

On-Line Smooth Trajectory Planning for Manipulators

by

Sonja Macfarlane

B.Sc.Eng., The University of New Brunswick, 1999

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

August 2001

© Sonja Macfarlane, 2001

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Mechanical Engineering

The University of British Columbia
Vancouver, Canada

Date August 31, 2001

Abstract

This work proposes a smooth, on-line, trajectory generation method for manipulators. The ability to compute a trajectory on-line increases the ease and speed with which manipulator programming can be done. It also reduces a manipulator's downtime and provides a certain amount of autonomy to the manipulator. These properties are especially desirable in industrial applications. Smooth trajectories are used to avoid controller and actuator saturation thereby reducing actuator wear, and improving tracking accuracy. They also improve manipulator motion times.

Existing trajectory generation strategies can be divided into two groups: those based on the dynamic model of the manipulator, and those based on the kinematic limits of the actuators. Trajectory generation methods based on the dynamic model of the manipulator typically cannot be implemented on-line due to the required optimization processes, which can have widely varying computational requirements. Furthermore, their complexity is prohibitive for industrial implementation. Those based on the kinematic limits of the actuators provide computationally simpler solutions and can be used on-line. However existing kinematic-based trajectory planning algorithms that are suitable for use on-line, either do not incorporate jerk limits or involve the use of transcendental functions to limit jerk. The latter method increases the computational load.

The trajectories proposed herein can be tracked using typical industrial controllers. The solution is based on forming a fifth order polynomial (quintic) trajectory following a sine-wave template, such that the velocity, acceleration and jerk limits are respected throughout the trajectory. All trajectories between two points can be planned by calculating the time, position, speed and acceleration at a maximum of seven control points, which describe the trajectory. This provides a tractable near time-optimal trajectory which can easily be generated on-line. The trajectory paths consist of straight lines joined by splines (blends) to ensure that the manipulator does not stop at each way-point. Limitations on the blend speed are proposed to ensure that the acceleration and jerk limits are respected during the blend. In addition, a high level planning architecture is provided for on-line trajectory planning.

An implementation of the proposed trajectory generation technique and its comparison to an industrial technique show improved motion times, consistent kinematic profiles and improved path tracking.

Table of Contents

Abstract	ii
Table of Contents.....	iii
List of Tables	viii
List of Figures.....	x
Terminology.....	xii
Nomenclature.....	xiii
Acknowledgements	xviii
1 Introduction	1
1.1. Background.....	1
1.2. Industrial Motivation.....	2
1.3. Research Objectives	3
1.4. Thesis Outline	3
2 Literature Review	5
2.1. Path Definition.....	5
2.2. Time-Optimal Trajectory Planning with Dynamic Limitations.....	6
2.3. Practical Implementation of PCTOM Trajectories	8
2.3.1. Controller Based Tracking of Purely Time-Optimal Motions	8
2.3.2. Incorporation of Torque Rate Limitations (Smooth Time-Optimal Motion).....	9
2.4. Trajectory Generation with Kinematic Limitations	10
2.4.1. On-Line Trajectory Generation.....	10
2.4.2. On-Line Kinematic Trajectory Planning	12
2.4.2.1. Pick-and-Place Operations	12
2.4.2.2. Multiple Way-Points with Prescribed Path	13
2.4.2.3. Kinematic Limitations at, and Between, Way-Points	14
2.5. Summary of Trajectory Planning Techniques	16
2.6. Problem Formulation	16
2.6.1. Implementation	18
2.6.1.1. Task Space Planning Versus Joint Space Planning.....	18

2.6.1.2. Path-Related Constraints	19
2.6.1.3. High Level Planner	19
2.7. Summary	20
3 Methodology.....	21
3.1. Trajectory Functions (Building Blocks).....	21
3.1.1. Sine wave template.....	23
3.1.2. Non-Limited Trajectory.....	25
3.1.3. Limited Trajectories	28
3.2. General Trajectory Generation Between Any Two Way-Points.....	30
3.3. Summary	35
4 Blends.....	36
4.1. Lloyd and Hayward's Blending Function	37
4.2. Geometric Blends.....	38
4.2.1. Conditions Necessary for a Geometric Blend.....	39
4.2.2. Velocity, Acceleration and Jerk Profiles for a 2D Geometric Blend.....	41
4.2.3. Limiting Acceleration and Jerk	43
4.2.4. Extension into n-dimensions	44
4.3. Summary	45
5 Connecting the Dots.....	46
5.1. Task Space Straight-Line Mode.....	47
5.1.1. Cartesian Dimensions	47
5.1.2. Orientation	49
5.1.3. Combining Cartesian and Orientation Trajectories and Generating a Trajectory	51
5.2. Joint Space Mode.....	52
5.2.1. Straight Line Joint Space Motion.....	52
5.2.2. Independent Joint Motion	53
5.3. Planning a Straight-Line Motion Between Two Way-Points With Variable Speeds.....	53
5.4. Backward Planning Issues and Trajectory Generation Between N Way-Points.....	56
5.5. Summary	58

6 Application to a Manipulator	61
6.1. Way-Points	62
6.2. High Level Planner	62
6.2.1. Preliminary Considerations.....	62
6.2.2. Vetting	63
6.2.3. Trajectory Planning and Parser Points.....	66
6.2.3.1. Flag Setting Example.....	67
6.3. Parser.....	68
6.3.1. Straight Line Parsing	68
6.3.2. Task Space Parsing.....	69
6.3.3. Blend Parsing	70
6.4. Summary	70
7 Experimental Results.....	71
7.1. Experimental Setup	71
7.2. Comparison to a Typical Industrial Trajectory Generation Method	73
7.2.1. Task-Space Stop Point to Stop Point Motion.....	74
7.2.1.1. Experiment 1: Task-Space Straight Line with Constant Orientation.....	74
7.2.1.2. Experiment 2: Task-Space Straight Line with Constant Orientation.....	78
7.2.1.3. Experiment 3: Task Space Motion with a Change in Orientation	82
7.2.1.4. Summary of Results for Task Space Stop Point to Stop Point Motions.....	83
7.2.2. Task-Space Multiple Way-Point Motion.....	84
7.2.3. Joint-Space Stop Point to Stop Point Motion.....	86
7.2.3.1. Experiment 1: Joint Space Straight Line.....	86
7.2.3.2. Experiment 2: Joint Space Straight Line.....	89
7.2.3.3. Summary of Joint Space Stop Point to Stop Point Motion Results	89
7.2.4. Joint Space Multiple Way-Point Motion	92
7.3. Comparison to SPCTOM	94
7.4. Computation Time Issues	95
7.5. Summary	95
8 Conclusions & Recommendations	97
8.1. Conclusions	97
8.2. Recommendations.....	98

Bibliography	100
---------------------------	------------

Appendices

A Jerk Considerations	104
------------------------------------	------------

B Quintic End Conditions	107
---------------------------------------	------------

B.1. Quintic End Conditions for a Trajectory Based on a Sustained Acceleration Pulse	107
B.2. Quintic End Conditions for a Trajectory Based on an Acceleration Pulse	108
B.3. Algorithm followed when $s_1 > s_2$	109

C Algorithm Proof.....	111
-------------------------------	------------

D Quintic Control Points for Case B (iiia)	114
---	------------

E Blend Acceleration and Jerk Limits.....	115
--	------------

E.1. Constant of Proportionality	116
E.2. Derivation of the Relationship Between the Maximum Acceleration and the Speed for a Geometric Blend.....	119
E.3. Derivation of the Relationship Between the Maximum Jerk and the Speed for a Geometric Blend.....	120

F Estimation of Task Space Kinematic Limits and Calculation of Joint Space Path Limits	124
---	------------

F.1. Joint Space Kinematic Limits.....	124
F.1.1. Joint Space Path Limits	125
F.2. Task Space Path Kinematic Limits	126

G Quaternions.....	128
---------------------------	------------

G.1. Rotation Matrices and Quaternions.....	128
G.2. Quaternion Multiplication and Transpose	129
G.3. Algorithm Used to Generate a Rotation Matrix at Each Sampling Interval	130

H Further Experimental Results	132
H.1. Stop Point to Stop Point Task Space Motions	132
H.1.1. Task Space Experiment 1.....	132
H.1.2. Task Space Experiment 2.....	133
H.1.3. Task Space Experiment 3.....	134
H.1.4. Task Space Experiment 4.....	135
H.1.5. Task Space Experiment 5.....	136
H.1.6. Task Space Experiment 6.....	137
H.1.6.1 Explanation of Results for Task Space Experiment 6	138
H.2. Multiple Way-Point Task Space Motions.....	139
H.2.1. Horizontal Rectangular Motion.....	139
H.2.2. Vertical Rectangular Motion.....	140
H.3. Stop Point to Stop Point Joint Space Motions	142
H.3.1. Joint Space Experiment 1.....	142
H.3.2. Joint Space Experiment 2.....	143
H.3.3. Joint Space Experiment 3.....	144
H.3.4. Joint Space Experiment 4.....	145
H.4. Multiple Way-Point Joint Space Motion	146

List of Tables

6.1.	Way-Point Speeds and Flags at the End of Planning	67
7.1.	Denavit-Hartenberg Parameters for the Positional dof of the SCORBOT ER-VII.....	71
7.2.	Denavit-Hartenberg Parameters for the CRS A465 Manipulator	72
7.3.	Joint Kinematic Limits for the A465.....	73
7.4.	Task Space Kinematic Limits for the A465	74
7.5.	Coordinates of the Straight Line Path Start and End Points for Task Space Experiment 1	76
7.6.	Motion Times and RMS Errors of Task Space Experiment 1	78
7.7.	Coordinates of the Straight Line Path Start and End Points for Task Space Experiment 2	80
7.8.	Motion Times and RMS Errors Corresponding to Task Space Experiment 2.....	81
7.9.	Coordinates of Orientation Motion Start and End Points.....	83
7.10.	Angle Between the Desired Rotation Axis and the Actual Rotation Axis at Time Increments for Task Space Experiment 3	83
7.11.	Coordinates of Way-Points Corresponding to Figure 7.12.	84
7.12.	Motion Times for Task Space Multiple Way-Point Path.....	85
7.13.	Coordinates of Joint Space Straight Line Start and End Points for Joint Space Experiment 1	86
7.14.	Motion Times and Accuracy with which the End Point is Reached for Joint Space Experiment 1 .	88
7.15.	Coordinates of Joint Space Straight Line Start and End Points for Joint Space Experiment 2	89
7.16.	Motion Times for Straight Line for Joint Space Experiment 2.....	89
7.17.	Coordinates of Way-Points for Multiple Way-Point Joint Space Path	92
7.18.	Motion Times for Multiple Way-Point Joint Space Path	93
7.19.	Motion Times for SPCTOM Trajectory.....	94
B.1.	Quintic Control Point Conditions for the SAP Algorithm.....	108
B.2.	Quintic Control Point Conditions for the Acceleration Pulse Algorithm.....	109
H.1.	Coordinates of the Straight Line Path Start and End Points for Task Space Experiment 1	132
H.2.	Results for Task Space Experiment 1	132
H.3.	Coordinates of the Straight Line Path Start and End Points for Task Space Experiment 2	133
H.4.	Results for Task Space Experiment 2	133
H.5.	Coordinates of the Straight Line Path Start and End Points for Task Space Experiment 3	134
H.6.	Results for Task Space Experiment 3	134
H.7.	Coordinates of the Straight Line Path Start and End Points for Task Space Experiment 4	135
H.8.	Results for Task Space Experiment 4	135

H.9.	Coordinates of the Straight Line Path Start and End Points for Task Space Experiment 5	136
H.10.	Results for Task Space Experiment 5	136
H.11.	Coordinates of the Straight Line Path Start and End Points for Task Space Experiment 6	137
H.12.	Results for Task Space Experiment 6	137
H.13.	Coordinates of Way-Points for Task Space Horizontal Rectangular Path	139
H.14.	Results for Horizontal Rectangular Path	139
H.15.	Coordinates of Way-Points for Task Space Vertical Rectangular Path	140
H.16.	Results for Task Space Vertical Rectangular Path	141
H.17.	Coordinates of Joint Space Straight Line Start and End Points for Joint Space Experiment 1	142
H.18.	Results for Joint Space Experiment 1	142
H.19.	Coordinates of Joint Space Straight Line Start and End Points for Joint Space Experiment 2	143
H.20.	Results for Joint Space Experiment 2	143
H.21.	Coordinates of Joint Space Straight Line Start and End Points for Joint Space Experiment 3	144
H.22.	Results for Joint Space Experiment 3	144
H.23.	Coordinates of Joint Space Straight Line Start and End Points for Joint Space Experiment 4	145
H.24.	Results for Joint Space Experiment 4	145
H.25.	Coordinates of Way-Points for Multiple Way-Point Joint Space Path	146
H.26.	Motion Time for the Multiple Way-Point Path at Varying Demanded Speeds	146

List of Figures

T.1.	Trajectory Segment Terminology	xii
2.1.	Proposed Acceleration Profiles.....	14
3.1.	Linear Segment with Parabolic Blend (LSPB) Trajectory Between Way-Points p ₁ and p ₂	22
3.2.	Sine Wave Template for a Positive Change in Acceleration.....	24
3.3.	Smooth Trajectory Generated with a Sequence of Quintics.....	26
3.4.	Speed Limited Trajectory.....	29
3.5.	Flowchart Indicating the Algorithm to Use Based on the Way-Point Speeds	31
3.6.	Illustration of Dlimit	33
3.7.	Illustration of Shortened Acceleration Segments.....	34
3.8.	Template Used to Determine τ in Subcase iib.....	34
4.1.	Blend in Time Domain.....	37
4.2.	Effect of tightness on Path.....	39
4.3.	2D Blend Geometry	40
4.4.	Velocity Profiles for a 2D Geometric Blend	42
4.5.	Acceleration Profiles for a 2D Geometric Blend.....	42
4.6.	Jerk Profiles for a 2D Geometric Blend	43
5.1.	Four Different Cases of Straight Line and Blend Motions.....	48
5.2.	Sketch of a Multiple Point Path	49
5.3.	Rotational Motion Trajectory	50
5.4.	Generation of Trajectories Between Any Two Way-Points p _i and p _{i+1}	54
5.5.	Cartesian Space Trajectory Reference Data Generation	55
5.6.	Joint Space Trajectory Reference Data Generation	56
5.7.	Three Possible Speed Profiles Between Three Way-Points.	57
5.8.	Trajectory Planning Between N Way-Points (Forward Planning).....	59
5.9.	Backplanning.....	60
6.1.	Interaction Between the Three Planning Units	61
6.2.	Vetting Procedure	64
6.3.	Elimination of Way-Points Due to Tightness Considerations	65
6.4.	Special Cases for Point Elimination in Vetting Algorithm.....	65
7.1.	Experimental Setup.....	72
7.2.	Comparison of Tracking Performance of Task Space Experiment 1	75

7.3.	Comparison of Speed During Motion of Task Space Experiment 1	76
7.4.	Comparison of Path Deviations of Task Space Experiment 1	77
7.5.	Acceleration Profiles (filtered) of Task Space Experiment 1	77
7.6.	Comparison of Speed During Motion of Task Space Experiment 2	79
7.7.	Comparison of Tracking Performance of Task Space Experiment 2	79
7.8.	Comparison of Path Deviation During Motion of Task Space Experiment 2	80
7.9.	Comparison of Acceleration Profiles (filtered) of Task Space Experiment 2	80
7.10.	Comparison of Orientation With Respect to World Frame During the Motion for Task Space Experiment 2	81
7.11.	Comparison of at Different Maximum Allowable Speeds	82
7.12.	Comparison of Paths at Different Speeds for Task Space Multiple Way-Point Path	85
7.13.	Comparison of Speed Profiles for Task Space Multiple Way-Point Path.....	86
7.14.	Comparison of Joint Velocities for Joint Space Experiment 1	87
7.15.	Comparison of Industrial Planner and Quintic Planner Acceleration Profiles (filtered) for Joint Space Experiment 1 Trajectory with Demanded Speed of 1016 mm/s (100% MMAS).....	88
7.16.	Comparison of Joint Velocities for Joint Space Experiment 2	90
7.17.	Comparison of Filtered Joint Acceleration Profiles for Different Jerk Limits	91
7.18.	Comparison of Joint Velocity Profiles With Different Jerk Limits	92
7.19.	Comparison of Joint Space Paths for Joint Space Multiple Way-Point Experiment	93
7.20.	Comparison of Quintic Concatenation and SPCTOM Joint Performance	95
7.21.	Comparison of Tracking for SPCTOM and Quintic Concatenation Method	95
A.1.	Jerk, Acceleration, Velocity and Position Profiles Corresponding to a Sine-Based Acceleration Ramp from 0 to amax.....	104
B.1.	Smooth Trajectory Generated with a Sequence of Quintics.....	108
B.2.	Speed Limited Trajectory (• indicate the quintic control points).....	109
E.1.	2D Blend Geometry	115
E.2.	Plot of Acceleration Profile for the 2D Blend of Figure E.1.....	117
E.3.	Figure Illustrating the Variables Used in Equations E.31 and E.32.....	119
E.4.	Jerk Profiles for the Blend of Figure E.1.....	121
E.5.	Diagram Indicating the Geometry Used to Derive Eq. E.51	123
F.1.	Kinematic Limit Projection for a 3D Straight Line Path.....	126
H.1.	Comparison of Paths at Different Speeds for Vertical Rectangular Path.....	140
H.2.	Comparison of Speed Profiles for Task Space Vertical Rectangular Path.....	140

Terminology

Herein, various terms are used to describe the trajectory segments. These terms are modified by the parameter that they describe, and are depicted in Figure T.1:

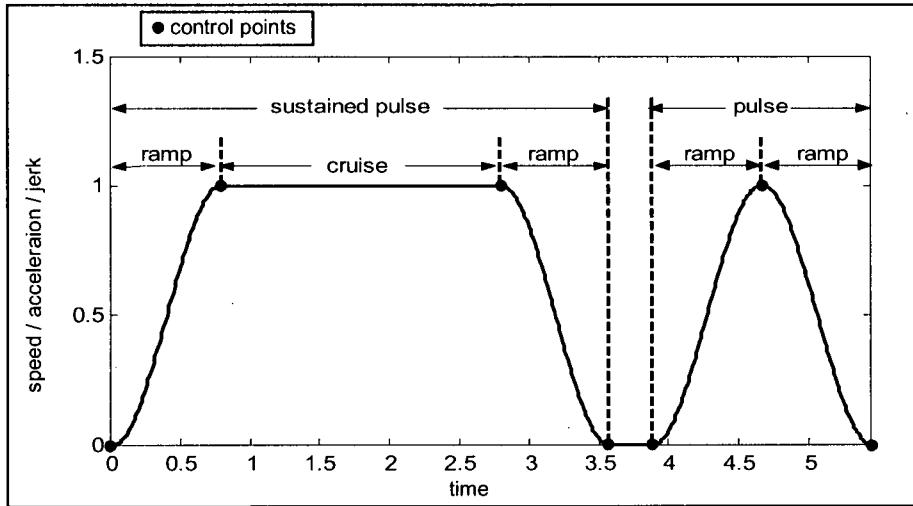


Figure T.1. Trajectory Segment Terminology

- ramp:** A change in speed, acceleration or jerk. For example, a speed ramp is a trajectory segment with a change in speed.
- cruise:** Indicates no change in the relevant parameter, e.g. a speed cruise is a constant speed trajectory segment.
- pulse:** Composed of a ramp-up followed by a ramp-down.
- sustained pulse:** Composed of a ramp-up, a cruise, followed by a ramp-down.

Three types of points are frequently referred to in this document. They are:

- way-points:** Points, provided by the user, designating positions that the machine must pass through. A way-point may be defined in either joint space or task space.
- control points:** Points that provide the end conditions for the quintic polynomials in a trajectory. Control points define a trajectory between two way-points. A control point is assigned time, position, velocity and acceleration.
- stop points:** Way-points with a zero speed condition.

Two spaces are frequently used in this document to define the position of a manipulator:

- task space:** The position vector is defined as a combination of the Cartesian coordinates (i.e. the positional coordinates, x , y , and z) and the orientation coordinates (ψ_x , ψ_y , ψ_z).
- joint space:** The position vector is defined as a vector of joint positions.

Nomenclature

Best	flag used in the high level of the trajectory planner
C	Christoffel symbols of the manipulator (centripetal and Coriolis coefficients)
D	Euclidian distance between two points
D_1	distance covered during a ramp from zero acceleration to a_{\max} in the SAP algorithm
D_{1z}	distance covered during a ramp from zero acceleration to a_{\max} in the SAP algorithm for the case z
D_2	distance covered during the acceleration cruise at a_{\max} in the SAP algorithm
D_{2z}	distance covered during the acceleration cruise at a_{\max} in the SAP algorithm for the case z
D_3	distance covered during a ramp down from a_{\max} to zero acceleration in the SAP algorithm
D_{3z}	distance covered during a ramp down from a_{\max} to zero acceleration in the SAP algorithm for the case z
D_a	distance covered during a SAP used to ramp from s_1 to s_{limit} ($=D_{1a} + D_{2a} + D_{3a}$)
D_b	distance covered during an acceleration pulse from s_2 to s_{limit}
D_{limit}	minimum distance required to be able to use the SAP when ramping from s_1 to s_{peak} and then from s_{peak} to s_2 .
D_{min}	minimum distance required to use the SAP algorithm
D_v	distance required to ramp between the two way-point speeds
D_w, D_x, D_y, D_z	distances required to ramp between two speeds
Desirable	flag used in the high level of the trajectory planner
F_w	world frame
G	vector of gravity terms
H	unit quaternion
$J(\gamma)$	Jacobian matrix at joint configuration γ
K	constant used to normalize a quaternion
L	variable used in determining the root of an equation
M	inertia matrix of the manipulator
N	number of way-points between which a trajectory should be planned at one time (size of planning window)
N_{const}	constant used to normalize a quaternion
P	variable used in determining the root of an equation

R_{xy}	rotation matrix describing the rotation necessary to move from the orientation at point x to the orientation at point y
$S(f)$	skew symmetric matrix corresponding to vector f
\mathbf{T}	vector of actuator torques
U, V	variables used in determining the root of an equation
W	number of points preceding the last <i>Best</i> point in the window of N way-points
Y	number of points given to the parser from the window of N way-points
Z	quaternion = $[Z_w, Z_x, Z_y, Z_z]$
$a(t)$	acceleration at time t
\mathbf{a}	acceleration vector of Cartesian degrees of freedom
\mathbf{a}_{1a}	vector of accelerations at the start of a blend
\mathbf{a}_{2a}	vector of accelerations at the end of a blend
$a_{blendMax}$	maximum acceleration occurring during a blend.
a_{max}	manufacturer maximum allowable acceleration (MMAA)
a_{peak}	peak acceleration reached during a point to point motion ($\leq a_{max}$)
b_1, m_1	intercept and slope of $x_1(\sigma)$
b_2, m_2	intercept and slope of $x_2(\sigma)$
b_x	polynomial coefficients (e.g. b_0, b_1, b_2)
dD	distance remaining after an acceleration pulse (used in the speed-limited case)
ds	change in speed
ds_{min}	minimum change in speed required to use the SAP algorithm
ds_{rd}	change in speed required for a ramp from maximum acceleration to zero acceleration in the SAP algorithm
dt	time required to ramp up to a_{peak} while respecting j_{max} ($\leq dt_{max}$)
dt_{max}	time required for a ramp up to a_{max} from zero acceleration while respecting j_{max}
\mathbf{f}	axis of the axis-angle representation of a rotation matrix = $[f_x, f_y, f_z]$
$g(t)$	quintic polynomial
$\hat{i}, \hat{j}, \hat{k}$	unit vectors forming an orthogonal frame
$j(t)$	jerk at time t
$j_{blendMax}$	maximum jerk occurring during a blend
j_{max}	maximum allowable jerk
k, l, m	coefficients of a cubic equation

n	number of actuators or joints
$p(t)$	position at time t
\mathbf{p}_e	coordinates of the point at the end of a straight line segment
\mathbf{p}_i	position at point i
\mathbf{p}_i	position vector at point i (in joint or task space)
\mathbf{p}_{ia}	vector of positions at the start of a blend around point p_i
\mathbf{p}_{ib}	vector of positions at the end of a blend around point p_i
\mathbf{p}_s	coordinates of the point at the start of a straight line segment
$q(t)$	trajectory polynomial
$\dot{q}(t), \ddot{q}(t), \dddot{q}(t)$	trajectory derivatives
r_{11}, \dots, r_{33}	elements of the rotation matrix
$s(t)$	speed at time t
$s\%$	speed expressed as a percentage of the Manufacturer Maximum Allowable Speed
s_a	speed attained at the end of a ramp from zero acceleration to maximum acceleration in the SAP algorithm
s_{az}	speed attained at the end of a ramp from zero acceleration to maximum acceleration in the SAP algorithm for the case z
s_b	speed attained at the end of the acceleration cruise in the SAP algorithm
s_{bz}	speed attained at the end of the acceleration cruise in the SAP algorithm for the case z
$s_{blendMax}$	maximum speed with which a geometric blend can be started or ended while respecting acceleration and jerk limits.
$s_{blendMaxAcc}$	maximum speed with which a geometric blend can be started or ended while respecting the acceleration limit.
$s_{blendMaxJerk}$	maximum speed with which a geometric blend can be started or ended while respecting the jerk limit.
s_i	speed at point i
s_{limit}	speed reached by an acceleration pulse at a_{max}
s_{max}	maximum allowable speed along a trajectory segment (\leq MMAS)
s_{peak}	peak speed reached during a trajectory between two way-points ($\leq s_{max}$)
t	time
t_{cart}	time required to carry out the desired change in position (Cartesian space) while respecting kinematic limits
t_k	time at sampling interval k

t_{rot}	time required to carry out the desired change in orientation while respecting kinematic limits
t_s	time at which $\mathbf{x}_1(t)$ and $\mathbf{x}_2(t)$ intersect
t_{sl}	time required to carry out a given straight line motion
t_λ	rotation time limit based on respecting the rotational acceleration limit
t_φ	rotation time limit based on respecting the rotational jerk limit
t_ω	rotation time limit based on respecting the rotational speed limit
tightness	variable that indicates the size of the blend
$\hat{\mathbf{u}}$	unit vector
\mathbf{v}	velocity vector of Cartesian degrees of freedom
\mathbf{v}_{1a}	vector of velocities at the start of a blend
\mathbf{v}_{2a}	vector of velocities at the end of a blend
\mathbf{v}_d	$= \mathbf{m}_2 - \mathbf{m}_1$
$\mathbf{x}_1(t), \mathbf{x}_2(t)$	position trajectory polynomials that are connected by a blend
$\mathbf{x}_1(\sigma), \mathbf{x}_2(\sigma)$	$\mathbf{x}_1(t)$ and $\mathbf{x}_2(t)$ expressed in the σ -domain

Δt	controller sampling interval
Δt_{jerk}	time allowed to ramp up to maximum acceleration (used in calculating the jerk limits of the actuators)
$\alpha(\sigma)$	blend function that smoothly increases from 0 to 1 over the interval $\sigma \in [0,1]$
$\beta(\sigma)$	blend function used to compensate for extra acceleration
γ	vector of joint angles
$\dot{\gamma}$	vector of joint velocities
$\ddot{\gamma}$	vector of joint accelerations
$\ddot{\gamma}_{\text{max}}$	vector of joint space acceleration limits
$\dddot{\gamma}_{\text{max}}$	vector of joint space jerk limits
θ	angle between the lines described by $\mathbf{x}_1(\sigma)$ and $\mathbf{x}_2(\sigma)$
κ	acceleration compensation factor used in blending
λ_{max}	maximum allowable rotational acceleration (manufacturer limit)
ρ	blend acceleration proportionality constant

σ	normalized time coordinate used in blending, $\sigma \in [0,1]$
σ_k	σ at sampling interval k
τ	half of the total blend time
τ_u	time required to complete half blend u
ϕ	angle of the axis-angle representation of a rotation matrix
χ	snap
Φ_{\max}	maximum allowable rotational jerk (manufacturer limit)
Ψ_x, Ψ_y, Ψ_z	roll, pitch, yaw angles
ω_{\max}	maximum allowable rotational velocity (\leq manufacturer limit)

dof	degrees of freedom
DH	Denavit-Hartenberg
LSPB	Linear Segment with Parabolic Blends
MMAA	Manufacturer Maximum Allowable Acceleration
MMAS	Manufacturer Maximum Allowable Speed
ORTS	Open architecture Real-Time operating System
PCTOM	Path-Constrained Time-Optimal Motion
PID	Proportional Integral Derivative
SAP	Sustained Acceleration Pulse
SPCTOM	Smooth Path-Constrained Time-Optimal Motion

Acknowledgements

I would like to thank my supervisor, Dr. Elizabeth Croft, for her guidance and support throughout my two years here, and for always being there to answer my numerous questions. I would also like to thank all the members of the IAL for their support and friendship, and for humouring me when I felt a little stressed-out. The assistance of the faculty and staff of the Mechanical Engineering Department was greatly appreciated.

A special thanks to my parents for always encouraging me in my chosen fields of study, and for nodding and smiling at the right times when I was explaining some deep, but perhaps useless, fact to them. And finally, thanks to my brothers, for being my brothers.

I would also like to acknowledge the financial support of the Natural Sciences and Engineering Research Council of Canada.

Chapter 1

Introduction

1.1. Background

Multi-axis machines are used in a variety of applications: pick-and-place operations, welding, machining, etc. Such machines can be divided into two units: the physical mechanism composed of links and actuators, and the control system. The number of actuators present in the mechanical system depends on the number of independent machine axes, or degrees of freedom. For example, a typical articulated six degree of freedom manipulator contains six rotating actuators. A five-axis high-speed CNC machining centre would contain three linear actuators and two rotating actuators.

A motion task given to the machine must ultimately be represented as a reference signal, which is sent to the control system. The control system acts to make the machine track the reference signal by activating the appropriate actuators. If the reference signal changes too quickly, given the dynamic limitations of the machine, the tracking of the reference signal will be poor, regardless of the control system design. Computer algorithms are designed to calculate an appropriate reference signal based on the desired task path and time-related limits (such as speed and acceleration). This reference signal is the trajectory, and can be defined as a locus of points in operational or joint space on which a time-law has been specified [42]. The generation of an appropriate trajectory is the problem that is being investigated in this thesis.

The path along which the trajectory is defined can be point-to-point; namely, the machine is required to move between the two points but is not given any fixed intermediate path. This type of path is useful in manipulator pick-and-place operations. A path can also be completely specified through use of geometric functions. This type of path is commonly used in CNC machining applications or in manipulator applications when obstacles are present, or when it is necessary to ensure that the end effector follows a specific path. Herein it is assumed that the path definition is provided, and the problem of path planning is not specifically addressed.

The control of the machine motion can be divided into two parts: motion planning and motion tracking. Motion planning involves generating the path and its time law, providing the controller's reference signal. Motion tracking, on the other hand, is concerned with improving the tracking of the reference signal.

Motion planning is often done off-line, typically when the trajectory generation algorithms are computationally intensive. However, it is often desirable to generate trajectories on-line so that changes can easily be made to the machine's trajectory, increasing the system's overall robustness and adaptability. For example, a manipulator may require the ability to recompute its trajectory on-line in order to avoid an unexpected obstacle that lies along a path on which it is currently moving [3]. In an automated robot workcell, a high level scheduler feeds a series of tasks to a manipulator in terms of waypoints, approach points and stop points. The manipulator must execute the task by generating paths and trajectories to these points and then following these trajectories using a control law. Use of an on-line planner reduces manipulator setup time and downtime, since the time required to plan the trajectories is shorter.

In industrial applications, it is common to use simple PID control laws, which do not take into account the system's nonlinearities, to track a reference signal. To compensate for tracking errors introduced by the system's nonlinearities, a more complicated controller must be used [17, 18, 27]. Alternatively, it is possible to design a trajectory that takes the system's nonlinearities into account and thus provides a reference signal that can be more easily tracked by common industrial controllers [4, 14].

1.2. Industrial Motivation

Increased productivity is an important industrial consideration. When a multi-axis machine limits the task speed, decreasing the machine's overall motion time will increase productivity [4, 17]. In addition, improving the tracking accuracy of the machine is always desirable since it results in more repeatable products or operations.

Tracking a purely time-optimal trajectory with a simple controller will saturate the actuators resulting in poor tracking, vibrations in the machine and increased machine wear [5, 30, 32, 40, 48, 52]. Specialized controllers have been developed in order to provide better tracking of time-optimal trajectories. However, it is unlikely that they will be widely implemented in industry due to their complicated form. Purely time-optimal trajectories have been modified to take into account further limitations of the actuators, for example jerk or torque rate limits, thereby avoiding controller saturation and resulting in improved tracking accuracy [7, 14, 45]. Herein, trajectories that are planned with jerk or torque rate limitations are termed smooth trajectories.

There exists a need for a *smooth* trajectory generation algorithm that can easily be integrated into existing industrial systems, that is, be implemented using a typical industrial controller. Such an algorithm should

be applicable on-line, provide adequate dynamic limitations, and allow the specification of the speeds at all the way-points.

1.3. Research Objectives

The objective of this work is to provide a method of generating smooth, near time-optimal trajectories, on-line. Therefore, the trajectory generation algorithm must be computationally simple. The trajectories must take into account the physical limitations of the machine, that is, not only their torque limitations but also their torque rate limitations. This work aims to produce a trajectory that will improve the motion time and the trajectory tracking of a standard industrial machine whether it is applied to the machine through a simple PID controller or a more complex controller.

The trajectory is to be planned such that it could be used to provide a time-law for any type of path, described in any space, provided that the corresponding path jerk, acceleration and velocity limits are derived.

A further objective is to apply this technique to a robotic manipulator. To ensure the practicality of the proposed approach, guidelines were provided by a robotics manufacturer [19]. The guidelines specified that the motion generator should include:

- the ability to provide straight line, jerk-limited motion between way-points with variable speeds;
- a cornering scheme for motion around trajectory way-points, such that the motion does not stop at each way-point;
- coordinated joint motion (i.e., all the joints must start and stop at the same time);
- an overall planning algorithm that should allow trajectory planning for a window of N way-points "on the fly" considering the desired position and velocity at each way-point.

Furthermore, the trajectory generation algorithm should be applicable in both joint and task space. That is, the trajectory way-points can be defined in either task space coordinates, with straight-line motion between the way-points, or the way-points can be defined in joint space, with the motion between the way-points being planned in joint space. Finally, it is desirable that no retrograde motion occur during the trajectory.

1.4. Thesis Outline

This chapter introduces the notion of motion planning for multi-axis machines. The research objectives were presented, and an outline of the thesis is provided below. The terminology specific to this thesis and frequently used herein is defined in the Terminology section (p. xii).

Chapter 2 discusses the work that has been done in the area of motion control, and justifies the approach used in this thesis.

Chapter 3 presents the method used to generate a trajectory between two points in a one-dimensional parametrized space. The basic trajectory planning algorithm described in Chapter 3 can be adapted for use on different types of machines such as manipulators, CNC machines, cranes, etc. where a reference trajectory is used to describe the machine's task.

This work then concentrates on the implementation of the trajectory generation algorithm of Chapter 3 on a robotic manipulator, based on the guidelines for robotic motion described in Section 1.3. In Chapter 4, the issues related to blending motions between straight line segments without stopping are addressed and a solution proposed. In Chapter 5, the method derived in Chapter 3, combined with blending is extended into n-dimensions. In addition, the overall trajectory generation algorithm designed for a manipulator system is explained in Chapter 6.

Chapter 7 presents experimental results comparing an implementation of the trajectory generation algorithm developed herein to an industrial planner, using a CRS A465 manipulator. Examples of motions in both joint and task space are presented. In addition, the method is compared to a time-optimal, with torque rate limitations, motion using a SCORBOT ER-VII manipulator.

Finally, Chapter 8 presents the conclusions of this research, as well as recommendations for future work in this area.

Chapter 2

Literature Review

This chapter discusses issues relevant to trajectory generation for manipulators. Existing trajectory generation methods that produce time-optimal motion and near time-optimal motion are discussed. Both off-line and on-line trajectory generation techniques are examined since the time-optimal, *off-line* trajectory generation algorithms provide a basis against which sub-optimal *on-line* trajectory generation algorithms can be compared.

Initially minimum time trajectories are explored since they address the need to increase productivity. However, these trajectories do not take into account the fundamental torque rate limitations of the actuators and result in controller saturation and poor tracking accuracy. Methods of improving the tracking of such trajectories are then discussed and trajectories planned considering the torque rate limitations, smooth trajectories, are expounded in Section 2.3. These methods all require the off-line generation of trajectories due to their high computational load.

On-line trajectory generation is desirable since it reduces manipulator setup time and downtime. These methods consider only the kinematic limitations of the manipulator and result in near-time optimal motion. The kinematic limits that must be considered to provide smooth motion are speed limits, acceleration limits (proportional to torque) and jerk limits (proportional to torque rate). These methods are discussed in Section 2.4 and emphasis is placed on the methods that limit the kinematics during the entire trajectory, not just at the way-points. In Section 2.6 the research objectives presented in Chapter 1 are expanded and justified.

2.1. Path Definition

A manipulator task may be defined as a series of way-points, through which the manipulator should pass (unconstrained motion). However, it is often desirable to specify the path between the way-points to avoid obstacles present in the workspace, for example, or to facilitate a task such as spray-painting or welding. In robot manipulations, such tasks are generally described by straight-line motions in task space joined by polynomial splines at the way-points. The splines pass through or near the way-points so that the manipulator does not have to stop at each way-point [33, 35, 50]. The objective in this constrained motion

is to accurately track the path [43]. Trajectory planning methods for both types of path specification have been developed and will be discussed below.

In this thesis, it is assumed that the path to be followed by the manipulator can be parametrized in terms of a single variable. The path is given in terms of the actuator positions (joint-space specification) or in terms of the placement of the tool tip in the workspace (task-space specification).

2.2. Time-Optimal Trajectory Planning with Dynamic Limitations

One of the most important issues in trajectory planning for manipulators is that of increasing productivity. Machine processing or handling time can be a system bottleneck. Such machines represent high capital costs, therefore increasing the number of machines is often not economically feasible. The capability of the machine actuators limits the motion time. However, increasing the size and power of the actuators is not always desirable since this would increase the inertia of the overall system. Such a solution can be largely self-defeating [4]. Another approach is to use minimum-time trajectories.

Time-optimal control of manipulators is a difficult problem due to the manipulators' nonlinear, coupled dynamics and their joint and torque limits [28]. The first attempts at solving this problem from a trajectory planning viewpoint resulted in suboptimal control solutions. Kahn and Roth (1971) formulated the time-optimal control problem for an unconstrained path subject to torque limitations using Pontryagin's principle [26]. The resulting set of equations was computationally intractable. Thus, the dynamic model of the manipulator was linearized so that a standard linear optimal control technique could be applied to the problem, and a near time-optimal trajectory obtained. However, linearization of the dynamics results in significant errors [39].

Geering et al. (1986) developed the structure of a time-optimal motion for unconstrained paths subject to torque constraints using Pontryagin's minimum principle, and showed that such motions were either bang-bang or bang-singular-bang in torque [24]. Chen and Desrochers proved that such a minimum-time control law requires that at least one of the control torques be saturated along the entire trajectory [13]. Sahar and Hollerbach developed a general solution for determining the time-optimal path of a point-to-point motion using joint-space tessellation to create a state-space search tree, which is searched to find the best solution [41]. The algorithm takes into account the full dynamic model of the manipulator and kinematic, dynamic and geometric constraints. Rajan, on the other hand, combined the use of Pontryagin's principle and dynamic search techniques to obtain a minimum-time point-to-point trajectory [39]. A path is first determined using a joint space search technique and then the algorithm in [4] is used

to determine the minimum-time law along the path. The path is then varied and the minimum time-law along the new path found. This is done iteratively until the minimum time path and trajectory are obtained.

Bobrow et al. [4] and Shin and McKay [44] considered the problem of moving a manipulator along a specified path in minimum-time while taking the manipulator torque limits into considerations. Shin and McKay parametrized the path in joint space while Bobrow parametrized it in task space. Both use phase plane techniques to determine the trajectory switching points, which indicate where a change from maximum acceleration to maximum deceleration or vice-versa should occur, in order to obtain minimum-time motion along the path. The resulting path-constrained time-optimal motion (PCTOM) maximized the velocity along the given path. The acceleration and velocity limits are calculated directly from the robot's dynamic equations and the given path. Pfeiffer and Johann provide an efficient manner of identifying the switching points [36]. Both Shin and McKay and Bobrow et al. showed that the optimal control along a parametrized path was bang-singular-bang. Furthermore, Chen and Desrochers showed that the PCTOM solution requires that only one control torque always be in saturation on every finite interval along the time-optimal trajectory [12].

Kim and Shin developed a minimum-time path planning method in joint space based on the dynamic model of the manipulator and subject to torque constraints [28]. The path was formed of a series of straight lines with specified path deviation at the corner points. By deriving bounds on the joint space acceleration from the manipulator torque limits based on a heuristic approximation, the problem was divided into a set of one-dimensional optimization problems which could easily be solved. Their approach differed from that of Bobrow et al. and Shin and McKay in that it allowed for path deviations at corner points.

However, as discussed, time-optimal trajectories based on the dynamic model of the machine and the actuator torque limitations result in discontinuous (bang-bang) actuator torque reference profiles. Such rapid torque fluctuations cannot be obtained from the actuators because of their physical limitations [5, 14, 30, 43], and as a result, the actuators fall behind the reference signal, leading to controller saturation, which produces vibrations in the machine and tracking errors. Furthermore, the computational burden of generating such trajectories for manipulators is very high. The methods described above require, in essence, the solution of a two point boundary value problem for n degrees of freedom for a non-linear system. This is done through dynamic programming and/or other optimization/search techniques. Work

by Croft et al. [16] and Sahar and Hollerbach [41] considered strategies for reducing the computational expenses of solving this problem through heuristics and approximation techniques.

2.3. Practical Implementation of PCTOM Trajectories

PCTOM trajectories result in at least one saturated actuator at all times. In addition, the torques may be discontinuous at the switching points. Such trajectories cannot be accurately tracked since they do not take into account the uncertainties of the dynamic model, the dynamics of the tracking controller and requirements for tracking accuracy [27]. Furthermore, the actuator saturation leaves no control authority to correct the tracking errors that will occur [17, 18, 27, 43]. Tracking accuracy is of prime importance in many applications such as welding, painting, gluing, etc. Therefore, PCTOM-type trajectories are of little use in practice unless these effects can be overcome. This can be done by introducing changes at the controller level [17, 18, 27, 43, 46] or by modifying the PCTOM trajectory [14, 45]. The first approach is discussed in section 2.3.1, while the second is discussed in Section 2.3.2.

2.3.1. Controller Based Tracking of Purely Time-Optimal Motions

Tam and Potts proposed a perturbation control scheme for tracking PCTOM trajectories [46]. In this scheme, the control signal, switching time and final time are updated based on the path deviation. The method was compared to computed torque control and was found to provide faster convergence to the path and less overshoot since it accounts for actuator saturation. However, this method does not eliminate actuator saturation: it is simply postponed.

If an actuator is already at its torque limit, it is obviously not possible to catch up to the nominal trajectory. One way of improving path tracking is to design the trajectories with conservative torque limits; another is to modify the trajectory on-line [17]. Dahl and Nielsen argue that the second method is preferable, since being conservative in the trajectory planning stage is inefficient for productivity reasons [17]. That is, it is hard to ensure that the conservative torque limits chosen are not too conservative. They propose a scheme in which the primary robot controller is left unchanged, while the time-optimal reference trajectory is modified using a secondary control loop, the path velocity controller. The path velocity controller modifies the nominal trajectory to achieve a trajectory that does not require more torque than is available. The path velocity controller limits the path acceleration by modifying the nominal velocity profile at the expense of increased motion time. This method provides good tracking accuracy and good utilization of the available torque range [18]. This method allows and adjusts reference trajectories that exceed the robot tracking capabilities and increases robot performance since it is able to approach the minimum possible motion times.

Shiller et al. [43] proposed an implementation which resulted in higher tracking accuracy at high speeds than that provided by Dahl and Nielsen's work [17, 18]. It is based on using trajectory preshaping to compensate for the delay introduced by the feedback controller used to track a PCTOM trajectory, as well as a simplified motor model that models system dynamics not taken into account in trajectory generation, and is used to modify the preshaping current command. Comparison with a typical linear segments with parabolic blends (LSPB) profile showed a six-fold reduction in tracking error.

Kieffer et al. noted that all the above schemes did not allow a minimum path deviation tolerance to be specified [27]. They proposed two schemes to adapt time-optimal trajectory planning algorithms for application to manipulators under computed torque control while tracking the paths to a prescribed tolerance. In the first scheme, the controller gains are chosen experimentally while tracking a PCTOM trajectory. The required compensation torques are then calculated to meet a prescribed tracking accuracy. A PCTOM with those new torque limits is calculated and used along with the experimentally determined controller gains. This scheme resulted in the elimination of actuator saturation, tracking accuracy within the desired tolerance, and an increase in motion time of between 6 and 23%. In the second scheme, the reference trajectory is generated on-line as a feedback function of the plant and the controller states. However, offline planning of the path acceleration policy is necessary. This scheme resulted in improved motion time while maintaining equivalent tracking accuracy compared to the first.

In 1987, Leahy and Saridis suggested a number of improvements to computed-torque compensation techniques aimed at improving tracking, namely friction compensation and improved frequency response [30]. They concluded that although these methods improved tracking accuracy, the controllers still could not overcome the jerk constraints that affected tracking accuracy, and as such, a trajectory generation algorithm that incorporated the jerk limits of the system was required. Bobrow et al. recognized that the discontinuous actuator torque required by the minimum-time solution cannot be exactly provided, given the dynamic properties of the actuators [4]. They concluded that the time constants relating the input voltage to the output torques were small enough to be neglected, but others have shown this assumption to be incorrect [5, 14, 29, 45, 48]. Methods that take into account the torque rates of the actuators giving the corresponding time-optimal motion are discussed in the next section.

2.3.2. Incorporation of Torque Rate Limitations (Smooth Time-Optimal Motion)

As discussed above, while PCTOM motion yields a solution for time-optimal motion based on bang-bang control theory [48], actuator torques cannot be varied instantaneously (i.e. the torque rate must be limited). The drive system cannot produce the large value of current necessary when a large change in

torque is demanded [5]. Discontinuities in the actuator trajectory cause vibrations in the machine structure leading to tracking inaccuracies and wear on the machine [5, 7, 15, 30, 31, 52]. It is often more cost effective to use trajectories that take the torque rate limitations into account than to redesign the machine [52]. The use of torque constraints ensures efficient use of joint actuators, while torque *rate* constraints resolve the problem of bang-bang behaviour of joint torques [48].

In 1986, Shin and McKay presented a solution to the PCTOM problem incorporating torque rate constraints [45]. This solution involved the use of a dynamic programming method, which forced the authors to make assumptions on the variation of certain quantities within discretized regions [48].

Tan and Potts advocated the use of a complete discretization of the minimum-time trajectory problem rather than using discretization in the computation of the time-optimal solution based on a continuous manipulator model [48]. They argued that robot manipulators are generally controlled at discrete sampling intervals and that discrete computation methods are necessary due to the highly non-linear dynamics of the manipulator. The solution incorporated joint torque constraints, joint velocity constraints and torque rate constraints and was based on a discrete model of the manipulator dynamics. The planning of a trajectory along a path with the above constraints and 10 discrete time intervals required 2 s of CPU computation time, and a 1 min computation time was required for a case with 40 discrete time intervals.

Tarkiainen and Shiller developed a method for computing a time-optimal trajectory that was bang-bang in jerk along a specified path, except at singular points and arcs where the acceleration was at its maximum or minimum [49]. Constantinescu and Croft also presented a method for determining smooth and time-optimal motions for path-constrained trajectories (SPCTOM) [15]. The controlled input of the optimization problem is the pseudo-jerk, and cubic splines are used to parametrize the state space trajectory. It was proved experimentally that SPCTOM trajectories with appropriate torque rate limits provide a faster motion than a PCTOM or a motion described with a single quintic subject to equivalent kinematic limits.

2.4. Trajectory Generation with Kinematic Limitations

2.4.1. On-Line Trajectory Generation

Manipulator dynamics are nonlinear and coupled. Thus, the time-optimal trajectory solution (with or without torque rate constraints) is, in general, too time-consuming for on-line implementation. Instead, the trajectory planning is done off-line, providing the necessary sequence of joint angles (and their subsequent derivatives depending on the machine control scheme in use). Path tracking is then carried out

on-line [48]. An alternative to considering the full dynamic model of the manipulator is to derive the kinematic limits of the manipulator (i.e. velocity, acceleration and jerk limits) and to plan a trajectory using simple functions that stay within these limits. It is usually assumed that there are piecewise constant bounds on machine velocity and acceleration. In reality, these bounds vary with position, angular velocity and payload mass and shape because of the manipulator dynamics are nonlinear functions [4, 44]. Therefore, the constant bounds must be chosen as the reasonable global lower bounds of the velocity and acceleration in the entire workspace. This results in underutilization of the machine capabilities since the task space limits are determined experimentally and are for a ‘worst case scenario’ [28]. However, kinematic methods are well suited for on-line applications since the algorithms tend to have short calculation times and analytical solutions. Given the limits, kinematic time-optimal solutions can be developed but will never reach the full dynamic optimality. In either case, unless jerk is limited, the system (actuators and controller) is unlikely to be able to track the trajectory, making optimality a moot point.

The extra computational effort and the complex mathematical formulations that are associated with off-line methods are often not justified. A pre-planned trajectory is not exactly followed by the manipulator since there are errors in the mathematical model of the machines, due to their complexity [48]. In addition, the ability to compute a trajectory on-line increases the ease and speed with which manipulator programming can be done. It also reduces a manipulator’s downtime and provides a certain amount of autonomy to the manipulator. A manipulator may require the ability to recompute its trajectory on-line in order to avoid an unexpected obstacle lying along a path on which it is currently moving [3]. For these more practical reasons which relate to industrial applications, as well as autonomous robotics applications, this research aims to develop an on-line trajectory generation algorithm.

Trajectory generation techniques based on limiting the manipulator kinematic variables that involve optimization have been developed. For example, Cao et al. suggested methods of obtaining smooth minimum-time motion along both constrained and unconstrained paths [7, 8]. The algorithm involves initially determining smooth and time-suboptimal cubic spline joint paths through minimization of an objective function. This is followed by scaling the resulting travelling time so that velocities, accelerations and jerks at the knot points of each joint trajectory are maximized within their limitations. Due to the computational load of the optimization process, the trajectory must again be planned off-line. Since the research objectives of this thesis involve the on-line generation of trajectories, only the trajectory generation methods involving kinematic limitations that do not require optimization will be discussed further.

2.4.2. On-Line Kinematic Trajectory Planning

2.4.2.1. Pick-and-Place Operations

Trajectory planning techniques have been developed for pick-and-place operations in which the manipulator is required to move an object from one position to another and the intermediate path is not defined [2, 25, 34]. In such techniques two way-points are always provided: the initial or pick configuration and the final or place configuration, generally specified in task space coordinates and having zero velocity conditions. The way-point coordinates are usually converted to joint space coordinates to allow the trajectory to be planned in joint space in order to reduce the computational complexity of the algorithms. Angeles et al. used C^2 continuous periodic cubic spline segments in joint space to form a trajectory joining two way-points with zero velocity and acceleration conditions [2]. Given a pre-defined task time, the maximum acceleration along the trajectory was minimized. Müller-Karger et al. also planned the trajectories in joint space but used a normalized hyperbolic function that was scaled to describe the desired trajectory and provided infinitely continuous trajectories [34]. This method also allowed the definition of a third way point that was used to avoid obstacles. Gosselin and Hadj-Messaoud proposed a solution that would ensure C^2 and C^3 continuity at the initial and final points using quintic and seventh order polynomials to join the initial and final configurations [25].

Frequently, two other way-points are defined in pick-and-place operations: a lift-off point and a set-down point so as to provide acceptable departure and arrival paths [35]. Gosselin and Hadj-Messaoud also proposed a solution for this case involving the use of a seventh order polynomial to link all four way-points while ensuring zero velocity and acceleration trajectory end conditions and a continuous jerk trajectory [25]. A second solution involving a ninth order polynomial to ensure C^3 continuity at the initial and final points of the trajectory was also developed. Gosselin and Hadj-Messaoud found that the 7th and 5th order polynomials were smooth monotonic functions, and therefore, the problem of undesirable oscillations could be avoided. However, the 9th order polynomial was not always monotonic. If more way-points must be defined in order to avoid obstacles, a single function used to join all the way-points with acceptable end conditions will require higher order polynomials whose shapes are increasingly hard to predict since they depend on the placement of the intermediate way-points.

All of these methods used normalized trajectory functions that were scaled to fit the desired way-points. This is possible since the start and end positions of the trajectories are given zero velocity and acceleration conditions and a single function is used to join all the way-points.

2.4.2.2. Multiple Way-Points with Prescribed Path

Often a manipulator task is defined in terms of constrained motion: that is the manipulator end effector must follow a specific path. In this case, the pick-and-place methods described above can not be directly applied. These methods do not allow the specification of more than three to five way-points, which may not be sufficient to avoid obstacles. Specifying more than five way-points would involve the use of high order polynomials whose shapes are hard to control, perhaps resulting in retrograde motion.

Planning in joint space is convenient since it avoids the computational burden of converting the trajectory points to joint space coordinates at every controller sampling interval, that would be necessary if the trajectory were planned in task space. However, it is more intuitive to specify way-points in task space. Thus, much work has been done involving converting task space way-point and intermediate point coordinates into joint space coordinates and planning the trajectory in joint space, resulting in approximate straight-line task space motion [11, 31, 47]. Taylor derived an algorithm that generates a sufficient number of intermediate points to guarantee that the deviation of the manipulator end effector from a desired straight-line path is within a prescribed tolerance when the trajectory is generated in joint space [47]. Taylor did not consider velocity or acceleration limitations. Lin et al. proposed an on-line method of generating a joint space trajectory through any number of way-points defined in task space [31]. The trajectory is planned joint by joint and the time to move between the way-points is equal to the controller sampling time. They also compare the use of X-spline and quartic spline trajectories, and found that the X-spline functions provide better path approximation (i.e. smaller path deviation) because continuity in acceleration is not enforced. Chang et al. proposed a similar on-line trajectory planning method, involving the approximation of the desired arbitrarily smooth task space path by a series of concatenated line segments intersecting at knot points [11]. Recursive joint trajectories are then derived by interpolating these knot points using quartic splines to ensure continuity of acceleration at the knot points and allow the acceleration along the segment to be minimized. The knot points are determined such that the resulting task space path deviation is small.

Trajectory planning of a similar type has also been done in task space. Bazaz and Tondu used three cubic splines to join way-points in task space providing continuity of position, velocity and acceleration at the way-points by imposing zero acceleration end conditions [3]. They are able to provide straight-line and circular segment paths by controlling the way-point velocities. They developed normalized velocity and acceleration profiles which are then scaled to fulfill the user's requirements. This also allows the intermediate velocity and acceleration values that will occur during the motion to be predicted.

2.4.2.3. Kinematic Limitations at, and Between, Way-Points

The methods discussed above only take into account the way-point positions, velocities and accelerations. They do not specify the velocity or acceleration profiles between the way-points. Instead only the acceleration between the points is minimized or only continuity in velocity is provided and, in certain cases, acceleration or jerk. However, it is important to ensure that the kinematic limits of the machine are respected not only at the way-points but also between them so as to provide feasible motion and reduced actuator wear [5, 7, 30, 32, 52]. Bazaz and Tondu's work, described above, does allow for the prediction of the velocity and acceleration profile and therefore the motion time could be modified so as to stay within velocity and acceleration limits [3].

The shape of the position profile is also important. Overshoot of the desired positions frequently occurs when high order polynomials are used and is undesirable in most tasks [9]. For example, position overshoot in assembly operations may cause the manipulator to contact the surface of the workspace or the object that it is to pick up, or could cause uneven welds in a welding operations.

A commonly used trajectory profile is the Linear Segments with Parabolic Blends (LSPB) [42]. It has the square wave acceleration profile shown in Figure 2.1(a). Fisher et al. provide a method of calculating the minimum time necessary for a motion based on coordinated joint motion (all joints reach the way-points at the same time) and limiting the velocity and acceleration of the LSPB motion [23]. In accordance with [4, 44], they found that for a minimum-time motion one joint is at its acceleration limit and another joint at its limiting velocity for all end effector motions. This is the time-optimal motion that could be obtained if actuators were able to change accelerations instantaneously. However, applying an LSPB profile to an actuator may introduce transients into the motion and excite structural vibrations, affecting path tracking and increasing actuator wear [9, 40].

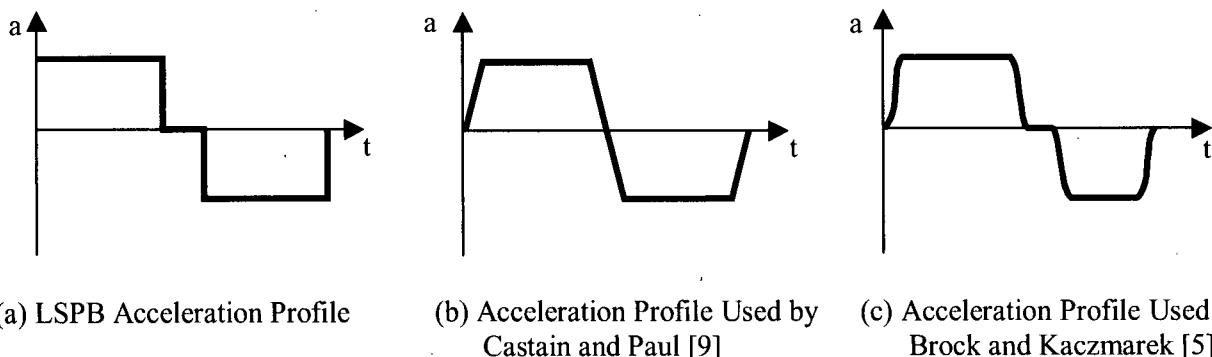


Figure 2.1. Proposed Acceleration Profiles

Castain and Paul modified the LSPB profile by limiting the rise time of the waveform (Figure 2.1(b)). That is, rather than allow the profile to change acceleration instantaneously, changes in acceleration follow a ramp profile (i.e. a trapezoidal acceleration profile) [9]. The rise time was assigned to be a fixed proportion of the time necessary to move between the two points. They found that this profile, the cubic-quadratic-cubic-squared profile, caused a 6.9% increase in motion time. All motions between two waypoints are described with a set of six splined cubic equations. The position and velocity at the two waypoints are specified. The trajectory is planned based on the initial and final point as well as an intermediate point determined by the algorithm. The accelerations at these three points are set to zero. The algorithm ensures that the acceleration limit is respected along the entire trajectory by calculating the planned accelerations and comparing them to the limit. If the acceleration limit is violated, the time is incremented and new accelerations are calculated. The algorithm can calculate a trajectory between any number of way-points and assumes that the first and last way-points provided have zero speed conditions. The resulting trajectories provided smooth motion and no overshoot or undershoot of the way-points. The main disadvantage of this method is that it requires iterative calculations to ensure that the acceleration limits are respected and does not take velocity limits into account. The algorithm assumes that the user provides the motion time or it uses a default time.

Erkorkmaz also used a trapezoidal acceleration profile to provide a smooth motion that stayed within jerk, acceleration and velocity limits [20]. However, in contrast with Castain and Paul's method [9], the profile could be formed of up to seven segments (a zero acceleration segment was included). The method requires solving a nonlinear system of equations using Newton-Raphson's algorithm to obtain the segment times. The reference trajectory is generated with varying interpolation periods and therefore requires re-sampling at the control frequency. This re-sampling was done using quintic polynomials, with boundary conditions estimated using third order polynomials, to ensure the C^2 continuity of the trajectory.

Brock and Kaczmarek reported that use of an LSPB profile for CNC machining caused contouring errors due to the system's inability to provide the required torque at the points where instantaneous changes in acceleration were required [5]. They suggested the use of a squared sine function to provide continuity of acceleration (Figure 2.1(c)). The squared sine function was used to ramp the acceleration whenever a change in acceleration was required. A typical reference profile was composed of seven segments: three constant acceleration segments and four squared sine function segments. The profile provided a sub-optimal motion that they applied to curvilinear paths while staying within acceleration and velocity limits. The jerk limitation is provided by adjusting the acceleration ramp time.

2.5. Summary of Trajectory Planning Techniques

In summary, trajectory planning techniques have been developed to improve productivity and tracking accuracy. Trajectories that are purely time-optimal [4, 44] and based on the machine dynamics do not take into account actuator torque rate limitations and lead to controller saturation when implemented. This results in poor tracking accuracy and machine wear. Controller-based and planning-based solutions to this problem have been proposed [8, 14, 17, 27]. However, it is unlikely that the controller-based methods will be implemented in an industrial setting since they require complex control laws. In addition, trajectory planning methods that incorporate the machine dynamics cannot be used on-line since they represent a high computational load.

Trajectory planning techniques based solely on the kinematic limits of the actuators have also been developed and can be implemented on-line. These methods are time suboptimal since the kinematic limits are determined experimentally and are for a “worst case scenario” [28]. Of most interest to the work described herein is the subset of kinematic methods that limits the acceleration, velocity and jerk during the entire trajectory between two way-points [5, 9, 20].

2.6. Problem Formulation

The objective of this research is to develop a method of generating *smooth* multi-axis machine trajectories, suitable for on-line applications. That is, trajectories that stay within velocity, acceleration and jerk limitations throughout the motion. The trajectories should be near time-optimal so as to increase the productivity of the machines. Acceleration and jerk limitations are considered in order to reduce wear and tear on the actuators, prolong machine life and reduce servicing. Tracking accuracy can also be improved by using smooth trajectories since the controller saturation is avoided, as was discussed in the preceding sections.

The algorithm should be computationally simple for on-line implementation. Thus, in this work, the proposed algorithm is based on providing kinematic limitations to the motion (i.e. limiting the jerk, acceleration and velocity of the motion), rather than directly incorporating the full dynamic model of the machine. Although this results in underutilization of the machine’s capabilities [4, 28, 41, 44], incorporating the dynamics of the machine would severely complicate the algorithm due to the inherent non-linearities, and prohibit use of the algorithm on-line. Further developments on this research could lead to the incorporation of linearized dynamics in the planning algorithms. However, this is not within the scope of this research.

The basic trajectory planning algorithm should provide a time law that can be used on any parametrizable path, described in any space, provided that the corresponding jerk, acceleration and velocity limits are derived. Thus, such a trajectory could be used on different types of machines such as manipulators, CNC machines, cranes, etc., where a reference trajectory is used to describe the machine's task.

This work is closest to that by Brock and Kaczmarek [5], Castain and Paul [9] and Erkorkmaz [20], described in Section 2.4.2.3. It ensures that the velocity, acceleration and jerk limits are respected along the entire trajectory and not just at the way-points (the entire profile must be defined). The trajectory must be C^2 continuous and a desired speed, which should be attained if physically possible, will be given to each way-point. Most methods described in literature do not allow for the specification of all these properties.

As discussed previously, Erkorkmaz limited the jerk, acceleration and velocity of the motion. However, the method requires optimization to determine the motion time, and cannot be implemented on-line. While Castain and Paul limited velocity and acceleration, they did not incorporate jerk limitations. The method proposed by Brock and Kaczmarek is closest to the requirements of this work since it is possible to choose the time parameters in such a manner that the jerk limits are respected. However, the use of sine functions leads to computational complexity that should be avoided if the algorithm is to be used on-line. The algorithm developed herein was developed independently of Brock's work, but the results are quite similar. Furthermore, Brock and Kaczmarek only discuss the case where the maximum allowable velocity between two way-points is greater than the velocity at the start and the end way-points. As well, their method is applied only in two dimensions but can be used along cubic spline paths. The work presented herein also considers the case where the maximum way-point speed is the limiting path speed, as well as cases involving speed and distance limitations (more typical of manipulator trajectory planning than the CNC trajectory planning proposed by Brock and Kaczmarek). This work is applicable to motion in any dimensional space, addresses the issues related to joint space and task space planning, and provides a method for transiting past way-points with a specified approach distance.

Another important consideration is that the resulting trajectory should not contain any oscillations in the position domain since undershoot or overshoot of way-points is often undesirable. For example, in the case of a manipulator, such oscillations may cause the manipulator to contact with an obstacle in the workspace. The manipulator may end up pushing a workpiece into the supporting surface or away from the manipulator causing damage to the manipulator or workpiece and/or failure of the task. This type of

behaviour can be hard to control when polynomial trajectories are used to describe the trajectory [9]. Thus, the trajectory must be formed in such a manner that no such oscillations occur.

2.6.1. Implementation

The second objective of this work is to implement the trajectory generation method for use with robotic manipulators based on guidelines provided by an industrial robot manufacturer [19]. The various issues related to this implementation are discussed below.

2.6.1.1. Task Space Planning Versus Joint Space Planning

In most cases, a manipulator path will be described in terms of task space coordinates. For applications such as gluing, arc welding, spraying or cutting, the specified path must be followed accurately. Therefore, it is necessary to plan the entire trajectory in task space rather than to approximate the path by interpolating knot points in joint space. Motions planned in task space also provide a motion that is predictable: that is, it is easy for the user to tell whether the manipulator end effector will contact an obstacle [47]. A path in joint space is not intuitively predictable. In addition, for certain applications such as carrying trays filled with fluids, keeping the end effector orientation constant during the motion may be necessary. This can easily be specified in task space, whereas a joint space motion does not provide such guarantees. Thus, the added computational burden of performing the inverse kinematics at every control interval is justifiable.

However, the value of the ability to plan a task in joint space should not be overlooked. It has been shown that the fastest motion between two way-points approximates a straight line in joint space [41]. Therefore, in areas of the workspace where no obstacles are present, it is desirable to move as fast as possible, and a straight-line joint-space path is the best option for on-line computation. In addition, motions planned in joint space will reach the joint space way-points with better accuracy since errors will not be introduced by the forward and inverse kinematics. Finally, following a straight-line path described in joint space will provide coordinated joint motion. Coordinated joint motion provides smooth well-behaved end effector motion that is path invariant with manipulator speed [23]. This provides predictable motion in systems where a path is taught to the manipulator at a low speed and the manipulator is to carry out the tasks at high speed. When the motion is run at high speeds, the robot will follow the same path it followed at the lower speed.

Therefore, a user can use joint space motions for gross motions when the manipulator is not carrying anything, and is simply transiting between two tasks (for example two pick-and-place operations). In

areas where many obstacles are present, or when the manipulator is carrying something, straight-line motion in task space can be used to ensure that no obstacles are contacted and/or that the object the manipulator is carrying is not damaged. The ability to plan in either joint or task space is provided in this work.

2.6.1.2. Path-Related Constraints

A straight-line motion provides a motion that is easily predictable from the user's point of view, simplifying task planning. However, if straight lines are used to connect the way-points, the manipulator must stop at each way-point due to the inability of its actuators to provide discontinuous velocity. Thus, it is common to connect the straight lines with curves that pass close to the way-point, but not necessarily through the way-point [33, 35, 50]. An industrial standard [19], followed in this work, is to blend within the angle defined by the two straight lines. This standard is followed in both task and joint space. It is necessary to ensure that the kinematic limits are respected during the curved portion of the motion.

Straight-line motion can be provided by superimposing the trajectory profile on the parametrized straight line described in either joint space or task space. However, task space planning possesses the additional complexity of planning the end effector orientation trajectory. Orientation cannot be treated as a vector, and smoothed Euler angles do not necessarily provide a smooth motion [22]. Thus, a method of providing smooth orientation interpolation between the way-points is presented in Chapter 5.

2.6.1.3. High Level Planner

As defined by Bazaz and Tondu, on-line trajectory generation is the computation of the next trajectory segment while the manipulator is executing the present trajectory segment [3]. An overall planning algorithm is needed to coordinate the on-line trajectory planning for multiple way-points. The high level planner obtains future way-points from the scheduler, and then calls lower level functions to calculate trajectories based on the type of motion the user requires (e.g. straight line, straight line followed by a blend, etc.). It is desirable that over a planned window of N way-points, the user-specified speeds (expressed as a percentage of maximum speed) be attained at each way-point, if physically feasible. Furthermore, the trajectory must always have a stop condition at the end of the planning window in case no new way-points are provided. If new way-points are provided, the imposed stop condition can be removed. In this manner, it is not necessary to stop at every way-point, and a significant speed can be reached along an extended trajectory. The choice of the number of points in the window will require a trade-off between the overall motion time and the amount of time the robot will delay (for planning) until

it starts the motion. As well, the larger the planning window, the greater the buffer available to handle trajectories with a large number of closely spaced way-points with high speed demands.

2.7. Summary

Purely time-optimal trajectories require bang-bang torque profiles, which are unattainable by the actuators due to physical limitations. Such trajectories result in controller saturation which reduces tracking performance and induces vibrations in the machine [14].

Trajectory generation algorithms, which take the torque rate limitations into account and use the dynamic model of the manipulator, provide smooth motion that the actuators can track more accurately and reduce wear in the machines. However, such algorithms rely on optimization of non-linear multiple degree of freedom equations to calculate the trajectory and are not practical for on-line industrial manipulator trajectory planning. An alternative is to consider only the kinematic limits of the actuators obtained from the dynamic models of the machines (velocity, acceleration and jerk) and to plan the trajectory based on these limits. The constant bounds on these variables are chosen as the effective lower bounds in the entire workspace. This results in an underutilization of the machine capabilities since the task space limits are determined experimentally and are for a “worst case scenario” [28]. However, such methods can be successfully implemented on-line. Existing methods in the literature that limit the jerk during the *entire* trajectory (i.e., smooth trajectories) generally require iteration, optimization or frequent calculation of transcendentals and are not suitable for on-line implementation.

The present research provides a method of generating near time-optimal smooth trajectories that can be applied on-line. To the author’s knowledge, this is the first-reported industrially-implemented, on-line, jerk-limited motion planner. The trajectories can be planned in task or joint space and are composed of straight-line segments connected with blends. Furthermore, an overall planning algorithm used to generate trajectories between multiple way-points was developed. This work has been implemented on a robotic manipulator, and experimental results demonstrating the utility of the approach will be presented. It will also be shown that the on-line method developed herein compares favourably to a time-optimal with torque rate limitations method, and provides a substantial improvement over a typical on-line motion planning approach.

Chapter 3

Methodology

In this chapter, the methodology used to construct an efficient, jerk limited, parametrized, one-dimensional trajectory between two way-points is outlined. The trajectory can then be extended to any parametrizable path, as will be shown in Chapter 5. The method has low computational complexity and, therefore, can be applied on-line.

As outlined in Chapter 2, an important step towards fulfilling the objectives of this research is the generation of trajectories for which speed, acceleration and jerk are limited while travelling between two specified way-points in either cartesian or joint space. The lowest order polynomial function which respects these kinematic limits at every trajectory way-point is a seventh-order polynomial function. Such a function would allow the specification of eight end conditions (i.e. initial and final: position, speed, acceleration and jerk). However, simply connecting any two way-points with such a polynomial does not guarantee that the kinematic limits are respected at all points between the two way-points. For example, the polynomial may cause retrograde motion or overshoot of the speed, acceleration and/or jerk limits [9, 10, 37]. It will be shown in this chapter, that by using a concatenation of fifth-order polynomials and a sine wave template, it is possible to construct an efficient trajectory between two way-points that respects all the required kinematic limits.

Section 3.1 provides the basis function used when a change in acceleration is required, and provides two different algorithms used to obtain changes in speed along a trajectory: the sustained acceleration pulse algorithm and the acceleration pulse algorithm. These algorithms are then used in Section 3.2 to generate trajectories between any two one-dimensionalized way-points; that is, the trajectory is parametrized along a straight line between any two multi-dimensional way-points.

3.1. Trajectory Functions (Building Blocks)

Linear segment with parabolic blends (LSPB) trajectories are commonly used by industrial manipulators and machine tool trajectories for their fast motion and low computational complexity. LSPB trajectories switch between maximum (a_{max}), minimum and zero accelerations to achieve time-optimal motion while respecting imposed speed limits (s_{max}) (Figure 3.1). However, this type of motion has infinite jerk, and as

a consequence, is not easily tracked, resulting in a longer motion time than expected and deviations from the prescribed path. If LSPB motion could be approximated using a series of smooth functions that are at least C^2 continuous, that is, continuous in their second derivative, the resulting motion would be fast and trackable.

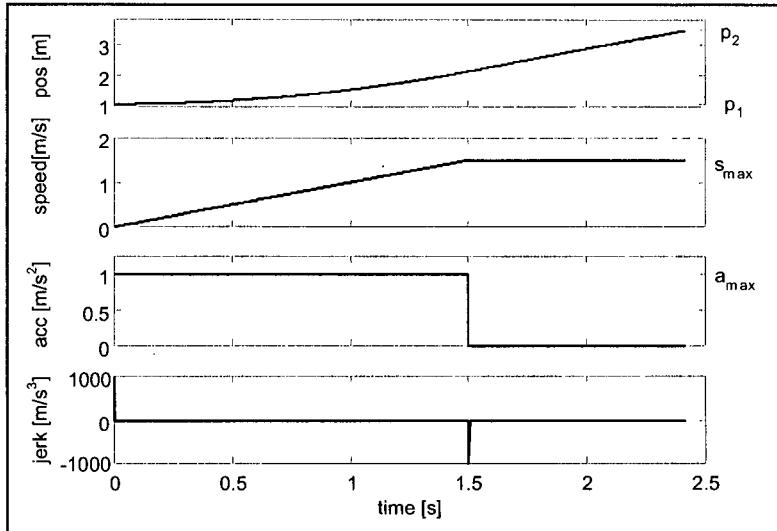


Figure 3.1. Linear Segment with Parabolic Blend (LSPB) Trajectory Between Way-Points p_1 and p_2 ^{*}

Fifth-order polynomials (quintics) are the lowest order polynomials for which it is possible to specify the end point positions, speeds and accelerations (Eq. 3.1),

$$g(t) = b_5 t^5 + b_4 t^4 + b_3 t^3 + b_2 t^2 + b_1 t + b_0, \quad (3.1)$$

where t is the time. This polynomial provides a quadratic jerk profile, as opposed to a cubic polynomial, which provides a constant jerk profile. At a minimum, a cubic trajectory will result in jerk discontinuities, and potentially, infinite jerk, at trajectory way-points.

Andersson used quintics for stop-point-to-stop-point motion, and derived the quintic motion time such that speed and acceleration limits are respected [1]. Using this method, it is also possible to ensure that jerk limits are respected. However, extending this work to cases where the initial and/or final speeds are non-zero would require the computational expense of repeatedly solving a third-order equation. Furthermore, the resulting trajectory will be slower than the one proposed herein and may have a number of undesirable oscillations in position, velocity and acceleration between the way-points.

* The maximum occurring jerk of 1000 m/s³ is a result of the sampling time interval of 1 ms used to generate the plots. The maximum jerk seen by the system will be a function of the controller sampling interval.

By assigning appropriate start and end conditions to the quintics, a smooth trajectory joining two specified way-points can be found, while satisfying acceleration continuity (limited jerk) at the trajectory end points. That is, a trajectory between two way-points is formed as a series of linked quintics, with each quintic having initial conditions equal to the end conditions of the previous quintic. Such a trajectory would have C^2 continuity. The intermediate points between the linked quintics are called “quintic control points”. In fact, the entire trajectory can be characterized by a series of these control points. Therefore, during the trajectory planning stage, it is only necessary to calculate the quintic control points corresponding to the trajectory. This provides a computationally simple and rapid method of planning the trajectories between a series of way-points, as described in Chapters 5 and 6.

Between the way-points, it is desirable to have the quintics follow a LSPB trajectory template without undesirable oscillations. This requires careful selection of the quintic control points such that each quintic will behave in a prescribed manner, respecting speed, acceleration and jerk limits. The LSPB pattern is used to provide the “switching” points for the trajectory. That is, the trajectory is still formed of a series of speed ramp and speed cruise segments, like the LSPB in Figure 3.1. However, in this work, changes in acceleration are generated using sine-shaped ramps, rather than the abrupt square wave signals that generate infinite jerk.

The sine wave template provides a constant relationship between its maximum amplitude and the maximum value of its derivative. Selecting the time for acceleration and deceleration ramps based on a sine wave with appropriate properties, allows the jerk limits to be respected. Furthermore, the sine shape is infinitely differentiable and provides a highly smooth trajectory profiles. The resulting acceleration/deceleration ramp is, therefore, smooth and there is no positional oscillation.

3.1.1. Sine wave template

As stated above, a change in acceleration is accomplished by following a sine wave template (Figure 3.2). A single quintic is used to approximate this motion. As will be shown, using a quintic to approximate the shape allows fast computation of the desired trajectory by avoiding both computation of transcendentals, and trajectory scaling. Since the end conditions are specified (initial and final: position, velocity and acceleration), the time allotted for the change in acceleration is the only free variable. This variable is used to ensure that the jerk limit is not violated.

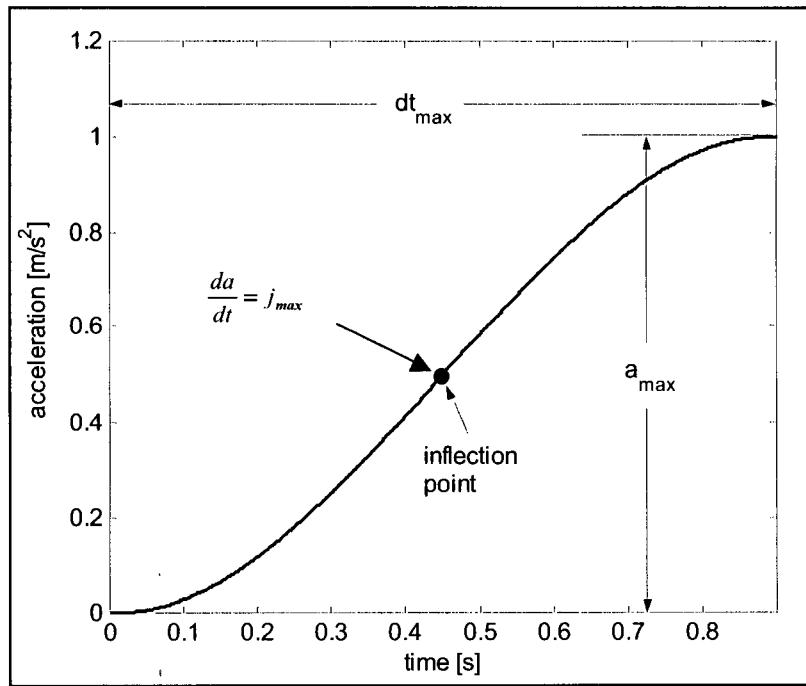


Figure 3.2. Sine Wave Template for a Positive Change in Acceleration Between Zero Acceleration and the Acceleration Limit, a_{max}

By setting the amplitude of the sine wave equal to half the acceleration limit, $a_{max}/2$, and the maximum slope equal to j_{max} , the maximum jerk, the minimum time that should be allotted for the acceleration ramp motion can be derived.

The acceleration equation for the sine wave motion can be written as:

$$a(t) = \frac{a_{max}}{2} \sin\left(\frac{\pi}{dt_{max}} t - \frac{\pi}{2}\right) + \frac{a_{max}}{2}, \quad (3.2)$$

where t is the time and dt_{max} is the total time allowed for the motion. Taking the derivative of Eq. 3.2 with respect to time gives the equation for the jerk,

$$j(t) = \frac{\pi a_{max}}{2dt_{max}} \cos\left(\frac{\pi}{dt_{max}} t - \frac{\pi}{2}\right). \quad (3.3)$$

The maximum slope of the acceleration sine wave profile occurs at $t = 0.5dt_{max}$. Setting Eq. 3.3 equal to j_{max} at $t = 0.5dt_{max}$ provides the equation for dt_{max} ,

$$dt_{max} = \frac{\pi a_{max}}{2 j_{max}}. \quad (3.4)$$

Integrating Eq. 3.2 provides the quintic speed and position end conditions corresponding to the sine wave template in Figure 3.2. The speed end conditions for the quintic are

$$s(0) = s_1, \quad (3.5)$$

$$s(dt_{max}) = \frac{a_{max} dt_{max}}{2} + s_1, \quad (3.6)$$

where s_1 is the initial velocity. The position end conditions are

$$p(0) = p_1, \quad (3.7)$$

$$p(dt_{max}) = a_{max} dt_{max}^2 \left(\frac{1}{4} - \frac{1}{\pi^2} \right) + s_1 dt_{max} + p_1, \quad (3.8)$$

where p_1 is the initial position. The acceleration end conditions are already specified as

$$a(0) = 0, \quad (3.9)$$

$$a(dt_{max}) = a_{max}. \quad (3.10)$$

A deceleration profile from can be derived in the same manner.

When a quintic with the end conditions developed above is used, the acceleration is limited to the interval $[-a_{max}, +a_{max}]$ and the jerk to the interval $[-0.98 j_{max}, +0.98 j_{max}]$. The motion also has no substantial retrograde or oscillatory effects in the position and velocity domains.

The jerk cannot be specified at the way-points, and therefore, the jerk profile is limited and has a limited discontinuity. It can be shown that by using the sine-wave template, the magnitude of this discontinuity is at most 11% of the jerk limit (Appendix A).

3.1.2. Non-Limited Trajectory

In Figure 3.3, the construction of a smooth trajectory between two way-points (p_1, s_1) and (p_2, s_2) , using a concatenation of quintics and the sine wave approximation is illustrated. In this example, the change in position required to accelerate from speed s_1 to speed s_2 is smaller than the distance between the two way-points. Thus in this case, the trajectory reaches the maximum speed, s_2 . This example represents the simplest case for a transition between two way-points.

In Figure 3.3, p_1 and p_2 are, respectively, the initial and final positions, and s_1 and s_2 are the initial and final speeds. The speed reached at the end of the ramp from zero acceleration to the maximum acceleration, a_{max} , is s_a . The time taken to ramp up to a_{max} is dt_{max} . The speed reached at the end of the

acceleration cruise at a_{max} is s_b . The maximum allowable jerk is j_{max} . The acceleration ramp up to a_{max} is followed by an acceleration cruise and then an acceleration ramp down to zero. The ramp down begins when the remaining required change in speed is equal to ds_{rd} , where,

$$ds_{rd} = \frac{a_{max} dt_{max}}{2}. \quad (3.11)$$

The acceleration ramp up and ramp down are symmetrical, simplifying the integration process. The speed ramp is also symmetrical about a straight line drawn from the start of the ramp to the end of the ramp.

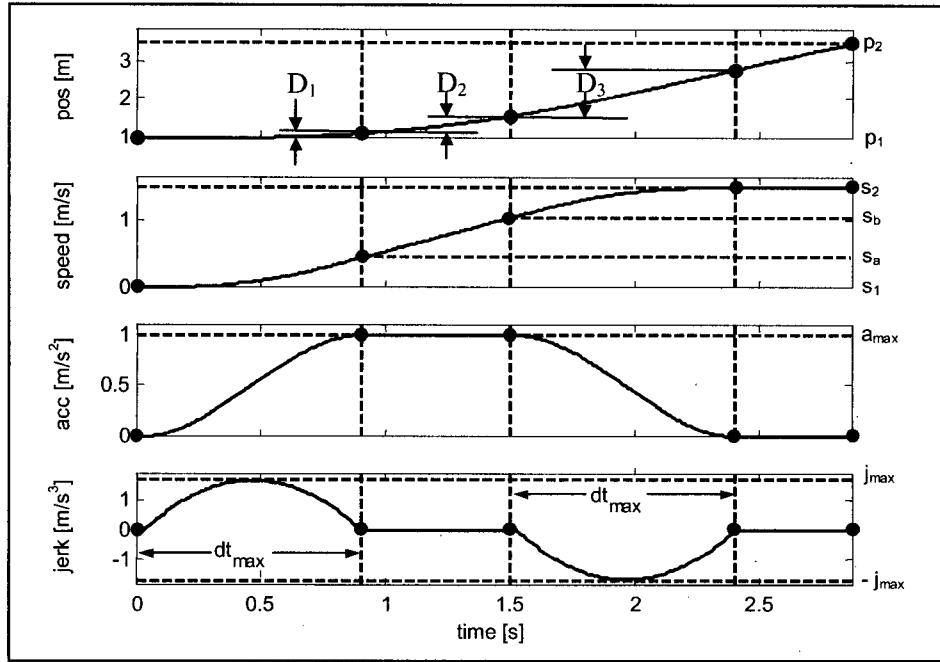


Figure 3.3. Smooth Trajectory Generated with a Sequence of Quintics (• indicate the quintic control points)

The corresponding LSPB trajectory time (Figure 3.1) for the same speed and position change is smaller than the quintic trajectory time (by dt_{max}), however, the quintic trajectory has bounded jerk, and is therefore trackable.

The quintic control points corresponding to the speed ramp are determined as specified in the Sustained Acceleration Pulse (SAP) Algorithm below. A deceleration trajectory can be derived in the same manner by reversing the end point speeds, planning the corresponding trajectory and then making the appropriate changes to the control points (Section 3.2).

Sustained Acceleration Pulse Algorithm (cf. Figure 3.3) ($s_2 > s_1 \geq 0$)

1. Calculate the total way-point to way-point distance, $D = \|p_2 - p_1\|$.
2. Calculate the two unknown control point speeds s_a ,

$$s_a = s_1 + \frac{a_{max} dt_{max}}{2}, \quad (3.12)$$

and s_b ,

$$s_b = s_2 - \frac{a_{max} dt_{max}}{2}. \quad (3.13)$$

3. Calculate the distances covered during:

- the ramp up to a_{max} ,

$$D_1 = a_{max} dt_{max}^2 \left(\frac{1}{4} - \frac{1}{\pi^2} \right) + s_1 dt_{max}. \quad (3.14)$$

- the acceleration cruise,

$$D_2 = \frac{s_b^2 - s_a^2}{2a_{max}}. \quad (3.15)$$

- the ramp down from a_{max} to zero acceleration

$$D_3 = a_{max} dt_{max}^2 \left(\frac{1}{4} + \frac{1}{\pi^2} \right) + s_b dt_{max}. \quad (3.16)$$

4. IF ($D_1 + D_2 + D_3 - D > 0$)

It is not possible to reach the demanded speed, s_2 , due to the distance, acceleration and jerk limitations. Therefore, reduce s_b and s_2 so that the distance constraint is respected (i.e., reduce the length of the acceleration cruise):

$$s_b = -a_{max} dt_{max} + \sqrt{(a_{max} dt_{max})^2 - 2a_{max} \left(\frac{a_{max} dt_{max}^2}{2} + s_1 dt_{max} - \frac{s_a^2}{2a_{max}} - D \right)}, \quad (3.17)$$

$$s_2 = s_b + \frac{a_{max} dt_{max}}{2}. \quad (3.18)$$

Recalculate D_2 and D_3 from Eqs. 3.15 and 3.16.

5. Assign the times, positions (using D_1 , D_2 , and D_3), speeds (s_1 , s_a , s_b and s_2) and accelerations (zero or a_{max}) corresponding to the four quintic control points that form the non-zero acceleration portion of the motion (refer to Section B.1, Appendix B).

To generate the entire trajectory depicted in Figure 3.3, the first four quintic control points are calculated based on the SAP Algorithm. A fifth quintic control point corresponding to the end of the speed cruise is

also necessary. In Section B.1, Appendix B, the control points conditions for the entire way-point to way-point trajectory can be found.

3.1.3. Limited Trajectories

For a short travel distance, D , between the two way-points p_1 and p_2 , an acceleration pulse of a_{max} may result in an overshoot of the distance D . This is the case when

$$D < D_{min}, \quad (3.19)$$

where

$$D_{min} = a_{max} dt_{max}^2 + 2s_1 dt_{max} \quad (3.20)$$

Again s_2 is assumed to be larger than s_1 . Furthermore, if the desired change in speed, $ds = (s_2 - s_1)$ is such that

$$ds < ds_{min}, \quad (3.21)$$

where

$$ds_{min} = dt_{max} a_{max}, \quad (3.22)$$

then an acceleration pulse of a_{max} will overshoot the desired speed change, ds .

In both cases, the maximum allowable acceleration, a_{max} , cannot be reached without overshooting a prescribed trajectory limit (D or ds). In such cases, the SAP algorithm described in Section 3.1.2 must be modified. The motion is now based on an acceleration ramp up to a lower acceleration, a_{peak} , and then an immediate ramp down to zero acceleration, i.e. an acceleration pulse of a_{peak} . If the motion is limited by the change in speed, the acceleration pulse may be followed by a speed cruise.

If $D < D_{min}$ and $ds > ds_{min}$, then only the travel distance limits the motion. In this case, the equation describing the distance, D , based on an acceleration pulse, can be found through integration of the jerk, acceleration and velocity profiles:

$$D = \frac{2j_{max}}{\pi} dt^3 + 2s_1 dt, \quad (3.23)$$

where dt is the time necessary to ramp up to a_{peak} .

In this case, dt can be calculated by solving Eq. 3.23:

$$dt^3 + \frac{\pi s_1}{j_{max}} dt - \frac{\pi D}{2j_{max}} = 0. \quad (3.24)$$

For $s_1 > 0$, Eq. 3.24 has only one positive real root in dt , which can be found directly (Appendix C) [38] or via a Taylor series approximation in an average of 1.5 iterations.

The peak acceleration, a_{peak} , is then

$$a_{peak} = \frac{2j_{max} dt}{\pi}. \quad (3.25)$$

If $D > D_{min}$ and $ds < ds_{min}$, then only the desired change in speed is limiting. In this case, the time necessary to ramp up to a_{peak} can be calculated from:

$$dt = \sqrt{\frac{\pi(s_2 - s_1)}{2j_{max}}}. \quad (3.26)$$

In the case of a motion limited by distance or speed, up to three quintics are necessary to describe the motion between two way-points: the acceleration ramp up quintic, the acceleration ramp down quintic, and, in the case of limitation by speed, the speed cruise quintic. A sample trajectory for the speed limited case is shown in Figure 3.4. The algorithm used to determine the quintic control points for the speed or distance limited case is given below. It can be shown that this algorithm converges to the Sustained Acceleration Pulse Algorithm at $D = D_{min}$.

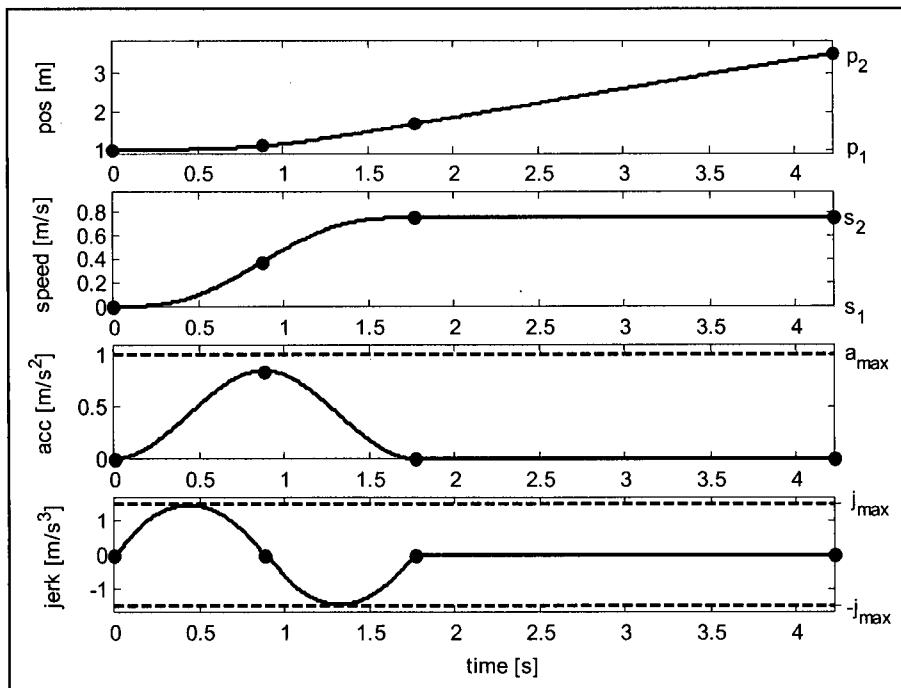


Figure 3.4. Speed Limited Trajectory (• indicate the quintic control points)

Acceleration Pulse Algorithm ($s_2 > s_1 \geq 0$)

1. IF ($ds < ds_{min}$ & $D < D_{min}$)

Calculate dt from Eq. 3.24, and a_{peak} from Eq. 3.25.

IF ($ds < a_{peak} dt$)

Recalculate dt from Eq. 3.26 and a_{peak} from Eq. 3.25.

ELSE IF($D < D_{min}$)

Calculate dt from Eq. 3.24, and a_{peak} from Eq. 3.25.

ELSE IF($ds < ds_{min}$)

Calculate dt from Eq. 3.26 and a_{peak} from Eq. 3.25.

2. Assign the time, position, speed and acceleration corresponding to the three quintic control points that form the acceleration pulse portion of the trajectory.

To generate the entire trajectory depicted in Figure 3.4, the first three quintic control points are calculated based on the Acceleration Pulse Algorithm. A fourth quintic control point (corresponding to the end of the speed cruise phase) is assigned based on the remaining distance:

$$dD = D - (s_2 + s_1)dt. \quad (3.27)$$

The quintic control point conditions for this case can be found in Section B.2, Appendix B.

3.2. General Trajectory Generation Between Any Two Way-Points

The algorithms described in Sections 3.1.2 and 3.1.3 can be used to determine a trajectory between any two way-points. Each way-point has a specified position and speed, while the acceleration at each of these way-points is set to zero to ensure continuity in acceleration and reduce the computational burden.

Herein, these trajectories are used along straight-line paths:

$$\mathbf{p}(t) = \mathbf{p}_1 + \frac{(\mathbf{p}_2 - \mathbf{p}_1)}{\|\mathbf{p}_2 - \mathbf{p}_1\|} q(t), \quad (3.28)$$

where \mathbf{p}_1 and \mathbf{p}_2 are respectively the positions of way-point (1) and way-point (2) and $q(t)$ is the trajectory generated using the control points described in this section. $q(t)$ starts at zero and ends at $\|\mathbf{p}_2 - \mathbf{p}_1\|$. Thus, a trajectory linking any two given way-points is calculated based on the distance between the points, the speeds that must be achieved at each of the way-points, as well as the acceleration and jerk limits. It should be noted that \mathbf{p}_1 and \mathbf{p}_2 may not be the scheduler-defined way-points, but rather the points situated at the end and the start of a blend, as described in Chapter 5.

In the following cases, D is the distance between the two way-points, point (1) and point (2); s_1, s_2 are the desired speeds at points (1) and (2) respectively; and the speed limit, s_{max} , is the highest allowable speed when travelling between points (1) and (2). Thus $s_{max} \geq (s_1, s_2)$. The maximum speed reached during the trajectory is s_{peak} .

If $s_2 < s_1$, the speeds are interchanged such that all motions can be treated as speed increases from s_1 to s_2 . The appropriate modifications are then made at the end of the trajectory calculations (Section B.3, Appendix B). This allows all motions to be separated into two categories based on speed. In category A, where $s_2 = s_{max}$, the maximum path speed is limited by the user-defined limit, s_2 . This category also includes motions that are limited by distance or speed. In category B, where $s_2 < s_{max}$, the maximum path speed is greater than that allowed at point (2). Figure 3.5 is the flowchart corresponding to the general algorithm outlined below.

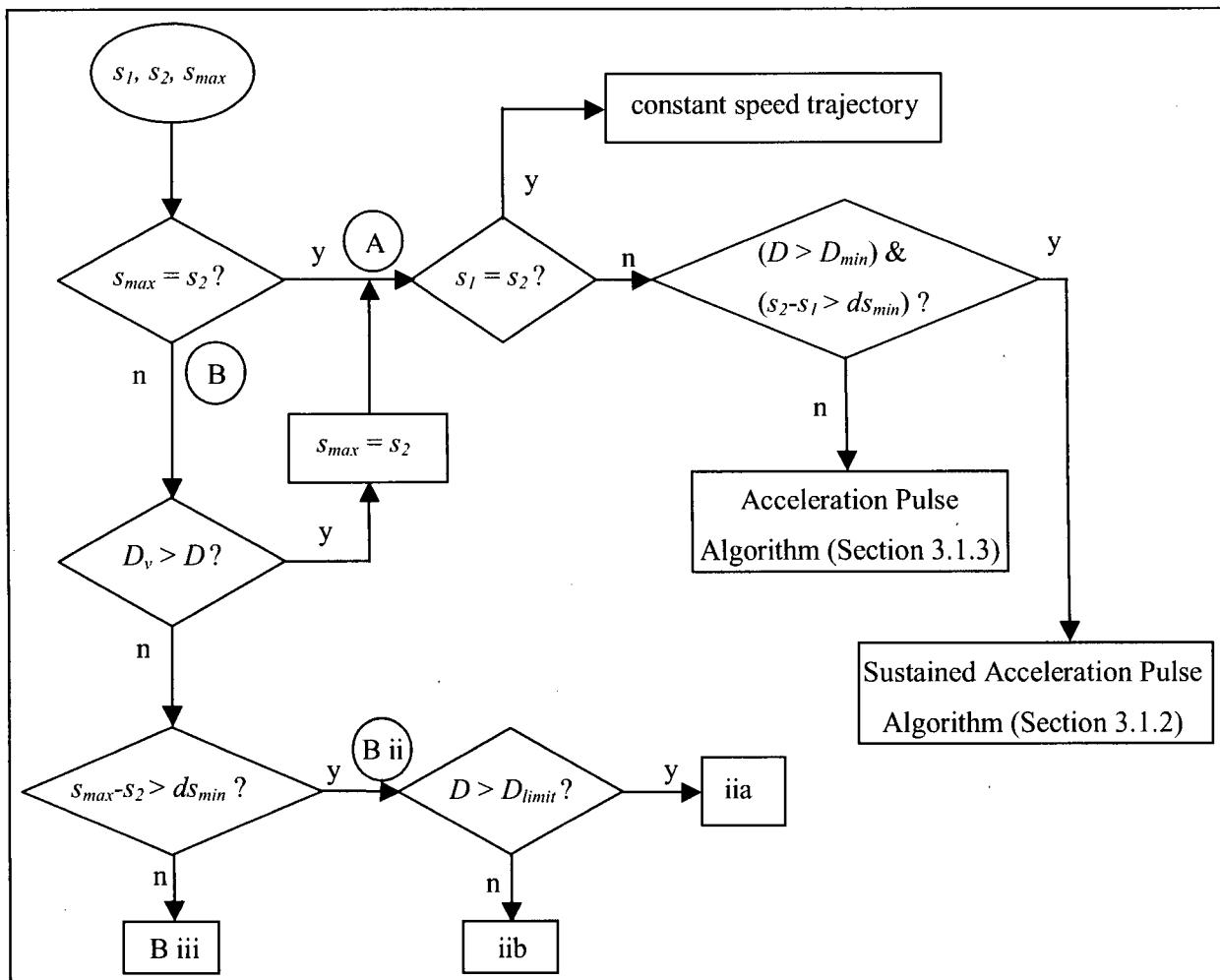


Figure 3.5. Flowchart Indicating the Algorithm to Use Based on the Way-Point Speeds

Case A. $s_{max} = s_2$

If $s_2 = s_1$, a constant speed trajectory can be used. The coefficients of Eq. 3.1, b_5, b_4, b_3, b_2 are set equal to zero, and Eq. 3.1 becomes a first-order polynomial. Otherwise, in order for the motion to be carried out as rapidly as possible, the motion begins with an acceleration phase:

IF $D > D_{min}$ (Eq. 3.20) & $s_2 - s_1 > ds_{min}$ (Eq. 3.22),

The trajectory outlined in Section 3.1.2 (with a sustained acceleration pulse) is used.

ELSE

The trajectory outlined in Section 3.1.3 (with an acceleration pulse) is used.

Case B. $s_{max} > s_2$

Several subcases are possible based on the distance, D , and the speeds s_1, s_2, s_{max} :

i. IF the distance necessary to ramp from s_1 to s_2 , D_v , is greater than D ,

reduce s_{max} such that $s_{max} = s_2$. Go to Case A.

ii. $s_{max} - s_2 > ds_{min}$, where ds_{min} is calculated from Eq. 3.22.

A minimum distance, D_{limit} , is calculated. D_{limit} is the minimum distance necessary to be able to use the sustained acceleration pulse algorithm to calculate trajectories when ramping up from s_1 to s_{limit} and then down from s_{limit} to s_2 (Figure 3.6). s_{limit} is the speed reached by an acceleration pulse from s_2 at a_{max} :

$$s_{limit} = a_{max} dt_{max} + s_2. \quad (3.29)$$

The calculation of D_{limit} , is based on integration of the acceleration profile shown in Figure 3.6:

$$D_{limit} = D_a + D_b, \quad (3.30)$$

where

$$D_a = D_{1a} + D_{2a} + D_{3a} \quad (3.31)$$

is the distance covered during the sustained acceleration pulse used to ramp up from s_1 to s_{limit} (refer to SAP algorithm, Section 3.1.2), and

$$D_b = a_{max} dt_{max}^2 + 2s_2 dt_{max} = (s_{limit} + s_2) dt_{max} \quad (3.32)$$

is the distance covered during the acceleration pulse used to ramp down from s_2 to s_{limit} .

D_{limit} is compared to the required distance, D , to determine whether the planned trajectory will be composed of acceleration pulses or sustained acceleration pulses.

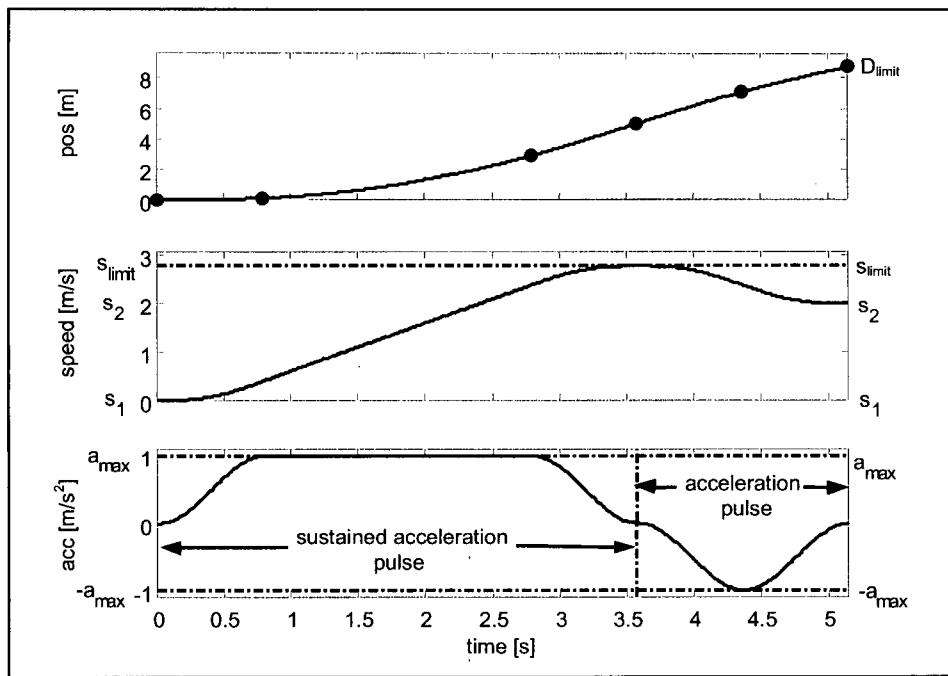


Figure 3.6. Illustration of D_{limit} (• indicate the quintic control points)

ii a. $D > D_{limit}$ - Use sustained acceleration pulses.

Sustained acceleration pulses are used to generate speed ramps from s_1 to s_{peak} and then from s_{peak} to s_2 .

Let D_x be the distance covered by a ramp from s_1 to s_{max} (using sustained acceleration pulse) and D_z be the distance covered by a ramp from s_2 to s_{max} (using sustained acceleration pulse).

IF $D > D_x + D_z$

$s_{peak} = s_{max}$, and a speed cruise at s_{max} is necessary.

ELSE

$s_{peak} < s_{max}$, and the acceleration cruises corresponding to the speed ramps to and from s_{max} are shortened based on D , as shown by the dotted line in Figure 3.7, and explained in Appendix D.

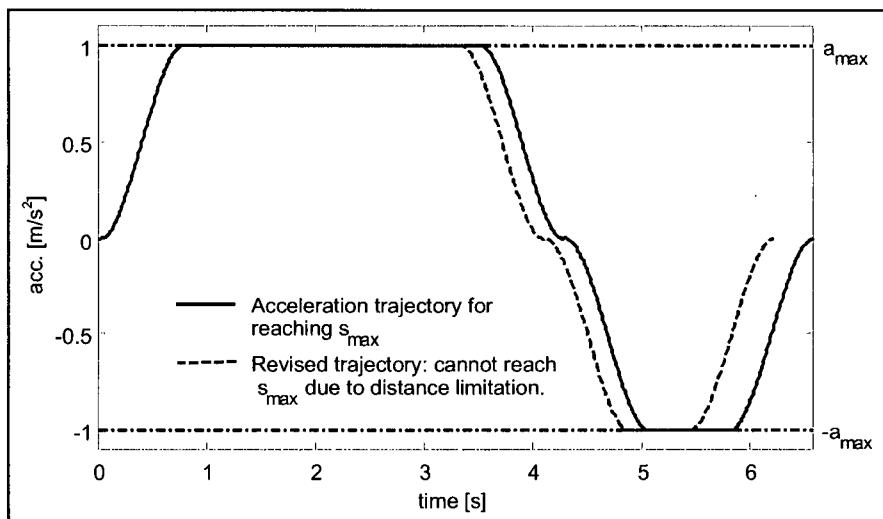


Figure 3.7. Illustration of Shortened Acceleration Segments

ii b. $D < D_{limit}$ - Use at least one acceleration pulse.

In this case, it can be shown that $s_{peak} < s_{max}$. s_{peak} is based on a speed ramp from s_1 to s_2 , followed by a speed pulse from s_2 to s_{peak} and back down to s_2 , as shown in Figure 3.8. This calculated value for s_{peak} is then used to generate the actual trajectory, consisting of a speed ramp from s_1 to s_{peak} , a speed cruise at s_{peak} followed by a speed ramp from s_{peak} to s_2 .

This calculation provides an efficient solution to this case over a more computationally intense (fourth order equation) solution for an optimal s_{peak} subject to the constraints of s_1 , s_2 , D , a_{max} , and j_{max} .

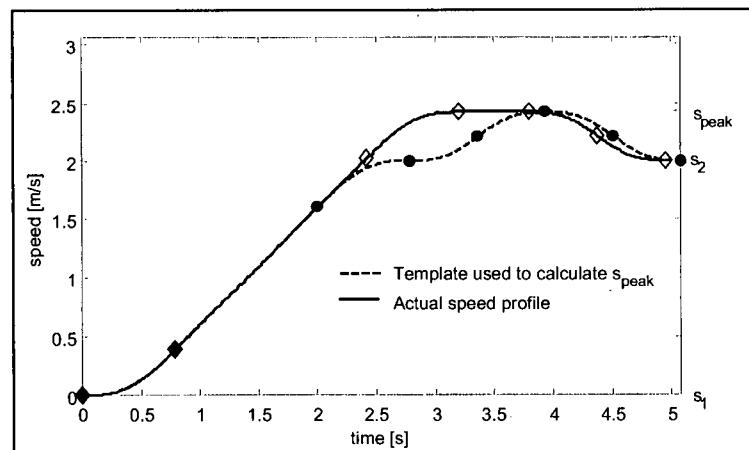


Figure 3.8. Template Used to Determine s_{peak} in Subcase iib.

iii. $s_{max} - s_2 < ds_{min}$ (Eq. 3.22)

In this case, an acceleration pulse is needed for the ramp down from s_{peak} to s_2 . As in subcase ii, a limiting distance is calculated, this time based on $s_{peak} = s_{max}$.

Let D_w be the distance required to ramp from s_2 to s_{peak} (using an acceleration pulse) and D_y be the distance to ramp required to ramp from s_1 to s_{peak} (using an acceleration pulse or a sustained acceleration pulse).

IF $D > D_w + D_y$

$s_{peak} = s_{max}$. A speed ramp is used to reach s_{peak} from s_1 , and a speed cruise segment at s_{peak} is added.

ELSE

The same approach described in subcase *iib* above is used.

Thus for Case B, the trajectory is composed of a speed ramp from s_1 to s_{peak} , a speed cruise at s_{peak} , if necessary, followed by a speed ramp from s_{peak} down to s_2 . A maximum of seven quintics can be used to describe the trajectory linking the two way-points.

3.3. Summary

In this chapter, a method of generating a trajectory between two way-points, based on a one dimensional parametrization of the straight line connecting the two points, is outlined. This trajectory is obtained as a concatenation of up to seven quintic segments (eight quintic control points). A sine-wave template is used when switching between zero and non-zero accelerations to ensure that the trajectory is smooth and that the jerk limit is reached. The trajectory is near-time-optimal since it closely follows a LSPB trajectory and utilizes the maximum allowable accelerations and jerks, as would be provided by the manufacturer, whenever possible. The resulting trajectory has the desired end conditions (in as much as is possible), and the kinematic limits such as maximum acceleration, jerk and velocity are respected throughout the trajectory. The trajectory displays no retrograde motion and is computationally simple to plan. It can be used to define the time-law along any parametrizable path, as will be shown in Chapter 5.

As shown in simulation and experimental results in Chapter 6, the trajectories generated are faster than typical industrial trajectories without requiring a high computational burden. They also compare favourably with the computationally expensive jerk-limited time-optimal motion [14].

Chapter 4

Blends

In the previous chapter, a smooth trajectory generation algorithm between two one-dimensional way-points, while respecting the path kinematic limits, was described. However, it is necessary, in many cases, to be able to plan a smooth trajectory through a number of n-dimensional way-points that describe the desired path.

If the point-to-point (straight-line) trajectory must pass exactly through every way-point, then the trajectory must halt at each way-point where the direction of the path changes. Otherwise, there will be discontinuities in the velocity profile of the trajectory. However, frequent stops at way-points slow down the overall motion. The speed that can be attained between the way-points is limited, and the overall average speed through a series of way-points is greatly reduced. These stops can be avoided by allowing the path to deviate slightly from the way-points, using blends to smoothly travel near the point while changing directions but not stopping [33].

The blending function, developed by Lloyd and Hayward for generating smooth transitions between two polynomial trajectories, was chosen, since it provides C^2 continuity during the blend [33]. In this work, this method is used to provide a smooth transition between any two straight-line segments that are pre-defined in n-dimensional space.

Straight line paths have the benefit of null normal acceleration and jerk components: all velocity, acceleration and jerk components are tangent to the path and are equivalent to the trajectory velocity, acceleration and jerk. During the blends, the normal components are no longer null, and must be considered, as will be discussed in Section 4.2.3. Lloyd and Hayward's blend methodology is extended herein to incorporate acceleration and jerk limits on the blended trajectory. Apart from blended paths, the details of generating curved paths are outside the scope of this thesis.

In Section 4.1, a brief introduction to the blends defined by Lloyd and Hayward is given. Then in Section 4.2, a subset of blends, namely *geometric* blends, is derived, and the method for limiting the acceleration and jerk during these blends is described. These blends are used in conjunction with the quintic

concatenation trajectories of Chapter 3. Together they provide a trajectory with C^2 continuity that passes within a specified distance of the given trajectory way-points.

4.1. Lloyd and Hayward's Blending Function

In Lloyd and Hayward [33], a blend is used to smoothly connect two trajectory polynomials $\mathbf{x}_1(t)$ and $\mathbf{x}_2(t)$. It is assumed that $\mathbf{x}_1(t)$ and $\mathbf{x}_2(t)$ are also smooth functions. Let the blend start at point \mathbf{p}_{1a} and end at point \mathbf{p}_{1b} (Figure 4.1). These points have corresponding velocities \mathbf{v}_{1a} and \mathbf{v}_{1b} .

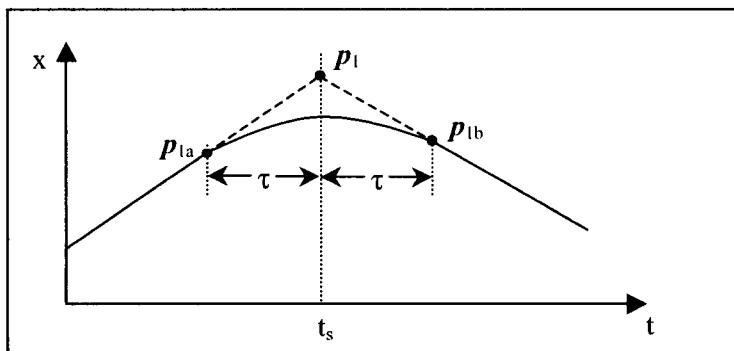


Figure 4.1. Blend in Time Domain [33]

The blend between the two polynomials $\mathbf{x}_1(t)$ and $\mathbf{x}_2(t)$ is designed in parametrized space with a new non-dimensional time coordinate,

$$\sigma = \frac{(t - t_s + \tau)}{2\tau}, \quad (4.1)$$

such that the blend occurs in the interval $\sigma \in [0,1]$. In Eq. 4.1 t is the time parameter, t_s is the time at which the trajectories intersect and τ is half the time allowed for switching between the two paths (Figure 4.1).

Lloyd and Hayward approximate the two polynomials by linear paths during the transition interval, such that :

$$\mathbf{x}_1(\sigma) = \mathbf{b}_1 + \mathbf{m}_1 \sigma, \quad (4.2)$$

$$\mathbf{x}_2(\sigma) = \mathbf{b}_2 + \mathbf{m}_2 \sigma. \quad (4.3)$$

where $\mathbf{b}_1, \mathbf{m}_1$ are respectively the position at $\sigma = 0$ and slope of $\mathbf{x}_1(\sigma)$, and $\mathbf{b}_2, \mathbf{m}_2$ are respectively the position at $\sigma = 0$ and slope of $\mathbf{x}_2(\sigma)$. $\mathbf{x}_1(\sigma)$ and $\mathbf{x}_2(\sigma)$ are intersecting paths defined within the interval $\sigma \in [0,1]$.

To ensure that the transitions between x_1 , x_2 and the blend are smooth, the blend polynomial, $x(\sigma)$, must satisfy the following boundary conditions :

$$x(0) = p_{1a}, \quad x(1) = p_{1b}, \quad (4.4)$$

$$\dot{x}(0) = 2\tau v_{1a}, \quad \dot{x}(1) = 2\tau v_{1b}, \quad (4.5)$$

$$\ddot{x}(0) = 4\tau^2 a_{1a}, \quad \ddot{x}(1) = 4\tau^2 a_{1b}, \quad (4.6)$$

where a_{1a} is the acceleration at the start of the blend and a_{1b} is the acceleration at the end of the blend.

A quintic polynomial could be used to generate a blend satisfying the above conditions. However, the quintic does not control the shape of the blend (that is, the path that it follows). To address this problem, Lloyd and Hayward blend $x_1(t)$ and $x_2(t)$ together using a convex average of the two polynomials. In addition, the average acceleration is minimized by adding a damping function, $\beta(\sigma)$, that does not affect the end conditions of the blend. The blend function is defined as :

$$x(\sigma) = x_1(\sigma) + \alpha(\sigma)(x_2(\sigma) - x_1(\sigma)) - \kappa\beta(\sigma)v_d, \quad (4.7)$$

where

$$\alpha(\sigma) = 6\sigma^5 - 15\sigma^4 + 10\sigma^3, \quad (4.8)$$

$$\beta(\sigma) = \sigma^6 - 3\sigma^5 + 3\sigma^4 - \sigma^3, \quad (4.9)$$

$$v_d = m_2 - m_1, \quad (4.10)$$

and κ is the parameter that controls the amount of acceleration compensation to apply. The resulting blend function is a sixth-order polynomial, if $x_1(t)$ and $x_2(t)$ are first-order polynomials.

4.2. Geometric Blends

Herein, Lloyd and Hayward's function is used for the case when the straight-line paths between waypoints are not changing with time. That is, they are known in advance, and the start and end position of the blends are given through the definition of a variable, *tightness*. *tightness* is a measure of how closely the trajectory must approach a given way-point before blending to the next way-point trajectory can begin. This measure is important for industrial applications where the user will typically define the path as a series of straight-line segments between way-points. The *tightness* signifies the importance of not deviating from the straight-line path; to avoid collisions with other objects, for example. A large tightness allows the manipulator to quickly pass through a series of points by "cutting" the corners at the waypoints. A small *tightness* distance indicates that following the straight-line path is of higher importance than quickly travelling past the way-points. Zero *tightness* indicates that unless the adjoining paths are colinear, the trajectory must stop at the intermediary way-point (to avoid a discontinuous velocity profile). In this case, the manipulator will follow a series of straight line paths (Figure 4.2).

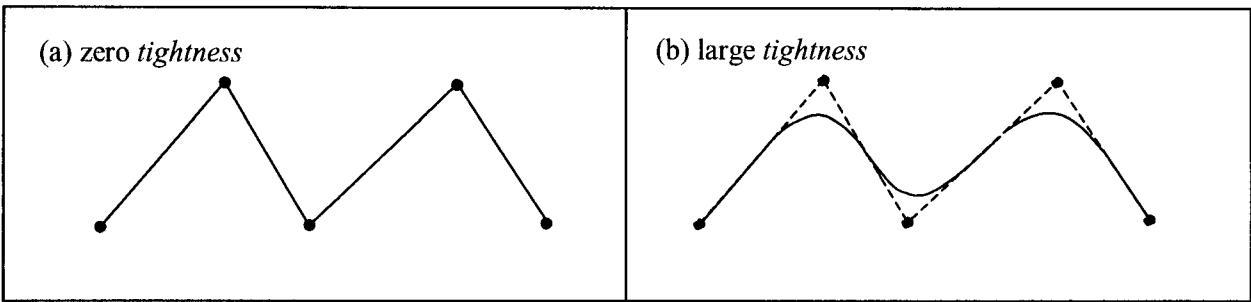


Figure 4.2. Effect of *tightness* on Path

In addition, the properties of the blends are set such that the blend is symmetrical about its midpoint. This provides the ability to limit the jerk and acceleration during the blend. Hereafter, such a blend is referred to as a ***geometric*** blend.

The blends are used to join intersecting straight line segments. The condition of C^2 continuity during the entire trajectory, constructed of straight line segments, with zero acceleration end conditions, connected to blends, is respected, since the blends are sixth order polynomials that start and end with zero acceleration.

4.2.1. Conditions Necessary for a Geometric Blend

In the following discussion a 2D case is considered. However, the method is easily extended into n-dimensions as shown in Subsection 4.2.4. Figure 4.3 shows the geometry of a 2D blend that is used to connect two intersecting straight lines $\mathbf{p}_1 - \mathbf{p}_0$ and $\mathbf{p}_2 - \mathbf{p}_1$. A distance, *tightness*, is provided by the user, to indicate how closely the point being blended around, \mathbf{p}_1 , must be approached. That is,

$$\|\mathbf{p}_1 - \mathbf{p}_{1a}\| = \|\mathbf{p}_1 - \mathbf{p}_{1b}\| = \text{tightness}, \quad (4.11)$$

$$\mathbf{p}_{1a} = \mathbf{p}_1 - \frac{(\mathbf{p}_1 - \mathbf{p}_0)}{\|\mathbf{p}_1 - \mathbf{p}_0\|} \text{tightness}, \quad (4.12)$$

and

$$\mathbf{p}_{1b} = \mathbf{p}_1 + \frac{(\mathbf{p}_2 - \mathbf{p}_1)}{\|\mathbf{p}_2 - \mathbf{p}_1\|} \text{tightness}, \quad (4.13)$$

where \mathbf{p}_{1a} is the position at the start of the blend and \mathbf{p}_{1b} is the position at the end of the blend.

In a 2D case, the distance *tightness* describes a circle, centred at \mathbf{p}_1 . The blend must lie within the circle. According to the industrial convention followed (cf. Section 2.6.2.2), the blend must be located within the isosceles triangle with vertices \mathbf{p}_{1a} , \mathbf{p}_1 and \mathbf{p}_{1b} . Planning a motion that lies outside this triangle may result in collision with other objects. Eq. 4.11 ensures that the paths intersect at $\sigma = 0.5$.

The two polynomial paths between which the blend is used are first-order. That is, the linearity assumption in Eqs. 4.2 and 4.3 is exact (i.e. the start and end accelerations are null). Thus, the variables in Eqs. 4.2 and 4.3 are:

$$\mathbf{b}_1 = \mathbf{p}_{1a}, \quad \mathbf{b}_2 = \mathbf{p}_{1b} - 2\tau \mathbf{v}_{1b}, \quad (4.14)$$

$$\mathbf{m}_1 = 2\tau \mathbf{v}_{1a}, \quad \mathbf{m}_2 = 2\tau \mathbf{v}_{1b}. \quad (4.15)$$

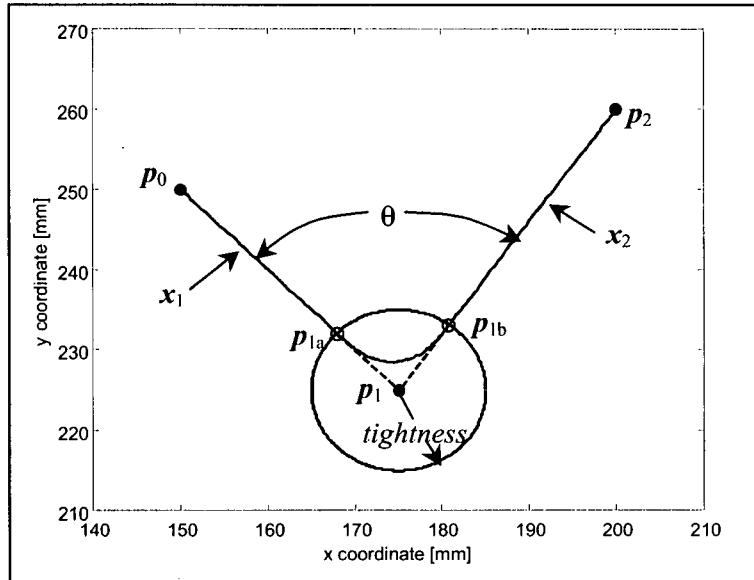


Figure 4.3. 2D Blend Geometry

Another requirement of a geometric blend is that the blend incoming velocity, \mathbf{v}_{1a} , must lie along the line between \mathbf{p}_0 and \mathbf{p}_1 , and the blend outgoing velocity, \mathbf{v}_{1b} , must be colinear with the line from \mathbf{p}_1 to \mathbf{p}_2 . In addition, the incoming and outgoing speeds, s_{1a} and s_{1b} , must be equal ($s_{1a} = s_{1b} = s$) so that the acceleration and jerk during the blend can be limited. This leads to the following definitions of the incoming and outgoing blend velocities :

$$\mathbf{v}_{1a}(t) = s \frac{(\mathbf{p}_1 - \mathbf{p}_{1a})}{\|\mathbf{p}_1 - \mathbf{p}_{1a}\|}, \quad (4.16)$$

$$\mathbf{v}_{1b}(t) = s \frac{(\mathbf{p}_{1b} - \mathbf{p}_1)}{\|\mathbf{p}_{1b} - \mathbf{p}_1\|}. \quad (4.17)$$

Given that $\|\mathbf{p}_1 - \mathbf{p}_{1a}\| = \|\mathbf{p}_1 - \mathbf{p}_{1b}\|$, and the two lines $\mathbf{x}_1(\sigma)$ and $\mathbf{x}_2(\sigma)$ intersect at $\sigma = 0.5$,

$$\begin{aligned} \mathbf{b}_1 + 0.5 \mathbf{m}_1 &= \mathbf{b}_2 + 0.5 \mathbf{m}_2 \\ \mathbf{p}_{1a} + \tau \mathbf{v}_{1a} &= \mathbf{p}_{1b} - \tau \mathbf{v}_{1b} \end{aligned} \quad (4.18)$$

Substituting Eqs. 4.11, 4.16 and 4.17 into Eq. 4.18, gives the equation for the half time allowed for the blend:

$$\tau = \frac{\text{tightness}}{s}. \quad (4.19)$$

Since the two paths \mathbf{x}_1 and \mathbf{x}_2 are linear and intersect at $\sigma = 0.5$, the optimal value of the acceleration compensation parameter, $\kappa = 15/2$, is used (suggested by Lloyd and Hayward). This ensures that the blend will remain within the isosceles triangle with vertices \mathbf{p}_{la} , \mathbf{p}_l and \mathbf{p}_{lb} .

A blend that has all the characteristics listed in this section has a fixed relationship between the maximum acceleration and jerk and the incoming/outgoing blend speed, and allows these variables to be limited as described in Section 4.2.3. It can be shown that for a blend that is not "geometric", the limiting equations for acceleration and jerk cannot be solved in a tractable analytical manner, although a numerical method could be applied.

The velocities at the end of the straight-line trajectories correspond to those at the start and the end of the geometric blend. In addition, when the quintic concatenation method is used to describe the straight line trajectories, the trajectories have zero acceleration end conditions and C^2 continuity. The geometric blend function also has zero acceleration end conditions and C^2 continuity. Thus, the geometric blend can be used to join straight line trajectory segments, maintaining C^2 continuity along the entire trajectory. Herein, the geometric blend is used and the methods described in Section 4.2.3 are employed to limit the acceleration and jerk during the blend.

4.2.2. Velocity, Acceleration and Jerk Profiles for a 2D Geometric Blend

In this section, an overview of the magnitude and individual components of the blend velocity, acceleration and jerk profiles is presented. The objective is to provide insight into the manner in which the acceleration and jerk are limited.

The velocity profiles for the x and y dimensions and the velocity magnitude in the σ -domain are shown in Figure 4.4 for the 2D blend example presented in Figure 4.3. Conversion to the time domain simply scales the profiles by a factor of $0.5\tau^1$. In this case, the velocity in the x -direction decreases slightly, while the velocity in the y -direction has a correspondingly slight increase in magnitude but also a change in sign, which is consistent with the change in path direction of Figure 4.3. It can be seen that the speeds

at the start and the end of the blend are equal. The speed decreases during the blend; in fact, in all cases during a geometric blend the imposed speed limit will be respected.

The acceleration profiles (Figure 4.5) have two local maxima (if the absolute values are considered). The profiles are symmetric about the line $\sigma = 0.5$. As is expected, the y -coordinate acceleration magnitude is much large than that of the x -coordinate. The largest jerks occur at the start and end of the blend (Figure 4.6), with the y -coordinate experiencing the largest jerk.

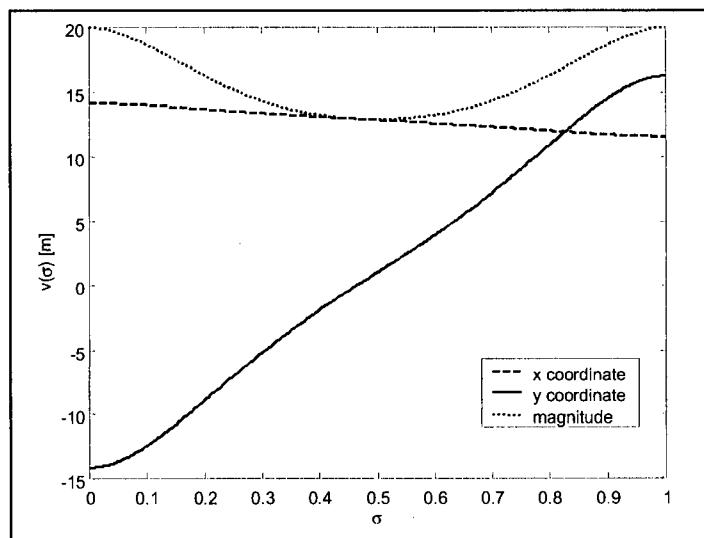


Figure 4.4. Velocity Profiles for a 2D Geometric Blend

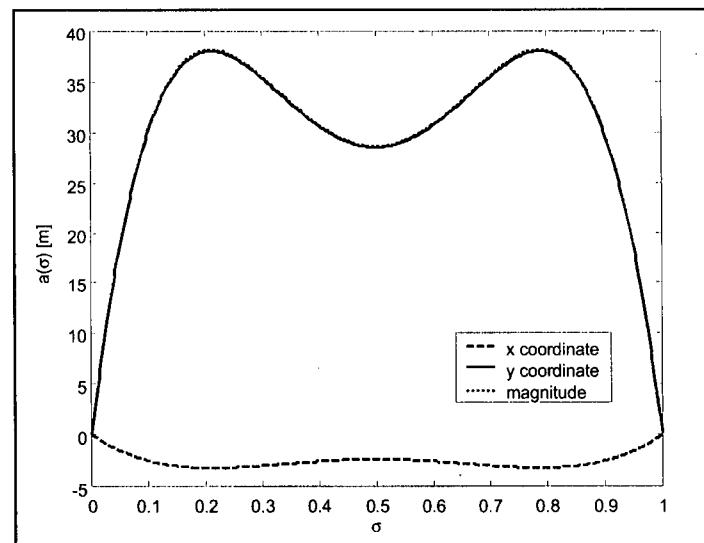


Figure 4.5. Acceleration Profiles for a 2D Geometric Blend

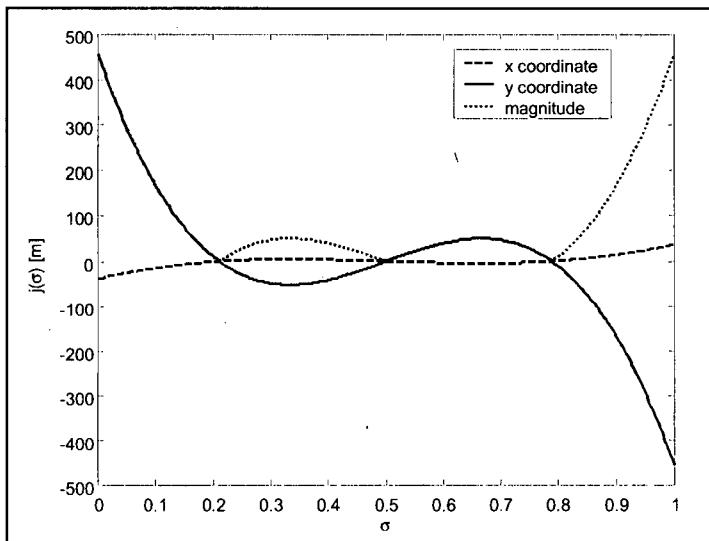


Figure 4.6. Jerk Profiles for a 2D Geometric Blend

These maximum jerks and accelerations occur at the same σ -domain location for all geometric blends. Therefore, the maximum acceleration and jerk magnitudes also occur in these fixed locations. In all cases, the profiles have the same shapes. They are simply scaled depending on the magnitude and direction of the variable changes. In other words, the location of the acceleration and jerk maxima depend on the fifth and sixth order polynomials $\alpha(\sigma)$ (Eq. 4.8) and $\beta(\sigma)$ (Eq. 4.9).

4.2.3. Limiting Acceleration and Jerk

The path of a blend is not linear, thus, the acceleration and jerk vectors are not tangent to the path. Therefore, limiting the acceleration and jerks during blends is slightly more complicated than limiting them along a straight-line path. Although Lloyd and Hayward's method minimizes the average acceleration during the blend, it does not provide a method for limiting the *maximum* acceleration and jerk during the blend [33]. It can be shown that given a geometric blend, the maximum acceleration occurring during the blend is

$$a_{blendMax} = \frac{s^2}{\rho \cdot tightness} \sqrt{\frac{1 + \cos \theta}{2}}, \quad (4.20)$$

where ρ is a proportionality constant ($\rho = 0.8$) and (cf. Figure 4.3)

$$\cos \theta = -\frac{(p_1 - p_{1a}) \bullet (p_{1b} - p_1)}{tightness^2}. \quad (4.21)$$

Eq. 4.20 is based on a constant acceleration assumption. Although the acceleration is not constant during the blend, it can be shown that the velocity is symmetrical about a constant acceleration line. That is, the areas above and below the constant acceleration line are equal. The maximum acceleration always occurs at the same two values of σ : $\sigma = 0.211$ and $\sigma = 0.7887$ (Figure 4.5, Appendix E). The proportionality constant, ρ , in Eq. 4.20 is the factor used to calculate the maximum blend acceleration based on the corresponding constant acceleration (refer to Appendix E, Section E.1).

It is, therefore, possible to ensure that the blend acceleration is limited by bounding the speed at the start and the end of the blend. That is, the maximum speed at which the blend can be started and ended, $s_{blendMaxAcc}$, subject to the maximum path acceleration, a_{max} , is

$$s_{blendMaxAcc} = \sqrt{\frac{\rho \cdot a_{max} \cdot tightness}{\sqrt{\frac{1 + \cos\theta}{2}}}}. \quad (4.22)$$

The maximum jerk during a blend occurs at the start and the end of the blend (Figure 4.6) and is given by (refer to Appendix E, Section E.3):

$$j_{blendMax} = \frac{15s^3}{2tightness^2} \cos\left(\frac{\theta}{2}\right). \quad (4.23)$$

Once again, the speed can be limited such that the maximum blend jerk is equal to the maximum allowable jerk, j_{max} :

$$s_{blendMaxJerk} = \left(\frac{2 \cdot j_{max} \cdot tightness^2}{15 \cos\left(\frac{\theta}{2}\right)} \right)^{\frac{1}{3}}. \quad (4.24)$$

Thus the maximum speed with which a blend can be started (or ended) while respecting the jerk and acceleration limits is

$$s_{blendMax} = \min(s_{blendMaxAcc}, s_{blendMaxJerk}), \quad (4.25)$$

where **min** is a function returning the smaller of the two speeds.

4.2.4. Extension into n-dimensions

The extension of the geometric blend definition and jerk and acceleration limitations presented in Sections 4.2.1 and 4.2.3 from two to n dimensions requires the manipulation of the distance *tightness*. In 3D, the given distance *tightness* defines a sphere centred at the way-point. The blend must lie within this

sphere. The blend is located on the plane defined by the two vectors $(\mathbf{p}_I - \mathbf{p}_{Ia})$ and $(\mathbf{p}_{Ib} - \mathbf{p}_I)$ and must not cross the planes formed by the vectors $\{\mathbf{p}_I - \mathbf{p}_{Ia}, \hat{\mathbf{u}}\}$ and $\{\mathbf{p}_{Ib} - \mathbf{p}_I, \hat{\mathbf{u}}\}$, where

$$\hat{\mathbf{u}} = (\mathbf{p}_I - \mathbf{p}_{Ia}) \times (\mathbf{p}_{Ib} - \mathbf{p}_I). \quad (4.26)$$

The equations provided in Sections 4.2.1 and 4.2.3 apply directly to the 3D case.

In 6D, the second case of interest in this work, a “hypercube” is determined based on the given distance of approach, *tightness*. A hyper-cube, rather than a hyper-sphere, is used since in 6D joint space, the *tightness* must be individually respected for each joint. Alternatively, a Cartesian *tightness* sphere for the manipulator end point could be imposed and then converted via the inverse Jacobian to a joint space ellipsoid. However, in this case the way-points are planned in joint space. Thus, the extra conversion of Cartesian limits is somewhat costly and artificial for trajectory planning.

Assuming that *tightness* applies to each dimension, a “hypercube” centred at \mathbf{p}_I , with edges corresponding to \pm *tightness* from the point \mathbf{p}_I , is defined. The point of intersection of the hypercube plane closest to \mathbf{p}_I and the 6D line, \mathbf{x}_I , is labelled \mathbf{p}_{Ia} . This leads to the definition of a new *tightness*, *tighthyper*, which is the distance between \mathbf{p}_I and \mathbf{p}_{Ia} , and \mathbf{p}_I and \mathbf{p}_{2a} :

$$\text{tighthyper} = \left(\frac{\text{tightness}}{\max(|\mathbf{p}_I - \mathbf{p}_0|)} \right) \|\mathbf{p}_I - \mathbf{p}_0\|, \quad (4.27)$$

where $\max(|\mathbf{p}_I - \mathbf{p}_0|)$ is a function that returns the largest coordinate difference between \mathbf{p}_I and \mathbf{p}_0 . *tightness* is then replaced by *tighthyper* in Eqs 4.12, 4.13, 4.19, 4.20-4.24. Eq. 4.27 ensures that all axes come within the specified limit.

4.3. Summary

The blends developed by Lloyd and Hayward [33] were introduced, and a special category of blends, geometric blends, was defined so that blend acceleration and jerk could easily be limited.

Trajectories formed of straight line segments connected with blends meet the C^2 continuity requirement because:

- (i) the straight line segments start and end with zero acceleration (Chapter 3) and are based on a quintic time law, and,
- (ii) the blends are sixth order polynomials that start and end with zero acceleration.

Chapter 5

Connecting the Dots

In Chapter 3, a method to generate a trajectory between two one-dimensional way-points while respecting the kinematic limits was described. In Chapter 4, a description of blends, which can be used to join straight line trajectories, while maintaining C^2 continuity, was provided. In this chapter, the method used to plan a smooth trajectory passing through or close to a number of n-dimensional way-points that describe the desired path is described.

Manipulator way-points can be given in either joint or task space. Thus, trajectories must be designed in either space, depending on the requirements of the user. The trajectory must then be converted into the appropriate controller command signals. A manipulator's position can be described in terms of its joint angles or in terms of its end effector position and orientation relative to a fixed coordinate system. The trajectory path may also be defined in either of these spaces. If the end effector must travel along a straight line in Cartesian space, it is most effective to plan the trajectory in task space, and then convert the task space data to joint space using inverse kinematics, and then proportionally to motor pulses.

For motions described in task space, the positional path is parametrized and the path length is used to determine the trajectory through a sequence of way-points. When changing direction between two straight line segments, the blends described in Chapter 4 can be used. The orientation motion is superimposed on the positional motion. The overall method developed in this work is described in Section 5.1.

For motions described in joint space, two methods have been developed. The first consists of moving along a straight line in joint space between the two way-points and using blends for changes in direction (just as in task space motion). The second consists of determining the limiting joint (i.e. the slowest joint) for the motion between the two way-points. The trajectory for the limiting joint is calculated based on the method of Chapter 3. The other joints' trajectories are determined based on the limiting joint's trajectory time. These methods are further developed in Section 5.2.

Finally, the method used to plan a trajectory through multiple way-points is described in Sections 5.3 and 5.4.

It should be noted that the generation of trajectory coefficients by finding quintic control points and the trajectory generation at every controller time step are two separate processes. Initially, a trajectory between N way-points is planned in such a way that it is feasible and the corresponding quintic control points and blend coefficients are calculated and stored, as described in this chapter. Once the trajectory has been planned, this information is sent to the "parser". The parser calculates the trajectory points at each controller time step based on Eq. 5.1 for straight lines and Eq. 4.7 for blends. The parser is described in Chapter 6.

5.1. Task Space Straight-Line Mode

In task space, the trajectory planning is done in two steps. First the Cartesian dimensions are considered, since they can be treated vectorially, then the orientations are smoothly interpolated between the way-points.

5.1.1. *Cartesian Dimensions*

In Cartesian space, the trajectory between N way-points is composed of straight-lines segments joined together with blends. A straight line path was chosen for the Cartesian dimensions (x,y,z) as outlined in Chapter 2 (Section 2.6.1.2). As discussed in Chapter 3 (Eq. 3.28), the straight line path is parametrized in terms of time,

$$\mathbf{p}(t) = \mathbf{p}_s + \frac{(\mathbf{p}_e - \mathbf{p}_s)}{\|\mathbf{p}_e - \mathbf{p}_s\|} q(t), \quad (5.1)$$

where \mathbf{p}_s and \mathbf{p}_e are respectively the initial and final position vectors of the path and $q(t)$ is the trajectory generated using the algorithm described in Chapter 3. $q(t)$ starts at zero and ends at $\|\mathbf{p}_e - \mathbf{p}_s\|$. Limits are placed on the trajectory derivatives, $\dot{q}(t)$, $\ddot{q}(t)$ and $\dddot{q}(t)$. These limits are respectively the Cartesian speed, acceleration and jerk limits (Appendix F, Section F.2).

To generate a blend, the blend coefficients, \mathbf{b}_1 , \mathbf{b}_2 , \mathbf{m}_1 and \mathbf{m}_2 (Eqs. 4.14, 4.15) as well as the blend half-time, τ (Eq. 4.19), must be calculated. Eq. 4.7 can then be used to generate the position along the blend at each time interval.

A motion from way-point \mathbf{p}_u to way-point \mathbf{p}_v in the Cartesian domain can be composed of various segments depending on the speeds given to these points (Figure 5.1):

- If both points are stop points (points with zero speed conditions), the motion is a straight line.
- If only \mathbf{p}_u is a stop point, the motion is composed of a straight line followed by a blend.

- If only p_v is a stop point, the motion is composed of a blend followed by a straight line.
- If neither point is a stop point, the motion is composed of a blend, followed by a straight line and another blend.

Thus, points p_s and p_e in Eq. 5.1 are not necessarily equal to way-points p_u and p_v .

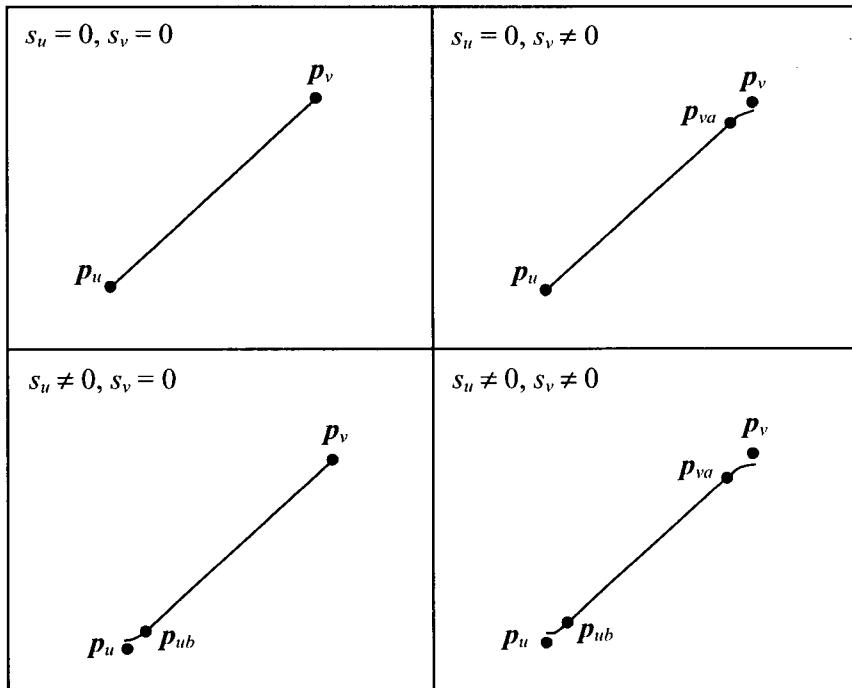


Figure 5.1. Four Different Cases of Straight Line and Blend Motions

Therefore, a motion, in Cartesian space, between the three way-points (p_0, p_1, p_2), where the second way-point is not a stop point, is composed of: (i) a straight line trajectory between points p_0 and p_{1a} , (ii) a 3D geometric blend between points p_{1a} and p_{1b} , as described in Chapter 4, and (iii) a straight line trajectory leaving from p_{1b} and either reaching p_2 (if p_2 is a stop point) or p_{2a} (if p_2 is not a stop point). This set of motions is repeated for any number of points (Figure 5.2). It is possible that a specified way-point speed cannot be reached because a previous way-point's speed is either too high or too low. The solution to this problem will be discussed in Section 5.4.

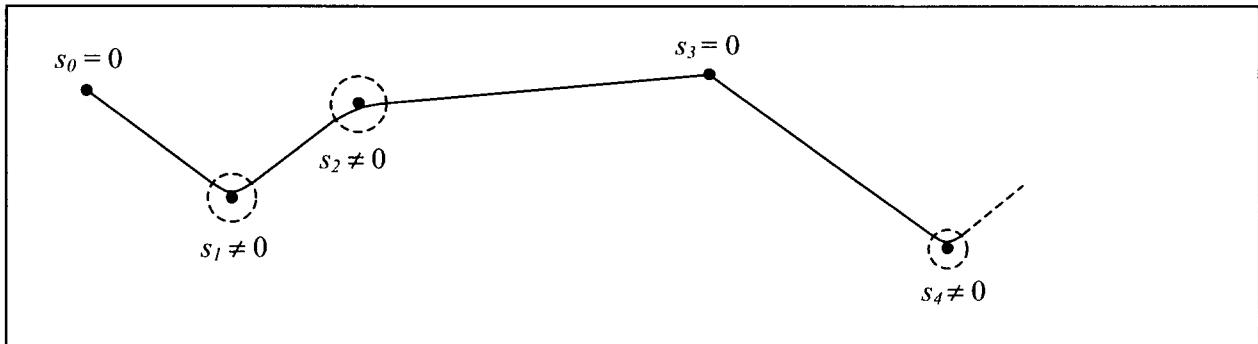


Figure 5.2. Sketch of a Multiple Point Path

5.1.2. Orientation

Blending was not used in the orientation domain since it is computationally intensive. This is due to the nonlinear aspect of the orientation angles, which makes matching of end conditions complex. Instead, a quintic is used to provide the trajectory for the orientation, based on an axis-angle representation of the orientation, and the rotational velocity and acceleration are set to zero at each way-point, as described below. Although the rotational motion stops at each way-point, the Cartesian motion typically does not stop at each way-point, unless very short Cartesian motions with large orientation changes are required. In this case, it is usually desirable for the positional motion to be slow or stopped. Otherwise, an overall view of the trajectory shows that the manipulator does not completely stop moving. Only the axes related to the rotational motion stop at each way-point. The axis-angle approach for orientation motion is desirable for a number of typical industrial manipulator motions, such as screwing, turning and pouring, where it is important that the end effector rotate about a single axis.

The orientation coordinates given at each point can be converted to rotation matrix form with respect to the world or machine base coordinate system. The change in orientation between two way-points (e.g. p_0 and p_1) can also be represented in the form of a rotation matrix:

$$R_{01} = R_{w0}^{-1} R_{w1} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}. \quad (5.2)$$

where R_{w0} and R_{w1} are the rotation matrices corresponding to the orientations, expressed in the world frame, at p_0 and p_1 respectively. Every rotation matrix can be expressed as a rotation of a given angle about an arbitrary axis in space [42]. Thus, to move smoothly between p_0 and p_1 in the orientation domain, the orientation trajectory must rotate about the axis of the rotation matrix, R_{01} . The angular rotation and acceleration about that axis must be continuous in order to ensure limited jerk in orientation space.

The orientation trajectory is based upon a stop point to stop point quintic (Figure 5.3). That is, a quintic with zero velocity and acceleration end conditions. The quintic starts at position zero and ends at position ϕ_2 ,

$$\phi_2 = \cos^{-1} \left(\frac{r_{11} + r_{22} + r_{33} - 1}{2} \right), \quad (5.3)$$

where r_{11}, r_{22}, r_{33} are obtained from Eq. 5.2. The time allowed for the quintic will be discussed in the next section.

Setting the start and end rotational velocities and acceleration to zero does not provide the most time-optimal solution. However, it makes the superimposition of the orientation and positional trajectory components tractable for on-line computation.

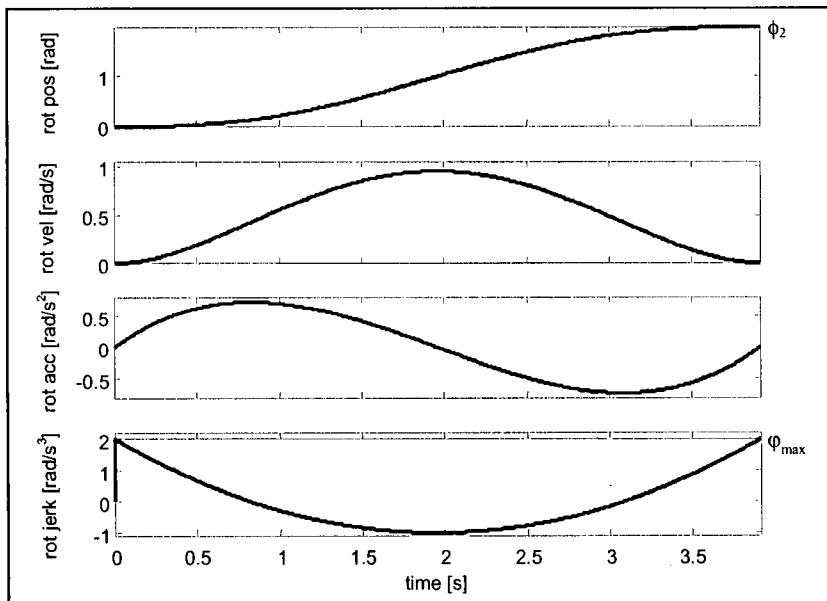


Figure 5.3. Rotational Motion Trajectory

It is possible to limit the angular velocity, acceleration and jerk about the axis, by allowing enough time for the motion based on:

$$t_\omega = \frac{15\phi_2}{8\omega_{max}}, \quad (5.4)$$

$$t_\lambda = \sqrt{\frac{10\phi_2}{\sqrt{3}\lambda_{max}}}, \quad (5.5)$$

and

$$t_\varphi = \left(\frac{60\phi_2}{\varphi_{max}} \right)^{\frac{1}{3}} \quad (5.6)$$

where t_ω is the minimum motion time necessary so as not to violate the rotational velocity limit, ω_{max} , t_λ is the minimum motion time necessary so as not to violate the rotational acceleration limit, λ_{max} , and t_φ is the minimum motion time necessary so as not to violate the rotational jerk limit, φ_{max} , when rotating about an axis by ϕ_2 radians with zero rotational velocity and acceleration end conditions.

5.1.3. Combining Cartesian and Orientation Trajectories and Generating a Trajectory

A task space motion is formed of an orientation motion superimposed upon a positional motion. In order to superimpose these two motions, they must have equal trajectory times.

The Cartesian motion time is composed of various components since it can be formed in four different ways as described in Section 5.1.1. A blend around a point, p_w , is divided into two, so that the first half is allotted to the motion between points p_{w-1} and p_w and the second half to the motion between points p_w and p_{w+1} . Thus, the Cartesian motion time between two way-points, p_u and p_v , has three components (some of which may be null): the first half-blend time, τ_u , the straight line motion time, t_{sl} , and the second half-blend motion time, τ_v . The total positional time,

$$t_{cart} = \tau_u + t_{sl} + \tau_v, \quad (5.7)$$

is the time necessary to move between way-points p_u and p_v in the Cartesian domain while respecting all Cartesian limits.

The orientation motion time is more flexible, provided that the rotational limits are respected. Therefore, once the optimal Cartesian motion time has been determined, it is compared to t_ω (Eq. 5.4), t_λ (Eq. 5.5) and t_φ (Eq. 5.6). If

$$t_{cart} \geq t_{rot}, \quad (5.8)$$

where $t_{rot} = \max(t_\omega, t_\lambda, t_\varphi)$, then t_{cart} is the time for the motion between p_u and p_v . One can note that there is an added advantage to using an orientation quintic with null speed and acceleration end conditions: its time can be easily modified without causing adverse effects, as long as Eq. 5.8 is satisfied (retrograde orientation motion is a low risk issue).

If Eq. 5.8 is not satisfied, then the motion cannot be carried out in time t_{cart} without violating the rotational limits. In this case, the second way-point, \mathbf{p}_v is redefined as a stop point, and t_{rot} is assigned as the motion time. The Cartesian trajectory is then recalculated. If t_{rot} is larger than the time necessary to move from \mathbf{p}_u and \mathbf{p}_v with \mathbf{p}_v as a stop point, then the positional components are held constant until the rotational motion has ended. This does not provide the fastest possible motion between the two points (since it might be possible to end the motion with a smaller than specified Cartesian velocity); however, it is a computationally efficient solution.

5.2. Joint Space Mode

5.2.1. Straight Line Joint Space Motion

Planning in joint space is done in a similar manner to planning the task space Cartesian motion. That is, the trajectory is planned along straight-line paths in n -dimensional joint space connected with blends (where n is the number of actuators). One can note that a straight line in joint space does not imply a straight-line task-space motion of the manipulator end effector. A straight-line joint-space path is used because it is the shortest path between the two joint space points. Furthermore, Sahar and Hollerbach show that the time-optimal motion (considering dynamic limits) between any two points approaches a straight line in joint space [41]. Finally, a straight line path also allows for coordinated axis motion. All axes reach the way-points at the same time, starting and stopping in a coordinated manner, and the motion path is independent of the speed along the path.

The straight line path is parametrized using Eq. 5.1. Limits are placed on the trajectory derivatives, $\dot{q}(t)$, $\ddot{q}(t)$ and $\dddot{q}(t)$, thus, the overall path limits are used rather than individual joint limits (refer to Appendix F, Section F.1).

In joint space, a motion between three way-points, $(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)$, (where the second way-point is not a stop point) is formed by a straight line trajectory between points \mathbf{p}_0 and \mathbf{p}_{1a} , a 6D blend between points \mathbf{p}_{1a} and \mathbf{p}_{1b} , as described in Chapter 4, followed by a straight line trajectory leaving from \mathbf{p}_{1b} and either reaching \mathbf{p}_2 (if \mathbf{p}_2 is a stop point) or \mathbf{p}_{2a} (if \mathbf{p}_2 is not a stop point). Again, this motion can be extended for any number of points (Figure 5.2), and the same four path possibilities exist in joint space as in task space (Section 5.1.1). It is possible that a specified way-point speed cannot be reached because of a previous point's speed being either too high or too low. The solution to this problem will be discussed in Section 5.4.

5.2.2. Independent Joint Motion

A further motion option was examined for joint space planning: that of planning the joint trajectories independently. That is, the limiting joint for a motion between two way-points is determined and the optimal motion for this joint planned using the method of Chapter 3. The other joint motions are then determined based on this time and the demanded way-point speeds using a series of concatenated quintics generated such that the individual joint jerk, acceleration and speed limitations are respected. The way-point speeds are attained, if feasible.

A motion between two joint space way-points may require some joints to move a large distance and others to move a smaller distance. Thus, it may be necessary to stop the joint travelling a short distance during the motion and then accelerate it to reach the demanded end speed when the slower joint is reaching the end of its trajectory. This leads to uncoordinated joint motion: that is all joints may not start and stop at the same time. In addition, this type of motion may produce varying trajectory paths as the way-point speeds are increased or decreased. For these reasons, it was determined that independent motion was not desirable. However, it was found that this type of motion is faster than straight line joint motion for a multiple way-point path because the way-point speeds can be specified and reached independently for each joint.

5.3. Planning a Straight-Line Motion Between Two Way-Points With Variable Speeds

In Chapters 3, 4, and the preceding sections all the components necessary to plan a trajectory between way-points with non-zero speeds have been established. This section provides an overview of how a trajectory between any two given way-points defined in either n-dimensional or task space is planned (Figures 5.4-5.6).

Assuming a motion between two way-points, p_i , and p_{i+1} is being planned, it is first necessary to check whether the acceleration and jerk limits will be violated during the blend around point p_{i+1} (if p_{i+1} is not a stop point). If the limits will be violated, the speed at p_{i+1} is reduced using Eq. 4.25. Then, the start and end points of the straight-line segment, p_s and p_e , are determined based on the blend requirements at each way-point. At this point, the planning method differs depending on whether the planning is being carried out in task space or joint space.

In the case of task space planning, the quintic control points corresponding to the trajectory along the straight line segment in Cartesian space are determined, along with the blend coefficients for a blend

around \mathbf{p}_{i+1} (Figure 5.5). The orientation parameter, ϕ_2 , (Eq. 5.3) is then calculated, and the method described in Section 5.1.3 is used to generate the overall trajectory.

Joint space or n-dimensional planning is much simpler: the quintic control points corresponding to the straight line segment are calculated, as are the blend coefficients for the blend around \mathbf{p}_{i+1} , if necessary (Figure 5.6).

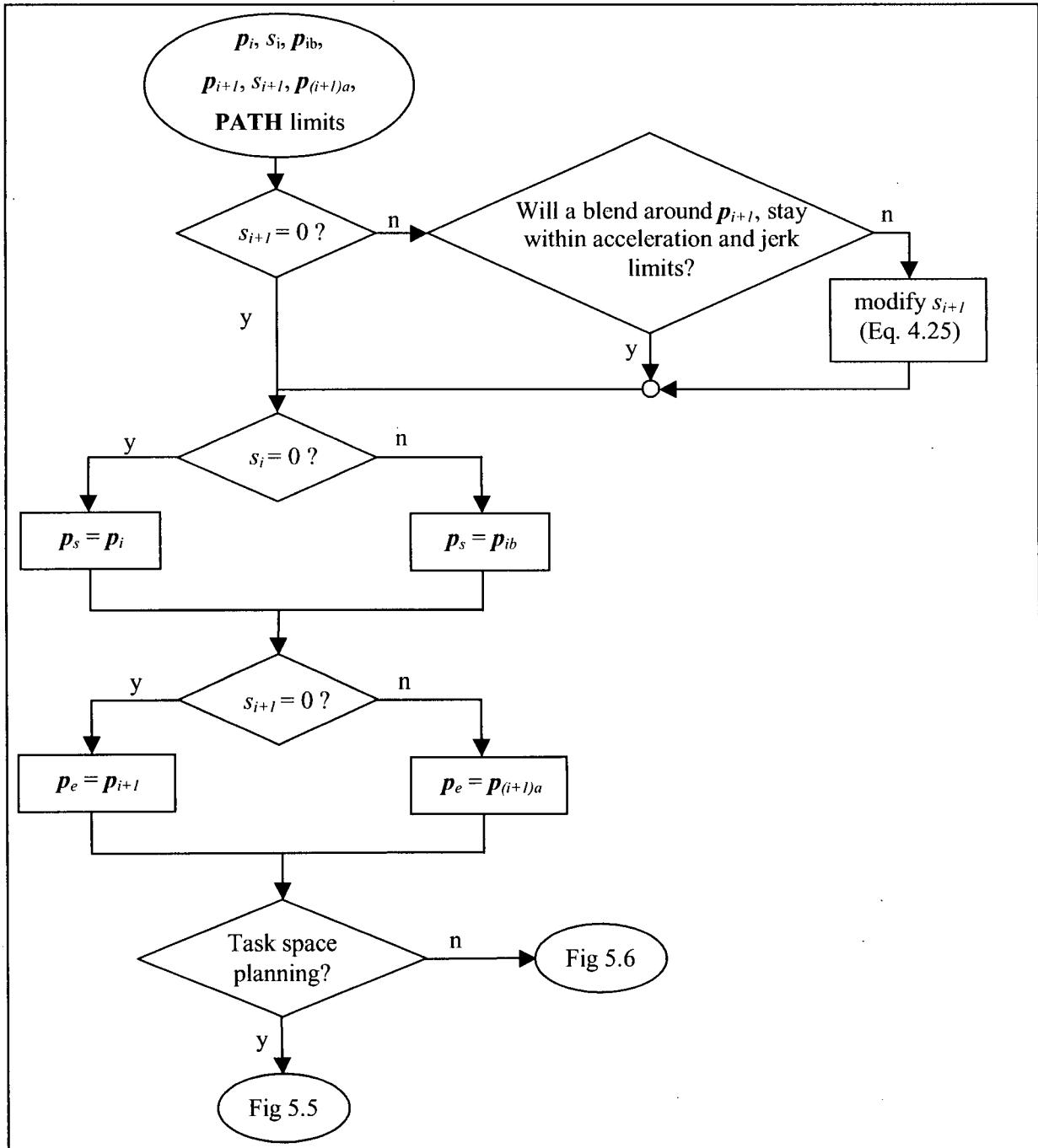


Figure 5.4. Generation of Trajectories Between Any Two Way-Points \mathbf{p}_i and \mathbf{p}_{i+1}

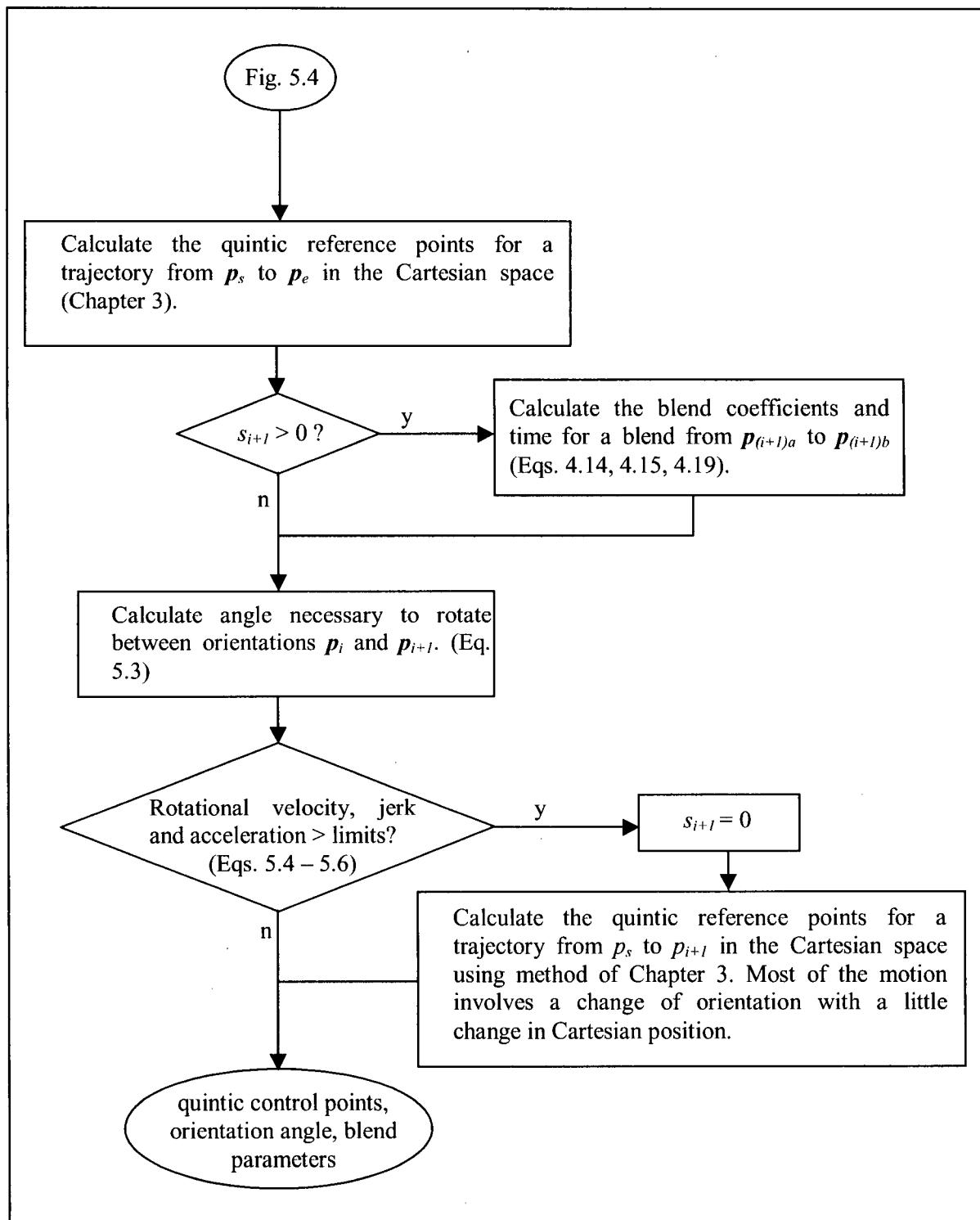


Figure 5.5. Cartesian Space Trajectory Reference Data Generation

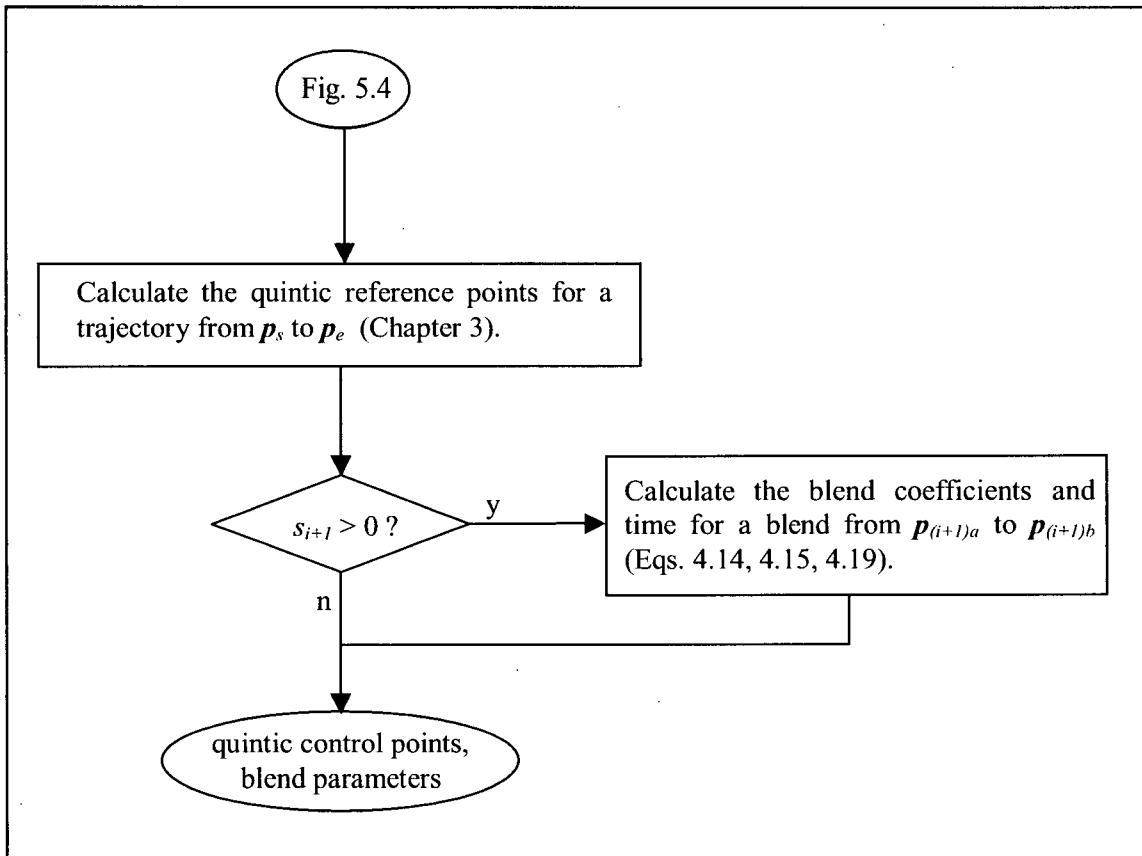


Figure 5.6. Joint Space Trajectory Reference Data Generation

5.4. Backward Planning Issues and Trajectory Generation Between N Way-Points

When way-points are specified, they are assigned a demanded speed that should be reached by the time trajectory arrives at the way-points, if possible. For example, a task space way-point can be specified (using roll, pitch, yaw angles) as $\mathbf{p} = (x, y, z, \psi_x, \psi_y, \psi_z, s\%)$, where $s\%$ varies between 0 and 100, 0 meaning \mathbf{p} is a stop point, 100 meaning that the machine should pass through the point at the manufacturer maximum allowable speed (MMAS) - often a “worst case” kinematic limit.

Given three way-points, \mathbf{p}_0 , \mathbf{p}_1 , and \mathbf{p}_2 , with demanded speeds $speed_0 = 0$, $speed_1 = s_1$, and $speed_2 = 0$, through which a trajectory must be fitted while respecting the kinematic limits, there are three possible scenarios (Figure 5.7):

- (a) The demanded speed, s_1 , at \mathbf{p}_1 is reached.
- (b) The demanded speed, s_1 , at \mathbf{p}_1 is not reached because of a distance limitation between points \mathbf{p}_0 and \mathbf{p}_1 .
- (c) The demanded speed, s_1 , at \mathbf{p}_1 is not reached because of a distance limitation between points \mathbf{p}_1 and \mathbf{p}_2 .

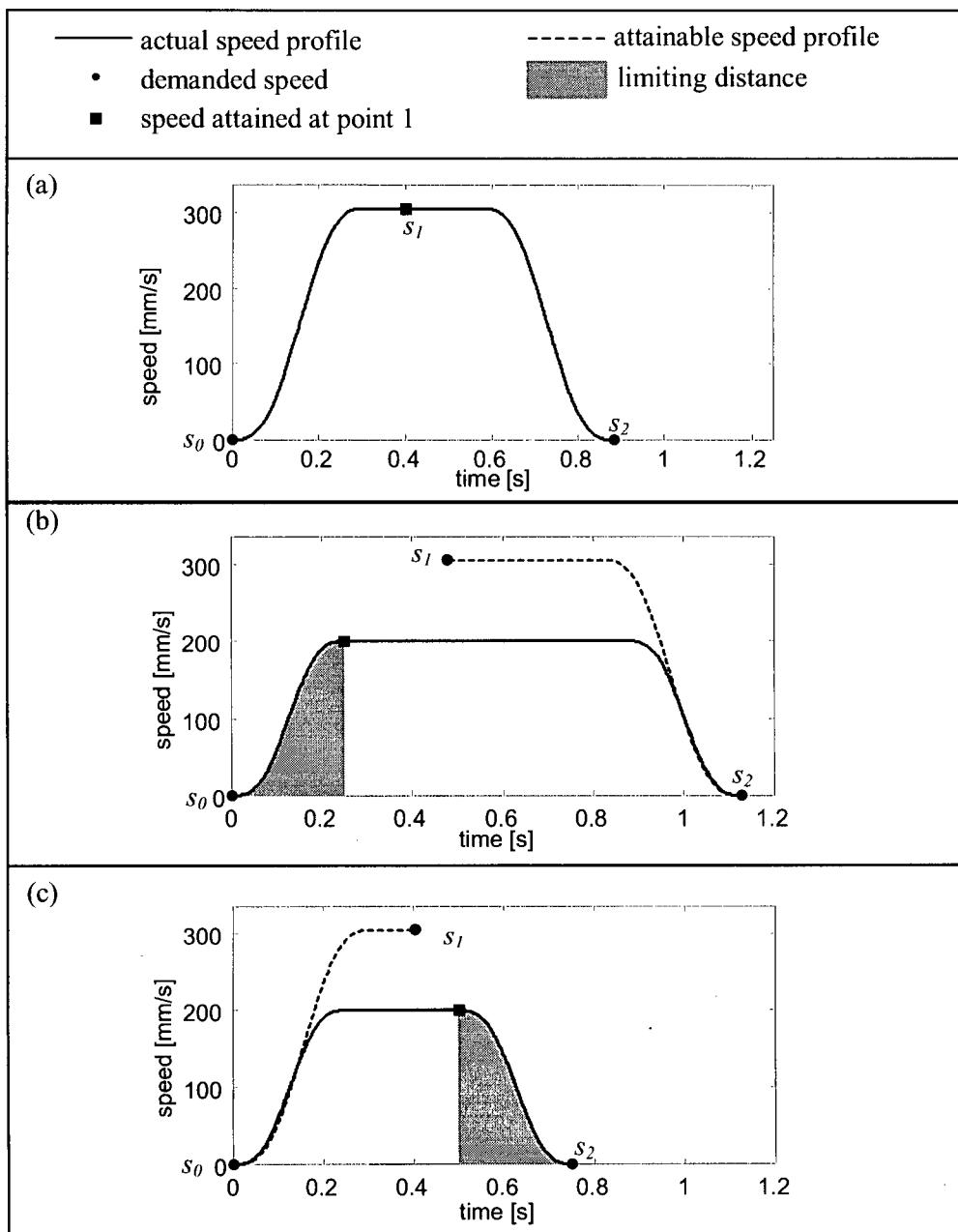


Figure 5.7. Three Possible Speed Profiles Between Three Way-Points.

The trajectory planning between the three points (p_0, p_1, p_2) , $N = 3$, is carried out as follows (Figures 5.8, 5.9):

1. Plan a trajectory between points p_0 and p_1 .

IF the demanded speed at point p_1 cannot be reached (case (b)),

Replace the demanded s_1 with the reachable s_1 .

2. Plan a trajectory between points p_1 and p_2 .

IF the demanded speed at point p_1 cannot be reached (since the zero speed condition at p_2 must be respected),

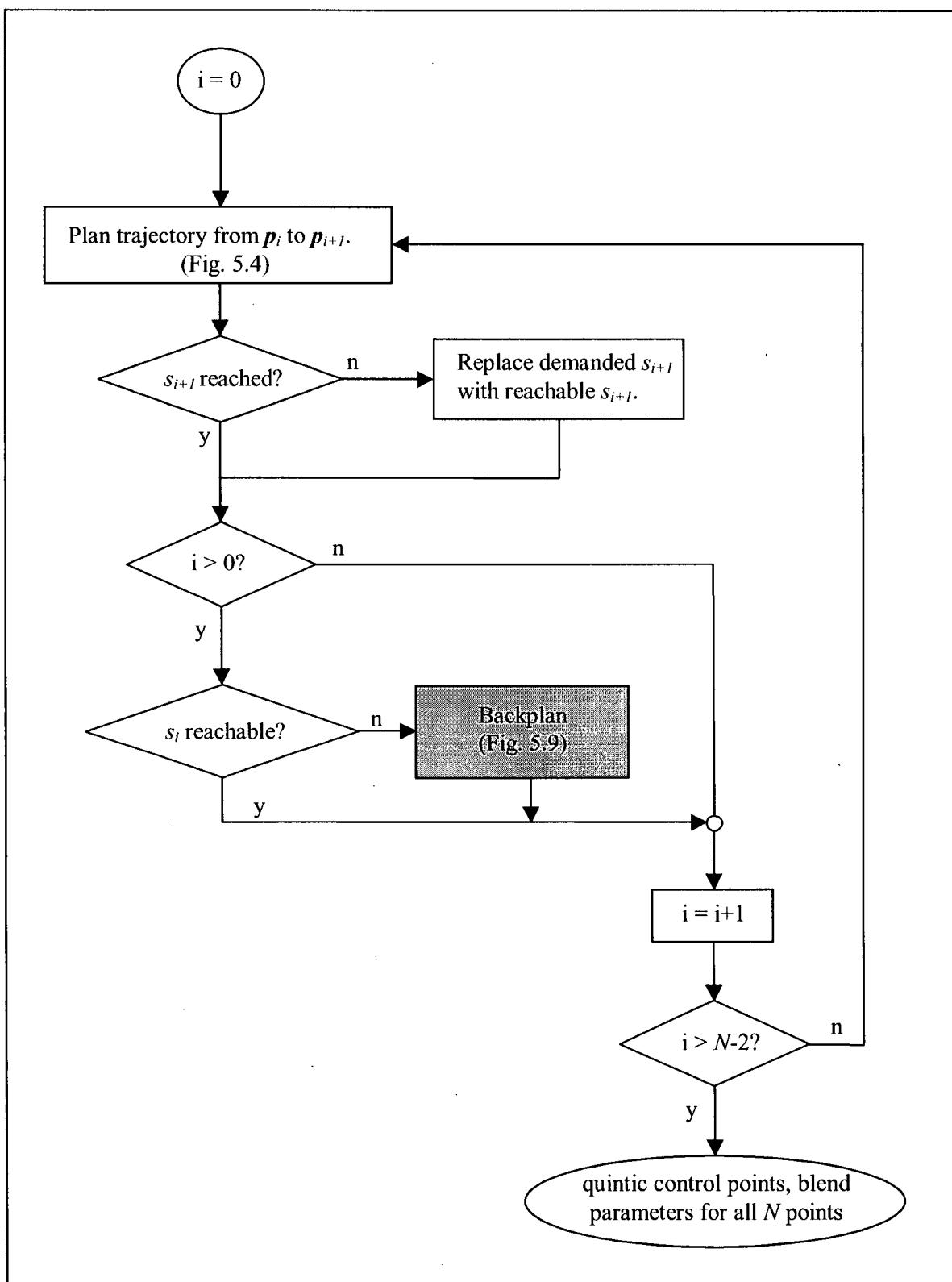
Replace the demanded s_l with the reachable s_l , and replan the trajectory between p_0 and p_l , with this new end speed. This approach is referred to as *backplanning*.

This trajectory planning method can be extended to any number of way-points. The planning is carried out from the first point to the last point, with backplanning done as necessary. Figure 5.8 and 5.9 show a flowchart of the trajectory planning method between N way-points. The calculated reference data (namely, the quintic control points and the blend parameters) are then used to generate trajectory points at every controller sampling interval (Chapter 6). It should be noted that by planning the trajectory based solely on the reference data, it is computationally simple to modify the trajectory when speed corrections are necessary. That is, only the control points and blend parameters must be modified.

5.5. Summary

In this chapter, a method for designing a smooth near time-optimal trajectory that passes through or close to many way-points, while respecting speed, acceleration and jerk limits, was described. The method is based on straight line motion in any n-dimensional space and the use of geometric blends for changing direction. The trajectory generation method described in Chapter 3 is used to generate the time law associated with the straight line paths. The geometric blends described in Chapter 4 are used to interconnect the straight-line paths.

Planning the trajectories is computationally simple. It requires only the computation of a series of reference data for each way-point to way-point motion: the quintic control points (a maximum of eight per way-point to way-point trajectory) and the blend parameters.

Figure 5.8. Trajectory Planning Between N Way-Points (Forward Planning)

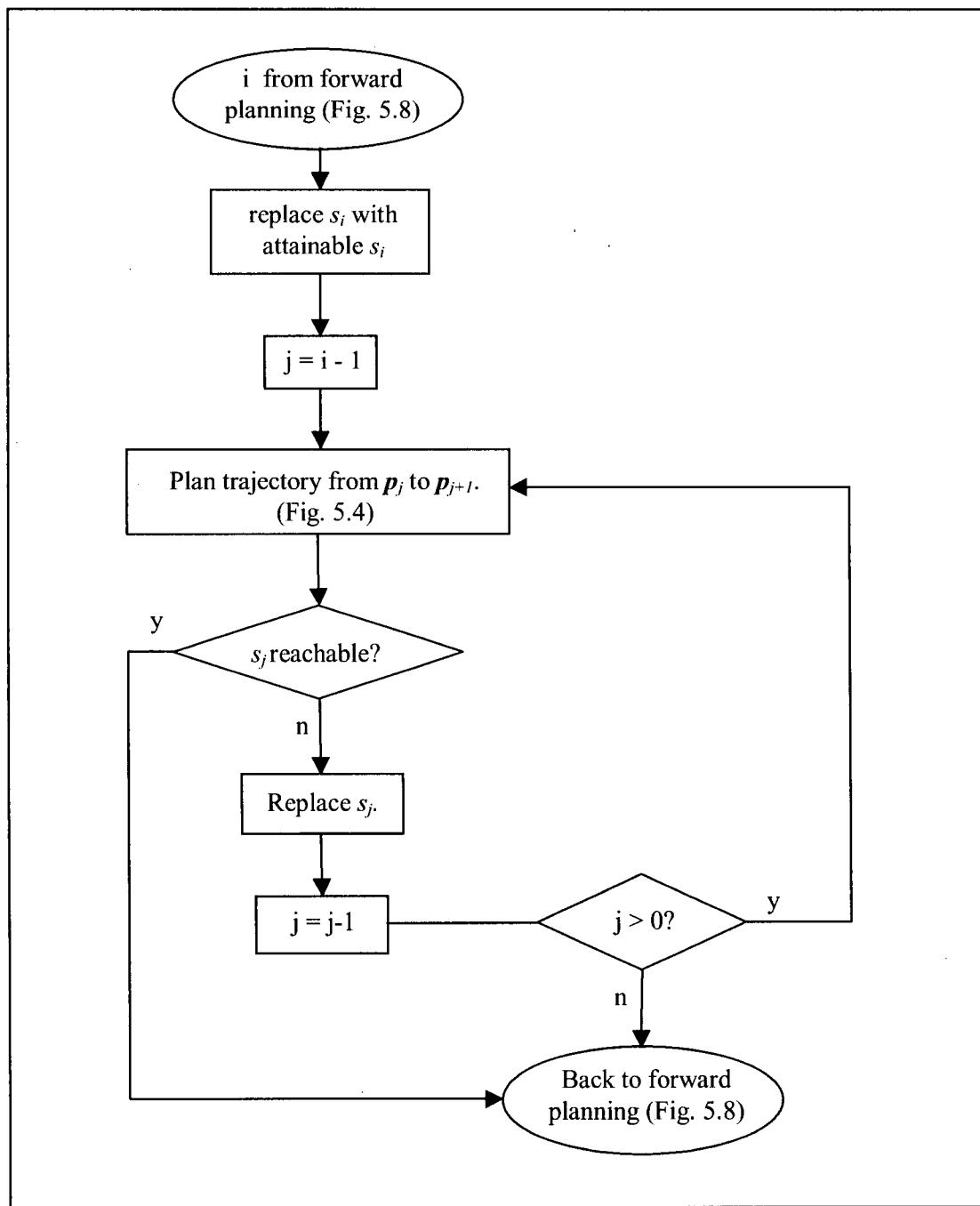


Figure 5.9. Backplanning

Chapter 6

Application to a Manipulator

In the overall trajectory planning algorithm there are three different cooperating systems (Figure 6.1): a low level, which calculates the quintic control points and blend coefficients (Chapter 5), a parser, which calculates the trajectory points corresponding to each sampling interval in task or joint space, and a high level, which coordinates the flow of information from the scheduler to the low level and then to the parser. The high level and parser functions discussed in this chapter are intended for an on-line manipulator system.

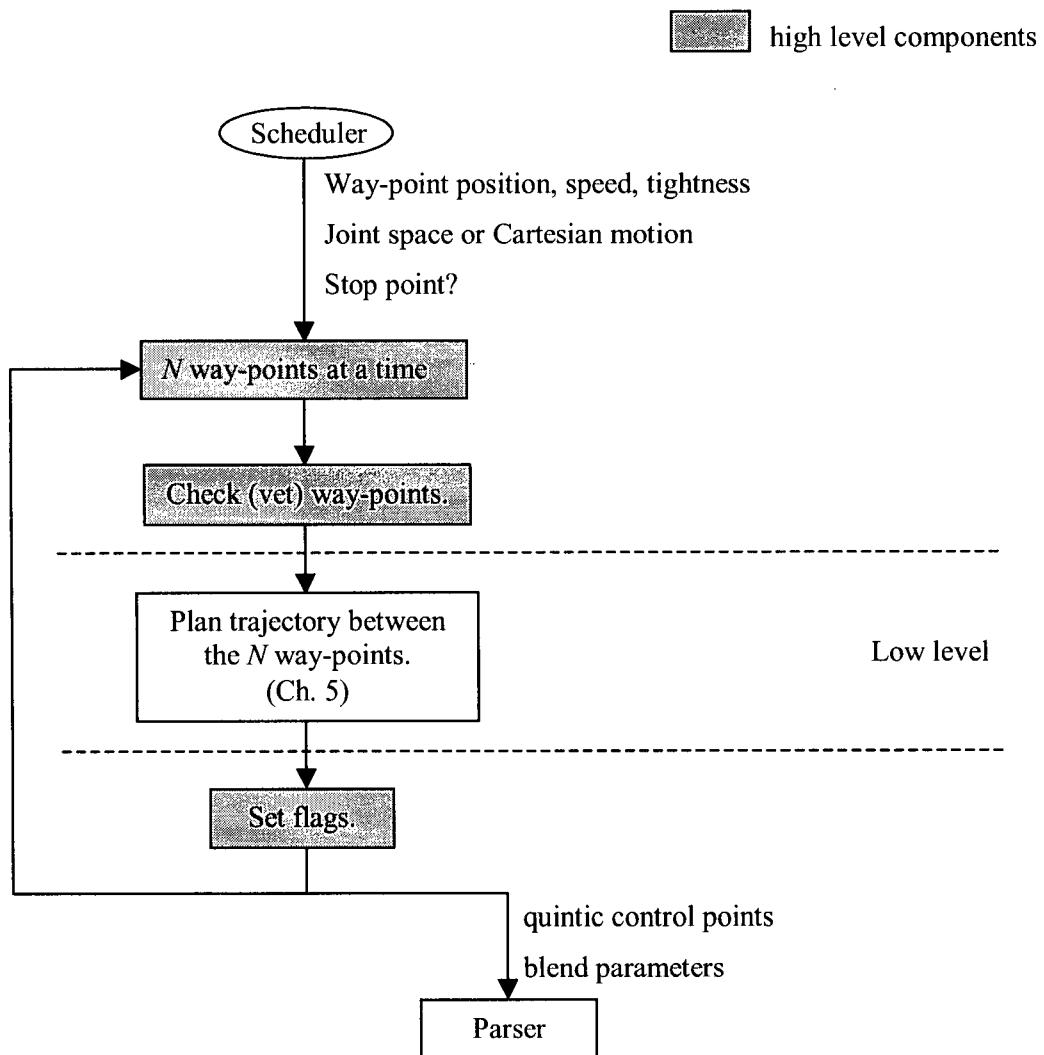


Figure 6.1. Interaction Between the Three Planning Units

6.1. Way-Points

The way-points are provided by a high level scheduler. This may be a scheduler program with an off-line set of pre-selected way-points, or an on-line workcell scheduling program. These way-points designate the manipulator path. Along with each way-point position, specified either in joint space or in task space, the scheduler must also indicate the speed the robot should reach at the way-point. This speed can be specified as a percentage of the maximum speed that the robot can attain, as determined by the manipulator manufacturer. The scheduler must also specify other properties related to the way-point, namely:

- the *tightness* associated with the way-point, i.e. how closely the way-point should be approached (cf. Chapter 4);
- whether the way-point is a stop point;

and those related to the motion that starts at the given way-point:

- whether the motion should be done in joint space or cartesian space;
- whether the wrist orientation should be maintained or allowed to vary. This property can only be specified in the case of a task space motion. It is important in the case where the robot must not tip its payload.

6.2. High Level Planner

When the high level planner receives way-points from the scheduler, it places them in a queue. The planner processes a maximum window of N way-points at a time. At the beginning of a planning sequence, the first N way-points are checked for scheduler inconsistencies, and a number of variables necessary for the low level trajectory planning are set. This process is referred to as vetting the way-points. Once vetting has been performed, the way-points are sent to the low level trajectory planning algorithm, and the trajectories between the entire window of N way-points are planned. The quintic control points and blend parameters associated with a certain number of those N way-points are then sent to the parser. The high level selects the succeeding way-points in the queue and the cycle recommences. This continues until no new way-points are available in the queue. This feature allows the trajectory to be computed on-line, since the time needed to compute the trajectory can be fixed by carefully selecting the window size, N . The following subsections explain the overall process in more detail.

6.2.1. Preliminary Considerations

Way-points can be delivered to the queue asynchronously. Therefore, the high level planner cannot predict the number of way-points it is likely to receive. When planning the trajectory between the current

window of N way-points, a zero speed condition is imposed by the algorithm on the N^{th} way-point to ensure that the manipulator will stop at this "last" way-point, in the event that no further way-points are delivered. In addition, if the manipulator completes the previously planned trajectory before the trajectory for the current window of N way-points is planned, the manipulator will be able to halt and wait for the computations to catch up. This case could occur when the N way-points are very close together, or when the number N is rather small. Thus, it is necessary to ensure that the planning window size, N , is large enough to avoid computation stalling. The algorithm-imposed zero speed condition is removed later in the planning, if possible, as described in Section 6.2.3. A point with a zero speed condition is called a *stop point*.

As mentioned above, the planning window size, N , should be chosen such that the planning can be done on-line. However, N must also be chosen in such a manner that it is possible to reach a significant speed when travelling between the N way-points (e.g. if $N = 1$, then each point would be a stop point, which would obviously lead to a slow trajectory).

6.2.2. Vetting

The vetting function is provided to check the scheduler way-point information. In the vetting function (Figure 6.2), the way-points are compared to verify that two consecutive way-points are not located "on top" of each other. In this case, the second of the two way-points is removed. Furthermore, if a way-point lies within the next way-point's *tightness* (encompassed point case, Figure 6.3), the second of the two way-points is eliminated. In reaching the first way-point, the second way-point's *tightness* area will be entered and, therefore, the second point can be considered to have been reached. If the *tightness* of two way-points overlap (overlapping *tightness* case, Figure 6.3), they are reduced proportionally such that they become tangent to each other. This is done so that it is possible to plan the two blends associated with the points. In the example provided in Figure 6.3, way-point p_2 is eliminated since p_1 lies within p_2 's *tightness*. The *tightness* of p_5 and that of p_6 are reduced proportionally such that they are tangent to each other. In some unusual encompassed point cases, it is possible that the resulting path will not enter an eliminated way-point's *tightness* area (Figure 6.4(a)). If such cases are of concern, the point should be treated as an overlapping *tightness* case, although this will result in a slower motion (Figure 6.4(b)). Once all point elimination has been performed, the last of the N way-points is set to be a stop point (algorithm-imposed zero speed condition) as explained in Section 6.2.1.

The vetting function also sets the variables related to the motion between two consecutive way-points. The motion information associated with the trajectory between two way-points is always assigned to the

first of the two way-points. The trajectory motion type is set to be either a straight line followed by a blend or a straight line followed by a stop point. This variable is used to determine which low level function is used to compute the trajectory quintic control points and blend parameters. As well, the maximum path speed between two way-points is set to be the maximum of the speeds associated with the way-point under consideration and the subsequent way-point.

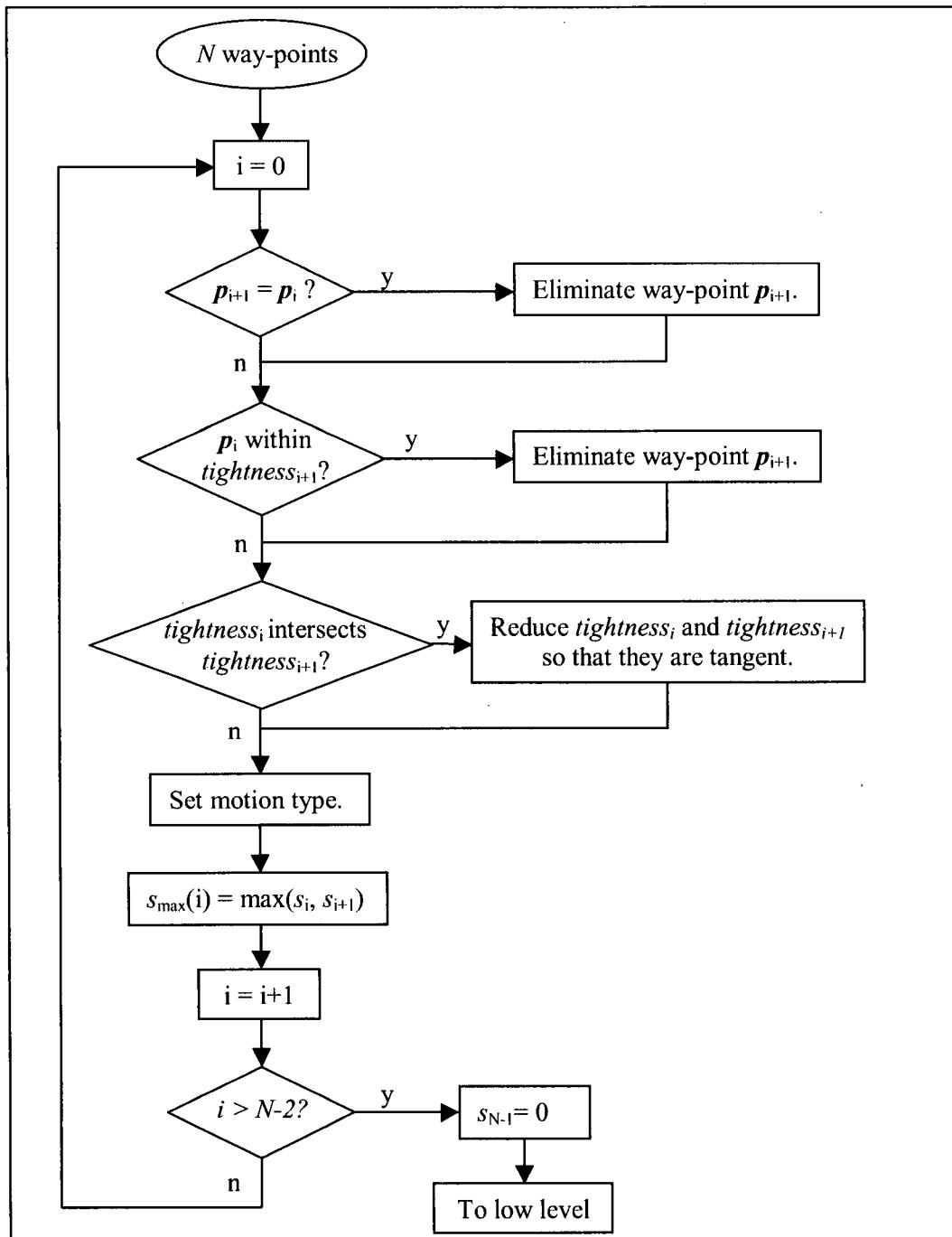


Figure 6.2. Vetting Procedure

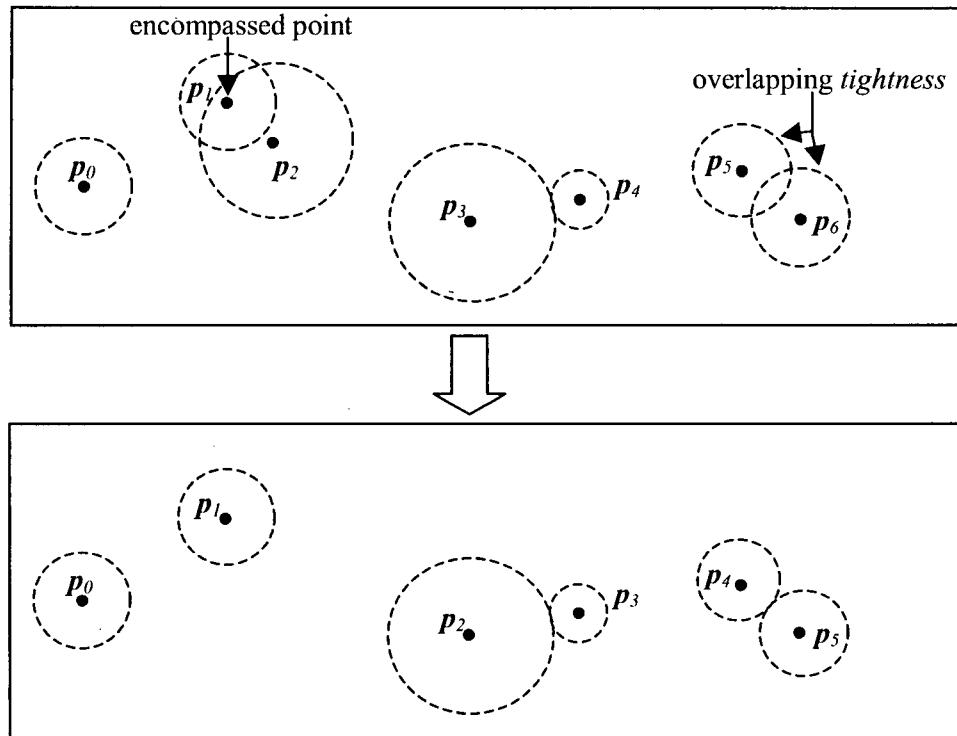
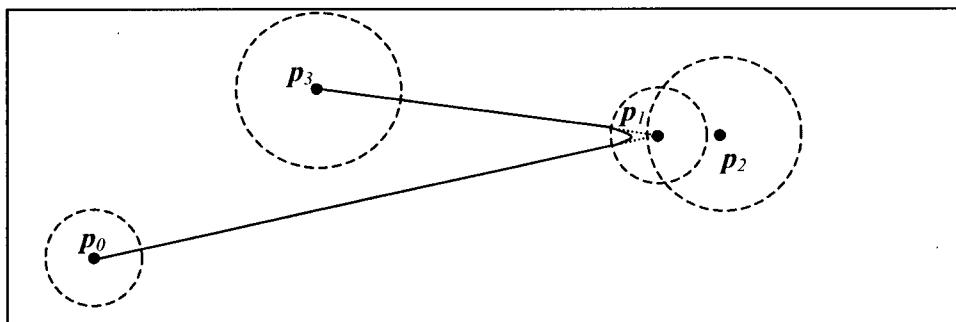
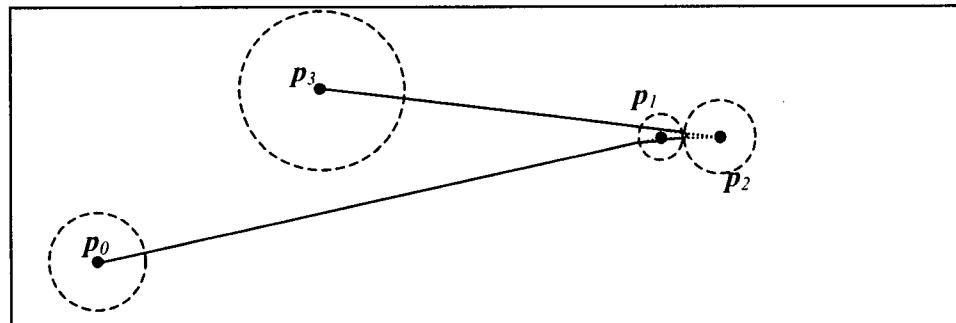


Figure 6.3. Elimination of Way-Points Due to *Tightness* Considerations



(a) Illustration of a case where the elimination of point p_2 would result in the path not entering the *tightness* area of point p_2 - encompassed point case.



(b) Suggested Solution: Reduce the *tightness* of way-points 1 and 2 proportionally - treat p_1 and p_2 as an overlapping *tightness* case.

Figure 6.4. Special Cases for Point Elimination in Vetting Algorithm

6.2.3. Trajectory Planning and Parser Points

Once the vetting has been performed, the trajectories are planned sequentially between each pair of way-points, as described in Chapter 5. It is then necessary to send the quintic control points and blend parameters to the parser. However, as mentioned in Section 6.2.1, a zero speed condition is imposed by the algorithm upon the last of the N way-points. Obviously keeping this restriction is not optimal, and it should be removed, if possible. This possibility was incorporated through the use of flags. The flags are also used to determine the number of planned trajectories that will be sent to the parser.

There are two flags: *Desirable*, and *Best*. A way-point is *Desirable* when the trajectory passes through this way-point at the speed requested by the scheduler. A way-point is *Best* when it is *Desirable* or when its speed is imposed by the previous way-point (i.e., the speed achieved is the highest possible given the circumstances). The *Best* flag indicates the latest way-point in the planning window that will not be affected by the addition of new way-points to the queue, or by the removal of the zero speed condition on the N^{th} way-point. Therefore, a way-point with an algorithm-imposed zero speed is never *Best*.

A way-point's flags are set once a trajectory between a pair of way-points has been calculated, and the corresponding back-planning performed. If the planning between the two way-points currently under consideration required backplanning of the previous way-points (Section 5.4), the *Best* and *Desirable* flags of the backplanned way-points are modified during the backplanning. An example is provided below, in Section 6.2.3.1.

Once planning in the window is completed, all W way-points preceding the last *Best* way-point in the window are passed to the parser since there is no way to improve the trajectories linking these way-points. If new points are available, the zero speed condition is removed from the last point in the window, and the next available W way-points in the queue are added to the window. This new list of way-points is processed by the high level. This approach provides an optimally planned trajectory since there are no algorithm-imposed stop points.

In the event that the time it will take the manipulator to reach the last of the W way-points is smaller than the time it will take the planner to process the next window of N way-points, it is necessary to provide Y ($Y > W$) way-points to the parser. In this case, the $(Y+1)^{\text{th}}$ way-point is marked as *Best* and its speed cannot be modified. Then Y new way-points are accepted from the queue. This obviously does not provide the most optimal trajectory, but it allows for continued on-line computation of the trajectory.

6.2.3.1. Flag Setting Example

In this illustrative numerical example, there are eight way-points in the window ($N = 8$) with desired speeds given in Table 6.1. The way-points are designated as p_1, p_2, \dots, p_8 , their corresponding speeds and flags as, respectively, s_i , $Best_i$, $Desirable_i$, etc. The speeds are given as a percent of the manufacturer speed limit. The planner algorithm proceeds as follows:

1. The speed at p_8 is set to zero (zero speed condition).
2. $Best_1$ is set to TRUE.
3. Forward plan from p_1 to p_2 : $s_2 = 40\%$ is reached, therefore, $Desirable_1$ = TRUE.
4. Forward plan from p_2 to p_3 : $s_3 = 55\%$. $Best_2$ = TRUE (p_2 's speed was imposed by p_1 , not p_3) and $Desirable_2$ = TRUE.
5. Forward plan from p_3 to p_4 : s_3 reduced to 25% to achieve $s_4 = 20\%$, due to distance limitations. Therefore, $Best_3$ = FALSE (p_3 's speed is imposed by p_4) and $Desirable_3$ = FALSE.
 - a. Backplan: from p_2 to p_3 : s_2 reduced to 35% to allow $s_3 = 25\%$. $Best_2$ = FALSE, $Desirable_2$ = FALSE.
 - b. Backplan: from p_1 to p_2 to allow $s_2 = 35\%$. No change in flags for p_1 .
6. Forward plan from p_4 to p_5 : $s_5 = 30\%$. Therefore, $Best_4$ = TRUE and $Desirable_4$ = TRUE.
7. Forward plan from p_5 to p_6 : $s_6 = 45\%$. Therefore, $Best_5$ = TRUE and $Desirable_5$ = TRUE.
8. Forward plan from p_6 to p_7 : $s_7 = 55\%$. Therefore, $Best_6$ = TRUE and $Desirable_6$ = FALSE.
9. Forward plan from p_7 to p_8 : s_7 reduced to 40% to allow $s_8 = 0$. Therefore, $Best_7$ = FALSE (p_7 's speed is imposed by p_8) and $Desirable_7$ = FALSE.
 - a. Backplan: from p_6 to p_7 : $s_6 = 45\%$. No change in flags.

The final values of the way-point flags are shown in Table 6.1. The information related to the trajectories linking p_1 to p_6 can be sent to the parser since even if the zero speed limitation is removed from p_8 , these trajectories will not be affected. Point p_6 will become the new first way-point in the window of N way-points, and five new way-points will be accepted ($W = 5$). When the new list is compiled, s_8 will be reset to the scheduler's demanded speed, and the trajectory segments between p_6 and p_8 will be recalculated.

way-point	1	2	3	4	5	6	7	8
demanded speed (%)	10	40	60	20	30	50	60	50
speed attained (%)	10	35	25	20	30	45	40	0
Best	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	FALSE	-
Desirable	TRUE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE	-

Table 6.1. Way-Point Speeds and Flags at the End of Planning

6.3. Parser

The function of the parser is to compute the n-dimensional reference signal corresponding to each sampling interval. The parser works in parallel with the high level and low level planners. That is, the parser computes trajectory points for each sampling interval from the previously planned trajectories, while the high and low level planners are generating new quintic control points and blend coefficients to be fed to the parser.

As mentioned in Section 5.1.1, a motion between two way-points, p_u and p_v , can be composed of various segments depending on the speeds of these two way-points:

- If both way-points are stop points, the motion is a straight line.
- If only p_u is a stop point, the motion is composed of a straight line followed by a blend.
- If only p_v is a stop point, the motion is composed of a blend followed by a straight line.
- If neither way-point is a stop point, the motion is composed of a blend, followed by a straight line and another blend.

Therefore, there are two different types of parsing corresponding to the components of the motions being generated: straight-line parsing and blend parsing. The parser recognizes which segments are present in the motion between two way-points and generates the appropriate controller commands based on the time data provided by the high level (i.e. a segment with zero time is not present in the motion).

The routines outlined in the next three sections are used to calculate the n-dimensional reference trajectory corresponding to the sampling intervals.

6.3.1. Straight Line Parsing

For a straight line, it is first necessary to determine the quintic coefficients, b_5, \dots, b_0 (Eq. 6.1) based on the quintic control points provided by the planner. By taking the first and second derivatives of Eq. 6.1 with respect to time, six equations, corresponding to the start and end points of the straight line, can be derived.

$$g(t) = b_5 t^5 + b_4 t^4 + b_3 t^3 + b_2 t^2 + b_1 t + b_0, \quad (6.1)$$

where t is the time, the path parameter.

Since the position, speed and acceleration are known at the start and the end of each quintic, the system of six equations and six unknowns (the quintic coefficients) can be solved. This system is shown in Eq 6.2:

$$\begin{bmatrix} 1 & t_1 & t_1^2 & t_1^3 & t_1^4 & t_1^5 \\ 0 & 1 & 2t_1 & 3t_1^2 & 4t_1^3 & 5t_1^4 \\ 0 & 0 & 2 & 6t_1 & 12t_1^2 & 20t_1^3 \\ 1 & t_2 & t_2^2 & t_2^3 & t_2^4 & t_2^5 \\ 0 & 1 & 2t_2 & 3t_2^2 & 4t_2^3 & 5t_2^4 \\ 0 & 0 & 2 & 6t_2 & 12t_2^2 & 20t_2^3 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{bmatrix} = \begin{bmatrix} p_1 \\ v_1 \\ a_1 \\ p_2 \\ v_2 \\ a_2 \end{bmatrix}, \quad (6.2)$$

where t_1, p_1, v_1, a_1 are the time, position, speed and acceleration at the start of the quintic, similarly, t_2, p_2, v_2, a_2 are the time, position, speed and acceleration at the end of the quintic, and b_5, \dots, b_0 are the unknown quintic coefficients. To simplify this system, the initial time parameter, t_1 , is set to zero for each quintic trajectory segment. The top of the matrix becomes diagonal and the system is reduced to a system of three equations and three unknowns, which is suitable for solving on a DSP-based trajectory planner. There are no transcendental calculations involved in the calculation of the quintic control points needed to describe the trajectory between two way-points. The limiting case, computationally, is that described in Section 3.2, case B, where seven quintics may be necessary. In this case, a maximum of 480 floating point operations is required to determine the quintic control points.

Once the quintic coefficients have been determined, it is possible to calculate the n-dimensional reference signal corresponding to each sampling interval by simply incrementing the time in Eq. 6.3.

$$\mathbf{p}(t_k) = \mathbf{p}_s + \frac{(\mathbf{p}_e - \mathbf{p}_s)}{\|\mathbf{p}_e - \mathbf{p}_s\|} g(t_k), \quad (6.3)$$

where

$$t_k = k \cdot \Delta t, \quad 0 \leq k \leq \frac{t_2}{\Delta t}, \quad (6.4)$$

\mathbf{p}_s and \mathbf{p}_e are respectively the initial and final position vectors of the straight line and Δt is the sampling interval. $g(t)$ starts at zero and ends at $\|\mathbf{p}_e - \mathbf{p}_s\|$.

6.3.2. Task Space Parsing

When the trajectory is defined in task space, the Cartesian portion of the trajectory is generated based on Eq. 6.3. The orientation portion of the trajectory is generated using a different method because the orientation axes must be orthogonal (i.e. orientation is not a vector). Eq. 6.1 is used to generate a quintic that rotates the orientation about a fixed axis in space, as explained in Section 5.1.2. Since the quintic provides an angle corresponding to each sampling instant, it is necessary to convert this axis-angle representation to the original orientation specification at each increment. This is done using quaternion representation. That is, the relative axis-angle representation is converted to quaternion form. The

necessary matrix manipulations are done in quaternion representation to obtain the orientation in terms of the world reference frame, and then converted to rotation matrix form, providing a form that can easily be mapped to the type of coordinates being used by the controller. This algorithm detailed in Appendix G.

6.3.3. Blend Parsing

To generate the n-dimensional trajectory corresponding to a blend, the blend parameters, \mathbf{b}_1 , \mathbf{b}_2 , \mathbf{m}_1 , \mathbf{m}_2 and τ , calculated during the low-level planning (Section 4.2.1), are used in Eq. 6.5:

$$\mathbf{x}(\sigma_k) = \mathbf{x}_1(\sigma_k) + \alpha(\sigma_k)(\mathbf{x}_2(\sigma_k) - \mathbf{x}_1(\sigma_k)) - \kappa\beta(\sigma_k)\mathbf{v}_d, \quad (6.5)$$

where

$$\sigma_k = \frac{t_k}{2\tau}, \quad (6.6)$$

$$t_k = k \cdot \Delta t, 0 \leq k \leq 2\tau, \quad (6.7)$$

$$\mathbf{x}_1(\sigma_k) = \mathbf{b}_1 + \mathbf{m}_1 \sigma_k, \quad (6.8)$$

$$\mathbf{x}_2(\sigma_k) = \mathbf{b}_2 + \mathbf{m}_2 \sigma_k, \quad (6.9)$$

$$\alpha(\sigma_k) = 6\sigma_k^5 - 15\sigma_k^4 + 10\sigma_k^3, \quad (6.10)$$

$$\beta(\sigma_k) = \sigma_k^6 - 3\sigma_k^5 + 3\sigma_k^4 - \sigma_k^3, \quad (6.11)$$

$$\mathbf{v}_d = \mathbf{m}_2 - \mathbf{m}_1, \quad (6.12)$$

and

$$\kappa = 15/2. \quad (6.13)$$

6.4. Summary

In this chapter, the issues related to the implementation of the trajectory generation method described in Chapter 5 in a manipulator system were discussed. Such an implementation requires a high level that controls the flow of information between the scheduler and the planning algorithm, as well as to the parser. The parser is the unit that computes the reference signal corresponding to each sampling interval. A flag is used to determine which points to pass from the planning algorithm to the parser with the aim of providing the fastest trajectory possible between the entire queue of way-points given the imposed kinematic limits and the time required to plan the window of way-points.

Chapter 7

Experimental Results

In this chapter, the quintic-based trajectory generation algorithm is compared to an industrial trajectory generator [6], and to the smooth, path-constrained time-optimal motion (SPCTOM) technique [14]. The first comparison highlights the improvements in the trajectories as compared to the industrial technique. These comparative approaches provide a good comparison metric and were readily available for testing.

7.1. Experimental Setup

The experimental setup used to test the trajectories is a CRS A465 6-dof manipulator. The manipulator is controlled with a C500C CRS controller. Two different trajectory planning algorithms were available for use with the C500C controller. The first is the algorithm previously used by CRS and has no jerk limitations. It is referred to hereafter as the industrial planner. The second is an industrial implementation on the C500C of the algorithms developed in this thesis and licensed by the University of British Columbia to CRS Robotics Corp. Herein, it is referred to as the quintic planner. The experiments compare the industrial and quintic trajectory planners over a number of different metrics. The encoder feedback is read from a debugging header in the C500C controller using an open architecture real-time operating system (ORTS) [21].

In addition, the trajectory generation method proposed herein is compared to the SPCTOM trajectory using a SCORBOT ER-VII 5-dof manipulator in the IAL at UBC [14]. For this experiment, the robot is controlled by a TMS320C32 DSP board interfaced with two MFIO-3B cards that each control three axes. ORTS is used to send the control signals to the robot actuators based on the pre-planned trajectories. A tuned discrete PID controller is used to control the manipulator. In this case, only the positional degrees of the manipulator were controlled. The Denavit-Hartenberg (DH) parameters for the SCORBOT ER-VII and the A465 are provided in Tables 7.1 and 7.2 respectively. The experimental setup is shown in Figure 7.1.

Link	θ [rad]	d [mm]	a [mm]	α [rad]
1	0	358.5	50.0	$-\frac{\pi}{2}$
2	0	-37.0	300.0	0
3	0	0.0	250.0	0

Table 7.1. Denavit-Hartenberg Parameters for the Positional dof of the SCORBOT ER-VII

Link	θ [rad]	d [mm]	a [mm]	α [rad]
1	0	330.2	0.0	$-\frac{\pi}{2}$
2	0	0.0	304.8	0
3	0	0.0	0.0	$\frac{\pi}{2}$
4	0	330.2	0.0	$-\frac{\pi}{2}$
5	0	0.0	0.0	$\frac{\pi}{2}$
6	0	76.2 + tool transform	0.0	0

Table 7.2. Denavit-Hartenberg Parameters for the CRS A465 Manipulator

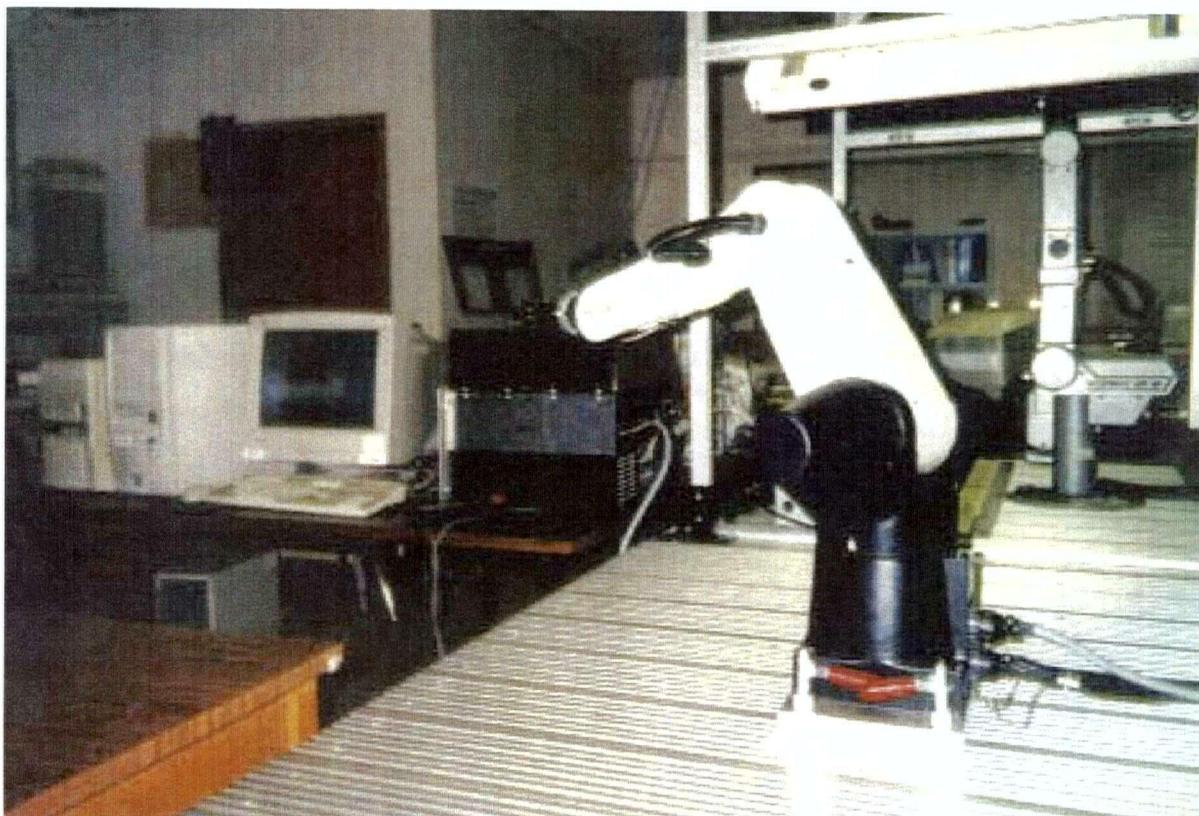


Figure 7.1. Experimental Setup (CRS A465 manipulator in the foreground; SCORBOT ER-VII manipulator in the background)

7.2. Comparison to a Typical Industrial Trajectory Generation Method

In this section, comparisons between the quintic planner and the industrial planner are made. Four exemplar types of motion are presented in this section, namely: (1) task-space stop point to stop point motion, (2) task-space multiple way-point motion, (3) joint-space stop point to stop point motion, (4) joint-space multiple way-point motion. The kinematic limits of the CRS A465 manipulator are given in Tables 7.3 and 7.4. Task space rotational jerk limitations were not included in the CRS implementation of the quintic planner. As mentioned, jerk limitations were not be implemented on the industrial planner.

The motions were run on the CRS A465 manipulator, five times each, at 10 different “demanded” path speeds for a total of 50 data sets per motion for each planner. The CRS system requires the demanded motion speed to be input as a percentage of the manufacturer maximum allowable speed (MMAS). The first five trials for each motion were carried out with a demanded speed of 10% of the MMAS. The demanded path speed was then incremented by 10%, up to the MMAS, and five trials were run at each increment. Since the same control laws (PID) are used for the industrial and quintic planner, any changes in path tracking or motion time for similar motions are due to the nature of the trajectories generated by the planner.

The experimental data provided in this section are averages reported to the appropriate accuracy based on two standard deviations of the data. Only the actuator encoder feedback was available for analysis. This precluded examination of the jerk profiles, since the jerk was indistinguishable from the noise of the signal. In addition, the motion times reported are the implemented motion times. That is, once the encoder speed reached a certain level (distinguishable from the noise of the signal), the motion was deemed to have commenced. The motion was deemed to have ended when the speed dropped below the signal noise level. Thus, the actual motion times can be compared between planners, but they are not identical to the motion times provided by the reference signals. The results provided in this section are a selection of the results available. The complete comparative results can be found in Appendix H, including an analysis of the relative improvements achieved using the quintic planner approach.

	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6
Speed (MMAS) [rad/s]	π	π	π	$\frac{96\pi}{101}$	$\frac{96\pi}{100}$	$\frac{96\pi}{101}$
Acceleration (MMAA) [rad /s²]	4π	4π	4π	$\frac{800\pi}{101}$	$\frac{800\pi}{100}$	$\frac{800\pi}{101}$
Jerk [rad /s³]	320π	320π	320π	$\frac{64000\pi}{101}$	$\frac{64000\pi}{100}$	$\frac{64000\pi}{101}$

Table 7.3. Joint Kinematic Limits for the A465

	Cartesian	Rotational
Speed (MMAS)	1016 mm/s	2 rad/s
Acceleration	2540 mm/s ²	10 rad/s ²
Jerk	81280 mm/s ³	-

Table 7.4. Task Space Kinematic Limits for the A465

7.2.1. Task-Space Stop Point to Stop Point Motion

In the straight-line task-space motions considered here, it is desired that the end effector accurately track the specified path. Therefore, the deviation of the end effector from the straight line is of importance, as well as the motion time. Since it is desired that the motions being compared in task space result in relatively similar joint paths to provide a proper basis for comparison, two types of experiments were carried out: motions that comprised purely positional changes and a motion that provided a change in task space orientation but no change in end effector position. To allow consistent comparison, it is necessary to separate these two types of motions since the industrial and quintic planner plan the orientation trajectories differently.

In the positional task-space experiments the motion times and the path deviations (shortest distance from each data point to the desired straight-line path) along the trajectory are compared, along with the maximum occurring path deviation. In the orientation experiment, the motion times, as well as the smoothness of the motions, are compared.

7.2.1.1. Experiment 1: Task-Space Straight Line with Constant Orientation

Figure 7.2 shows a comparison of the industrial and quintic task-space trajectories, as tracked by the CRS controller, along the straight-line path between the two points given in Table 7.5. Both planners generated a straight line reference signal. However, because of the way that the quintic task space trajectory is shaped, the controller is able to more accurately track the desired path at all demanded speeds. Table 7.6 gives a summary comparison of the deviations from the straight line path for both planners at various demanded speeds (refer to Section H.1.1, Appendix H for the complete results). In this example, the average and maximum path deviations obtained using the quintic planner with the demanded path speed equal to the MMAS (1016 mm/s) are better than those provided by tracking of the industrial planner trajectory at a demanded speed of 102 mm/s (10% MMAS). Both planners exhibit an increase in average and maximum path deviation as the maximum allowable path speed is increased.

The quintic planner generates a faster trajectory when the demanded path speed is below 406 mm/s (40% MMAS). The industrial planner accelerations do not reach the manufacturer maximum allowable

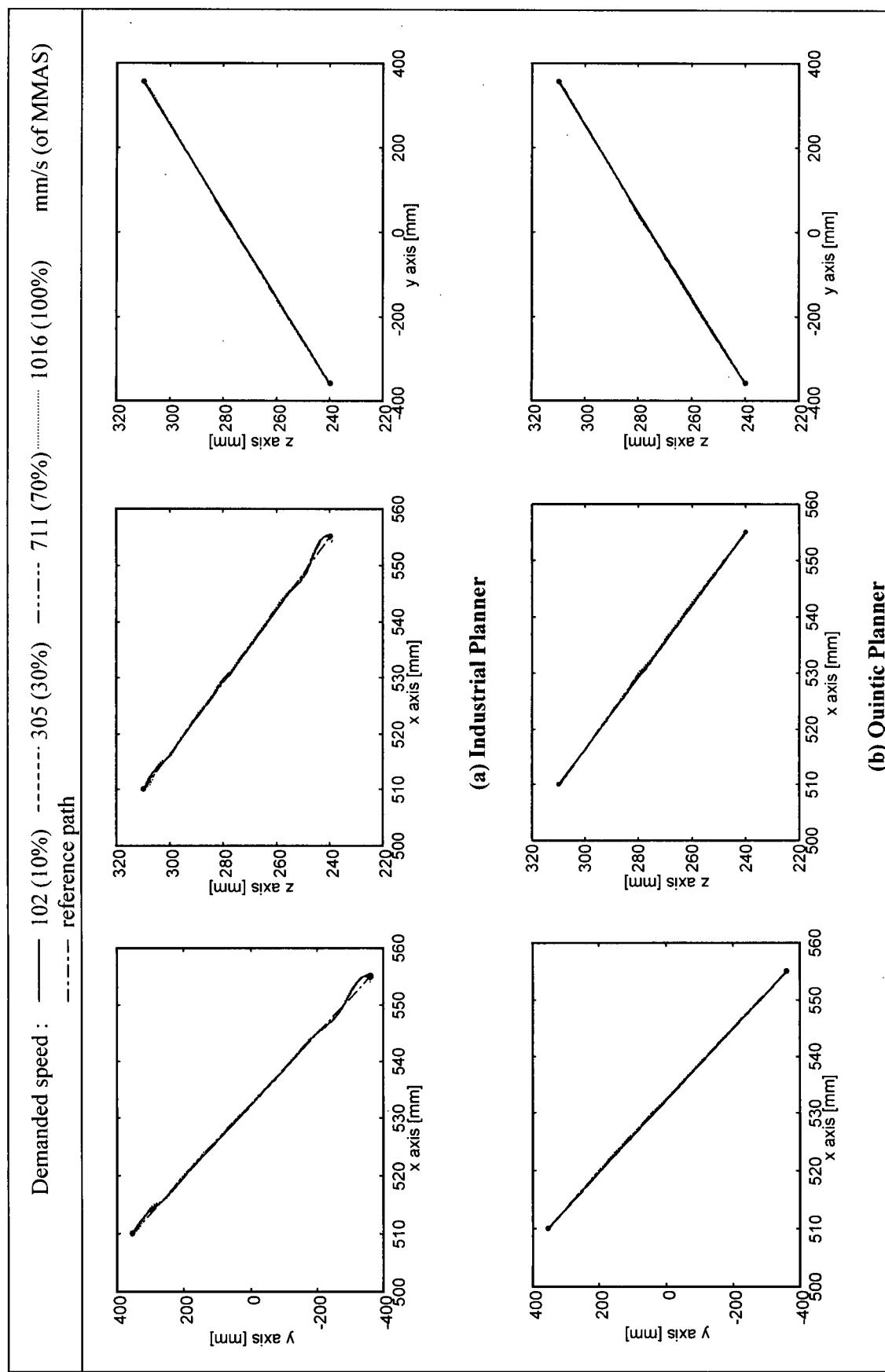


Figure 7.2. Comparison of Tracking Performance of Task Space Experiment 1

acceleration limit (MMAA) at these speeds. As the demanded path speeds increase, the industrial planner increases its maximum path acceleration until it eventually violates the MMAA by a significant margin (cf. Figures 7.3(a) and 7.5(b)). One can note that at 100% maximum speed, the industrial planner demands too high an acceleration and the resulting speed profile exhibits oscillations (Figure 7.3(a)). The quintic planner, on the other hand, requires a consistent maximum acceleration level for all desired path speeds, and the acceleration profile generated does not significantly violate the MMAA even at the maximum speed (Figures 7.3(b) and 7.5 (b)).

Point	x [mm]	y [mm]	z [mm]	yaw [rad]	pitch [rad]	roll [rad]
1	510	355	310	0.165	0.0731	-0.0113
2	555	-360	240	0.165	0.0731	-0.0113

Table 7.5. Coordinates of the Straight Line Path Start and End Points for Task Space Experiment 1 (tool transform of 10.5 mm in z-direction)

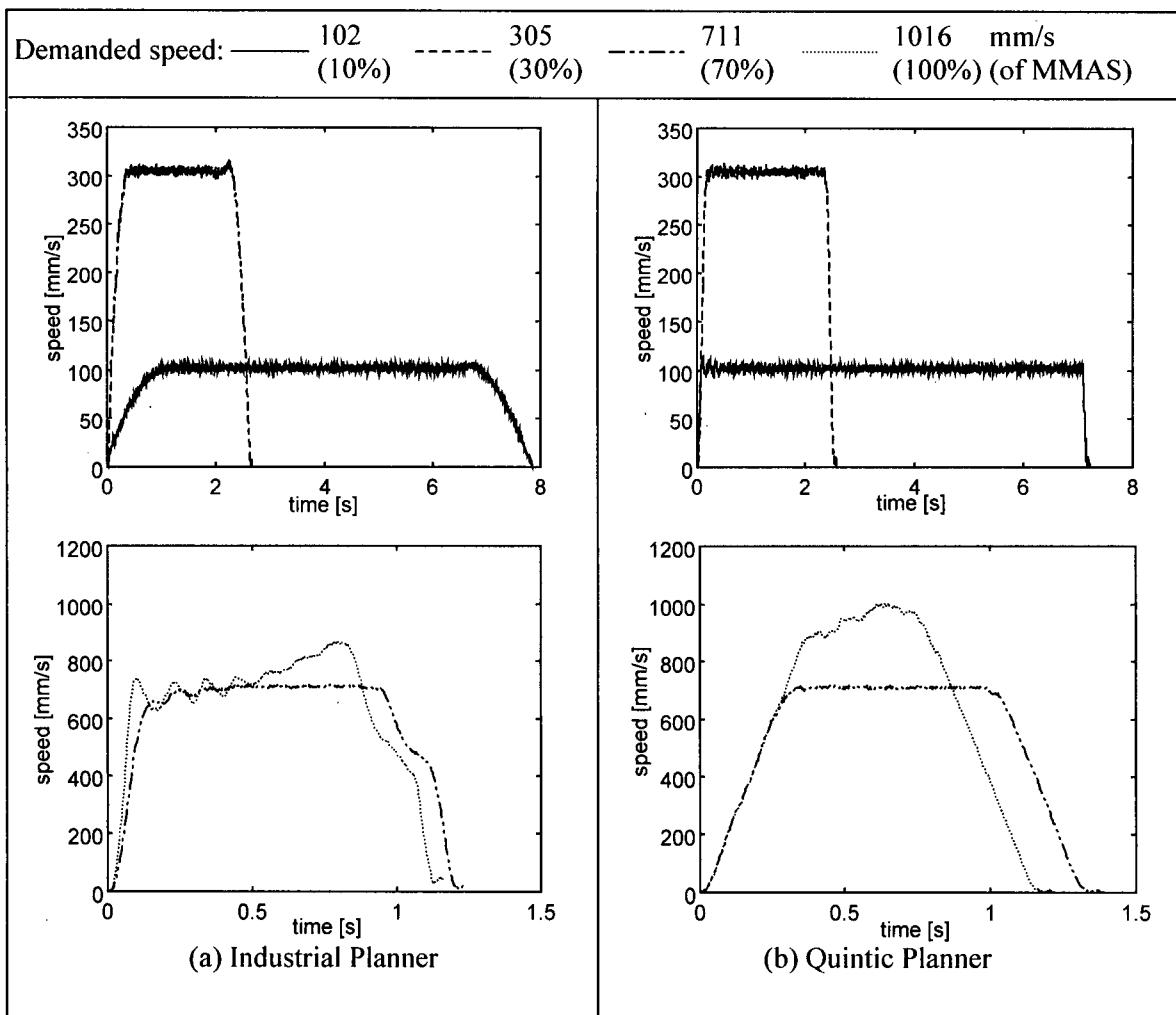


Figure 7.3. Comparison of Speed During Motion of Task Space Experiment 1

In Figure 7.4, the path deviation plots for two different demanded path speeds are shown with respect to time. Comparison of the plots in Figure 7.4 to the acceleration profiles of Figure 7.5 demonstrates the role of the acceleration and jerk limits in path tracking. An increase in path deviation is experienced at the beginning and end of each motion due to the non-zero acceleration components, which increase the dynamic load on the system. The path deviations are larger when the industrial planner is used. A comparison of Figures 7.4(a) and 7.5(a) shows that these larger path deviations are related to the jerk profiles since, in this case (at 10% MMAS), the industrial planner maximum acceleration and deceleration are far below those generated by the quintic planner.

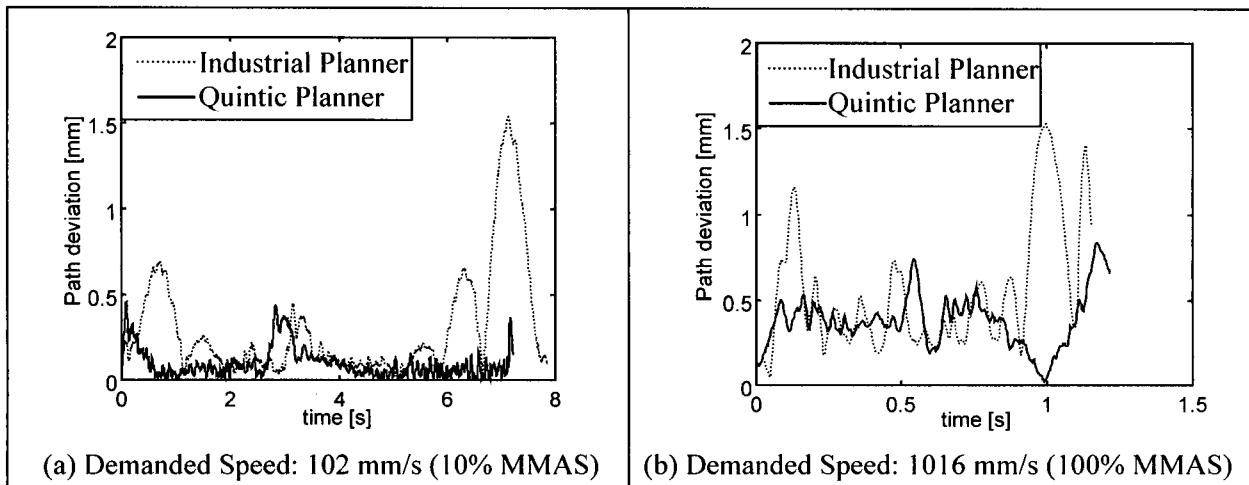


Figure 7.4. Comparison of Path Deviations of Task Space Experiment 1

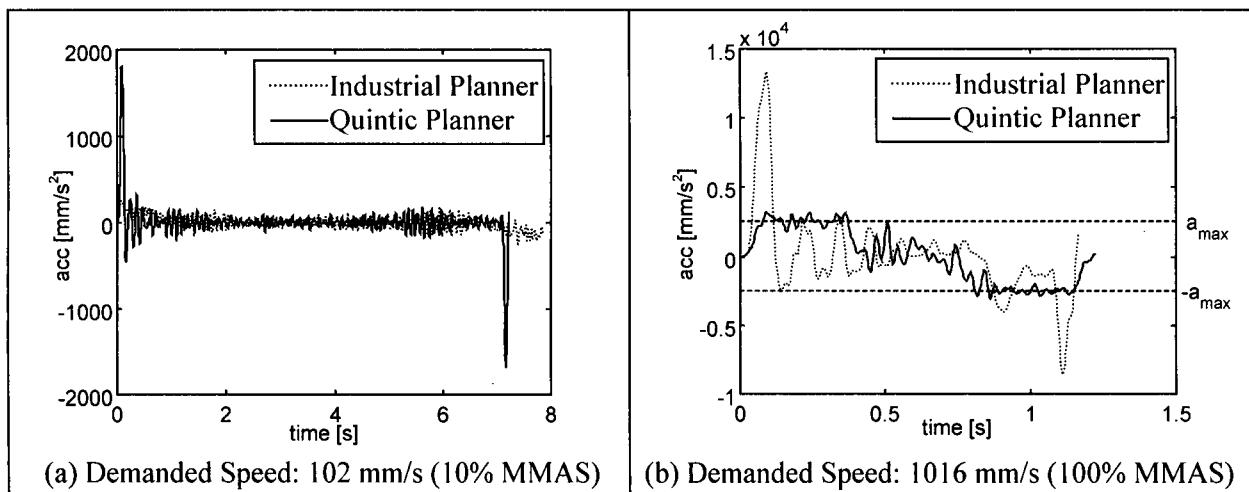


Figure 7.5. Acceleration Profiles (filtered) of Task Space Experiment 1 (Note different acceleration scales)

Demanded Speed Along Path [mm/s (MMAS)]	102 (10%)	305 (30%)	711 (70%)	1016 (100%)
Industrial Planner Motion Time [s]	7.9	2.7	1.2	1.1
Quintic Planner Motion Time [s]	7.2	2.6	1.4	1.2
Industrial Planner Path Deviation RMS [mm]	0.5	0.5	0.6	0.6
Quintic Planner Path Deviation RMS [mm]	0.1	0.2	0.3	0.4
Industrial Planner Maximum Path Deviation [mm]	1.5	1.5	1.6	1.5
Quintic Planner Maximum Path Deviation [mm]	0.5	0.6	0.9	0.8

Table 7.6. Motion Times and RMS Errors of Task Space Experiment 1

7.2.1.2. Experiment 2: Task-Space Straight Line with Constant Orientation

The example path of Section 7.2.1.1 provided a motion with a distance sufficiently large that the MMAS could be reached during the motion. The motion distance in Task Space Experiment 2 (Table 7.7) is limited such that the quintic planner limits the path speed attainable because of the shortened distance. That is, the MMAS cannot be reached, even if such a speed is demanded.

In Figure 7.6, the maximum speed attainable with the quintic planner is just above 400 mm/s, while that of the industrial planner is just below 1000 mm/s. In fact, it was seen that the industrial planner speed profile remained the same with demanded path speeds of 80% (813 mm/s), 90% (914 mm/s) and 100% (1016 mm/s) of the MMAS. That is, the high acceleration demanded for this motion causes overshoot of the demanded path speed (at 80 and 90 % MMAS). On the other hand, at these speeds, the quintic planner provides a trajectory that accelerates to the maximum attainable speed given the jerk and acceleration limits, before decelerating to zero speed at the end of the motion.

When the industrial planner is used, the path deviation increases substantially as the speed increases (Table 7.8, Figure 7.7). The quintic planner provides low average path deviations by respecting the speed, acceleration and jerk limits. However, both planners provide roughly equivalent path deviations at comparable motion times. As expected, the path deviation increases during a non-zero acceleration phase (Figures 7.8 and 7.9). The industrial planner trajectory was based on an acceleration which violated the MMAA for demanded speeds above 203 mm/s (30% MMAS). The maximum speed reached by the industrial planner trajectories during the motion is higher than that reached by the corresponding quintic planner trajectories. Consequently, the motion times corresponding to the industrial planner trajectories are smaller than those resulting from the quintic trajectories. However, large path deviations occur when the acceleration limits are exceeded (Figures 7.8(b) and 7.9(b)). Figure 7.10 shows that the quintic trajectories allow the controller to maintain the end effector orientation closer to the desired (constant) orientation than the industrial planner.

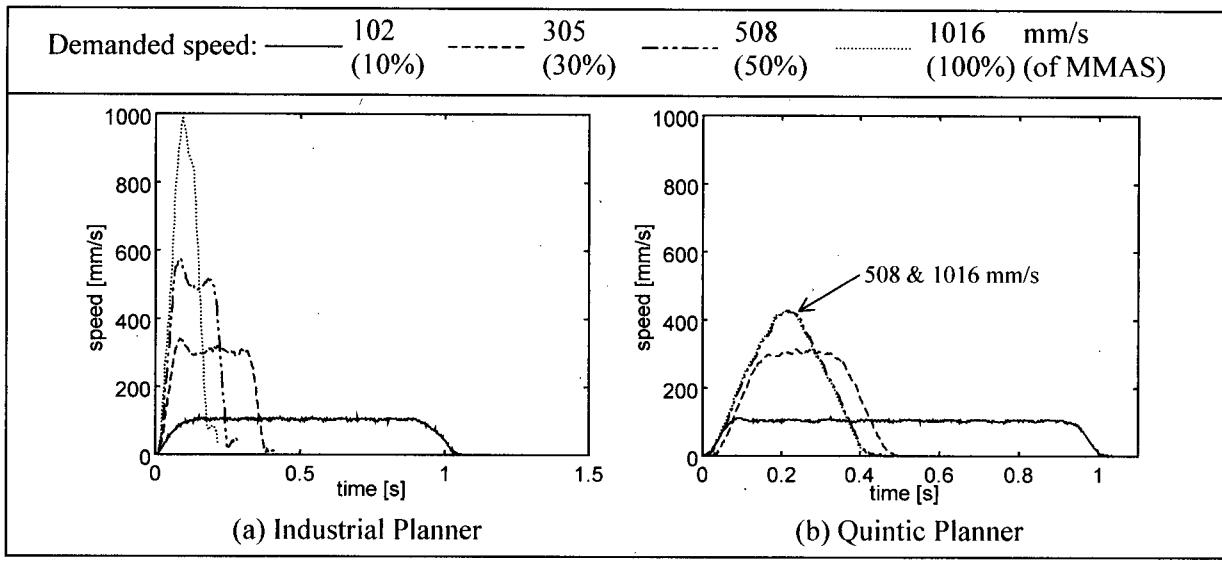


Figure 7.6. Comparison of Speed During Motion of Task Space Experiment 2

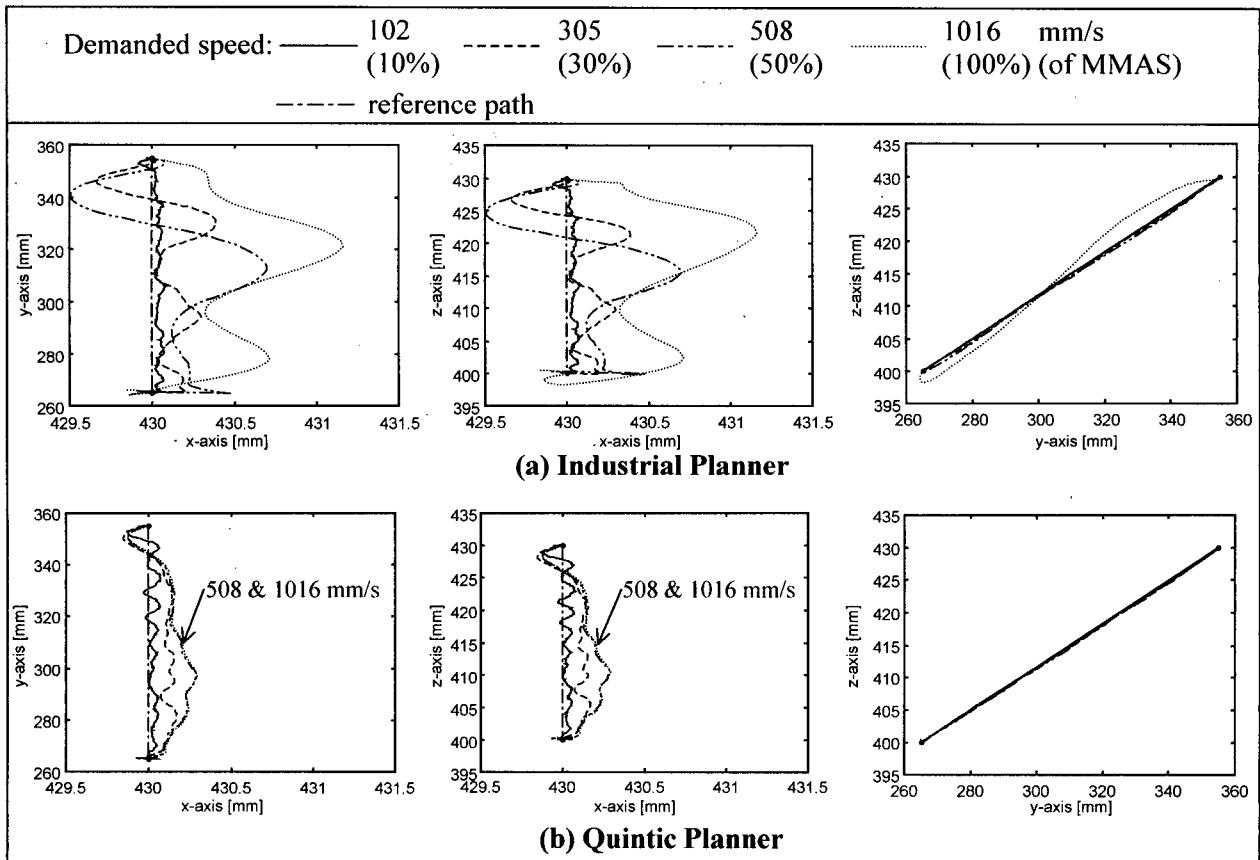


Figure 7.7. Comparison of Tracking Performance of Task Space Experiment 2 (Note that x-axis scale has been magnified)

Point	x [mm]	y [mm]	z [mm]	yaw [rad]	pitch [rad]	roll [rad]
1	430	355	430	$\frac{2\pi}{9}$	$\frac{\pi}{360}$	$\frac{\pi}{90}$
2	430	265	400	$\frac{2\pi}{9}$	$\frac{\pi}{360}$	$\frac{\pi}{90}$

Table 7.7. Coordinates of the Straight Line Path Start and End Points for Task Space Experiment 2

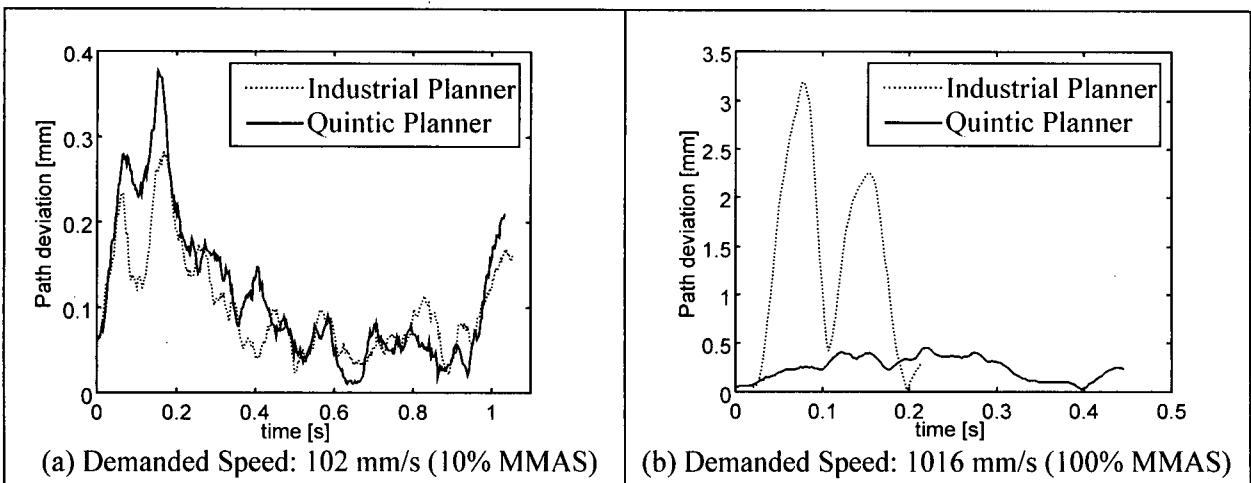


Figure 7.8. Comparison of Path Deviation During Motion of Task Space Experiment 2

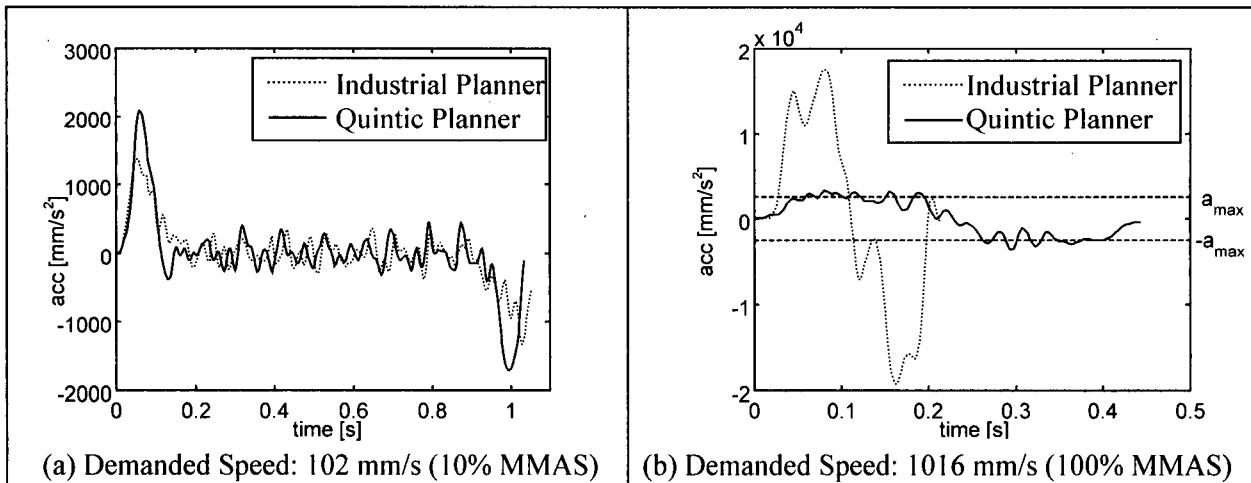


Figure 7.9. Comparison of Acceleration Profiles (filtered) of Task Space Experiment 2 (Note different acceleration scales)

Demanded Speed Along Path [mm/s (MMAS)]	102 (10%)	305 (30%)	508 (50%)	1016 (100%)
Industrial Planner Motion Time [s]	1.1	0.4	0.3	0.2
Quintic Planner Motion Time [s]	1.0	0.5	0.5	0.5
Industrial Planner Path Deviation RMS [mm]	0.1	0.3	0.5	1.6
Quintic Planner Path Deviation RMS [mm]	0.2	0.3	0.3	0.3
Industrial Planner Maximum Path Deviation [mm]	0.3	0.6	0.8	3.2
Quintic Planner Maximum Path Deviation [mm]	0.4	0.5	0.5	0.5

Table 7.8. Motion Times and RMS Errors Corresponding to Task Space Experiment 2

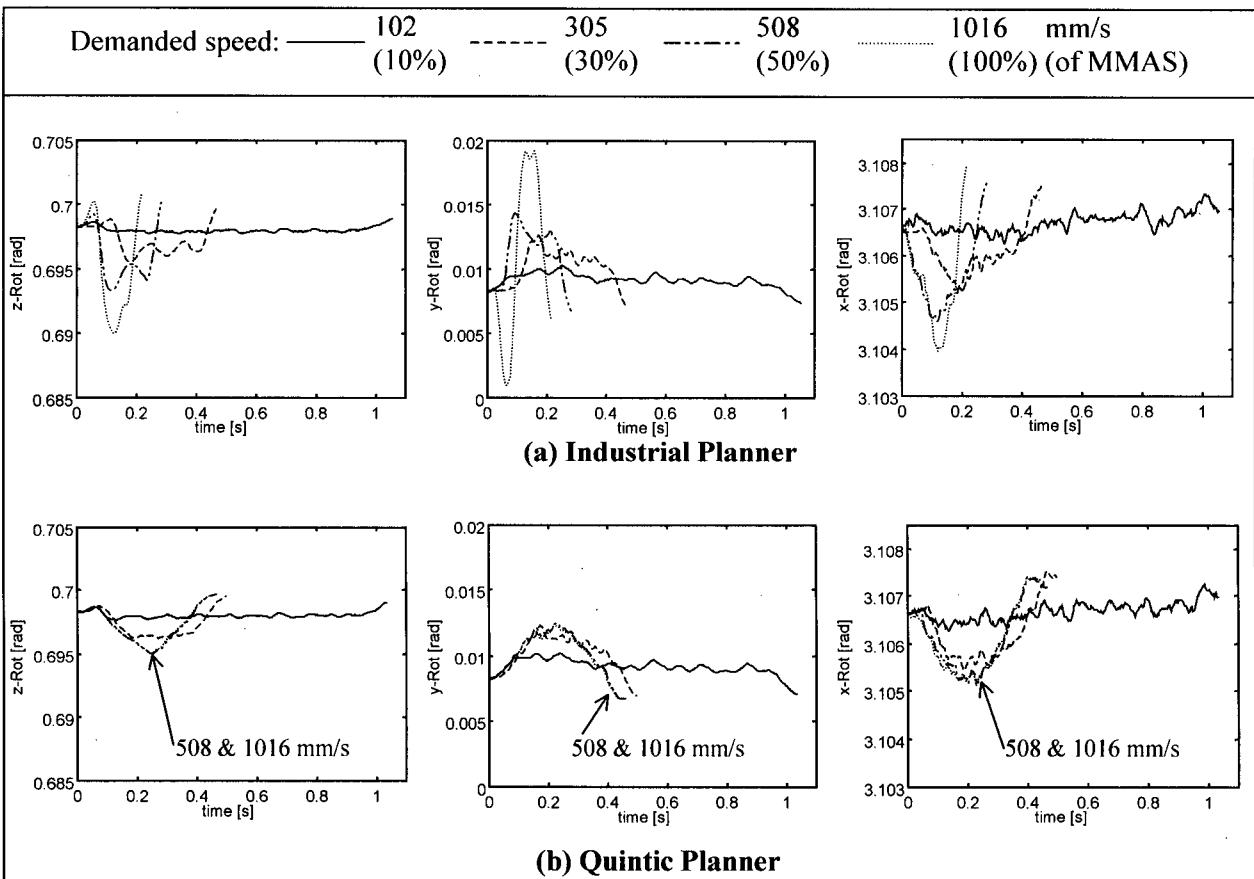


Figure 7.10. Comparison of Orientation With Respect to World Frame During the Motion for Task Space Experiment 2

7.2.1.3. Experiment 3: Task Space Motion with a Change in Orientation

To analyze a motion with a change in orientation an axis-angle representation is utilized. An orientation motion will be smooth if the derivatives of the angle trajectory, $\dot{\phi}(t)$, are continuous. As explained in Chapter 5, the quintic planner uses a quintic trajectory for $\dot{\phi}(t)$ with zero speed and acceleration end conditions between two way-points. The axis of rotation is held constant during the motion. Limits are placed on $\dot{\phi}(t)$ and its derivatives. As shown in Figure 7.11, for a motion which only requires a change in orientation (Table 7.9), the industrial planner simply computes one motion at the manufacturer limits. That is, it does not respect the demanded speeds. The quintic planner, on the other hand, respects the demanded speed and also provides a much smoother $\dot{\phi}(t)$ profile. However, in this example, the quintic planner motion time is always higher than that of the industrial planner, due to the imposed and respected acceleration limits.

One measure of smoothness for orientation is the amount of “wobble” associated with the motion. That is, for a motion described by an axis-angle representation, how much the path orientation deviates from this axis. The objective is to have a smooth motion about a single axis for operations such as pouring liquids or turning nuts or screws. Table 7.10 shows the angle between the desired axis of rotation and the actual axis of rotation, given at time increments along the path. The industrial planner motion clearly does not rotate about the desired axis, while the quintic planner motion does, although the angle between the desired and actual axis increases as the maximum path speed is increased.

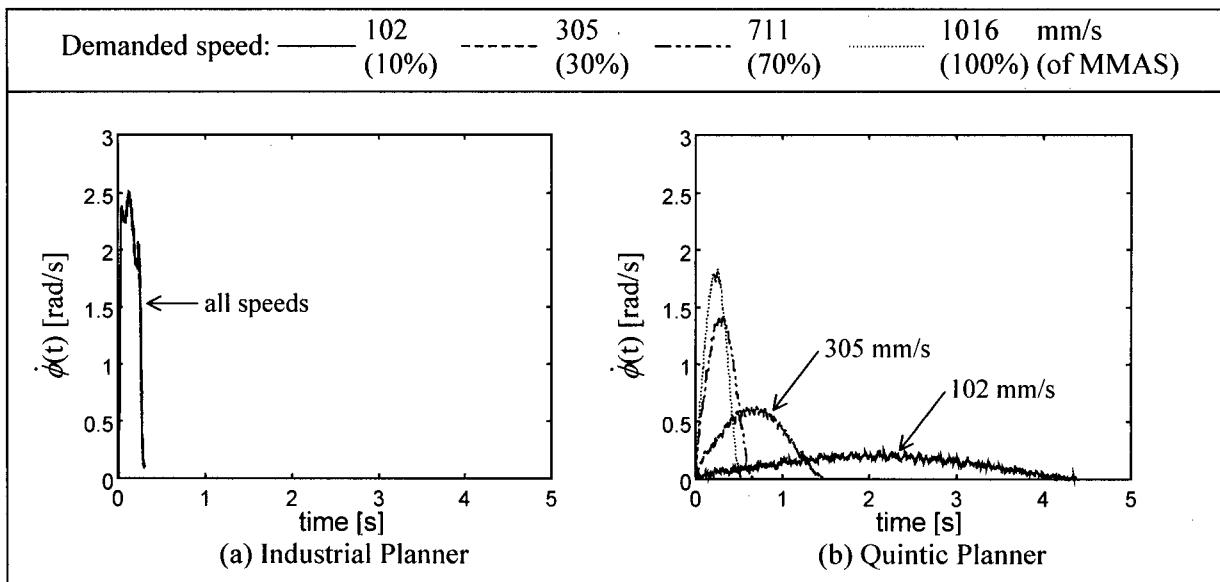


Figure 7.11. Comparison of $\dot{\phi}(t)$ at Different Maximum Allowable Speeds

Point	x [mm]	y [mm]	z [mm]	yaw [rad]	pitch [rad]	roll [rad]
1	590	-325	220	$-\frac{\pi}{18}$	$-\frac{\pi}{12}$	$\frac{\pi}{180}$
2	590	-325	220	$-\frac{\pi}{6}$	$\frac{\pi}{20}$	$\frac{\pi}{200}$

Table 7.9. Coordinates of Orientation Motion Start and End Points

		Industrial Planner		Quintic Planner		
Demanded Speed Along the Path [mm/s (MMAS)]		all speeds	102 (10%)	305 (30%)	711 (70%)	1016 (100%)
Motion Time [s]		0.3	4.3	1.5	0.7	0.5
Angle Between the Desired Rotation Axis and the Actual Rotation Axis [rad] at	25%	max.	0.12	0.0043	0.0088	0.012
	25%	min.	0.11	0.0016	0.0062	0.0095
	50%	max.	0.077	0.0012	0.0033	0.0067
	50%	min.	0.058	0.00045	0.0021	0.0058
	75%	max.	0.040	0.0010	0.0012	0.0035
	75%	min.	0.014	0.00062	0.00033	0.0024
	100%	max.	0.0032	0.00080	0.0014	0.0016
	100%	min.	0.00097	0.00011	0.0013	0.0012

Table 7.10. Angle Between the Desired Rotation Axis and the Actual Rotation Axis at Time Increments for Task Space Experiment 3

7.2.1.4. Summary of Results for Task Space Stop Point to Stop Point Motions

The three experiments discussed above, as well as an analysis of data for four other paths (Appendix H, Section H.1) demonstrate that the quintic planner provides faster motions at lower demanded path speeds, with lower path deviations. The shorter motion times are due, in part, to the industrial planner not reaching the acceleration limit. It was seen that when, for a given demanded speed, the industrial planner provided a faster motion, it did so by violating the manufacturer maximum allowable acceleration (Appendix H, Section H.1). The quintic planner also provides the ability to limit the kinematic parameters during a change in orientation while the industrial planner does not.

Comparison of maximum path deviations at comparable motion times for showed that in most cases, the quintic planner provided considerably improved maximum path deviation (ranging from 14% to 47% improvement). In one case, the quintic planner provided poorer maximum path deviation at comparable motion times, because the acceleration limit was too high for the given task-space path (refer to Section

H.1.6.1, Appendix H). However, in all cases, implementation of the quintic planner trajectories resulted in lower maximum path deviations at a demanded speed of 1016 mm/s (100% MMAS). That is, use of the quintic planner guarantees a lower maximum path deviation at the maximum demanded speed. The same conclusions apply to the average path deviations.

In summary, for point-to-point motions, the quintic planner provides a motion that respects the kinematic limits, optimizing the trajectory within these limits. At low demanded speeds, the quintic planner provides a faster motion since it reaches the maximum acceleration possible given the velocity and jerk constraints. At high speeds the quintic planner provides a slower motion than the industrial planner because the industrial planner violates the acceleration constraints, resulting in a trajectory that causes large path deviations when the controller tries to track it.

7.2.2. Task-Space Multiple Way-Point Motion

A multiple way-point task-space motion example is provided in this section. The way-points are given in Table 7.11. For the industrial planner, as the maximum allowable speed increases, the path shape is modified, i.e., the blend radius increases (Figure 7.12(a)). The quintic planner, on the other hand, generates the same path regardless of the demanded speed (Figure 7.12(b)). At all demanded speeds, the quintic planner provides a faster overall motion (Table 7.12).

Point	x [mm]	y [mm]	z [mm]	yaw [rad]	pitch [rad]	roll [rad]
1	315	-300	390	$\frac{5\pi}{9}$	$\frac{\pi}{2}$	$\frac{11\pi}{18}$
2	470	-300	390	$\frac{5\pi}{9}$	$\frac{\pi}{2}$	$\frac{11\pi}{18}$
3	470	300	390	$\frac{5\pi}{9}$	$\frac{\pi}{2}$	$\frac{11\pi}{18}$
4	315	300	390	$\frac{5\pi}{9}$	$\frac{\pi}{2}$	$\frac{11\pi}{18}$
5	315	-300	390	$\frac{5\pi}{9}$	$\frac{\pi}{2}$	$\frac{11\pi}{18}$

Table 7.11. Coordinates of Way-Points Corresponding to Figure 7.12.

The fact that the quintic planner path is invariable over a range of demanded speeds is important in that it provides a predictable motion. When the industrial planner is used to design a motion at low speeds that is later run at a high speed, the path changes and the manipulator end effector may subsequently contact an object in the workspace.

As shown in Figure 7.13, at a demanded path speed of 102 mm/s (10% MMAS), there is only a slight decrease in speed at the third and fourth way-points when the industrial planner is used. The corresponding acceleration and jerk components will be very large. The quintic planner provides limited jerk and acceleration when cornering (or blending).

The difference between the motion times of the industrial and quintic trajectories increases as the speed increases due mainly to the higher speeds attained along the straight line segments with the quintic trajectories. It was also found that the industrial planner motion time increased slightly when the maximum path speed was increased to 90 (914 mm/s) and 100% (1016 mm/s) MMAS (refer to Section H.2.1, Appendix H, for complete results).

A second multiple way-point example is provided in Section H.2.2, Appendix H, and corroborates the above conclusions.

Demanded Speed Along the Path [mm/s (% MMAS)]	102 (10)	406 (40)	508 (50)	1016 (100)
Industrial Planner Motion Time [s]	15.12	4.75	4.17	3.90
Quintic Planner Motion Time [s]	14.96	4.13	3.66	2.99

Table 7.12. Motion Times for Task Space Multiple Way-Point Path

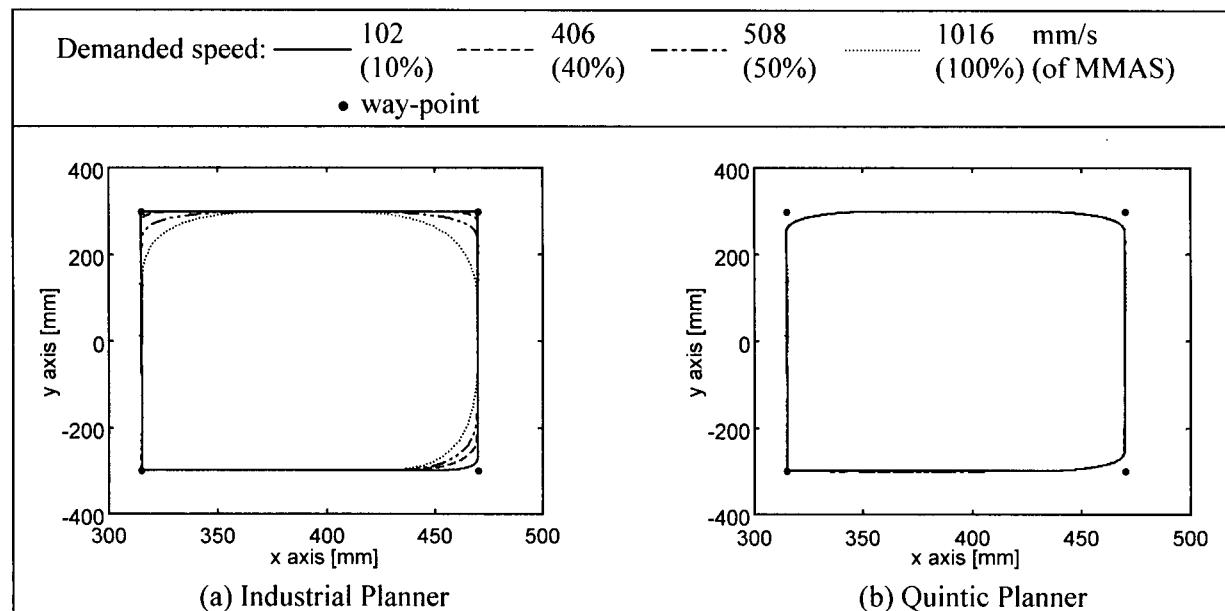


Figure 7.12. Comparison of Paths at Different Speeds for Task Space Multiple Way-Point Path

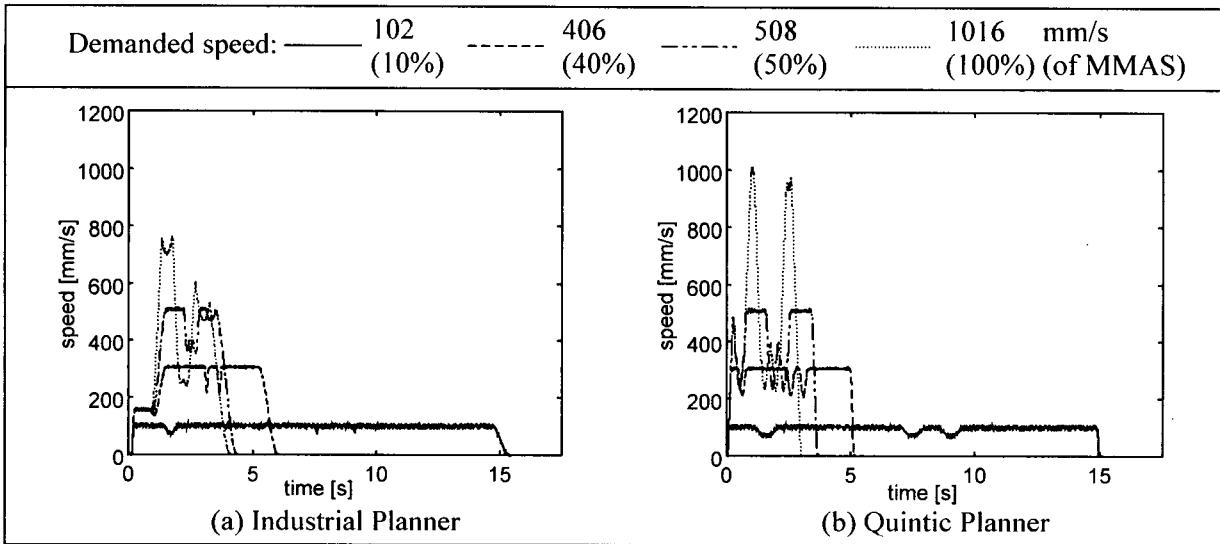


Figure 7.13. Comparison of Speed Profiles for Task Space Multiple Way-Point Path

7.2.3. Joint-Space Stop Point to Stop Point Motion

For stop point to stop point motions in joint space, the most important considerations are the motion times and trajectory end-point accuracy. It is not important that the joint space path be followed accurately since this type of motion is used to move rapidly between way-points when the robot is transiting between tasks (It is, however, important that the path be invariable with path speed). Thus, in this section, the difference between the actual and desired joint positions at the end of the motion is used as a comparison metric to determine how accurately the way-points are reached.

7.2.3.1. Experiment 1: Joint Space Straight Line

For the joint space motion between the two points given in Table 7.13, the quintic planner provides a faster motion than that of the industrial planner at all demanded speeds (Table 7.14, Figure 7.14). The industrial planner provides a slower motion because it uses a lower deceleration limit than acceleration limit, while the quintic planner uses the same acceleration and deceleration limit (Figure 7.15).

Point	Joint 1 [rad]	Joint 2 [rad]	Joint 3 [rad]	Joint 4 [rad]	Joint 5 [rad]	Joint 6 [rad]
1	$\frac{7\pi}{18}$	$-\frac{\pi}{9}$	$\frac{13\pi}{18}$	$\frac{\pi}{6}$	$\frac{2\pi}{9}$	$-\frac{5\pi}{18}$
2	$-\frac{4\pi}{9}$	$-\frac{5\pi}{9}$	$\frac{11\pi}{18}$	$-\frac{\pi}{3}$	$-\frac{\pi}{6}$	$-\frac{7\pi}{18}$

Table 7.13. Coordinates of Joint Space Straight Line Start and End Points for Joint Space Experiment 1

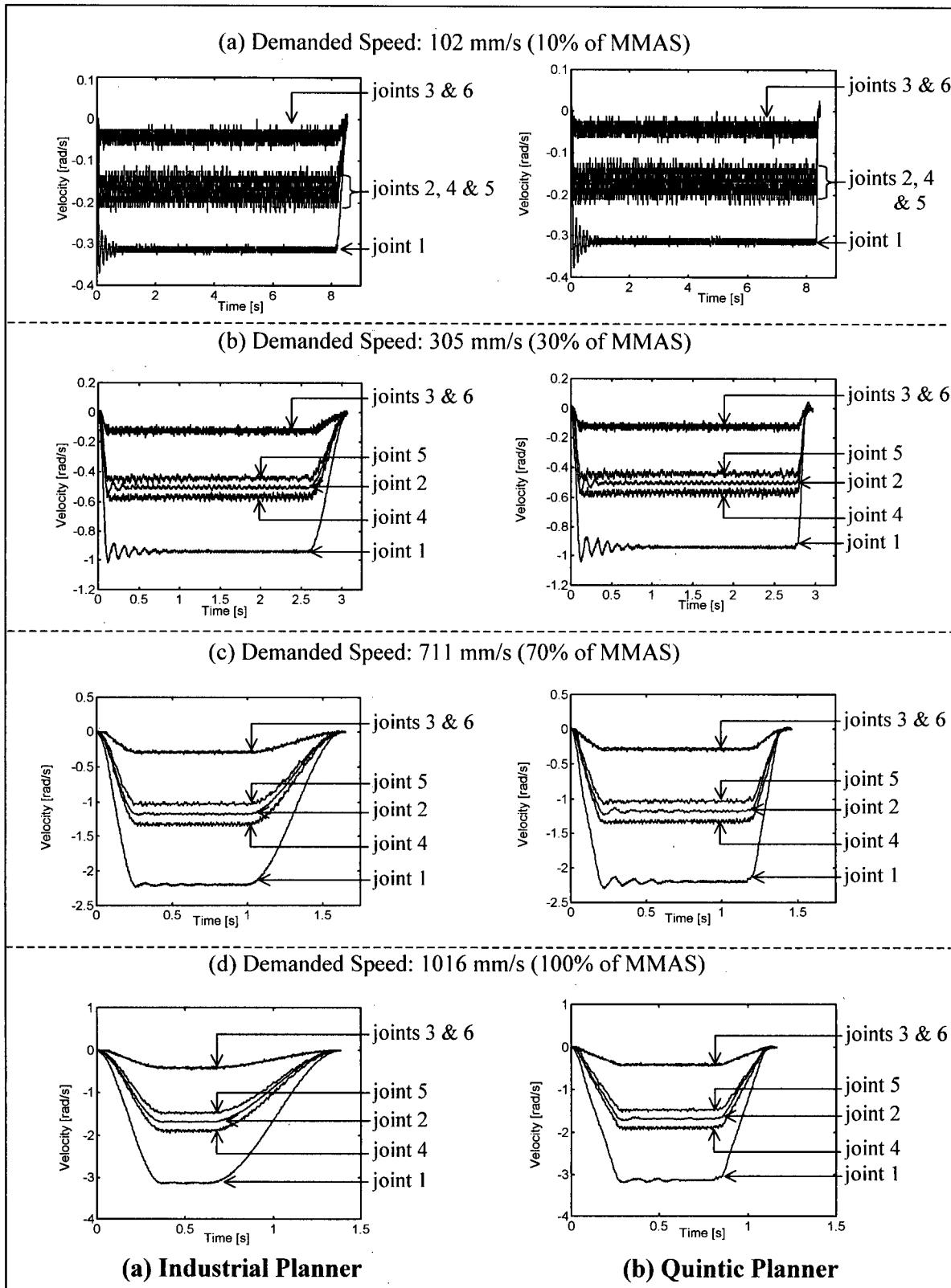


Figure 7.14. Comparison of Joint Velocities for Joint Space Experiment 1

Both the quintic and industrial planner trajectories cause the manipulator to reach the end of the trajectory with equivalent accuracy for comparable motion times (Table 7.14). In both cases, the accuracy with which the end point is reached decreases with increasing demanded speeds (refer to Section H.3.1, Appendix H for complete results).

		Industrial Planner	Quintic Planner						
Demanded Speed Along Path [mm/s (MMAS)]		102 (10%)		305 (30%)		711 (70%)		1016 (100%)	
Motion Time [s]		8.5	8.4	3.1	2.9	1.6	1.4	1.4	1.1
Deviation of Joint	1 [rad]	0.001	0	0	0	0	0	0	0
	2 [rad]	0	0.001	0.001	0.001	0.001	0.002	0.001	0.002
	3 [rad]	0	0.001	0	0.001	0.001	0.001	0.001	0.001
	4 [rad]	0	0	0.002	0.003	0.003	0.004	0.004	0.004
	5 [rad]	0	0.001	0.002	0.002	0.003	0.004	0.003	0.004
	6 [rad]	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001

Table 7.14. Motion Times and Accuracy with which the End Point is Reached for Joint Space Experiment 1

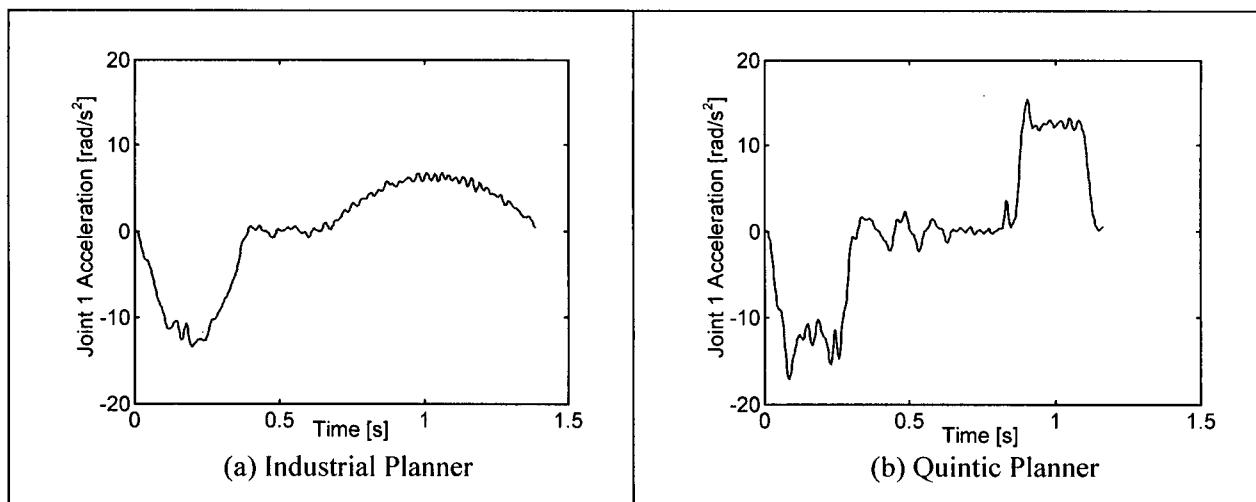


Figure 7.15. Comparison of Industrial Planner and Quintic Planner Acceleration Profile (filtered) for Joint Space Experiment 1 Trajectory with Demanded Speed of 1016 mm/s (100% MMAS)

7.2.3.2. Experiment 2: Joint Space Straight Line

The second joint space experiment was for a shorter distance, such that the peak speed was limited by end-point requirements (Table 7.15). The quintic planner did not allow the speed to increase once the demanded speed was above 508 mm/s (50% MMAS). The industrial planner actually produced motions that were slower when higher path speeds were demanded (Figure 7.16, Table 7.16). In fact, with the industrial planner, for a demanded speed of 1016 mm/s (100% MMAS) the motion time is actually larger than that for a demanded speed of 102 mm/s (10% MMAS). This occurs when a distance limitation is present, as explained in Section H.3.2 (Appendix H).

Point	Joint 1 [rad]	Joint 2 [rad]	Joint 3 [rad]	Joint 4 [rad]	Joint 5 [rad]	Joint 6 [rad]
1	$-\frac{\pi}{9}$	$-\frac{\pi}{3}$	$\frac{17\pi}{18}$	0	$-\frac{\pi}{9}$	0
2	$-\frac{5\pi}{36}$	$-\frac{5\pi}{18}$	$\frac{33\pi}{36}$	$-\frac{\pi}{18}$	$-\frac{\pi}{12}$	$\frac{\pi}{18}$

Table 7.15. Coordinates of Joint Space Straight Line Start and End Points for Joint Space Experiment 2

Demanded Speed Along the Path [mm/s (MMAS)]	102 (10%)	305 (30%)	711 (70%)	1016 (100%)
Industrial Planner Motion Time [s]	0.8	0.6	0.8	0.9
Quintic Planner Motion Time [s]	0.7	0.3	0.3	0.3

Table 7.16. Motion Times for Straight Line for Joint Space Experiment 2

7.2.3.3. Summary of Joint Space Stop Point to Stop Point Motion Results

The complete results for joint space stop point to stop point motion, as well as two further examples can be found in Section H.3, Appendix H. The quintic planner provides faster motions than the industrial planner at all demanded speeds because the industrial planner utilizes a lower deceleration and jerk than acceleration and jerk.

The method used by the industrial planner gives poor results over small motion distances. An increase in demanded speed results in an unexpected and undesirable increase in motion time. The quintic planner, on the other hand, reaches the demanded speed or the highest speed possible given the motion distance and kinematic limits.

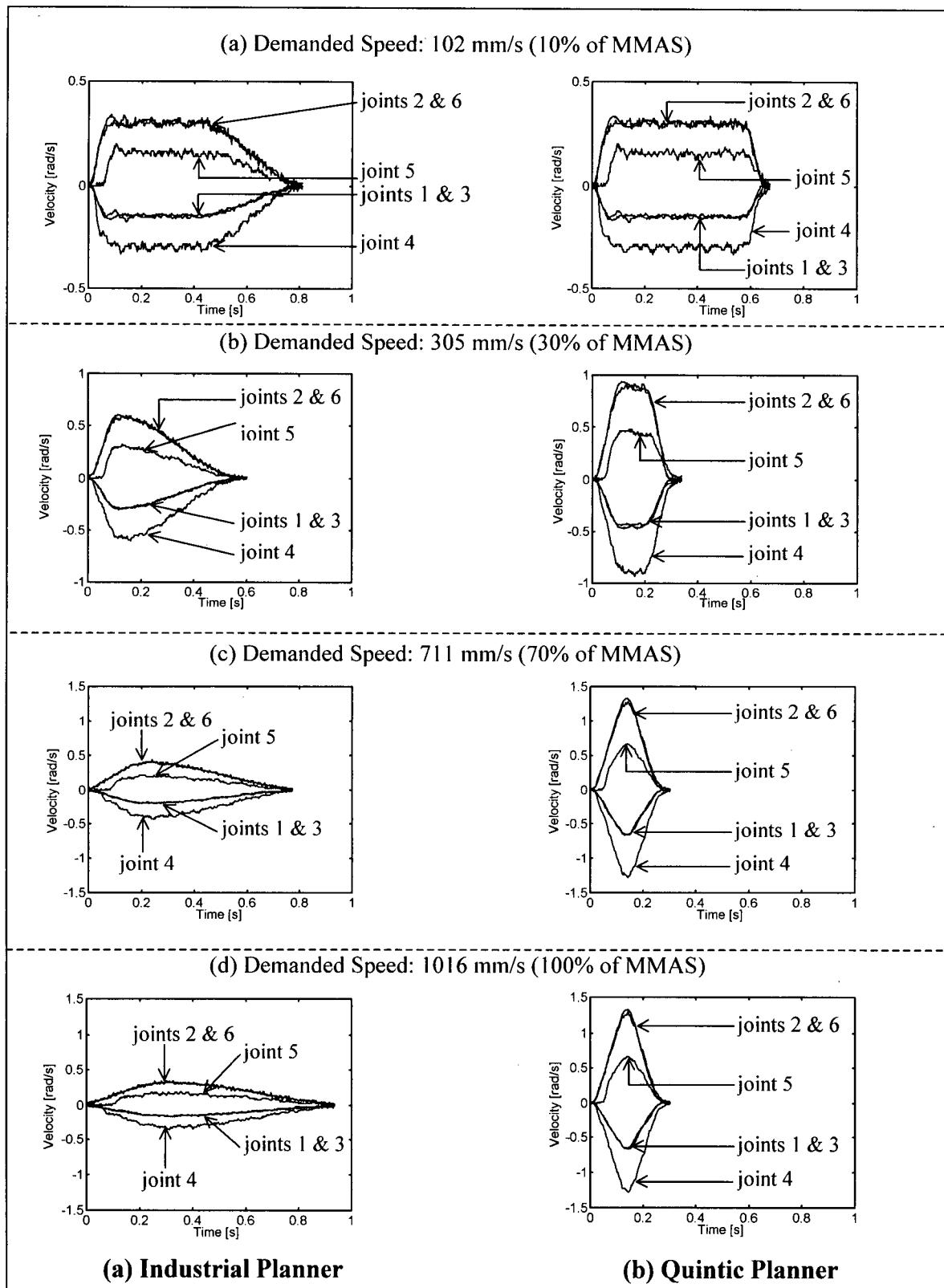


Figure 7.16. Comparison of Joint Velocities for Joint Space Experiment 2

With the imposed kinematic parameters, all the joint velocity profiles utilizing the quintic planner exhibit oscillations during the first part of the speed cruise region. In addition, the resulting acceleration profiles slightly overshoot the acceleration limits. An example is provided below. Since this is most visible for the joint that reached the highest path speed, the acceleration of joint 1 for the joint space experiment 1 at a demanded speed of 305 mm/s (30% MMAS) is shown in Figure 7.17(a).

As shown in Figure 7.17 (b), when the joint jerk limits are reduced such that the acceleration ramp time is equal to 150 ms, the resulting experimental acceleration profile follows the reference acceleration profile much more closely. In addition, the magnitude of the oscillations in the velocity profile is reduced (Figure 7.18, most visible in joints 1 and 2).

With the original joint jerk limits, the time allowed for a ramp to maximum acceleration, dt_{max} , is 20 ms. As shown in Figure 7.17(a), during the ramp down to minimum acceleration, the reference signal is not followed due either to controller or actuator saturation. However, when the jerk limit was reduced (Figure 7.17(b)), the acceleration ramp was followed more closely. This indicates that the manufacturer joint jerk limits are set too high. That is, the manipulator was unable to follow the trajectory signal or the controller was saturated. Further work, beyond the scope of this thesis, is required to determine the correct joint jerk limits necessary to avoid controller saturation. This will require integrated monitoring of the reference signal, which cannot currently be done from the encoder debugging header provided on the C500C controller.

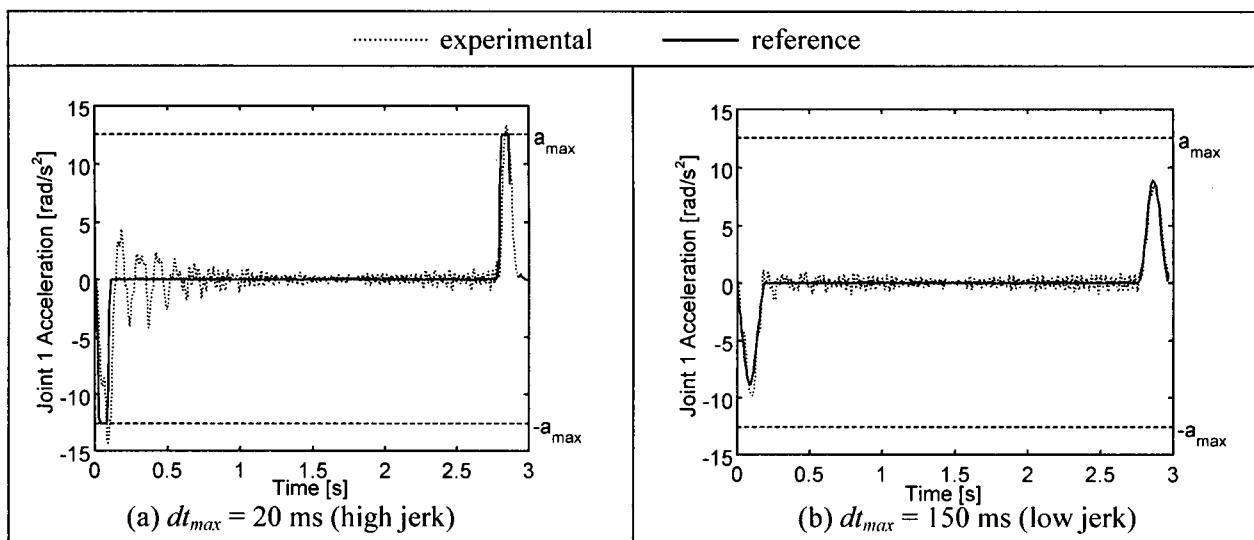


Figure 7.17. Comparison of Filtered Joint Acceleration Profiles for Different Jerk Limits (Joint Space Experiment 1, demanded speed of 305 mm/s (30% MMAS))

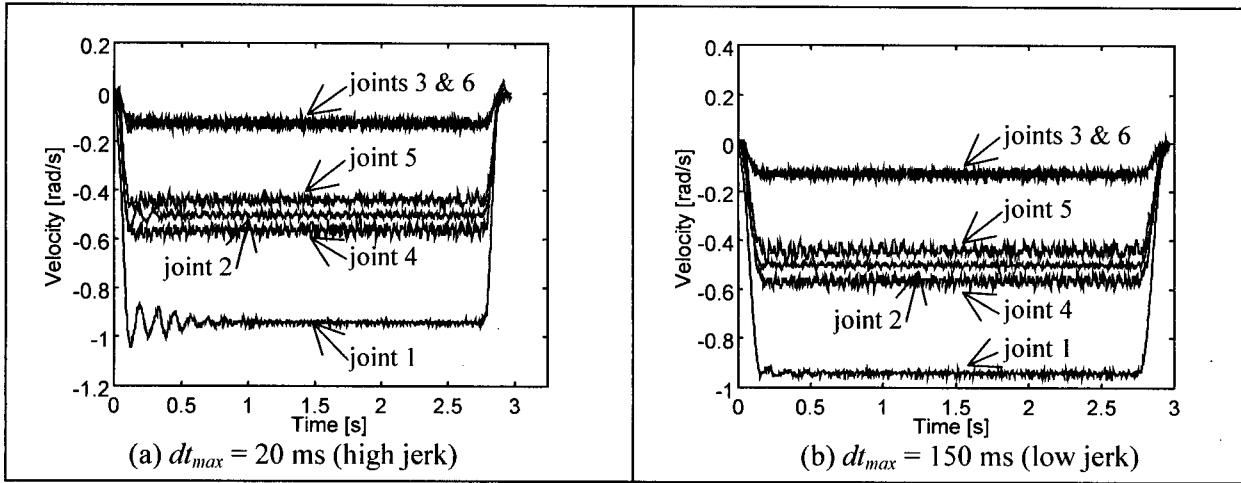


Figure 7.18. Comparison of Joint Velocity Profiles With Different Jerk Limits (Joint Space Experiment 1, demanded speed of 305 mm/s (30% MMAS))

7.2.4. Joint Space Multiple Way-Point Motion

A joint space motion between the points given in Table 7.17 demonstrates that the quintic planner provides a consistently faster motion between the way-points at all maximum path speeds (Table 7.18). The industrial planner path was modified as the demanded speed was increased; the blend size increasing with demanded speed (Figure 7.19(a)). The quintic planner path remained constant with increasing speed (Figure 7.19(b)). At low demanded speeds, the industrial planner path will require large jerks due to the sharp blends.

Point	Joint 1 [rad]	Joint 2 [rad]	Joint 3 [rad]	Joint 4 [rad]	Joint 5 [rad]	Joint 6 [rad]
1	$\frac{\pi}{6}$	$-\frac{2\pi}{3}$	$\frac{10\pi}{9}$	0	0	0
2	$\frac{\pi}{3}$	$-\frac{7\pi}{18}$	$\frac{17\pi}{18}$	$\frac{\pi}{9}$	$-\frac{\pi}{18}$	$-\frac{\pi}{9}$
3	$-\frac{\pi}{3}$	$-\frac{7\pi}{18}$	$\frac{17\pi}{18}$	$-\frac{\pi}{9}$	$-\frac{\pi}{18}$	$\frac{\pi}{9}$
4	$-\frac{\pi}{3}$	$-\frac{5\pi}{18}$	π	$-\frac{\pi}{6}$	$-\frac{2\pi}{9}$	$\frac{11\pi}{18}$
5	$\frac{\pi}{6}$	$-\frac{2\pi}{3}$	$\frac{10\pi}{9}$	0	0	0

Table 7.17. Coordinates of Way-Points for Multiple Way-Point Joint Space Path

Demanded Speed Along the Path [mm/s (MMAS)]	102 (10%)	305 (30%)	711 (70%)	1016 (100%)
Industrial Planner Motion time [s]	21.32	7.31	3.47	2.88
Quintic Planner Motion Time [s]	21.21	7.18	3.28	2.69

Table 7.18. Motion Times for Multiple Way-Point Joint Space Path

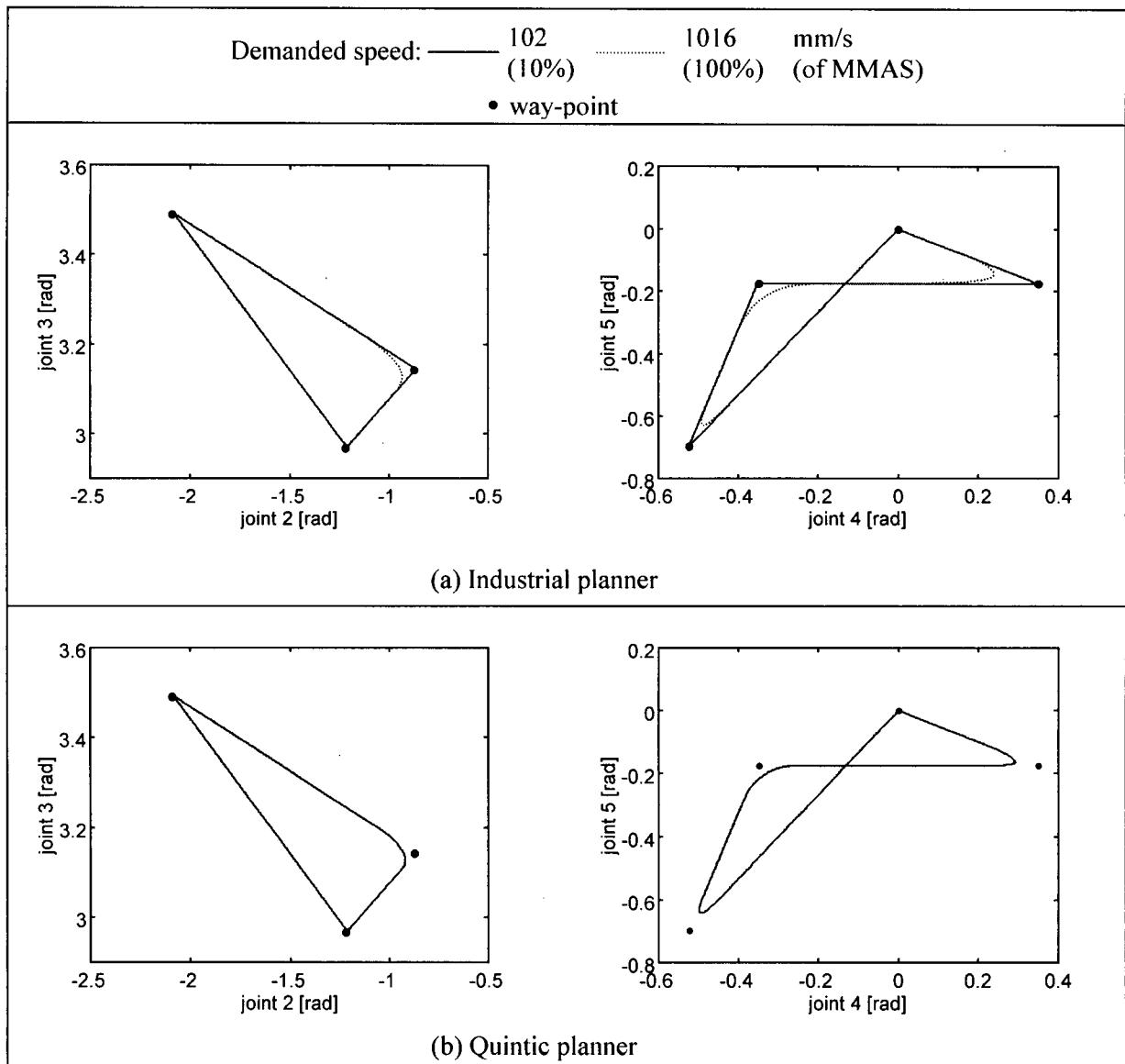


Figure 7.19. Comparison of Joint Space Paths for Joint Space Multiple Way-Point Experiment (plots chosen for best view of blends)

7.3. Comparison to SPCTOM

It has been shown that Smooth Path-Constrained Time-Optimal Motion (SPCTOM) with low torque rate limits results in a faster and more accurate motion than an optimal trajectory that does not incorporate torque rate limits [14]. The quintic concatenation method was compared to the SPCTOM with low torque rate limits using the positional *dof* of a SCORBOT ER VII. The path followed, described in parametric form, is

$$\begin{aligned}x(\sigma) &= 400 \text{ mm} \\y(\sigma) &= 300s - 100 \text{ mm} \\z(\sigma) &= 200s + 300 \text{ mm} \\\sigma &= 0, \dots, 1.\end{aligned}\tag{7.1}$$

The SPCTOM trajectory was designed for torque saturation limits of 10 N.m and torque rate limits of 10 N.m/s for each joint.

The quintic concatenation trajectory was generated with maximum allowable speed, v_{max} , of 370 mm/s, maximum acceleration, a_{max} , of 890 mm/s², and jerk limit, j_{max} , of 4450 mm/s³, resulting in similar torque and torque rate demands.

As shown in Figures 7.20, 7.21 and Table 7.19, the two methods are comparable. Thus, the quintic concatenation method, although slightly slower, provides a smooth trajectory, which has the same properties as the SPCTOM, but is computationally simpler and therefore faster to generate. The large RMS tracking errors are due, in part, to controller model errors and the control scheme being used, namely, independent joint control.

Trajectory	Motion Time [s]	RMS Tracking Error [rad]			RMS Path Deviation [mm]	Maximum Path Deviation [mm]
		jt 1	jt 2	jt 3		
Quintic Concatenation	1.7	0.039	0.039	0.028	5	8
SPCTOM	1.5	0.042	0.034	0.026	4	8

Table 7.19. Motion Times for SPCTOM Trajectory

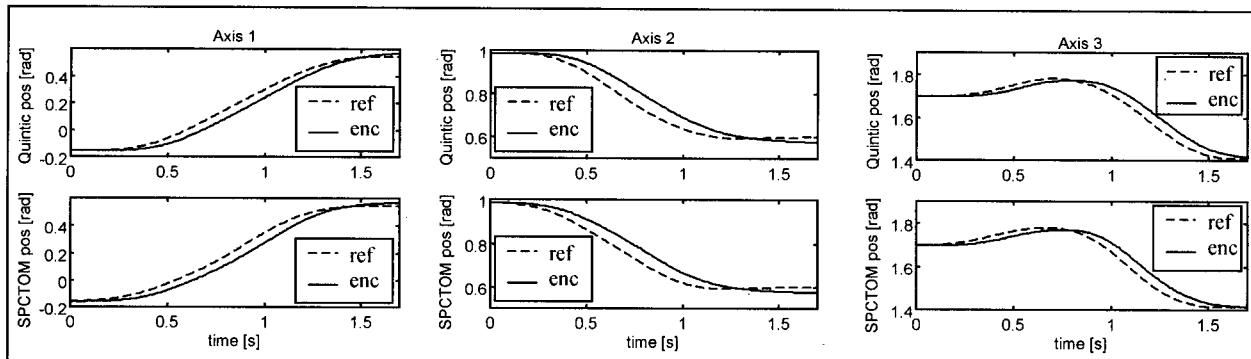


Figure 7.20. Comparison of Quintic Concatenation and SPCTOM Joint Performance

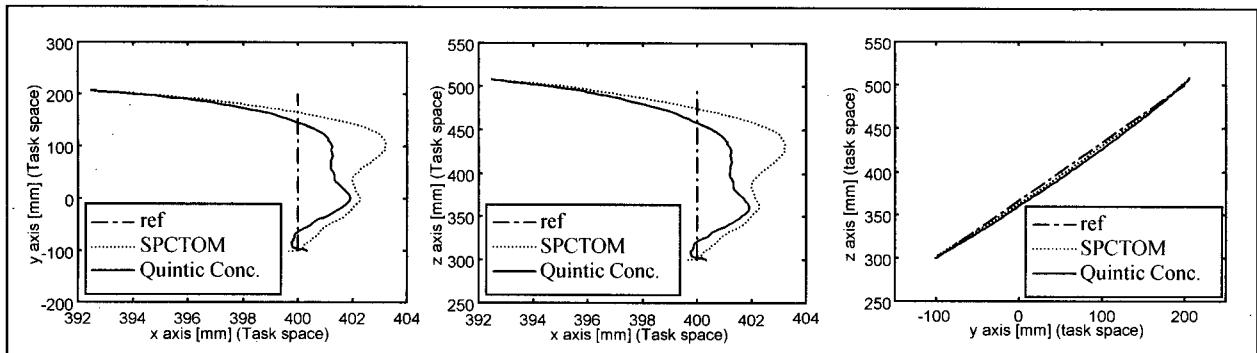


Figure 7.21. Comparison of Tracking for SPCTOM and Quintic Concatenation Method

7.4. Computation Time Issues

The typical computation time for both the industrial and quintic trajectories is on the order of 1 ms for a point to point trajectory. The computation time for a point to point smooth path-constrained time-optimal motion is on the order of 30 s; that is, four orders of magnitude greater than that of the kinematic methods presented. Most importantly, dynamic methods generally involve optimization and therefore do not have a fixed computation time. The quintic planning method described herein is unique among the jerk-limited motion planning schemes reviewed in Chapter 2 in that it demonstrates an analytical solution approach that does not require any iterative calculations. That is, it has a fixed and known computation time, and has been implemented on-line in an industrial controller setting.

7.5. Summary

In this chapter the quintic planner was compared to an industrial planner and to a smooth, time-optimal motion planner. The industrial and quintic planner trajectories were tracked using the same controller.

It was found that, in task space stop point to stop point motions, the quintic planner provided faster motion and decreased average and maximum path deviation at low demanded path speeds. At higher demanded path speeds, the industrial planner provided faster motions but poorer path deviations at similar motion times since the trajectories violated the acceleration limits. For multiple way-point task space motions, the quintic planner provides consistently faster motion times and a constant motion path. The industrial planner trajectories are slower and vary the path with the maximum demanded path speed.

In joint space, the quintic planner provided consistently faster point-to-point and multiple way-point motions since the industrial planner uses a lower deceleration than acceleration. In distance-limited cases, as the demanded path speed was increased, the industrial planner produced increasingly slower motions. The quintic planner always provides motion times that decrease or remain constant as the demanded speed is increased.

Comparison to a SPCTOM trajectory showed that the quintic trajectories result in slightly poorer tracking accuracy and slightly longer motion time. However, the quintic trajectories can be generated on-line, while the SPCTOM trajectories must be implemented off-line.

Chapter 8

Conclusions & Recommendations

8.1. Conclusions

In this thesis, a smooth, on-line, trajectory generation method has been presented. The trajectory path consists of straight lines interconnected with geometric blends, which pass a specified distance from the way-points. The smoothness of the trajectories is obtained through appropriate selection of the jerk limits, and planning trajectories that respect these limits.

The straight-line trajectories are formed using concatenated quintic segments that impose C^2 continuity on the trajectory and limit the velocity, acceleration and jerk during the trajectory. In addition, undesirable oscillations in the position and velocity domains are avoided. The blend function used to connect the straight line trajectories also imposes C^2 continuity, and a method as been proposed herein to limit the acceleration and jerk during the blend.

Both the straight-line and blend trajectories have zero acceleration end conditions. Thus, by design, the overall trajectory has a continuous acceleration profile along with limited velocity, acceleration and jerk. Utilizing conservative, but reasonable kinematic limits, this approach provides an efficient, near time-optimal, on-line planning method.

The ability to plan in both task and joint space is provided. The task space motion can be used in tasks when the path followed by the manipulator end effector is of importance. Joint space motion can be used when transiting between tasks, in areas of the workspace where no obstacles are present, since joint-space motion provides a faster motion between any two way-points than task-space motion.

A high level system that allows trajectories between multiple way-points to be planned on-line was also developed. The trajectory is planned by calculating quintic control points (up to eight per way-point to way-point trajectory) and the blend parameters, which are used at a later stage, by the parser, to generate the controller reference signal.

The trajectory generation method was implemented using a typical industrial controller. Comparison of the quintic planner to an industrial planner with no jerk limitations showed that the quintic planner provided faster motions in joint space with equivalent end point accuracy. In task space, the quintic planner provided an overall reduction in maximum and average path deviations. In addition, the quintic planner provides trajectories that consistently utilize the maximum possible acceleration and speed. In task space, as the demanded speed is increased, the industrial planner also increases the acceleration, eventually violating the manufacturer maximum allowable acceleration. As the demanded speed is increased, the quintic planner provides trajectories with decreased or constant motion times. The industrial planner, on the other hand, provides increased motion time with increasing demanded speed in joint space when the planned motion is short.

The successful implementation of the quintic concatenation trajectories on a commercial industrial controller demonstrates that the method requires a computational load low enough for use on-line. In addition, the method is compared to a smooth dynamic-based trajectory, the smooth, path-constrained, time-optimal motion (SPCTOM), and it is shown that similar, if slightly poorer, motion times and tracking accuracy are obtained through use of the quintic trajectories. However, the quintic trajectories can be planned and generated much more rapidly than the SPCTOM trajectories.

8.2. Recommendations

This research has provided an improved method of calculating trajectories on-line, which can be used to avoid controller saturation and the resulting wear on the machine by appropriate selection of the jerk limits. This methodology can easily be extended to accommodate other basis functions. However, there are a number of improvements that can be explored. These, as well as a number of research directions, are discussed below.

- (1) The orientation motion quintic trajectory could be replaced by a concatenated quintic trajectory when the orientation motion time is limiting. However, this would lead to a more iterative algorithm.
- (2) The trajectory generation methodology provided herein for straight-line paths was based on minimizing the jerk discontinuities. However, the blends used to join the straight-line segments cause large discontinuities in jerk at the blend start and end points. The basis function (sine wave template) could be modified to allow a bang-bang jerk profile, providing a faster motion time but resulting in even larger jerk discontinuities. This change would only require the modification of the quintic end conditions and the trajectory would in fact become a concatenation of cubics, similar to the work done by Castain and Paul [9]. However, no iterative calculations would be required if the method

proposed herein were used, and the motion time would be calculated directly by the planner. Alternatively, the blend functions could be modified in order to decrease the size of the jerk discontinuity.

- (3) The above considerations lead to another area that should be investigated: the role of snap (the derivative of jerk with respect to time) in trajectory generation. It was found that the SCORBOT-ER VII manipulator deviated greatly from the trajectory shortly after the start of the blends, most often at high speeds. This may be due to the large change in jerk that occurs at these points. However, designing a trajectory with limited snap would increase the motion time.
- (4) One could also investigate the incorporation of local kinematic limits, based on the robot configuration and its expected joint velocities and accelerations. This would allow the kinematic limits to be optimized for the given robot configuration, increasing the motion time and tracking accuracy of the manipulator.
- (5) Finally, when switching between joint space planning and task space planning, it is necessary to stop the manipulator at the intermediate way-point. Motion time would be improved with the ability to switch from task space planning to joint space planning or vice-versa without requiring the manipulator to stop.

The investigation of these topics would further contribute to the area of on-line trajectory planning.

Bibliography

- [1] Andersson, R., L., *A Robot Ping-Pong Player: Experiment in Real-Time Intelligent Control*, Cambridge, MIT Press (1988).
- [2] Angeles, J., Alivizatos, A., Zsombor-Murray, P., "The Synthesis of Smooth Trajectories for Pick-and-Place Operations", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 18, No 1., (1988): 173-178.
- [3] Bazaz, S., Tondu, B., "3-Cubic Spline for On-Line Cartesian Space Trajectory Planning of an Industrial Manipulator", *International Workshop on Advanced Motion Control*, (1998): 493-498.
- [4] Bobrow, J. E., Dubowsky, S., Gibson, J. S., "Time-Optimal Control of Robotic Manipulators Along Specified Paths", *International Journal of Robotics Research*, Vol. 4, No. 3, (1985): 3-17.
- [5] Brock, S., Kaczmarek, T., "2 D Command Preprocessor with Jerk Limit for Machine Tool Drives", *IEEE International Symposium on Industrial Electronics*, Vol. 1, (1996): 230-235.
- [6] *C500C Controller User Guide*, CRS Robotics, 2000.
- [7] Cao, B., Dodds, G. I., "Time-Optimal and Smooth Joint Path Generation for Robot Manipulators", *Proceedings of the International Conference on Control*, (1994): 1122-1127.
- [8] Cao, B., Dodds, G. I., Irwin, G. W., "Time-Optimal and Smooth Constrained Path Planning for Robot Manipulators", *Proceedings of the IEEE International Conference on Robotics and Automation*, (1994): 1853-1858.
- [9] Castain, R. H., Paul, R. P., "An On-Line Dynamic Trajectory Generator", *The International Journal of Robotics Research*, Vol. 3, No. 1, (1984): 68-72.
- [10] Chand, S., Doty, K.L., "On-Line Polynomial Trajectories for Robot Manipulators", *International Journal of Robotics Research*, Vol. 4, No. 2, (1985): 38-48.
- [11] Chang, Y.-H., Lee, T.-T., Liu, C.-H., "On-Line Approximate Cartesian Path Trajectory Planning for Robotic Manipulators", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 22, No. 3, (1992): 542-547.
- [12] Chen, Y. B., Desrochers A. A., "Structure of Minimum-Time Control Law for Robotic Manipulators With Constrained Paths", *Proceedings of the IEEE International Conference on Robotics and Automation*, (1989): 971-976.
- [13] Chen Y. B., Desrochers A. A., "Time-Optimal Control for Two-Degree of Freedom Robot Arms", *IEEE Transactions on Robotics and Automation*, Vol. 6, No. 3, (1990): 1210-1215.
- [14] Constantinescu, D., *Smooth Time Optimal Planning for Industrial Manipulators*, M.A.Sc., Department of Mechanical Engineering, University of British Columbia (1998).

- [15] Constantinescu, D., Croft, E. A., "Smooth and Time-Optimal Trajectory Planning for Industrial Manipulators Along Specified Paths", *Journal of Robotic Systems*, Vol. 17, No. 5, (2000): 233-249.
- [16] Croft, E. A., Benhabib, B., Fenton, R.G., "Near-Time Optimal Robot Motion Planning for On-Line Applications", *Journal of Robotic Systems*, Vol. 12, No. 8, (1995): 553-567.
- [17] Dahl, O., Nielsen, L., "Torque-Limited Path Following by On-Line Trajectory Time Scaling", *IEEE Transactions on Robotics and Automation*, Vol. 6, No. 5, (1990): 554-561.
- [18] Dahl, O., "Path-Constrained Robot Control with Limited Torques - Experimental Evaluation", *IEEE Transactions on Robotics and Automation*, Vol. 10, No. 5, (1994): 658-669.
- [19] Driedger, K., *Letter*, "Requirements Document: Robot Motion and Trajectory Planning", CRS Robotics (January 10, 2000).
- [20] Erkorkmaz, K., *High Speed Contouring Control for Machine Tool Drives*, M.A.Sc. Thesis, Department of Mechanical Engineering, University of British Columbia (1999).
- [21] Erol, N.A., Altintas, Y., "Open Architecture Modular Tool Kit for Motion and Process Control", *ASME International Mechanical Congress and Exposition*, ASME Publication MED, Vol. 6-1 (1997): 15-22.
- [22] Fang, Y. C., Hsieh, C. C., Kim, M. J., Chang, J. J., Woo, T. C., "Real Time Motion Fairing With Unit Quaternions", *Computer-Aided Design*, Vol. 30, No. 3, (1998): 191-198.
- [23] Fisher, W. D., Mujtaba, M. S., "Minimum Ratio-Locked Profile Times for Robot Trajectories", *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 2, (1988): 1054-1060.
- [24] Geering, H. P., Guzzella, L., Hepner, A. R., Onder, C. H., "Time-Optimal Motions of Robots in Assembly Tasks", *IEEE Transactions on Automatic Control*, Vol. AC-31, No. 6, (1986): 512-518.
- [25] Gosselin, C., Hadj-Messaoud, A., "Automatic Planning of Smooth Trajectories for Pick-and-Place Operations", *Advances in Design Automation*, Vol. 2, (1991): 531-536.
- [26] Kahn, M.E., Roth, B., "The Near-Minimum-Time Control of Open-Loop Articulated Kinematic Chains", *Journal of Dynamic Systems, Measurement, and Control*, Vol. 93, (Sept. 1971): 164-172.
- [27] Kieffer, J., Cahill, A. J., James, M. R., "Robust and Accurate Time-Optimal Path-Tracking Control for Robot Manipulators", *IEEE Transactions on Robotics and Automation*, Vol. 13, No. 6, (1997): 880-890.
- [28] Kim, B. K., Shin, K. G., "Minimum-Time Path Planning for Robot Arms and Their Dynamics", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-15, No. 2, (1985): 213-223.
- [29] Kyriakopoulos, K. J., Saridis, G. N., "Minimum Jerk Path Generation", *Proceedings of the IEEE International Conference on Robotics and Automation*, (1988): 364-369.
- [30] Leahy, M. B., Saridis, G. N., "Compensation of Unmodeled PUMA Manipulator Dynamics", *IEEE International Conference on Robotics and Automation*, (1987): 151-156.

- [31] Lin, C.-S., Chang, P.R., "Joint Trajectories of Mechanical Manipulators for Cartesian Path Approximation", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-13, No. 6, (1983): 1094-1102.
- [32] Lin, C. S., Chang, P. R., Luh, J. Y. S., "Formulation and Optimization of Cubic Polynomial Joint Trajectories for Industrial Robots", *IEEE Transactions on Automatic Control*, Vol. AC-28, No. 12, (1983): 1066-1074.
- [33] Lloyd, J., Hayward, V., "Trajectory Generation for Sensor-Driven and Time-Varying Tasks", *International Journal of Robotics Research*, Vol. 12, No. 4, (1993): 380-393.
- [34] Müller-Karger, C., Granados Mirena, A., Scarpati López, J., "Hyperbolic Trajectories for Pick-and-Place Operations to Elude Obstacles", *IEEE Transactions on Robotics and Automation*, Vol. 16, No. 3, (2000): 294-300.
- [35] Paul, R. P., *Robot Manipulators: Mathematics, Programming, and Control*, MIT Press (1981).
- [36] Pfeiffer, F., Johanni, R., "A Concept for Manipulator Trajectory Planning", *IEEE Journal of Robotics and Automation*, Vol. RA-3, No. 2, (1987): 115-123.
- [37] Piazzoli, A., Visioli, A., "Global-Minimum-Jerk Trajectory Planning of Robot Manipulators", *IEEE Transactions on Industrial Electronics*, Vol. 47, No. 1, (2000): 140-149.
- [38] Press, W. H., Flannery, B., P., Vetterling, W., T., Teukolsky, S., A., *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, pp. 184-185, 1992.
- [39] Rajan, V. T., "Minimum Time Trajectory Planning", *IEEE International Conference on Robotics and Automation*, (1985): 759-764.
- [40] Red, E., "A dynamic Optimal Trajectory Generator for Cartesian Path Following", *Robotica*, Vol. 18, Part 5, (2000): 451-458.
- [41] Sahar G., Hollerbach, J.M., "Planning of Minimum-Time Trajectories for Robot Arms", *The International Journal of Robotics Research*, vol. 5, no. 3, (1986): 90-100.
- [42] Sciavicco L., Siciliano, B., *Modeling and Control of Robot Manipulators*, McGraw-Hill Companies, Inc. (1996).
- [43] Shiller, Z., Chang, H., Wong, V., "The Practical Implementation of Time-Optimal Control for Robotic Manipulators", *Robotics and Computer-Integrated Manufacturing*, Vol. 12, No. 1, (1996): 29-39.
- [44] Shin, K. G., McKay, N. D., "Minimum-Time Control of Robotic Manipulators with Geometric Path Constraints", *IEEE Transactions on Automatic Control*, Vol. AC-30, No. 6, (1985): 531-541.
- [45] Shin, K. G., McKay, N. D., A "Dynamic Programming Approach to Trajectory Planning of Robotic Manipulators", *IEEE Transactions on Automatic Control*, Vol. AC-31, No. 6, (1986): 491-500.
- [46] Tam, H. Y., "Minimum Time Closed-Loop Tracking of A Specified Path by Robot", *Proceedings of the 29th Conference on Decision and Control*, (1990): 3132-3137.

- [47] Taylor, R. H., "Planning and Execution of Straight Line Manipulator Trajectories", *IBM Journal of Research and Development*, Vol. 23, No. 4, (1979): 424- 436.
- [48] Tan, H. H., Potts, R. B., "Minimum Time Trajectory Planner for the Discrete Dynamic Robot Model With Dynamic Constraints", *IEEE Journal of Robotics and Automation*, Vol. 4, No. 2, (1988): 174- 185.
- [49] Tarkiainen, M., Shiller, Z., "Time Optimal Motions of Manipulators with Actuator Dynamics", *Proceedings of the IEEE Conference on Robotics and Automation*, Vol. 2, (1993): 725-730.
- [50] Volpe, R., "Task Space Velocity Blending for Real-Time Trajectory Generation", *Proceedings of the IEEE International Conference on Robotics and Automation*, v.2, (1993): 680-687.
- [51] Ward, J.P., *Quaternions and Cayley Numbers*, Kluwer Academic Publishers (1997).
- [52] Yang, D.C.H., Chou, J.J., "Automatic Generation of Piecewise Constant Speed Motion with Smooth Transition for Multi-Axis Machines", *Journal of Mechanical Design*, Vol. 116, (June 1994): 581- 586.

Appendix A

Jerk Considerations

A quintic specified by the sine wave template, such as the one shown in Figure A.1, has end conditions

$$p(0) = p_1, \quad (\text{A.1})$$

$$p(dt_{max}) = a_{max} dt_{max}^2 \left(\frac{1}{4} - \frac{1}{\pi^2} \right) + s_1 dt_{max} + p_1, \quad (\text{A.2})$$

$$s(0) = s_1, \quad (\text{A.3})$$

$$s(dt_{max}) = \frac{a_{max} dt_{max}}{2} + s_1, \quad (\text{A.4})$$

$$a(0) = 0, \quad (\text{A.5})$$

$$a(dt_{max}) = 0, \quad (\text{A.6})$$

where p_1 is the initial position, s_1 is the initial velocity, a_{max} is the acceleration limit, and dt_{max} is set to

$$dt_{max} = \frac{\pi a_{max}}{2 j_{max}} \quad (\text{A.7})$$

so that the jerk limit, j_{max} , will not be violated during an acceleration ramp. For the trajectory developed herein, formed of concatenated quintic segments, the largest jerk discontinuity occurs during an acceleration pulse. It will be shown below that the maximum jerk discontinuity is equal to 11% of the jerk limit.

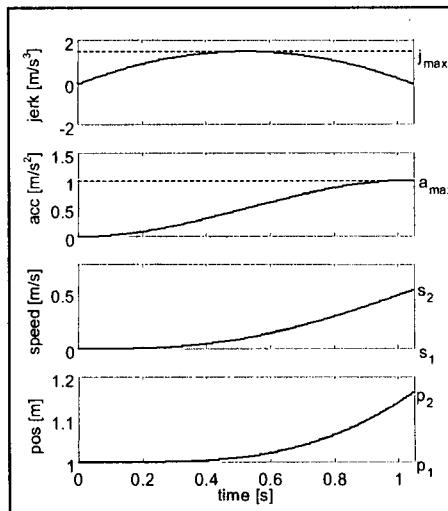


Figure A.1. Jerk, Acceleration, Velocity and Position Profiles Corresponding to a Sine-Based Acceleration Ramp from 0 to a_{max}

The distance covered during an acceleration ramp from zero to the maximum acceleration, a_{max} , based on the sine wave profile is

$$D_1 = p(dt_{max}) - p_1(0) = a_{max} dt_{max}^2 \left(\frac{1}{4} - \frac{1}{\pi^2} \right) + s_1 dt_{max}. \quad (\text{A.8})$$

The quintic coefficients corresponding to this quintic are:

$$b_5 = \left(4 - \frac{48}{\pi^2} \right) \frac{j_{max}^3}{\pi^3 a_{max}^2}, \quad (\text{A.9})$$

$$b_4 = \left(-5 + \frac{60}{\pi^2} \right) \frac{j_{max}^2}{\pi^2 a_{max}}, \quad (\text{A.10})$$

$$b_3 = \left(2 - \frac{20}{\pi^2} \right) \frac{j_{max}}{\pi}, \quad (\text{A.11})$$

$$b_2 = a_{max}, \quad (\text{A.12})$$

$$b_1 = s_1, \quad (\text{A.13})$$

$$b_0 = p_1. \quad (\text{A.14})$$

The equation corresponding to the jerk is:

$$j(t) = 60b_5 t^2 + 24b_4 t + 6b_3, \quad (\text{A.15})$$

and to the snap:

$$\chi(t) = 120b_5 t + 24b_4. \quad (\text{A.16})$$

Therefore, the maximum jerk may occur at either of the quintic extremities:

$$t_1 = 0, \quad (\text{A.17})$$

$$t_2 = dt_{max}, \quad (\text{A.18})$$

or when $\chi(t) = 0$:

$$t_3 = -\frac{b_4}{5b_5} = \frac{dt_{max}}{2}. \quad (\text{A.19})$$

It can be shown that at these points:

$$j(t_1) = 6b_3 = \frac{6}{\pi} \left(2 - \frac{20}{\pi^2} \right) j_{max} = -0.0505 j_{max}, \quad (\text{A.20})$$

$$j(t_2) = 60b_5 dt_{max}^2 + 24b_4 dt_{max} + 6b_3 = \frac{6}{\pi} \left(2 - \frac{20}{\pi^2} \right) j_{max} = -0.0505 j_{max}, \quad (\text{A.21})$$

$$j(t_3) = 15b_5 dt_{max}^2 + 12b_4 dt_{max} + 6b_3 = \left(-3 + \frac{60}{\pi^2} \right) \frac{j_{max}}{\pi} = 0.980 j_{max}. \quad (\text{A.22})$$

Therefore, the maximum jerk occurring during the acceleration ramp is equal to $0.980j_{max}$. In addition, the jerk at the start and the end of the quintic is equal to $-0.0505j_{max}$. Thus, for a trajectory formed of a concatenation of quintics, the worst discontinuity in jerk will occur during an acceleration pulse (a concatenation of two acceleration ramps). The magnitude of this discontinuity will be equal to $0.011j_{max}$ or 11% of the jerk limit.

However, as seen in Chapter 4, the jerk at the extremities of a blend can be equal to the jerk limit. Thus, when a straight-line segment is connected to a blend, there is a maximum jerk discontinuity of $1.05j_{max}$ during the motion.

Appendix B

Quintic End Conditions

B.1. Quintic End Conditions for a Trajectory Based on a Sustained Acceleration Pulse

The quintic end conditions (time, position, velocity, and acceleration) for a trajectory between two waypoints based on a sustained acceleration pulse are provided in Table B.1. The quintic control points are shown in Figure B.1 (taken from Chapter 3, cf. Figure 3.3). The Sustained Acceleration Pulse (SAP) algorithm is repeated below for clarity.

Sustained Acceleration Pulse Algorithm ($s_2 > s_1 \geq 0$)

1. Calculate the total way-point to way-point distance, $D = \|p_2 - p_1\|$.
2. Calculate the two unknown control point speeds s_a ,

$$s_a = s_1 + \frac{a_{max} dt_{max}}{2}, \quad (\text{B.1})$$

and s_b ,

$$s_b = s_2 - \frac{a_{max} dt_{max}}{2}. \quad (\text{B.2})$$

3. Calculate the distances covered during:

- the ramp up to a_{max} ,

$$D_1 = a_{max} dt_{max}^2 \left(\frac{1}{4} - \frac{1}{\pi^2} \right) + s_1 dt_{max}. \quad (\text{B.3})$$

- the acceleration cruise,

$$D_2 = \frac{s_b^2 - s_a^2}{2a_{max}}. \quad (\text{B.4})$$

- the ramp down from a_{max} to zero acceleration

$$D_3 = a_{max} dt_{max}^2 \left(\frac{1}{4} + \frac{1}{\pi^2} \right) + s_b dt_{max}. \quad (\text{B.5})$$

4. IF ($D_1 + D_2 + D_3 - D > 0$)

It is not possible to reach the demanded speed, s_2 , due to the distance, acceleration and jerk limitations. Therefore, reduce s_b and s_2 so that the distance constraint is respected (i.e., reduce the length of the acceleration cruise):

$$s_b = -a_{max} dt_{max} + \sqrt{\left(a_{max} dt_{max}\right)^2 - 2a_{max} \left(\frac{a_{max} dt_{max}^2}{2} + s_1 dt_{max} - \frac{s_a^2}{2a_{max}} - D\right)}, \quad (\text{B.6})$$

$$s_2 = s_b + \frac{a_{max} dt_{max}}{2}. \quad (\text{B.7})$$

Calculate D_2 and D_3 from Eqs B.4 and B.5.

5. Assign the times, positions (using D_1 , D_2 , and D_3), speeds (s_1 , s_a , s_b and s_2) and accelerations (zero or a_{max}) corresponding to the four quintic control points that form the non-zero acceleration portion of the motion.

(Control Point) i^1	time (t_{qi})	position (p_{qi})	velocity (v_{qi})	acceleration (a_{qi})
1	0	0	s_1	0
2	dt_{max}	D_1	s_a	a_{max}
3	$t_{q2} + (s_b - s_a)/a_{max}$	$D_1 + D_2$	s_b	a_{max}
4	$t_{q3} + dt_{max}$	$D_1 + D_2 + D_3$	s_2	0
5	$t_{q4} + (D - (D_1 + D_2 + D_3))/s_2$	D	s_2	0

Table B.1. Quintic Control Point Conditions for the SAP Algorithm

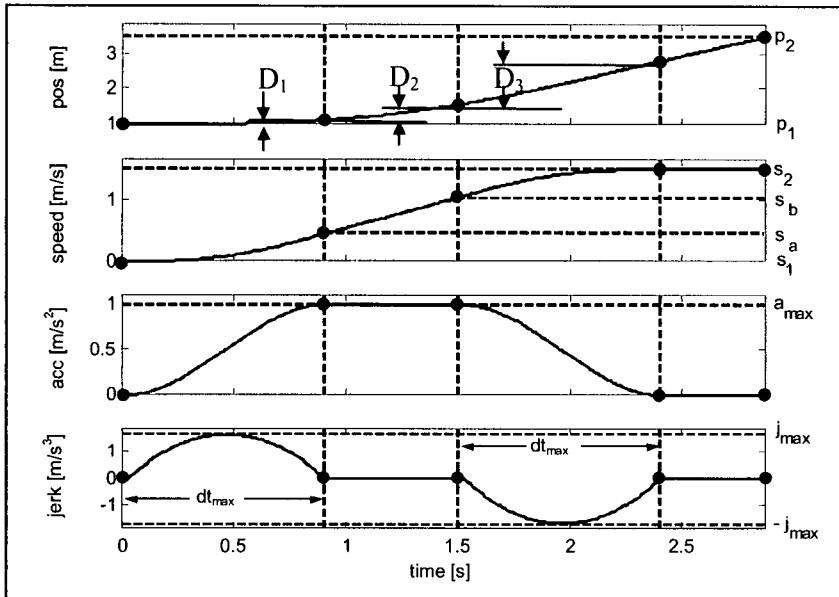


Figure B.1. Smooth Trajectory Generated with a Sequence of Quintics (• indicate the quintic control points)

B.2. Quintic End Conditions for a Trajectory Based on an Acceleration Pulse

The quintic end conditions (time, position, velocity, and acceleration) for a trajectory between two waypoints based on an acceleration pulse are provided in Table B.2. The quintic control points are shown in Figure B.2 (taken from Chapter 3, cf. Figure 3.4). The Acceleration Pulse algorithm is repeated below for clarity.

Acceleration Pulse Algorithm ($s_2 > s_1 \geq 0$)

1. IF ($ds < ds_{min}$ & $D < D_{min}$)
 - Calculate dt from Eq. 3.24, and a_{peak} from Eq. 3.25.
 - IF ($ds < a_{peak} dt$)
 - Recalculate dt from Eq. 3.26 and a_{peak} from Eq. 3.25.
 - ELSE IF ($D < D_{min}$)
 - Calculate dt from Eq. 3.24 and a_{peak} from Eq. 3.25.
 - ELSE IF ($ds < ds_{min}$)
 - Calculate dt from Eq. 3.26 and a_{peak} from Eq. 3.25.
2. Assign the time, position, speed and acceleration corresponding to the three quintic control points that form the acceleration pulse portion of the trajectory.

¹ Control point 5 is only necessary if a speed cruise is present (i.e. if $D - (D_1 + D_2 + D_3) > 0$)

(Control Point) i^2	time (t_{qi})	position (p_{qi}) ³	velocity (v_{qi})	acceleration (a_{qi})
1	0	0	s_1	0
2	dt	D_{q1}	$s_1 + a_{peak}dt/2$	a_{peak}
3	$2dt$	$D_{q1} + D_{q2}$	$s_1 + a_{peak}dt$	0
4	$t_{q3} + (D - p_{q3})/v_{q3}$	D	v_{q3}	0

Table B.2. Quintic Control Point Conditions for the Acceleration Pulse Algorithm

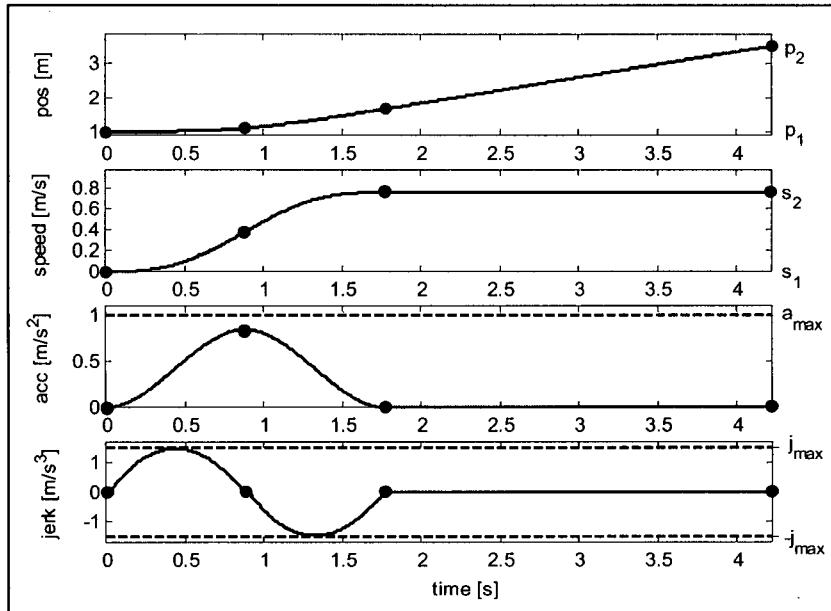


Figure B.2. Speed Limited Trajectory (• indicate the quintic control points)

B.3. Algorithm followed when $s_1 > s_2$

When the speed of the first way-point is greater than that of the second way-point, the two speeds are interchanged and the trajectory is calculated following the algorithm outlined in Section 3.2. The quintic control points are then modified to reflect the actual way-point speeds, as shown in the algorithm below. This algorithm is also used to modify the control points for a ramp down from maximum path speed, s_{peak} , to the speed at the end of the trajectory, s_2 , when s_{peak} is larger than s_2 (Section 3.2, Case B).

² Quintic control point 4 is only necessary for a speed limited case.

$$^3 D_{q1} = a_{max} dt^2 \left(\frac{1}{4} - \frac{1}{\pi^2} \right) + s_1 dt$$

$$D_{q2} = a_{max} dt^2 \left(\frac{1}{4} + \frac{1}{\pi^2} \right) + v_{q2} dt$$

Algorithm Used to Modify the Control Points for a Deceleration Trajectory

j = number of control points

FOR i=1:j,

1. Adjust the control point time:

$$t_{qi} = t_{qj} - t_{q(j-i+1)}$$

2. Adjust the control point position:

$$p_{qi} = p_{qj} - p_{q(j-i+1)}$$

3. Adjust the control point velocity:

$$v_{qi} = v_{q(j-i+1)}$$

4. Adjust the control point acceleration:

$$a_{qi} = -a_{q(j-i+1)}$$

END

Appendix C

Algorithm Proof

In this appendix it is shown that there is only one positive real root in dt (and two complex roots) for Eq. C.1 (Eq. 3.24), if $s_1 \geq 0$, $D > 0$ and $j_{max} > 0$.

$$dt^3 + \frac{\pi s_1}{j_{max}} dt - \frac{\pi D}{2 j_{max}} = 0, \quad (\text{C.1})$$

where D is the distance (positive, real, non-null, number), j_{max} is the jerk limit (positive, real, non-null, number) s_1 is the speed at the first way-point (positive real number). It is important that the real root be positive since dt is a time that must be greater than zero.

It has been shown that a cubic equation of the form

$$t^3 + kt^2 + lt + m = 0 \quad (\text{C.2})$$

has one real root and two complex roots if k , l , and m are real and $U^2 > V^3$ [38], where

$$U = \frac{2k^3 - 9kl + 27m}{54}, \quad (\text{C.3})$$

$$V = \frac{k^2 - 3l}{9}. \quad (\text{C.4})$$

For Eq. C.1,

$$k = 0, \quad (\text{C.5})$$

$$l = \frac{\pi s_1}{j_{max}}, \quad (\text{C.6})$$

$$m = -\frac{\pi D}{2 j_{max}}, \quad (\text{C.7})$$

(k , l , and m are real) and therefore,

$$U = -\frac{\pi D}{4 j_{max}}, \quad (\text{C.8})$$

$$V = -\frac{\pi s_1}{3 j_{max}}. \quad (\text{C.9})$$

If $s_l \geq 0$, $D > 0$ and $j_{max} > 0$, then

$$U^2 = \left(-\frac{\pi D}{4j_{max}} \right)^2 = \left(\frac{\pi D}{4j_{max}} \right)^2. \quad (\text{C.10})$$

and

$$V^3 = \left(-\frac{\pi s_l}{3j_{max}} \right)^3 = -\left(\frac{\pi s_l}{3j_{max}} \right)^3, \quad (\text{C.11})$$

Therefore, $U^2 > V^3$ and there is only one real root in dt for Eq C.1.

Furthermore, this real root is equal to

$$t = (L + P) - \frac{k}{3}, \quad (\text{C.12})$$

where

$$L = -\text{sign}(U) \left[|U| + \sqrt{U^2 - V^3} \right]^{\frac{1}{3}}, \quad (\text{C.13})$$

and

$$P = \begin{cases} \frac{V}{L}, & (L \neq 0) \\ 0, & (L = 0) \end{cases}. \quad (\text{C.14})$$

Therefore, for the equation under consideration (Eq. C.1),

$$L = \left[|U| + \sqrt{U^2 - V^3} \right]^{\frac{1}{3}}, \quad (\text{C.15})$$

and since $U^2 > V^3$, $L \geq |U|$, and the real root, dt_1 :

$$dt_1 = \left(L + \frac{V}{L} \right). \quad (\text{C.16})$$

If $V = 0$, then $L = (2|U|)^{1/3}$ and

$$dt_1 = \left(\frac{\pi D}{2j_{max}} \right)^{\frac{1}{3}}. \quad (\text{C.17})$$

In this case, $dt_1 > 0$ since both D and j_{max} are positive non-null numbers.

Otherwise $V < 0$, and $dt_1 > 0$ only if

$$L^2 > -V. \quad (\text{C.18})$$

This is in fact the case, since after intermediary calculations Eq C.18 can be transformed into

$$0 > 2V^{3/2}|U|. \quad (\text{C.19})$$

Since $V^{3/2} < 0$, Eq.C.19 holds and $dt_l > 0$. Thus, the single real root of Eq C.1 is always a positive real number as long as $s_l \geq 0$, $D > 0$ and $j_{max} > 0$.

Appendix D

Quintic Control Points for Case B (iiA)

This material pertains to Section 3.2, case B, subcase iiA. When the maximum speed, s_{max} cannot be reached due to distance limitations, it is necessary to shorten the acceleration cruises so that the distance, D , is respected. As defined in the section, D_x is the distance covered by a sustained acceleration pulse used to ramp up from s_1 to s_{max} , and D_z is the distance covered by a sustained acceleration pulse used to ramp down from s_{max} to s_2 . Since $D > D_{limit}$ (Eq. 3.30), two sustained acceleration pulses will be used to move as fast as possible between the two way-points, however, in this case, $D < D_x + D_z$, and therefore, the maximum speed reached, s_{peak} will be lower than s_{max} :

$$s_{peak} = -\frac{a_{max} dt_{max}}{2} + \sqrt{a_{max} D + \frac{s_{ax}^2 + s_{az}^2}{2} - a_{max} dt_{max} (s_1 + s_2)}. \quad (\text{D.1})$$

where s_{ax} and s_{az} were found using the SAP algorithm for each motion.

The speeds s_{ax} and s_{az} and the distances D_{lx} and D_{lz} remain the same as calculated using the SAP algorithm, however, the speeds s_{bx} and s_{bz} are changed as follows:

$$s_{bx} = s_{bz} = s_{peak} - \frac{a_{max} dt_{max}}{2}. \quad (\text{D.2})$$

The distances D_{2x} , D_{2z} , D_{3x} and D_{3z} are also modified:

$$D_{2x} = \frac{s_{bx}^2 - s_{ax}^2}{2a_{max}}, \quad (\text{D.3})$$

$$D_{3x} = a_{max} dt_{max}^2 \left(\frac{1}{4} + \frac{1}{\pi^2} \right) + s_{bx} dt_{max}, \quad (\text{D.4})$$

$$D_{2z} = \frac{s_{bz}^2 - s_{az}^2}{2a_{max}}, \quad (\text{D.5})$$

$$D_{3z} = a_{max} dt_{max}^2 \left(\frac{1}{4} + \frac{1}{\pi^2} \right) + s_{bz} dt_{max}. \quad (\text{D.6})$$

This provides the information necessary to calculate the quintic control points for the motion of case 2(iiA) when $D < D_x + D_z$.

Appendix E

Blend Acceleration and Jerk Limits

It was stated in Chapter 4 that the maximum acceleration and jerk attained during a blend are

$$a_{blendMax} = \frac{s^2}{\rho \cdot tightness} \sqrt{\frac{1 + \cos\theta}{2}}, \quad (E.1)$$

$$j_{blendMax} = \frac{15s^3}{2tightness^2} \cos\left(\frac{\theta}{2}\right), \quad (E.2)$$

where, s is the speed, ρ is a proportionality constant, *tightness* indicates how close to the point the path must pass, and θ is as shown in Figure E.1. The derivation of these two equations is provided in this Appendix.

Figure 4.3 is reproduced here for reference purposes (Figure E.1). This appendix assumes that the blend under consideration is a geometric blend as outlined in Section 4.2.

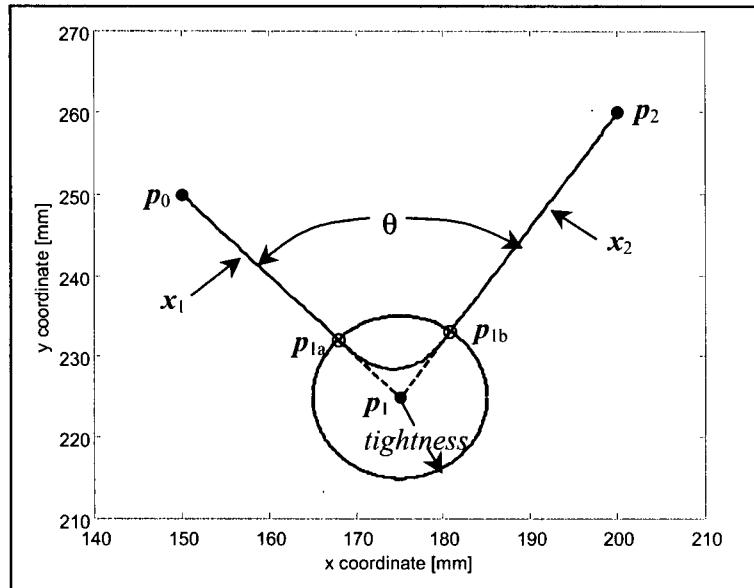


Figure E.1. 2D Blend Geometry

E.1. Constant of Proportionality

In this section it will be shown that there is a constant relationship between the maximum acceleration during a geometric blend and the constant absolute acceleration required to transit between the initial and final blend velocities. If the change in velocities were done linearly, the corresponding acceleration would be constant, $a_{constant}$. Establishing the relationship between $a_{constant}$ and the maximum blend acceleration, $a_{blendMax}$, is of interest since the constant acceleration equations of motion could then be used to determine a relationship between the maximum permitted acceleration, a_{max} , and the change in velocity during the blend. It will be shown below that the magnitude of $a_{constant}$ is directly proportional to the magnitude of $a_{blendMax}$ through the proportionality constant, ρ :

$$a_{constant} = \rho \cdot a_{blendMax} \quad (\text{E.3})$$

The equation for a blend between two straight lines x_1 and x_2 , is reproduced here from Chapter 4 (Eq. 4.7),

$$x(\sigma) = x_1(\sigma) + \alpha(\sigma)(x_2(\sigma) - x_1(\sigma)) - \kappa\beta(\sigma)v_d, \quad (\text{E.4})$$

where in the geometric blend case,

$$x_1(\sigma) = b_1 + m_1 \sigma, \quad (\text{E.5})$$

$$x_2(\sigma) = b_2 + m_2 \sigma, \quad (\text{E.6})$$

$$\alpha(\sigma) = 6\sigma^5 - 15\sigma^4 + 10\sigma^3, \quad (\text{E.7})$$

$$\beta(\sigma) = \sigma^6 - 3\sigma^5 + 3\sigma^4 - \sigma^3, \quad (\text{E.8})$$

$$v_d = m_2 - m_1, \quad (\text{E.9})$$

$$b_1 = p_{1a}, \quad b_2 = p_{1b} - 2\tau v_{1b}, \quad (\text{E.10})$$

$$m_1 = 2\tau v_{1a}, \quad m_2 = 2\tau v_{1b}, \quad (\text{E.11})$$

and 2τ is the blend time.

Initially, the location and magnitude of the maximum blend acceleration must be determined. This requires developing the equation describing the jerk in the σ -domain. Taking the first, second and third derivatives of $x(\sigma)$ with respect to σ , leads to the following equations (from which the null terms have been removed):

$$v(\sigma) = \dot{x}(\sigma) = \dot{x}_1(\sigma) + \alpha(\sigma)(\dot{x}_2(\sigma) - \dot{x}_1(\sigma)) + \dot{\alpha}(\sigma)(x_2(\sigma) - x_1(\sigma)) - \kappa\dot{\beta}(\sigma)v_d, \quad (\text{E.12})$$

$$a(\sigma) = \ddot{x}(\sigma) = 2\dot{\alpha}(\sigma)(\dot{x}_2(\sigma) - \dot{x}_1(\sigma)) + \ddot{\alpha}(\sigma)(x_2(\sigma) - x_1(\sigma)) - \kappa\ddot{\beta}(\sigma)v_d, \quad (\text{E.13})$$

$$\mathbf{j}(\sigma) = \ddot{\mathbf{x}}(\sigma) = 3\ddot{\alpha}(\sigma)(\dot{x}_2(\sigma) - \dot{x}_1(\sigma)) + \ddot{\alpha}(\sigma)(x_2(\sigma) - x_1(\sigma)) - \kappa\ddot{\beta}(\sigma)\mathbf{v}_d. \quad (\text{E.14})$$

Substituting Eqs. E.5-E.9 and their derivatives into Eq. E.14

$$\begin{aligned} j(\sigma) = & 60(b_2 - b_1) + 45(m_2 - m_1) + \sigma[-360(b_2 - b_1) - 300(m_2 - m_1)] + \\ & \sigma^2[360(b_2 - b_1) + 450(m_2 - m_1)] - 180\sigma^3(m_2 - m_1) \end{aligned} \quad (\text{E.15})$$

Therefore, the local acceleration maxima and minima are located at :

$$\sigma_1 = \frac{1}{2} + \frac{1}{6}\sqrt{3}, \quad (\text{E.16})$$

$$\sigma_2 = \frac{1}{2} - \frac{1}{6}\sqrt{3}, \quad (\text{E.17})$$

$$\sigma_3 = \frac{4(b_2 - b_1) + 3(m_2 - m_1)}{2(m_2 - m_1)}, \quad (\text{E.18})$$

and the two extremities $\sigma=0$, $\sigma=1$. These locations are plotted on Figure E.2.

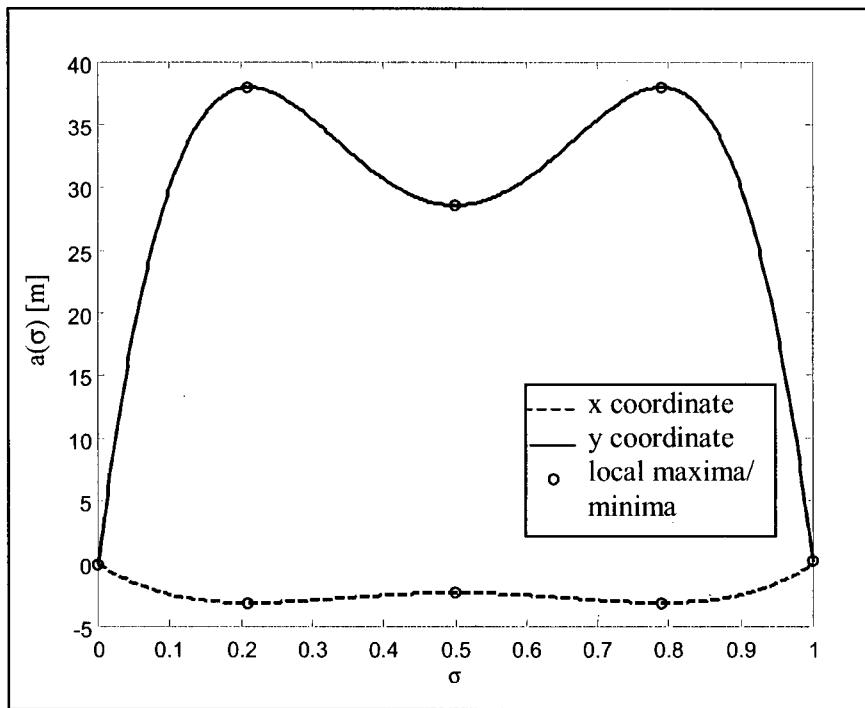


Figure E.2. Plot of Acceleration Profile for the 2D Blend of Figure E.1.

Solving for the numerical value of σ_3 requires defining b_2-b_1 and m_2-m_1 in terms of the blend positions and velocities. Given that a geometric blend is being considered,

$$\mathbf{v}_{lb} - \mathbf{v}_{la} = s \left(\frac{\mathbf{p}_{lb} - 2\mathbf{p}_l + \mathbf{p}_{la}}{tightness} \right), \quad (\text{E.19})$$

and from Eq. 4.19,

$$\tau = \frac{tightness}{s}, \quad (\text{E.20})$$

then

$$\mathbf{v}_{lb} - \mathbf{v}_{la} = \frac{(\mathbf{p}_{lb} - 2\mathbf{p}_l + \mathbf{p}_{la})}{\tau}. \quad (\text{E.21})$$

Thus, from Eqs. E. 10, E. 11 and E.21 ,

$$\mathbf{b}_2 - \mathbf{b}_1 = \mathbf{p}_{lb} - 2\tau \mathbf{v}_{lb} - \mathbf{p}_{la} = -\mathbf{p}_{lb} + 2\mathbf{p}_l - \mathbf{p}_{la}, \quad (\text{E.22})$$

$$\mathbf{m}_2 - \mathbf{m}_1 = 2\tau(\mathbf{v}_{lb} - \mathbf{v}_{la}) = 2(\mathbf{p}_{lb} - 2\mathbf{p}_l + \mathbf{p}_{la}). \quad (\text{E.23})$$

Substitution of Eqs E.22 and E.23 into Eq. E.18, gives $\sigma_3 = 0.5$.

Thus, provided that all the conditions stipulated for a geometric blend are met, the local acceleration maxima and minima always occur at the same path location and corresponding time location in every dimension (as can be seen in Figure E.2). Substituting the five σ values corresponding to the local maxima and minima locations into the acceleration equation (E.13) leads to the equations corresponding to the values of the local maxima and minima :

$$a(\sigma_1) = 5.774(\mathbf{b}_2 - \mathbf{b}_1) + 4.137(\mathbf{m}_2 - \mathbf{m}_1) = 2.500(\mathbf{p}_{lb} - 2\mathbf{p}_l + \mathbf{p}_{la}), \quad (\text{E.24})$$

$$a(\sigma_2) = -5.774(\mathbf{b}_2 - \mathbf{b}_1) - 1.637(\mathbf{m}_2 - \mathbf{m}_1) = 2.500(\mathbf{p}_{lb} - 2\mathbf{p}_l + \mathbf{p}_{la}), \quad (\text{E.25})$$

$$a(\sigma_3) = 0.938(\mathbf{m}_2 - \mathbf{m}_1) = 1.876(\mathbf{p}_{lb} - 2\mathbf{p}_l + \mathbf{p}_{la}), \quad (\text{E.26})$$

$$a(0) = a(1) = 0. \quad (\text{E.27})$$

The absolute acceleration profile has two equal local maxima and one local minimum (assuming a positive acceleration profile). The acceleration values of interest are those corresponding to $a(\sigma_1)$ and $a(\sigma_2)$, since these are the maximum absolute accelerations occurring during a blend (Figure E.2). That is

$$a_{blendMax} = 2.500(\mathbf{p}_{lb} - 2\mathbf{p}_l + \mathbf{p}_{la}). \quad (\text{E.28})$$

The maximum path blend acceleration is obtained by taking the magnitude of Eq. E.28.

The constant acceleration vector, when accelerating linearly from \mathbf{m}_1 to \mathbf{m}_2 is:

$$\mathbf{a}_{constant} = \frac{\mathbf{m}_2 - \mathbf{m}_1}{1-0} = \mathbf{m}_2 - \mathbf{m}_1. \quad (\text{E.29})$$

Therefore, from Eqs. E.3, E.23, E.28, and E.29, the relationship between the constant acceleration and the maximum blend acceleration vectors is

$$\rho = \frac{\mathbf{a}_{constant}}{\mathbf{a}_{blendMax}} = \frac{2.000(\mathbf{p}_{lb} - 2\mathbf{p}_1 + \mathbf{p}_{la})}{2.500(\mathbf{p}_{lb} - 2\mathbf{p}_1 + \mathbf{p}_{la})} = 0.800. \quad (\text{E.30})$$

That is, each element of $\mathbf{a}_{blendMax}$ is proportional to its corresponding element in $\mathbf{a}_{constant}$ by ρ .

E.2. Derivation of the Relationship Between the Maximum Acceleration and the Speed for a Geometric Blend

Based on a constant acceleration motion between \mathbf{v}_{la} and \mathbf{v}_{lb} , and dividing the acceleration into x and y components as shown in Figure E.3,

$$a_x = \frac{s^2(1 + \cos\theta)}{2tightness}, \quad (\text{E.31})$$

and

$$a_y = \frac{s^2 \sin\theta}{2tightness}. \quad (\text{E.32})$$

Then, the magnitude of the constant acceleration is :

$$a_{constant} = \frac{s^2}{tightness} \sqrt{\frac{1 + \cos\theta}{2}}. \quad (\text{E.33})$$

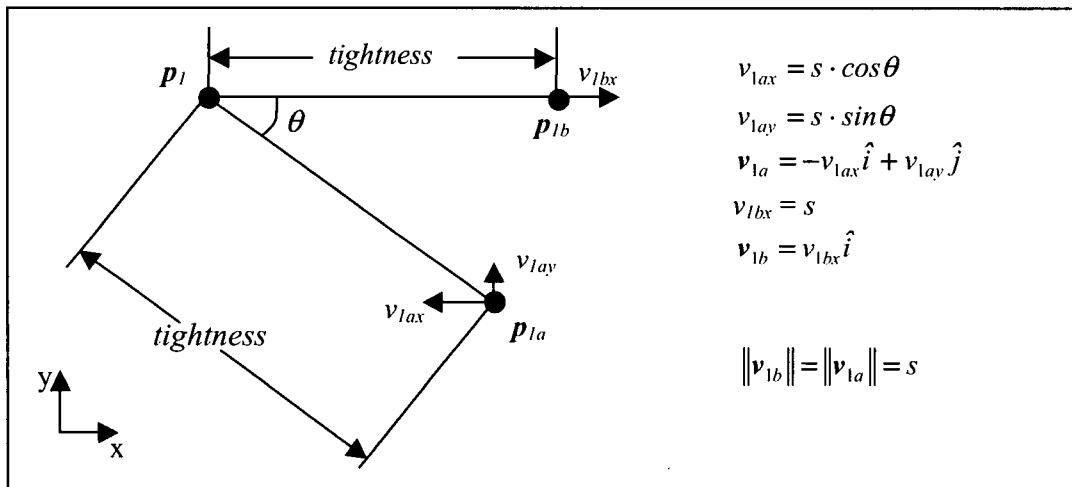


Figure E.3. Figure Illustrating the Variables Used in Equations E.31 and E.32

Thus, using the result from Section E.1, (Eq. E.30) and Eq. E.33 the magnitude of the maximum blend acceleration is defined as:

$$a_{blendMax} = \frac{s^2}{\rho \cdot tightness} \sqrt{\frac{1 + \cos \theta}{2}}. \quad (\text{E.34})$$

To ensure that the magnitude of the maximum blend acceleration is no larger than that allowed (a_{max}), it is possible to limit the incoming/outgoing speed of the blend, i.e.,

$$s \leq \sqrt{\frac{\rho \cdot a_{max} \cdot tightness}{\sqrt{\frac{1 + \cos \theta}{2}}}}, \quad (\text{E.35})$$

so that the maximum allowable acceleration, a_{max} , is respected during a *geometric* blend.

E.3. Derivation of the Relationship Between the Maximum Jerk and the Speed for a Geometric Blend

The relationship between the maximum jerk and the incoming and outgoing speed of the blend can be derived much in the same manner as in Section E.2. In this case, to find the location of the local maximum we must set the derivative of jerk with respect to the parameter σ , namely the snap, χ , to zero and solve for the corresponding roots :

$$\chi(\sigma) = \frac{d^4 x(\sigma)}{d\sigma^4} = 4\ddot{\alpha}(\sigma)(\dot{x}_2(\sigma) - \dot{x}_1(\sigma)) + \frac{d^4 \alpha(\sigma)}{d\sigma^4}(x_2(\sigma) - x_1(\sigma)) - \kappa \frac{d^4 \beta(\sigma)}{d\sigma^4}(\sigma)v_d. \quad (\text{E.36})$$

Substituting Eqs. E.5-E.9 and their derivatives into Eq. E.36,

$$\chi(\sigma) = -360(b_2 - b_1) - 300(m_2 - m_1) + \sigma[720(b_2 - b_1) + 900(m_2 - m_1)] - 540\sigma^2(m_2 - m_1). \quad (\text{E.37})$$

The roots of Eq. E.37 are

$$\sigma_{1,2} = \frac{12(b_2 - b_1) + 15(m_2 - m_1) \pm \sqrt{16(b_2 - b_1)^2 + 16(b_2 - b_1)(m_2 - m_1) + 5(m_2 - m_1)^2}}{18(m_2 - m_1)}. \quad (\text{E.38})$$

When Eqs. E.22 and E.23 are substituted into Eq. E.38, it can be shown that

$$\sigma_1 = \frac{1}{3}, \sigma_2 = \frac{2}{3}. \quad (\text{E.39})$$

There are four locations at which the jerk may be maximum : σ_1 , σ_2 and the blend extremities (i.e. $\sigma=0$ and $\sigma=1$). From Eq. E.15, E.22 and E.23, the value of the jerk at these locations can be found to be :

$$j(0)=30(p_{1b}-2p_1+p_{1a}), \quad (\text{E.40})$$

$$j\left(\frac{1}{3}\right)=-\frac{10}{3}(p_{1b}-2p_1+p_{1a}), \quad (\text{E.41})$$

$$j\left(\frac{2}{3}\right)=\frac{10}{3}(p_{1b}-2p_1+p_{1a}), \quad (\text{E.42})$$

and

$$j(1)=-30(p_{1b}-2p_1+p_{1a}). \quad (\text{E.43})$$

These locations are plotted on Figure E.4.

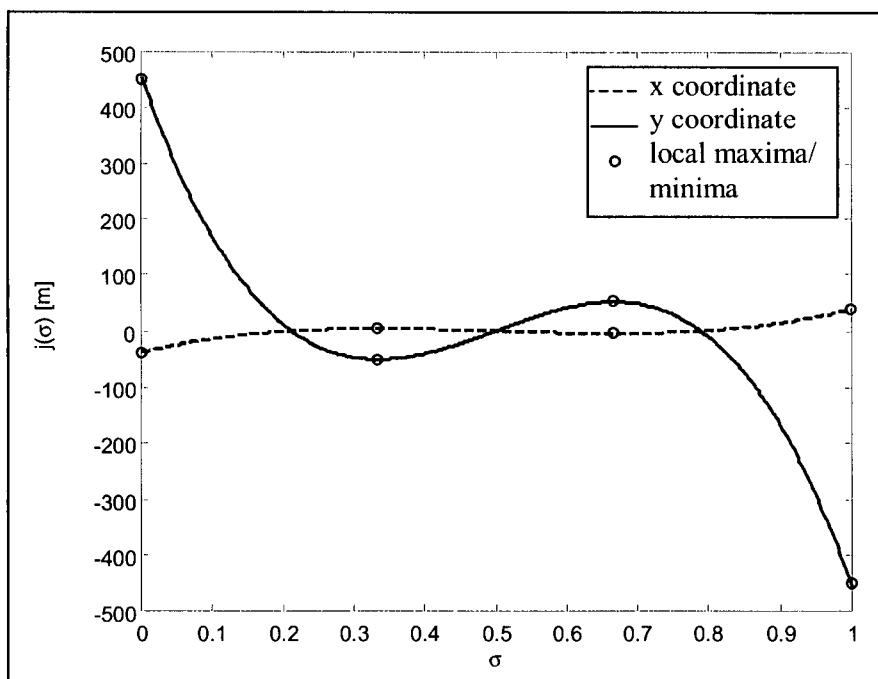


Figure E.4. Jerk Profiles for the Blend of Figure E.1

The maximum absolute jerks always occurs at the extremities, that is, at the start and the end of the blend (Figure E.4). To find the relationship between the speed and the magnitude of the jerk in the time domain, it is necessary to convert the jerk and velocity from the σ -domain to the time domain using the identity

$$j(t)=\frac{j(\sigma)}{8\tau^3}, \quad (\text{E.44})$$

and from Eq.23

$$m_2 - m_1 = 2\tau(v_{1b} - v_{1a}), \quad (\text{E.45})$$

giving

$$\frac{j(\sigma=0)}{m_2 - m_1} = 4\tau^2 \frac{j(t=0)}{v_{lb} - v_{la}} = 15. \quad (\text{E.46})$$

The magnitude of the maximum blend jerk is then

$$j_{blendMax} = \frac{15\|v_{lb} - v_{la}\|}{4\tau^2}. \quad (\text{E.47})$$

However, from Eqs. E.20 and E.21:

$$j_{blendMax} = \frac{15s^3\|p_{lb} - 2p_i + p_{la}\|}{4tightness^3}. \quad (\text{E.48})$$

Given that

$$\|p_{lb} - 2p_i + p_{la}\| = \|p_{lb} - p_i + p_{la} - p_i\| = 2\sqrt{tightness^2 - \frac{\|p_{lb} - p_{la}\|^2}{4}}, \quad (\text{E.49})$$

and

$$\cos\left(\frac{\theta}{2}\right) = \frac{\sqrt{tightness^2 - \frac{\|p_{lb} - p_{la}\|^2}{4}}}{tightness}, \quad (\text{E.50})$$

(Figure E.5), it can be shown that

$$j_{blendMax} = \frac{15s^3 \cos\left(\frac{\theta}{2}\right)}{2tightness^2}. \quad (\text{E.51})$$

Therefore,

$$s \leq \left(\frac{2 \cdot j_{max} \cdot tightness^2}{15 \cos\left(\frac{\theta}{2}\right)} \right)^{\frac{1}{3}} \quad (\text{E.52})$$

so that the maximum jerk during the blend is no more than the maximum allowable jerk, j_{max} .

One can note that either Eqs. E.48 or Eq. E.51 may be used to limit the jerk. Use of Eq. E.48 is preferred since it does not require the computation of an inverse cosine.

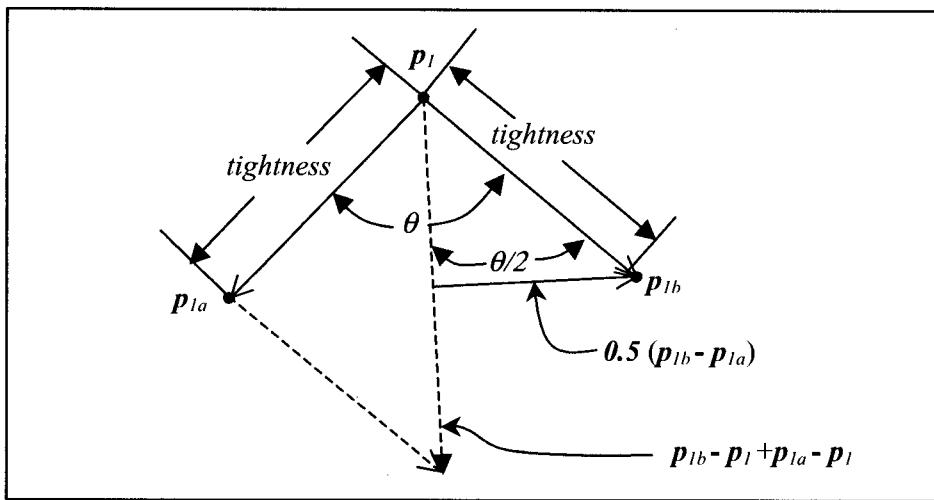


Figure E.5. Diagram Indicating the Geometry Used to Derive Eq. E.51

Appendix F

Estimation of Task Space Kinematic Limits and Calculation of Joint Space Path Limits

The problem of determining reliable joint and Cartesian space kinematic limits has not been thoroughly treated in literature. Kinematic limits (as derived from worst case dynamic limitations) are highly configuration dependent. Most authors develop their work based on the assumption that such limits can be found or estimated, but do not provide details. The manipulator kinematic limits derived using the methods described below are simply estimates. They will not ensure that the manipulator dynamic limits are respected due to the high configuration dependence. However, the estimates are conservative enough that the manipulator should generally be within its physical capacity, while not being so conservative that the manipulator's capacity is underused.

F.1. Joint Space Kinematic Limits

The joint space kinematic limits necessary for the trajectory generation algorithm consist of the maximum *path* speed, acceleration and jerk.

Actuator torque and speed limits can be obtained from manufacturer data sheets. The manufacturer speed limit provides the actuator speed limit, while the joint actuator acceleration limit can be derived through use of the manipulator Lagrangian dynamic equation:

$$M(\gamma)\ddot{\gamma} + \dot{\gamma}^T C(\gamma)\dot{\gamma} + G(\gamma) = T. \quad (\text{F.1})$$

where $\gamma \in \mathbf{R}^n$ is the vector of joint positions (n corresponds to the number of joints), $T \in \mathbf{R}^n$ is the vector of actuator torques, $M(\gamma) \in \mathbf{R}^{n \times n}$ is the inertia matrix of the manipulator, $C(\gamma) \in \mathbf{R}^{n \times n \times n}$ is a third order tensor representing the coefficients of the centrifugal and coriolis forces, and $G(\gamma) \in \mathbf{R}^n$ is the vector of gravity terms.

To determine the joint space acceleration limit, $\ddot{\gamma}_{max}$, the actuator torque and speed limit are used in

$$\ddot{\gamma} = M(\gamma)^{-1} [T - \dot{\gamma}^T C(\gamma)\dot{\gamma} - G(\gamma)]. \quad (\text{F.2})$$

and the joint angles are incremented. The minimum acceleration for each joint is used to determine the maximum allowable acceleration vector. Alternatively, an optimization technique can be used to determine the limits.

Generally, the actuator torque rate limits are not provided by the manufacturer. In fact, the choice of the jerk limit is a trade-off between the smoothness of the motion and the motion time. As the jerk limit is increased, the trajectory starts to resemble an LSPB trajectory causing poorer tracking. However, too low a jerk limit compromises the speed of the motion without providing much increase in tracking accuracy. Therefore, the joint jerk limits are set based on an estimate of the time it should take to ramp up to the maximum joint acceleration, Δt_{jerk} ,

$$\ddot{\gamma}_{max} = \frac{\dot{\gamma}_{max}}{\Delta t_{jerk}}. \quad (F.3)$$

These manipulator-specific limits are the individual joint limits, and must only be determined once. The path limits are computed as described in Section F.1.1.

F.1.1. Joint Space Path Limits

Herein, the manipulator trajectory is generated based on joint space straight-line paths joined by blends (Chapter 5). The trajectory generation algorithm requires one-dimensional kinematic limits along the joint space path since the trajectory is initially designed in one dimension (Chapter 3). However, for industrial systems, the joint space limits are given in terms of the individual joints, rather than as a single number applicable to the entire path, as is commonly done for the task space limits. The particular joint that imposes the limit will differ depending on the straight line or blend motion. Thus, the one-dimensional kinematic limits must be generated for each trajectory segment. This is done through projection of the individual joint limits into the space of the trajectory path, as explained below. It should be noted that the speed, acceleration and jerk along the path may be limited by different joints.

a. Projection for Straight Line Segments

The projection factor for joint i between two way-points p_0 and p_1 is

$$Proj_i = \frac{|p_1(i) - p_0(i)|}{\|p_1 - p_0\|}. \quad (F.4)$$

The speed limits at the start and the end of the straight line, the maximum speed limit, the acceleration limit and the jerk limit for joint i are obtained by dividing the nominal joint limit by the projection factor (Figure F.1):

$$\text{Path Kinematic Limit}_i = \frac{\text{Kinematic Limit}_i}{\text{Proj}_i}. \quad (\text{F.5})$$

The path joint limits are the minima of the limits obtained for all the joints, namely:

$$\text{Path Joint Limit} = \min(\text{Path Kinematic Limit}), \quad (\text{F.6})$$

where **min** is a function that chooses the minimum Path Kinematic Limit.

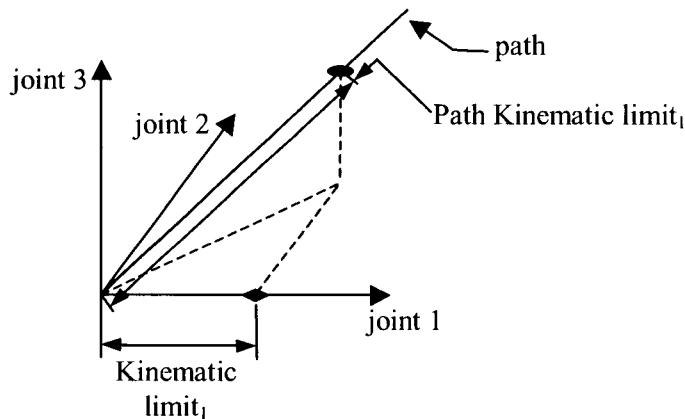


Figure F.1. Kinematic Limit Projection for a 3D Straight Line Path

b. Projection for Blend Segments

The geometric blend segments only require computation of acceleration and jerk limits since the maximum speed is fixed by the straight lines joined by the blend (the maximum speed in a blend occurs at its end points). These limits are computed in a similar manner to that described in (a), except that in this case p_f is the point at the end of the blend and p_0 is the point that is being blended around.

F.2. Task Space Path Kinematic Limits

The task space kinematic limits necessary for the trajectory generation algorithm consist of the maximum **Cartesian path** speed, acceleration and jerk, and the maximum **rotational path** speed, acceleration and jerk limits.

These limits are computed from the joint space limits based on differential kinematics. The task space speed is related to the joint space speed through

$$\boldsymbol{v} = \mathbf{J}(\boldsymbol{\gamma})\dot{\boldsymbol{\gamma}}, \quad (\text{F.7})$$

where $\mathbf{J}(\boldsymbol{\gamma}) \in \mathbf{R}^{n \times n}$ the Jacobian matrix. Through differentiation with respect to time, it can be shown that the task space acceleration is related to the joint space speed and acceleration through

$$\mathbf{a} = \mathbf{J}(\boldsymbol{\gamma})\ddot{\boldsymbol{\gamma}} + \dot{\mathbf{J}}(\boldsymbol{\gamma})\dot{\boldsymbol{\gamma}} . \quad (\text{F.8})$$

The limits are determined through iteration of Eqs. F.7 and F.8 based on varying the manipulator configuration while $\ddot{\boldsymbol{\gamma}} = \ddot{\boldsymbol{\gamma}}_{max}$ and $\dot{\boldsymbol{\gamma}} = \dot{\boldsymbol{\gamma}}_{max}$. The Cartesian space speed and acceleration limits are chosen as the minimum of the average speeds and accelerations in the x, y and z directions, respectively (since the motion could be along any of the axes). The orientation limits are taken to be the minimum of the kinematic limits of joints 4, 5 and 6.

The maximum jerk limit is set based on the estimated time it should take to ramp up to the maximum task space acceleration, a_{max} , Δt_{jerk} ,

$$j_{max} = \frac{a_{max}}{\Delta t_{jerk}} . \quad (\text{F.9})$$

As opposed to the joint space limits, the task space limits are only determined once (i.e. no projection onto the path is used), as is common in industrial trajectory generation routines.

Appendix G

Quaternions

G.1. Rotation Matrices and Quaternions

The change in end effector orientation between two way-points (e.g. p_0 and p_1) can be represented in the form of a rotation matrix:

$$R_{01} = R_{w0}^{-1} R_{w1} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, \quad (\text{G.1})$$

where R_{w0} and R_{w1} are the rotation matrices corresponding to the orientations with respect to a world frame, F_w , at way-points p_0 and p_1 respectively. Every rotation can also be expressed using an axis-angle formulation [42]. Thus, to move smoothly about a single axis (i.e. not “wobbling”) between the end effector orientations at p_0 and p_1 , the end effector frame must rotate about the axis of the rotation matrix, R_{01} :

$$\mathbf{f} = \frac{1}{2\sin\phi_2} \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix}, \quad (\text{G.2})$$

by an angle,

$$\phi_2 = \cos^{-1}\left(\frac{r_{11} + r_{22} + r_{33} - 1}{2}\right). \quad (\text{G.3})$$

This can also be represented in quaternion form as:

$$Z = \begin{bmatrix} \cos\left(\frac{\phi_2}{2}\right) \\ f_x \sin\left(\frac{\phi_2}{2}\right) \\ f_y \sin\left(\frac{\phi_2}{2}\right) \\ f_z \sin\left(\frac{\phi_2}{2}\right) \end{bmatrix} = \begin{bmatrix} Z_w \\ Z_x \\ Z_y \\ Z_z \end{bmatrix}. \quad (\text{G.4})$$

Z is a unit quaternion because the sum of the squares of its four components is equal to 1.

To transform a unit quaternion into a rotation matrix, the following equation is used [42]:

$$R = \begin{bmatrix} f_x^2(1 - c_{\phi_2}) + c_{\phi_2} & f_x f_y (1 - c_{\phi_2}) - f_z s_{\phi_2} & f_x f_z (1 - c_{\phi_2}) + f_y s_{\phi_2} \\ f_x f_y (1 - c_{\phi_2}) + f_z s_{\phi_2} & f_y^2 (1 - c_{\phi_2}) + c_{\phi_2} & f_y f_z (1 - c_{\phi_2}) - f_x s_{\phi_2} \\ f_x f_z (1 - c_{\phi_2}) - f_y s_{\phi_2} & f_y f_z (1 - c_{\phi_2}) + f_x s_{\phi_2} & f_z^2 (1 - c_{\phi_2}) + c_{\phi_2} \end{bmatrix}, \quad (\text{G. 5})$$

where $c_{\phi_2} = \cos(\phi_2)$ and $s_{\phi_2} = \sin(\phi_2)$, and from Eq. G.4,

$$\phi_2 = 2\cos^{-1}(Z_w), \quad (\text{G. 6})$$

and

$$\begin{aligned} f_x &= \frac{Z_x}{\sin\left(\frac{\phi_2}{2}\right)} \\ f_y &= \frac{Z_y}{\sin\left(\frac{\phi_2}{2}\right)} \\ f_z &= \frac{Z_z}{\sin\left(\frac{\phi_2}{2}\right)} \end{aligned} \quad (\text{G. 7})$$

A computationally efficient manner of calculating Eq. G.5 is as follows:

$$R = f \cdot f^T + \cos(\phi_2) \cdot \left(\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - f \cdot f^T \right) + \sin(\phi_2) \cdot S(f), \quad (\text{G. 8})$$

where f^T is the transpose of vector f and $S(f)$ is the skew symmetric matrix corresponding to f .

G.2. Quaternion Multiplication and Transpose

A quaternion can also be described as:

$$Z = Z_w + Z_x \hat{i} + Z_y \hat{j} + Z_z \hat{k}, \quad (\text{G. 9})$$

where $\hat{i}^2 = \hat{j}^2 = \hat{k}^2 = -1$ and $\hat{i}\hat{j} = \hat{k}$, $\hat{j}\hat{k} = -\hat{i}$ [51, p.62]. Thus, the multiplication of two unit quaternions H and Z can be calculated as:

$$HZ = [Z^-]H, \quad (\text{G. 10})$$

where

$$Z^- = \begin{bmatrix} Z_w & -Z_x & -Z_y & -Z_z \\ Z_x & Z_w & Z_z & -Z_y \\ Z_y & -Z_z & Z_w & Z_x \\ Z_z & Z_y & -Z_x & Z_w \end{bmatrix}. \quad (\text{G. 11})$$

The transpose of a unit quaternion is simply:

$$Z^T = \begin{bmatrix} Z_w \\ -Z_x \\ -Z_y \\ -Z_z \end{bmatrix}. \quad (\text{G.12})$$

G.3. Algorithm Used to Generate a Rotation Matrix at Each Sampling Interval

As described in Chapter 5, a quintic trajectory, $\phi(t)$, that starts at 0 and ends at ϕ_2 , smoothly interpolates between the orientations p_0 and p_1 . From this, the rotation matrix describing the orientation of each sampling instant point with respect to the world frame is generated for use in the inverse kinematics. The algorithm used to generate the rotation matrix with respect to the world frame at every sampling instant is provided below. It should be noted that the manipulations could also have been done using an axis-angle representation rotation matrix instead of the quaternion form.

Algorithm To Generate a Rotation Matrix at Each Sampling Interval

1. Compute Z_{w0} and Z_{wl} , which correspond respectively to the rotation matrices of p_0 and p_l (Eq. G.4).
2. Compute Z_{0l} , the quaternion corresponding to the change in orientation between p_0 and p_l (Eqs. G.10 and G.12):

$$Z_{0l} = Z_{w0}^T Z_{wl} \quad (\text{G.13})$$

3. Calculate the normalization constant:

$$N_{\text{const}} = Z_{0l_x}^2 + Z_{0l_y}^2 + Z_{0l_z}^2. \quad (\text{G.14})$$

4. FOR each quintic trajectory point, $\phi(i)$,

- a. Calculate the first quaternion element,

$$Z_w = \cos\left(\frac{\phi(i)}{2}\right). \quad (\text{G.15})$$

- b. IF ($1 - Z_w^2 > 0$)

$$K = \sqrt{\frac{N_{\text{const}}}{1 - Z_w^2}} \quad (\text{G.16})$$

ELSE

$$K = 1 \quad (\text{G.17})$$

END

- c. Compute the quaternion corresponding to a rotation of $\phi(i)$ about axis f with respect to the frame of p_0 :

$$Z_{0i} = \begin{bmatrix} Z_w \\ Z_{0i_x} \\ \frac{K}{Z_{0i_y}} \\ Z_{0i_z} \\ \frac{K}{Z_{0i_z}} \end{bmatrix}. \quad (\text{G.18})$$

- d. Compute the quaternion representing the rotation of p_i with respect to the world reference frame (Eq. G.10)

$$Z_{wi} = Z_{w\theta} Z_{0i}. \quad (\text{G.19})$$

- e. Compute the corresponding rotation matrix, R_{wi} (Eq. G.5)).

END

Appendix H

Further Experimental Results

H.1. Stop Point to Stop Point Task Space Motions

H.1.1. Task Space Experiment 1

Point	x [mm]	y [mm]	z [mm]	yaw [rad]	pitch [rad]	roll [rad]
1	510	355	310	0.165	0.0731	-0.0113
2	555	-360	240	0.165	0.0731	-0.0113

Table H.1. Coordinates of the Straight Line Path Start and End Points for Task Space Experiment 1

	Industrial Planner	Quintic Planner	
Demanded Speed Along Path [mm/s (% of MMAS)]	Motion Time [s]	Path Deviation RMS [mm]	Maximum Path Deviation [mm]
102 (10%)	7.9	7.2	0.5
203 (20%)	3.9	3.7	0.5
305 (30%)	2.7	2.6	0.5
406 (40%)	2.0	2.0	0.5
508 (50%)	1.6	1.7	0.5
610 (60%)	1.4	1.5	0.5
711 (70%)	1.2	1.4	0.6
813 (80%)	1.2	1.3	0.6
914 (90%)	1.1	1.3	0.6
1016 (100%)	1.1	1.2	0.6

Table H.2. Results for Task Space Experiment 1

- The tool transform used in this case was 10.5 mm.
- At comparable motion times, the quintic planner provides a 38% to 47% improvement in maximum path deviation, and a 33% to 40% improvement in average path deviation.
- At demanded speeds below 406 mm/s (40% of the Manufacturer Maximum Allowable Speed (MMAS)), implementation of the quintic planner trajectories provides a faster motion and better path deviations as compared to the industrial planner trajectories.
- At demanded speeds above 406 mm/s (40% MMAS), the industrial planner violates the acceleration limit.

H.1.2. Task Space Experiment 2

Point	x [mm]	y [mm]	z [mm]	yaw [rad]	pitch [rad]	roll [rad]
1	430	355	430	$\frac{2\pi}{9}$	$\frac{\pi}{360}$	$\frac{\pi}{90}$
2	430	265	400	$\frac{2\pi}{9}$	$\frac{\pi}{360}$	$\frac{\pi}{90}$

Table H.3. Coordinates of the Straight Line Path Start and End Points for Task Space Experiment 2

	Industrial Planner	Quintic Planner				
Demanded Speed Along Path [mm/s (% of MMAS)]	Motion Time [s]		Path Deviation RMS [mm]		Maximum Path Deviation [mm]	
102 (10%)	1.1	1.0	0.1	0.2	0.3	0.4
203 (20%)	0.6	0.6	0.2	0.2	0.4	0.5
305 (30%)	0.4	0.5	0.3	0.3	0.6	0.5
406 (40%)	0.3	0.5	0.4	0.3	0.7	0.5
508 (50%)	0.3	0.5	0.5	0.3	0.8	0.5
610 (60%)	0.2	0.5	0.6	0.3	1.1	0.5
711 (70%)	0.2	0.5	1.2	0.3	2.4	0.5
813 (80%)	0.2	0.5	1.5	0.3	2.9	0.5
914 (90%)	0.2	0.5	1.5	0.3	3.0	0.5
1016 (100%)	0.2	0.5	1.6	0.3	3.2	0.5

Table H.4. Results for Task Space Experiment 2

- The tool transform used in this case was 0 mm.
- At a demanded speed of 203 mm/s (20% MMAS), the quintic planner trajectories provided 20% poorer maximum path deviation, while providing equivalent average path deviation. No other comparisons are possible since no other times are equivalent.
- It is shown that the quintic planner trajectories limit the motion times, and as a consequence, the maximum and average path deviations during the motion. The large increases in maximum and average path deviations with an increase in demanded speed, obtained with the industrial planner, are a result of violated kinematic constraints.
- At demanded speeds above 203 mm/s (20% MMAS), the industrial planner violates the acceleration limit.

H.1.3. Task Space Experiment 3

Point	x [mm]	y [mm]	z [mm]	yaw [rad]	pitch [rad]	roll [rad]
1	-355	-450	280	$-\frac{5\pi}{18}$	$\frac{11\pi}{36}$	0
2	365	-500	410	$-\frac{5\pi}{18}$	$\frac{11\pi}{36}$	0

Table H.5. Coordinates of the Straight Line Path Start and End Points for Task Space Experiment 3

	Industrial Planner	Quintic Planner				
Demanded Speed Along Path [mm/s (% of MMAS)]	Motion Time [s]		Path Deviation RMS [mm]		Maximum Path Deviation [mm]	
102 (10%)	8.0	7.3	0.4	0.2	0.9	0.9
203 (20%)	4.0	3.8	0.4	0.4	1.1	0.9
305 (30%)	2.7	2.6	0.5	0.4	1.4	0.9
406 (40%)	2.1	2.1	0.6	0.5	1.5	0.9
508 (50%)	1.7	1.7	0.6	0.6	1.6	1.0
610 (60%)	1.4	1.5	0.7	0.6	1.8	1.1
711 (70%)	1.2	1.4	0.8	0.6	1.8	1.3
813 (80%)	1.1	1.3	0.8	0.7	2.0	1.2
914 (90%)	1.1	1.3	0.8	0.7	1.8	1.3
1016 (100%)	1.0	1.2	0.9	0.7	1.9	1.4

Table H.6. Results for Task Space Experiment 3

- The tool transform used in this case was 105 mm.
- At comparable motion times, the quintic planner provides a 22% to 40% improvement in maximum path deviation, and a 13% to 17% improvement in average path deviation.
- At demanded speeds below 406 mm/s (40% MMAS), implementation of the quintic planner trajectories provides a faster motion and better path deviations as compared to the industrial planner trajectories.
- At demanded speeds above 508 mm/s (50% MMAS), the industrial planner violates the acceleration limit.

H.1.4. Task Space Experiment 4

Point	x [mm]	y [mm]	z [mm]	yaw [rad]	pitch [rad]	roll [rad]
1	-355	-450	280	0	$\frac{\pi}{2}$	$\frac{4\pi}{6}$
2	365	-380	410	0	$\frac{\pi}{2}$	$\frac{4\pi}{6}$

Table H.7. Coordinates of the Straight Line Path Start and End Points for Task Space Experiment 4

	Industrial Planner	Quintic Planner				
Demanded Speed Along Path [mm/s (% of MMAS)]	Motion Time [s]		Path Deviation RMS [mm]		Maximum Path Deviation [mm]	
102 (10%)	8.0	7.3	0.4	0.2	1.1	1.1
203 (20%)	4.0	3.8	0.4	0.3	1.2	1.1
305 (30%)	2.7	2.6	0.5	0.4	1.3	1.1
406 (40%)	2.0	2.0	0.5	0.5	1.3	1.0
508 (50%)	1.7	1.7	0.6	0.5	1.5	1.1
610 (60%)	1.4	1.5	0.7	0.6	1.7	1.1
711 (70%)	1.2	1.4	0.7	0.6	1.6	1.1
813 (80%)	1.1	1.3	0.8	0.6	1.7	1.0
914 (90%)	1.0	1.2	0.9	0.7	1.9	1.1
1016 (100%)	1.0	1.2	0.9	0.7	1.7	1.2

Table H.8. Results for Task Space Experiment 4

- The tool transform used in this case was 105 mm.
- At comparable motion times, the quintic planner provides a 23% to 35% improvement in maximum path deviation, and a 14% to 17% improvement in average path deviation.
- At demanded speeds below 406 mm/s (40% MMAS), implementation of the quintic planner trajectories provides a faster motion and better path deviations as compared to the industrial planner trajectories.
- At demanded speeds above 406 mm/s (40% MMAS), the industrial planner violates the acceleration limit.

H.1.5. Task Space Experiment 5

Point	x [mm]	y [mm]	z [mm]	yaw [rad]	pitch [rad]	roll [rad]
1	520	320	660	-0.0716	0.0113	0
2	410	455	210	-0.0716	0.0113	0

Table H.9. Coordinates of the Straight Line Path Start and End Points for Task Space Experiment 5

	Industrial Planner	Quintic Planner				
Demanded Speed Along Path [mm/s (% of MMAS)]	Motion Time [s]		Path Deviation RMS [mm]		Maximum Path Deviation [mm]	
102 (10%)	5.3	4.9	0.2	0.2	0.6	0.4
203 (20%)	2.7	2.5	0.3	0.2	0.7	0.5
305 (30%)	1.8	1.8	0.3	0.3	0.7	0.5
406 (40%)	1.4	1.4	0.3	0.3	0.7	0.6
508 (50%)	1.1	1.2	0.4	0.3	0.8	0.7
610 (60%)	0.9	1.1	0.5	0.4	0.9	0.6
711 (70%)	0.8	1.0	0.5	0.4	1.0	0.7
813 (80%)	0.8	1.0	0.6	0.4	1.1	0.6
914 (90%)	0.7	1.0	0.6	0.4	1.1	0.7
1016 (100%)	0.7	1.0	0.7	0.4	1.3	0.7

Table H.10. Results for Task Space Experiment 5

- The tool transform used in this case was 105 mm.
- At comparable motion times, the quintic planner provides a 14% to 29% improvement in maximum path deviation, and equivalent average path deviation.
- At demanded speeds below 305 mm/s (30% MMAS), implementation of the quintic planner trajectories provides a faster motion and better path deviations as compared to the industrial planner trajectories.
- At demanded speeds above 406mm/s (40% MMAS), the industrial planner violates the acceleration limit.

H.1.6. Task Space Experiment 6

Point	x [mm]	y [mm]	z [mm]	yaw [rad]	pitch [rad]	roll [rad]
1	410	455	210	-0.0716	0.0113	0
2	520	320	660	-0.0716	0.0113	0

Table H.11. Coordinates of the Straight Line Path Start and End Points for Task Space Experiment 6

Demand Speed Along Path [mm/s (% of MMAS)]	Motion Time [s]		Path Deviation RMS [mm]		Maximum Path Deviation [mm]	
102 (10%)	5.3	4.9	0.2	0.2	0.5	0.8
203 (20%)	2.7	2.5	0.3	0.3	0.6	0.8
305 (30%)	1.8	1.8	0.3	0.3	0.6	0.8
406 (40%)	1.4	1.4	0.4	0.4	0.7	0.8
508 (50%)	1.1	1.2	0.4	0.4	0.9	0.8
610 (60%)	0.9	1.1	0.5	0.5	1.0	0.8
711 (70%)	0.8	1.0	0.5	0.5	1.1	0.8
813 (80%)	0.8	1.0	0.6	0.5	1.4	0.8
914 (90%)	0.7	1.0	0.9	0.5	2.6	0.8
1016 (100%)	0.7	1.0	1.5	0.5	4.7	0.8

Table H.12. Results for Task Space Experiment 6

- The tool transform used in this case was 105 mm.
- At demanded speeds of 305 mm/s (30% MMAS) and 406 mm/s (40% MMAS), the quintic planner provides a 13% to 25% poorer maximum path deviation, and equivalent average path deviation.
- For a motion time of 1.1 s, the quintic planner provides an improvement of 11% in maximum path deviation and a 20% poorer average path deviation.
- At demanded speeds below 305 mm/s (30% MMAS), the implementation of the quintic planner trajectories provides a faster motion, equivalent average path deviation and poorer maximum path deviation.
- As the demanded speed increases, there is a large increase in maximum path deviation for the industrial planner trajectories.
- At demanded speeds above 305 mm/s (30% MMAS), the industrial planner violates the acceleration limit.

H.1.6.1 Explanation of Results for Task Space Experiment 6

The maximum path deviation remains constant with increasing demanded speed for the quintic planner. It occurs during or shortly after the initial change in speed (during the first acceleration phase). For the industrial planner trajectories, at high demanded speeds, the acceleration limit is violated, resulting in large maximum path deviations, which occur during the acceleration phase. However, at demanded speeds below 305 mm/s (30% of MMAS), the industrial planner trajectories do not cause such large path deviations since the maximum acceleration is lower than the limit. This would indicate that the acceleration limit is too high for this manipulator path.

This path has end-points identical to the path end-points of Section H.1.5; they are simply interchanged. The maximum path deviation for the path of Section H.1.5 occurred near the end of the trajectory, while the maximum path deviation for the path of Section H.1.6 occurred near the beginning of the trajectory. This indicates that the given combination of trajectory manipulator configuration, speed and acceleration at this point may violate the joint limits.

H.2. Multiple Way-Point Task Space Motions

H.2.1. Horizontal Rectangular Motion

Point	x [mm]	y [mm]	z [mm]	yaw [rad]	pitch [rad]	roll [rad]
1	315	-300	390	$\frac{5\pi}{9}$	$\frac{\pi}{2}$	$\frac{11\pi}{18}$
2	470	-300	390	$\frac{5\pi}{9}$	$\frac{\pi}{2}$	$\frac{11\pi}{18}$
3	470	300	390	$\frac{5\pi}{9}$	$\frac{\pi}{2}$	$\frac{11\pi}{18}$
4	315	300	390	$\frac{5\pi}{9}$	$\frac{\pi}{2}$	$\frac{11\pi}{18}$
5	315	-300	390	$\frac{5\pi}{9}$	$\frac{\pi}{2}$	$\frac{11\pi}{18}$

Table H.13. Coordinates of Way-Points for Task Space Horizontal Rectangular Path

	Industrial Planner	Quintic Planner
Demanded Speed Along Path [mm/s (% of MMAS)]		Motion Time [s]
102 (10%)	15.12	14.96
203 (20%)	8.00	7.57
305 (30%)	5.81	5.13
406 (40%)	4.75	4.13
508 (50%)	4.17	3.66
610 (60%)	3.90	3.38
711 (70%)	3.76	3.20
813 (80%)	3.74	3.09
914 (90%)	3.80	3.02
1016 (100%)	3.90	2.99

Table H.14. Results for Horizontal Rectangular Path

- The tool transform used for this motion was 105 mm.
- At all demanded speeds, the quintic planner provides a faster motion than the industrial planner does.

H.2.2. Vertical Rectangular Motion

Point	x [mm]	y [mm]	z [mm]	yaw [$^{\circ}$]	pitch [$^{\circ}$]	roll [$^{\circ}$]
1	510	115	240	0	0	0
2	510	465	240	0	0	0
3	510	465	575	0	0	0
4	510	115	575	0	0	0
5	510	115	240	0	0	0

Table H.15. Coordinates of Way-Points for Task Space Vertical Rectangular Path

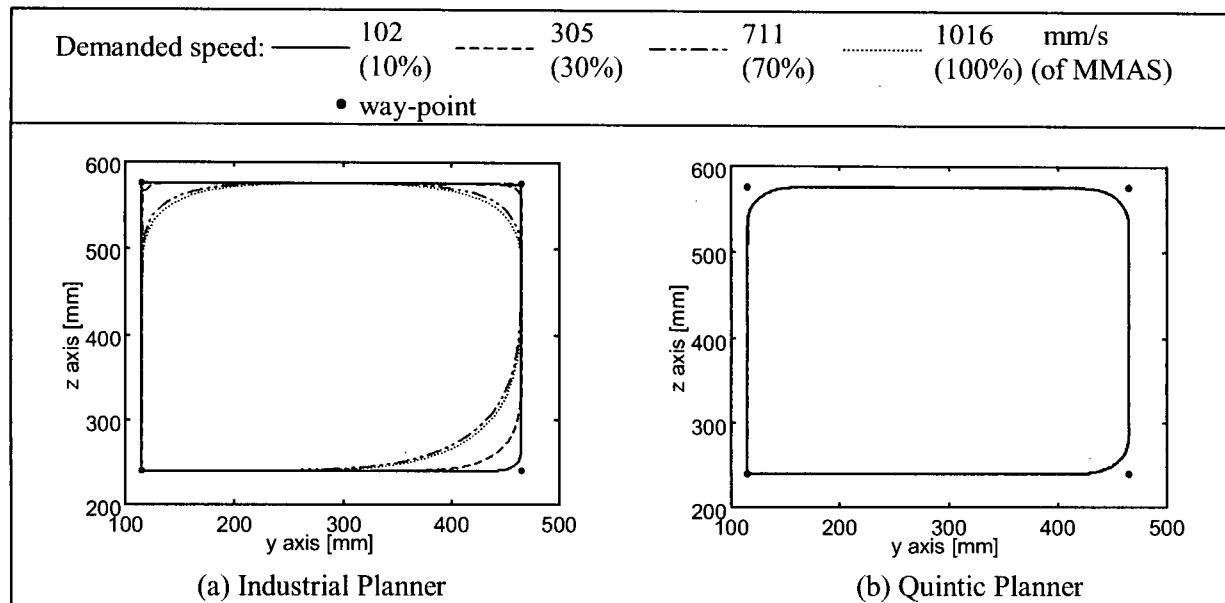


Figure H.1. Comparison of Paths at Different Speeds for Vertical Rectangular Path

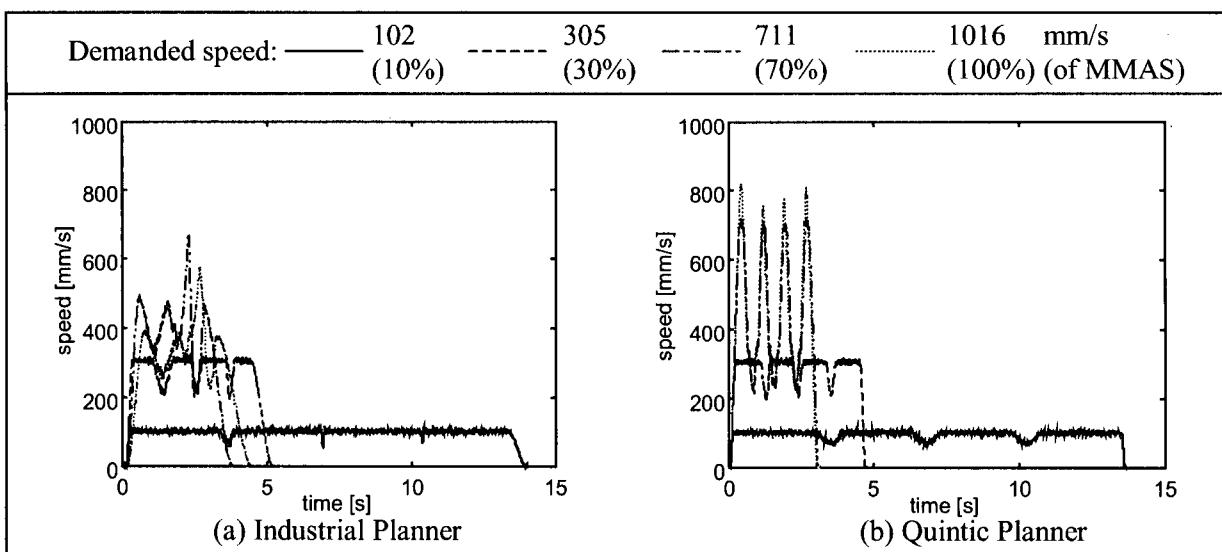


Figure H.2. Comparison of Speed Profiles for Task Space Vertical Rectangular Path

	Industrial Planner	Quintic Planner
Demanded Speed Along Path [mm/s (% of MMAS)]	Motion Time [s]	
102 (10%)	13.74	13.58
203 (20%)	7.08	6.88
305 (30%)	4.96	4.67
406 (40%)	4.00	3.78
508 (50%)	3.48	3.38
610 (60%)	3.42	3.16
711 (70%)	3.55	3.05
813 (80%)	3.73	3.01
914 (90%)	3.95	3.01
1016 (100%)	4.17	3.01

Table H.16. Results for Task Space Vertical Rectangular Path

- The tool transform used for this motion was 105 mm.
- At all demanded speeds, the quintic planner provides a faster motion than the industrial planner does.
- At demanded speeds above 711 mm/s (70% MMAS), the motion is limited by distance for the quintic planner.
- At demanded speeds above 610 mm/s (60% MMAS), the industrial planner motion time increases with increasing demanded speed, since the maximum path acceleration is reduced. This occurs when one of the path segments between two way-points becomes a pure blending motion (i.e. it is formed of two blend halves).

H.3. Stop Point to Stop Point Joint Space Motions

H.3.1. Joint Space Experiment 1

Point	Joint 1 [rad]	Joint 2 [rad]	Joint 3 [rad]	Joint 4 [rad]	Joint 5 [rad]	Joint 6 [rad]
1	$\frac{7\pi}{18}$	$-\frac{\pi}{9}$	$\frac{13\pi}{18}$	$\frac{\pi}{6}$	$\frac{2\pi}{9}$	$-\frac{5\pi}{18}$
2	$-\frac{4\pi}{9}$	$-\frac{5\pi}{9}$	$\frac{11\pi}{18}$	$-\frac{\pi}{3}$	$-\frac{\pi}{6}$	$-\frac{7\pi}{18}$

Table H.17. Coordinates of Joint Space Straight Line Start and End Points for Joint Space Experiment 1

	Industrial Planner	Quintic Planner	Deviation of Joint [10^{-3} rad]											
Demanded Speed Along Path [mm/s (% of MMAS)]	Motion Time [s]		1	2	3	4	5	6						
102 (10%)	8.5	8.4	1	0	0	1	0	1	0	0	0	1	1	1
203 (20%)	4.4	4.3	0	0	0	1	0	1	1	1	1	2	1	1
305 (30%)	3.1	2.9	0	0	1	1	0	1	2	3	2	2	1	1
406 (40%)	2.4	2.2	0	0	1	1	1	1	2	3	2	3	1	1
508 (50%)	2.0	1.8	0	0	1	1	1	1	3	3	3	3	1	1
610 (60%)	1.8	1.6	0	0	1	2	1	1	3	4	3	3	1	1
711 (70%)	1.6	1.4	0	0	1	2	1	1	3	4	3	4	1	1
813 (80%)	1.5	1.3	0	0	1	2	1	1	3	4	3	4	1	1
914 (90%)	1.4	1.2	0	0	1	1	1	1	4	4	3	4	1	1
1016 (100%)	1.4	1.1	0	0	1	2	1	1	4	4	3	4	1	1

Table H.18. Results for Joint Space Experiment 1

- The quintic planner provides a faster motion at all demanded speeds, and equivalent accuracy at the end points.

H.3.2. Joint Space Experiment 2

Point	Joint 1 [rad]	Joint 2 [rad]	Joint 3 [rad]	Joint 4 [rad]	Joint 5 [rad]	Joint 6 [rad]
1	$-\frac{\pi}{9}$	$-\frac{\pi}{3}$	$\frac{17\pi}{18}$	0	$-\frac{\pi}{9}$	0
2	$-\frac{5\pi}{36}$	$-\frac{5\pi}{18}$	$\frac{33\pi}{36}$	$-\frac{\pi}{18}$	$-\frac{\pi}{12}$	$\frac{\pi}{18}$

Table H.19. Coordinates of Joint Space Straight Line Start and End Points for Joint Space Experiment 2

		Industrial Planner	Quintic Planner										
Demanded Speed Along Path [mm/s (% of MMAS)]	Motion Time [s]	Deviation of Joint [10^{-3} rad]											
		1	2	3	4	5	6	1	2	3	4	5	6
102 (10%)	0.8	0.7	0	0	0	1	0	0	0	1	1	1	0
203 (20%)	0.5	0.4	0	0	0	0	0	0	0	1	2	2	0
305 (30%)	0.6	0.3	0	0	0	0	0	0	1	1	1	3	0
406 (40%)	0.6	0.3	0	0	0	0	0	0	1	1	1	3	0
508 (50%)	0.7	0.3	0	0	0	0	0	0	0	1	1	3	0
610 (60%)	0.7	0.3	0	0	0	0	0	0	0	1	1	3	0
711 (70%)	0.8	0.3	0	0	0	0	0	0	0	1	1	3	0
813 (80%)	0.8	0.3	0	0	0	0	0	0	0	1	1	3	0
914 (90%)	0.9	0.3	0	0	0	0	0	0	1	1	1	3	0
1016 (100%)	0.9	0.3	0	0	0	0	0	0	1	1	1	3	0

Table H.20. Results for Joint Space Experiment 2

- The quintic planner provides a faster motion at all demanded speeds, and equivalent accuracy at the end points.
- Use of the industrial planner results in an increase in motion time as the demanded speed is increased above 610 mm/s (60% MMAS). This occurs due to the distance limitation. The industrial planner algorithm computes the time necessary to ramp up to and then down from the demanded speed. If the distance covered during these two ramps is larger than the required distance, the acceleration and deceleration limits are reduced proportionally such that the required distance is met, rather than lowering the peak speed reached along the path, as is done with the quintic planner. Since a longer time is required to ramp up to and down from 100% of MMAS than 90% of MMAS, for example, and both cases are limited by the distance, the motion at 100% of MMAS will take longer.

H.3.3. Joint Space Experiment 3

Point	Joint 1 [rad]	Joint 2 [rad]	Joint 3 [rad]	Joint 4 [rad]	Joint 5 [rad]	Joint 6 [rad]
1	$\frac{5\pi}{18}$	$-\frac{\pi}{9}$	$\frac{3\pi}{4}$	0	$-\frac{5\pi}{36}$	0
2	$-\frac{\pi}{4}$	$-\frac{2\pi}{9}$	$\frac{7\pi}{9}$	$-\frac{7\pi}{36}$	$-\frac{\pi}{36}$	$\frac{7\pi}{36}$

Table H.21. Coordinates of Joint Space Straight Line Start and End Points for Joint Space Experiment 3

	Industrial Planner	Quintic Planner	Deviation of Joint [10^{-3} rad]											
Demanded Speed Along Path [mm/s (% of MMAS)]	Motion Time [s]		1	2	3	4	5	6						
102 (10%)	5.4	5.4	1	1	0	0	0	0	0	1	0	1	0	1
203 (20%)	2.9	2.7	0	1	0	1	0	0	0	1	1	1	0	1
305 (30%)	2.0	1.9	0	0	0	1	0	0	1	1	1	2	1	1
406 (40%)	1.6	1.5	0	0	0	0	0	0	1	1	1	1	1	1
508 (50%)	1.4	1.2	0	0	0	0	0	0	1	1	1	1	1	1
610 (60%)	1.3	1.1	0	0	0	0	0	0	1	2	1	1	1	1
711 (70%)	1.2	1.0	0	1	0	0	0	0	1	2	1	1	1	1
813 (80%)	1.1	0.9	0	0	0	0	0	0	1	2	1	1	1	2
914 (90%)	1.1	0.9	0	0	0	1	0	0	1	2	1	1	1	1
1016 (100%)	1.1	0.8	0	0	0	0	0	0	1	2	1	1	1	1

Table H.22. Results for Joint Space Experiment 3

- The quintic planner provides a faster motion at all demanded speeds, and equivalent accuracy at the end points.

H.3.4. Joint Space Experiment 4

Point	Joint 1 [rad]	Joint 2 [rad]	Joint 3 [rad]	Joint 4 [rad]	Joint 5 [rad]	Joint 6 [rad]
1	$\frac{11\pi}{36}$	$-\frac{11\pi}{36}$	$\frac{10\pi}{9}$	$\frac{2\pi}{9}$	$-\frac{11\pi}{36}$	$-\frac{\pi}{9}$
2	$-\frac{5\pi}{18}$	$-\frac{15\pi}{36}$	$\frac{37\pi}{36}$	0	$-\frac{\pi}{9}$	0

Table H.23. Coordinates of Joint Space Straight Line Start and End Points for Joint Space Experiment 4

	Industrial Planner	Quintic Planner	Deviation of Joint [10^{-3} rad]											
Demanded Speed Along Path [mm/s (% of MMAS)]	Motion Time [s]		1	2	3	4	5	6						
102 (10%)	6.0	5.9	1	0	0	0	0	1	0	0	1	0	0	1
203 (20%)	3.1	3.0	0	0	0	0	0	1	1	1	1	1	1	1
305 (30%)	2.2	2.1	0	0	0	0	0	1	2	1	2	1	1	1
406 (40%)	1.8	1.6	0	0	0	0	0	1	2	1	2	1	1	1
508 (50%)	1.5	1.3	0	0	0	0	0	1	2	2	2	1	1	1
610 (60%)	1.3	1.2	0	0	0	0	0	1	2	2	2	1	1	1
711 (70%)	1.2	1.1	0	0	0	0	0	2	2	2	2	1	1	1
813 (80%)	1.2	1.0	0	0	0	0	0	2	2	2	2	1	1	1
914 (90%)	1.1	0.9	0	0	0	0	0	1	2	1	2	0	1	1
1016 (100%)	1.1	0.9	0	0	0	0	0	2	2	1	2	1	1	1

Table H.24. Results for Joint Space Experiment 4

- The quintic planner provides a faster motion at all demanded speeds, and equivalent accuracy at the end points.

H.4. Multiple Way-Point Joint Space Motion

Point	Joint 1 [rad]	Joint 2 [rad]	Joint 3 [rad]	Joint 4 [rad]	Joint 5 [rad]	Joint 6 [rad]
1	$\frac{\pi}{6}$	$-\frac{2\pi}{3}$	$\frac{10\pi}{9}$	0	0	0
2	$\frac{\pi}{3}$	$-\frac{7\pi}{18}$	$\frac{17\pi}{18}$	$\frac{\pi}{9}$	$-\frac{\pi}{18}$	$-\frac{\pi}{9}$
3	$-\frac{\pi}{3}$	$-\frac{7\pi}{18}$	$\frac{17\pi}{18}$	$-\frac{\pi}{9}$	$-\frac{\pi}{18}$	$\frac{\pi}{9}$
4	$-\frac{\pi}{3}$	$-\frac{5\pi}{18}$	π	$-\frac{\pi}{6}$	$-\frac{2\pi}{9}$	$\frac{11\pi}{18}$
5	$\frac{\pi}{6}$	$-\frac{2\pi}{3}$	$\frac{10\pi}{9}$	0	0	0

Table H.25. Coordinates of Way-Points for Multiple Way-Point Joint Space Path

Industrial Planner Quintic Planner

Demanded Speed Along Path [mm/s (% of MMAS)]	Motion Time [s]
102 (10%)	21.32
203 (20%)	10.79
305 (30%)	7.31
406 (40%)	5.58
508 (50%)	4.55
610 (60%)	3.89
711 (70%)	3.47
813 (80%)	3.21
914 (90%)	3.02
1016 (100%)	2.88
	21.21
	10.67
	7.18
	5.43
	4.40
	3.72
	3.28
	2.98
	2.80
	2.69

Table H.26. Motion Time for the Multiple Way-Point Path at Varying Demanded Speeds

- The quintic planner consistently provides a faster motion at a given demanded speed.