

# Smooth Time Optimal Trajectory Planning for Industrial Manipulators

by

Daniela Constantinescu

M.Sc., Transilvania University, Brasov, Romania, 1995

B.Sc., University of Brasov, Romania, 1986

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF  
THE REQUIREMENTS FOR THE DEGREE OF  
MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

(Department of Mechanical Engineering)

We accept this thesis as conforming  
to the required standard

.....  
.....  
.....  
.....

THE UNIVERSITY OF BRITISH COLUMBIA

September 1998

© Daniela Constantinescu, 1998

# Abstract

This work proposes and demonstrates a strategy for planning smooth path-constrained time-optimal trajectories for manipulators. Such trajectories are obtained by limiting the actuator jerks required by the planned motion.

Existing planning strategies incorporate the smoothness requirement either as smoothness of the actuator torques or as smoothness of the joint trajectories. The smoothness requirement is desirable for reducing strain on robot actuators while still requiring low cycle times. In this work, the trajectory smoothness is defined in the phase plane and the planning observes the limits on the actuator jerks.

The solution proposed for determining the optimal trajectories consists of approximating the time optimal control problem by a nonlinear parameter optimization problem which is solved using the flexible tolerance method. It is shown that the approximate solution converges to the time optimal motion when the actuator jerks become very high.

A number of simulations are performed to demonstrate the proposed strategy. These simulations show that actuator jerk limits have a negative impact on robot motion time, but they do not give any indication about robot trajectory feasibility. This aspect is studied through further simulations and experiments on an industrial robot. The results of this work show that the tracking accuracy is directly related to the actuator jerk limits. Therefore, it is necessary to impose such limits when planning feasible optimal trajectories.

Finally, the performance of the smooth time optimal motion is compared to the performance of both the non-jerk limited optimal trajectory, as well as a smooth quintic trajectory. For similar actuator jerks and controller effort, the smooth path-constrained time-optimal trajectory results in

a significantly shorter motion time with nearly the same tracking accuracy as a quintic polynomial. Based on the results in this work, actuator jerk limits are shown to provide an improved method of achieving a compromise between high tracking accuracy, smooth joint behaviour, and optimal motion time.

# Contents

<b>Abstract</b>	<b>ii</b>
<b>Table of Contents</b>	<b>iv</b>
<b>List of Tables</b>	<b>vii</b>
<b>List of Figures</b>	<b>viii</b>
<b>Glossary</b>	<b>xi</b>
<b>Acknowledgements</b>	<b>xiv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Time-Optimal Motion . . . . .	3
1.3 Industrial Motivation . . . . .	5
1.4 Research Objectives . . . . .	6
1.5 Thesis Outline . . . . .	6
<b>2 Problem Formulation</b>	<b>8</b>
2.1 Path-Constrained Time-Optimal Motion . . . . .	8
2.1.1 Characteristics . . . . .	8
2.1.2 Practical Implementation Issues . . . . .	13
2.2 Research Addressing the Issues in PCTOM . . . . .	13
2.3 Smooth Path-Constrained Time-Optimal Control . . . . .	17
2.3.1 Problem Formulation . . . . .	18

---

2.3.2	Path Constraints . . . . .	21
2.3.3	Torque Limits . . . . .	23
2.3.4	Jerk Limits . . . . .	26
2.3.5	Admissible States . . . . .	28
2.3.6	System Dynamics . . . . .	31
2.4	Summary . . . . .	33
<b>3</b>	<b>Methodology</b>	<b>34</b>
3.1	SPCTOM Trajectory Characteristics . . . . .	34
3.2	Approximating Functions in Robotics . . . . .	37
3.3	Proposed Method . . . . .	40
3.3.1	Basis Functions . . . . .	40
3.3.2	Optimization Variables . . . . .	41
3.3.3	Numerical Solution . . . . .	45
3.4	Summary . . . . .	48
<b>4</b>	<b>Simulations</b>	<b>49</b>
4.1	SPCTOM Planning . . . . .	50
4.1.1	Parabolic Path . . . . .	50
4.1.2	Circular Path . . . . .	54
4.2	Discussion . . . . .	60
4.3	SPTOM Performance Investigation . . . . .	61
4.4	Summary . . . . .	70
<b>5</b>	<b>Experiments</b>	<b>72</b>
5.1	Experimental Setup . . . . .	72
5.2	Experimental Results . . . . .	73
5.3	Summary . . . . .	77
<b>6</b>	<b>Conclusions and Recommendations</b>	<b>78</b>
6.1	Summary . . . . .	78
6.2	Recommendations . . . . .	79

---

<b>Bibliography</b>	<b>81</b>
<b>Appendices</b>	<b>86</b>
<b>A The Flexible Tolerance Method</b>	<b>86</b>
A.1 Overview of the FTM . . . . .	86
A.2 Overview of the FPM . . . . .	88
A.3 Implementation Details . . . . .	89
<b>B Robot Dynamic Models</b>	<b>95</b>
B.1 Elbow Manipulator Dynamic Model . . . . .	96
B.2 SCORBOT ER VII Dynamic Model . . . . .	97

# List of Tables

2.1	Robot parameters . . . . .	12
2.2	Actuator parameters . . . . .	28
4.1	Robot parameters (repeated). . . . .	50
4.2	Actuator parameters (repeated). . . . .	50
4.3	Location of the PCTOM switching points in the $s - \dot{s}$ plane. . . . .	51
4.4	Optimal motion times for the elbow manipulator along the parabolic path. . . . .	52
4.5	Optimal motion times for the elbow manipulator along the circular path. . . . .	56
4.6	Optimal motion times for the elbow manipulator along the circular path when additional knot points are inserted. . . . .	58
4.7	PCTOM and SPCTOM motion times for the examples in Sections 4.1.1 and 4.1.2. . . . .	60
4.8	The SCORBOT ER VII Denavit-Hartenberg parameters. . . . .	61
4.9	SCORBOT ER VII estimated masses and inertias. . . . .	62
4.10	Actuator torque and jerk bounds for the SCORBOT ER VII. . . . .	62
4.11	PCTOM, SPCTOM, and quintic trajectory motion times for the SCORBOT ER VII along the straight line path. . . . .	64
5.1	Experimental results for the PCTOM, SPCTOM, and quintic trajectories. . . . .	75

# List of Figures

2.1	The elbow manipulator. . . . .	10
2.2	PCTOM plotted in the $s - \dot{s}$ phase plane. . . . .	11
2.3	PCTOM trajectory in the $s - \ddot{s}$ phase plane. . . . .	11
2.4	Actuator torques for the PCTOM trajectory. . . . .	12
2.5	Admissible region in the $\dot{s}^2 - \ddot{s}$ plane. . . . .	24
2.6	Torque limits in the $s - \dot{s} - \ddot{s}$ space and the VLC. . . . .	25
2.7	Admissible region in the $\ddot{s} - \dddot{s}$ plane. . . . .	26
2.8	Control constraints in the $s - \dot{s} - \ddot{s}$ space. . . . .	28
2.9	Different jerk limits in the $s - \dot{s} - \ddot{s}$ space. . . . .	29
2.10	Admissible states in the $s - \dot{s} - \ddot{s}$ space. . . . .	30
2.11	SMVLC for different actuator jerk limits. . . . .	32
3.1	Influence of the end slopes and the height of the trajectory on the motion time. . . . .	37
3.2	Sample trajectories in the $s - \dot{s}$ plane and their motion times. . . . .	41
3.3	Switching points of the PCTOM trajectory. . . . .	42
3.4	The optimization variables. . . . .	43
3.5	Motion time and its level curves with respect to several parameters varied over their ranges. . . . .	44
3.6	Plot of a phase plane trajectory resulting from a point in the inadmissible region of the parameter space. . . . .	44
3.7	The parameter space in the Flexible Tolerance Method. . . . .	47
4.1	PCTOM for the elbow manipulator along the parabolic path. . . . .	52

4.2	SPCTOM for the elbow manipulator along the parabolic path - Example 1 (medium jerk limits). . . . .	53
4.3	SPCTOM for the elbow manipulator along the parabolic path - Example 2 (large jerk limits). . . . .	53
4.4	SPCTOM for the elbow manipulator along the parabolic path - Example 3 (low jerk limits). . . . .	53
4.5	Actuator jerks for the SPCTOM in Example 2 and the PCTOM for the elbow manipulator along the parabolic path. . . . .	54
4.6	PCTOM in the $s - \dot{s}$ phase plane for the elbow manipulator along the circular path. . . . .	56
4.7	PCTOM for the elbow manipulator along the circular path. . . . .	56
4.8	SPCTOM for the elbow manipulator along the circular path - Example 1 (medium jerk limits). . . . .	57
4.9	SPCTOM for the elbow manipulator along the circular path - Example 2 (high jerk limits). . . . .	57
4.10	SPCTOM for the elbow manipulator along the circular path - Example 3 (low jerk limits). . . . .	57
4.11	Peak actuator jerks for the SPCTOM and the PCTOM for the elbow manipulator along the circular path. . . . .	58
4.12	SPCTOM for the elbow manipulator along the circular path (additional knot points inserted) - Example 1 (medium jerk limits). . . . .	59
4.13	SPCTOM for the elbow manipulator along the circular path (additional knot points inserted) - Example 2 (high jerk limits). . . . .	59
4.14	SPCTOM for the elbow manipulator along the circular path (additional knot points inserted) - Example 3 (low jerk limits). . . . .	59
4.15	PCTOM trajectory in the $s - \dot{s}$ phase plane for the SCORBOT ER VII along the linear path. . . . .	63
4.16	PCTOM for the SCORBOT ER VII along the linear path. . . . .	65
4.17	SPCTOM for the SCORBOT ER VII along the linear path (additional knot points inserted) - Example 1 (medium jerk limits). . . . .	65
4.18	SPCTOM for the SCORBOT ER VII along the linear path (additional knot points inserted) - Example 2 (high jerk limits). . . . .	65

4.19	SPCTOM for the SCORBOT ER VII along the linear path (additional knot points inserted) - Example 3 (low jerk limits).	66
4.20	Quintic trajectory for the SCORBOT ER VII along the linear path.	66
4.21	Controller tracking performance for the PCTOM, quintic, and SPCTOM trajectories.	67
4.22	Desired and simulated torques for the PCTOM trajectory.	67
4.23	Desired and simulated torques for the SPCTOM trajectory - Example 1 (medium jerk limits).	68
4.24	Desired and simulated torques for the SPCTOM trajectory - Example 2 (high jerk limits).	68
4.25	Desired and simulated torques for the SPCTOM trajectory - Example 3 (low jerk limits).	69
4.26	Desired and simulated torques for the quintic trajectory.	69
5.1	Experimental setup.	73
5.2	Experimental results for the PCTOM trajectory implemented on the SCORBOT ER VII.	74
5.3	Experimental results for the SPCTOM trajectory (Example 1 - jerk limits of 100Nm/sec) implemented on the SCORBOT ER VII.	74
5.4	Experimental results for the SPCTOM trajectory (Example 2 - jerk limits of 1000Nm/sec) implemented on the SCORBOT ER VII.	74
5.5	Experimental results for the SPCTOM trajectory (Example 3 - jerk limits of 10Nm/sec) implemented on the SCORBOT ER VII.	75
5.6	Experimental results for the quintic trajectory implemented on the SCORBOT ER VII.	75
5.7	Experimental results for the SPCTOM trajectory (jerk limits of 20Nm/sec) implemented on the SCORBOT ER VII.	76
5.8	Experimental results for the SPCTOM trajectory (jerk limits of 50Nm/sec) implemented on the SCORBOT ER VII.	77
A.1	The line searches during the replacement of the different types of infeasible points generated by the FPM.	92
A.2	Flow diagram of the FTM for the SPCTOM problem - Part 1.	93
A.3	Flow diagram of the FTM for the SPCTOM problem - Part 2.	94

# Glossary

$s$	The path parameter.
$\dot{s}$	The first time derivative of the path parameter, the pseudo-velocity.
$\ddot{s}$	The second time derivative of the path parameter, the pseudo-acceleration.
$\overset{\cdot\cdot}{s}$	The third time derivative of the path parameter, the pseudo-jerk.
$\mathbf{q}$	The column vector of joint variables.
$\dot{\mathbf{q}}$	The first time derivative of the joint positions, the vector of joint velocities.
$\ddot{\mathbf{q}}$	The second time derivative of the joint positions, the vector of joint accelerations.
$\overset{\cdot\cdot}{\mathbf{q}}$	The third time derivative of the joint positions, the vector of joint jerks.
$\mathbf{M}$	The inertia matrix of the manipulator.
$\mathbf{C}$	The Christoffel's symbols (centripetal and Coriolis coefficients).
$\mathbf{G}$	The vector of gravity terms.
$\mathbf{r}$	The Cartesian coordinates of the end-effector.
$\mathbf{T}$	The vector of actuator torques.
$\mathbf{T}_{min}$	The vector of minimum actuator torques.
$\mathbf{T}_{max}$	The vector of maximum actuator torques.
$\dot{\mathbf{T}}$	The first time derivative of the actuator torques, the vector of actuator jerks.
$\dot{\mathbf{T}}_{min}$	The vector of minimum actuator jerks.
$\dot{\mathbf{T}}_{max}$	The vector of maximum actuator jerks.
$\mathcal{A}$	The coefficients of the pseudo-acceleration in the inequalities expressing the actuator torque constraints in the phase space.

- $\mathcal{B}$**  The coefficients of the squared pseudo-velocity in the inequalities expressing the actuator torque constraints in the phase space.
- $\mathcal{C}$**  The free term in the inequalities expressing the actuator torque constraints in the phase space.
- $\mathbf{a}$**  The coefficients of the pseudo-jerk in the inequalities expressing the actuator jerk constraints in the phase space.
- $\mathbf{b}$**  The coefficients of the pseudo-velocity and pseudo-acceleration product in the inequalities expressing the actuator jerk constraints in the phase space.
- $\mathbf{c}$**  The coefficients of the third power of the pseudo-velocity in the inequalities expressing the actuator jerk constraints in the phase space.
- $\mathbf{d}$**  The coefficients of the pseudo-velocity in the inequalities expressing the actuator jerk constraints in the phase space.
- $\dot{s}_{max,T}$  Maximum admissible pseudo-velocity when only the actuator torque constraints are considered.
- $\ddot{s}_{min,T}$  Minimum admissible pseudo-acceleration when only the actuator torque constraints are considered.
- $\ddot{s}_{max,T}$  Maximum admissible pseudo-acceleration when only the actuator torque constraints are considered.
- $\dot{s}_{max,J}$  Maximum admissible pseudo-velocity when only the actuator jerk constraints are considered.
- $\ddot{s}_{min,J}$  Minimum admissible pseudo-acceleration when only the actuator jerk constraints are considered.
- $\ddot{s}_{max,J}$  Maximum admissible pseudo-acceleration when only the actuator jerk constraints are considered.
- $\ddot{\ddot{s}}_{min}$  Minimum admissible pseudo-jerk when the actuator jerk constraints are considered.
- $\ddot{\ddot{s}}_{max}$  Maximum admissible pseudo-jerk when the actuator jerk constraints are considered.

---

$f$	The optimization objective function (motion time).
$h_i$	The optimization equality constraints.
$g_i$	The optimization inequality constraints.
$\mathbf{x}$	A point in the parameter space.
$\Phi$	The flexible tolerance criterion.
$T$	The measure of the constraint violation.
TOC	Time Optimal Control.
OCT	Optimal Control Theory.
PMP	Pontryagin's Maximum Principle.
TPBVP	Two-Point Boundary Value Problem.
dof	Degrees of Freedom.
C.O.M.	Center of Mass.
DH	Denavit-Hartenberg.
PCTOM	Path-Constrained Time-Optimal Motion.
VLC	Velocity Limit Curve.
SPCTOM	Smooth Path-Constrained Time-Optimal Motion.
JL-VLC	Jerk-Limited Velocity Limit Curve.
SMVLC	Smooth Motion Velocity Limit Curve.
FTM	Flexible Tolerance Method.
FPM	Flexible Polyhedron Method.
DFP	Davidon-Fletcher-Powell.
ORTS	Open architecture Real-time Operating System.
LQG	Linear Quadratic Gaussian.
PID	Proportional Integral Derivative.
RMS	Root Mean Square.

# Acknowledgements

My thanks go to:

- My supervisor, Dr. Elizabeth Anne Croft, for her guidance, expertise and encouragement during the course of this work, and for never getting tired of proofreading this thesis.
- Dr. Beno Benhabib of the University of Toronto, for his invaluable discussions regarding this work.
- Kaan Erkorkmaz, for providing continuous support for the ORTS and always having an answer to my questions.
- Gord Wright, for making my experimental setup a reality.
- All my colleagues in the IAL and MAL labs at UBC, for their support and good humour.
- My husband and my family, for everything.

I also acknowledge the financial support of the Natural Sciences and Engineering Research Council of Canada and the Faculty of Graduate Studies at UBC.

# Chapter 1

## Introduction

### 1.1 Background

The Robotic Industries Association defines a robot as “a reprogrammable multi-functional manipulator designed to move materials, parts, tools, or specialized devices, through variable programmed motions for the performance of a variety of tasks” [3]. Both mobile robots and fixed base manipulators conform to this definition. In this work, however, the focus is on fixed base manipulators commonly found in industry, and herein described as “industrial robots”.

By the above definition, a robot may be decomposed into two parts: (i) a mechanism (namely, joints and links) together with actuators, capable of performing an infinite range of motions; and, (ii) computer algorithms together with a control and a sensing system, capable of planning and controlling these motions. Since the issues involved in motion control are complex, they are typically made more tractable by dividing them into *motion planning* problems and *motion tracking* problems.

Motion planning may be done either *on-line*, while the robot is executing the task, or *off-line*, that is, before the execution starts. On-line planning is often more desirable, since it increases the robustness of the robot’s behaviour in the face of uncertainties and modelling errors. However, off-line planning is used when the required planning algorithms are computationally intensive.

Depending on the specification of the task, motion planning entails either planning a trajectory along a specified path, or, planning both a path and a trajectory along the path. A *trajectory* is

a curve in state space that describes the system evolution in time. The trajectory incorporates the system dynamics. A *path* is a curve in Cartesian space (or in joint space) that describes the spatial evolution of the end-effector (or the joints). The path reflects the geometric constraints on the task, for example, the joint limits.

Robot motions are specified in two different ways. When the end-effector is constrained to a predetermined path, the motion is called *path-constrained*. Applications of path-constrained manipulation include welding, gluing and spraying. Since the path is imposed by the task, only trajectory planning is addressed.

When only the initial and final positions (and velocities) of the end-effector are imposed, the motion is called *point-to-point*. Pick and place operations are point-to-point manipulations. In this case both a path and a trajectory must be determined. Path planning is now subsumed by trajectory planning. However, this global problem is difficult due to the highly nonlinear and coupled dynamics of articulated robots. Most solutions are limited to two degree of freedom (dof) manipulators due to the exponential increase in complexity of the problem with increasing dof. Such solutions are generally considered only in off-line planning schemes when obstacles are not a concern. More often, however, path planning and trajectory planning are treated as independent problems and addressed separately. For example, this is the case when the trajectory is planned on-line or when the working environment is cluttered.

Once the trajectory is determined by the motion planner, it is sent to the tracking controller. This low level controller ensures that the robot's position and velocity match the reference signals. While many different advanced controllers have been proposed, most often in industrial practice a simple PID controller is used. Although such a controller does not consider the nonlinear robot dynamics, at high control loop frequencies it can keep the manipulator reasonably close to a typical industrial trajectory. This approach usually requires the robot to operate well under its full dynamic capabilities.

Thus, in the general context of motion planning, the present research is concerned with determining *industrially feasible* time-optimal trajectories for path-constrained tasks. These are trajectories which allow the robot to move in an optimal manner within the constraints of the robot actuators and controller capabilities.

## 1.2 Time-Optimal Motion

Since any manipulation can be performed in many ways, a desirable choice is to have the robot execute the motion in an optimal manner according to some relevant criterion. Optimality criteria for industrial manipulation tasks include: minimum energy consumption [50], [64], [65], [66]; minimum jerk [33], [34], [57]; and minimum effort [40]. However, the economic drive for maximum task productivity is so strong that much more research has been dedicated to time-optimal motions [5], [11], [14], [24], [26], [31], [38], [45], [54], [55], [56], [58], [59], [60].

Planning minimum time motions for industrial manipulators is a time optimal control (TOC) problem for a highly nonlinear and coupled multi-degree of freedom dynamic system. Thus, no explicit closed-form solution may be expected from Optimal Control Theory (OCT) based on the application of Pontryagin's Maximum Principle (PMP). For nonlinear systems, OCT only provides necessary conditions for optimality. Hence, only a set of candidate controls can be deduced using the general theory.

These difficulties stimulated the research in time-optimal motion planning. Different algorithms have been developed for solving the problem and much research has been concerned with studying the structure of the solution; see, for example, [5], [11], [12], [23], [41], [60], [61]. This research has shown that the optimal controls are either of the bang-bang or of the bang-singular-bang type. The singular controls are not directly characterized by PMP.

The first near-time optimal solution was proposed by Kahn and Roth [31] for point-to-point motions. They linearized the differential equations describing the dynamics of the robot, applied a transformation to decouple the linearized dynamics and then solved the TOC problem analytically. Hollerbach [29] showed that the coupling terms could not be neglected in the dynamics. That is, a manipulator is intrinsically nonlinear and coupled, and thus does not behave like a linear system slightly perturbed by nonlinearities.

Since then, methods for obtaining the time-optimal motion along a specified path have been reported by Bobrow et al. [5], and Shin and McKay [55]. Both these methods maximize the velocity of the manipulator along the given path and give the result in terms of a switching curve in the phase plane. Using essentially the same algorithm, Pfeiffer and Johanni [45] and Slotine and Yang [59] improved the efficiency of determining the switching points. Shiller and Lu [54] extended

the method to deal with singular controls.

For point-to-point manipulations, two approaches have been pursued. In the first approach, the TOC problem is solved globally, using either OCT or search techniques. In the former approach, PMP is applied to the Hamiltonian formulation of the robot dynamics [14], [13], [24], and the resulting two point boundary value problem (TPBVP) is solved by shooting methods. Alternatively, joint-space [47] or state-space [19], [26] search techniques are employed. In these methods, the space between the initial and final position (state) is tessellated and the linearized robot dynamics are used to compute the motion time between the tessellation points. The time-optimal trajectory is then found by a graph search. Regardless of the approach, the global TOC problem for point-to-point motions has been demonstrated only for simple two degrees of freedom manipulators in uncluttered environments.

In the second approach, path planning and trajectory planning are treated separately. The path is restricted to a class of suitably chosen functions under the assumption that the optimal path is smooth. The methods devised for determining the optimal path include: a joint space search technique [46]; a parameter optimization technique [4]; the use of the dynamic properties of the manipulator [52]. The minimum time trajectory along the path is then determined using the algorithm in [5]. This disjunction of the global problem greatly simplifies its solution. This simplification has been exploited to plan motions for realistic manipulators, with more than two degrees of freedom [4]. It has also been used in on-line planning schemes [16].

In all the above works, manipulation tasks are planned in a time-optimal fashion. Hence, all of them require bang-bang or bang-singular-bang actuator torques. Such torques cannot be achieved by physical robots. Thus, a need arises for smoothing the controls before such minimum time strategies can be practically implemented. This need has been recognized for more than 15 years [38], [64], and it has been addressed both at the planning level and at the tracking level. At the tracking level, trajectory preshaping [51] and filtering of the control signals [37] have been used to smooth the optimal controls required by the planned minimum time trajectories. At the planning level, actuator voltages as controls [63] and cost functions including actuator effort [8], [37], [50], [66] have been used to generate trajectories which result in smooth optimal controls. This is done by taking into account a more complete model of the manipulator dynamics. The resulting trajectory is then feasible for implementation by existing (non-specialized) industrial robot controllers.

In this work, smooth time optimal control is achieved through the second approach, namely, at the planning level. However, rather than require a more rigorous model of the robot (which is generally unavailable), constraints on the allowable actuator jerks are applied. Again, the objective is to generate optimal trajectories that are feasible for implementation on existing industrial manipulators.

### 1.3 Industrial Motivation

The research in time optimal robot control addresses the industry's demand for increased productivity. Whenever the process speed is limited by the robot itself, rather than by the task that it is performing, minimum time path tracking results in reduced cycle time. Shorter cycle times can increase productivity and, thus, decrease the cost per operation.

To achieve the goal of time optimal control, both the trajectory planner and the tracking controller are designed to exploit the full capability of the robot. This ensures minimum execution time for the task. However, the algorithms, both at the planning and at the tracking level, use only a simplified model of the manipulator. Link flexibility, actuator dynamics and joint friction are usually ignored and the robot dynamic model is considered to be accurately known. These modeling assumptions are required by the complexity of the motion planning problems. However, such assumptions are problematic in the implementation of time optimal control to real systems.

If implemented on a physical manipulator with a non-specialized controller, time optimal control will result in vibrations in the links and strain and wear on the actuators. Premature wear of the mechanical structure and frequent failures of the actuators can result [37]. Hence, in practical terms, the implementation of time optimal motion could decrease the return on capital investment in the robot. Furthermore, the productivity of the entire workcell may be reduced due to the down time for robot repairs. Such events defeat the purpose of time optimal control.

Thus, from an industrial point of view, the need arises for a more realistic optimal control: a control that achieves a desirable balance between the minimum task completion time and the practical limitations of the robot system.

## 1.4 Research Objectives

The view adopted herein is that increased productivity, namely shorter time for performing a task, is important from an economic perspective. Nonetheless, the physical limitations of the robot, such as link and joint flexibility, must be acknowledged in an industrial implementation. Therefore, the general objective herein is to determine *feasible* optimal trajectories for path-constrained tasks. The benefit of such trajectories is that, while being time-optimal, they will also be suited for direct implementation on a commercial robot using non-specialized industrial controllers.

In this context, the specific objectives are: (i) formulation of the TOC problem for path constrained tasks such that it includes constraints through which different degrees of trajectory smoothness can be imposed; (ii) development of an algorithm for solving this problem; and, (iii) establishing that this algorithm results in improved robot performance, as measured by the tracking accuracy.

## 1.5 Thesis Outline

This chapter introduced robotic motion planning and the time-optimal motion research. In summary, the practical limitations of minimum time motions led to the industrial motivation of the present work. The research objectives were presented and an outline of the thesis is given herein.

Chapter 2 discusses the issues involved in the physical implementation of path-constrained time-optimal motions (PCTOM). Then, existing methods of addressing these issues are presented. Finally, a new formulation of the PCTOM planning problem is presented, in which smoothness requirements on the motion are incorporated as limits on the rate of variation of the actuator torques.

In Chapter 3, the methodology for the solution of the proposed formulation is described. As the TOC problem is transformed into an optimization problem, this transformation is justified and detailed.

Simulations are performed using this optimization and the results are presented and discussed in Chapter 4.

---

An optimal trajectory planned using the algorithm developed in this work and a PCTOM trajectory planned according to the algorithm in [45] are implemented on the SCORBOT ER VII robot in the IAL laboratory. The experimental results and a comparison between robot performances along the two trajectories are given in Chapter 5.

Finally, Chapter 6 presents the conclusions of this research together with the recommendations for future work.

## Chapter 2

# Problem Formulation

This chapter discusses time-optimal control of manipulators along specified paths and highlights the issues arising in the practical implementation of such control. Existing methods of addressing these issues, both at the trajectory planning and at the trajectory tracking level, are presented. A new formulation for time-optimal control is then introduced and its advantages summarized.

### 2.1 Path-Constrained Time-Optimal Motion

Path-constrained time-optimal motions (PCTOM) are minimum time motions along a prespecified path. At the planning stage, a joint torques history is derived considering actuator torque limits, joint angle and joint velocity limits. The manipulator is modelled as a serial chain of rigid links, with each joint individually powered. The friction in the joints and the actuator dynamics are most often neglected. At the tracking level, a controller is designed that best can track the planned trajectory. Hence, the robot dynamic capabilities are used at the maximum in PCTOM.

#### 2.1.1 Characteristics

Path-constrained time-optimal trajectories have been found both by using dynamic programming [56], [58], or by identifying the switching points in the position-velocity phase plane [5], [45], [55]. The advantage of the dynamic programming approach is that the path need not be parame-

terized. Also, performance indices other than minimum time may be easily incorporated. However, the phase plane approach is computationally very efficient and has been used in on-line planning schemes [16].

In the dynamic programming approach, the state space is discretized into a grid of points. At each point, the next allowable state-points on the grid are defined by imposing velocity and acceleration constraints [58], or velocity, acceleration and jerk constraints [56]. The cost of moving to each of the allowable state-points is the motion time. This cost is computed assuming that the acceleration is constant over one step. The dynamic programming algorithm is then applied to determine the minimum-time trajectory over all of the points.

In the phase-plane approach, the path parameter and its first time derivative, the *pseudo-velocity*, are taken as the state. The second time derivative of the path parameter, the *pseudo-acceleration*, is the control. The nonlinear robot dynamics and actuator constraints are transformed into state-dependent constraints on the control. For each point along the path, the constraints also determine a maximum allowable end-effector pseudo-velocity such that the actuators can keep the end-effector on the path. This maximum pseudo-velocity as a function of the path parameter is called the *velocity limit curve*.

The optimal trajectory is derived by selecting the admissible control that produces the largest pseudo-velocity at each point along the path, without violating the velocity limit curve. The solution is given in terms of a switching curve in the phase plane and an efficient method for identifying the switching points is provided in [45]. A method for dealing with the case of singular control is given in [54].

In all of the above cited works, one actuator torque is always saturated. Bobrow et al. [5] and Shin and McKay [55] have shown that the optimal control takes values only on the bounds of its admissible set. Bobrow et al. [5] have also shown that, as a consequence, at least one actuator torque is always saturated. Hence, the optimal policy requires that at least one actuator torque switches value instantaneously between saturation levels. An example is presented next to illustrate this bang-bang nature of the optimal actuator torques.

A typical PCTOM trajectory of a 3-dof manipulator (Figure 2.1) and the velocity limit curve (VLC) are plotted in Figure 2.2 in the  $s-\dot{s}$  phase plane, where  $s$  is the path parameter and  $\dot{s}$  the

end-effector pseudo-velocity. The robot kinematic and dynamic parameters are taken from [45]

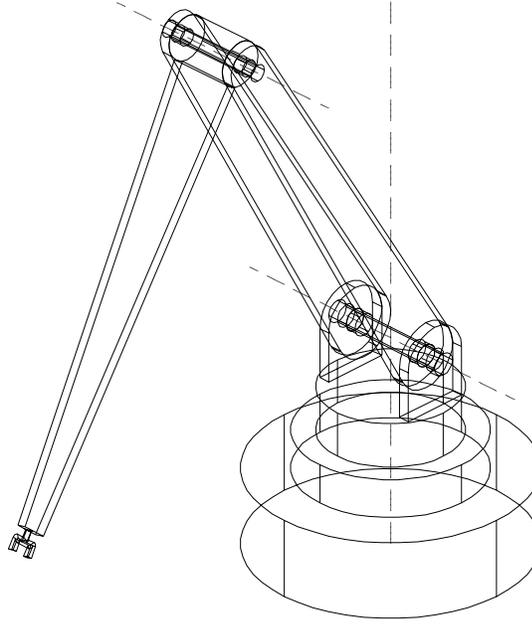


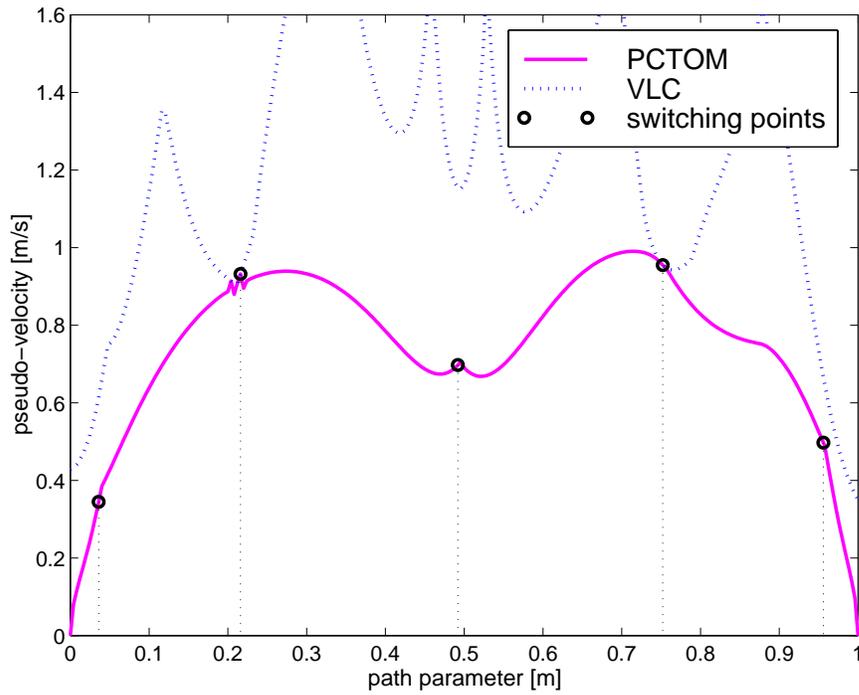
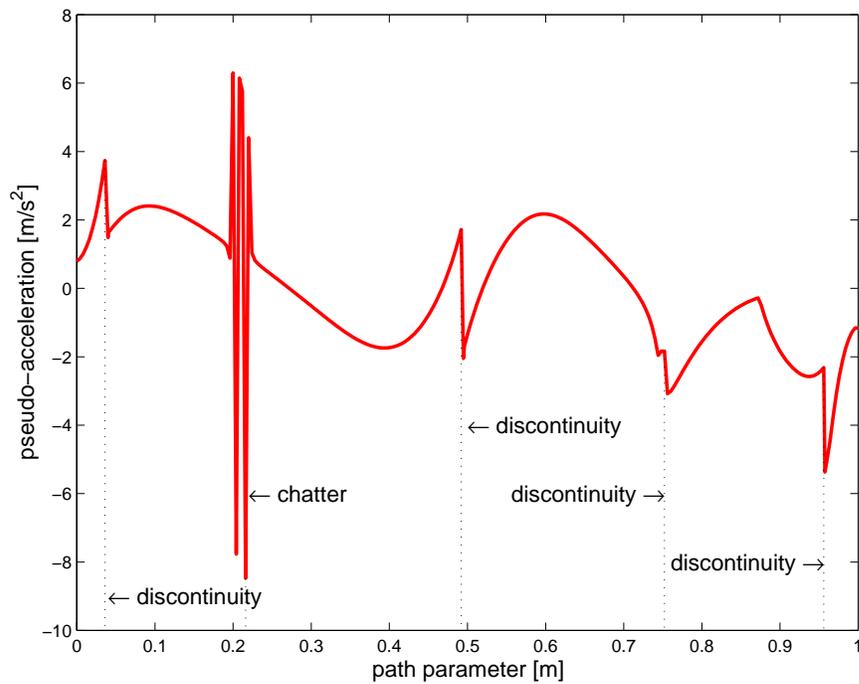
Figure 2.1: The elbow manipulator.

and are given in Table 2.1. In Table 2.1, C.O.M. is the center of mass of link  $i$  with respect to frame  $i$  and  $I_x, I_y, I_z$  are the central moments of inertia of link  $i$  with respect to axes parallel to the axes of frame  $i$ . The prescribed path, taken from the same source, is a parabola given in parametric form as:

$$\begin{aligned}
 x(s) &= 0.5 \\
 y(s) &= 20 \cdot s^3 - 30 \cdot s^2 + 10 \text{ } \mathit{cdots} s \\
 z(s) &= s - 0.5 \\
 s &= 0 \dots 1.
 \end{aligned}
 \tag{2.1}$$

The algorithm for computing the trajectory is also from [45].

As shown in Figure 2.2, the PCTOM pseudo-velocity is continuous along the entire path and differentiable everywhere except at the five switching points. These are the points along the trajectory where the pseudo-acceleration,  $\ddot{s}$ , switches from maximum to minimum or vice-versa. Plotting

Figure 2.2: PCTOM plotted in the  $s - \dot{s}$  phase plane.Figure 2.3: PCTOM trajectory in the  $s - \ddot{s}$  phase plane.

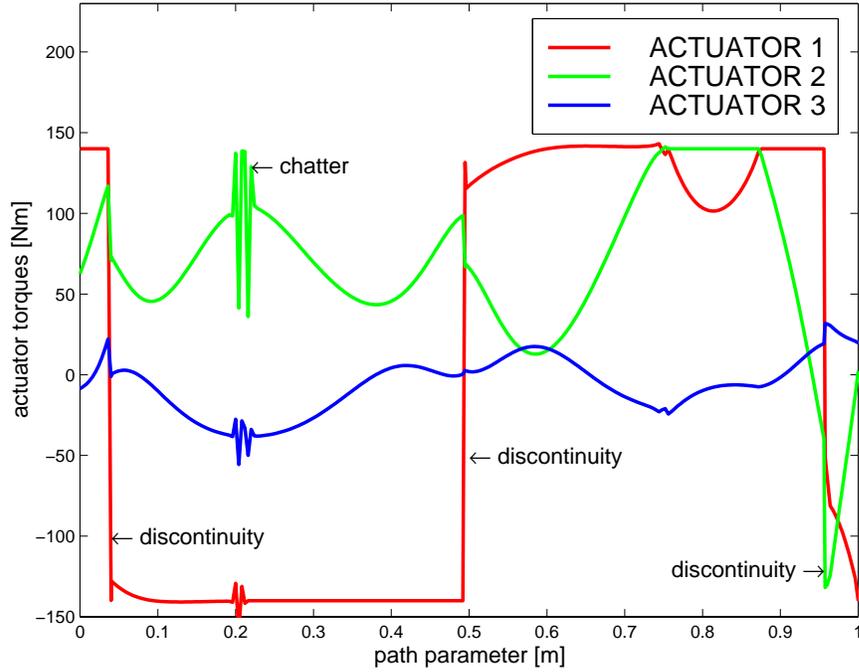


Figure 2.4: Actuator torques for the PCTOM trajectory.

Table 2.1: Robot parameters

Link [m]	C.O.M. [m]	Mass [kg]	$I_x$ [kgm <sup>2</sup> ]	$I_y$ [kgm <sup>2</sup> ]	$I_z$ [kgm <sup>2</sup> ]
$l_1 = 0.00$	$l_{c1} = 0.05$	$m_1 = 0.0$	$I_{x1} = 0.00$	$I_{y1} = 5.0$	$I_{z1} = 0.0$
$l_2 = 0.75$	$l_{c2} = 0.20$	$m_2 = 6.6$	$I_{x2} = 0.10$	$I_{y2} = 0.6$	$I_{z2} = 0.6$
$l_3 = 0.75$	$l_{c3} = 0.15$	$m_3 = 4.2$	$I_{x3} = 0.02$	$I_{y3} = 0.2$	$I_{z3} = 0.3$

the same PCTOM trajectory in the  $s$ - $\ddot{s}$  phase plane, Figure 2.3, shows the discontinuities in the pseudo-acceleration at the switching points. The actuator torques necessary to generate this motion are plotted in Figure 2.4.

These figures illustrate one of the problematic characteristics of the PCTOM trajectory: the pseudo-acceleration and the motor torques have discontinuities at the switching points and chatter occurs when the trajectory in the  $s$ - $\ddot{s}$  phase plane touches the VLC.

### 2.1.2 Practical Implementation Issues

Time-optimal trajectories such as the one presented in the previous section represent a measure of the maximum robot tip velocity at which the actuators can still keep it along the path. However, as shown, they require instantaneous changes in joint accelerations and in joint actuator torques.

Such discontinuities in actuator torques reflect the system model used in computing the PCTOM trajectories. Namely, the robot links are considered to be completely rigid and the actuator dynamics are ignored. While these modelling assumptions are sufficient for low speed trajectories, they can be expected to have an impact on a minimum time motion.

As a result of these assumptions, two negative effects arise, and each effect compounds the other. First, since the actuators cannot generate discontinuous torques due to their own internal dynamics, the joint output lags behind the reference signal. This decreases the tracking accuracy, and activates the tracking controller. Thus, after each switching point, the joint actuators exhibit chatter. Second, this chatter of the joint actuators induces vibrations in the robot links. These vibrations aggravate the wear on the robot and reduce the tracking performance. As a result, the tracking controller is activated more often and additional strain is imposed on the actuators.

Vibrations are a fundamental limiting factor in any high speed manipulation. They result in robot wear and reduce the life span of the actuators. In an industrial application, where the robot must perform the same task hundreds of times per day, fast motions are desirable, but the structural wear and actuator lifetime are also important. The typical industrial solution is to detune the robot feedback controller by limiting its bandwidth so as not to excessively excite these vibrations. In path-following manipulations, however, the tracking performance may be adversely affected by limiting the frequency content of the control signal and other solutions may be required. Thus, all of these issues must be addressed in an industrial implementation of PCTOM trajectories.

## 2.2 Research Addressing the Issues in PCTOM

The practicalities involved in path-constrained time-optimal control were recognized shortly after efficient algorithms for trajectory computation were developed and the bang-bang nature of the controls was proven. Methods have been devised for addressing these problems both at the

tracking controller level and at the trajectory planning level.

A fundamental problem associated with time-optimal control is to ensure a prescribed tracking accuracy. This is the objective of any path following manipulation task. The tracking performance is affected by unmodelled link flexibility and joint friction and disturbances occurring during the performance of the task. Nonetheless, algorithms have been proposed that consider only the effect of disturbances on the tracking performance. Since, in time-optimal control, one actuator is always saturated while the other actuators adjust their values to keep the end-effector on the path, there is no control input available to compensate for modelling errors and disturbances. Thus, performance deteriorates when unexpected control saturation occurs.

Improved tracking accuracy may be guaranteed from the planning stage by keeping some torque in reserve [7], hence determining near-time optimal trajectories. Time-optimal trajectories with increased tracking performance may be obtained by accounting for the actuator dynamics. This approach is adopted in Tarkiainen and Shiller [63]. In their work, the actuator constraints are imposed as limits on the motor voltages. The system consisting of the robot and its actuators is a third order dynamic system with the *pseudo-jerk*, the third time derivative of the path parameter, as the control variable. The time-optimal policy is shown to be bang-bang or bang-singular-bang in the pseudo-jerk and an algorithm is given to determine the optimal trajectory. The algorithm ignores the case of singular controls, since these can be avoided by appropriately choosing the path [54] or by convexifying the set of admissible controls [49]. The optimal torques are continuous in this formulation and the trajectory is smooth. However, the method does not allow the path planner to directly impose a desired degree of trajectory smoothness.

The concern for improving the tracking performance exists not only at the planning, but also at the tracking level. Solutions proposed at the controller level include: (i) a secondary controller for on-line trajectory time scaling [17]; (ii) an LQG controller for adjusting the switching times on-line based on the measured performance of the system [62]; (iii) a closed-loop trajectory generator that is robust to experimentally identified levels of disturbance [32]; (iv) a trajectory preshaping scheme that compensates for actuator dynamics [51]. Although these methods increase the tracking accuracy, they do not directly address the problem of discontinuous reference torques or completely compensate for the resulting transient system behaviour. In [51], however, the overshoot of the nominal torques and the control chatter after each bang are acknowledged. It is pointed out that

the overshoot is quite significant and requires that conservative bounds on the admissible torques be used during the planning stage.

While good tracking accuracy is highly desirable for path-constrained motions, robot wear and the life span of the actuators are important considerations in any industrial application of time-optimal control. Unfortunately, it is much more difficult to measure the cumulative effect that a specific algorithm may have on vibrations or wear. Acknowledgements of this issue exist in the literature [8], [33], [34], [38], [45], [50], [56], and it is recognized that, in practice, some degree of smoothness is necessary. However, in this review, only one method was identified for generating and evaluating a smooth trajectory which explicitly considers vibrations [66]. This method is based on the iterative adjusting of the weights in the cost function till a smooth input is obtained that results in reduced vibrations in the joints and no oscillations at the end-point of the trajectory.

At the planning level, the smoothness requirement may be considered with respect to the trajectory smoothness or the smoothness of the actuator torques. Since the actuator torques are computed as continuous functions of the trajectory parameters, a smooth trajectory translates into smooth control signals.

In Lin et al. [38], smooth trajectories are planned by choosing cubic polynomials to approximate the joint paths. This results in trajectories with continuous joint velocities and accelerations. The motion time is then minimized subject to the physical constraints on joint velocities, accelerations, and jerks. The limits used are conservative, because they are constant over the entire robot work space. Hence, they are determined for the worst case, i.e. the arm fully extended (maximum inertia).

The same joint path parameterization is used in Cao et al. [8], together with a quadratic cost function involving time and joint accelerations. The limits on the joint variables are imposed after the optimization is solved, through time scaling. The resulting motion has optimal joint velocity profiles, in the sense that the variance of the joint accelerations is minimum in the natural motion, i.e. with zero initial and final states. Then, for the same limits on the motion acceleration, the motion vibration is minimum. As a result, the strain energy and the mechanical wear are approximately minimal.

The actuator torques may also be smoothed directly by adding an “energy” term to the objective

function [45], [50], [56], [64], [66]. This term convexifies the cost and the Hamiltonian with respect to the control and creates a minimum that is not necessarily on the control boundaries. In [56] constraints on the joint jerks are imposed explicitly, while the algorithms in [50] and [66] allow weighting of the energy term. By modifying this weight, Shiller [50] achieved a desired compromise between minimum time and increased accuracy, while Wappenhans et al. [66] obtained motions with reduced low-frequency vibrations and no residual oscillations.

At the tracking and control level, Newman [43] developed a feedback control scheme as a combination of bang-bang control with sliding-mode control. This feedback law drives the system to imitate the desirable and feasible set of dynamics, which are specified as a state constraint corresponding to the motion along the switching curve of the time-optimal control. The concern with unmodelled dynamics is explicit. An analysis in terms of a design parameter of the controller is performed. This analysis shows how the parameter should be chosen such that high-frequency dynamics of the system does not destabilize the controller. Thus, the proposed controller is robust to parametric uncertainty and unmodelled flexibility of the robot structure.

Li et al. [37] explicitly acknowledged vibrations as the limiting factor in time-optimal control. Ways of compromising between minimum motion time, and robot wear and actuator lifetime for point-to-point motions were considered in their work. Both vibrations and wear were addressed in two ways: (i) by planning time-optimal motions together with different torque parameterizations; (ii) by using cost functionals involving terms with various exponential powers of the actuator torques with piecewise linear, continuous controls. By penalizing the terms containing powers of the actuator torques, varying degrees of smoothness of the controls were achieved. Two strategies were recommended for time-optimal control. One involved prefiltering the time-optimal torques and then matching the bandwidth of the low-pass filter to the bandwidth of the feedback controller. The other utilized the derivative of the torque as the control. However, the lowest vibration levels corresponded to a quadratic cost functional involving the second derivative of the torque.

In summary, actuator chatter, large overshoot of the nominal torques, decreased tracking accuracy, vibrations, and robot wear arise in the direct implementation of PCTOM trajectories due to the bang-bang nature of the actuator torques required by such trajectories. These difficulties have been addressed by a large body of research, and methods have been devised for overcoming these problems, both at the trajectory planning level and at the tracking controller level. At the

controller level, solutions include: sophisticated controllers, more accurate models of the robot-actuators system, filtering of the reference torques, and parameterizations of the control signals. However, such approaches do not directly address the cause or completely solve the effects of these difficulties. At the planning level, proposed solutions include smooth joint path parameterizations, objective functions incorporating an energy term, the use of actuator voltages as the controls, and the restriction of controls to some suitably chosen set of functions, i.e., piecewise linear and continuous or piecewise quadratic and continuous controls.

In the following section, a new formulation is proposed for the planning of time-optimal trajectories for path-constrained tasks. This formulation avoids the disadvantages of PCTOM trajectories by choosing the derivatives of the actuator torques as the controls.

## 2.3 Smooth Path-Constrained Time-Optimal Control

The view adopted in this work and supported by the reviewed research is that physical manipulators cannot exhibit discontinuous actuator torques. When trying to achieve such a behaviour, robot links can be subjected to severe vibrations that may lead to increased robot wear and failure of the actuator themselves.

Herein it is considered that it is important to address these problems from the trajectory planning level and not leave them entirely to the tracking controller. The reasons for advocating these measures at the planning level are twofold. First, eliminating the torque discontinuities from the planning stage will reduce the controller effort and thus better tracking performance and a longer life span of the actuators are expected. Also, more room is left for controller robustness in face of model uncertainties. This is important, especially considering that an accurate dynamic model of the robot is difficult to obtain, and all trajectory planning schemes rely on such a model. Second, since the planned trajectories will be free from the problems associated with PCTOM trajectories, it will be possible to implement them on industrial manipulators using their own “general purpose” controllers.

Using this approach, minimum time remains the criterion for planning the trajectory. However, the physical limitations of the robot are considered by requiring that the optimal trajectory be smooth. In this work, trajectory smoothness is imposed through limits on the rate of variation

of the actuator torques, hereafter termed *actuator jerks*. Thus, different degrees of smoothness of a trajectory may be achieved by adjusting the limits on the actuator jerks. The time-optimal trajectory resulting when both actuator jerk and actuator torque limits are imposed will be called the *smooth, path-constrained, time-optimal motion* (SPCTOM).

### 2.3.1 Problem Formulation

The SPCTOM planning problem is formulated hereafter for the case of a manipulation task starting and ending at rest. Thus, the problem that is addressed in this work is:

$$\min_{\dot{\mathbf{T}} \in \Omega} J = \int_0^{t_f} 1 dt, \quad (2.2)$$

subject to the standard model of the manipulator dynamics:

$$\mathbf{M}(\mathbf{q}) \cdot \ddot{\mathbf{q}} + \dot{\mathbf{q}}^T \cdot \mathbf{C}(\mathbf{q}) \cdot \dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \mathbf{T}, \quad (2.3)$$

the boundary conditions:

$$\mathbf{q}(0) = \mathbf{q}_0 ; \mathbf{q}(t_f) = \mathbf{q}_f ; \dot{\mathbf{q}}(0) = \dot{\mathbf{q}}(t_f) = \mathbf{0} ; \ddot{\mathbf{q}}(0) = \ddot{\mathbf{q}}(t_f) = \mathbf{0}, \quad (2.4)$$

the path constraints:

$$\mathbf{r} = \mathbf{r}(s), \quad (2.5)$$

the actuator torque limits:

$$\mathbf{T}_{min} \leq \mathbf{T} \leq \mathbf{T}_{max} \quad (2.6)$$

and the actuator jerk limits:

$$\dot{\mathbf{T}}_{min} \leq \dot{\mathbf{T}} \leq \dot{\mathbf{T}}_{max} \quad (2.7)$$

where  $\mathbf{T}_{min}$ ,  $\mathbf{T}_{max}$ ,  $\dot{\mathbf{T}}_{min}$ , and  $\dot{\mathbf{T}}_{max}$  are the actuator torque and jerk limits, respectively. Further-

more,  $\mathbf{q} \in \mathbf{R}^n$  is the vector of joint positions,  $\mathbf{T} \in \mathbf{R}^n$  is the vector of actuator torques,  $\dot{\mathbf{T}} \in \mathbf{R}^n$  is the vector of actuator jerks,  $\mathbf{M}(\mathbf{q}) \in \mathbf{R}^{n \times n}$  is the inertia matrix of the manipulator,  $\mathbf{C}(\mathbf{q}) \in \mathbf{R}^{n \times n \times n}$  is a third order tensor representing the coefficients of the centrifugal and Coriolis forces,  $\mathbf{G}(\mathbf{q}) \in \mathbf{R}^n$  is the vector of gravity terms,  $\mathbf{r} \in \mathbf{R}^3$  is a  $C^1$  continuous curve parametrized by  $s$ , which may be, for example, the arc length, and  $n$  is the number of dof of the manipulator. For simplicity, viscous and static friction terms have been neglected in the dynamics.

In the above formulation, the actuator jerks represent the bounded controls. However, these controls would later have to be integrated, in order to derive the actual system inputs in terms of desired actuator torques. Since the Lagrangian form of the robot dynamics incorporates only the actuator torques, the third order dynamics is derived. Differentiation of Equation (2.3) with respect to time results in:

$$\mathbf{M}(\mathbf{q}) \cdot \ddot{\mathbf{q}} + \dot{\mathbf{M}}(\mathbf{q}) \cdot \dot{\mathbf{q}} + \ddot{\mathbf{q}}^T \cdot \mathbf{C}(\mathbf{q}) \cdot \dot{\mathbf{q}} + \dot{\mathbf{q}}^T \cdot \dot{\mathbf{C}}(\mathbf{q}) \cdot \dot{\mathbf{q}} + \dot{\mathbf{q}}^T \cdot \mathbf{C}(\mathbf{q}) \cdot \ddot{\mathbf{q}} + \dot{\mathbf{G}}(\mathbf{q}) = \dot{\mathbf{T}}. \quad (2.8)$$

In terms of optimal control theory, the third order dynamics of the manipulator, Equation (2.8), are the dynamics of the system, with  $\dot{\mathbf{T}}$  representing the  $n$ -dimensional bounded controls. In the usual fashion, Equation (2.8) can be written in the form of the system state equation:

$$\dot{\mathbf{x}} = F(\mathbf{x}, \mathbf{u}) = \mathbf{F}(\mathbf{x}) + \mathbf{K}(\mathbf{x})\mathbf{u}, \quad (2.9)$$

where the state of the system is:

$$\mathbf{x}^T = (\mathbf{q}^T \quad \dot{\mathbf{q}}^T \quad \ddot{\mathbf{q}}^T)_{1 \times 3n}, \quad (2.10)$$

while:

$$\mathbf{F}(\mathbf{x}) = \begin{pmatrix} \dot{\mathbf{q}} \\ \ddot{\mathbf{q}} \\ -\mathbf{M}^{-1}(\mathbf{q})(\dot{\mathbf{M}}(\mathbf{q})\dot{\mathbf{q}} + \ddot{\mathbf{q}}^T \mathbf{C}(\mathbf{q}) \dot{\mathbf{q}} + \dot{\mathbf{q}}^T \dot{\mathbf{C}}(\mathbf{q}) \dot{\mathbf{q}} + \dot{\mathbf{q}}^T \mathbf{C}(\mathbf{q}) \ddot{\mathbf{q}} + \dot{\mathbf{G}}(\mathbf{q})) \end{pmatrix}_{3n \times 1}, \quad (2.11)$$

$$\mathbf{K}(\mathbf{x}) = \begin{pmatrix} \mathbf{0}_{n \times n} \\ \mathbf{0}_{n \times n} \\ \mathbf{M}^{-1}(\mathbf{q}) \end{pmatrix}_{3n \times n}, \quad (2.12)$$

and:

$$\mathbf{u} = \dot{\mathbf{T}}_{n \times 1}. \quad (2.13)$$

Thus, the optimal control problem is reformulated as follows. Find the control  $\dot{\mathbf{T}}$ :

$$\dot{\mathbf{T}} : [t_0, t_f] \longrightarrow \mathbf{\Omega} \subset \mathbf{R}^n, \quad (2.14)$$

where  $\mathbf{\Omega}$  is the set of admissible controls, that drives the system (2.9) in minimum time from an initial state:

$$\mathbf{x}_0^T = \begin{pmatrix} \mathbf{q}_0^T & \mathbf{0}_{1 \times n} & \mathbf{0}_{1 \times n} \end{pmatrix}_{1 \times 3n}, \quad (2.15)$$

to a final given state:

$$\mathbf{x}_f^T = \begin{pmatrix} \mathbf{q}_f^T & \mathbf{0}_{1 \times n} & \mathbf{0}_{1 \times n} \end{pmatrix}_{1 \times 3n} \quad (2.16)$$

subject to the limits on the states:

$$\mathbf{T}_{min} \leq \mathbf{M}(\mathbf{q}) \cdot \ddot{\mathbf{q}} + \dot{\mathbf{q}}^T \cdot \mathbf{C}(\mathbf{q}) \cdot \dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) \leq \mathbf{T}_{max}, \quad (2.17)$$

and the limits on the controls:

$$\dot{\mathbf{T}}_{min} \leq \dot{\mathbf{T}} \leq \dot{\mathbf{T}}_{max}. \quad (2.18)$$

Formulation of the problem solution using Pontryagin's Maximum Principle results in a TPBVP in a  $3n$ -dimensional state space.

### 2.3.2 Path Constraints

The dynamic system described by Equation (2.8) has  $3n$  degrees of freedom. However, the path constraints (2.5) parameterize the tip position by a single parameter  $s$  and reduce the order of the system to 3. By expressing the joint positions, velocities, accelerations, and jerks as functions of the path parameter  $s$ , the actuator torque limits and jerk limits are transformed into limits on  $\ddot{s}$ , the pseudo-acceleration, and  $\ddot{\ddot{s}}$ , the pseudo-jerk, respectively.

The joint values corresponding to a given end-effector position and orientation,  $\mathbf{q} = \mathbf{q}(\mathbf{r})$  can be determined for each value of the path parameter  $s$  by solving the manipulator inverse kinematic equations and then substituting the path constraints (2.5):

$$\mathbf{q} = \mathbf{q}(\mathbf{r}(s)) \quad (2.19)$$

The dependence of the joint velocities  $\dot{\mathbf{q}}$  on the path parameter and its first derivative, the pseudo-velocity, is determined using the Jacobian matrix of the forward kinematic map:

$$\dot{\mathbf{r}} = \mathbf{J}(\mathbf{q}) \cdot \dot{\mathbf{q}} \quad (2.20)$$

If the manipulator is not at a singular position, the Jacobian is invertible and:

$$\dot{\mathbf{q}} = \mathbf{J}^{-1} \cdot \dot{\mathbf{r}} = \mathbf{J}^{-1} \cdot \frac{d\mathbf{r}}{ds} \frac{ds}{dt} = \mathbf{J}^{-1} \cdot \mathbf{r}' \dot{s} = \mathbf{q}' \dot{s}, \quad (2.21)$$

where:

$$\mathbf{q}' = \mathbf{J}^{-1} \cdot \mathbf{r}', \quad (2.22)$$

and  $'$  denotes the derivative with respect to the path parameter.

From Equation (2.21), the joint accelerations and jerks are obtained by successive differentia-

tions with respect to time:

$$\dot{\mathbf{q}} = \mathbf{q}' \dot{s} \quad (2.23)$$

$$\Rightarrow \ddot{\mathbf{q}} = \mathbf{q}'' \dot{s}^2 + \mathbf{q}' \ddot{s} \quad (2.24)$$

$$\Rightarrow \dddot{\mathbf{q}} = \mathbf{q}''' \dot{s}^3 + 3 \cdot \mathbf{q}'' \cdot \dot{s} \ddot{s} + \mathbf{q}' \ddot{\ddot{s}}, \quad (2.25)$$

where  $\mathbf{q}'$ ,  $\mathbf{q}''$ , and  $\mathbf{q}'''$  can be derived using the inverse of Equation (2.22):

$$\mathbf{r}' = \mathbf{J} \cdot \mathbf{q}' \quad (2.26)$$

$$\Rightarrow \mathbf{r}'' = \frac{d\mathbf{J}}{ds} \cdot \mathbf{q}' + \mathbf{J} \cdot \mathbf{q}'' \quad (2.27)$$

$$\Rightarrow \mathbf{r}''' = \frac{d^2\mathbf{J}}{ds^2} \cdot \mathbf{q}' + 2 \cdot \frac{d\mathbf{J}}{ds} \cdot \mathbf{q}'' + \mathbf{J} \cdot \mathbf{q}'''. \quad (2.28)$$

Now, from Equation (2.27):

$$\mathbf{q}'' = \mathbf{J}^{-1} \cdot \left( \mathbf{r}'' - \frac{d\mathbf{J}}{ds} \cdot \mathbf{q}' \right), \quad (2.29)$$

and from Equation (2.28):

$$\mathbf{q}''' = \mathbf{J}^{-1} \cdot \left( \mathbf{r}''' - \frac{d^2\mathbf{J}}{ds^2} \cdot \mathbf{q}' - 2 \cdot \frac{d\mathbf{J}}{ds} \cdot \mathbf{q}'' \right). \quad (2.30)$$

Substituting Equations (2.23), (2.24) and (2.25) in (2.8) yields the jerk equation:

$$\begin{aligned} & \mathbf{M} \cdot (\mathbf{q}''' \cdot \dot{s}^3 + 3 \cdot \mathbf{q}'' \cdot \dot{s} \cdot \ddot{s} + \mathbf{q}' \cdot \ddot{\ddot{s}}) + \frac{d\mathbf{M}}{ds} \cdot \dot{s} \cdot (\mathbf{q}'' \cdot \dot{s}^2 + \mathbf{q}' \cdot \ddot{s}) + \\ & + (\mathbf{q}'' \cdot \dot{s}^2 + \mathbf{q}' \cdot \ddot{s})^T \cdot \mathbf{C} \cdot \mathbf{q}' \cdot \dot{s} + \mathbf{q}'^T \cdot \dot{s} \cdot \frac{d\mathbf{C}}{ds} \cdot \dot{s} \cdot \mathbf{q}' \cdot \dot{s} + \\ & + \mathbf{q}'^T \cdot \dot{s} \cdot \mathbf{C} \cdot (\mathbf{q}''^T \cdot \dot{s}^2 + \mathbf{q}' \cdot \ddot{s}) + \frac{d\mathbf{G}}{ds} \cdot \dot{s} = \dot{\mathbf{T}}. \end{aligned} \quad (2.31)$$

The actuator jerk bounds become:

$$\dot{\mathbf{T}}_{min} \leq \mathbf{a}(s) \cdot \ddot{\ddot{s}} + \mathbf{b}(s) \cdot \dot{s} \cdot \ddot{s} + \mathbf{c}(s) \cdot \dot{s}^3 + \mathbf{d}(s) \cdot \dot{s} \leq \dot{\mathbf{T}}_{max}, \quad (2.32)$$

where,

$$\mathbf{a}_{n \times 1}(s) = \mathbf{M} \cdot \mathbf{q}', \quad (2.33)$$

$$\mathbf{b}_{n \times 1}(s) = 3 \cdot \mathbf{M} \cdot \mathbf{q}'' + \frac{d\mathbf{M}}{ds} \cdot \mathbf{q}' + 2 \cdot \mathbf{q}'^T \cdot \mathbf{C} \cdot \mathbf{q}', \quad (2.34)$$

$$\mathbf{c}_{n \times 1}(s) = \mathbf{M} \cdot \mathbf{q}''' + \frac{d\mathbf{M}}{ds} \cdot \mathbf{q}'' + \mathbf{q}''^T \cdot \mathbf{C} \cdot \mathbf{q}' + \mathbf{q}'^T \cdot \frac{d\mathbf{C}}{ds} \cdot \mathbf{q}' + \mathbf{q}'^T \cdot \mathbf{C} \cdot \mathbf{q}'', \quad (2.35)$$

$$\mathbf{d}_{n \times 1}(s) = \frac{d\mathbf{G}}{ds} \cdot \dot{s}. \quad (2.36)$$

The matrices  $\frac{d\mathbf{M}}{ds}$  and  $\frac{d\mathbf{G}}{ds}$  and the third order tensor  $\frac{d\mathbf{C}}{ds}$  are robot dependent.

Substituting Equations (2.23) and (2.24) into (2.3) yields the torque equation:

$$\mathbf{M} \cdot \mathbf{q}' \cdot \ddot{s} + \mathbf{M} \cdot \mathbf{q}'' \cdot \dot{s}^2 + \mathbf{q}'^T \cdot \dot{s} \cdot \mathbf{C} \cdot \mathbf{q}' \cdot \dot{s} + \mathbf{G} = \mathbf{T} \quad (2.37)$$

and the actuator torque bounds become:

$$\mathbf{T}_{min} \leq \mathcal{A}(s) \cdot \ddot{s} + \mathcal{B}(s) \cdot \dot{s}^2 + \mathcal{C}(s) \leq \mathbf{T}_{max} \quad (2.38)$$

where:

$$\mathcal{A}_{n \times 1}(s) = \mathbf{M} \cdot \mathbf{q}', \quad (2.39)$$

$$\mathcal{B}_{n \times 1}(s) = \mathbf{M} \cdot \mathbf{q}'' + \mathbf{q}'^T \cdot \mathbf{C} \cdot \mathbf{q}', \quad (2.40)$$

$$\mathcal{C}_{n \times 1}(s) = \mathbf{G}. \quad (2.41)$$

### 2.3.3 Torque Limits

The actuator torque bounds (2.38) represent a set of  $2n$  inequalities of the form:

$$T_{i,min} \leq A_i \cdot \ddot{s} + B_i \cdot \dot{s}^2 + C_i \leq T_{i,max}, \quad i = 1, \dots, n, \quad (2.42)$$

where  $A_i$ ,  $B_i$ ,  $C_i$ ,  $T_{i,min}$ , and  $T_{i,max}$  are the elements of the vectors  $\mathcal{A}(s)$ ,  $\mathcal{B}(s)$ ,  $\mathcal{C}(s)$ ,  $\mathbf{T}_{min}$ , and  $\mathbf{T}_{max}$ , respectively. For some value of the path parameter  $s$ , the feasible region of  $\dot{s}$  and  $\ddot{s}$ , satisfying (2.42) and  $\dot{s} \geq 0$ , forms a polygon in the  $\dot{s}^2 - \ddot{s}$  plane. Such a polygon is shown schematically in

Figure 2.5 for a 3-dof manipulator.

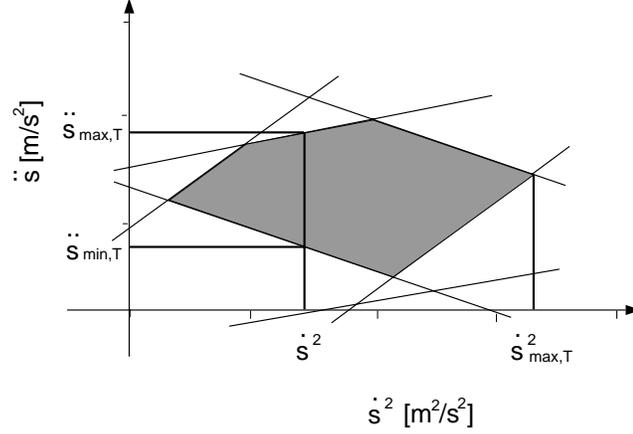


Figure 2.5: Admissible region in the  $\dot{s}^2 - \ddot{s}$  plane.

As discussed in [54], for each value of the path parameter  $s$ , the bounds on the pseudo-velocity  $\dot{s}$  and pseudo-acceleration  $\ddot{s}$  due to the torque constraints (2.38) are:

$$\dot{s} \leq \dot{s}_{max,T}(s) \quad (2.43)$$

$$\ddot{s}_{min,T}(s, \dot{s}) \leq \ddot{s} \leq \ddot{s}_{max,T}(s, \dot{s}), \quad (2.44)$$

where:

$$\ddot{s}_{min,T}(s, \dot{s}) = \max_i \left\{ \min \left( \frac{T_i - C_i - B_i \dot{s}^2}{A_i} \right) \right\}, \quad i = 1, \dots, n, \quad (2.45)$$

$$\ddot{s}_{max,T}(s, \dot{s}) = \min_i \left\{ \max \left( \frac{T_i - C_i - B_i \dot{s}^2}{A_i} \right) \right\}, \quad i = 1, \dots, n, \quad (2.46)$$

and

$$\dot{s}_{max,T}(s) = \min_{i,j} \left\{ \max \left( \frac{A_j(T_i - C_i) - A_i(T_j - C_j)}{A_j B_i - A_i B_j} \right) \right\}, \quad i, j = 1, \dots, n, \quad (2.47)$$

when  $A_i \neq 0$  and  $A_j B_i - A_i B_j \neq 0$ . The subscript  $T$  is used to discriminate the pseudo-velocity and pseudo-acceleration bounds due to the torque constraints (2.38) from those due to the jerk constraints (2.32), which will be denoted with the subscript  $J$ . If  $A_i = 0$  for some  $i$ , then this actuator does not bound the pseudo-acceleration at this point along the path. The limits are determined by the other  $n - 1$  actuators. The case that  $A_i = 0$  for all  $i$  is unlikely, since  $\mathbf{M}$  is generally positive definite, except for some few particular cases [1], and  $\mathbf{q}'$  is a vector tangent to the path. If  $A_j B_i - A_i B_j = 0$ , then the  $i$ th and  $j$ th actuators do not contribute to the velocity limit, or the maximum velocity is infinite.

In the  $s - \dot{s} - \ddot{s}$  phase space, the pseudo-acceleration bounds (2.44) represent a lower and an upper surface which meet and vanish along the VLC on the  $s - \dot{s}$  plane. In Figure 2.6<sup>1</sup>, these surfaces have been plotted for the actuator torque limits given in Table 2.2. All the states inside these surfaces are states that obey the actuator torque constraints (2.17).

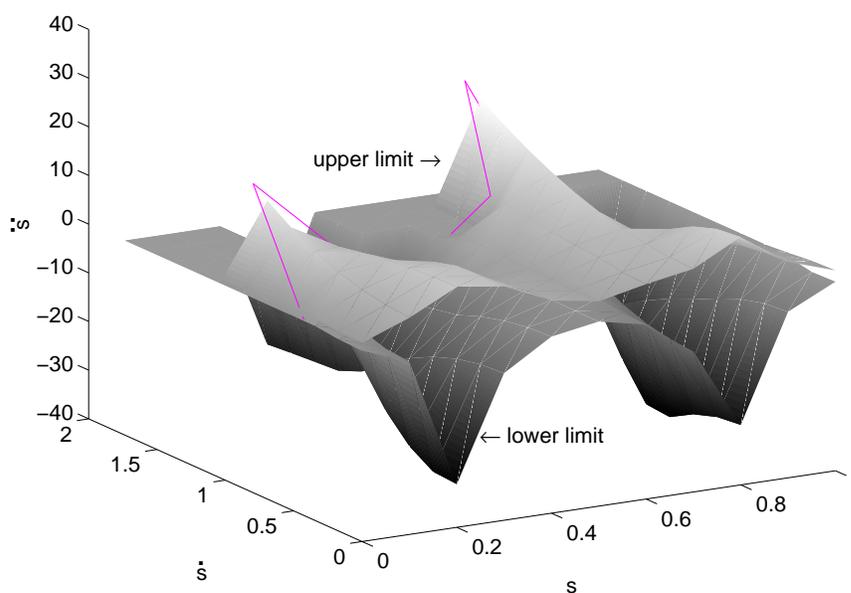


Figure 2.6: Torque limits in the  $s - \dot{s} - \ddot{s}$  space and the VLC (dark line).

The VLC represents an upper bound for any feasible trajectory in the  $s - \dot{s}$  plane.

<sup>1</sup>Unless otherwise specified, all the figures in this and the following sections refer to the elbow manipulator with the dynamic parameters given in Table 2.1 moving along the parabolic path specified by Equations (2.1).

### 2.3.4 Jerk Limits

A similar approach can be used to determine the pseudo-velocity, pseudo-acceleration and pseudo-jerk bounds due to the actuator jerk limits. Thus, the actuator jerk bounds (2.32) represent a set of  $2n$  inequalities of the form:

$$\dot{T}_{i,min} \leq a_i \ddot{s} + b_i^* \dot{s} + c_i^* \leq \dot{T}_{i,max}, \quad i = 1, \dots, n \quad (2.48)$$

where  $a_i$ ,  $b_i$ ,  $c_i$ ,  $d_i$ ,  $\dot{T}_{i,min}$ , and  $\dot{T}_{i,max}$  are the elements of the vectors  $\mathbf{a}$ ,  $\mathbf{b}$ ,  $\mathbf{c}$ ,  $\mathbf{d}$ ,  $\dot{\mathbf{T}}_{min}$ , and  $\dot{\mathbf{T}}_{max}$ , respectively, and  $b_i^* = b_i \dot{s}$  and  $c_i^* = c_i \dot{s}^3 + d_i \dot{s}$ . For a set of values of the path parameter  $s$  and the pseudo-velocity  $\dot{s}$ , the feasible region of  $\ddot{s}$  and  $\ddot{\dot{s}}$  satisfying (2.48), forms a polygon in the  $\ddot{s} - \ddot{\dot{s}}$  plane. Such a polygon is shown schematically in Figure 2.7 for a 3-dof manipulator.

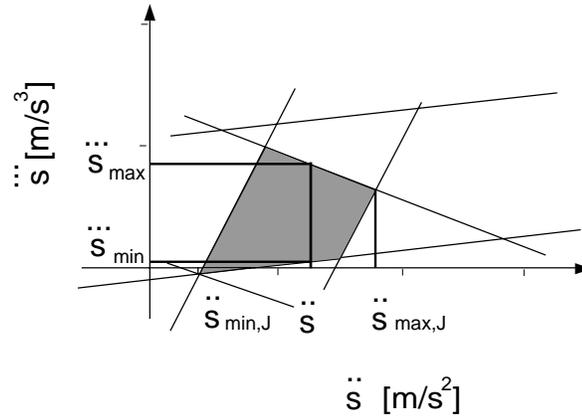


Figure 2.7: Admissible region in the  $\ddot{s} - \ddot{\dot{s}}$  plane.

As shown in Figure 2.7, for each set of values for  $s$  and  $\dot{s}$ , the bounds on the pseudo-acceleration  $\ddot{s}$  and the pseudo-jerk  $\ddot{\dot{s}}$  due to the jerk constraints (2.32) are:

$$\ddot{s}_{min,J}(s, \dot{s}) \leq \ddot{s} \leq \ddot{s}_{max,J}(s, \dot{s}) \quad (2.49)$$

$$\ddot{\dot{s}}_{min}(s, \dot{s}, \ddot{s}) \leq \ddot{\dot{s}} \leq \ddot{\dot{s}}_{max}(s, \dot{s}, \ddot{s}) \quad (2.50)$$

where:

$$\ddot{s}_{min}(s, \dot{s}, \ddot{s}) = \max_i \left\{ \min_{\dot{T}_i} \left( \frac{\dot{T}_i - c_i^* - b_i^* \ddot{s}}{a_i} \right) \right\}, \quad i = 1, \dots, n, \quad (2.51)$$

$$\ddot{s}_{max}(s, \dot{s}, \ddot{s}) = \min_i \left\{ \max_{\dot{T}_i} \left( \frac{\dot{T}_i - c_i^* - b_i^* \ddot{s}}{a_i} \right) \right\}, \quad i = 1, \dots, n, \quad (2.52)$$

and

$$\ddot{s}_{min,J}(s, \dot{s}) = \max_{i,j} \left\{ \min_{\dot{T}_i, \dot{T}_j} \left( \frac{a_j(\dot{T}_i - c_i^*) - a_i(\dot{T}_j - c_j^*)}{a_j b_i^* - a_i b_j^*} \right) \right\}, \quad i, j = 1, \dots, n, \quad (2.53)$$

$$\ddot{s}_{max,J}(s, \dot{s}) = \min_{i,j} \left\{ \max_{\dot{T}_i, \dot{T}_j} \left( \frac{a_j(\dot{T}_i - c_i^*) - a_i(\dot{T}_j - c_j^*)}{a_j b_i^* - a_i b_j^*} \right) \right\}, \quad i, j = 1, \dots, n, \quad (2.54)$$

assuming that  $a_i \neq 0$  and  $a_j b_i^* - a_i b_j^* \neq 0$ . Again, the subscript  $J$  indicates that the  $\ddot{s}$  and  $\dot{s}$  bounds are due to the jerk constraints (2.32). If  $a_i = 0$  for some  $i$ , then this actuator does not bound the pseudo-jerk at this point along the path. The limits are determined by the other  $n - 1$  actuators. The case that  $a_i = 0$  for all  $i$  is unlikely for the same reasons as in the case of the torque constraints, since  $\mathbf{a} = \mathcal{A}$ . If  $a_j b_i^* - a_i b_j^* = 0$ , then the  $i$ th and  $j$ th actuators do not contribute to the pseudo-acceleration limits, or any pseudo-acceleration is admissible. This is the case with any manipulator starting from rest, for example.

In the  $s - \dot{s} - \ddot{s}$  space, the pseudo-acceleration constraints (2.49) represent two limiting surfaces which meet and vanish along a curve. The projection of this curve on the  $s - \dot{s}$  plane is called the jerk limited velocity limit curve (JL-VLC). The limiting surfaces are plotted in Figure 2.8 for the first column of actuator jerk limits given in Table 2.2.

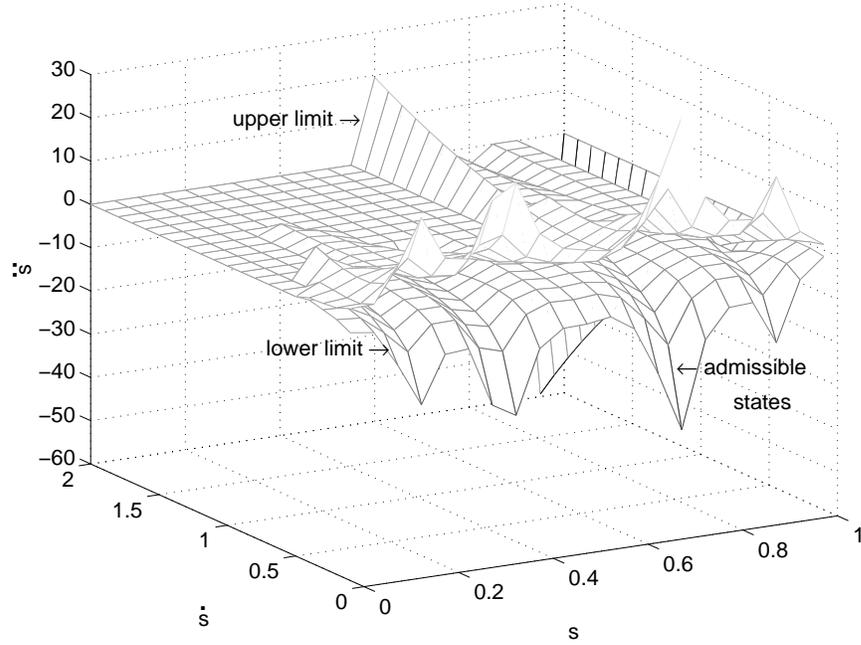
The JL-VLC expresses the bound on the pseudo-velocity due to the jerk constraints (2.32):

$$\dot{s} \leq \dot{s}_{max,J}(s). \quad (2.55)$$

It represents an upper bound on any feasible trajectory in the  $s - \dot{s}$  plane. For each value of the path parameter, this maximum allowable pseudo-velocity due to the jerk constraints is defined as

Table 2.2: Actuator parameters

Torque limits	Medium jerk limits	High jerk limits	Low jerk limits
$T[Nm]$	Example 1 $\dot{T}_{ex1}[Nm/sec]$	Example 2 $\dot{T}_{ex2}[Nm/sec]$	Example 3 $\dot{T}_{ex3}[Nm/sec]$
$T_1 = 140$	$\dot{T}_{11} = 500$	$\dot{T}_{12} = 5000$	$\dot{T}_{13} = 140$
$T_2 = 140$	$\dot{T}_{21} = 500$	$\dot{T}_{22} = 2000$	$\dot{T}_{23} = 140$
$T_3 = 50$	$\dot{T}_{31} = 100$	$\dot{T}_{32} = 1000$	$\dot{T}_{33} = 50$

Figure 2.8: Control constraints in the  $s - \dot{s} - \ddot{s}$  space.

the  $\dot{s}$  value at which the feasible region in the  $\ddot{s} - \ddot{s}$  plane reduces to a point:

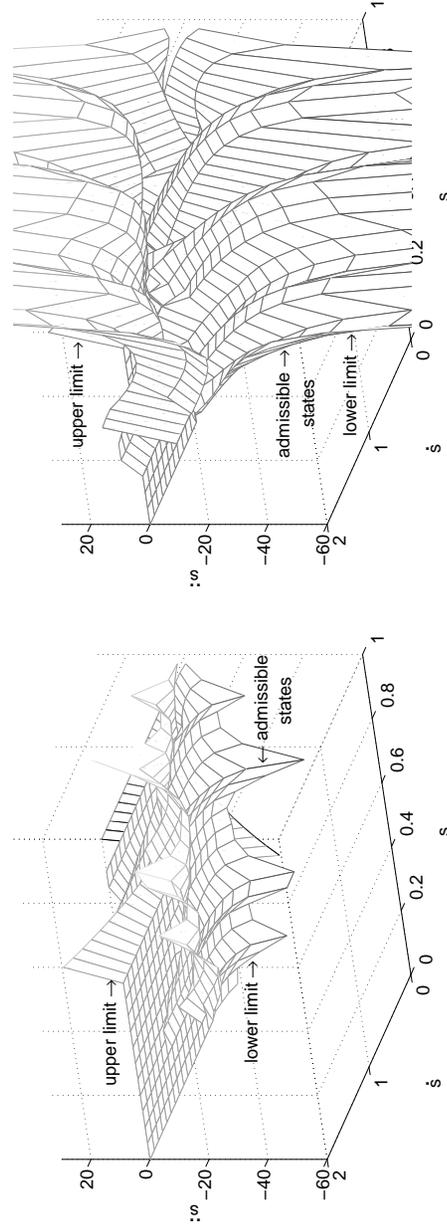
$$\ddot{s}_{min,J}(s, \dot{s}_{max,J}) = \ddot{s}_{max,J}(s, \dot{s}_{max,J}). \quad (2.56)$$

This maximum can be computed by a numerical search which determines the value of  $\dot{s}$  that satisfies Equation (2.56) for a given value of  $s$ .

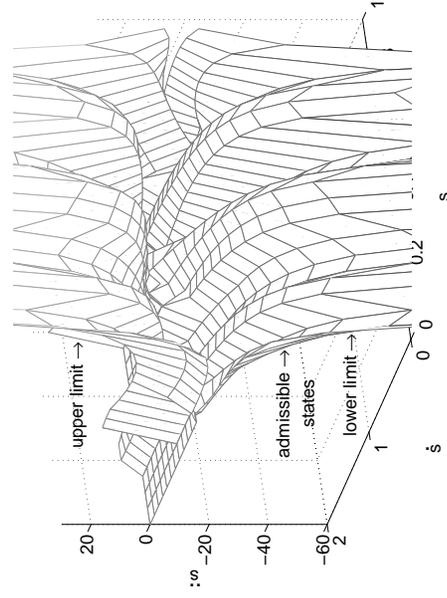
### 2.3.5 Admissible States

In the formulation of the SPCTOM problem proposed herein, the actuator jerk limits are

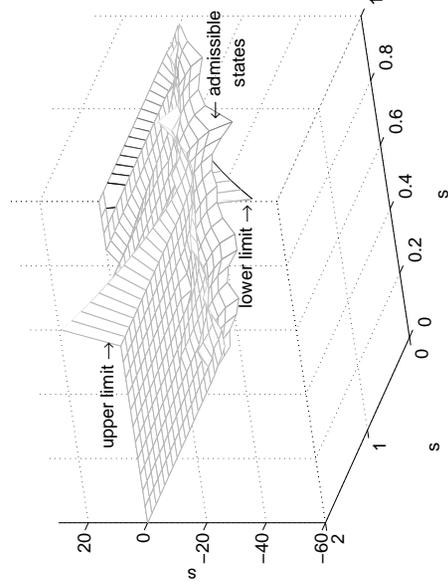
imposed as a means for adjusting the smoothness of the trajectory. Hence, they are independent of the actuator torque limits. This independence is shown in Figure 2.9, where the limiting surfaces due to the jerk bounds are plotted for the three example actuator jerk limits in Table 2.2. In Figure 2.10, the torque and the jerk constraints are plotted in the phase space for all three examples.



(a) Example 1 (medium jerk limits)



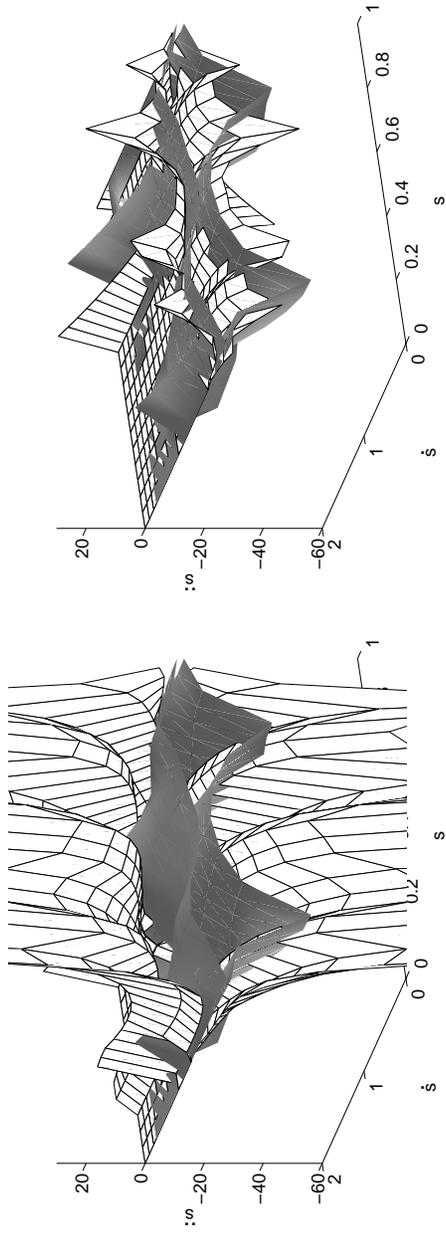
(b) Example 2 (high jerk limits)



(c) Example 3 (low jerk limits)

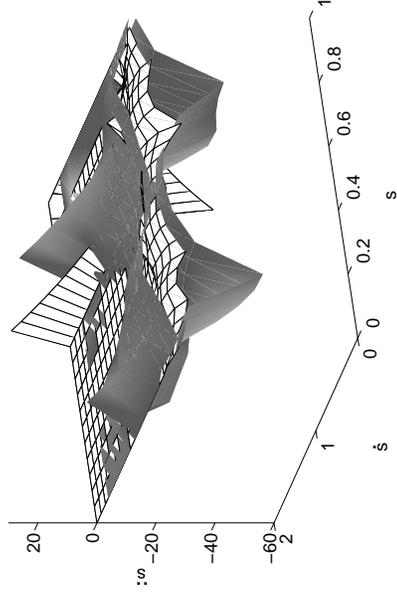
Figure 2.9: Different jerk limits in the  $s - \dot{s} - \ddot{s}$  space.

The independence of the actuator torque and jerk limits is reflected in a new constraint on the



(a) Example 1 (medium jerk limits): meshed surfaces - jerk limits ; shaded surfaces - torque limits.

(b) Example 2 (high jerk limits): meshed surfaces - jerk limits ; shaded surfaces - torque limits.



(c) Example 3 (low jerk limits): meshed surfaces - jerk limits ; shaded surfaces - torque limits.

Figure 2.10: Admissible states in the  $s - \dot{s} - \ddot{s}$  space.

pseudo-velocity:

$$\dot{s} \leq \min \{ \dot{s}_{max,T}(s), \dot{s}_{max,J}(s) \}, \quad (2.57)$$

and a new constraint on the pseudo-acceleration:

$$\max \{ \ddot{s}_{min,T}(s), \ddot{s}_{min,J}(s) \} \leq \ddot{s} \leq \min \{ \ddot{s}_{max,T}(s), \ddot{s}_{max,J}(s) \}. \quad (2.58)$$

Equation 2.57 defines a global velocity limit curve, called the *smooth motion velocity limit curve* (SMVLC). In the  $s - \dot{s}$  plane, the SMVLC is an upper bound on any feasible trajectory. Herein, the SMVLC is computed by a numerical search through bisection.

The SMVLC corresponding to the three examples in Table 2.2 are plotted in Figure 2.11. Figure 2.11 demonstrates that the SMVLC is determined by a combination of both actuator torque and jerk limits. Depending on the restrictions of the jerk limits, they can determine the SMVLC almost entirely, as shown in the third example, or they can have little influence on it, as shown in the second example.

### 2.3.6 System Dynamics

For the reduced system of order 3 where the path constraints (2.5) are incorporated into the dynamics (2.8), the states of the system are  $\mathbf{x} = (s \quad \dot{s} \quad \ddot{s})^T$ , while the *pseudo-jerk*  $\ddot{s}$  is the scalar control  $u$ .

The smooth time-optimal path-constrained trajectory planning problem can be formulated now as follows. Determine a scalar control:

$$u : [t_0, t_f] \longrightarrow U \subset \mathbf{R}, \quad (2.59)$$

where  $U$  is the set of admissible controls, such that:

$$\min_u J = \int_0^{t_f} 1 dt, \quad (2.60)$$

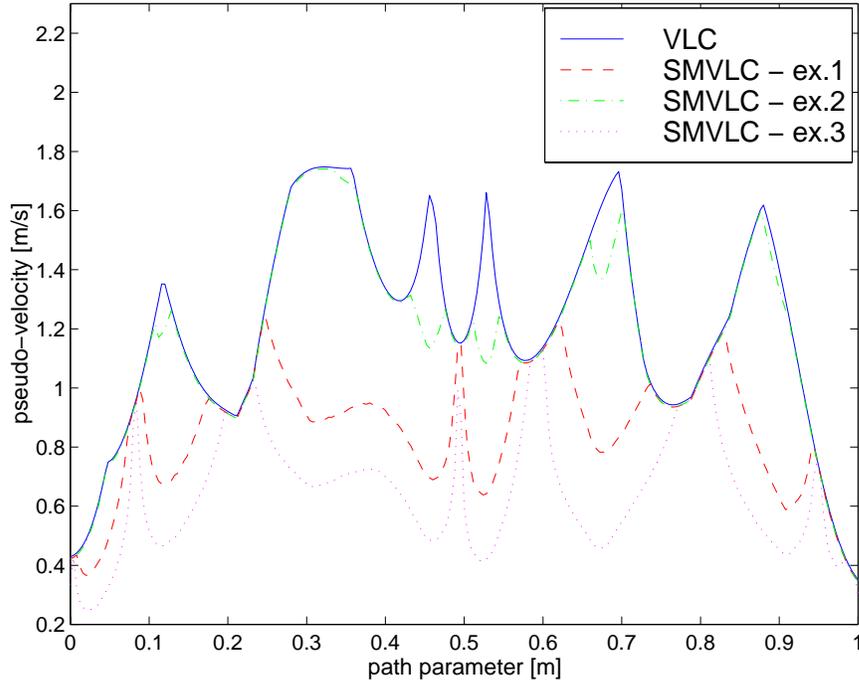


Figure 2.11: SMVLC for different actuator jerk limits.

subject to the system dynamics:

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \cdot \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \cdot u \quad (2.61)$$

the boundary conditions:

$$\begin{aligned} \mathbf{x}_0 &= (s_0 \quad \dot{s}_0 \quad \ddot{s}_0)^T \\ \mathbf{x}_f &= (s_f \quad \dot{s}_f \quad \ddot{s}_f)^T, \end{aligned} \quad (2.62)$$

the state inequality constraints:

$$\dot{s} \leq \min\{\dot{s}_{max,T}(s), \dot{s}_{max,J}(s)\}, \quad (2.63)$$

$$\max\{\ddot{s}_{min,T}(s, \dot{s}), \ddot{s}_{min,J}(s, \dot{s})\} \leq \ddot{s} \leq \min\{\ddot{s}_{max,T}(s, \dot{s}), \ddot{s}_{max,J}(s, \dot{s})\}, \quad (2.64)$$

and the state-dependent control inequality constraints:

$$\ddot{s}_{min}(s, \dot{s}, \ddot{s}) \leq \ddot{s} \leq \ddot{s}_{max}(s, \dot{s}, \ddot{s}). \quad (2.65)$$

In this final formulation, the system dynamics is described by a system of three first order differential equations with constant coefficients. Thus, the complexity of the TOC problem is shifted from the system dynamics to the nonlinear state-dependent state and control constraints.

## 2.4 Summary

PCTOM trajectories are not suited for direct implementation in an industrial application because they require discontinuous actuator torques. A physical robot has unmodelled link flexibility and actuator dynamics. In this situation, bang-bang joint torques induce arm vibrations. These characteristics decrease the tracking performance in path following applications resulting in strain on the actuators and the mechanical structure. Wear becomes a concern.

Recent research has addressed the need for smooth actuator torques both at the trajectory planning and at the trajectory tracking level. However, none of the reviewed works proposes a method for smoothing the torques individually, based on the dynamic characteristics of the robot and the path.

Herein, the path-constrained time-optimal trajectory planning problem is reformulated such that the actuator jerks are the controls and limits on both joint torques and joint jerks are considered. The actuator jerk limits may be chosen such that different degrees of smoothness are imposed on the actuator torques.

The independence of the torque and jerk limits is reflected in the SMVLC, which is determined by both limits. The SMVLC is computed herein using a bisection search.

## Chapter 3

# Methodology

In this chapter, a methodology is proposed for solving the problem formulated in Chapter 2. Namely, determine the trajectory that drives the manipulator along a prescribed path from a given initial state to a final desired state in minimum time, subject to limits on the actuator torques and their rates of variation, the actuator jerks. This is the SPCTOM trajectory planning problem.

Herein, this TOC problem is transformed into a nonlinear parameter optimization problem. Previous work on solving such optimal control problems through nonlinear programming techniques is reviewed and the flexible tolerance method (FTM) [27] is identified as a suitable candidate method for solving the SPCTOM problem. The formulation of the nonlinear optimization is presented and an overview of the FTM is given.

### 3.1 SPCTOM Trajectory Characteristics

The SPCTOM problem as defined in Section 2.3.6 is a TOC problem for a first order linear system with constant coefficients, nonlinear state-dependent state and control constraints and preimposed initial and final states. Such TOC problems have been solved either by applying the PMP to derive the necessary conditions for optimality and then using multiple shooting methods to solve the resulting two point boundary value problem (TPBVP) [6] or by a search for the switching points, using either dynamic programming [56] or specific algorithms [5], [45], [54], [59].

Two difficulties arise in the application of these approaches in the present case. First, the

complexity of the dynamic programming algorithms grows exponentially with the phase space dimension, rendering the method infeasible for more than two dimensions. As defined, the SPCTOM problem has a three dimensional phase space. Second, the other two approaches (based on PMP and the search for the switching points) depend on the bang-bang or bang-singular-bang structure of the optimal controls. This structure has been proven using results from OCT regarding systems with state dependent control constraints [35]. However, no results have been proven using OCT concerning the necessary optimality conditions for systems with state and control constraints which are independently active. Thus, for the SPCTOM problem, it is not guaranteed that the optimal controls take values only on the boundary of the admissible set.

Hereafter, the SPCTOM trajectory planning problem will be analyzed and solved in the  $s - \dot{s}$  plane. By doing so, another advantage is obtained besides surmounting the difficulties presented above. Namely, in the  $s - \dot{s}$  plane both end points are fixed, while in the time domain the final point is free. Thus, the TOC problem lends itself to a nonlinear parameter optimization problem in this phase plane.

To state the optimization problem, two results will be needed : (i) first, that any feasible trajectory, i.e., a trajectory complying with the constraints (2.63)-(2.65), is a smooth curve in the  $s - \dot{s}$  plane, i.e., it has a continuous first derivative; and, (ii) second, that in the  $s - \dot{s}$  phase plane, both the trajectory slopes at the path end-points and the height of the trajectory, i.e., the pseudo-velocity along the path, affect the motion time. These two results are established below.

The smoothness of a feasible trajectory can be proved by considering the control constraints (2.65). These inequalities impose that, at any point along the trajectory, with the exception of the first and last point (when the manipulator is starting from or stopping at rest), the pseudo-jerk  $\ddot{s}$  be limited. Then, the pseudo-acceleration  $\dot{s}$  is continuous along the entire trajectory and so is the pseudo-velocity  $s$ . This result can be established considering a feasible trajectory in the  $s - \dot{s}$  plane. Such a trajectory is a curve  $s(\dot{s})$ . Therefore, its first derivative is  $\frac{ds}{d\dot{s}}$ . Considering that:

$$\ddot{s} = \frac{d\dot{s}}{ds} \cdot \frac{ds}{dt} = \dot{s} \cdot \frac{d\dot{s}}{ds} \quad (3.1)$$

then:

$$\frac{d\dot{s}}{ds} = \frac{\ddot{s}}{\dot{s}} \quad . \quad (3.2)$$

As both  $\ddot{s}$  and  $\dot{s}$  are continuous functions along the entire trajectory, it follows that  $\frac{d\dot{s}}{ds}$  is continuous along the entire trajectory. The end points present no difficulty, since at these points both the pseudo-velocity and the pseudo-acceleration are zero. Thus, the values of  $\frac{d\dot{s}}{ds}$  at these points can be obtained as:

$$\left(\frac{d\dot{s}}{ds}\right)_0 = \lim_{s \searrow s_0} \frac{d\dot{s}}{ds} \quad (3.3)$$

and

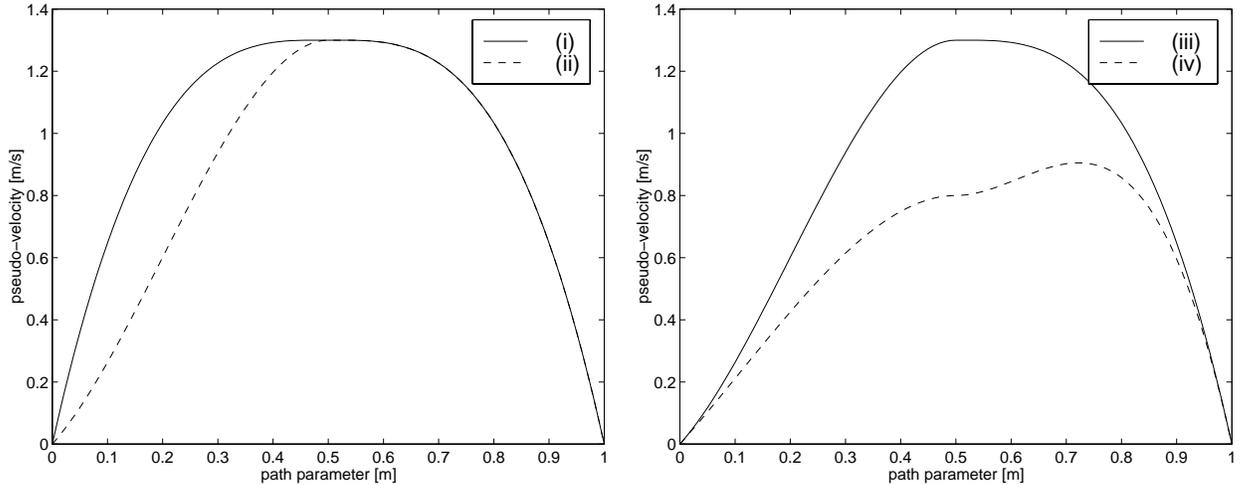
$$\left(\frac{d\dot{s}}{ds}\right)_f = \lim_{s \nearrow s_f} \frac{d\dot{s}}{ds} \quad (3.4)$$

The second result needed for solving the SPCTOM problem is that the optimal trajectory motion time is determined both by its end slopes and its height in the phase plane. The influence of each of these two elements is illustrated in Figure 3.1(a) and 3.1(b). In Figure 3.1(a), two trajectories are shown that differ from each other over some path parameter interval due to the different initial slopes in the phase plane. Considering the identity:

$$t_f = \int_{s_0}^{s_f} \frac{ds}{\dot{s}} \quad , \quad (3.5)$$

one can see that trajectory (i) results in a shorter motion time than trajectory (ii). Similarly, in Figure 3.1(b), the two trajectories have the same slopes at the end points, but (iii) is higher than (iv) over the path parameter interval, hence it has a shorter motion time. The combined effect of these two factors determines the motion time.

In order to determine the optimal smooth curve in the  $s - \dot{s}$  phase plane that does not violate actuator torque and jerk limits, two issues have to be addressed. First, the trajectory has to be approximated in some manner, such that it can be represented by a finite number of parameters. Second, an optimization technique must be selected to determine the optimal values of these parameters.



(a) Trajectories with different initial slopes.

(b) Trajectories with the same initial slopes

Figure 3.1: Influence of the end slopes and the height of the trajectory on the motion time.

Herein, the trajectory is approximated by cubic polynomials. By this choice, the smoothness requirement is satisfied. Furthermore, there is a great deal of material available on using such polynomials for function approximation.

## 3.2 Approximating Functions in Robotics

In optimal motion planning robotics research, approximating functions have been used for planning paths [4], [9], [10], [16], [20], [25], [30], [39], [40], [46], [53] and trajectories [28], [38], [48]. Using such functions, infinite dimensional optimization problems have been converted to finite dimensional optimizations. Theoretically, an infinite number of such approximating functions should be used. However, for practical purposes, approximations with a finite number of functions may still produce paths or trajectories sufficiently close to the true optimal ones.

However, such an approach leads to further problems, namely : (i) the choice of the basis functions used in the approximation; (ii) treatment of the equality and inequality constraints; (iii) the choice of the optimization algorithm. These problems are usually interlinked and as such the choice of one affects the choice of the others. Generally speaking, however, cubic splines have been the basis functions of choice for path approximation. They are selected for their properties of

smoothness, differentiability, and simplicity of parameterization. Also, spline theory shows that any smooth curve can be approximated to an arbitrary degree of accuracy using an adequate number of splines [18]. Moreover, a small number of splines is generally sufficient to approximate a smooth curve well.

This philosophy was originally applied for determining time optimal paths. The path was parameterized in some appropriate manner and the algorithm in [5] was used to compute the minimum time motion along this path. Then, an optimization method suited for incorporating the specific constraints was used to find the optimal path by varying the path parameters.

Rajan [46] assumed that the time optimal path should be smooth (sharp corners would require the manipulator to slow down) and that it could adequately be approximated by a low number of cubic splines. Hence, he parameterized the joint space path by one spline, discretized the parameter space and searched it to find the minimum. A gradient descent technique was applied afterwards to locate the minimum more precisely.

The same type of approach was extended to path planning in cluttered environments by Gilbert and Johnson [25], [30], Dubowsky et al. [20], Shiller and Dubowsky [52], [53], and Bobrow [4].

Gilbert and Johnson [25], [30] used basis functions at least twice differentiable for path parameterization. The path was assumed to be linear in the basis functions. Obstacle constraints were incorporated as special state constraints and the constrained optimization was transformed into an unconstrained one by use of penalty functions. The distance functions were shown to be continuously differentiable if parts were modeled by strictly convex sets and simple gradient formulas for them were derived. The optimal path was found through a Broyden-Fletcher-Goldfarb-Shanno-type algorithm that optimized the basis functions coefficients of an initially feasible path.

Dubowsky et al. [20] studied different path parameterizations. Thus, straight lines connected by circular arcs, perturbations about a straight line by a Fourier Series and cubic Bezier splines were used for path representation. Among them, splines were found to allow easy manipulation of the path's shape through a limited number of parameters. Obstacle and joint constraints were incorporated into the cost through weighted penalties. This approach had the desirable feature of keeping the manipulator some distance away from the constraints. For solving the parameter optimization problem, both a Quasi-Newton method and the Pattern Search method were applied.

A cubic B spline path parameterization in conjunction with the Pattern Search method was applied by Shiller and Dubowsky [52] to plan time optimal paths tangent to the Acceleration Lines. The same authors [53] extended then the technique to incorporate gripper and payload constraints. They found that the Pattern Search method, which did not require derivatives of the cost function, performed better in the majority of the problems.

Bobrow [4] used a similar approach for determining time optimal motions while incorporating joint limits, actuator torque limits, and collision avoidance constraints. He parameterized the path by cubic B splines, since B splines allow arbitrary smoothing of the path. The gradients of the objective function and the constraints were estimated numerically, and the vertices defining the path were varied till the minimum transversal time path was found. The problem was cast as a sequence of unconstrained minimizations with cubic interior penalty functions for the constraints. Each unconstrained minimization was solved by the Davidon-Fletcher-Powell (DFP) search technique.

Croft et al. [16] determined near-time optimal paths as a step towards on-line time optimal motion planning. They used a cubic Hermite spline for path parameterization together with a two-step optimization algorithm. At the first level, analytical formulations and heuristic rules were used to select the best initial path direction and an initial path approximation. At the second level, the flexible polyhedron method (FPM) was employed to determine the optimal path parameters. The near-time optimal path was chosen as the minimum curvature one, with the intent of avoiding sharp corners. As the analysis indicated that the cost was a smooth function of the two variable parameters, a Quasi-Newton method (e.g., the DFP method) was indicated as an alternative minimization technique.

Finally, at the path planning level, Martin and Bobrow [40] used cubic B splines for parameterizing a minimum effort joint space path. The path was assumed to depend linearly on the path parameters and the Newton-Euler formulation of robot dynamics was expressed in terms of the product of exponentials formula. This formulation allowed the authors to compute analytic gradients of the objective function with respect to the path parameters and thus a steepest descent technique was applied to find the optimal path.

Cubic [9], [28], [38] or quartic splines [9], [10] were also used for parameterizing joint space trajectories. However, in the case of trajectory planning, the TOC problem was not solved through nonlinear programming, but rather through scaling the trajectory subjected to joint, velocity,

acceleration, and jerk limits.

### 3.3 Proposed Method

Similar to the research reviewed in Section 3.2, the present work solves a TOC problem through a mathematical programming technique. That is, the SPCTOM trajectory is computed through a search in the  $s - \dot{s}$  plane for the minimum time smooth trajectory. However, in this approach, rather than parameterizing the Cartesian path or the joint space path, the state space trajectory is parameterized. By doing so, the TOC problem is formulated as a fixed end points problem which lends itself to solving through nonlinear optimization. Nevertheless, in using a state space trajectory parameterization, the choices of basis functions and of trajectory parameters must be justified.

#### 3.3.1 Basis Functions

As shown in Section 3.1, the SPCTOM trajectory is a smooth curve in the  $s - \dot{s}$  plane; i.e., it is  $C^2$  continuous. Hence, the lowest degree polynomials that can be used to approximate it are cubics.

Higher degree polynomials present no an advantage here, since the extra degrees of freedom they offer are not needed. Also, the optimal trajectory is obtained from a small number of splines. Therefore, the higher the polynomial degree of the splines, the more oscillatory they become. As a result, higher order polynomials will actually result in a poorer representation of the SPCTOM trajectory.

In view of this, cubic polynomials are chosen as basis functions herein. The base trajectory is obtained by splining them together through selected knot points. The coefficients of the cubic polynomials are computed using the Matlab Spline Toolbox. The selection of the knot points, as well as other properties of the polynomials, are discussed in the following section.

### 3.3.2 Optimization Variables

The parameters defining the trajectory are the pseudo-velocities of the end-effector at predetermined points along the Cartesian path (knot points) and the slopes of the trajectory in the  $s - \dot{s}$  plane at the path end points. Thus, the pseudo-velocities and the trajectory end slopes are the optimization variables in this problem. These variables control the motion time : the higher the knot points (as located in the phase plane), the shorter the trajectory transversal time. On the other hand, the end slopes control the speed at which the actuator torques leave or approach their static equilibrium values. Therefore, steeper slopes result in shorter motion time.

The trajectories shown in Figure 3.2<sup>1</sup> illustrate the influence of the trajectory parameters on the motion time. The solid line represents an exemplar trajectory. The dot-dashed line is the trajectory obtained when the initial trajectory slope was increased by 70%; the dashed line was obtained by increasing the final trajectory slope by 70%; and the dotted line was obtained by increasing the third knot-point pseudo-velocity by 70%.

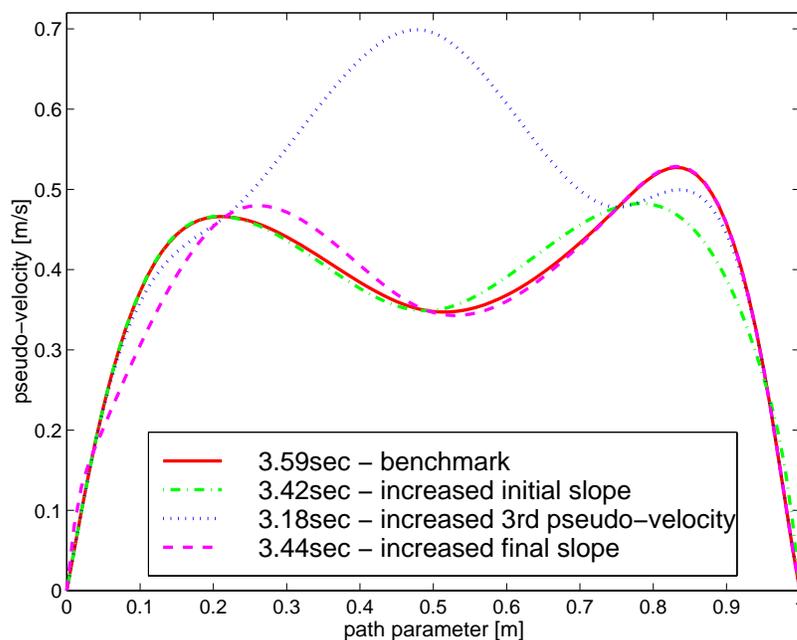


Figure 3.2: Sample trajectories in the  $s - \dot{s}$  plane and their motion times.

<sup>1</sup>All the figures in this chapter are plotted for the elbow manipulator with the dynamic parameters given in Table 2.1 moving along the parabolic path specified by Equations (2.1).

In order to define the spline basis functions, the location of the knot points along the path must be specified. Fixing the location of the knot points along the Cartesian path results in lost degrees of freedom in the search for the optimal path. However, this is a necessary step in transforming the TOC problem into a tractable optimization.

In this work, the location of the knot points is chosen to be the same as that of the PCTOM trajectory switching points (Figure 3.3). These switching points indicate where each joint actuator is, in turn, saturated due to the generalized dynamic forces of the robot's motion under PCTOM. Since PCTOM is the limit for SPCTOM, these switching points are, in the limit, the same for SPCTOM and provide a reasonable estimate of the location of the SPCTOM switching points along the parameterized path. Further work to improve these switching points is beyond the scope of this thesis.

Extra knot points could be chosen; however, as the number of the PCTOM trajectory switching points can be considerable (five for the PCTOM trajectory in Figure 3.3), the addition of more knot points would significantly increase the number of optimization variables (from seven to twelve in this example). For simpler PCTOM trajectories, such as those having a single switching point, using more knot points is a reasonable option which is investigated in Chapter 4.

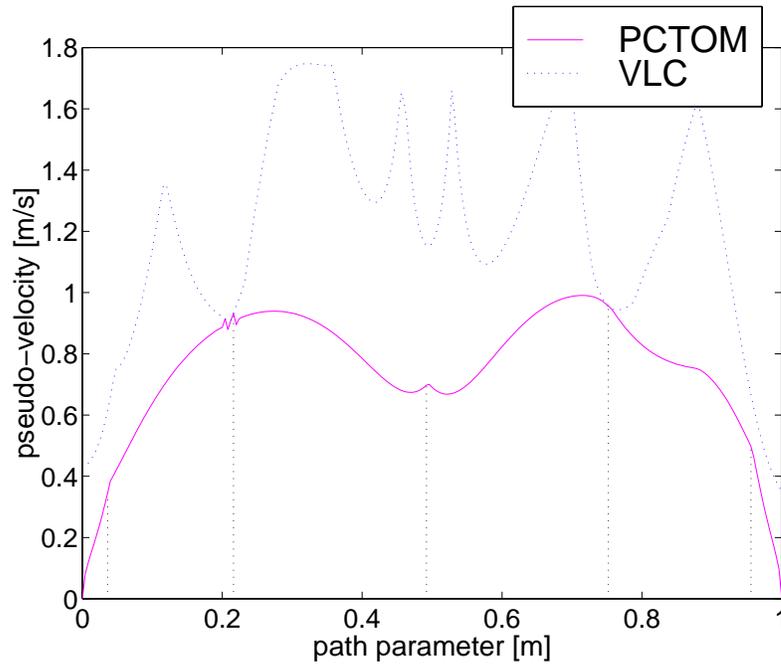


Figure 3.3: Switching points of the PCTOM trajectory.

The optimization variables are shown in Figure 3.4, for an exemplar splined trajectory. The knot points are indicated with the “o” symbol and the end slopes are marked with tangent lines at the beginning and end of the trajectory. The vector of optimization variables,  $\mathbf{x}$ , is defined as:

$$\mathbf{x} = \left( \frac{d\dot{s}}{ds} \Big|_0, \dot{s}_1, \dots, \dot{s}_p, \frac{d\dot{s}}{ds} \Big|_f \right)^T, \quad (3.6)$$

where the values with the index  $m$  correspond to the limiting PCTOM (the dashed line in Figure 3.4), while the other values correspond to the splined trajectory (the solid line), and  $p$  is the number of switching points of the PCTOM trajectory. The optimization variables are normalized since the end slopes vary over a much wider range than the pseudo-velocities. The normalization is performed with respect to the PCTOM trajectory because this trajectory always represents an upper bound for the smooth optimal motion, namely, time optimal motion without smoothness constraints.

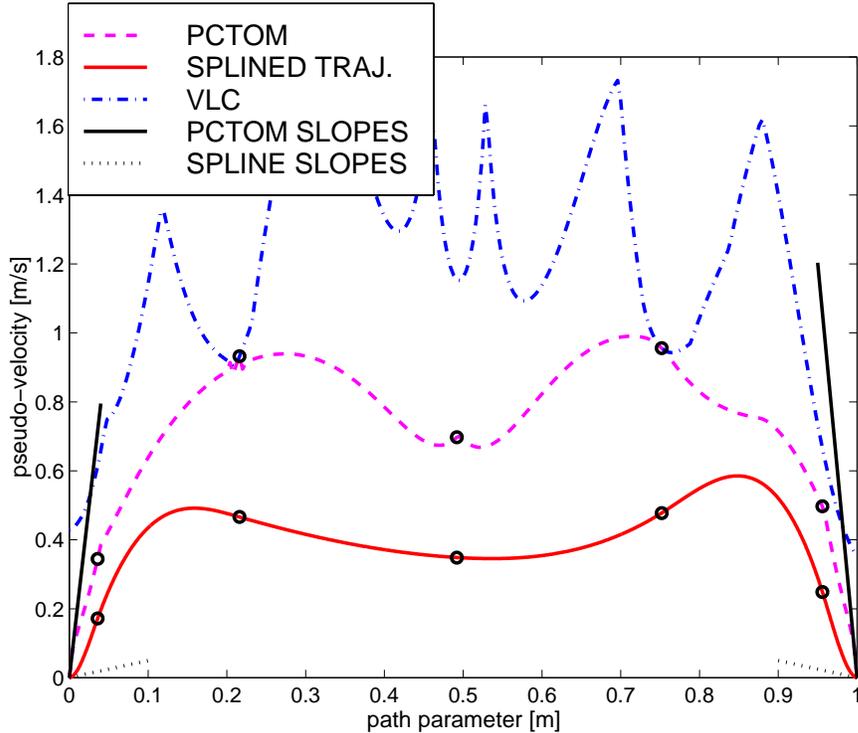


Figure 3.4: The optimization variables. The knot-points are indicated with the “o” symbol.

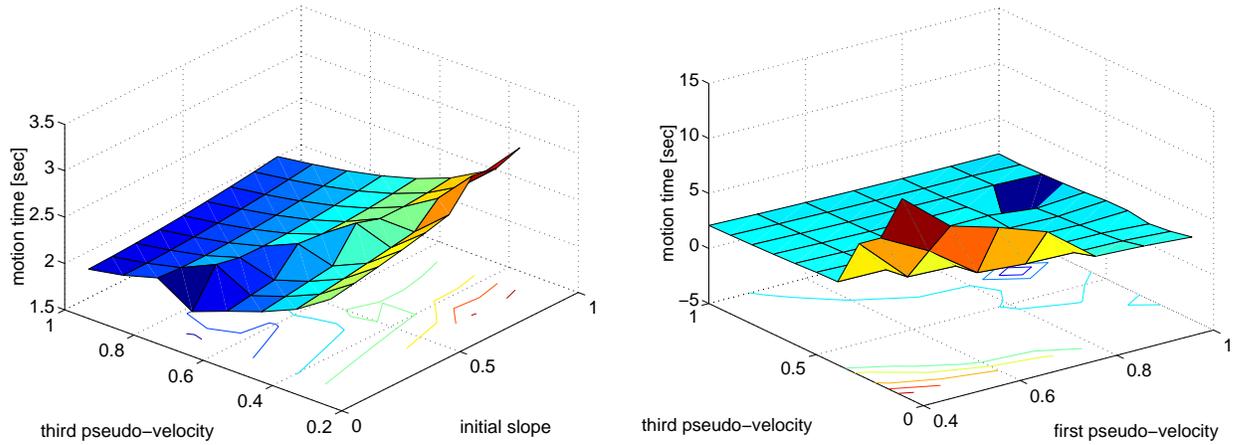


Figure 3.5: Motion time and its level curves with respect to several parameters varied over their ranges.

The selection of an optimization method depends on the relationship between the identified optimization vector,  $\mathbf{x}$ , and the cost function, namely, motion time. As is typically the case with trajectory optimization, the motion time is not linear in the variables. The relationship between the cost function and the parameters is illustrated in Figure 3.5, where the trajectory transversal time is plotted with respect to two of the variables over their range, namely from zero to one. Furthermore, there exist islands of *inadmissibility* in the parameter space. That is, there are points in the parameter space that result in trajectories that require negative pseudo-velocities. This phenomenon was first observed by Shin and McKay [55]. An example of such a case is shown in Figure 3.6.

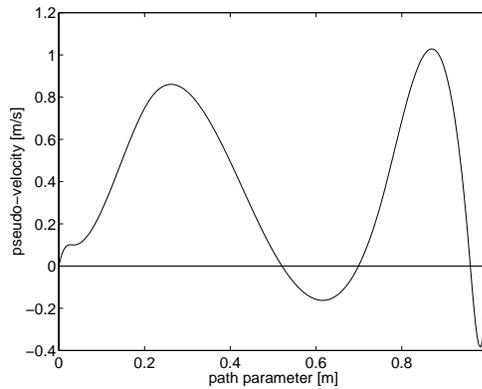


Figure 3.6: Plot of a phase plane trajectory resulting from a point in the inadmissible region of the parameter space. The resulting trajectory is physically meaningless (cannot be achieved).

Thus, the SPCTOM problem is recast as follows : the SPCTOM trajectory results from splining cubic polynomials in the  $s - \dot{s}$  phase plane through a set of optimized  $\mathbf{x}$  knot points. The knot point locations along the path are the same as the location of the switching points of the PCTOM trajectory. The end-effector pseudo-velocities at these locations and the trajectory slopes at the path end points are determined through an optimization process. The end-effector pseudo-velocities must all be positive, i.e., the trajectory must be admissible. The splined trajectory must be within actuator torque and actuator jerk limits and take minimum time.

### 3.3.3 Numerical Solution

The above problem is solved by finding the optimal trajectory end slopes and the pseudo-velocities for the knot points in the parameter space for the splined trajectory. Therefore, the numerical solution of the SPCTOM trajectory planning problem is an approximation of the infinite dimensional OCP by a  $p + 2$  dimensional optimization problem. The approximation is motivated by the complexity of the OCP.

There are a number of considerations involved in such a numerical solution for an OCP, namely:

- The incorporation of equality constraints, i.e. system dynamics.
- The treatment of inequality constraints, i.e. actuator torque and jerk constraints.
- The choice of a descent algorithm.
- The different functions and their derivatives evaluation.
- The determination of a good initial feasible approximate solution.
- The achievement of a reasonable computational cost.

Clearly, there is no algorithmic approach that best addresses these issues for any problem. Therefore, the method proposed herein is based on the particulars of the SPCTOM trajectory planning; namely:

- The cost function and the constraints are nonlinear functions of the trajectory parameters. Furthermore, they are not analytic functions, but can only be computed numerically.

- Objective function evaluations are expensive.
- Analytical derivatives of the cost function and the constraints are not available.
- The transversal time may not be differentiable with respect to the trajectory parameters, primarily because of the numerical approximations involved in its computation [53].
- The solution is expected to lie on the constraint boundary.

Two options exist for solving a constrained optimization: (i) transformation of the constrained optimization problem into an unconstrained problem by incorporating the constraints into the objective through penalties; (ii) directly solving the constrained optimization problem. Since a penalty function approach drives the solution away from the constraints, where the solution is expected to be found, a constrained optimization is more appropriate for planning SPCTOM trajectories. Furthermore, a method that does not require analytical or numerical derivatives is required. Finally, a method that minimizes objective function evaluations is important due to the computation expense of such evaluations.

In view of this, the optimization is carried out using the flexible tolerance method (FTM) in [27]. The FTM is a mathematical programming technique that improves the value of the objective function by using information provided by both feasible points, as well as certain nonfeasible points termed *near-feasible* points (Figure 3.7). As the search proceeds, the near-feasibility limits (determined by the tolerance  $\Phi$ ) are made more restrictive, until in the limit only feasible points are accepted. The details of the FTM, as well as the implementation of the algorithm, are presented in Appendix A. However, the FTM features that make it suited for determining SPCTOM trajectories are reviewed here. Namely:

- It allows for nonlinear objective function and nonlinear equality and/or inequality constraints.
- It transforms all equality and inequality constraints into one single inequality constraint, which results in a problem simplification.
- It does not require derivatives.
- It uses information about both feasible and near-feasible points, thus maintaining the search close to the constraint boundary.

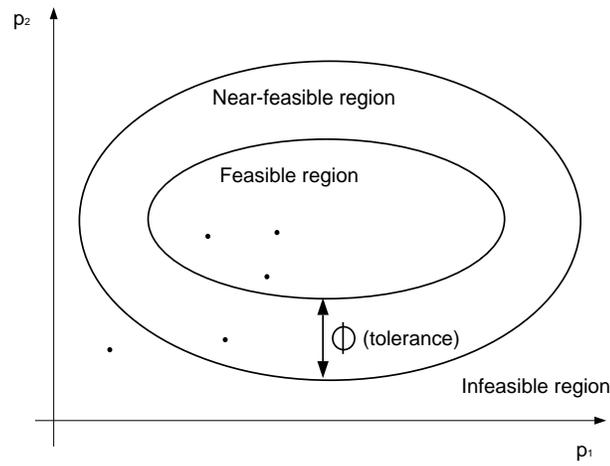


Figure 3.7: The parameter space in the Flexible Tolerance Method - Example for a two-dimensional optimization.

- It is computationally economical, because it does not spend a considerable amount of computation effort in satisfying rigorous feasibility requirements.
- It has successfully been applied to complex engineering problems with up to nine variables and forty-four constraints [15], [22].

As other constrained optimization techniques, the FTM requires an initial guess of a feasible solution. However, guessing such a solution is not difficult. If the path is feasible (namely, within the static and kinematic robot workspace), then there always exists a feasible trajectory (although it could be infinitely slow).

The question of whether the search has returned the global optimum has no definite answer. This is the case with any nonlinear optimization. In this case the objective function, i.e. motion time, is a complex function of many variables and it appears to have many local minima. To increase the confidence that the global optimum has been achieved, it is proposed herein to rerun the optimization starting with a number of different initial guesses.

## 3.4 Summary

This chapter developed an optimization solution for the SPCTOM trajectory planning problem formulated in Chapter 2. The method proposed is based on solving an infinite dimensional OCP by a finite dimensional optimization. This approach uses cubic polynomials splined together through knot points to approximate the SPCTOM trajectory in the  $s - \dot{s}$  plane. Cubic splines have been used in robotics before for approximating optimal paths or optimal trajectories in the time domain. However, this is the first time that they have been used for trajectory interpolation in the phase plane.

The knot point location along the Cartesian path is chosen to be the same as the location of the switching points of the PCTOM trajectory. This provides a compromise between an acceptable number of optimization variables and the dynamic characteristics of the robot along the given path.

The computation of the optimal trajectory reduces to a search for the optimal knot points in the parameter space. The search is performed using the FTM since it allows for nonlinear cost function, as well as equality and/or inequality constraints. Furthermore, the algorithm has low requirements on an initial feasible trajectory and can find optimal solutions that are on the constraint boundary.

As with any other nonlinear optimization algorithm, no guarantee exists that the global optimal trajectory has been determined. Therefore, the search for the optimal parameters is performed multiple times, each time starting from another point in the parameter space. The trajectory with minimum transversal time is then chosen as the computed SPCTOM trajectory.

## Chapter 4

# Simulations

In this chapter, SPCTOM trajectories are planned using the methodology developed in Chapter 3. As well, the effect of selecting more knot points than the number of switching points for the related PCTOM trajectory is investigated. A planning level comparison is performed between the SPCTOM trajectory, the PCTOM trajectory and a standard smooth polynomial trajectory. Finally, the performance of each of these trajectories is compared using a simulated robot and controller model.

The simulations presented herein serve several purposes: (i) they demonstrate the methodology developed in Chapter 3; (ii) they allow the assessment of whether or not the proposed method is suitable for computing SPCTOM trajectories; (iii) they establish a correlation between the limits imposed on the actuator jerks and the quality of the trajectory (as reflected by the tracking accuracy); and (iv) they permit an evaluation of the performance of a SPCTOM trajectory compared to other trajectories.

The trajectory performance is evaluated from the point of view of both improving task productivity and reducing the strain on the robot and the controller. Thus:

- Motion time reflects task productivity.
- Tracking accuracy determines the quality of the trajectory.
- Actuator torques smoothness reflects the robot and actuator strain.

## 4.1 SPCTOM Planning

In this section, the methodology developed in Chapter 3 is demonstrated through simulations. The effect of choosing more knot points than the number of switching points of the related PCTOM trajectory is studied.

The robot considered throughout this section is the elbow manipulator introduced in Chapter 2. Again, the robot dynamic parameters, the torque limits, and the two exemplar paths are taken from [45]. The robot data is repeated here for convenience (Tables 4.1 and 4.2). The dynamic model of the robot is detailed in Appendix B. The three different jerk limits have been selected to demonstrate the planning results. The “low jerk limits” represent a very smooth motion with low stress on the manipulator. The “high jerk limits” represent the other end of the range, approaching PCTOM motion in the limit. The “medium jerk limits” fill out the range.

Table 4.1: Robot parameters (repeated).

Link [m]	C.O.M. [m]	Mass [kg]	$I_x[kgm^2]$	$I_y[kgm^2]$	$I_z[kgm^2]$
$l_1 = 0$	$l_{c1} = 0.05$	$m_1 = 0$	$I_{x1} = 0$	$I_{y1} = 5$	$I_{z1} = 0$
$l_2 = 0.75$	$l_{c2} = 0.2$	$m_2 = 6.6$	$I_{x2} = 0.1$	$I_{y2} = .6$	$I_{z2} = 0.6$
$l_3 = 0.75$	$l_{c3} = 0.15$	$m_3 = 4.2$	$I_{x3} = 0.02$	$I_{y3} = .2$	$I_{z3} = 0.3$

Table 4.2: Actuator parameters (repeated).

Torque limits	Medium jerk limits	High jerk limits	Low jerk limits
$T[Nm]$	Example 1 $\dot{T}_{ex1}[Nm/sec]$	Example 2 $\dot{T}_{ex2}[Nm/sec]$	Example 3 $\dot{T}_{ex3}[Nm/sec]$
$T_1 = 140$	$\dot{T}_{11} = 500$	$\dot{T}_{12} = 5000$	$\dot{T}_{13} = 140$
$T_2 = 140$	$\dot{T}_{21} = 500$	$\dot{T}_{22} = 2000$	$\dot{T}_{23} = 140$
$T_3 = 50$	$\dot{T}_{31} = 100$	$\dot{T}_{32} = 1000$	$\dot{T}_{33} = 50$

### 4.1.1 Parabolic Path

In the following examples, the end-effector moves along the parabolic path given by Equa-

tions (2.1), namely:

$$\begin{aligned}x(s) &= 0.5 \\y(s) &= 20s^3 - 30s^2 + 10s \\z(s) &= s - 0.5 \\s &= 0, \dots, 1.\end{aligned}$$

The PCTOM trajectory has five switching points (Figure 3.3). The location of these points in the  $s - \dot{s}$  plane is given in Table 4.3. The slopes of the PCTOM trajectory in this plane are 19.875 at the starting point and  $-24.07$  at the ending point. As explained in Chapter 3, these values are used for scaling the SPCTOM variables.

Table 4.3: Location of the PCTOM switching points in the  $s - \dot{s}$  plane.

	Switching point				
	1	2	3	4	5
Path parameter	$s_1 = 0.036$	$s_2 = 0.216$	$s_3 = 0.492$	$s_4 = 0.752$	$s_5 = 0.956$
Pseudo-velocity	$\dot{s}_1 = 0.344$	$\dot{s}_2 = 0.932$	$\dot{s}_3 = 0.697$	$\dot{s}_4 = 0.955$	$\dot{s}_5 = 0.497$

Hence, in this example, the vector of optimization variables for SPCTOM is:

$$\mathbf{x} = \left( \frac{d\dot{s}}{ds_0} \quad \dot{s}_1 \quad \dot{s}_2 \quad \dot{s}_3 \quad \dot{s}_4 \quad \dot{s}_5 \quad \frac{d\dot{s}}{ds_f} \right)^T. \quad (4.1)$$

The SPCTOM trajectories are determined for the three different jerk limits in Table 4.2. These trajectories are obtained through a nonlinear programming method. The cost function, namely, the motion time, is multi-modal. Therefore, it is not possible to decide if the global optimum has been determined. To improve the chance of determining the global minimum, four different initial guesses are used. In the end, the optimization is restarted using three of these optima (the best one is excluded) as initial guess values. The SPCTOM trajectory is computed by splining cubics through the optimal knots obtained in the optimization. The motion times obtained after each optimization, as well as after the restart, are shown in Table 4.4.

The PCTOM trajectory and the SPCTOM trajectories for the three examples (i.e. the fastest trajectory in each case), together with the corresponding actuator torques and jerks, are plotted in

Figures 4.1-4.4.

Table 4.4: Optimal motion times for the elbow manipulator along the parabolic path [sec].

Initial guess	SPCTOM			PCTOM
	Example 1 Medium jerk limits	Example 2 High jerk limits	Example 3 Low jerk limits	
1	$t_1 = 3.03$	$t_2 = 1.91$	$t_3 = 4.08$	$t = 1.72$
2	$t_1 = 3.45$	$t_2 = 2.05$	$t_3 = 4.24$	
3	$t_1 = 3.85$	$t_2 = 2.61$	$t_3 = 5.07$	
4	$t_1 = 2.80$	$t_2 = 2.06$	$t_3 = 4.97$	
Restart	$t_1 = 2.74$	$t_2 = 1.91$	$t_3 = 3.93$	

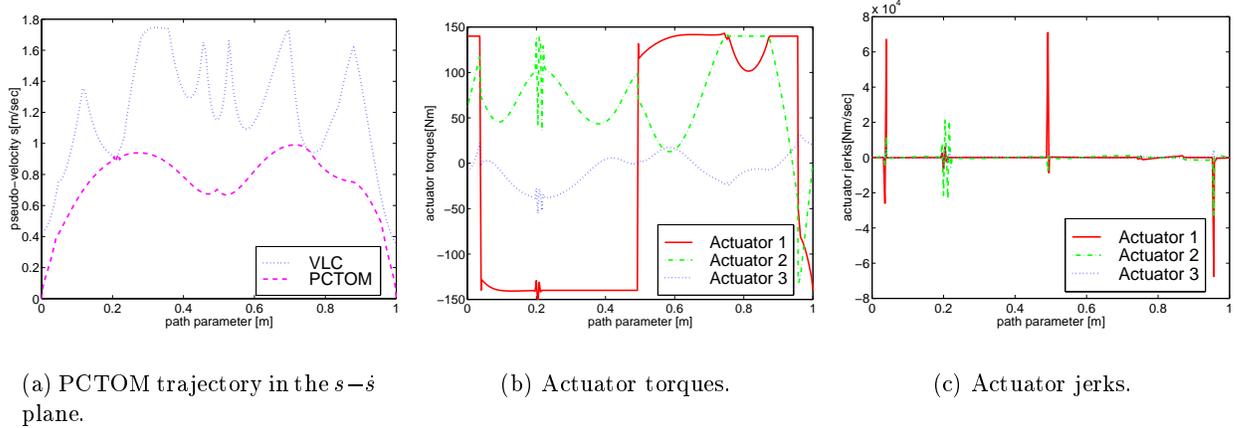


Figure 4.1: PCTOM for the elbow manipulator along the parabolic path.

As shown in Table 4.4, by restarting the optimization either the previous best trajectory is obtained or a new trajectory with a lower motion time is found. No guarantees exist that the trajectory obtained after the restart is the globally optimal one. However, if, after the restart, no or very little improvement is obtained, the confidence that the global optimum has been achieved increases. As this is the case with the three examples presented here, it is expected that the trajectories found approach the SPCTOM trajectory in each case.

The second example (Figure 4.3) further supports the confidence that the global optima have been obtained. In this example, the actuator jerk limits are very high. Hence, the trajectory is determined by the limits on the actuator torques and the increase in motion time is very small compared to the PCTOM: 1.91 seconds compared to 1.72 seconds. Although one would expect both

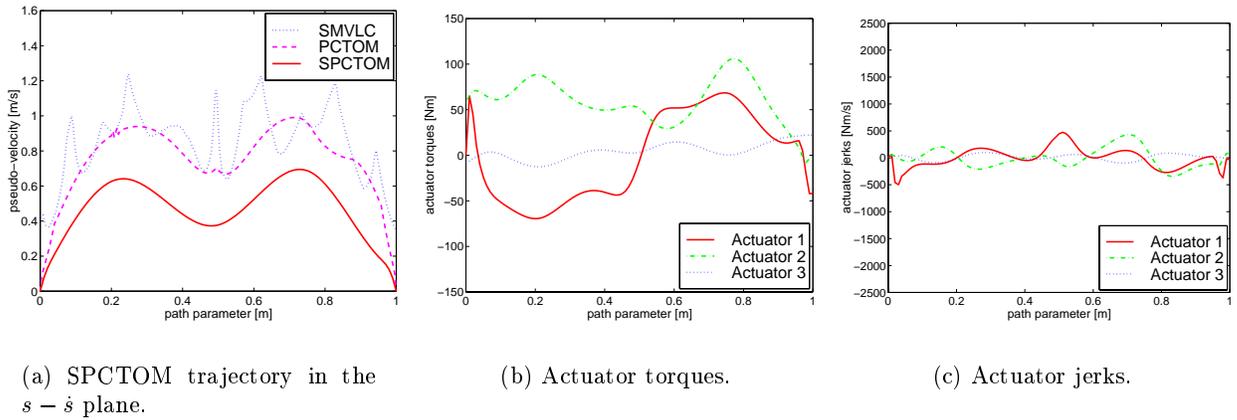


Figure 4.2: SPCTOM for the elbow manipulator along the parabolic path - Example 1 (medium jerk limits).

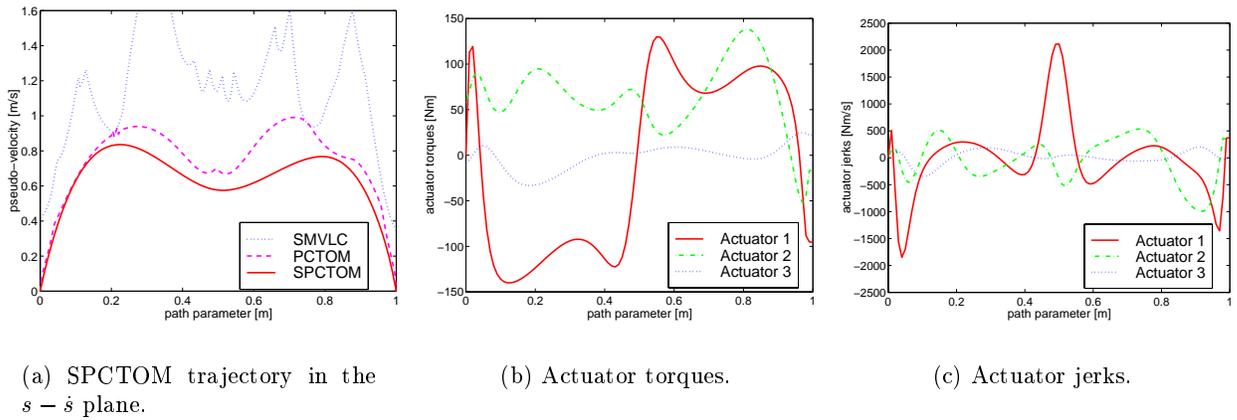


Figure 4.3: SPCTOM for the elbow manipulator along the parabolic path - Example 2 (large jerk limits).

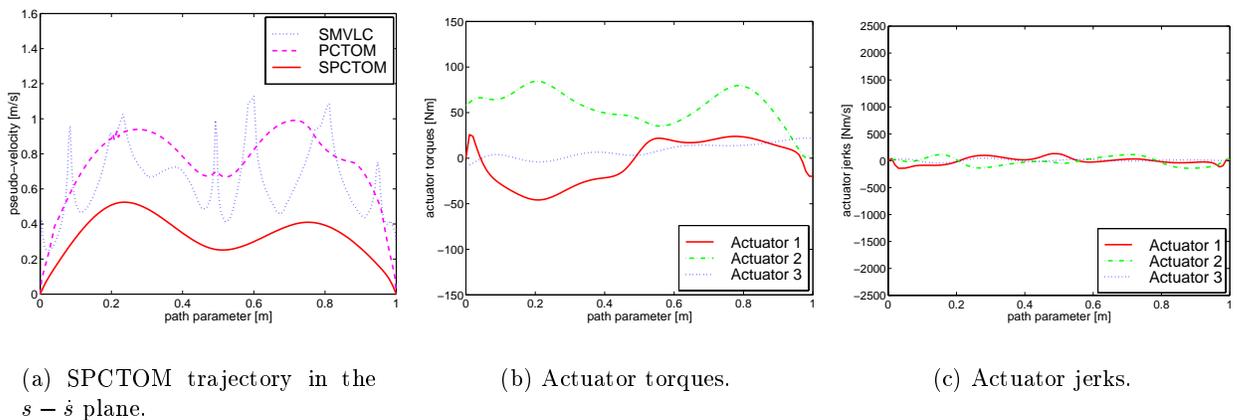


Figure 4.4: SPCTOM for the elbow manipulator along the parabolic path - Example 3 (low jerk limits).

trajectories to yield the same motion times, there are two reasons for the increase in motion time for the SPCTOM: (i) the limitations of the parameterization chosen in the  $s - \dot{s}$  phase plane and (ii) the significant decrease in peak actuator jerks for SPCTOM (solid lines) compared to PCTOM (dotted lines), as shown in Figure 4.5. The logarithmic pseudo-jerk scale in Figure 4.5 shows that the peak PCTOM jerk is almost two orders of magnitude greater than the peak SPCTOM jerk.

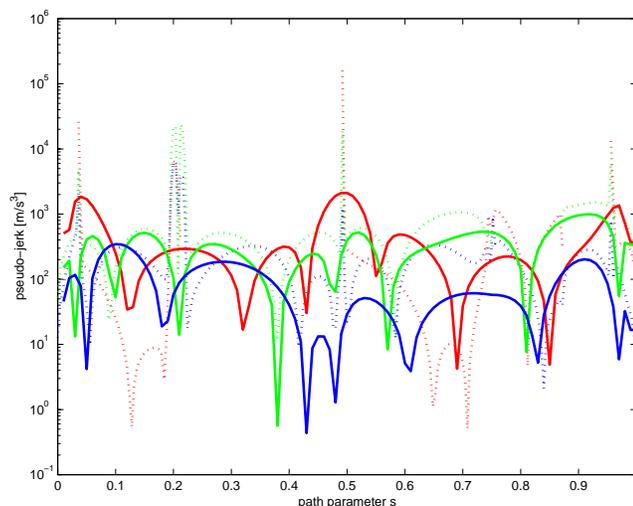


Figure 4.5: Actuator jerks for the SPCTOM in Example 2 (solid lines) and the PCTOM (dotted lines) for the elbow manipulator along the parabolic path.

### 4.1.2 Circular Path

In this section, a second example is presented. This second example is used for two purposes: (i) to illustrate a different path; and (ii) to investigate the influence of the number of knot points on the results of the optimization.

The robot is the same elbow manipulator as before, but the path is chosen to be a circular path

given in parametric form as:

$$\begin{aligned}
 x(s) &= 0.85 - 0.65 \cdot \cos(5.5s + 0.4) \\
 y(s) &= 0.65 \cdot \sin(5.5s + 0.4) \\
 z(s) &= 0.1 \\
 s &= 0 \dots 1.
 \end{aligned} \tag{4.2}$$

Again, the path is taken from [45], but is scaled such that the path parameter varies between 0 and 1.

The PCTOM in this example is shown in Figure 4.6. As shown in the figure, there is only one switching point. That is, the minimum time motion is bang-bang: accelerating with maximum allowable acceleration and then decelerating with maximum allowable deceleration. The transversal time of the PCTOM trajectory is  $t = 1.07$  sec.

Following the methodology proposed in Chapter 3, the base trajectory representing the SPCTOM consists of two cubics splined together at  $s = 0.496$  along the Cartesian path and the vector of optimization variables is:

$$\mathbf{x} = \left( \begin{array}{ccc} \frac{ds}{ds_0} & \dot{s}_1 & \frac{ds}{ds_f} \end{array} \right)^T. \tag{4.3}$$

The SPTOM trajectories are determined for all three actuator jerk limits in Table 2.2 following the procedure outlined in the previous section. Namely, for each set of jerk limits, the optimization is started with four different initial guesses. The best trajectory is discarded and the other three are used to restart the process. The results are given in Table 4.5. The PCTOM trajectory, the SPCTOM trajectories, and the corresponding optimal actuator torques and jerks, are plotted in Figures 4.7-4.10.

As Table 4.5 shows, the increase in motion time for the SPCTOM trajectory as compared to the PCTOM trajectory is significant. There are two factors that result in a SPCTOM that is slower than the PCTOM: (i) the SPCTOM trajectory parameterization; and (ii) the substantially lower peak actuator jerks for SPCTOM (Figure 4.11).

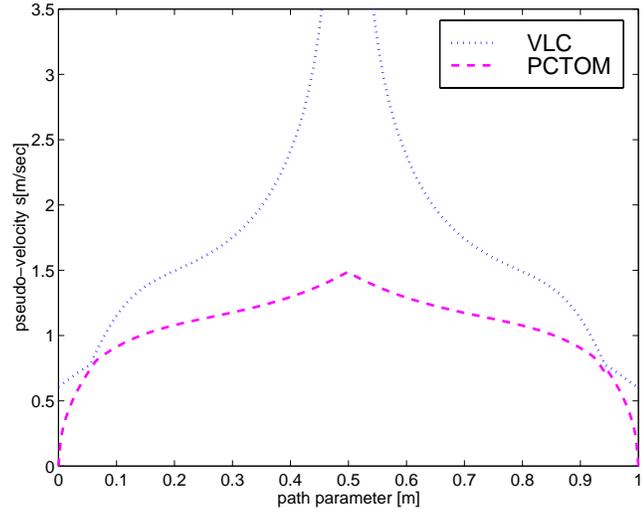
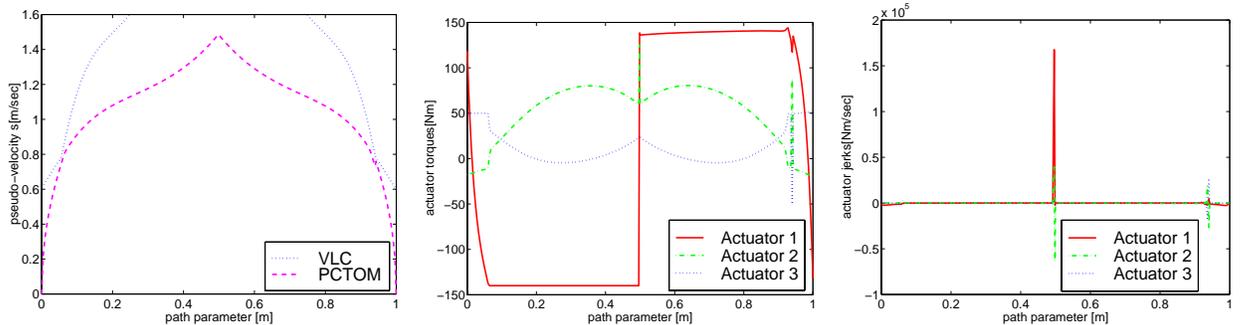


Figure 4.6: PCTOM in the  $s - \dot{s}$  phase plane for the elbow manipulator along the circular path.

Table 4.5: Optimal motion times for the elbow manipulator along the circular path [sec].

Initial guess	SPCTOM			PCTOM
	Medium jerk limits Example 1	High jerk limits Example 2	Low jerk limits Example 3	
1	$t_1 = 2.73$	$t_2 = 1.56$	$t_3 = 3.31$	$t = 1.07$
2	$t_1 = 2.14$	$t_2 = 1.57$	$t_3 = 2.91$	
3	$t_1 = 2.18$	$t_2 = 1.57$	$t_3 = 3.05$	
4	$t_1 = 2.26$	$t_2 = 1.59$	$t_3 = 2.90$	
Restart	$t_1 = 2.13$	$t_2 = 1.57$	$t_3 = 2.91$	



(a) PCTOM trajectory in the  $s - \dot{s}$  plane.

(b) Actuator torques.

(c) Actuator jerks.

Figure 4.7: PCTOM for the elbow manipulator along the circular path.

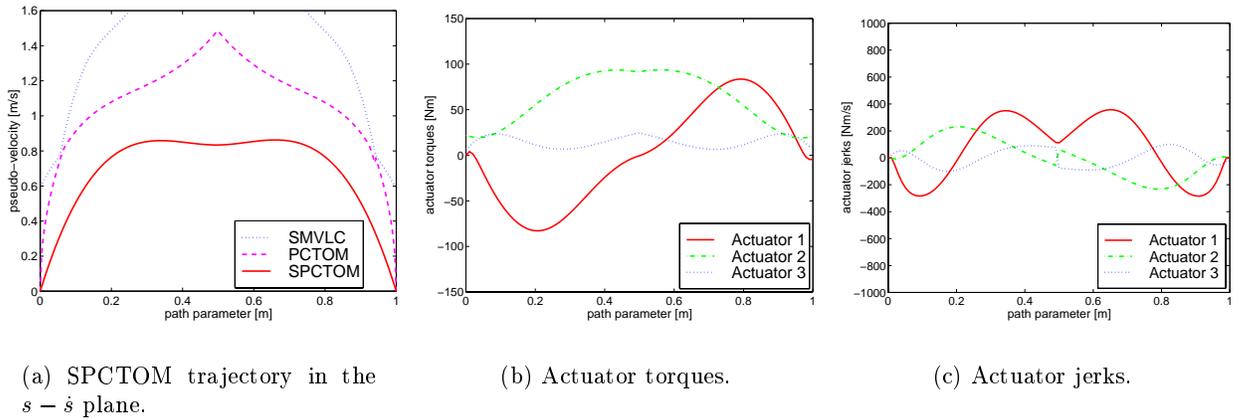


Figure 4.8: SPCTOM for the elbow manipulator along the circular path - Example 1 (medium jerk limits).

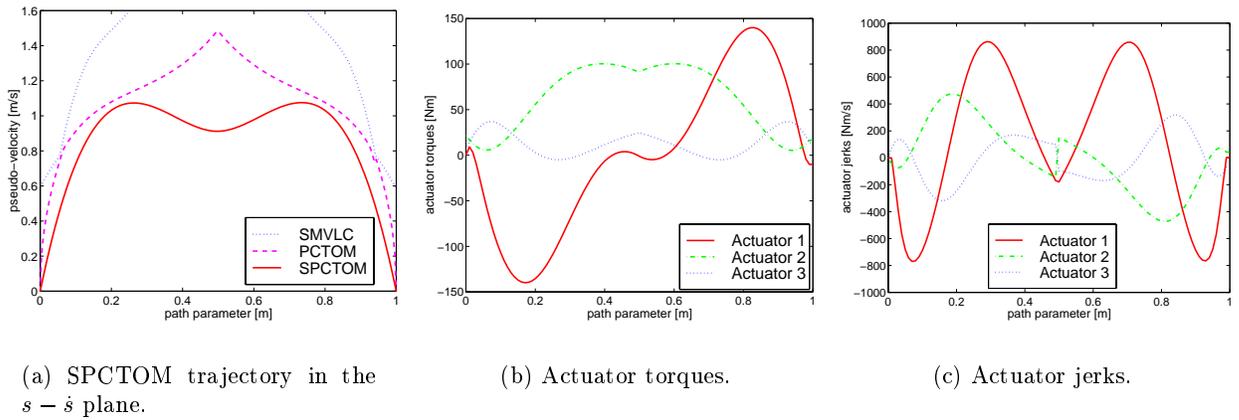


Figure 4.9: SPCTOM for the elbow manipulator along the circular path - Example 2 (high jerk limits).

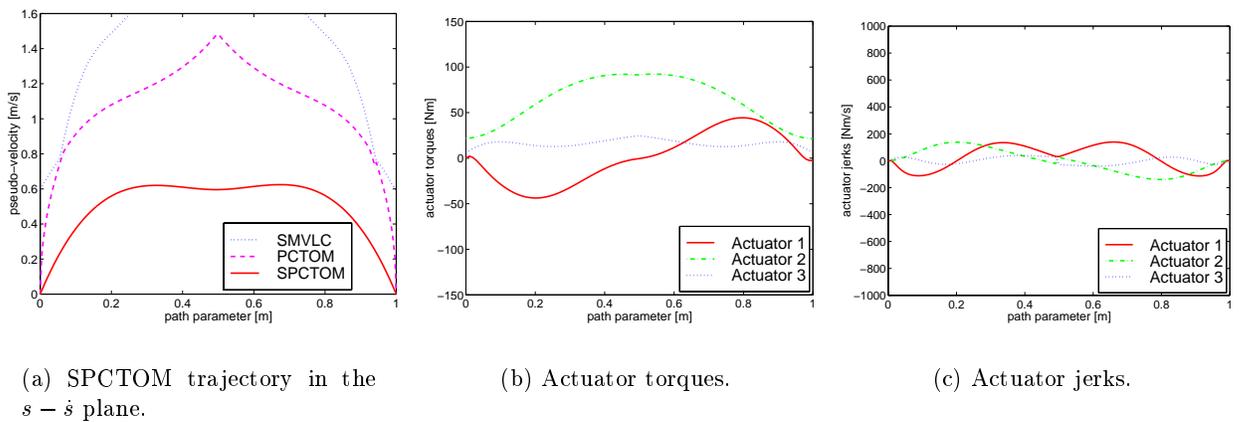


Figure 4.10: SPCTOM for the elbow manipulator along the circular path - Example 3 (low jerk limits).

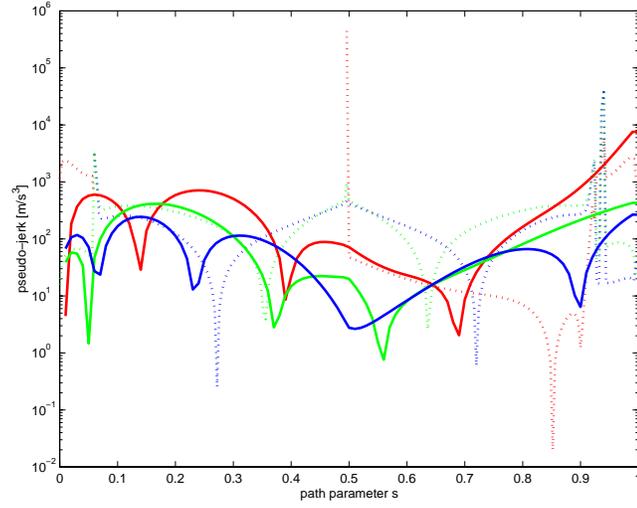


Figure 4.11: Peak actuator jerks for the SPCTOM (solid lines) and the PCTOM (dotted lines) for the elbow manipulator along the circular path.

Since the SPCTOM trajectory is represented by only two cubics splined together, one can consider whether the trajectory parameterization is adequate. To resolve this issue, two extra knot points are chosen, located at equal distances from the path end points and the switching point. Now, the optimal trajectory is approximated by four cubics and the vector of optimization variables is:

$$\mathbf{x} = \left( \frac{ds}{ds_0} \quad \frac{\dot{s}_1}{1.127} \quad \frac{\dot{s}_1}{1.48} \quad \frac{\dot{s}_1}{1.31} \quad \frac{ds}{ds_f} \right)^T. \quad (4.4)$$

The results obtained using this trajectory parameterization are given in Table 4.6. The SPCTOM trajectories for all three actuator jerk limits are plotted in Figures 4.12-4.14.

Table 4.6: Optimal motion times for the elbow manipulator along the circular path when additional knot points are inserted [sec].

Initial guess	SPCTOM			PCTOM
	Medium jerk limits	High jerk limits	Low jerk limits	
	Example 1	Example 2	Example 3	
1	$t_1 = 1.97$	$t_2 = 1.46$	$t_3 = 2.83$	$t = 1.07$
2	$t_1 = 2.00$	$t_2 = 1.33$	$t_3 = 2.92$	
3	$t_1 = 2.02$	$t_2 = 1.30$	$t_3 = 2.92$	
4	$t_1 = 2.15$	$t_2 = 1.33$	$t_3 = 2.73$	
Restart	$t_1 = 1.97$	$t_2 = 1.29$	$t_3 = 2.73$	

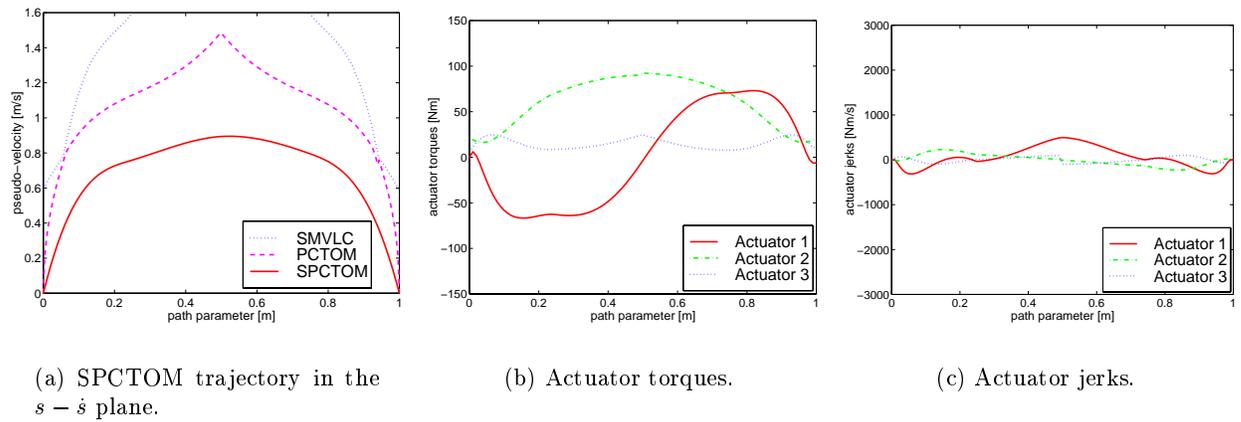


Figure 4.12: SPCTOM for the elbow manipulator along the circular path (additional knot points inserted) - Example 1 (medium jerk limits).

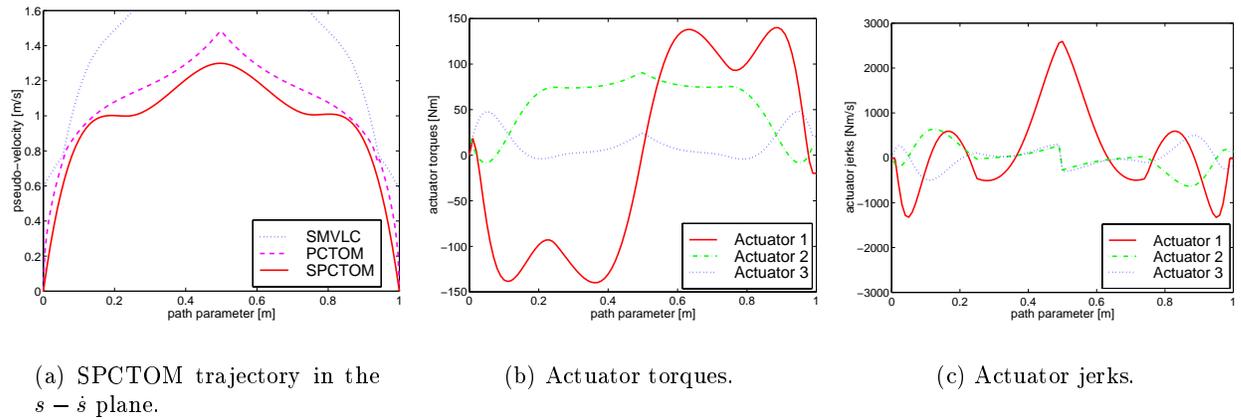


Figure 4.13: SPCTOM for the elbow manipulator along the circular path (additional knot points inserted) - Example 2 (high jerk limits).

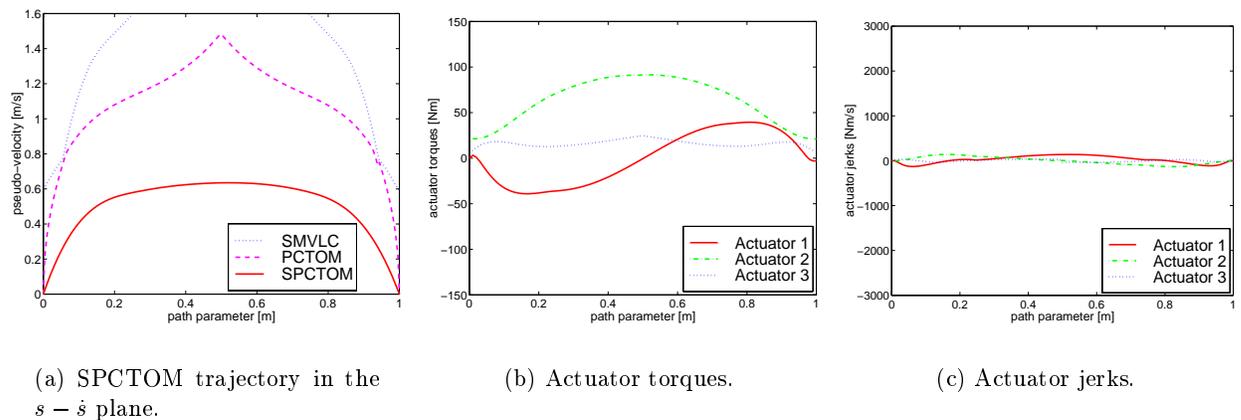


Figure 4.14: SPCTOM for the elbow manipulator along the circular path (additional knot points inserted) - Example 3 (low jerk limits).

The results in Table 4.6 show that the SPCTOM trajectory is better parameterized by more than two cubics. In view of this, it is recommended that at least three knot points are used when computing the SPCTOM. By doing so, the limitations involved in the trajectory parameterization are reduced while the increase in the computational cost is negligible.

## 4.2 Discussion

Motion times for both the PCTOM and the SPCTOM for all three examples presented in Sections 4.1.1 and 4.1.2 are summarized in Table 4.7.

Table 4.7: PCTOM and SPCTOM motion times for the examples in Sections 4.1.1 and 4.1.2 [sec].

Parabolic path				Circular path						
PCTOM	SPCTOM			PCTOM	SPCTOM					
	medium	high	low		One knot			Three knots		
					medium	high	low	medium	high	low
1	2	3	1	2	3	1	2	3		
1.72	2.74	1.91	3.92	1.07	2.13	1.57	2.90	1.97	1.29	2.72

From the results from Table 4.7 and other simulation examples run during the course of this research, the following conclusions can be drawn:

- The SPCTOM trajectory parameterization is adequate when knot point locations are the same as the PCTOM switching point locations if the PCTOM has more than one switching point. If the PCTOM has only one switching point, using two additional knot points (equally spaced between the path end points and the switching point) results in an improved trajectory approximation.
- Bounding the actuator jerks yields increased motion time for the SPCTOM compared to the PCTOM. The more restrictive the bounds are, the more significant the increase in motion time is.
- If the actuator jerk bounds are set sufficiently high, then the actuator torques become the active constraints (as in the case of second set of bounds for the robot along the parabolic

path, Figure 4.3(b)- 4.3(c)). The slight increase in motion time is then due to the smoothing of the actuator torques.

### 4.3 SPTOM Performance Investigation

The influence of the actuator jerk bounds on the trajectory traversal time has been analyzed in the previous section. Herein, their influence on the controller tracking performance is investigated through simulations. As shown in the previous section, from a planning level perspective, the smoothness of the trajectory has a negative impact on the motion time. That is, the higher the actuator jerks are, the lower the motion time is, with the PCTOM trajectory being the fastest possible motion given the limits on the actuator torques. These planning level results do not, however, give an indication of the system performance, i.e., the tracking accuracy. Thus, it is not possible to decide how to strike a balance between these conflicting requirements: namely, increased task productivity, as reflected in the transversal time, and good system performance, as reflected by the tracking accuracy. The following simulations address this issue.

Since the experiments reported in the following Chapter 5 are carried out on the SCORBOT ER VII robot in the Industrial Automation Laboratory (IAL) at UBC, the simulations performed herein use the same manipulator. Its model is built in Simulink using estimated dynamic parameters. The SCORBOT ER VII Denavit-Hartenberg parameters are given in Table 4.8, while estimated link masses and inertias are given in Table 4.9. Only the robot positional degrees of freedom are considered. Hence, for the purpose of the simulations performed here, the SCORBOT ER VII is considered a 3-dof elbow manipulator. The detailed dynamic model of the robot is presented in Appendix B.

Table 4.8: The SCORBOT ER VII Denavit-Hartenberg parameters.

Link	$\theta[rad]$	$d[m]$	$a[m]$	$\alpha[rad]$
1	$\theta_1 = 0$	$d_1 = 0.3585$	$a_1 = 0.050$	$\alpha_1 = -\frac{\pi}{2}$
2	$\theta_2 = 0$	$d_2 = -0.037$	$a_2 = 0.300$	$\alpha_2 = 0$
3	$\theta_3 = 0$	$d_3 = 0.0$	$a_3 = 0.250$	$\alpha_3 = 0$

No parameter identification has been performed because the high friction in the joints negates

Table 4.9: SCORBOT ER VII estimated masses and inertias.

Mass [kg]	$I_x[kgm^2]$	$I_y[kgm^2]$	$I_z[kgm^2]$
$m_1 = 0$	$I_{x1} = 0.00$	$I_{y1} = 0.05$	$I_{z1} = 0.00$
$m_2 = 6.6$	$I_{x2} = 0.10$	$I_{y2} = 0.60$	$I_{z2} = 0.60$
$m_3 = 4.2$	$I_{x3} = 0.02$	$I_{y3} = 0.20$	$I_{z3} = 0.30$

Table 4.10: Actuator torque and jerk bounds for the SCORBOT ER VII.

Torque limits	Medium jerk limits	High jerk limits	Low jerk limits
	Example 1	Example 2	Example 3
$T[Nm]$	$\dot{T}_1[Nm/sec]$	$\dot{T}_2[Nm/sec]$	$\dot{T}_3[Nm/sec]$
$T_1 = 10$	$\dot{T}_{11} = 100$	$\dot{T}_{12} = 1000$	$\dot{T}_{13} = 10$
$T_2 = 10$	$\dot{T}_{21} = 100$	$\dot{T}_{22} = 1000$	$\dot{T}_{23} = 10$
$T_3 = 10$	$\dot{T}_{31} = 100$	$\dot{T}_{32} = 1000$	$\dot{T}_{33} = 10$

the possibility of obtaining accurate results (as determined by previous attempts<sup>1</sup>). Friction itself has been modelled as Coulomb and viscous friction, with the Coulomb friction coefficients 2.0[Nm] and the viscous friction coefficients 0.2Nmsec for all three links. While an accurate robot model would yield better results, it is seldom the case in industrial practice that such a model is available. Therefore, a comparison between the tracking accuracy for the PCTOM and the SPCTOM even with a poor robot model is considered to have relevance.

The actuator torque limits were computed using data in the actuator manufacturer catalogue. As before, the three sets of actuator jerk limits were set arbitrarily, Table 4.10.

The path is a straight line given in parametric form as:

$$\begin{aligned}
 x(s) &= 0.4 \\
 y(s) &= 0.3s - 0.1 \\
 z(s) &= 0.2s + 0.3 \\
 s &= 0, \dots, 1.
 \end{aligned} \tag{4.5}$$

Since the PCTOM of the SCORBOT ER VII along this path has a single switching point (Figure 4.15), two additional knot points are used for parameterizing the SPCTOM trajectory. Along the path, the three knot points are located at  $s = 0.25$ ,  $s = 0.49$ , and  $s = 0.75$  respectively. The

<sup>1</sup>S. Gu, UBC IAL, Personal communication

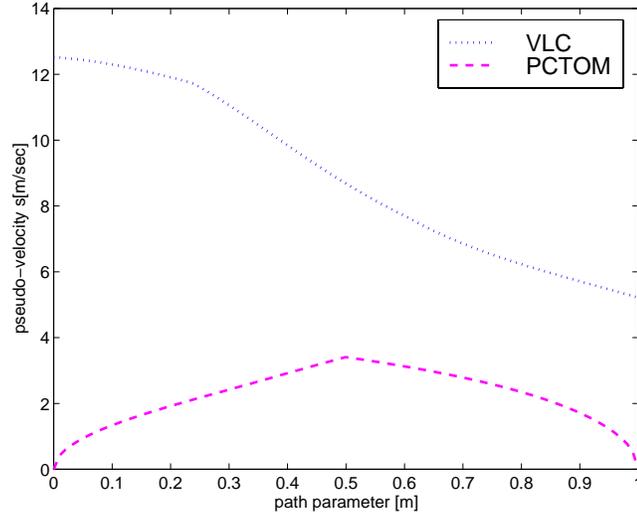


Figure 4.15: PCTOM trajectory in the  $s - \dot{s}$  phase plane for the SCORBOT ER VII along the linear path.

vector of optimization variables is:

$$\mathbf{x} = \left( \frac{d\dot{s}}{ds} 0 \quad \dot{s}_1 \quad \dot{s}_1 \quad \dot{s}_1 \quad \frac{d\dot{s}}{ds} f \right)^T. \quad (4.6)$$

A fifth trajectory used in simulations is a standard quintic trajectory. The motion time for the quintic trajectory is given by [2]:

$$t_q = \max\{t_v, t_a\}, \quad (4.7)$$

where:

$$t_v = \frac{15d}{8v_{max}} \quad (4.8)$$

$$t_a = \sqrt{5.77 \frac{d}{a_{max}}}, \quad (4.9)$$

with  $d$  being the distance traveled by the end-effector during the motion and  $v_{max}$  and  $a_{max}$  the (estimated) maximum end-effector velocity and acceleration. Once the motion time has been

determined, the quintic trajectory is found in parametric form by solving for the coefficients in:

$$s(t) = a_5t^5 + a_4t^4 + a_3t^3 + a_2t^2 + a_1t + a_0, \quad (4.10)$$

by applying the boundary conditions:

$$\begin{aligned} s(0) &= 0, \\ \dot{s}(0) &= 0, \\ \ddot{s}(0) &= 0, \\ s(t_q) &= 1, \\ \dot{s}(t_q) &= 0, \\ \ddot{s}(t_q) &= 0. \end{aligned} \quad (4.11)$$

The joint trajectories are then obtained by applying the inverse kinematics of the SCORBOT ER VII.

The motion times for the PCTOM trajectory, the SPCTOM trajectories corresponding to the different jerk limits, and the quintic trajectory are given in Table 4.11. In Figures 4.16-4.20, all these trajectories are plotted in the  $s - \dot{s}$  phase plane together with the corresponding actuator torques and jerks.

Table 4.11: PCTOM, SPCTOM, and quintic trajectory motion times for the SCORBOT ER VII along the straight line path [sec].

PCTOM	SPCTOM			Quintic
	1	2	3	
0.59	0.735	0.7	1.5	2.0

A PID independent joint controller with position feedback is used to compare the tracking performance for these trajectories. It has been tuned for each link individually [36] for critical damping and a rise time of 200msec. The gains of the corresponding discrete time controller have been then determined for a sampling frequency of 200Hz.

In the simulations, actuator torques saturate at 10Nm, which is the torque limit considered during planning.

The tracking performance of the PID controller for all five trajectories is plotted in Figure 4.21,

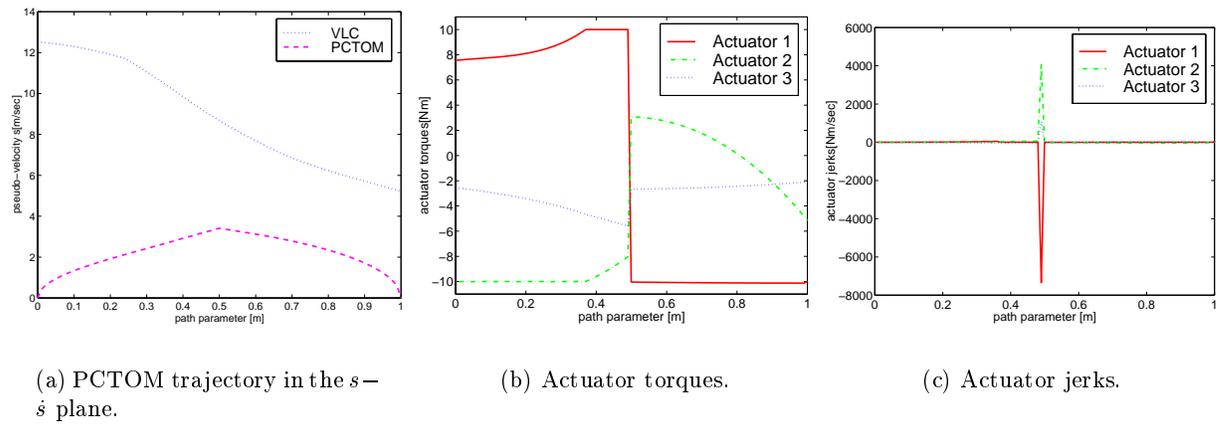


Figure 4.16: PCTOM for the SCORBOT ER VII along the linear path.

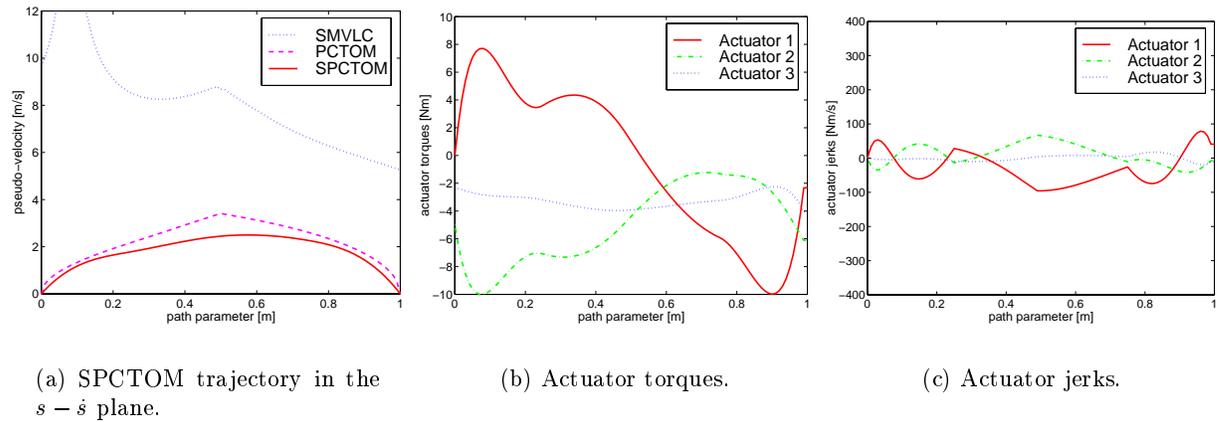


Figure 4.17: SPCTOM for the SCORBOT ER VII along the linear path (additional knot points inserted) - Example 1 (medium jerk limits).

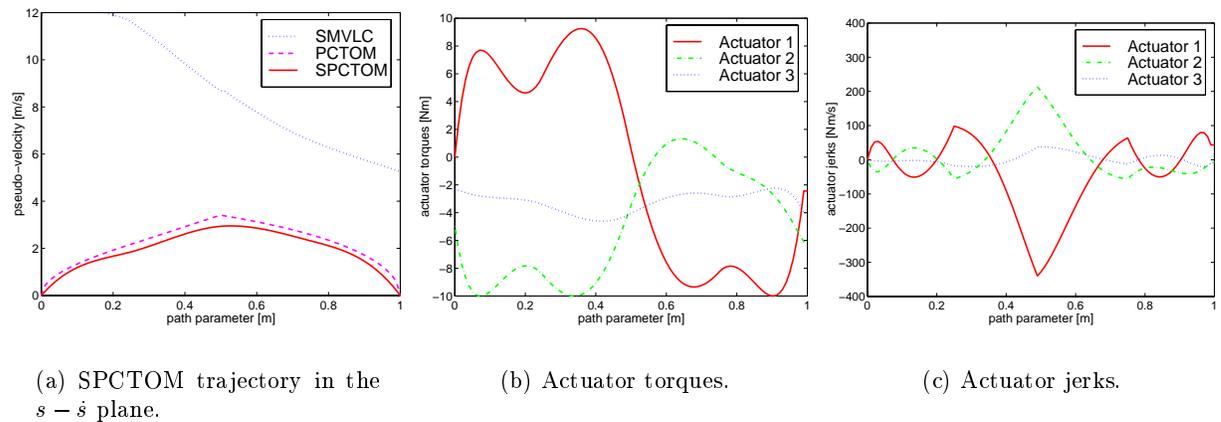


Figure 4.18: SPCTOM for the SCORBOT ER VII along the linear path (additional knot points inserted) - Example 2 (high jerk limits).

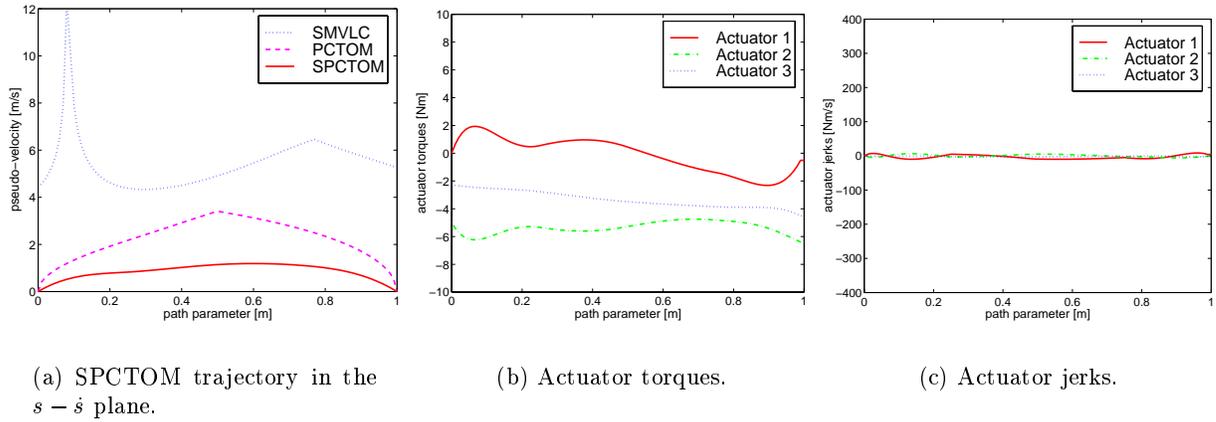


Figure 4.19: SPCTOM for the SCORBOT ER VII along the linear path (additional knot points inserted) - Example 3 (low jerk limits).

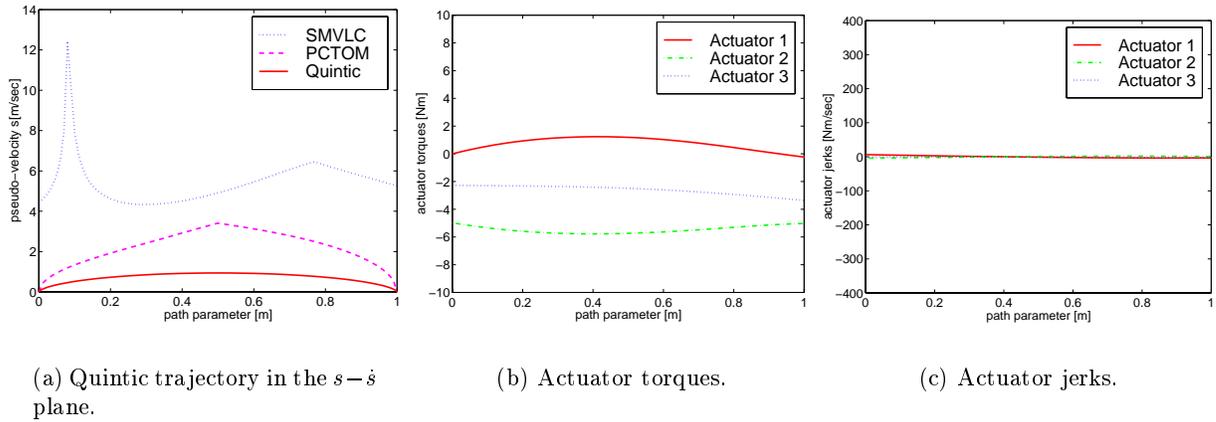


Figure 4.20: Quintic trajectory for the SCORBOT ER VII along the linear path.

while the planned and simulated actuator torques are plotted in Figures 4.22-4.26.

As seen in Figure 4.21, due to actuator torque saturation, the controller cannot keep the end-effector on the path when the actuator jerks are too high. This is the case with the PCTOM trajectory and the SPCTOM trajectories corresponding to actuator jerk limits of 100Nm/sec and 1000Nm/sec (labeled 'spctom1' and 'spctom2', respectively, in Figure 4.21). This result shows that actuator jerk limits are extremely important for the ability of the system to track a planned trajectory, especially in the face of poorly modelled system dynamics. As expected, the smoother the trajectory, i.e., the lower the actuator jerk limits, the higher the tracking accuracy of the controller.

Comparing the SPCTOM trajectory to the PCTOM and the quintic trajectories, the following

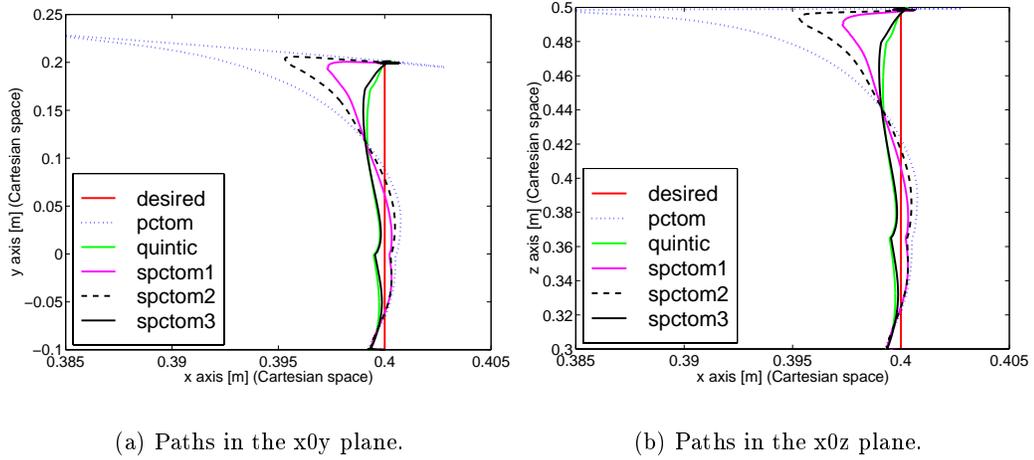


Figure 4.21: Controller tracking performance for the PCTOM, quintic, and SPCTOM trajectories.

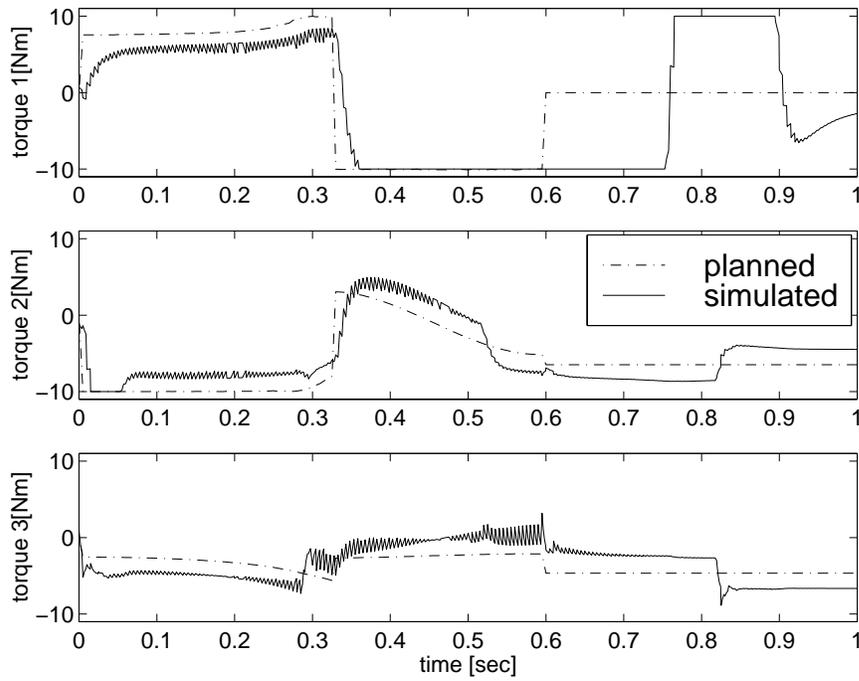


Figure 4.22: Desired and simulated torques for the PCTOM trajectory.

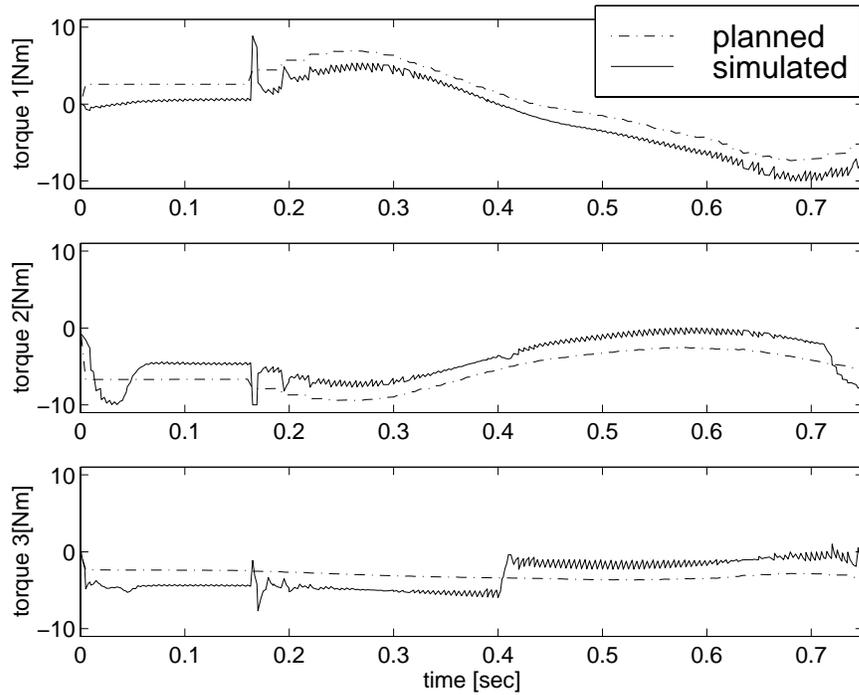


Figure 4.23: Desired and simulated torques for the SPCTOM trajectory - Example 1 (medium jerk limits).

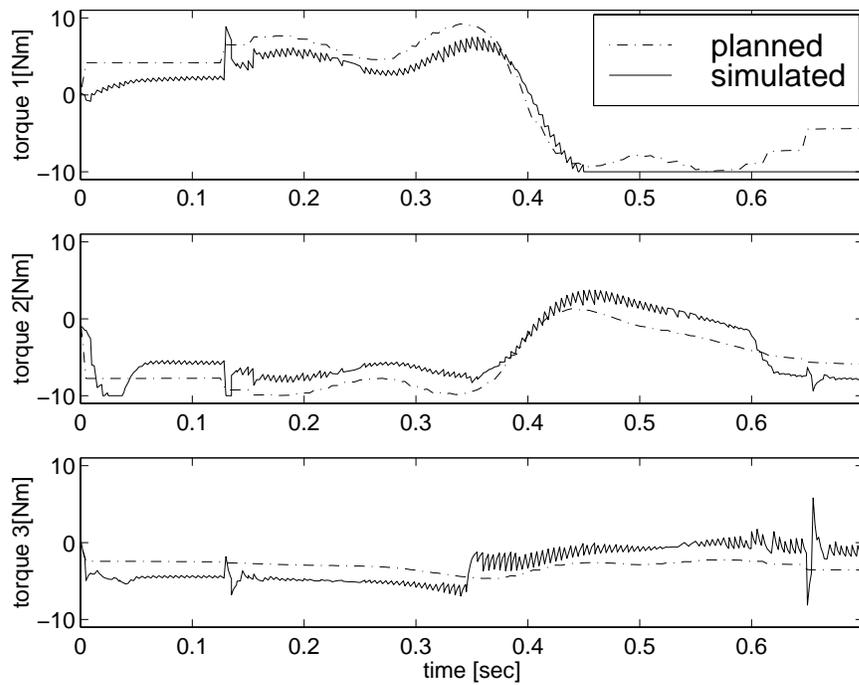


Figure 4.24: Desired and simulated torques for the SPCTOM trajectory - Example 2 (high jerk limits).

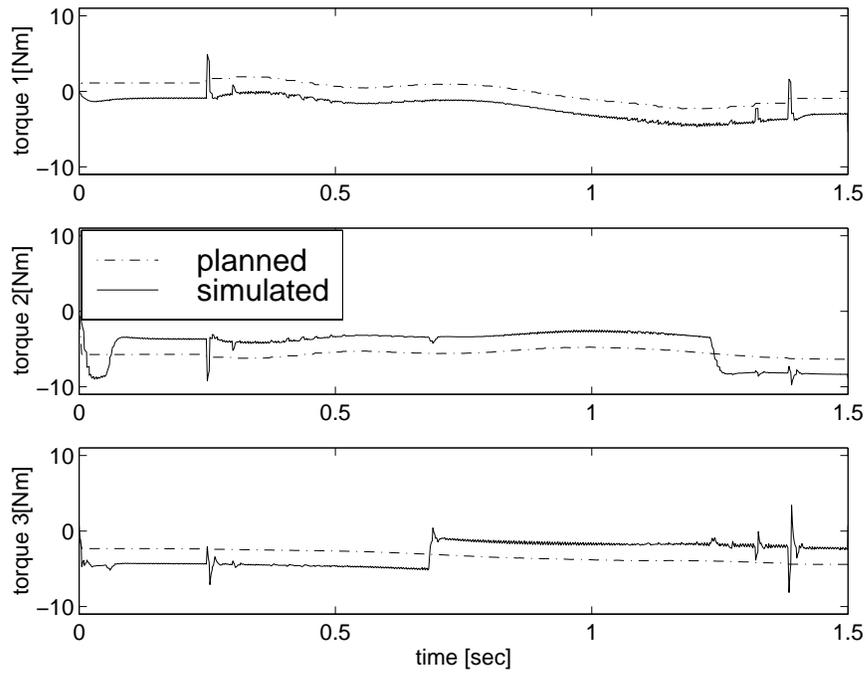


Figure 4.25: Desired and simulated torques for the SPCTOM trajectory - Example 3 (low jerk limits).

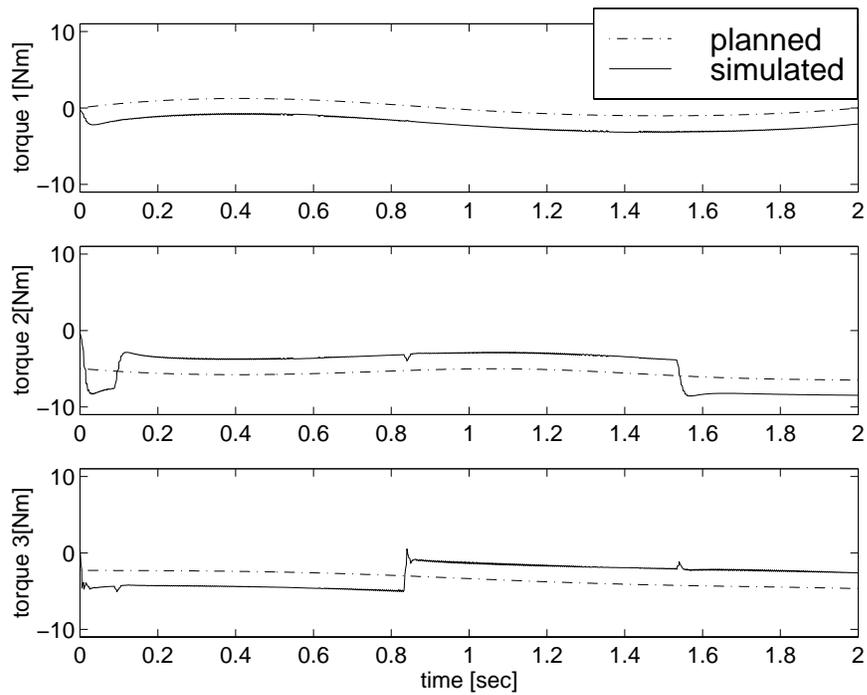


Figure 4.26: Desired and simulated torques for the quintic trajectory.

conclusions can be drawn:

- While the PCTOM trajectory represents the theoretical optimal motion when actuator torque limits are imposed, it is not feasible for direct implementation on a poorly modelled system with an unsophisticated controller, since it cannot be tracked by such a controller. On the other hand, the SPCTOM trajectory, while maintaining a measure of optimality, can also be tracked by the controller. Thus, it represents a feasible time-optimal motion.
- Evaluated against the quintic trajectory, the SPCTOM trajectory yields an improved motion time for comparable controller tracking performance. This shows that fixed actuator jerk limits are a much better way of defining optimal smooth trajectories over quintics with fixed velocity and acceleration limits. The reason for this is that jerk limits are actuator related and position independent, whereas the velocity and acceleration capabilities of the robot are configuration dependent, thus the limits imposed on them are worst case limits.
- Actuator jerks also represent a measure of the actuator torque smoothness. The lower the jerks are, the smoother the torques become. Thus, based on the relationship between torque smoothness and robot and actuator strain, [8], lower jerks also result in decreased system strain and wear.

## 4.4 Summary

In this chapter, two objectives were pursued. First, the methodology proposed in Chapter 3 for planning an SPCTOM trajectory was demonstrated. Second, the performance of the SPCTOM trajectory compared to the PCTOM and to a quintic trajectory was investigated.

The examples presented in Section 4.1 show that the FTM is adequate for planning SPCTOM trajectories. This conclusion is drawn from the fact that there is only a slight increase in the motion time of the SPCTOM with high jerk limits compared to the motion time of the PCTOM. The increase is attributed to the chosen trajectory parameterization in the phase plane, as well as to the decrease in actuator jerks. Thus, it is expected that the FTM has converged to near the global optimum for the SPCTOM and that the methodology proposed in Chapter 3 is suitable for planning SPCTOM trajectories.

The limitations involved in selecting a number of knot points equal to the number of switching points of the related PCTOM trajectory were investigated in Section 4.1 through an example. The example demonstrates that the parameterization of the SPCTOM trajectory in the phase plane by only two cubics is inadequate. Therefore, it is proposed that two additional knot points be inserted, since the added computational burden is negligible.

In section 4.3, the performance of the SPCTOM trajectory is compared to that of the PCTOM and quintic trajectories through simulations. These simulations show that, while limited actuator jerks have a negative influence on motion time, they need to be imposed in order to ensure the required tracking accuracy when the available system model is poor. While sophisticated controllers might track the PCTOM trajectory of an accurately identified system, it is shown that in this case the PID independent joint controller often used in industrial practice cannot keep the end-effector of an approximately modelled robot along the path. Practically, it is difficult to obtain an accurate manipulator model and most systems are controlled by PID controllers. Therefore, imposing actuator jerk limits and planning SPCTOM trajectories accordingly results in the planning of feasible time optimal motions.

Actuator jerk limits not only make the SPCTOM trajectories readily implementable on a manipulator, they are also a better way of defining the optimal trajectory smoothness over quintics with global velocity and acceleration limits, because they are actuator dependent and configuration independent.

In conclusion, the SPCTOM trajectory represents a balance between the conflicting requirements of minimum motion time and specified tracking accuracy: it results in lower motion time compared to a quintic trajectory with similar tracking performance and, though slower than the PCTOM trajectory, it can be tracked by a typical industrial controller.

## Chapter 5

# Experiments

In chapter 4, the SPCTOM trajectory was compared with the PCTOM and quintic trajectories through simulations. In this chapter, the same comparison is done through experiments performed on the SCORBOT ER VII manipulator in the IAL at UBC.

As previously stated, the view taken in this work and justified by simulations is that a desired degree of trajectory smoothness, which directly reflects upon system tracking performance, can be imposed through limits on the actuator jerks. This smoothness can compensate for modelling inaccuracies and thus feasible time optimal trajectories can be determined. These are trajectories that are readily implementable in an industrial setting, i.e., on a manipulator for which an accurate model is difficult to obtain and which is controlled by a PID independent joint controller. The goal of the present chapter is to provide the experimental proof of this assertion, while investigating appropriate jerk limits for practical applications.

### 5.1 Experimental Setup

The experimental setup used to test the different trajectories is a SCORBOT ER VII 5-dof robot in the IAL at UBC (Figure 5.1). The robot is controlled by a TMS320C32 DSP board, interfaced with two axis control cards, each capable of handling three axes simultaneously. An open architecture real-time operating system (ORTS) [21] is used in the implementation of the control algorithm and in reading the pre-planned trajectories and feeding them to the control loop

at the controller frequency. The axis control cards and the real-time operating system ORTS were developed in-house by the Manufacturing Automation Laboratory, UBC. For the purpose of the experiments reported here, only the positional degrees of freedom of the robot are considered, thus the robot is treated as a 3-dof elbow manipulator with the DH parameters given in Table 4.8. A tuned discrete PID algorithm is used to provide the control law. This setup simulates typical conditions in industry, where the robot is equipped with a closed architecture discrete PID independent joint controller.



Figure 5.1: Experimental setup.

## 5.2 Experimental Results

The PCTOM, SPCTOM, and quintic trajectories were all implemented on the SCORBOT ER VII and the encoder readings, as well as the actuator control voltages and output currents were recorded. The results of the experiments are plotted in Figures 5.2-5.6, and summarized in Table 5.1.

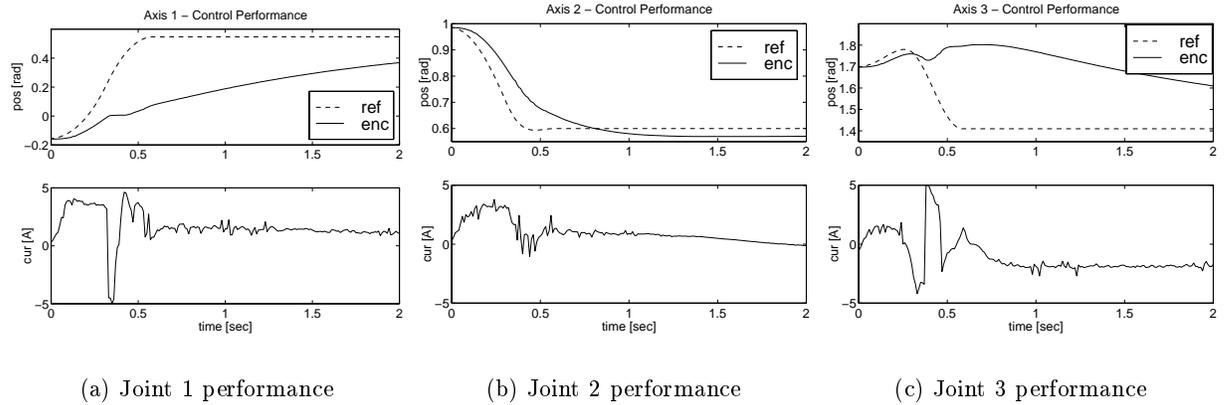


Figure 5.2: Experimental results for the PCTOM trajectory implemented on the SCORBOT ER VII.

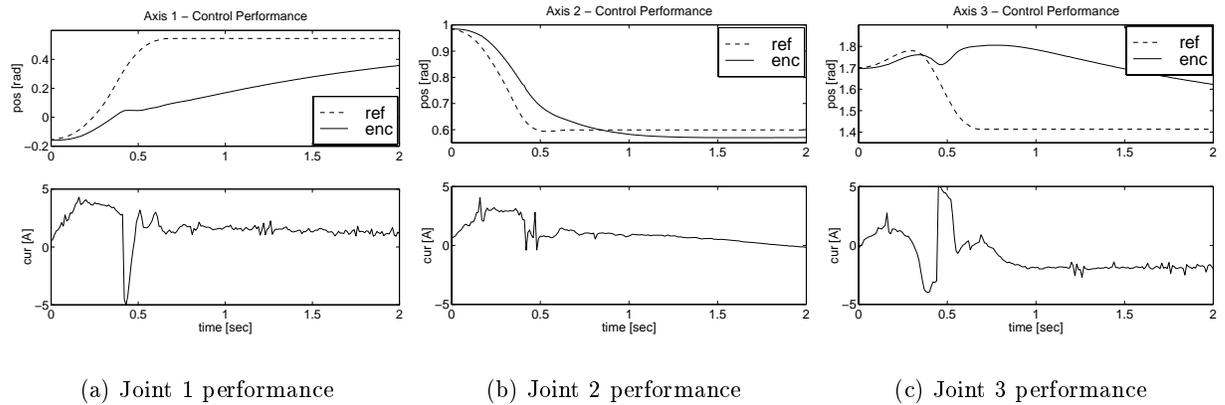


Figure 5.3: Experimental results for the SPCTOM trajectory (Example 1 - jerk limits of 100Nm/sec) implemented on the SCORBOT ER VII.

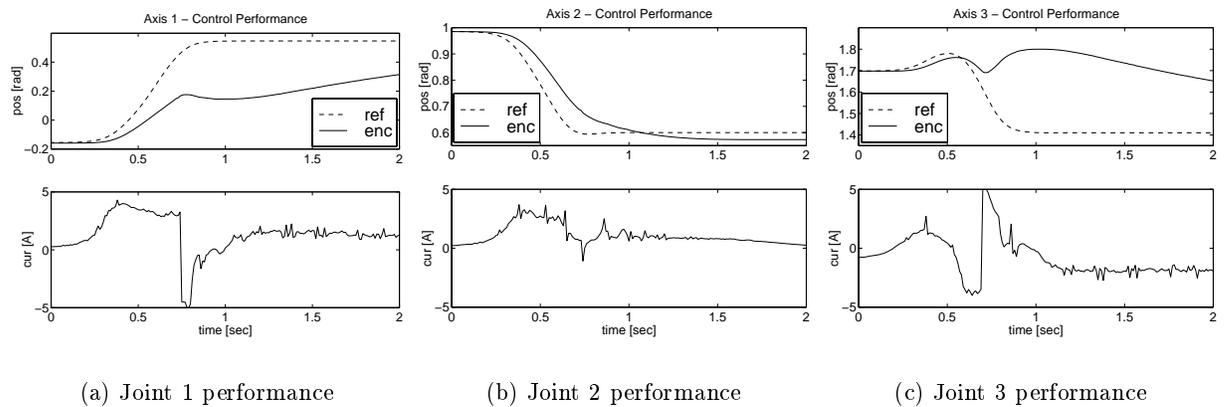


Figure 5.4: Experimental results for the SPCTOM trajectory (Example 2 - jerk limits of 1000Nm/sec) implemented on the SCORBOT ER VII.

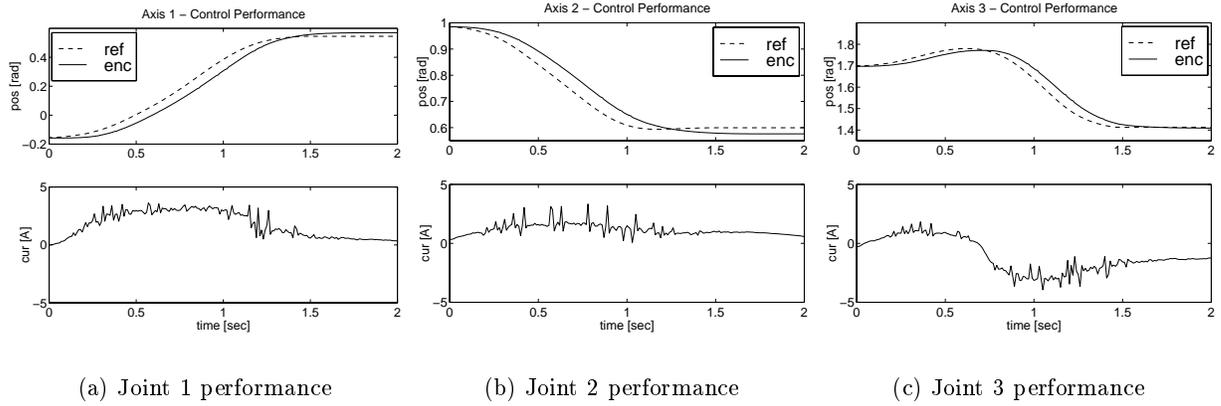


Figure 5.5: Experimental results for the SPCTOM trajectory (Example 3 - jerk limits of 10Nm/sec) implemented on the SCORBOT ER VII.

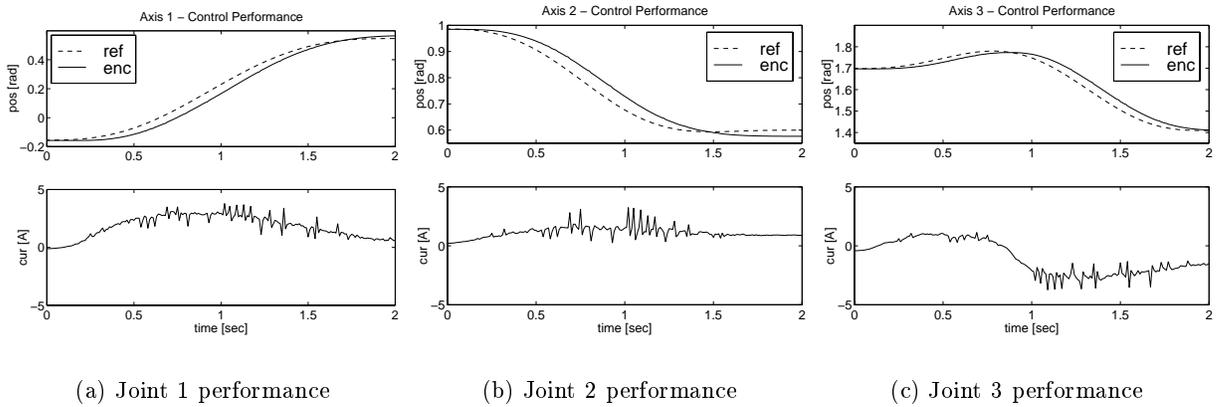


Figure 5.6: Experimental results for the quintic trajectory implemented on the SCORBOT ER VII.

Table 5.1: Experimental results for the PCTOM, SPCTOM, and quintic trajectories.

Trajectory	Jerk limits [Nm/sec]	Motion time [sec]	Maximum tracking error [cm]	RMS tracking error 1 [ $^{\circ}$ ]	RMS tracking error 2 [ $^{\circ}$ ]	RMS tracking error 3 [ $^{\circ}$ ]
PCTOM	$\infty$	4.0	14.0	17.5	3.1	15.8
SPCTOM 1	1000	4.0	13.4	17.2	2.8	14.8
SPCTOM 2	100	4.0	12.5	15.9	2.6	14.8
SPCTOM 3	10	1.5	3.1	2.6	1.9	1.5
Quintic	7	2.0	2.5	2.3	1.8	1.4

These experimental results support the results from Chapter 4. Namely, for high actuator jerk limits, the controller cannot keep the end-effector on the path. Figures 5.2, 5.3, and 5.4 show that trajectories with high jerks result in increased tracking errors, which, in turn, activate the controller, saturating the actuators. Whenever this happens, the end-effector leaves the path. Such a trajectory is an infeasible trajectory. For the case of the SCORBOT ER VII manipulator, actuator jerk limits more than an order of magnitude higher than actuator torque limits are not sufficient to ensure that the end-effector follows the planned path. While this result was not predicted by the simulations, it does not, however, present itself as totally unexpected. Due to the large errors involved in modelling the system, one would expect that the simulation results would overestimate the system capabilities.

The experimental performance of the SPCTOM trajectory corresponding to the low jerk limits, i.e. 10Nm/sec, is similar to its simulated performance. Thus, while being tracked by the controller with the same accuracy and the same effort as the quintic trajectory, it results in reduced motion time. This indicates that actuator jerk limits are preferable when determining smooth time optimal motions over global velocity and acceleration limits.

The SPCTOM trajectories above suggest that, for the SCORBOT ER VII, realistic jerk limits are approximately equal to the actuator torque limits. To verify this proposition, SPCTOM trajectories were determined for jerk limits of 50Nm/sec and 20Nm/sec and the robot was required to follow them. As seen in Figures 5.7 and 5.8, the jerk limits of 50Nm/sec are still too high for the system tracking capabilities. The SPCTOM trajectory corresponding to the 20Nm/sec limits results in lower motion time and slightly decreased tracking performance.

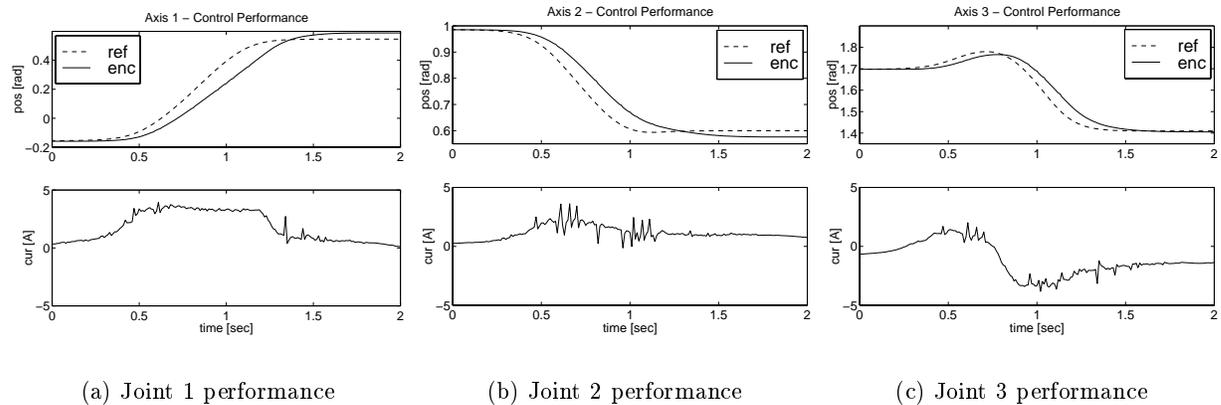
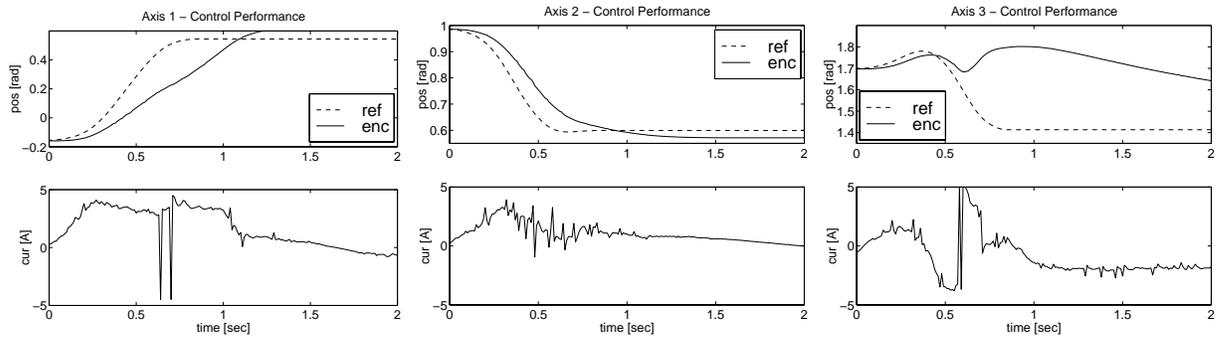


Figure 5.7: Experimental results for the SPCTOM trajectory (jerk limits of 20Nm/sec) implemented on the SCORBOT ER VII.



(a) Joint 1 performance

(b) Joint 2 performance

(c) Joint 3 performance

Figure 5.8: Experimental results for the SPCTOM trajectory (jerk limits of  $50\text{Nm}/\text{sec}$ ) implemented on the SCORBOT ER VII.

### 5.3 Summary

In this chapter, experimental results obtained when implementing the PCTOM, SPCTOM corresponding to different jerk limits, and quintic trajectories were presented. These results confirmed that the actuator jerk limits are instrumental in determining feasible time optimal trajectories when an accurate dynamic model of the system is not available. They also demonstrated that trajectory smoothness is directly related to the tracking performance of the controller.

No specific rules were obtained for how to choose appropriate jerk limits. However, the experiments have shown that these limits should be fairly conservative when applied to inaccurately modelled systems. Thus, actuator jerk limits can compensate for large model errors. The trade-off is an increase in motion time.

## Chapter 6

# Conclusions and Recommendations

### 6.1 Summary

In this thesis, a strategy for planning SPCTOM trajectories has been presented. These time optimal trajectories are determined by imposing actuator jerk limits on the proposed motion. Such trajectories can be directly implemented on an industrial robot when the dynamic parameters are not accurately known, without the need for a sophisticated controller. They represent, therefore, feasible time optimal motions.

The SPCTOM trajectories provide a practical compromise between the need for shorter manipulation task completion times and the need for increased tracking accuracy and decreased vibrations of the robot system.

Previous planning strategies have incorporated the smoothness requirement as smoothness of the joint space trajectory and have imposed limits on the joint velocities, accelerations, and jerks or on the end-effector velocity and acceleration. In this thesis, the trajectory smoothness is defined in the phase space and limits on the actuator jerks are used in the planning approach.

The solution proposed for determining SPCTOM trajectories consists in transforming the TOC problem into a nonlinear parameter optimization problem and solving it using the FTM in [27]. The adequacy of this optimization technique is shown by setting the jerk limits very high and showing that the SPCTOM trajectory converges towards the PCTOM trajectory, within the limits

of the chosen trajectory parameterization.

While the negative effect of lowering the actuator jerk limits on the motion time is visible from the planning stage, the influence of the jerk limits on the trajectory feasibility is studied through simulations and experiments. Both demonstrate that a trajectory which requires high actuator jerks is not a feasible trajectory for a poorly modelled system controlled by a simple PID independent joint controller. SPCTOM trajectories, however, represent feasible time optimal motions and therefore they have industrial relevance. Moreover, the limits on the actuator jerks are related to the tracking performance of the controller and the control effort of the actuators.

The simulations and the experiments are also conducted with the goal of comparing the SPCTOM trajectory to the smooth quintic trajectories. The results are favourable to the SPCTOM trajectory: for similar actuator jerks and controller effort for both trajectories, the SPCTOM trajectory is characterized by a significantly shorter motion time. This result shows that SPCTOM trajectories with fixed actuator jerk limits provide a superior method of defining optimal smooth motions over quintics with fixed velocity and acceleration limits. This is due to the fact that velocity and acceleration limits are position dependent and, therefore, are determined for the worst case scenario. The actuator jerk limits, on the other hand, are position independent. Thus, they represent a uniform measure for the trajectory smoothness over the entire robot workspace.

The disadvantage of planning a SPCTOM trajectory over planning a quintic trajectory is the required computational effort. While an optimal quintic involves practically no computational load, a SPCTOM is planned through a nonlinear optimization and requires considerable computational effort. Therefore, at this stage, the SPCTOM trajectory can only be used when the planning is done off-line.

## 6.2 Recommendations

In this thesis, SPCTOM trajectories with limited actuator jerks were proven to represent feasible trajectories even for a system with roughly estimated dynamic parameters controlled by an unsophisticated PID controller, being therefore readily implementable in any robotic system. Moreover, they compared favourably to the optimal quintic trajectories. However, several research directions which could improve the present work immediately suggest themselves. They are presented herein.

(1) At the theoretical level, one could investigate the optimality conditions for the SPCTOM. This could potentially lead to the derivation of an algorithm for computing SPCTOM similar to that presented in [5] for computing the PCTOM. The advantage of such an efficient search would be its much lower computational requirements, which would make SPCTOM trajectories suitable for on-line planning. This would considerably increase the practical significance of such trajectories, since they could be used in planning any path-constrained manipulation task.

(2) At the experimental level, one could investigate the selection of the actuator jerk limits. The research reported here indicates that they should be in the same range as the actuator torque limits. However, more work is required to determine whether they are only actuator dependent or they are also influenced by factors like model inaccuracies or friction. All these are reasonable assumptions and they need to be verified by experimental research.

(3) Finally, one could investigate how jerk limits can be directly related to robot vibrations and wear. This correlation might be difficult to establish, since it entails developing a reasonably easy to use measure for vibrations and their influence on robot wear. Furthermore, it could require longitudinal studies with access to working industrial robots. Nevertheless, it would provide an objective basis for determining what the optimal task productivity should be when both short term and long term consequences would be considered.

The research directions outlined here would be challenging and they would contribute to defining task time optimality in a more realistic way, with a more careful consideration for the physical limitations of a robotic system. To various degree, they would also contribute to the OCT, as well as to the problem of on-line time-optimal trajectory planning.

# Bibliography

- [1] S.K. Agrawal. Inertia Matrix Singularity of Spatial Series-Chain Manipulators with Point Masses. *ASME Transactions, Journal of Dynamic Systems, Measurement, and Control*, 31:131–136, 1991.
- [2] R.L. Andersson. *A Robot Ping-Pong Player: Experiment in Real-Time Intelligent Control*. MIT Press, Cambridge, MA, 1988.
- [3] Robotic Industries Association. ANSI/RIA R15.06-1992 Standard, 1992.
- [4] J.E. Bobrow. Optimal Robot Path Planning using Minimum-Time Criterion. *IEEE Transactions on Robotics and Automation*, 4:443–450, 1988.
- [5] J.E. Bobrow, S. Dubowsky, and J.S. Gibson. Time-Optimal Control of Robotic Manipulators Along Specified Paths. *The International Journal of Robotics Research*, 4(3):3–17, 1985.
- [6] H.G. Bock and K.-J. Plitt. A Multiple Shooting Algorithm for Direct Solution of Optimal Control Problems. In *9th IFAC World Congress*, pages 1853–1858, Budapest, Hungary, 1984.
- [7] J. Butler and M. Tomizuka. A Suboptimal Reference Generation Technique for Robotic Manipulators Following Specified Paths. *Robotics Research*, 14:37–42, 1989.
- [8] B. Cao, G.I. Dodds, and G.W. Irwin. Time-Optimal and Smooth Constrained Path Planning for Robot Manipulators. In *IEEE International Conference on Robotics and Automation*, pages 1853–1858, San Diego, California, 1994.
- [9] Y.-H. Chang, T.-T. Lee, and C.-H. Liu. On-Line Cartesian Path Trajectory Planning for Robot Manipulators. In *IEEE International Conference on Robotics and Automation*, pages 62–67, 1988.
- [10] Y.-H. Chang, T.-T. Lee, and C.-H. Liu. On-Line Approximate Cartesian Path Trajectory Planning for Robotic Manipulators. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(3):542–547, May/June 1992.
- [11] Y. Chen and A.A. Desrochers. Structure of Minimum-Time Control Law for Robotic Manipulators with Constrained Paths. In *IEEE International Conference on Robotics and Automation*, pages 971–976, Piscataway, New Jersey, 1989.
- [12] Y. Chen and A.A. Desrochers. A Proof of the Structure of the Minimum-Time Control Law of Robotic Manipulators using a Hamiltonian Formulation. *IEEE Transactions on Robotics and Automation*, 6(3):388–393, June 1990.

- 
- [13] Y. Chen and J. Huang. An Improved Computation of Time-Optimal Control Trajectory for Robotic Point-To-Point Motion. *International Journal of Control*, 65(5):177–194, 1996.
- [14] Y. Chen, J. Huang, and T.Y. Wen. A Continuation Method for Time-Optimal Control Synthesis for Robotic Point-To-Point Motion. In *32nd Conference on Decision and Control*, pages 1628–1632, San Antonio, Texas, December 1993.
- [15] W.L. Cleghorn, J.F. Fu, and R.G. Fenton. A General Method for Optimum Design of Gear Boxes through Nonlinear Programming. In *ASME Design and Automation Conference*, volume DE-2, pages 153–160, Montreal, Quebec, 1989.
- [16] E.A. Croft, B. Benhabib, and R.G. Fenton. Near-Time Optimal Robot Motion Planning for On-line Applications. *Journal of Robotic Systems*, 12(8):553–567, 1995.
- [17] O. Dahl and L. Nielsen. Torque-Limited Path Following by On-Line Trajectory Time Scaling. *IEEE Transactions on Robotics and Automation*, 6(5):554–561, 1990.
- [18] Carl de Boor. *A Practical Guide to Splines*. Springer-Verlag, New York, 1978.
- [19] B.R. Donald and P.G. Xavier. Time-Safety Tradeoffs and a Bang-Bang Algorithm for Kinodynamic Planning. In *IEEE International Conference on Intelligent Robots and Systems*, pages 552–557, Sacramento, California, April 1991.
- [20] S. Dubowsky, M.A. Norris, and Z. Shiller. Time Optimal Trajectory Planning for Robotic Manipulators. In *IEEE International Conference on Robotics and Automation*, pages 1906–1912, San Francisco, California, April 1986.
- [21] N.A. Erol and Y. Altintas. Open Architecture Modular Tool Kit for Motion and Process Control. In *ASME International Mechanical Engineering Congress and Exposition, ASME Publication MED*, pages 15–22, Dallas, Texas, 1997.
- [22] R.G. Fenton and J. Xu. Kinetic Synthesis of Mechanisms Using the Flexible Tolerance Algorithm. In *Proceedings of the 7th World Congress of the Theory of Machines and Mechanisms*, volume 1, pages 37–40, Sevilla, Spain, 1987.
- [23] J.Y. Fourquet. Optimal Control Theory and Complexity of the Time-Optimal Problem for Rigid Manipulators. In *IEEE International Conference on Intelligent Robots and Systems*, pages 84–90, Yokohama, Japan, July 1993.
- [24] H.P. Geering, L. Guzzella, S.A.R. Hepner, and C.H. Onder. Time-Optimal Motions of Robots in Assembly Tasks. *IEEE Transactions on Automatic Control*, AC-31(6):512–518, 1986.
- [25] E.G. Gilbert and D.W. Johnson. Distance Functions and their Applications to Robot Path Planning in the Presence of Obstacles. *IEEE Journal of Robotics and Automation*, RA-1:21–30, 1985.
- [26] G. Heinzinger, P. Jacobs, J. Canny, and B. Paden. Time-Optimal Trajectories for a Robot Manipulator: A Provably Good Approximation Algorithm. In *IEEE International Conference on Robotics and Automation*, pages 150–156, Cincinnati, Ohio, May 1990.
- [27] D.M. Himmelblau. *Applied Nonlinear Programming*. McGraw-Hill, 1989.

- [28] M Hoffman, A. Malowany, and J. Angeles. Near-Minimum-Time Trajectories for Pick-and-Place Operations. In *ASME International Conference on Computers in Engineering*, pages 433–438, San Francisco, California, August 1988.
- [29] J.M. Hollerbach. Dynamic Scaling of Manipulator Trajectories. *ASME Transactions, Journal of Dynamic Systems, Measurement, and Control*, 106:102–106, 1984.
- [30] D.W. Johnson and E.G. Gilbert. Minimum-Time Robot Path Planning in the Presence of Obstacles. In *24th Conference on Decision and Control*, pages 1748–1753, Ft. Lauderdale, Florida, December 1985.
- [31] M.E. Kahn and B. Roth. The Near-Minimum Time Control of Open-Loop Articulated Kinematic Chains. *ASME Transactions, Journal of Dynamic Systems, Measurement, and Control*, pages 164–172, 1971.
- [32] J.C. Kieffer, A.J. Cahill, and M.R. James. Robust and Accurate Time-Optimal Path-Tracking Control for Robot Manipulators. *IEEE Transactions on Robotics and Automation*, 13:880–889, December 1997.
- [33] K. Kyriakopoulos and G. Saridis. Minimum Jerk Path Generation. In *IEEE International Conference on Robotics and Automation*, pages 364–369, Philadelphia, Pennsylvania, April 1988.
- [34] K. Kyriakopoulos and G. Saridis. Minimum Jerk for Trajectory Planning and Control. *Robotica*, 12:109–113, 1994.
- [35] G. Leitman. *The Calculus of Variations and Optimal Control*. Plenum Press-New York and London, 1981.
- [36] F.L. Lewis, C.T. Abdallah, and D.M. Dawson. *Control of Robot Manipulators*. Macmillan Publishing Company, New York, 1993.
- [37] J. Li, R.W. Longman, V.H. Schultz, and H.G. Bock. The Choice of Appropriate Cost Functionals for Optimizing the Operating Speed of Robots. In *AAS/AIAA Astrodynamics*, pages 1821–1840, Sun Valley, Idaho, 1997.
- [38] C.S. Lin, P.R. Chang, and J.Y.S. Luh. Formulation and Optimization of Cubic Polynomial Joint Trajectories for Industrial Robots. *IEEE Transactions on Automatic Control*, AC-28(12):1066–1074, 1983.
- [39] J.Y.S. Luh and C.S. Lin. Approximate Joint Trajectories for Control of Industrial Robots along Cartesian Paths. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-14(3):444–450, May/June 1984.
- [40] B.J. Martin and J.E. Bobrow. Determination of Minimum -Effort Motions for General Open Chains. In *IEEE International Conference on Robotics and Automation*, pages 1160–1165, Piscataway, New Jersey, 1995.
- [41] J.M. McCarthy and J.E. Bobrow. The Number of Saturated Actuators and Constraint Forces-During Time-Optimal Movement of a General Robotic System. In *IEEE International Conference on Robotics and Automation*, pages 542–546, Nice, France, 1992.

- [42] J.A. Nelder and R. Mead. A Simplex Method for Function Minimization. *Computer Journal*, 4:308–313, 1964.
- [43] W.S. Newman. Robust Near-Time Optimal Control. *IEEE Transactions on Automatic Control*, 35(7):841–844, 1990.
- [44] D. Paviani and D.M. Himmelblau. Constrained Nonlinear Optimization by Heuristic Programming. *Operations Research*, 17:872–882, 1969.
- [45] F. Pfeiffer and R. Johanni. A Concept for Manipulator Trajectory Planning. *IEEE International Journal of Robotics and Automation*, RA-3(2):115–123, 1987.
- [46] V.T. Rajan. Minimum-Time Trajectory Planning. In *IEEE International Conference on Robotics and Automation*, pages 759–764, St. Louis, Montana, 1985.
- [47] G. Sahar and J.M. Hollerbach. Planning of Minimum-Time Trajectories for Robot Arms. In *IEEE International Conference on Robotics and Automation*, pages 751–758, St. Louis, Montana, 1985.
- [48] Z. Shiller. Interactive Time-Optimal Robot Motion Planning and Work-Cell Layout Design. In *IEEE International Conference on Robotics and Automation*, pages 964–969, Phoenix, Arizona, May 1989.
- [49] Z. Shiller. On Singular Points and Arcs in Path Constrained Time Optimal Motions. *Advances in Robotics, ASME Winter Annual Meeting, Anaheim, California*, DSC-42:141–147, 1992.
- [50] Z. Shiller. Time-Energy Optimal Control of Articulated Systems with Geometric Path Constraints. In *IEEE International Conference on Robotics and Automation*, pages 2680–2685, San Diego, California, 1994.
- [51] Z. Shiller, H. Chang, and V. Wong. The Practical Implementation of Time-Optimal Control for Robotic Manipulators. *Robotics and Computer-Integrated Manufacturing*, 12(1):29–39, 1996.
- [52] Z. Shiller and S. Dubowsky. The Acceleration Map and Its Use in Minimum Time Motion Planning of Robotic Manipulators. In *ASME International Conference on Computers in Engineering*, pages 229–234, New York, New York, August 1987.
- [53] Z. Shiller and S. Dubowsky. Robot Path Planning with Obstacles, Actuator, Gripper, and Payload Constraints. *The International Journal of Robotics Research*, 8(6):3–18, 1989.
- [54] Z. Shiller and H.H. Lu. Computation of Path Constrained Time Optimal Motions with Dynamic Singularities. *ASME Transactions, Journal of Dynamic Systems, Measurement, and Control*, 114(1):34–40, March 1992.
- [55] K.G. Shin and N.D. McKay. Minimum-Time Control of Robotic Manipulators with Geometric Path Constraints. *IEEE Transactions on Automatic Control*, AC-30(6):531–541, June 1985.
- [56] K.G. Shin and N.D. McKay. A Dynamic Programming Approach to Trajectory Planning of Robotic Manipulators. *IEEE Transactions on Automatic Control*, AC-31(6):491–500, June 1986.

- 
- [57] D. Simon. The Application of Neural Networks to Optimal Robot Trajectory Planning. *Robotics and Autonomous Systems*, 11:23–34, 1993.
- [58] S. Singh and M.C. Leu. Optimal Trajectory Generation for Robotic Manipulators using Dynamic Programming. *ASME Transactions, Journal of Dynamic Systems, Measurement, and Control*, 109:88–96, June 1987.
- [59] J.J.E. Slotine and H.S. Yang. Improving the Efficiency of Time-Optimal Path-Following Algorithms. *IEEE Transactions on Robotics and Automation*, 5(1):118–124, February 1989.
- [60] E.D. Sontag and H.J. Sussmann. Remarks on the Time-Optimal Control of Two-Links Manipulators. In *24th Conference on Decision and Control*, pages 1646–1652, Fort Lauderdale, Florida, December 1985.
- [61] E.D. Sontag and H.J. Sussmann. Time-Optimal Control of Manipulators. In *IEEE International Conference on Robotics and Automation*, pages 1692–1697, San Francisco, California, April 1986.
- [62] H.Y. Tam. Minimum Time Closed-Loop Tracking of a Specified Path by Robot. In *29th Conference on Decision and Control*, pages 971–976, Honolulu, Hawaii, December 1990.
- [63] M. Tarkiainen and Z. Shiller. Time Optimal Motions of Manipulators with Actuator Dynamics. In *IEEE International Conference on Robotics and Automation*, pages 725–730, Atlanta, Georgia, 1993.
- [64] M. Vukobratovic and M. Kircanski. A Method for Optimal Synthesis of Manipulator Robot Trajectories. *ASME Transactions, Journal of Dynamic Systems, Measurement, and Control*, 104:188–193, June 1982.
- [65] D. Wang and Y. Hamam. Optimal Trajectory Planning of Manipulators with Collision Detection and Avoidance. *The International Journal of Robotics Research*, 11(5):460–468, October 1992.
- [66] H. Wappenhans, J. Holzl, J Steinle, and F. Pfeiffer. Optimal Trajectory Planning with Application to Industrial Robots. *International Journal of Advanced Manufacturing Technology*, 9:49–55, 1994.

# Appendix A

## The Flexible Tolerance Method

The Flexible Tolerance Method FTM was introduced by Paviani and Himmelblau [44]. The detailed description of the algorithm and its concepts is given in [27]. Herein, an overview of the method is presented along with the details of its implementation for the SPCTOM trajectory planning problem.

In this appendix, the symbols used within the main body of the thesis are overloaded. This is done in order to maintain the common mathematical notation. All symbols are redefined as they are used, but the definitions apply only to this appendix.

### A.1 Overview of the FTM

Using the FTM, the general optimization problem is:

$$\text{Minimize: } f(\mathbf{x}) \quad \mathbf{x} \in R^n \tag{A-1}$$

$$\text{Subject to: } h_i(\mathbf{x}) = 0 \quad i = 1, \dots, l \tag{A-2}$$

$$g_i(\mathbf{x}) \geq 0 \quad i = l + 1, \dots, m, \tag{A-3}$$

where  $f(\mathbf{x})$  is the objective function,  $h_i(\mathbf{x})$  are the equality constraints,  $g_i(\mathbf{x})$  are the inequality constraints, and  $R^n$  is the  $n$ -dimensional space where  $\mathbf{x}$  lies. This problem is reformulated into the

following, simpler but equivalent problem with only one constraint:

$$\min: \quad f(\mathbf{x}) \quad \mathbf{x} \in R^n \quad (\text{A-4})$$

$$\text{subject to: } \Phi^{(k)} - T(\mathbf{x}) \geq 0, \quad (\text{A-5})$$

where  $\Phi^{(k)}$  is the value of the flexible tolerance criterion at the  $k$ th stage of the optimization. This criterion also serves as the stopping condition for the search. The cost function  $f(\mathbf{x})$  and the equality and inequality constraints in Equations (A-2) and (A-3) may be linear and/or non-linear functions of the variables in  $\mathbf{x}$ . The basic strategy of the method is to improve the value of the cost function by using information provided by feasible points, as well as certain nonfeasible points called *near-feasible* points. The limits on the near-feasibility are made more restrictive as the search advances until, in the limit, only feasible points are accepted. As a result of this strategy, the algorithm does not spend a considerable portion of computation time on satisfying rigorous feasibility constraints and its computational efficiency is greatly improved.

In (A-5),  $T(\mathbf{x})$  is a positive functional of all the equality and/or inequality constraints of the original problem and it is used as a measure of the constraint violation, while  $\Phi$  is selected as a positive decreasing function of the  $\mathbf{x}$  points in  $R^n$ . Thus,  $\Phi$  serves as both a limit on the constraint violation throughout the search and as a criterion for the termination of the search.

The tolerance criterion is used to categorize points in  $R^n$ . At the  $k$ -th stage of the optimization, a point  $\mathbf{x}^{(k)}$  is said to be:

1. Feasible, if  $T(\mathbf{x}) = 0$
2. Near-feasible, if  $0 \leq T(\mathbf{x}) \leq \Phi^{(k)}$
3. Nonfeasible, if  $T(\mathbf{x}) > \Phi^{(k)}$ .

A small value of  $T(\mathbf{x}^{(k)})$  implies that  $\mathbf{x}^{(k)}$  is relatively near to the feasible region, and a large value of  $T(\mathbf{x}^{(k)})$  implies that  $\mathbf{x}^{(k)}$  is relatively far from the feasible region.

The region of near-feasibility is therefore defined by:

$$\Phi^{(k)} - T(\mathbf{x}) \geq 0. \quad (\text{A-6})$$

On a transition from  $\mathbf{x}^{(k)}$  to  $\mathbf{x}^{(k+1)}$ , the move is said to be feasible if  $0 \leq T(\mathbf{x}^{(k+1)}) \leq \Phi^{(k)}$ , and

nonfeasible if  $\Phi^{(k)} < T(\mathbf{x}^{(k+1)})$ .

The flexible polyhedron method (FPM) of Nelder and Mead [42] is used to minimize  $f(\mathbf{x})$  when the constraint (A-5) is not active. The general idea is to reduce  $\Phi^{(k)}$  as the search progresses, thus tightening the near-feasibility region, and to segregate the minimization of  $f(\mathbf{x})$  from the steps taken to satisfy the constraint (A-5). For a given value of  $\Phi^{(k)}$ , the value of  $T(\mathbf{x})$  at  $\mathbf{x}^{(k+1)}$  will be:

- either  $T(\mathbf{x}^{(k+1)}) \leq \Phi^{(k)}$ , in which case  $\mathbf{x}^{(k+1)}$  is either a feasible or near-feasible point and the move is accepted as a permitted move;
- or  $T(\mathbf{x}^{(k+1)}) > \Phi^{(k)}$ , in which case  $\mathbf{x}^{(k+1)}$  is nonfeasible and a point  $\mathbf{x}$  in the feasible or near-feasible region must be found in lieu of  $\mathbf{x}^{(k+1)}$ .

It is possible to use the same minimization technique for replacing an infeasible  $\mathbf{x}^{(k+1)}$  with a feasible point, but any algorithm that achieves this purpose is sufficient.

In summary, the FTM is a nonlinear constraint optimization technique in which all equality and inequality constraints are transformed into one single inequality constraint. This constraint defines the near-feasibility limit at each stage of the optimization. The algorithm improves the value of the cost function using the FPM as the “outer” optimization when the constraint is not active. When the FPM generates an infeasible point (the constraint becomes active), the FTM algorithm uses another “inner” optimization to replace it with a feasible or near-feasible point. The strategy is to progressively tighten the near-feasibility region and to separate the two optimizations.

## A.2 Overview of the FPM

The Flexible Polyhedron Method, FPM [42], is briefly presented here since it is used in the implementation of the FTM, as the “outer”, unconstrained optimization.

The FPM is an unconstrained multi-dimensional minimization which requires only function evaluations. In advancing the search, the method uses information about the value of the objective function at some points called the vertices of the polyhedron. In an  $n$ -dimensional search space, information about  $n + 1$  points is needed.

The polyhedron is a geometrical figure consisting of  $n + 1$  points and all their interconnecting line segments, polygonal faces, etc. At any stage of the optimization, the polyhedron encloses a finite  $n$ -dimensional volume.

The algorithm replaces the point with the largest value of the objective function (“highest” point) with a lower point. One stage of the search consists of the following steps:

- The highest point is reflected through the centroid of the remaining  $n$  points and the reflection point is kept if it is higher than the lowest point but lower than at least two other points.
- If the reflection point is lower than the lowest point, it is expanded in the same direction. The expansion point is kept if it is lower than the lowest point. Otherwise, the reflection point is kept.
- If the reflection failed, the highest point is contracted toward the centroid and the contraction point is kept if it is lower than the highest point.
- If contraction failed, all points are pulled around the lowest (best) point.

Thus, the polyhedron tends to conserve its volume, but it might expand when a good search direction is found, or it might contract when it reaches a “valley floor”.

The search is initialized by providing it with  $n + 1$  points and it is stopped when the decrease in the objective function value is smaller than some predetermined tolerance.

### A.3 Implementation Details

The details of the FTM implementation for computing SPCTOM trajectories are presented herein. All the references will be made to the elbow manipulator in Table 2.1 along the parabolic path (2.1) and figures related to this example will be used to illustrate the notation.

As explained in Chapter 3, the vector of optimization variables is:

$$\mathbf{x} = \left( \begin{array}{cccccc} \frac{ds}{ds} 0 & \dot{s}_1 & \dots & \dot{s}_p & \frac{ds}{ds} f \\ \frac{ds}{ds} m,0 & \dot{s}_{m,1} & & \dot{s}_{m,p} & \frac{ds}{ds} m,f \end{array} \right)^T. \quad (\text{A-7})$$

As shown in Figure 3.3, for the chosen example,  $p = 5$ , i.e., there are 5 switching points in the PCTOM, and the dimension of the space to be searched is  $n = 7$ .

The objective function  $f(\mathbf{x})$  is the motion time. It is computed assuming that the pseudo-jerk is controlled and is kept constant over each step. Thus, if two consecutive points along the trajectory are considered, then the time taken by the manipulator to move from  $s_i, \dot{s}_i, \ddot{s}_i$  to  $s_{i+1}, \dot{s}_{i+1}, \ddot{s}_{i+1}$

is determined by solving the system:

$$\dot{s}_{i+1} = \dot{s}_i + \ddot{s}_i \cdot t + \frac{\dddot{s}_i \cdot t^2}{2} \quad (\text{A-8})$$

$$s_{i+1} = s_i + \dot{s}_i \cdot t + \frac{\ddot{s}_i \cdot t^2}{2} + \frac{\dddot{s}_i \cdot t^3}{6}, \quad (\text{A-9})$$

where  $t$  and  $\ddot{s}_i$  are considered unknown. The state is then updated according to:

$$\ddot{s}_{i+1} = \ddot{s}_i + \dddot{s}_i \cdot t. \quad (\text{A-10})$$

The inequality constraints (A-3) for the problem of SPCTOM are defined as:

$$g_{4(i-1)+1}(\mathbf{x}) = 1 - \max_{\dot{s}(s)} \frac{T_i}{T_{max,i}}, \quad (\text{A-11})$$

$$g_{4(i-1)+2}(\mathbf{x}) = 1 - \max_{\dot{s}(s)} \frac{T_i}{T_{min,i}}, \quad (\text{A-12})$$

$$g_{4(i-1)+3}(\mathbf{x}) = 1 - \max_{\dot{s}(s)} \frac{\dot{T}_i}{\dot{T}_{max,i}}, \quad (\text{A-13})$$

$$g_{4(i-1)+4}(\mathbf{x}) = 1 - \max_{\dot{s}(s)} \frac{\dot{T}_i}{\dot{T}_{min,i}}, \quad (\text{A-14})$$

where  $i$  takes values from 1 to dof, the number of degrees of freedom of the manipulator. This definition ensures that whenever any of the joint torques and/or joint jerks exceeds its limits, the respective constraint becomes negative. For the case presented in Section 3.3.2, Figure 3.4, the robot is an elbow manipulator and only the positional degrees of freedom are considered; therefore dof= 3.

The inequality constraints  $g_i(\mathbf{x}) \geq 0, i = 1, \dots, 4 \cdot \text{dof}$ , with  $g_i(\mathbf{x})$  defined by Equations (A-11)-(A-14), represent a direct enforcement of the actuator torque and jerk constraints (2.38) and (2.32). The actuator constraints are used directly instead of the state constraints (2.63)-(2.64) and the control constraints (2.65) to simplify the computation.

$T(\mathbf{x})$  is a positive functional of all the equality and/or inequality constraints. It is defined as:

$$T(\mathbf{x}) = \begin{cases} \max_i(1 - g_i(\mathbf{x})) & \text{if } \exists i \text{ such that } g_i(\mathbf{x}) \geq 1 \\ 0 & \text{otherwise.} \end{cases} \quad (\text{A-15})$$

This definition differs from the one in [27]. In [27],  $T(\mathbf{x})$  is defined as:

$$T(\mathbf{x}) = \left[ \sum_{i=1}^m h_i^2(\mathbf{x}) + \sum_{i=l+1}^m \mathcal{U}_i g_i^2(\mathbf{x}) \right]^{1/2}, \quad (\text{A-16})$$

where  $\mathcal{U}_i$  is the Heaviside operator such that  $\mathcal{U}_i = 0$  for  $g_i(\mathbf{x}) \geq 0$  and  $\mathcal{U}_i = 1$  for  $g_i(\mathbf{x}) < 0$ . Therefore, if  $\sum_{i=1}^l h_i^2(\mathbf{x})$  is convex and the  $g_i(\mathbf{x}), i = l + 1 \dots, m$ , are concave functions, then  $T(\mathbf{x})$  is a convex function with a global minimum at  $T(\mathbf{x}) = 0$ . Since for the SPCTOM problem the inequality constraints are not analytic functions, but can only be computed numerically, no information is available about their convexity. Then, the advantage of Equation (A-16) as the definition of  $T(\mathbf{x})$  is lost. Equation (A-15), on the other hand, is easier to compute and interpret.

In the implementation presented here, the tolerance criterion  $\Phi$  is selected to be:

$$\Phi^{(k)} = \min\{\Phi^{(k-1)}; \kappa \sum_{i=1}^{n+1} \|x_i^{(k)} - x_{centr}^{(k)}\|\} \quad (\text{A-17})$$

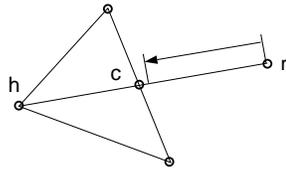
with  $\kappa$  a constant that is appropriately chosen, and  $x_{centr}^{(k)}$  the centroid of the polyhedron after the elimination of the point with the highest value of the cost. The initial value of the tolerance criterion,  $\Phi^0$ , is set to 0.2. Thus, the tolerance criterion is a positive non-increasing function of the sequence of points  $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(k)}, \dots, \mathbf{x}^*$  generated during the progression of the search.

Unlike the implementation of the FTM in [27] (where the FPM is used for both the outer and the inner optimizations), the strategy for the inner optimization used herein (that is, for replacing an infeasible point by a feasible or near-feasible one) is a simple line search through bisection. The choice is justified by the high computational cost involved in the motion time and the constraint evaluations. Depending on which point became infeasible, the line search is performed as follows:

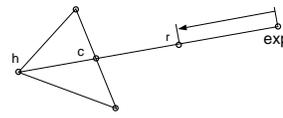
- If a reflection yielded the infeasible point, the line from the infeasible vertex toward the one from which it resulted is searched. The search stops before the centroid (in order to preserve search space dimension). This situation is schematically depicted in Figure A.1(a) for a polyhedron with three vertices.
- If an extension yielded the infeasible point, the line from the infeasible vertex toward the reflected one is performed. The search stops at the reflected point (Figure A.1(b)).
- If a contraction yielded the infeasible point, the line from the infeasible vertex toward the one

from which it resulted is performed (Figure A.1(c)).

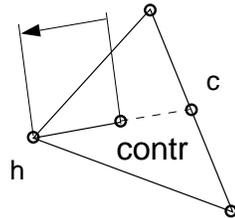
- If more vertices became infeasible as the result of a collapse or a change in the tolerance criterion, each infeasible vertex is moved toward the origin. As vertices closer to origin represent slower motions, this strategy is guaranteed to find feasible points. However, the move is not performed through bisection, but in small steps with predetermined length (Figure A.1(d)).



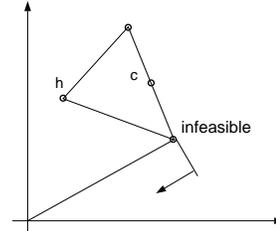
(a) Infeasible reflection.



(b) Infeasible expansion.



(c) Infeasible contraction.



(d) Infeasible vertex due to a change in the tolerance.

Figure A.1: The line searches during the replacement of the different types of infeasible points generated by the FPM.

The algorithm of the FTM as implemented for the SPCTOM problem is presented in Figures A.2 and A.3. The terminology is the one used in the FPM of Nelder and Mead [42].

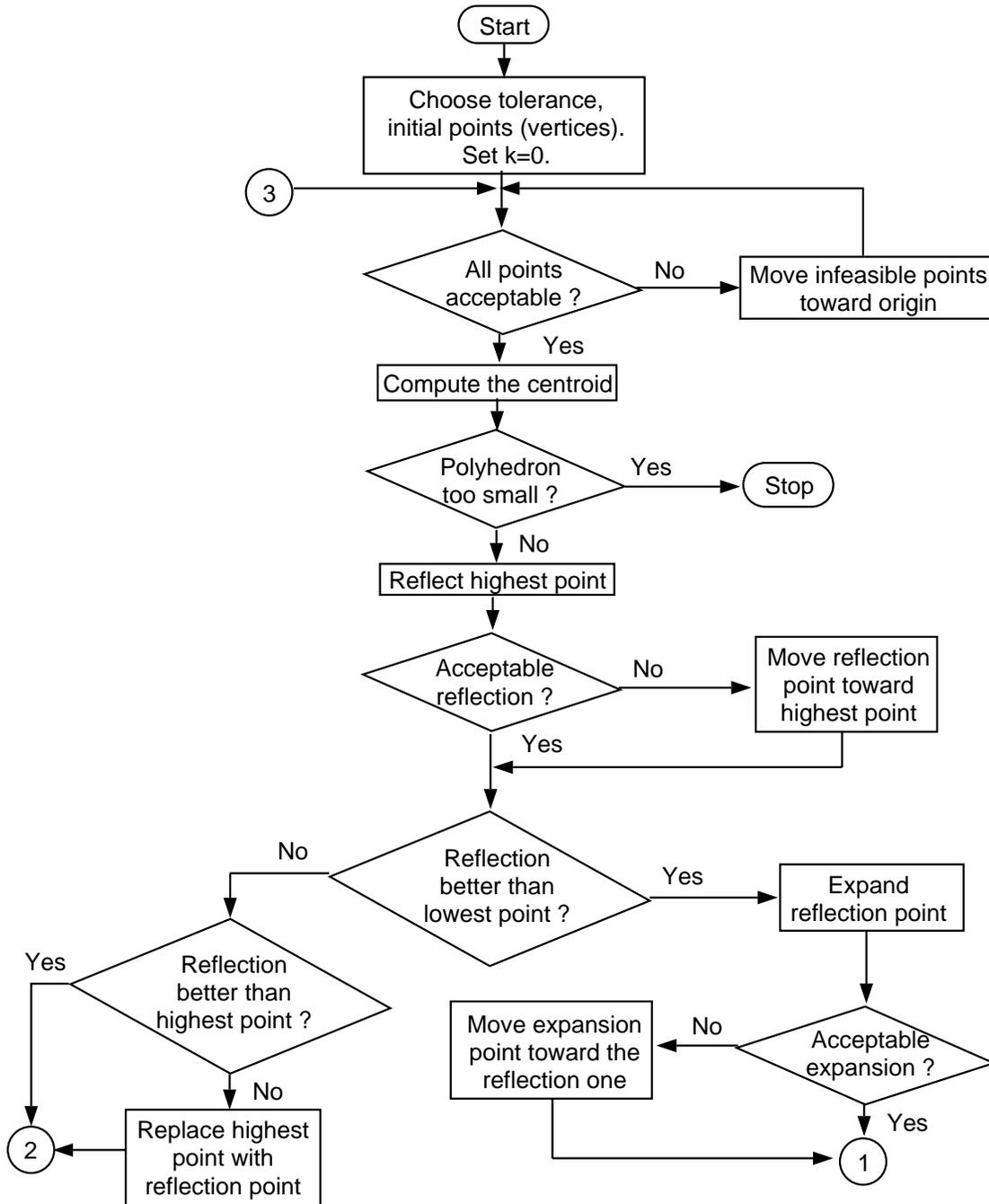


Figure A.2: Flow diagram of the FTM for the SPCTOM problem - Part 1.

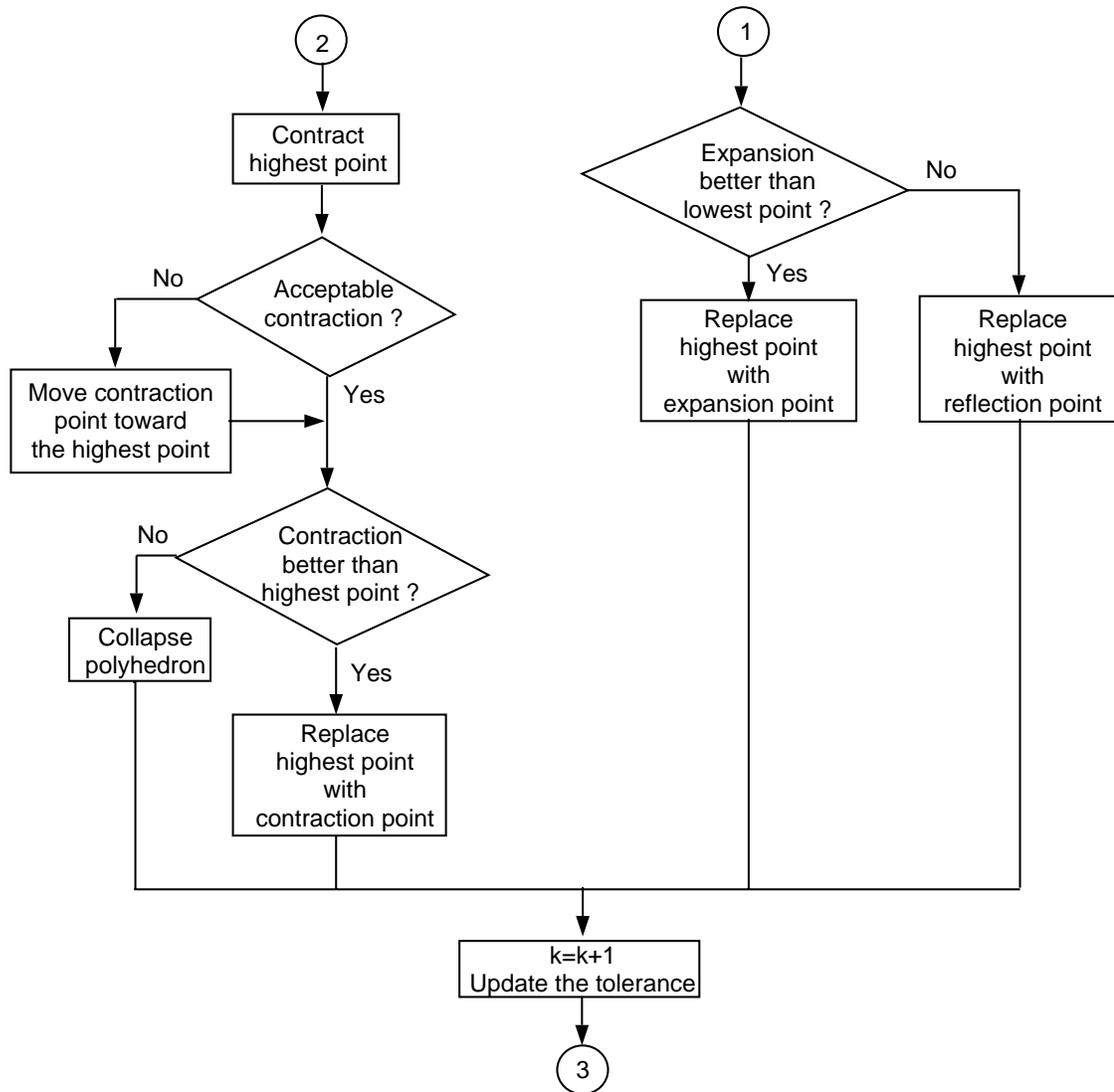


Figure A.3: Flow diagram of the FTM for the SPCTOM problem - Part 2.

## Appendix B

# Robot Dynamic Models

In the simulations presented in Chapter 4, two manipulators are used: (i) the elbow manipulator in [45] and (ii) the SCORBOT ER VII in the IAL. The dynamic models of these robots are detailed in this appendix.

As shown in Chapter 2, the manipulator Lagrangian dynamics is given by:

$$\mathbf{M}(\mathbf{q}) \cdot \ddot{\mathbf{q}} + \dot{\mathbf{q}}^{\mathbf{T}} \cdot \mathbf{C}(\mathbf{q}) \cdot \dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \mathbf{T}, \quad (\text{B-1})$$

where  $\mathbf{q} \in \mathbf{R}^n$  is the vector of joint positions,  $\mathbf{T} \in \mathbf{R}^n$  is the vector of actuator torques,  $\mathbf{M}(\mathbf{q}) \in \mathbf{R}^{n \times n}$  is the inertia matrix of the manipulator,  $\mathbf{C}(\mathbf{q}) \in \mathbf{R}^{n \times n \times n}$  is a third order tensor representing the coefficients of the centrifugal and Coriolis forces,  $\mathbf{G}(\mathbf{q}) \in \mathbf{R}^n$  is the vector of gravity terms, and  $\dot{\phantom{x}}$  denotes the derivative with respect to time.

The third order dynamics, needed in order to incorporate the actuator jerks, is derived from Equation (B-1) by differentiation with respect to time:

$$\mathbf{M}(\mathbf{q}) \cdot \ddot{\mathbf{q}} + \dot{\mathbf{M}}(\mathbf{q}) \cdot \ddot{\mathbf{q}} + \ddot{\mathbf{q}}^{\mathbf{T}} \cdot \mathbf{C}(\mathbf{q}) \cdot \dot{\mathbf{q}} + \dot{\mathbf{q}}^{\mathbf{T}} \cdot \dot{\mathbf{C}}(\mathbf{q}) \cdot \dot{\mathbf{q}} + \dot{\mathbf{q}}^{\mathbf{T}} \cdot \mathbf{C}(\mathbf{q}) \cdot \ddot{\mathbf{q}} + \dot{\mathbf{G}}(\mathbf{q}) = \dot{\mathbf{T}}. \quad (\text{B-2})$$

Equations (B-1) and (B-2) show that the robot model is known if the inertia matrix, the Christoffel symbols, and the gravity terms are known. All the other terms involved in Equations (B-1)-(B-2) are derived by differentiation with respect to time.

## B.1 Elbow Manipulator Dynamic Model

The dynamic parameters of the elbow manipulator in [45] are given in Table 4.1. Using them, the inertia matrix, the Christoffel symbols, and the gravity terms are computed and given hereafter.

The non-zero components of the inertia matrix,  $\mathbf{M}(\mathbf{q})$ , are:

$$M_{11} = 1.89c_{q_3} + 8.4455 + 2.4295c_{2q_2} + 1.89c_{2q_2+q_3} + 0.896c_{2q_2+2q_3} \quad (\text{B-3})$$

$$M_{22} = 3.78c_{q_3} + 6.7771 \quad (\text{B-4})$$

$$M_{23} = 1.89c_{q_3} + 1.812 \quad (\text{B-5})$$

$$M_{32} = 1.89c_{q_3} + 1.812 \quad (\text{B-6})$$

$$M_{33} = 1.812, \quad (\text{B-7})$$

where  $c_q$  is a notation for  $\cos(q)$ .

The Christoffel symbols are computed from the inertia matrix, according to:

$$C_{ijk} = \frac{1}{2} \left( \frac{\partial M_{ij}}{\partial q_k} + \frac{\partial M_{ik}}{\partial q_j} - \frac{\partial M_{jk}}{\partial q_i} \right). \quad (\text{B-8})$$

Performing all the calculations, the non-zero Christoffel symbols are obtained as:

$$C_{112} = -1.89s_{2q_2+q_3} - 0.896s_{2q_2+2q_3} - 2.4295s_{2q_2} \quad (\text{B-9})$$

$$C_{113} = -0.945s_{2q_2+q_3} - 0.896s_{2q_2+2q_3} - 0.945s_{q_3} \quad (\text{B-10})$$

$$C_{121} = -1.89s_{2q_2+q_3} - 0.896s_{2q_2+2q_3} - 2.4295s_{2q_2} \quad (\text{B-11})$$

$$C_{113} = -0.945s_{2q_2+q_3} - 0.896s_{2q_2+2q_3} - 0.945s_{q_3} \quad (\text{B-12})$$

$$C_{211} = 1.89s_{2q_2+q_3} + 0.896s_{2q_2+2q_3} + 2.4295s_{2q_2} \quad (\text{B-13})$$

$$C_{223} = -1.89s_{q_3} \quad (\text{B-14})$$

$$C_{232} = -1.89s_{q_3} \quad (\text{B-15})$$

$$C_{233} = -1.89s_{q_3} \quad (\text{B-16})$$

$$C_{311} = 0.945s_{2q_2+q_3} + 0.896s_{2q_2+2q_3} + 0.945s_{q_3} \quad (\text{B-17})$$

$$C_{322} = -1.89s_{q_3}, \quad (\text{B-18})$$

where  $s_q$  is a notation for  $\sin(q)$ .

Finally, the non-zero components of the vector of gravity terms are:

$$G_2 = 66.5118c_{q_2} + 24.7212c_{q_2+q_3} \quad (\text{B-19})$$

$$G_3 = 24.7212c_{q_2+q_3}. \quad (\text{B-20})$$

## B.2 SCORBOT ER VII Dynamic Model

The DH and dynamic parameters of the first three links (positional dof) of the SCORBOT ER VII in the IAL are given in Tables 4.8 and 4.9, respectively. Using them, after performing all the required calculations, the elements of the inertia matrix of the SCORBOT ER VII are:

$$M_{11} = 0.1575c_{2q_2+q_3} + 0.033c_{q_2} + 0.1575c_{q_3} + 0.0478c_{2q_2+2q_3} + 0.27725c_{2q_2} + 0.0525c_{q_2+q_3} + 0.9499 \quad (\text{B-21})$$

$$M_{12} = -0.019s_{q_2+q_3} - 0.012s_{q_2} \quad (\text{B-22})$$

$$M_{13} = -0.019s_{q_2+q_3} \quad (\text{B-23})$$

$$M_{21} = -0.019s_{q_2+q_3} - 0.012s_{q_2} \quad (\text{B-24})$$

$$M_{22} = 0.315c_{q_3} + 0.94 \quad (\text{B-25})$$

$$M_{23} = 0.1575c_{q_3} + 0.1656 \quad (\text{B-26})$$

$$M_{31} = -0.019s_{q_2+q_3} \quad (\text{B-27})$$

$$M_{32} = 0.1575c_{q_3} + 0.1656 \quad (\text{B-28})$$

$$M_{33} = 0.1656, \quad (\text{B-29})$$

the non-zero elements of the Christoffel symbols are:

$$C_{112} = -0.1575s_{2q_2+q_3} - 0.0478s_{2q_2+2q_3} - 0.0165s_{q_2} - 0.02625s_{q_2+q_3} - 0.27725s_{2q_2} \quad (\text{B-30})$$

$$C_{113} = -0.07875s_{2q_2+q_3} - 0.0478s_{2q_2+2q_3} - 0.07875s_{q_3} - 0.02625s_{q_2+q_3} \quad (\text{B-31})$$

$$C_{121} = -0.1575s_{2q_2+q_3} - 0.0478s_{2q_2+2q_3} - 0.0165s_{q_2} - 0.02625s_{q_2+q_3} - 0.27725s_{2q_2} \quad (\text{B-32})$$

$$C_{122} = -0.012c_{q_2} - 0.019c_{q_2+q_3} \quad (\text{B-33})$$

$$C_{123} = -0.019c_{q_2+q_3} \quad (\text{B-34})$$

$$C_{131} = -0.07875s_{2q_2+q_3} - 0.0478s_{2q_2+2q_3} - 0.07875s_{q_3} - 0.02625s_{q_2+q_3} \quad (\text{B-35})$$

$$C_{132} = -0.019c_{q_2+q_3} \quad (\text{B-36})$$

$$C_{133} = -0.019c_{q_2+q_3} \quad (\text{B-37})$$

$$C_{211} = 0.1575s_{2q_2+q_3} + 0.0478s_{2q_2+2q_3} + 0.0165s_{q_2} + 0.02625s_{q_2+q_3} + 0.27725s_{2q_2} \quad (\text{B-38})$$

$$C_{223} = -0.1575s_{q_3} \quad (\text{B-39})$$

$$C_{232} = -0.1575s_{q_3} \quad (\text{B-40})$$

$$C_{233} = -0.1575s_{q_3} \quad (\text{B-41})$$

$$C_{311} = 0.07875s_{2q_2+q_3} + 0.0478s_{2q_2+2q_3} + 0.07875s_{q_3} + 0.02625s_{q_2+q_3} \quad (\text{B-42})$$

$$C_{322} = 0.1575s_{q_3}, \quad (\text{B-43})$$

and the non-zero components of the gravity vector result as:

$$G_2 = -3.2373c_{q_2} - 5.15c_{q_2+q_3} \quad (\text{B-44})$$

$$G_3 = -5.15c_{q_2+q_3}. \quad (\text{B-45})$$