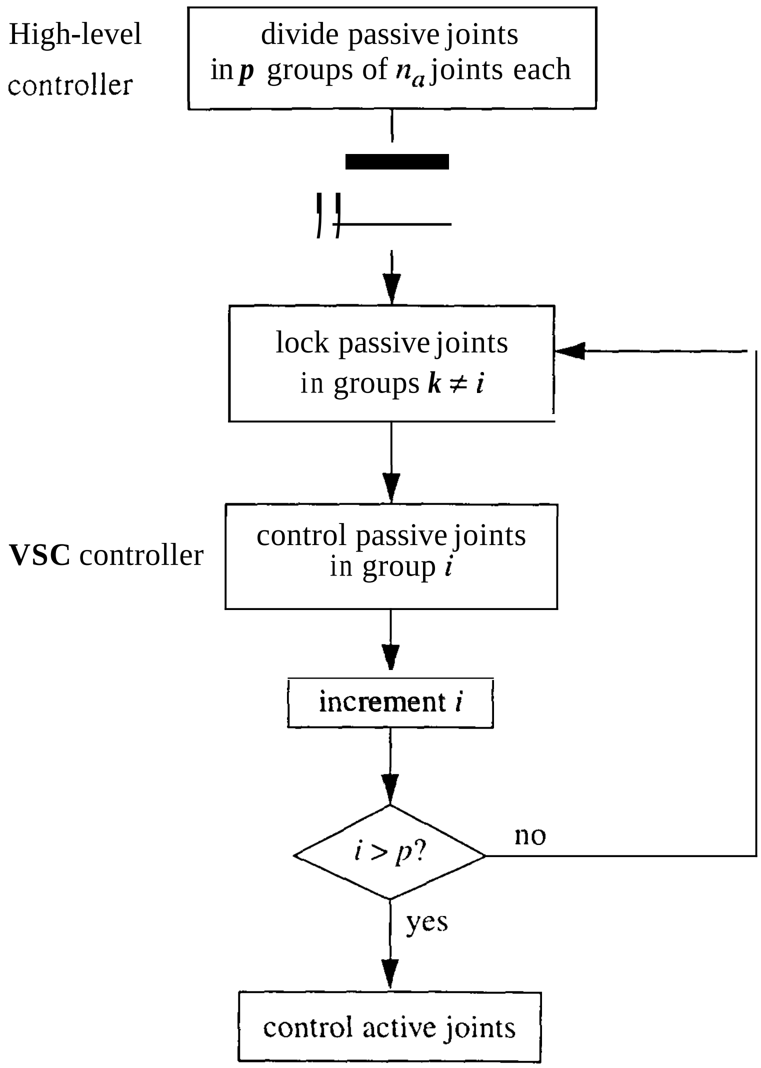


Figure 4.9: Flow diagram of the proposed method to control a joints of an underactuated manipulator with more passive than active joints.

2, 1}. Note that control from $q(t_0)$ to q^d requires $f = 4$ phases, three of them for the control of the passive joints. Note also that there is a total of nine different configurations of the manipulator, and six possible ways of reaching q^d from $q(t_0)$. More generally, the case $n_a = 1$ requires n control phases, for a total of $2^{n_p} + 1$ different configurations, and $n_p!$ possible ways of reaching q^d from $q(t_0)$.

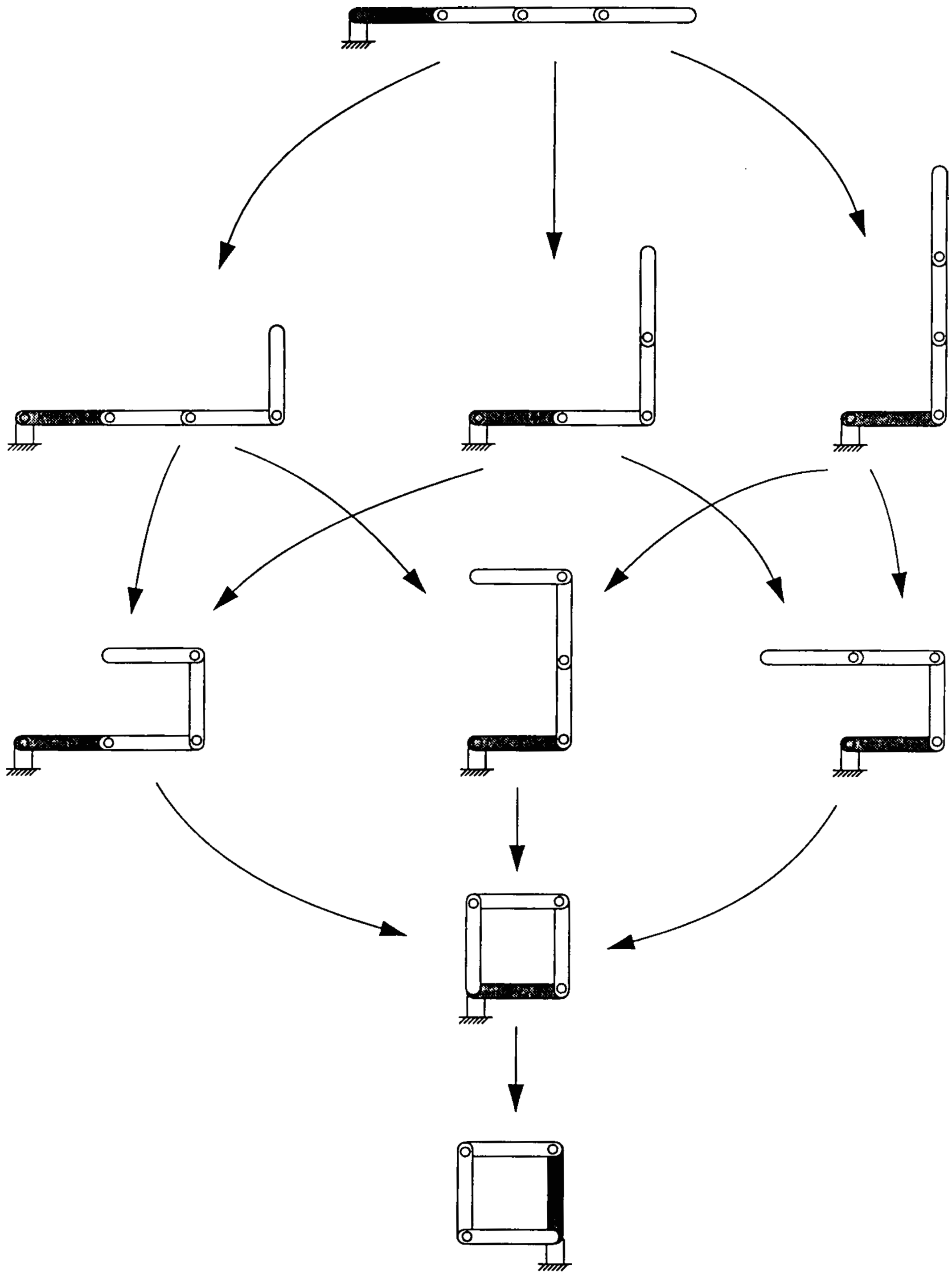


Figure 4.10: An underactuated manipulator with four links and one actuator can be controlled from an initial to a final configuration in six different ways.

We formulate the problem of selecting the optimal locking strategy among the $n_p!$ possible ones as an optimization problem. Consider the state transition diagram in Figure 4.11, where each state corresponds to the configurations in Figure 4.10. If we assign to all possible transitions a cost directly related to a critical optimization index, subsequent searching of the tree for the minimum cost path between the top-most and the bottom-most states will yield the locking strategy **that** is globally optimal with respect to the optimization index chosen. **A** very important index to be minimized, and the one we adopt in this work, is energy consumption, critical, for example, in space manipulation.

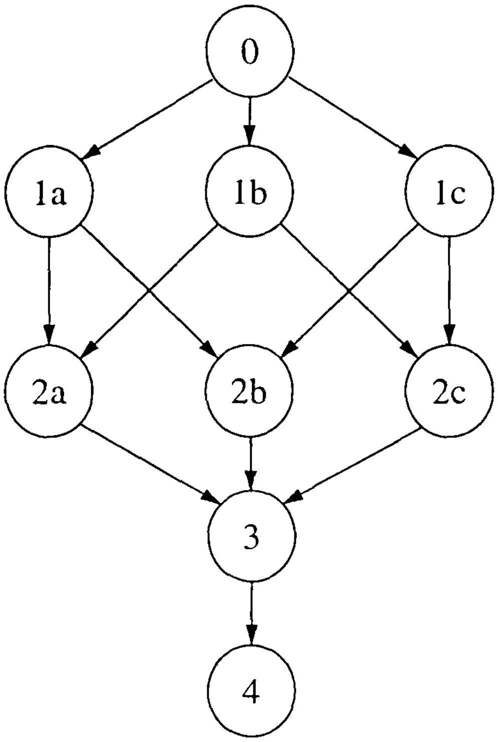


Figure 4.11: State transition diagram corresponding to the configurations in Figure 4.10.

We propose to use the sum of the reciprocal of the torque coupling indices at the configurations before and after the transition as the transition cost. To see why this is reasonable, recall the torque history of Figure 4.5 for $t < 1.4$ s. Note that the active joint "kicks" the passive one toward the desired position, and then "kicks" it in the other direction *to* prevent it from overshooting the set-point. From Equation (3.35), one can see that the larger the norm of W_{ua}, the smaller the norm of τ_a has to be for the same unit-norm acceleration $\ddot{q}_u$ (after the noninertial and gravitational effects are compensated for). In other words, the larger the norm of W_{ua} and, consequently, the magnitude of ρ_τ, the smaller the torque we have to supply to the actuators to produce the same motion. Furthermore, from Figure 4.5 one can see that the largest part of the torque is applied at the very beginning and closer to the end of the motion. If the torque coupling index at these instants is maximized, the spikes in the torque curve will be smaller and energy consumption will be reduced.

Even though it looks promising, the proposed optimization cost is not obvious to compute when ρ_τ is a function of both q_a and q_p. When control strategies P or $\boldsymbol{AP}$ are executed, we control q_c at the expense of not controlling q_a directly. Consequently, we cannot predict the value of q_a either before or after any transition (except, of course, before the first and after the last one). Therefore, it is impossible to compute $\rho_\tau(q_a, q_p)$. To overcome this difficulty, we propose to average the value of ρ_τ with respect to all possible values of the active joints' angles:

$$\bar{\rho}_\tau(q_p) = \frac{\int_{q_a} \rho_\tau^2(q)dq_a}{\int_{q_a} dq_a} \tag{4.42}$$

The use of $\bar{\rho}_\tau$ as the transition cost instead of ρ_τ may lead us to eventually choose a minimum cost path that does not attain the greatest possible energy consumption minimization. We can expect, however, that in the long run operation of the manipulator (i.e., for several different pairs of initial and final positions) energy minimization will be attained. The numerical results presented in the sequence confirm this expectation.

We consider now the last transition in the diagram in Figure 4.11. This is the step during which the active joints are controlled, with all passive ones locked. No matter what number we choose for this transition's cost, it will add equally to the cost of all possible control strategies (since all of them necessarily go through this transition to reach the final desired configuration). Consequently, we can arbitrarily set this transition's cost to zero.

After computing all transitions' costs, we search the corresponding tree for the minimum cost path. This can be done via exhaustive search, dynamic programming, or with tree-search algorithms such as **A*.** Any of these choices will find the path that is globally optimum with respect to the optimization index chosen.

We now discuss the case $n_a \neq 1$, which is an extension of the previous case. Two sub-cases can be distinguished:

(i) n_p is a multiple of n_a ($n_p = kn_a$, with $k \geq 2$ an integer number): in this case, the passive joints are controlled in groups of n_a joints each; k phases are necessary for their control, before the active ones can be controlled. The total number of phases, f, different configurations, c, and possible ways of reaching the desired configuration from the initial position, r, are respectively equal to:

$$
\begin{aligned}
f &= k + 1 \\
c &= \binom{n_p}{0} + \binom{n_p}{n_a} + \binom{n_p}{2n_a} + \dots + \binom{n_p}{kn_a} + 1 \\
r &= \binom{n_p}{n_a} \times \binom{n_p - n_a}{n_a} \times \binom{n_p - 2n_a}{n_a} \times \dots \times \binom{n_p - (k-1)n_a}{n_a} \\
&= \binom{kn_a}{n_a} \times \binom{(k-1)n_a}{n_a} \times \binom{(k-2)n_a}{n_a} \times \dots \times \binom{n_a}{n_a}
\end{aligned}
\tag{4.43}
$$

Note that these numbers respectively reduce to n, $2^{n_p} + 1$, and $n_p!$ when $n_a = 1$

(ii) n_p is not a multiple of n_a ($n_p = k_1 n_a + k_2$, $k_1 \geq 2$, $k_2 < n_a$, with k_1 and k_2 integer numbers): in this case, we first control $k_1 n_a$ passive joints in groups of n_a joints each, using k_1 control phases; then we control the remaining k_2 passive joints in one phase, and finally we control the active ones in one phase. The parameters f, c, and r in this case are equal to:

$$
\begin{aligned}
f &= k_1 + 2 \\
c &= \binom{n_p}{0} + \binom{n_p}{n_a} + \binom{n_p}{2n_a} + \dots + \binom{n_p}{k_1 n_a} + 2 \\
r &= \binom{n_p}{n_a} \times \binom{n_p - n_a}{n_a} \times \dots \times \binom{n_p - (k_1 - 1)n_a}{n_a} \\
&\quad - \binom{k_1 n_a + k_2}{n_a} \times \binom{(k_1 - 1)n_a + k_2}{n_a} \times \dots \times \binom{n_a + k_2}{n_a}
\end{aligned}
\tag{4.44}
$$

Example 4.3: Consider a 3-link planar underactuated manipulator with rotary joints, operating on a horizontal plane, with $I_a = \{1\}$, $I_p = \{2,3\}$ and parameters as in Table 4.8. The objective is to drive the manipulator from $q(t_0) = [0^o, 0^o, 0^o]$ to $q^d = [90^o, 90^o, 90^o]$. According to the discussion in this section, this objective is attainable by executing control strategy P twice, followed by control strategy **A.** Table 4.11 presents the VSC controller parameters adopted during each control strategy.

Table 4.11: VSC controller parameters.

Control strategy	Γ (s^{-1})	P (s^{-1})	ε (rd s^{-1})
P_1	10	10	0.2
P_2	10	10	0.2
A	10	10	0.2

To control the manipulator from $q(t_0)$ to q^d, two control sequences are possible, namely, {3,2,1} and {2,3,1}. They are shown in Figure 4.12. The numbers beside the arrows indicate the cost of each transition. The simplicity of this tree allows us to determine by inspection that, *for this* pair $[q(t_0), q^d]$, the optimal control sequence is {2,3,1}. Figure 4.13 presents the simulated motion of the manipulator, and Table 4.12 summarizes the results.

When the manipulator is mounted vertically, and the control gains are adopted as in Table 4.13, the resulting motion is that shown in Figure 4.14. The initial configuration is $q(t_0) = [-90^o, 0^o, 0^o]$ and set-point is $q^d = [90^o, 90^o, 90^o]$. The VSC performance is summarized in Table 4.14. ■

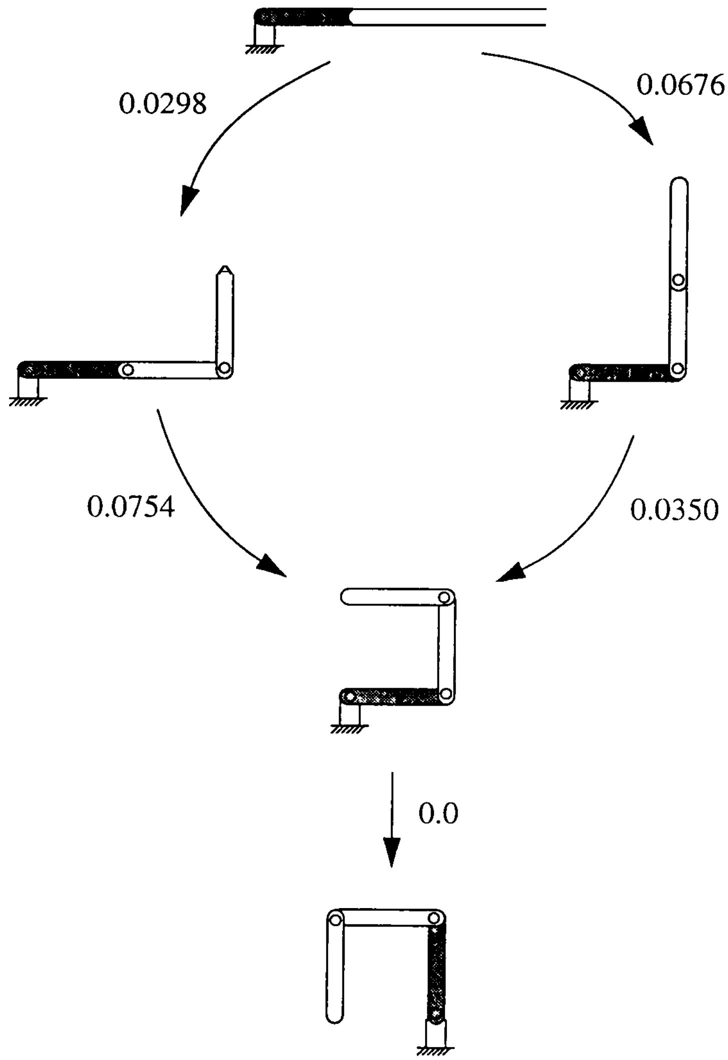


Figure 4.12: The two possible control sequences of an underactuated manipulator with three links and two passive joints.

Table 4.12: Performance of the **VSC.**

Joint	*Time to reach set-point* (s)	*Steady-state error* (°)
1	8.70	0.0014
2	2.15	0.0200
3	**4.30**	0.0202

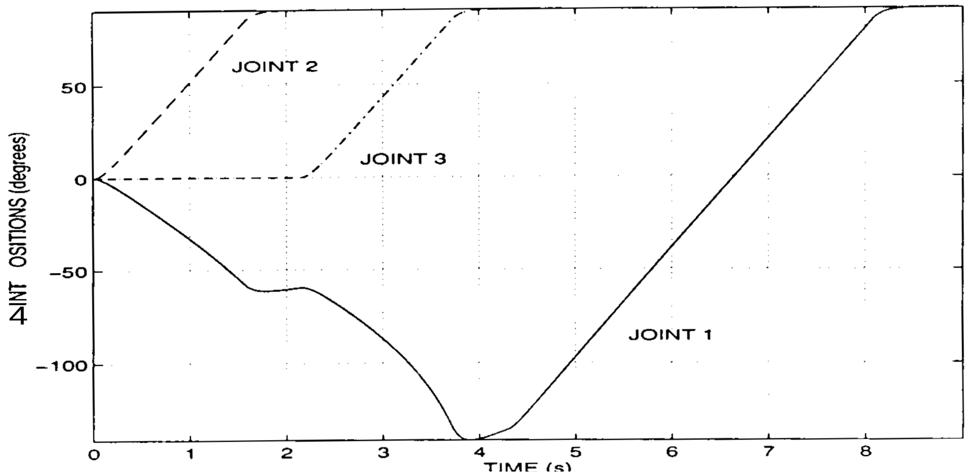


Figure 4.13: Simulated control of a 3-link underactuated manipulator with two passive joints.

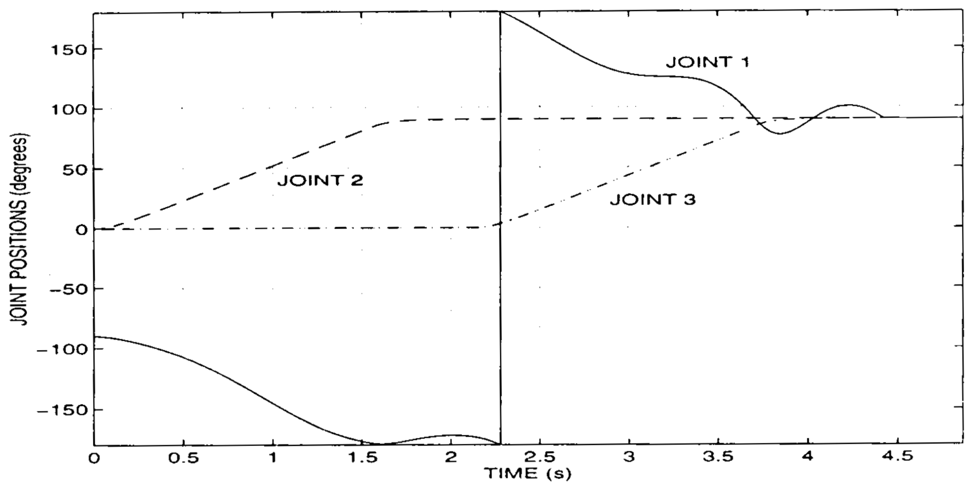


Figure 4.14: Simulated control of a 3-link underactuated manipulator with two passive joints under the influence of gravitational acceleration.

Example 4.4: We illustrate our claim that the optimal control strategy provides energy minimization in the long run operation of the manipulator (rather than on every individual PTP motion). Suppose the 3-link robot of the previous example is mounted on a space structure, and that it has to move in a PTP fashion to inspect bolts located at the set of Cartesian points shown in Figure 4.15, in the order shown. Set **A** consists of 20 PTP

Table 4.13: VSC controller parameters when the 3-link manipulator is mounted vertically.

Control strategy	Γ (s^{-1})	P (s^{-1})	E (rd s^{-1})
P_1	10	10	0.2
P_2	10	10	0.2
A	30	60	0.2

Joint	*Time to reach set-point* (s)	*Steady-state error* (°)
1	4.56	0.0709
2	2.15	0.0199
3	4.42	−0.0039

Cartesian point 1 to 2, 2 to 3, and so on.

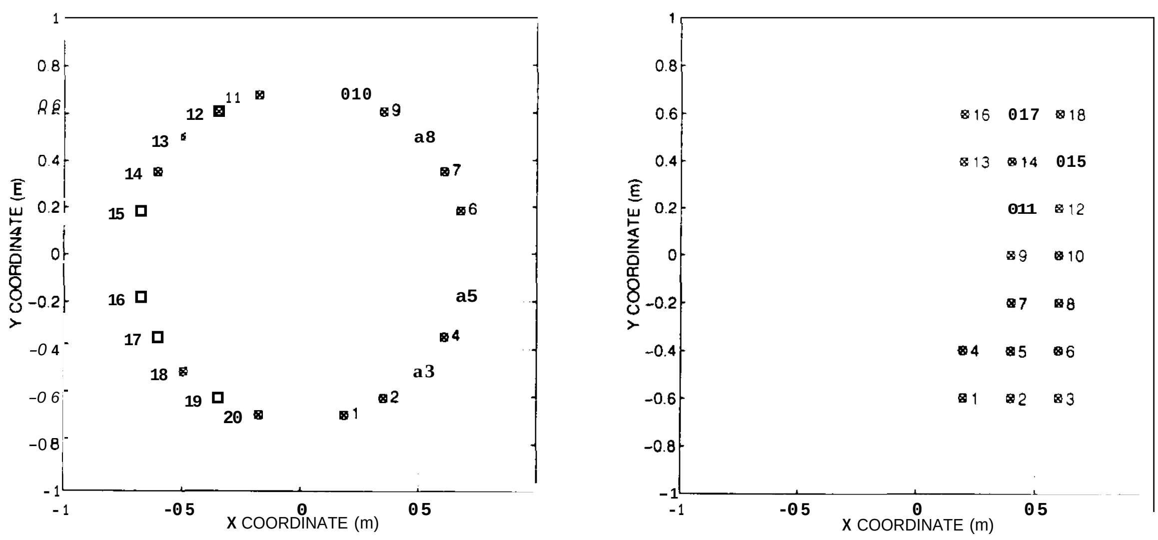


Figure 4.15: Two sets of Cartesian points to be reached by the 3-link manipulator.

We simulate the motion of the manipulator between each pair of Cartesian points i and $i+1$ using both control sequences {3,2,1} and {2,3,1}, and measure the amount of energy spent during each PTP motion as the sum of the absolute values of the torques applied at the first joint at every time step:

$$W_i = \sum_k \left|\tau_1(k\Delta t)\right| \tag{4.45}$$

When the actuator is a DC motor, W_i is directly proportional to the current applied to the motor and, therefore, to the actual electric energy spent on the motion.

Tables 4.15 and 4.16 present the amount of energy W_i spent for each sequence, the true optimal sequence, and the sequence chosen according to the optimal control method, as a function of the pairs of Cartesian points in sets **A** and B (due to the symmetry of the points in set **A,** we have to present only the data relative to points 1 through 5; the data repeats itself for the other points). **As** one can see, out of 37 PTP motions, our method led us to choose only one incorrect sequence (set B, motion from point 6 to 7). This incorrect choice was due to the fact that joint 1 ended up situated further from its set-point after controlling the passive ones when control sequence S_2 was executed. ■

Table 4.15: Energy spent to drive the **3-link** manipulator with two passive joints along the Cartesian points in set **A.**

Initial and final points	W_i for {3,2,1} (N m)	W_i for {2,3,1} (N m)	*True optimal strategy*	*Strategy chosen*
1 to 2	1078.0	1146.7	S_1	S_1
2 to 3	846.0	907.2	S_1	S_1
3 to 4	832.5	924.0	S_1	S_1
4 to 5	1208.1	1350.6	S_1	S_1
5 to 6	696.0	696.0	S_1	S_1

Table 4.16: Energy spent to drive the **3-link** manipulator with two passive joints along the Cartesian points in set B.

initial and final points	W_i for {3,2,1} (N m)	W_i for {2,3,1} (N m)	*True optimal strategy*	*Strategy chosen*
1 to 2	1049.2	1192.0	S_1	S_1
2 to 3	806.7	997.3	S_1	S_1
3 to 4	2382.1	1099.0	S_2	S_2
4 to 5	1187.8	1668.0	S_1	S_1
5 to 6	1232.7	1671.2	S_1	S_1
6 to 7	2007.2	2407.3	S_1	S_2
7 to 8	1948.9	2288.3	S_1	S_1
8 *to* 9	2285.7	1090.5	S_2	S_2
9 to 10	1095.7	1182.2	S_1	S_1
10 to 11	996.4	973.0	S_2	S_2
11 *to* 12	1185.7	1344.7	S_1	S_1
12 *to* 13	1299.7	3333.1	S_1	S_1
13 to 14	2094.7	1572.4	S_2	S_2
14 to 15	1149.9	1153.4	S_1	S_1
15 *to* 16	1042.5	1387.7	S_1	S_1
16 *to* 17	1402.4	1268.6	S_2	S_2
17 to 18	1054.2	1058.7	S_1	S_1

4.7 Cartesian control

As mentioned in Chapter 3, one is able to command an underactuated manipulator to follow not only an n_a-dimensional trajectory in joint space, but in Cartesian space as well. This task is achieved by first obtaining an open-loop relationship between the accelerations of the active Cartesian coordinates and the active torques, and by then designing a controller to force these coordinates to follow a desired trajectory. The open-loop relationship between $\ddot{x}_a$ and τ_a is found from (3.43); factoring out $\ddot{x}_p$ in the second line of that equation and substituting the result on its first line we obtain:

$$\tau_a = [M_a - M_a K_p (M_p K_p)^{-1} M_p](K_a \ddot{x}_a - K \dot{J} \dot{q}) - M_a K_p (M_p K_p)^{-1} b_p + b_a \quad (4.46)$$

Following a reasoning similar to that presented in Section 4.3, one may verify that the active Cartesian coordinates are controllable if $M_p K_p$ is a full rank matrix and the manipulator is not in a singular configuration.

Define the following n_a-dimensional sliding surface:

$$s_a = \Gamma_a \tilde{x}_a + \dot{\tilde{x}}_a \quad (4.47)$$

where Γ_a is an $n_a \times n_a$ diagonal positive-definite matrix. As before, we replace $\ddot{x}_a$ in (4.46) by

$$u = \Gamma_a \dot{\tilde{x}}_a + \ddot{x}_a^d + P_a \operatorname{sgn}(s_a) \quad (4.48)$$

to guarantee that x_a converges to its desired set-point or follows **a** desired trajectory.

Example 4.5: Consider again the 2-link underactuated manipulator of Example 4.1, operating on a gravity-free environment. Because this manipulator has only one actuator, we can control only one Cartesian coordinate. Let the active Cartesian coordinate be the x

coordinate, and its desired trajectory be sinusoidal, with amplitude 0.26 m and period 1 s. Figures 4.16 and 4.17 present the desired and the simulated motion of x_a, respectively, as well as the error along the trajectory, when the robust control method is utilized with $\delta = 1$ and 6 = 1.3. One can see that, because of the parameter uncertainty the error along the trajectory is amplified, nonetheless, the performance is still acceptable. ■

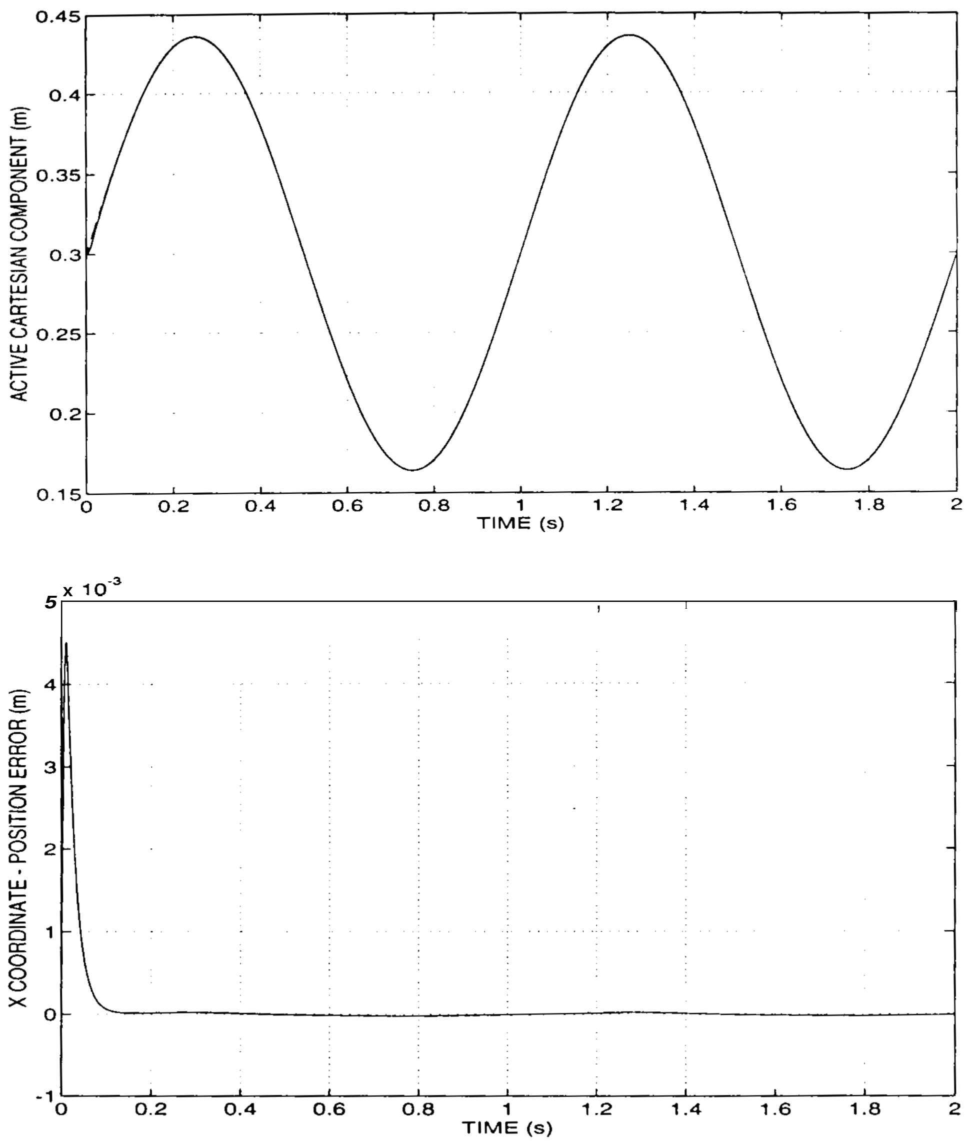


Figure 4.16: Simulated motion of a 2-link underactuated manipulator following a trajectory in the active Cartesian space, with 6 = 1.

Top: active Cartesian coordinate; bottom: error along the trajectory.

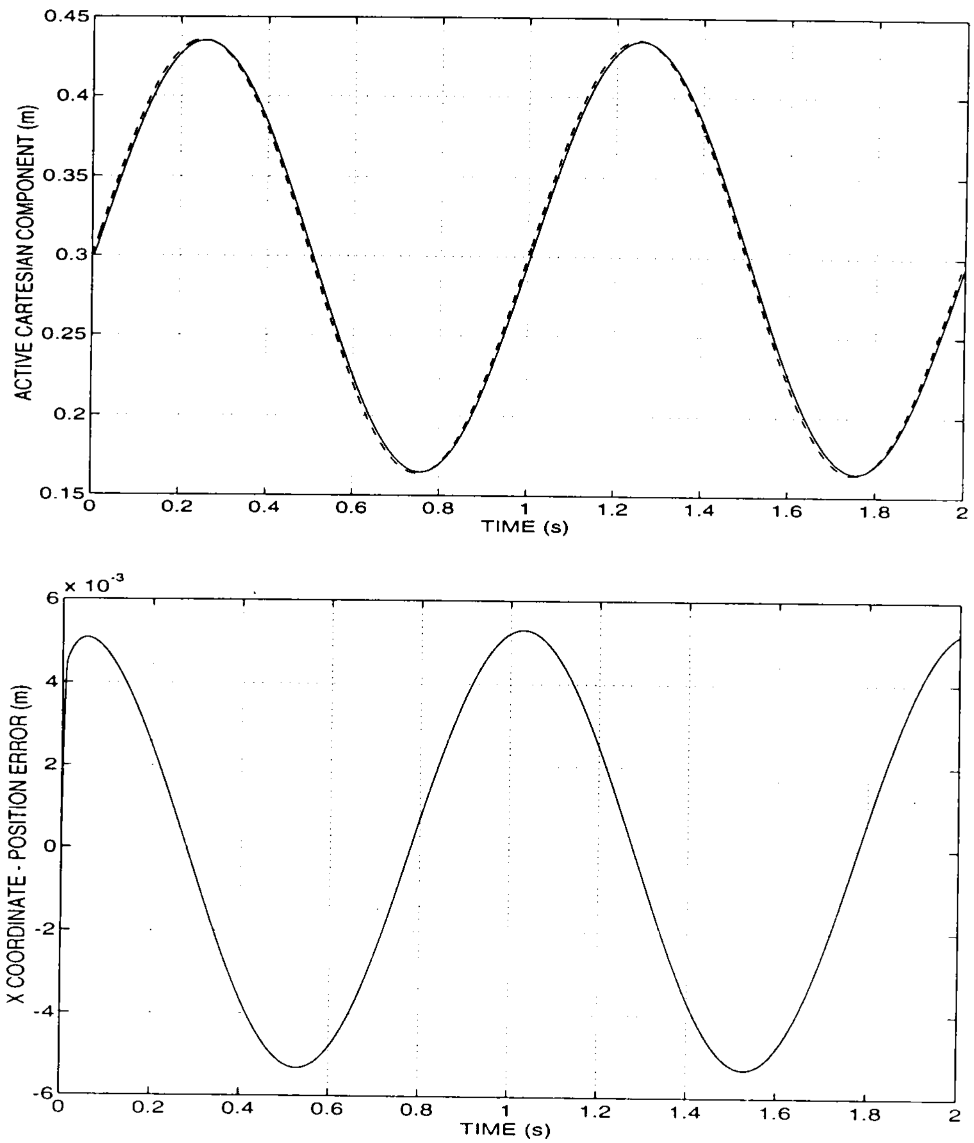


Figure 4.17: Simulated motion of a 2-link underactuated manipulator following a trajectory in the active Cartesian space, with $\delta = 1.3$.

Top: active Cartesian coordinate; bottom: error along the trajectory.

4.8 UARM testbed

To verify in practice the validity of the control method proposed, we built a 3-link planar rotary underactuated manipulator equipped with one actuator and two brakes. In this section, we present the details of this mechanism, which is called the UnderActuated Robot Manipulator (UARM).

4.8.1 Mechanical design and components

The UARM is pictured in Figure 4.18, and shown schematically in Figure 4.19. **All** joints are equipped with incremental encoders that provide angular position information. The angular velocities are obtained via numerical differentiation and low-pass filtering. The first joint is actuated by a DC motor, and the second and third joints are equipped with brakes.

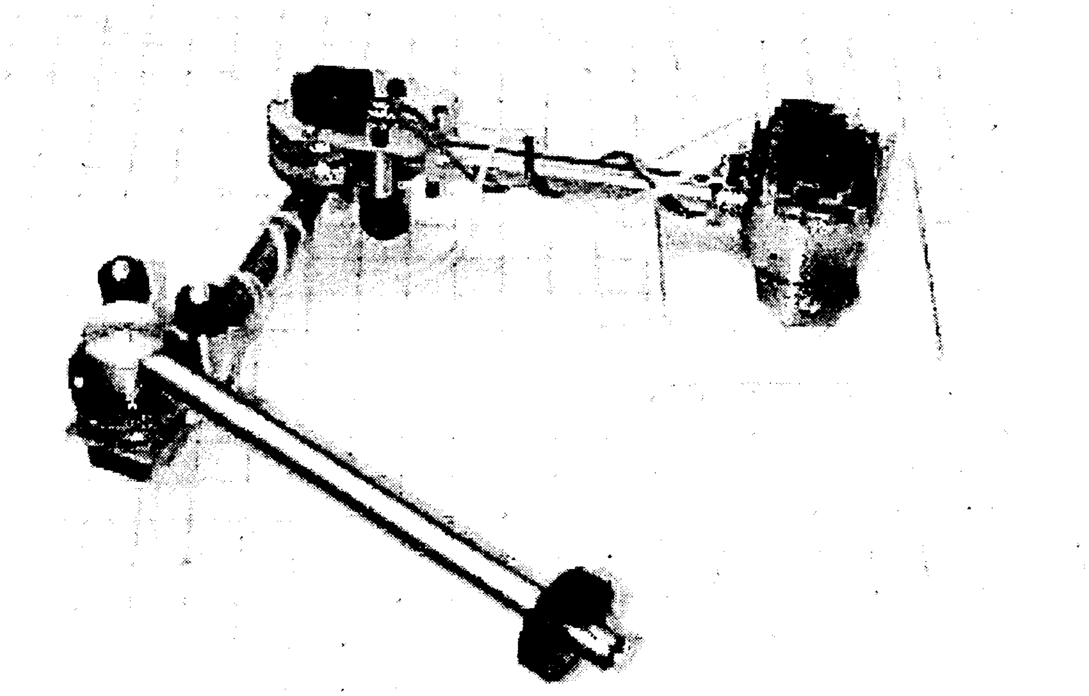


Figure 4.18: Underactuated robot manipulator (UARM).

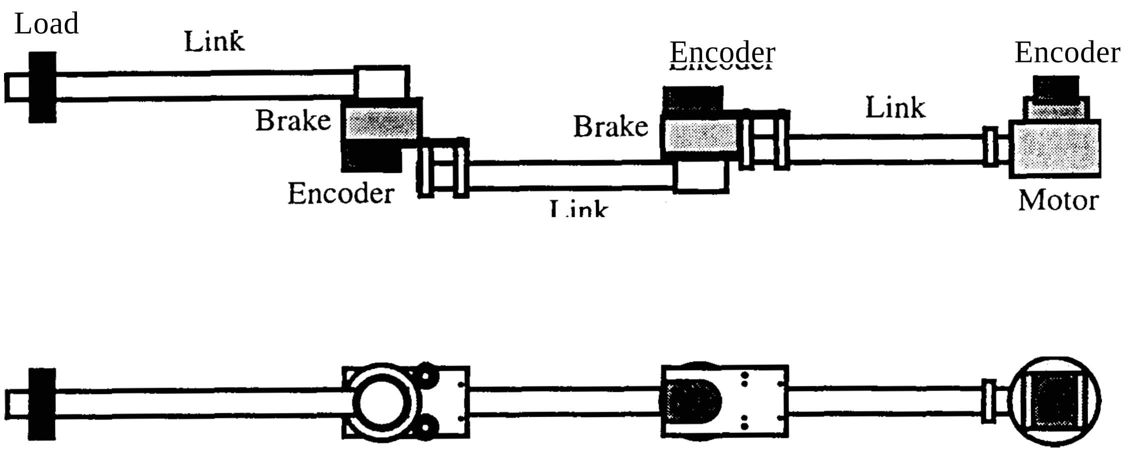


Figure 4.19: Schematic representation of UARM.

The UARM's kinematic and dynamic parameters were obtained from the niechanical drawings and the measurement of most of its parts. These parameters are given in Table 4.17. To simulate the control of a 2-link underactuated manipulator using the UARM, we keep the third joint locked at its extended position. The second and third links are then combined into one single, larger link. The resulting kinematic and dynamic parameters of this 2-link manipulator are given in Table 4.18.

link	$m_i\,(Kg)$	$I_i\,(Kg\,m^2)$	$l_i\,(m)$	$l_{ci}\,(m)$
1	0.994	0.0270	0.262	0.108
2	0.564	0.0167	0.208	0.150
3	0.368	0.0155	0.256	0.161

The actuator at the first joint is a DC motor with a 60:1 reduction gear mechanism. Its excursion is limited by the hardware, which turns off the power to the motor when the joint angle is outside the range $[-120^{\circ}, 120^{\circ}]$. This eliminates the need for a mechanical joint limiter. The brakes used in the second and third joints are on/off brakes with rated static

Table 4.18: Dynamic parameters of the **UARM** when joint 3 is locked at its extended position.

link	$m_i\,(Kg)$	$I_i\,(Kg\,m^2)$	$l_i\,(m)$	$l_{ci}\,(m)$
1	0.994	0.0270	0.262	0.108
2	0.932	0.0322	0.464	0.237

torque 50 lb-in, operating on a **24** VDC voltage. The brakes are manufactured by Inertia Dynamics (model number FB-22). In order to avoid damage to the manipulator, a mechanical joint limiter was devised for the second and third joints; it gives these joints an excursion of 240° degrees (from –120° to +120°). The encoders mounted at each joint are manufactured by Hewlett Packard, model HEDS 5540, and generate 1024 counts for each complete joint revolution. Additional hardware logic is utilized to provide ***a*** resolution of 4096 counts per revolution.

4.8.2 Control software

The UARM runs a software control system based on the Chimera 3.2 operating system, a multiprocessor real-time operating system which supports software written as ***a*** group of independent modules [46]. The basic architecture of the control software was taken from the DM^2 robot, also developed at the Space Robotics Laboratory at Carnegie Mellon University, but the modular design of this software made the adaptation straightforward.

The software architecture consists of two levels — a high-level system which interprets commands from a host workstation and monitors the execution of these commands, and a low-level system in which the basic control system is built from communication between a set of independent real-time software modules.

The low-level real-time modules each run as one (or more) separate threads, cycle at independent frequencies, and can be allocated at runtime to any of the processors on the system without modification. Modules can communicate through a variety of means, but real-time control modules generally exchange data at the beginning and end of each execution cycle through Chimera's state variable table mechanism. For example, a trajectory-generation module will typically output reference joint positions to a state variable at the end of each cycle; and a controller module like that shown in Figure 4.20 will input state variables representing reference and measured joint positions, and output actuator torque values to a state variable at the end of each cycle. The high-level system communicates with these modules through a high-speed message-passing system.

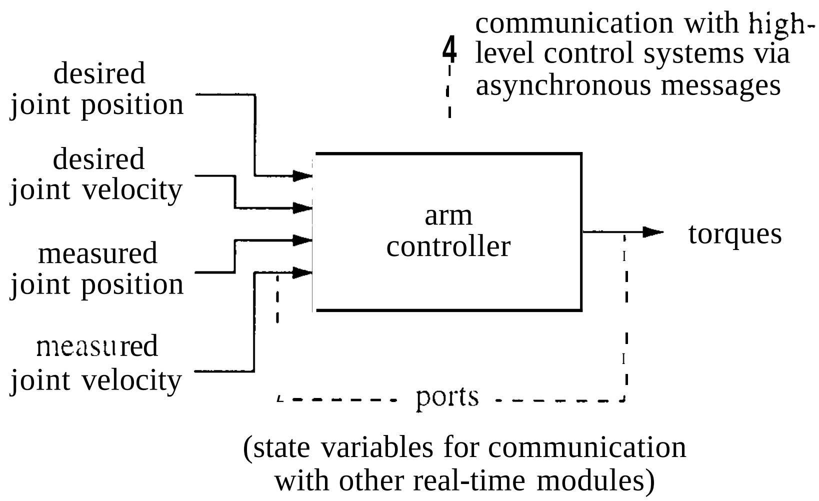


Figure 4.20: Arm controller module's inputs and outputs.

The purpose of the low-level real-time controller modules is to provide strong controller performance while insulating higher-level modules from the details of robot's hardware configuration. The UARM uses a simple arrangement of these modules, shown in Figure 4.21. The modules which interface to the robot hardware are the encoder and actuator modules. The encoder module writes joint positions to the state variable table, and the actuator module writes torque values from the state variable table to the joint actuator and the joint brakes. A trajectory-generation module provides reference values for the joint positions and velocities, and the controller module reads measured and reference joint positions from the state variable table and outputs torque values to the table.

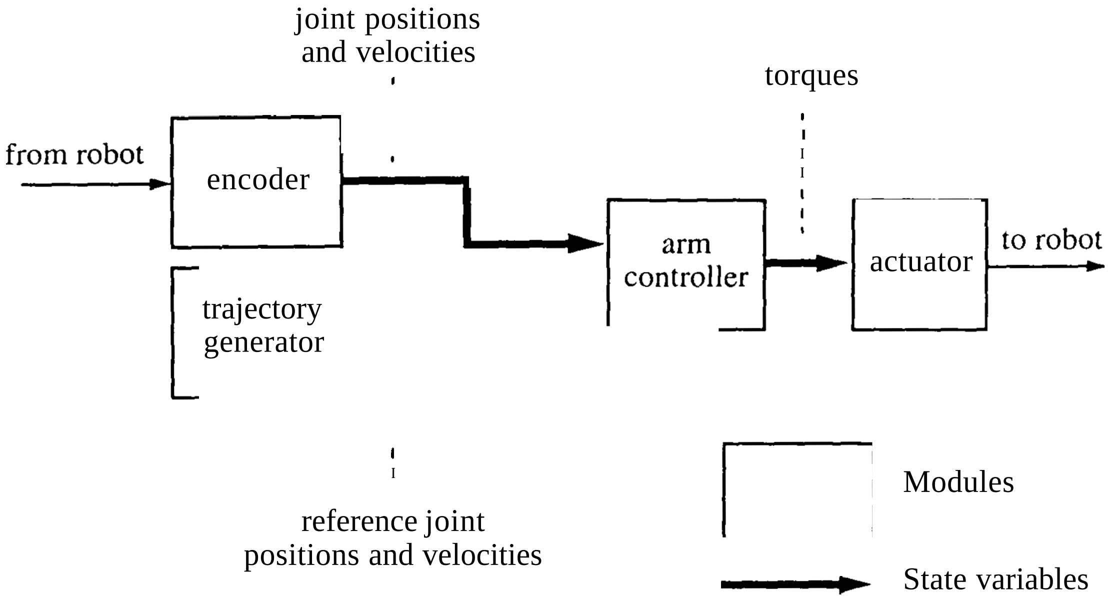


Figure 4.21: Real-time control architecture: combination of modules for control of UARM.

Both the trajectory-generator module and the controller module operate by using a set of *control-objects* which are selected from a library of objects at runtime. Trajectory-generation objects can correspond 'to functions such as set-point generators, linear-interpolators, or cubic spline followers. Real-time controllers can be created through combinations of control objects such as PID controllers, variable structure controllers, and torque limiters. By indicating which control objects to use, a high-level system is able to specify which trajectory-generation or control strategies to use at a particular time. Default controllers and the parameters of these controllers are specified in a configuration file which is read at the beginning of the software's execution.

Creation of new control-objects is facilitated by a code-generation tool which reads information about the controller from the developer and then uses this information to create a source-code skeleton from a template file. Filling in the code skeleton to produce a functional control-object takes just a few minutes. Two lines of code are needed to interface the new object to the library of usable control-objects for a given module.

High level control within the software system is provided by a software library consisting of a text-based command-parser and a separate thread which executes these commands and checks their status at a fixed cycle rate.

This library is designed as a framework for high-level code to exist within a Chimera program. The command parser allows a user to specify tasks for the robot to perform. These commands are grouped recursively into lists of commands that are to be executed either sequentially or in parallel. This way of scheduling tasks allows for simple specification of complicated jobs consisting of subtasks which may be executed quickly and in parallel when possible, or sequentially if later tasks depend on the completion of previous tasks.

High-level commands are defined in terms of a combinations of *atomic* tasks, which together span the technical ability of the robot. These atomic commands are written as C++ classes, and are called by passing commands which resemble function calls in the C-language to the interpreter.

4.9 Experiments

We implemented the foregoing joint space robust control methods on the UARM. Because the UARM has only one actuator, we can reproduce here only the results of Examples 4.1 and 4.3, but not those of Example 4.2. We start with the results of controlling the UARM as if it were a 2-link underactuated manipulator with $I_a = \{1\}$ and $I_u = \{2\}$. Figure 4.22 presents the joint angles history for two experiments with different initial and final conditions. Table 4.19 summarizes the results. **As** one can see, the experiments validate the simulations presented in this chapter: by executing control strategy $\boldsymbol{P}$ followed by strategy $\boldsymbol{A}$, the VSC is able to smoothly control the robot to an equilibrium position.

Operating the UARM as a 3-link underactuated manipulator with $I_a = \{1\}$ and $I_u = \{2,3\}$ gives the result shown in Figure 4.23, summarized in Table 4.20. The VSC executes the sequence of control strategies $\boldsymbol{P}$-$\boldsymbol{P}$-$\boldsymbol{A}$ to bring the manipulator to the desired equilibrium positions.

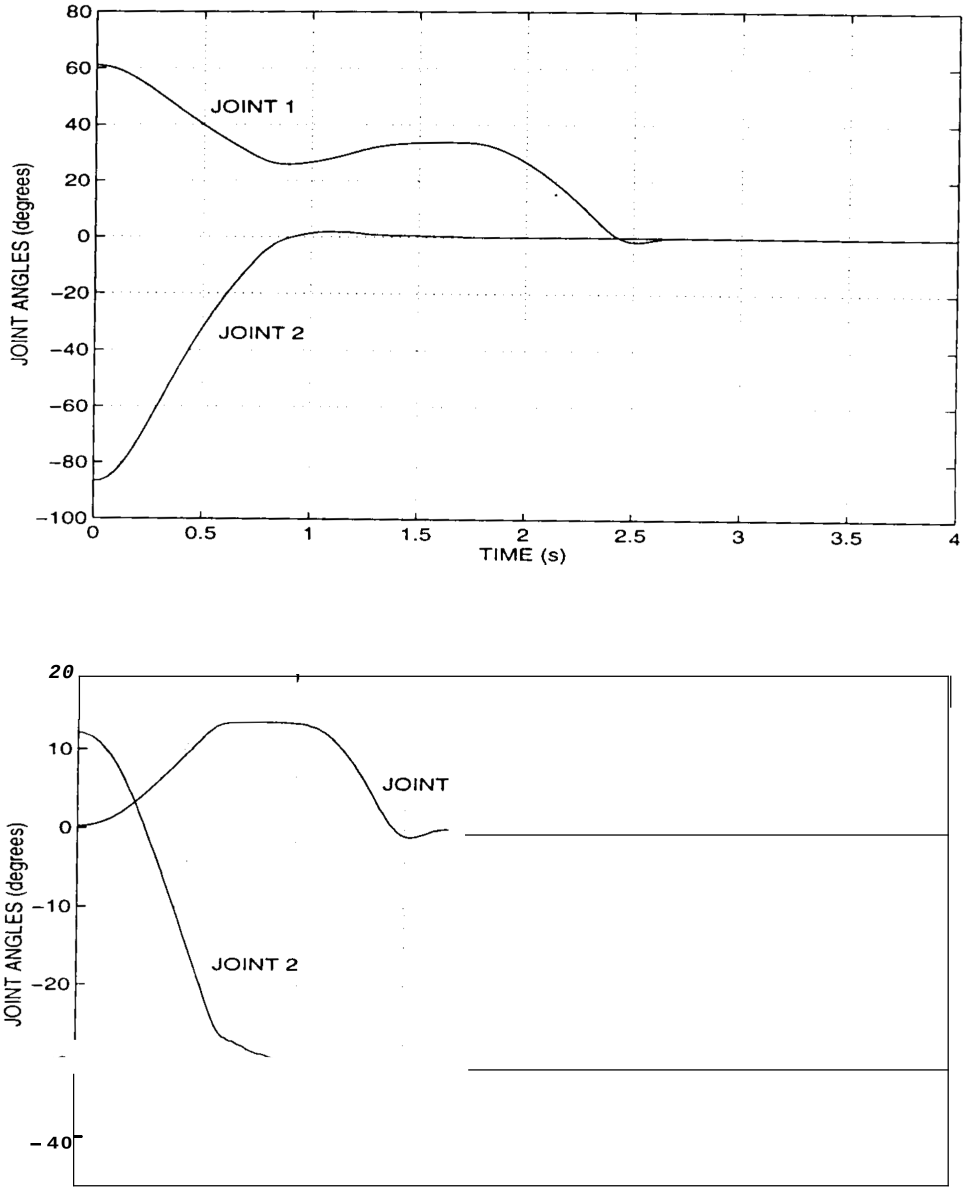


Figure 4.22: Experimental evaluation of the robust control on a 2-link underactuated manipulator.

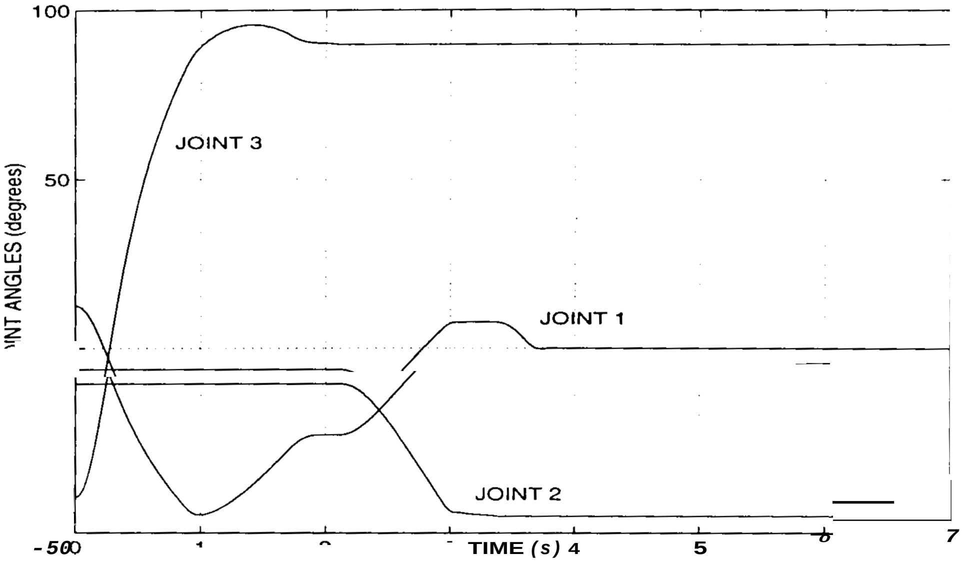


Figure 4.23: Experimental evaluation of the robust control on a 3-link underactuated manipulator with one actuator.

Experiment	*Joint*	*Time to reach set-point* (s)	*Steady-state error* (°)
1	1	1.70	0.0000
	2	2.75	0.0000
2	1	1.00	0.0090
	2	1.85	0.0000

Joint	*Time to reach set-point* (s)	*Steady-state* ***error*** (°)
1	2.20	0.0060
2	3.40	*0.0000*
3	3.90	*0.0000*

Chapter 5

Motion Planning for Underactuated Manipulators

In the previous chapters, we discussed the kinematic and dynamic properties of underactuated manipulators and developed a robust control methodology for these systems. In this chapter, we extend those results by considering static workspace obstacles and propose an approach to navigate the manipulator's links away from them. We propose the ***constrained*** *configuration space* concept to represent the space of all possible configurations of an underactuated manipulator constrained by nonholonomic differential equations. We then determine collision-free motions for all three possible control strategies ($(A, P$, and $AP)$, and combine several of these motions into one global collision-free trajectory connecting the start and the goal configuration.

5.1 Constrained configuration space

Path and motion planning for manipulators working on static environments is usually performed in configuration space, i.e., the space of all possible configurations of the manipulator. In our case, the joint space and the configuration space are coincident. When the manipulator is not equipped with passive joints, the point representing it in configuration space can move in any direction (provided that the manipulator does not collide with obstacles and that joints remain within their limits). On the other hand, the point representing an underactuated manipulator in configuration space cannot move in any direction. In fact, the nonholonomic constraints due to the presence of unlocked passive joints force the manipulator to move along a hypersurface described by these constraints. For this reason, we use the term *constrained configuration space* (CCS) to refer to the configuration (or joint) space of underactuated manipulators. Our objective in the next section is to characterize the motion of the underactuated manipulator inside the CCS so that collision-free trajectories can be obtained.

5.2 Characterization of the CCS trajectories

In the constrained configuration space, not all coordinates are independently controlled. While we control q_c, we cannot control q_r, and the coordinates not being controlled may lead the manipulator to collide with other machines or objects present in its workspace. We need to characterize the motion of q_r while q_c is being controlled, so that we can check for collisions or out-of-range conditions before commanding the manipulator to execute any motion. The nonholonomic constraints **(3.7)** cannot be used for this purpose, because they imply that the motion of the manipulator depends on its current position and velocity. Therefore, motions starting from the same positions may be substantially different if the velocities differ. What we wish to obtain is a description of the motion that depends only on the current position of the manipulator. It has been shown, however, that in general it is impossible to obtain an algebraic expression of the form $g(q, t) = 0$ from the integration of the nonholonomic constraints **(3.7)** **[34].** (It is important to remember that integration of the nonholonomic constraints makes sense only when the robot is being controlled by strategies $\boldsymbol{P}$ or $\boldsymbol{AP}$.) Here we discuss a method to overcome this difficulty by studying these constraints and showing how they can be integrated approximately to any desired accuracy.

Equation **(3.7)** can be integrated to first-order constraints if the gravitational forces acting on the passive joints are constant, and, in addition, if the following relationship is satisfied [34]:

$$S = \frac{\partial(\dot{q}^T M \dot{q})}{\partial q_p} = 0 \tag{5.1}$$

The first requirement leads us to consider in the sequence manipulators with passive joints mounted horizontally or operating in gravity-free environments. The $1 \times n_p$ vector S is the *sensitivity* of ***the kinetic energy*** with respect to the passive joints' positions. Equation (5.1) implies that the kinetic energy of the manipulator must not be a function of the passive joints' positions. It is true when either (I) M is a constant matrix; (11) M is a function of q_a only, i.e., not a function of q_p.

Condition (I) is very restrictive; a manipulator must be specially designed to have a constant inertia matrix [55] (except, for example, when the manipulator consists only of mutually perpendicular prismatic joints). Condition (11) is as restrictive as condition (I), because the inertia matrix of robot manipulators usually depends on the angles of all joints. One can verify that, even for **a** simple 2-link underactuated manipulator with joint 1 active and joint 2 passive, neither condition (I) nor condition (11) is satisfied, unless the mechanism is specially designed with a mass distribution that locates the center of mass of the second link exactly at the second joint (see Equation **(3.24)).**

From the discussion above, it is unlikely that manipulators have inertia matrices obeying either condition (I) or (11). On the other hand, the joint velocities of manipulators can be commanded at will. We then study the following condition which makes Equation (5.1) true: (111) the manipulator's velocity $\dot{q}$ is zero. Condition (111) is not as restrictive as conditions (I) and (11), for it does not impose rules on the structure of the manipulator. It allows one to integrate the second-order constraints (3.7) to the form:

$$M_{uc}\dot{q}_c + M_{ur}\dot{q}_r + k = 0 \tag{5.2}$$

where ***k*** is a constant dependent on the initial velocities. If, in addition, $M_u(q)\dot{q}$ defines an involutive distribution, then (3.7) can be integrated to the form $g(q, t) = 0$ [34].

One might claim that condition (111) is of no practical use, for it implies that the manipulator does not move. If, however, the joint velocities are "small" enough to keep S close to zero during the manipulator's motion, then (5.2) will be a good approximation of the integral of (3.7). This allows us to obtain first-order nonholonomic constraints from (3.7) without the downside of having to consider only manipulators with a constant inertia matrix or whose passive joints are cyclic.

The sensitivity S is a function of q, $\dot{q}$, and the manipulator's kinematic and dynamic parameters. We can intuitively expect that S grows larger (and, therefore, does not allow us to approximately integrate (3.7) to (5.2)) when we force the robot to move either faster or farther from its initial position. For any given manipulator, one can simulate its motion with all possible control strategies ***(A,*** P, or ***AP)*** and compute the value of S along a desired trajectory of q_c. In practice, the nonholonomic constraints vanish when control strategy ***A*** is used, and we have to measure S only along a P or an ***AP*** motion. We demonstrate this in the next example.

Example 5.1: Consider a 2-link underactuated manipulator whose first joint is active and whose second joint is passive.[1] For simulation purposes we adopt the dynamic parameters shown in Table 5.1. The sensitivity S is given by:

$$S = \frac{\partial(\dot{q}^T M \dot{q})}{\partial q_2} = -2 m_2 l_1 l_{c_2} \sin q_2 (\dot{q}_1^2 + \dot{q}_1 \dot{q}_2) \qquad (5.3)$$

Table 5.1: Dynamic parameters of a 2-link manipulator with one actuator.

Link	$m_i\,(Kg)$	$I_i\,(Kg\,m^2)$	$l_i\,(m)$	$l_{ci}\,(m)$
1	0.994	0.0270	0.262	0.108
2	0.932	0.0322	0.464	0.237

For this robot, an ***AP*** motion is not possible, since we can control, at any instant, only joint 1 or joint 2 but not both concurrently. To check whether S is kept sufficiently small along a

1. If joint 1 is passive and joint **2** is active, the manipulator is completely holonomic and the nonholonomic constraints are integrable to the form $g(q, t) = 0$ [56].

P motion, we simulated the motion of the robot starting from rest from the initial position $q(t_0) = [0^o, 0^o]$. We recorded the maximum absolute value of S for different combinations of the angular displacement of q_2 (which we represent by Δq_2), and its average angular velocity, $\dot{q}_2$. To avoid locking the passive joints while they were moving and, therefore, to avoid momentum from being transferred to the manipulator's base, all joint trajectories were designed as 5th order polynomials, with zero initial and final velocity and acceleration. The results are presented in Figure 5.1, as contour curves of $\max(|S|)$. Note how S grows larger when we (i) command q_2 to move faster through a fixed angular displacement, or (ii) command q_2 to move farther with a fixed average angular velocity. For every angular displacement desired for q_2, we can use the contour curves in Figure 5.1 to find the maximum possible angular velocity that guarantees that S is kept below a pre-defined threshold S_{min}. For example, we can command q_2 to move 20 degrees at an average angular velocity of **20** degrees/s and always keep S below 0.001. ■

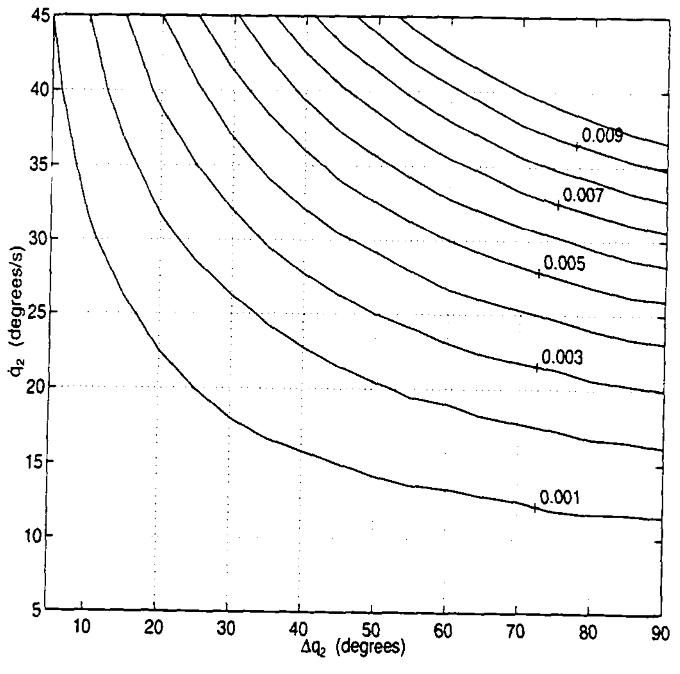


Figure 5.1: Contour curves of $\max(|S|)$ along P motions followed with different lengths and speeds.

The previous example shows that we can control an underactuated manipulator at reasonable speeds and yet still maintain the value of its sensitivity S very close to zero. Therefore, we can assume that Equation (5.1) is satisfied and that the second-order nonholonomic constraints **(3.7)** are integrable to (5.2). We furthermore assume that all motions start from rest, which implies that $\boldsymbol{k} = 0$. Because $\dot{q}_c$ is actively controlled, $\dot{q}_r$ can be found from (5.2):

$$\dot{q}_r = -M^{\#}_{ur} M_{uc} \dot{q}_c = G_{rc} \dot{q}_c \tag{5.4}$$

Equation (5.4) implies that the motion of the manipulator depends only on its current position, because G_{rc} is a function only of the manipulator's position. Equation (5.4) allows one to plan collision-free trajectories for underactuated manipulators, by commanding q_c to follow a "small" displacement Δq_c *starting from* ***rest*** and inferring the resulting displacement Δq_r via numerical integration of **(5.4)**. The combination of several "small" displacements of q_c allows one to drive the manipulator across "large" distances in the CCS while checking for the corresponding motion of q_r. This method is very similar in nature to the planner described in [20], Chapter 9, which consists of numerically integrating (5.4) and searching for a path of q_c that brings a mobile robot to a desired configuration, while, at the same time, checking for collisions with configuration space obstacles. The difference is that now we can switch between $\boldsymbol{A}$, P, and $\boldsymbol{AP}$ motions to follow different surfaces in the CCS to find a collision-free path to the goal configuration. We will refer to such surfaces as $\boldsymbol{A}$, P, or $\boldsymbol{AP}$ surfaces, according to the control strategy being executed.

Example 5.2: The 2-link manipulator with dynamic parameters in Table 5.1 operates inside a workspace defined by $\Theta = \{(q_1, q_2): -120^\circ \le q_1 \le 120^\circ, -120^\circ \le q_2 \le 120^\circ\}$. When joint 1 is controlled, with joint **2** locked (an $\boldsymbol{A}$ motion), q_r has dimension zero and q moves along the $\boldsymbol{A}$ surface $q_2 = k_2$, with k_2 a constant. On the other hand, when joint ***2*** is controlled (a P motion), q moves along the $\boldsymbol{P}$ surface described by:

$$\dot{q}_1 = -\frac{M_{22}}{M_{21}}\dot{q}_2 = -\frac{m_2 l_{c_2}^2 + I_2}{m_2\left[l_{c2}^2 + l_1 l_{c_2} c_2\right] + I_2}\dot{q}_2 \tag{5.5}$$

The *velocity gain* ***matrix*** $G_{rc}(q_2) = -M_{22} / M_{21}$ (in this case a scalar) is defined everywhere inside Θ since M_{21} is positive and bounded away from zero inside the workspace. ■

We now possess all the elements we need to characterize the constrained configuration space trajectory of an underactuated manipulator. When we use control strategy ***A***, the manipulator moves on the n_a-dimensional ***A*** surface $q_p = k_p$ (with k_p a constant vector), which is perpendicular to all passive joints' axes. When we use control strategies *P* or *AP*, the manipulator moves on the n-dimensional P or ***AP*** surface (5.4).

Example 5.3: Figure 5.2 presents the ***A*** and ***P*** surfaces for the 2-link underactuated manipulator inside the workspace Θ. The discretization of the ***CCS*** is equal to 20' in both axes. **As** expected, the ***A*** surfaces (represented by dashed lines) are 1-dimensional and perpendicular to the axis of the passive joint. The ***P*** surfaces (represented by full lines), on the other hand, are 2-dimensional and obey Equation (5.2). We simulated the motion of the manipulator inside Θ by combining several ***P*** and several ***A*** motions, starting and ending at $q(t_0) = q^d = [0^{\circ}, -80^{\circ}]$. Each P motion consisted of a 10' displacement of joint 2 at 10°/s; and each **A** motion of a displacement of joint 1 at 20°/s (remember that, when control strategy **A** is used, the nonholonomic constraints vanish and the joints' velocities are limited only by the torque their actuator can apply). Here again we used 5th order polinomial joint trajectories to avoid momentum transfer to the base. The resulting trajectory is shown in bold lines in Figure 5.2. One can see that the P motions follow the ***P*** surfaces precisely, and so do the ***A*** motions with respect to the ***A*** surfaces. ■

5.3 Planning collision-free motions

As we demonstrated in the previous section, we can predict a robot's motion in the constrained configuration space, no matter what type of motion ***(A, P***, or ***AP)*** is executed.

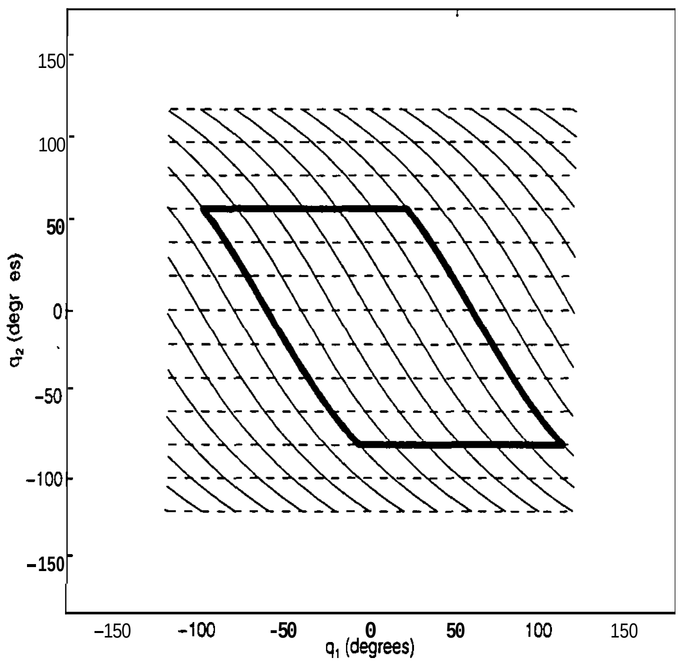


Figure 5.2: *A* surfaces (dashed lines) and P surfaces (full lines) for the 2-link planar underactuated manipulator. When the sensitivity S is small, the manipulator motion in the CCS follows the *A* and P surfaces.

In this section, we present an approach to drive the robot in a collision-free fashion inside the CCS. To make the presentation easier we will use the 2-link manipulator as a testbed. In this case, the dimension of the CCS allows us to obtain a graphical solution to the path-planning problem. We then discuss the generalization of this solution to manipulators with larger numbers of joints.

The inputs to the obstacle avoidance method are a bitmap representation of the CCS obstacles, and a graphical representation of the *A* and P surfaces, as in Figure 5.2. The discretization of the CCS plays an important role which will be described later. Note that if the obstacles are described by a set of polygons or surfaces, one can apply, for example, Pérez's method [25] to obtain the bitmap. In our case, let the obstacles be the two bins shown in Figure 5.3, the manipulator's initial configuration be $q(t_0) = [0^\circ, 0^\circ]$, and its goal configuration be $q^d = [90^\circ, 0^\circ]$. Figure 5.4 presents the CCS obstacles and joint limits, with the 'X' marks representing the initial and goal configurations.

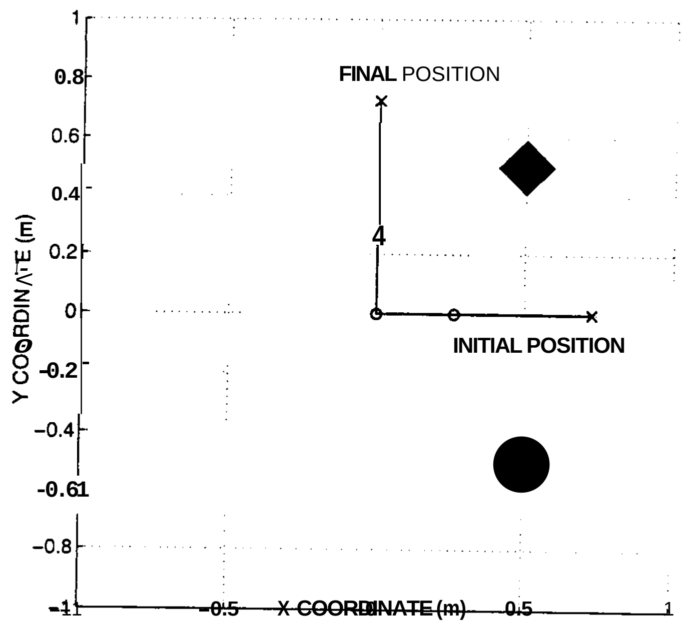


Figure 5.3: Obstacles in the workspace of the 2-link manipulator.

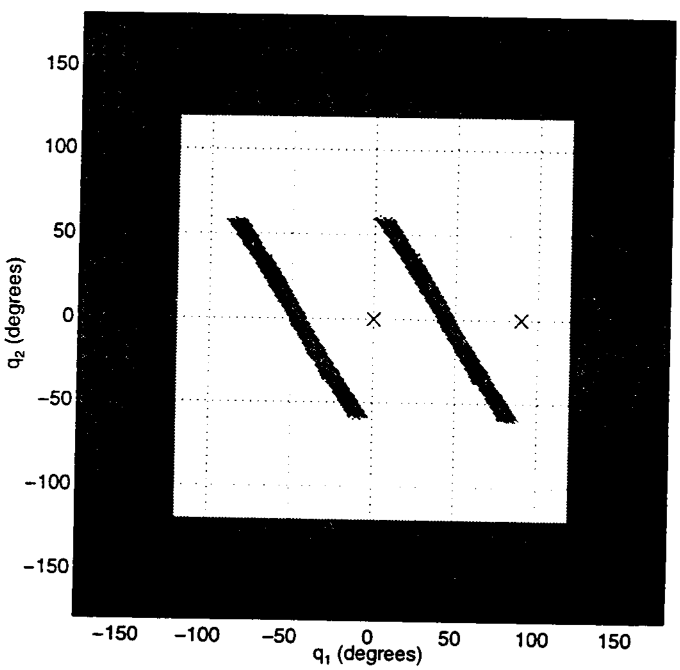


Figure 5.4: Initial and goal configurations,joint limits, and obstacles in the constrained configuration space.

To reach the goal configuration, we must drive the manipulator around the rightmost obstacle (the upper bin). To this end, we superimpose the ***A*** and ***P*** surfaces of Figure 5.2 on top of Figure 5.4, as shown in Figure 5.5. One can see that an **A** and a ***P*** surface pass directly over the initial configuration. If the manipulator follows the ***A*** surface toward the right, it hits the rightmost obstacle. On the other hand, if the manipulator follows the ***P*** surface until it reaches the circle located at $4 = [-55^{\circ}, 80^{\circ}]$, it avoids both obstacles. Next, the manipulator can be made to move along the ***A*** surface described by $q_2 = 80^{\circ}$, until it reaches the circle at $q = [24^{\circ}, 80^{\circ}]$. The manipulator then can follow another ***P*** surface to reach the circle at $q = [80^{\circ}, 0^{\circ}]$, and, finally, can reach the goal configuration following the ***A*** surface $q_2 = 0^{\circ}$. Note that the last ***A*** motion is necessary because no ***P*** surface passes directly over the goal configuration. **A** finer discretization of the CCS would solve this problem. The resulting collision-free trajectory is shown with a bold line in Figure 5.6. In that figure, we also present the robot at the initial and goal configurations, as well as at the configurations marked by the circles in Figure 5.5. Figure *5.7* presents the joint trajectories along the entire motion. It is important to note that: (i) the motions between the via points represented by circles in Figure 5.5 are *closed-loop* motions, executed with the variable structure controller; (ii) each ***P*** motion was subdivided into several motions of 10° followed at an average speed of 10°/s each, for the sake of keeping the sensitivity S below 0.001 at all times.

The schematic representation of the manipulator in the bottom of Figure 5.6 is valuable for understanding the whole obstacle avoidance method. Because the manipulator cannot swing directly from $q(t_0) = [0^{\circ}, 0^{\circ}]$ to $q^d = [90^{\circ}, 0^{\circ}]$ without hitting the upper bin with the second link, it folds the second link to make the link avoid the bin. After the manipulator clears from the obstacle, it unfolds the second link back to its extended position, and finally swings to the goal configuration.

The path-planning method described above can be automatically implemented in four steps: (1) "unwarp" the ***P*** surfaces so that the warped grid of ***A*** and ***P*** surfaces shown in Figure 5.2 becomes a regular rectangular grid. The unwarping includes mapping the CCS obstacles to their new location in the rectangular grid; *(2)* assign to each square of the

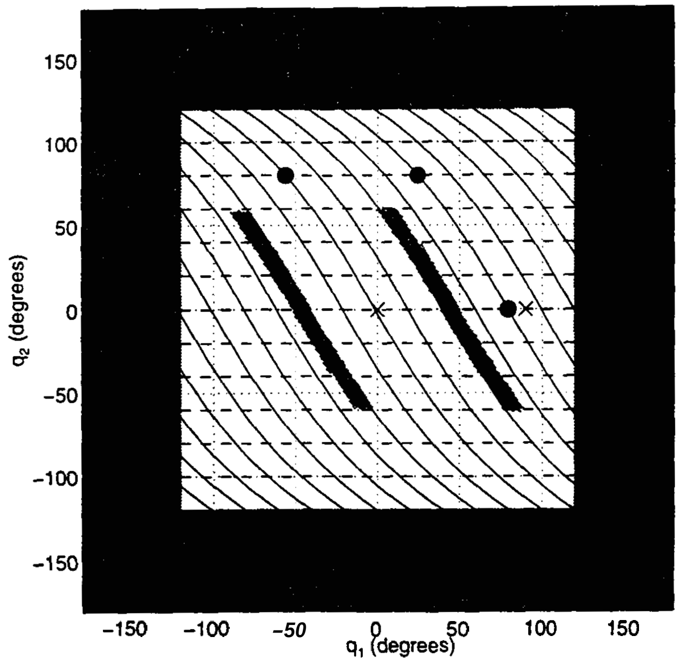


Figure 5.5: The via points of the collision-free trajectory lie on the intersection of **A** and ***P*** surfaces.

rectangular grid a cost which is equal to the number of switchings (between ***A***, *P*, or ***AP*** motions) necessary to reach the goal configuration if that square were the initial configuration; (3) search the grid for the trajectory with the minimum cost, i.e., with the least number of switchings; and **(4)** warp the resulting rectangular trajectory to the original grid of ***A*** and P surfaces. It is important to note that: (i) a graphical representation of the ***A*** and P surfaces is *not* required for this planner to work, only their implicit description as in Equation **(5.4)** is needed; (ii) step 1 has to be performed only once for every manipulator and for every set of obstacles; (iii) step ***2*** has to be performed only once for a given goal configuration; (iv) steps 3 **and 4** have to be performed only once for every start configuration.

Such an automatic planner can be extended to manipulators with more than two degrees-of-freedom. The **A**, ***P***, and ***AP*** surfaces are mapped to an n-dimensional grid (implemented as an n-tree), and the search for a trajectory with minimum cost is performed in this grid. The complexity of the planner is exponential on the number of joints, and is of the order of $g^n 2^{n_a} \binom{n}{n_a}$, where g is the discretization of the rectangular grid. The complexity

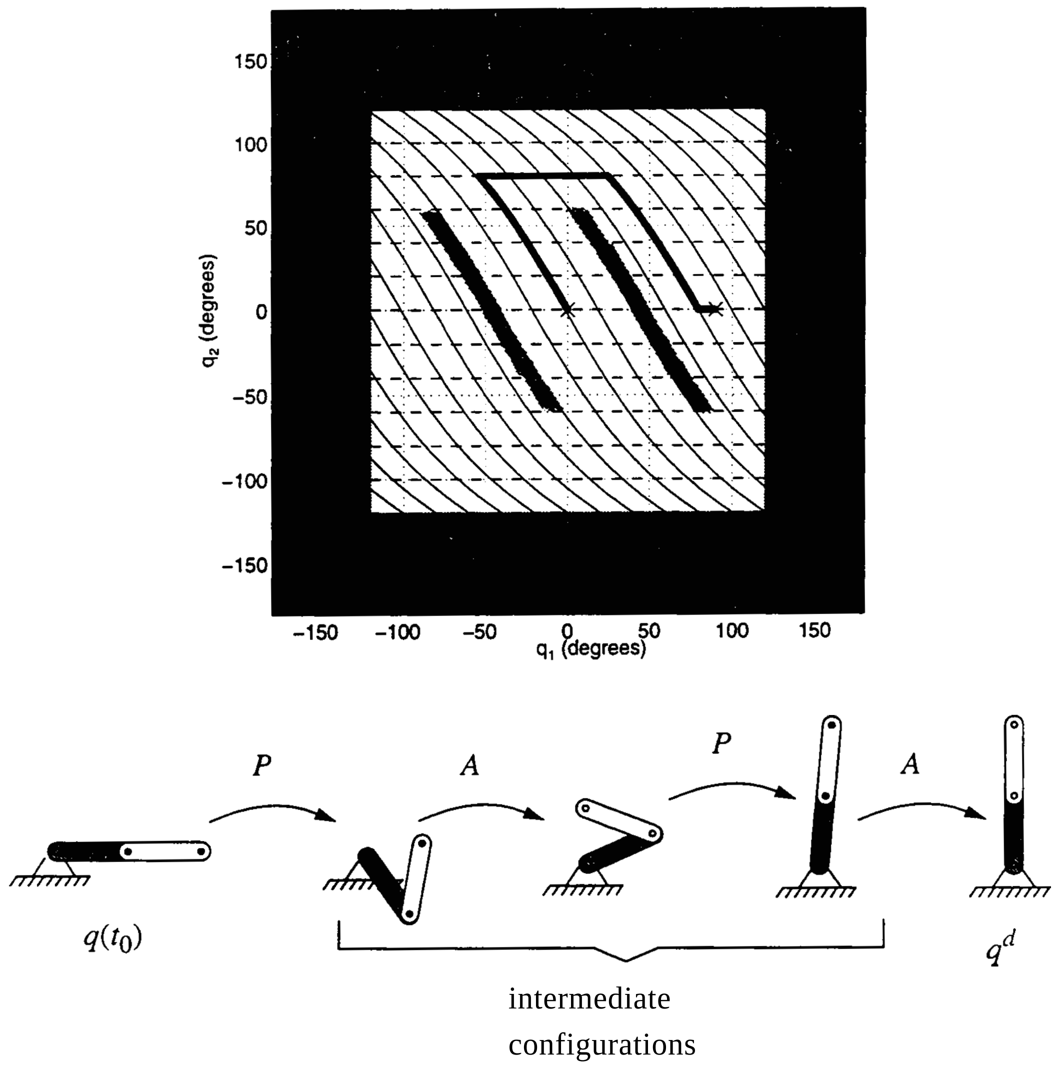


Figure *5.6:* Collision-free trajectory for the 2-link manipulator. The bottom drawings show the configuration of the manipulator during the motion. The letters indicate the type of motion performed between configurations.

is computed as follows: from every configuration (corresponding to a hypercube in the n-dimensional grid) we can control n_a out of the n existing joints (thus the term $\binom{n}{n_a}$), and each of these joints can be controlled in two directions (positive and negative, thus the term 2^{n_a}). Finally, we have to repeat the process for all g^n configurations.

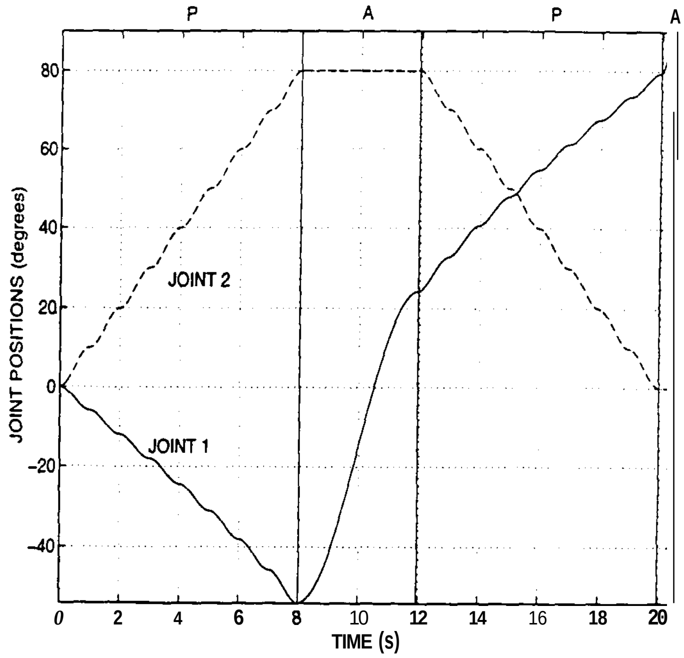


Figure 5.7: Joint angles history corresponding to the CCS motions in Figure 5.6.

5.4 Experiment

We implemented the foregoing obstacle avoidance method on the 3-link UARM described in Chapter **4.** By keeping the third joint locked at its extended position, we were able to simulate the behavior of a 2-link underactuated manipulator. In this condition, the kinematic and dynamic parameters of the **UARM** are given in Table 5.1.

In the experiment, we adopted a finer, 5° discretization of the CCS for the A surfaces. The discretization of the P surfaces was kept at 20°. To overcome the dry friction in the actuator, we adopted, for the P motions, an average speed three times as large as we did for the simulation study, i.e., $30^{\circ}/s$. From Figure 5.1 one can see that, at this average speed, we can command the manipulator to follow a 10° displacement without incurring the value of the sensitivity S to grow above 0.001.

Figure 5.8 presents a zoomed version of the actual *CCS* trajectory executed on the UARM. Both ***P*** motions consist of several 10° displacements at 30°/s, following 5th order polynomial trajectories. Note that, because of the finer discretization of the ***A*** surfaces, the manipulator had to fold the second link to 65° during the first part of the motion, compared to 80° in the simulation.

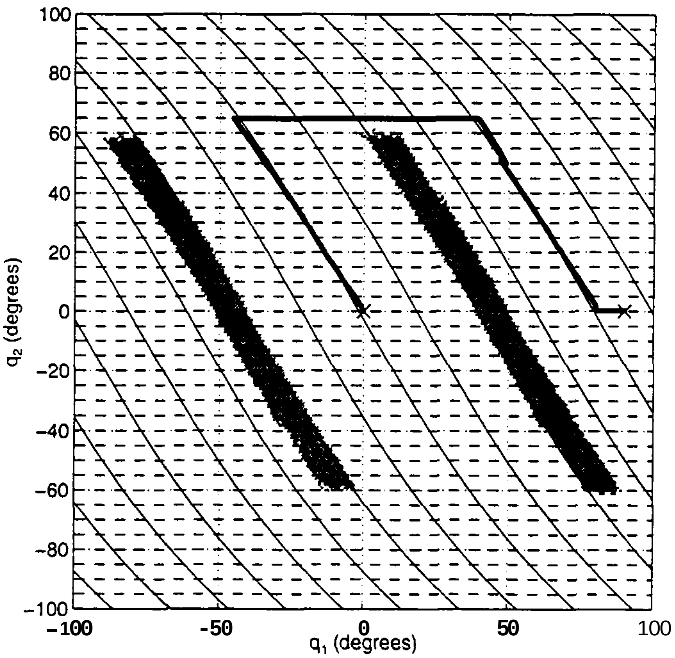


Figure 5.8: Experimental evaluation of the collision-free planning method on the UARM.

Recall that the 1st-order nonholonomic constraints (5.2) are integrable to holonomic constraints of the form $g(q, t) = 0$ if $M_u(q)\dot{q}$ defines an involutive distribution. This condition is always true when $n_a = 1$, i.e., when the manipulator has only one actuator. In fact, for the 2-link manipulator with joint 1 active and joint **2** passive, Equation (5.5) is equivalent to:

$$q_1(t) - q_1(t_0) = a\{f[a, b, q_2(t_0)] - f[a, b, q_2(t)]\} \tag{5.6}$$

where $a = m_2 l_{c_2}^2 + I$, $b = m_2 l_1 l_{c_2}$, and $f(a, b, \theta) = \frac{2}{\sqrt{a^2 - b^2}} \operatorname{atan}\left[\frac{\sqrt{a^2 - b^2}}{a + b} \tan\left(\frac{\theta}{2}\right)\right]$.

We did not use (5.6) to characterize the CCS trajectories of the 2-link manipulator studied for two reasons. First, one cannot assume that, in general, $M_u(q)\dot{q}$ defines an involutive distribution. Approximation (5.2), however, is valid for *any* manipulator whose sensitivity S is sufficiently small. Second, even when $M_u(q)\dot{q}$ does define an involutive distribution, obtaining a closed-form solution like (5.6) is'time-consuming and may require resorting to the Frobenius theorem (except for simple cases). It is important to note that, even when a closed-form is obtained, the underlying assumption that S must be kept small along the trajectory *cannot* be ignored.

Chapter 6

Conclusion

6.1 Contributions

In this dissertation, we present a complete study of the fundamental characteristics of underactuated manipulators, and develop methodologies for modeling, controlling, and planning the motion of these systems. The original contributions of this dissertation are summarized below.

Modeling

We present a complete study of the kinematic and dynamic properties of underactuated manipulators, including:

- a general formulation of their dexterity, applied to three important dexterity measures; and an optimization method to determine the passive joints' locking angles which maximize the manipulator's dexterity;
- the development of the coupling index, to measure the dynamic coupling between the active and the passive joints; the coupling index was utilized for the analysis, design, and control of underactuated manipulators;
- a proof of the controllability of the passive joints over the entire state space of these joints' positions and velocities; and a proof that the concepts of dynamic

coupling and controllability are strongly related, for a nonzero coupling index implies controllability of the passive joints.

Control

We developed robust and optimal sequential control methods for underactuated manipulators, including:

- an enumeration of all three control strategies that can be utilized for the control of a subset of the manipulators' joints;
- utilization of a variable structure controller to control a subset of the generalized coordinates (either joint positions or Cartesian coordinates) to converge to an equilibrium position or to follow a time-varying trajectory;
- automatic generation of the control sequence of the passive joints for manipulators with more passive than active joints, via the maximization of the coupling index and consequent minimization of energy consumption.

Planning

We proposed and developed the following important concept and method:

- we proposed the concept of the constrained configuration space to describe the space of the manipulator's configurations when constrained by nonintegrable differential equations;
- we developed a graphical method to plan the collision-free motion of an underactuated manipulator inside its constrained configuration space. The method includes: integration of the second-order nonholonomic constraints to first-order ones; characterization of the manipulator's motion via the numerical integration of the first-order constraints; generation of collision-free trajectories for each control strategy; and switching among the control strategies to generate an overall collision-free trajectory.

Experiments

We built a planar underactuated manipulator with one active and two passive joints, and demonstrated the feasibility of the following original methodologies:

- sequential robust control of a 2-link underactuated manipulator to an equilibrium point, and sequential robust optimal control of a 3-link underactuated manipulator to an equilibrium point;
- collision-free motion of a 2-link underactuated manipulator among workspace obstacles to an equilibrium point.

6.2 Future work

In this dissertation, we laid out the foundations of the dynamics and control of underactuated manipulators. The theory developed can be extended to other types of underactuated systems, as long as the systems share some of the underactuated manipulators' basic characteristics. The following are possible extensions of this work:

- *control of teams of cooperative underactuated manipulators rigidly grasping a common object:* a team of robots of this kind may or may not be nonholonomic, depending on the total number of active joints in the system. When the number of DOFs of the task to be executed is smaller than the total number of active joints, one can apply the Cartesian space modeling presented in this dissertation and obtain an openloop relationship between the active torques and the accelerations of the object in Cartesian space. Consequently, the variable structure controller can be utilized for the stabilization of the object around an equilibrium point or a time-varying trajectory. Research efforts in this direction have already begun [24];
- *control of underactuated manipulators without brakes:* the lack of brakes makes the problem of controlling all joints of an underactuated manipulator to an equilibrium point more difficult, for the passive joints might not stay in a fixed position when other joints are being controlled. Current results are those of Arai [3], who demonstrated

the controllability of a 3-link manipulator with two actuators, and of Suzuki et al. [47], who demonstrated in practice control of 2-link manipulator with one actuator. The motion planning method proposed in this dissertation might be applicable to the control of underactuated manipulators without brakes, as long as the constant gravity requirement is satisfied. When the passive joints cannot be locked, the surfaces followed by the manipulator inside the configuration space are modified, but can still be followed accurately if the sensitivity of the kinetic energy with respect to the passive joints' positions is negligible;

- *control of a space manipulator system:* it has been shown that a regular or underactuated manipulator mounted on a free-floating satellite is dynamically equivalent to a fixed-base manipulator whose first joint is a passive spherical one not equipped with a brake [23]. Moreover, the second-order nonholonomic constraints imposed by this virtual passive joint are automatically integrable to first-order ones. Consequently, if one succeeds in applying our motion planning method to underactuated manipulators without brakes as explained above, then it will also be possible to apply it to the control of space manipulator systems;
- *control of locomotion aids for the disabled:* artificial legs for disabled people with above-the-knee prosthesis are typically underactuated, for the mechanical knee is usually composed of passive elements. If one designs an artificial leg with a brake and provides it with simple sensory information, it might be possible to control the knee to generate a more natural stance and enhance the walking comfort of the disabled person;
- *control of large underactuated machines:* a crane is another typical underactuated system, for the angle of the cable which supports the load is not directly controllable. These machines are usually made to follow slow trajectories so that the cable does not oscillate beyond control. The application of model-based methods to the control of a crane might lead to rapid and precise automatic manipulation of, for example, containers to and from cargo ships, consequently minimizing loading time and costs.

Chapter 7

References

[1] H. Arai and S. Tachi, "Position control of a manipulator with passive joints using dynamic coupling." *IEEE Transactions on Robotics and Automation*, vol. **7,** no. **4,** Aug. 1991, pp. **528-534.**

[2] H. Arai, K. Tanie, and **S.** Tachi, "Dynamic control of a manipulator with passive joints in operational space." *ZEEE Transactions on Robotics and Automation,* vol. 9, no. 1, Feb. 1993, pp. 85-93.

[3] H. Arai, "Controllability of a 3-DOF manipulator with a passive joint under a nonholonomic constraint." *Proceedings of the I996 IEEE International Conference on Robotics and Automation,* vol. **4,** pp. 3707-37 13.

[4] N.S. Bedrossian, "Nonlinear control of an underactuated two-link manipulator." *Proceedings of the 1993 American Control Conference,* vol. **3,** pp. **2234-2238.**

[5] M.D. Berkemeier and **R.S.** Fearing, "Control experiments on an underactuated robot with application to legged locomotion." *Proceedings of the I994 IEEE International Conference on Robotics and Automation,* pp. 149-**154.**

[6] W.J. Chung, Y. Nakamura and O.J. Sørdalen, "Prototyping a nonholonomic manipulator." *Proceedings of the 1995 IEEE International Conference on Robotics and Automation,* pp. 2029-2036.

[7] J. J. Craig, *Introduction to robotics: mechanics and control.* Addison-Wesley, Reading, 1989, 2nd ed.

[8] S. Dubowsky and E. Papadopoulos, "The kinematics, dynamics, and control of free-flying and free-floating space robotic systems." *IEEE Transactions on Robotics and Automation,* vol 9, no. 5, Oct. 1993, pp. 531-543.

[9] C. Fernandes, L. Gurvits, and Z. Li, "Near-optimal nonholonomic motion planning for a system of coupled rigid bodies." *IEEE Transactions on Automatic Control,* vol. 39, no. 3, Mar. 1994, pp. 450-463.

[10] H.M. Frost, *An introduction to biomechanics.* C.C. Thomas, Springfield, 1967.

[11] C. Gosselin and J. Angeles, "A global performance index for the kinematic optimization of robotic manipulators." *Transactions of the ASME, Journal of Mechanical Design,* vol. 113, Sept. 1991, pp. 220-226.

[12] Y.-L. Gu and Y. Xu, "Under-actuated robot systems: dynamic interaction and adaptive control." *Proceedings of the 1994 IEEE Conference on Systems, Man and Cybernetics,* San Antonio, TX, USA, pp. 958-963.

[13] J. Hauser and R.M. Murray, "Nonlinear controllers for non-integrable systems: the Acrobot example." *Proceedings of the I990 American Control Conference,* pp. 669-671.

[14] J.Y. Hung, G. Weibing, and J.C. Hung, "Variable structure control: a survey." *IEEE Transactions on Industrial Electronics,* vol. 40, no. 1, Feb. 1993, pp. 2-22.

[15] **A.** Isidori, *Nonlinear control systems,* Springer Verlag, Berlin, 1995, 3rd ed.

[16] **A.** Jain and G. Rodriguez, "A spatial operator algebra for manipulator modeling and control.'' *The International Journal of Robotics Research*, vol. 10, Aug. 1991, **pp.** 371-381.

[17] **A.** Jain and G. Rodriguez, "An analysis of the kinematics and dynamics of underactuated manipulators." *IEEE Transactions on Robotics and Automation*, vol. 9, no. 4, **Aug.** 1993, pp. 411-422.

[18] E.M. Laassel, P. Loslever, and J.C. Angue, "Patterns of relations between lower limb angle excursions during normal gait." *Journal of Biomedical Engineering*, vol. 14, no. **4,** Jul. 1992, pp. 313-320.

[19] P. Lancaster and M. Tisrnenetsky, *The theory of matrices.* Academic Press, Orlando, 1985, 2nd ed.

[20] J.-C. Latombe, *Robot motion planning.* Kluwer Academic Publishers, Boston, 1991, **pp.** 431-437.

[21] C. Lee and Y. Xu, *Actuability of underactuated manipulators.* Technical Report CMU-TR-RI-94-13, Carnegie Mellon University, May 1994.

[22] F.L. Lewis, C.T. Abdallah, and D.M. Dawson, *Control of robot manipulators.* MacMillan Publishing Company, New York, 1993.

[23] B. Liang, **Y.** Xu, and M. Bergerrnan, *Mapping a space manipulator to a dynamically equivalent manipulator.* Technical Report CMU-RI-TR-96-33, Carnegie Mellon University, 1996.

[24] Y.-H. Liu and Y. Xu, "Cooperation of multiple manipulators with passive joints." Submitted to the *1997 IEEE International Conference on Robotics and Automation.*

[25] T. Lozano Pérez, "A simple motion-planning algorithm for general robot manipulators." *IEEE Journal of Robotics and Automation*, vol. 3, no. 3, Jun. 1987, pp. 224-238.

[26] **A.A.** Maciejewski, "Fault tolerant properties of kinematically redundant manipulators." *Proceedings of the 1990 IEEE International Conference on Robotics and Automation,* pp. 638-642.

[27] Mathworks, Inc., *MATLAB: high-performance numeric computation and visualization software.* Natlick, Mass., vol. 1 (Matlab user's guide), 1992.

[28] D.R. Meldrum, G.F. Franklin, and P.J. Wiktor, "Control of manipulators with some unactuated joints." *Proceedings of the 12th World Congress of the International Federation of Automatic Control,* vol. 3, 1993, pp. 345-348.

[29] R. Mukherjee, "Reorientation of a structure in space using a 3-link manipulator." *Proceedings of the I993 International Conference on Intelligent Robots and Systems,* pp. 2079-2086.

[30] R. Mukherjee and D. Chen, "Control of free-flying underactuated space manipulators to equilibrium manifolds." *IEEE Transactions on Robotics and Automation,* vol. 9, no. 5, Oct. 1993, pp. 561-570.

[31] R.M. Murray and S.S. Sastry, "Nonholonomic motion planning: steering using sinusoids." *IEEE Transactions on Automatic Control,* vol. 38, no. 5, May 1993, pp. 700-716.

[32] K. Nakakuki, H. Fujimoto, and K. Yamafuji, "Motion control of a robot composed of three serial links **with** a curved contour." *Proc. of the 1993 JSME International Conference on Advanced Mechatronics,* pp. 804-809.

[33] **Y.** Nakamura, *Advanced robotics:* redundancy and optimization. Addison-Wesley, Reading, 1991.

[34] G. Oriolo and **Y.** Nakamura, "Free-joint manipulators: motion control under second-order nonholonomic constraints." *IEEE International Workshop on Intelligent Robots and Systems*, 1991, pp. 1248-1253.

[35] J. Ostrowski, **A.** Lewis, R. Murray, and J. Burdick, "Nonholonomic mechanics and locomotion: the snakeboard example." *Proceedings of the 1994 International Conference on Robotics and Automation*, 1994, pp. 2391-2397.

[36] E. Papadopoulos and **S.** Dubowsky, "Failure recovery control for space robotic systems." *American Control Conference*, vol. 2, 1991, pp. 1485-1490.

[37] C.J.J. Paredis and P.K. Khosla, "Global trajectory planning for fault tolerant manipulators." 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 2, 1995, pp. 428-434.

[38] C.J.J. Paredis **and** P.K. Khosla, "Mapping tasks into fault tolerant manipulators." *Proceedings of the I994 International Conference on Robotics and Automation*, pp. 696-703.

[39] **A.K.** Pradeep, P.J. Yoder, R. Mukundan, and R.J. Schilling, "Crippled motion in robots." *IEEE Transactions on Aerospace and Electronic Systems*, vol. 24, no. 1, Jan. 1988, pp. 2-13.

[40] R.G. Roberts, "Quantifying the local fault tolerance of a kinematically redundant manipulator." *Proceedings of the 1996 American Control Conference*, pp. 1889-1893.

[41] R.G. Roberts and **A.A.** Maciejewski, **"A** local measure of fault-tolerance for kinematically redundant manipulators." *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, **Aug.** 1996, pp. 543-552.

[42] E.J. Routh, *Dynamics of a system of rigid bodies*. Dover, New York, 1960, 7 ed, pp. 317.

[43] F. Saito, T. Fukuda, and F. Arai, "Swing and locomotion control for a two-link brachiation robot." *IEEE Control Systems Magazine,* Feb. 1994, pp. 5-12.

[44] O.J. Sørdalen and C.C. de Wit, "Exponential control law for a mobile robot: extension to path following." *Proceedings of the 1992 International Conference on Robotics and Automation,* 1992, pp. 2158-2163.

[45] M.W. Spong, "Swing up of control the Acrobot." *Proceedings of the 1994 International Conference on Robotics and Automation,* 1994, pp. 2356-2361.

[46] D. Stewart, R. Volpe, and P.K. Khosla. "Integration of real-time software modules for reconfigurable sensor-based control systems." *IEEE International Conference on Intelligent Robots and Systems,* pp. 325-332, 1992.

[47] T. Suzuki, M. Koinuma, and Y. Nakamura, "Chaos and nonlinear control of a nonholonomic free-joint manipulator.'' *Proceedings of the I996 IEEE International Conference on Robotics and Automation,* vol. *3*, **pp.** 2668-2675.

[48] S. Takashima, "Control of gymnast on a high bar." *IEEE/RSJ International Workshop on Intelligent Robots and Systems,* 1991, pp. 1424-1429.

[49] *Y.* Umetani and K. Yoshida, "Resolved motion rate control of space manipulators with generalized Jacobian matrix." *IEEE Transactions on Robotics and Automation,* vol. *5*, no. 3, Jun. 1993, pp. 303-314.

[50] C.C. de Wit, **A.D.** Ndoudi-Likoho, and A. Micaelli, "Feedback control for a train-like vehicle." *Proceedings* ***of*** *the 1994 International Conference on Robotics and Automation,* 1994, pp. 14-19.

[51] Y. Xu and H. **Y.** Shum, "Dynamic control and coupling of a free-flying space robot system." *Journal of Robotic Systems,* vol. 11, no. 7, Oct. 1994, pp. 573-589.

[52] M. Yamakita, K. Nonaka, and K. Furuta, "Swing up control of a double pendulum." *Proceedings of the 1993 American Control Conference*, vol. 3, pp. 2229-2233.

[53] T. Yoshikawa, *Foundations of robotics: analysis and control.* MIT Press, Cambridge, 1990.

[54] T. Yoshikawa and K. Hosoda, "Modeling of flexible manipulators using virtual rigid links and passive joints." *Proceedings of the 1991 International Workshop on Intelligent Robots and Systems*, pp. 967-972.

[55] K. Youcef-Toumi and H. Asada, "The design of open-loop manipulator arms with decoupled and configuration-invariant inertia tensors." *Transactions of the ASME, Journal of Dynamic Systems, Measurement, and Control*, vol. 109, Sep. 1987, pp. 268-275.

[56] K.-H. Yu, T. Takahashi, and H. Inooka, "Dynamics and motion control of a two-link robot manipulator with a passive joint." *Proceedings of the 1995 International Conference on Intelligent Robots and Systems*, vol. 2, pp. 311-316.

Appendix A

Differential geometry definitions and results

The study of the controllability of nonlinear systems usually requires tools and results from the field of differential geometry. In our study of the controllability of underactuated manipulators, we refer to such results, which are here summarized for the reader's convenience. **All** definitions and the theorem in this appendix are based on Isidori's book [15].

Definition A.1: **A** vector field on an open subset U of $\Re^n$ is a smooth function mapping U into $\Re^n$. The Lie Bracket of two vector fields f and g is the vector field defined by

$$[f,g] = \frac{\partial g}{\partial x} \cdot f(x) - \frac{\partial f}{\partial x} \cdot g(x) \tag{A.1}$$

Iterated Lie brackets of the form $[f,[f,..[f,g]]]$ are denoted recursively as $ad_f^k g(x) = [f, ad_f^{k-1} g(x)]$ with $ad_f^0 g(x) = g(x)$.

Definition A.2: Let $\lambda(x)$ be a real-valued function and f a vector field, both defined on a subset U of $\Re^n$. The derivative of λ along f, denoted by $L_f\lambda$, is defined by

$$L_f\lambda = \frac{\partial\lambda}{\partial x}f(x) = \sum_{i=1}^{n}\frac{\partial\lambda}{\partial x_i}f_i(x) \tag{A.2}$$

Repeated derivatives of λ along f are denoted by $L_f^k\lambda(\mathbf{x})$, where $L_f^0\lambda(\mathbf{x}) = \lambda(\mathbf{x})$

Definition A.3: A k-dimensional distribution **A** on an open subset X of $\Re^n$ is a map which assigns a k-dimensional subspace of $\Re^n$ to each $x \in$ X, provided that, for each $x_0 \in X$, there exists an open set $U \subseteq X$ containing x_0 and k vector fields $f_1, \ldots, \mathbf{f}_k$ such that (i) $\{f_1(x), \ldots, f_k(x)\}$ is a linearly independent set for every $x \in U$, and (ii) $\Delta(x) = \mathrm{span}\{f_1(x), \ldots, f_k(x)\}$, $\forall x \in U$.

Definition A.4: A distribution **A** is involutive if $[f, g] \in$ **A** whenever the vector fields $f, g \in \Delta$.

Definition AS: A distribution **A** is invariant under the vector field f, or f-invariant, if $[f, h] \in$ **A**, $\forall h \in$ **A**.

Definition A.6: Suppose that **A** is defined as $\Delta(x) = \mathrm{span}\{f_1(x), \ldots, f_k(x)\}$. **A** point x is a *regular* ***point*** of **A** if $\Delta(x)$ is actually a k-dimensional subspace of $\Re^n$ (or, in other words, if the rank of the matrix $[f_1(x), \ldots, f_k(x)]$ is constant and equal to ***k).***

Definition A.7: A multivariable nonlinear system of the form

$$\begin{aligned} \dot{x} &= f(x) + \sum_{i=1}^{m} u_i g_i(x) = f(x) + g(x)u \\ Y_, &= h_1(x) \\ &\ldots \\ y_m &= h_m(x) \end{aligned} \tag{A.3}$$

has a vector *relative degree* $\{r_1, \ldots, r_m\}$ at a point x_0 if (i) $L_{g_j}L_f^k h_i(x) = 0$ for all $1 \le j \le m$, for all $k < r_i - 1$ for all $1 \le i \le m$, and for all x in a neighborhood of x_0; and (ii) the $m \times m$ matrix $A(x)$ defined by $[a_{ij}] = L_{g_j}L_f^{r_i - 1}h_i(x)$ is nonsingular at $x = x_0$.

State-space Exact Linearization Problem: Given a set of vector fields f and $g_1, \ldots, g_m$ (respectively known as *drift* and *control* vector fields) associated with the nonlinear system (A.3), and an initial state x_0, find, if possible, a neighborhood U of x_0, a pair of feedback functions $\alpha(x)$ and $\beta(x)$ defined on U, a coordinates transformation $z = \Phi(x)$ also defined on U, a matrix $A \in \Re^{n \times n}$ and a matrix $B \in \Re^{n \times m}$ such that the pair (A, B) is controllable and:

$$\left[\frac{\partial \Phi}{\partial x}(f(x) + g(x)\alpha(x))\right]\Bigg|_{x = \Phi^{-1}(z)} = Az \tag{A.4}$$

$$\left[\frac{\partial \Phi}{\partial x}(g(x)\beta(x))\right]\Bigg|_{x = \Phi^{-1}(z)} = B \tag{A.5}$$

Theorem A.1: Suppose that the matrix $g(x_0)$ in **(A.3)** has rank m. Then, the state-space exact linearization problem is solvable if, and only if, (i) for each $0 \le i \le n-1$, x_0 is a regular point of the distributions $\Delta_i = \mathrm{span}\{ad_f^k g_j : 0 \le k \le i, 1 \le j \le m\}$; (ii) the distribution Δ_{n-1} has dimension n; and (iii) for each $0 \le i \le n-2$, the distribution Δ_i is involutive.

Appendix B

Simulation Environment

To test the feasibility of the proposed control methods, we created a simulation environment that allows us to study planar underactuated manipulators with rotary joints. The simulator allows the user to assign an actuator or brake to each joint independently, as well as to change the dynamic and kinematic parameters of the links. The user may also introduce arbitrary parameter uncertainties to test the robustness of any desired controller (although the current version supports only the variable structure controller proposed in this dissertation). Figure B.1 presents a snapshot of the simulation environment, developed with Matlab [27].

The graphical interface is divided into the following areas (the following area names are identical to the uicontrol's names used in Matlab; for more information on Matlab and uicontrol's, see [27]):

msg_frame: holds all the buttons used to enter commands, all messages, and a prompt for entering numeric data;

movie–axes: the main axes where the simulation is displayed;

q_axes, coupling–axes, tau–axes: axes used to display, respectively, the joint angles, the coupling index, and the joint torques;

The *msg_frame* uicontrol is further sub-divided into the following areas:

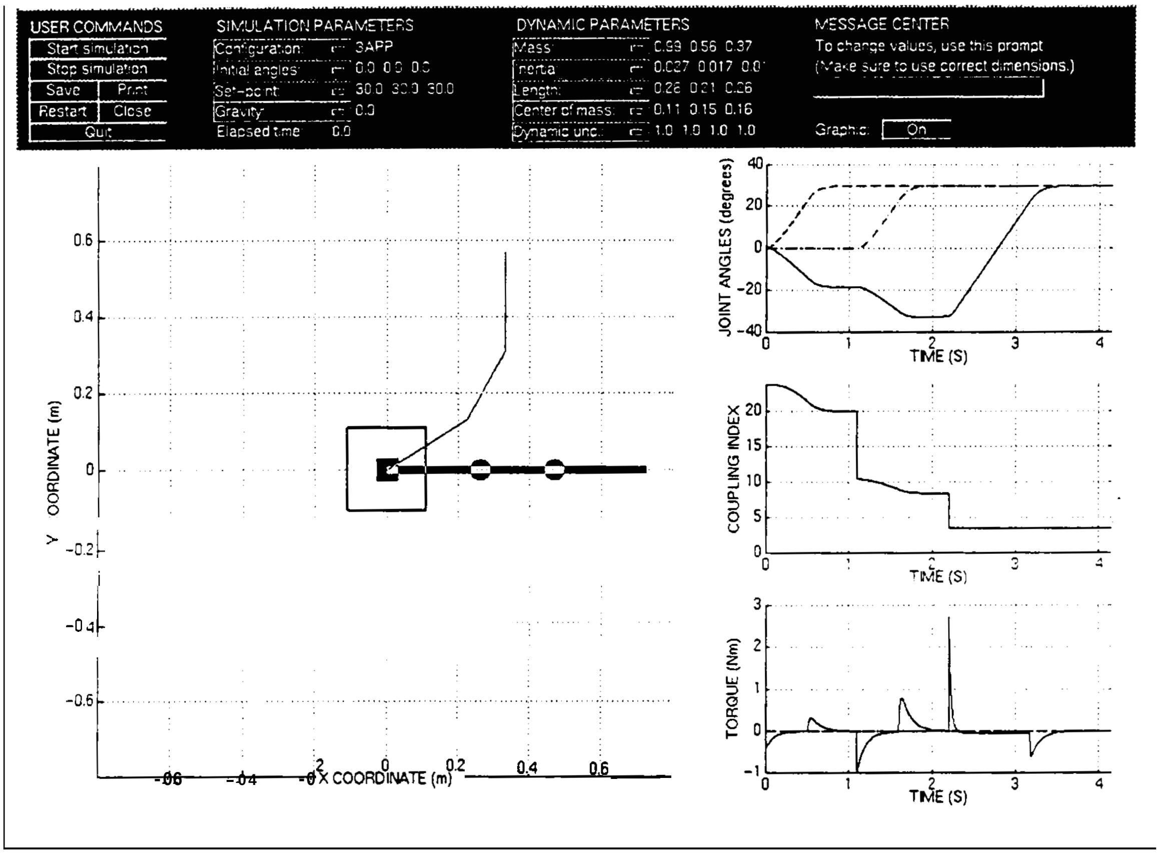


Figure B.1: Underactuated manipulator graphical user interface.

User Commands: the buttons here perform the following tasks:

Start Simulation: starts the simulation. Because this button is interruptible (so that one can stop the simulation at any time), we made it become invisible during the simulation. It becomes visible again when the simulation ends or when the user presses the *Stop Simulation* button. **If** anything goes wrong during the simulation (such as a failed matrix inversion), this button will not return to the visible state. In this case, the user may simply press *Stop Simulation* and the button will show up;

Stop Simulation: stops the simulation at any point, and returns the *Start Simulation* button to the visible state;

Save: saves the variables 'time', **'q'** (joint angles), 'qd' (joint velocities), 'q_d' (desired joint angles), and ‘stat’ (simulation statistics) in the file ‘data.mat’. Each variable is appended with ‘save’ before it is actually saved (e.g., ‘time’ is saved as ‘timesave’);

Print: prints a snapshot of all four axes (defined above) in the encapsulated postscript file ‘uarm.eps’;

Restart: restarts the graphical interface by closing it and then recalling it. Restart is *to* be used when changes are made to any of the buttons, messages, axes, etc;

Close: closes the graphical interface and cleans Matlab’s workspace of all variables;

Quit: closes the graphical interface and exits Matlab;

Simulation Parameters: this area displays the parameters that define the simulation. The following data is displayed and can be changed:

Configuration: defines the number of links (2 or 3), the number of active and passive joints, and their location. To change the configuration, click on the button, — do not release it — and choose the desired configuration. If you press and release the button, you have to wait until the manipulator is redisplayed, or Matlab will not change the configuration; then choose the desired configuration. Possible configurations are defined by the number of links, and the type of joints. For example, 2AP represents a 2-link manipulator with the first joint active and the second passive; 3PAP represents a 3-link manipulator with the first and third joints passive, and the second active;

Initial angles: defines the initial angles for the manipulator. To choose the default initial conditions, click on *default;* to choose randomly selected angles, click on *random.* To choose your own initial angles, first enter them as an $n \times 1$ matrix in the prompt provided at the message center (for example, for three links, enter *[23.0 34.0 –5.0]*), click on *Initial angles* and choose *User defined.* Note that angles must be entered in *degrees,* not in radians. If you enter incorrect data (for example, if the size of the matrix entered is not $n \times 1$, or if you forget the brackets) the message "Invalid data! Default values set" will appear below the prompt and default values will be assigned to the initial conditions;

Set-point: defines the set-point for the joint angles. To change the set-point, follow the procedures given above for changing the initial angles;

Gravity: defines the gravitational acceleration in m/s^2. It is a scalar defining the gravitational pull in the y axis. To choose a particular value, enter it in the prompt located at the message center, then click on *Gravity* and choose *User defined;*

Elapsed time: displays the simulated time as it progresses. The wall clock time elapsed is displayed on Matlab's command window after the simulation is over;

Dynamic parameters: this area displays the parameters that define the manipulator. The default values are those of the **UARM.** The following data is displayed and can be changed:

Mass: defines the masses of the links in ***Kg.*** To change the masses, follow the procedures given above for changing the initial angles;

Inertia: defines the inertias of the links in $Kg\,.m^2$. Because we are dealing only with planar manipulators, each link's inertia is a scalar. To change the inertias, follow the procedures given above for changing the initial angles;

Length: defines the lengths of the links in *m.* To change the lengths, follow the procedures given above for changing the initial angles. Note that, when you change the lengths, the centers-of-mass of the links are automatically defined to be half of the links' lengths;

Center of mass: defines the centers of mass of the links in *m*. To change the centers of mass, follow the procedures given above for changing the initial angles. If you attempt to define a center-of-mass that is greater than a particular link's length, the message "Invalid data! Default values set" will appear below the prompt and default values will be assigned to the centers of mass of all links;

Dynamic unc: defines the degree of uncertainty δ on the dynamic and kinematic parameters. It must be a 1×4 vector; if the user enters a scalar, that number will be converted into a vector with identical entries. Use this feature to test the robustness of a particular control law with respect to parameter uncertainty. The dynamic uncertainty *(du)* multiplies the manipulator's nominal parameters to generate estimates that are used for control purposes (the manipulator is always simulated with the nominal parameters), according to the following formulas:

$$\begin{aligned} \hat{m} &= du(1)m \\ \hat{l} &= du(2)l \\ \hat{l}_c &= du(3)l_c \\ \hat{I} &= du(4)I \end{aligned} \tag{B.1}$$

Message Center: provides a prompt for entering numeric data (as explained above). It also contains a button that allows toggling (on/off) the display of the manipulator's movement. The current state of the button is equal to its string. For example, if the button reads 'on,' then graphical display is enabled (and the simulation runs slower). When it reads 'off,' the simulation runs faster, but you do not get to see the manipulator moving.

The controller gains cannot be directly modified using the graphical interface. They can be modified by editing the file *uarm_gains.m.* This file is automatically loaded every time the simulation is started.

After choosing all values for a particular simulation, press *Start simulation* and you will see the manipulator performing its movement towards the set-point (provided that *Graphic* is in the 'on' state). The simulation ends when any one of these conditions occur:

- all joints reach their set-points; the simulation runs for another 0.3 s and stop;
- the user presses *Stop simulation*; the simulation stops immediately;
- a numerical error occurs; the simulation stops immediately.

In the first two cases, -as soon as the simulation is over the *Srart simulation* button returns to the visible state, and the joint angles, coupling index, and joint torques will be plotted in their respective axes. In the third case, no plot will appear, and the user has to press *Stop simulation* to regain control of the *Start simulation* button.

In addition to choosing the initial angles and set-points as described above, the user may also directly choose a Cartesian point as the initial and final destination. Just click (with any mouse button) anywhere in the *movie_axes*, wait until the cursor changes to a cross-hair, and then click on the desired Cartesian point. If you use the first button, you choose the initial configuration; if you press the second or third button (or press any key), you choose the set-point.

For further information, or to receive a copy of all the files that compose the simulation environment, contact us at *mbergerm@cs.cniu.edu* or *xu+ @cs.cmri.edu.*

Appendix C

Author's Publications

The following is the complete list of publications derived from the work described in this dissertation:

[1] M. Bergerman and Y. Xu, "Robust joint and Cartesian control of underactuated manipulators." *Transactions of the* ***ASME****, Journul of Dynamic Systems, Measurement, and Control,* vol. 118, no. 3, Sep. 1996, pp. 557-565.

[2] M. Bergerman and Y. Xu, "Optimal control sequence of underactuated manipulators." *Proceedings of the 1996 IEEE International Conference on Robotics and Automation,* Minneapolis, MN, **USA,** vol. 4, Apr. 1996, pp. 3714-3719.

[3] M. Bergerman, C. Lee and Y. Xu, "A dynamic coupling index for underactuated manipulators." *Journal of Robotic Systems,* vol. 12, no. 10, Oct. 1995, pp. 693-707.

[4] M. Bergerman, C. Lee, and Y. Xu, "Dynamic coupling of underactuated manipulators." *Proceedings of the 4th IEEE Conference on Control Applications,* Albany, NY, USA, Sep. 1995, pp. 500-505.

[5] M. Bergerman, C. Lee and Y. Xu, "Experimental study of an underactuated manipulator." *IEEE International Conference on Intelligent Robots and Systems,* Pittsburgh, PA, USA, vol. 2, Aug. 1995, pp. 317-322.

[6] M. Bergerman and **Y.** Xu, "Robust control of underactuated manipulators: analysis and implementation." *Proceedings of the I994 IEEE International Conference on Systems, Man and Cybernetics,* San Antonio, TX, USA, October 1994, pp. 925-930.