## NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

## AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

# A METHOD FOR GENERATING A CONTROLLABLE AND QUASI TIME OPTIMAL ROBOT TRAJECTORY

By

Sofian Amara

THESIS SUBMITTED
TO THE SCHOOL OF GRADUATE STUDIES
AS PARTIAL FULLFILMENT FOR THE REQUIREMENTS OF THE
DEGREE OF MASTERS OF APPLIED SCIENCE

at the
DEPARTMENT OF MECHANICAL ENGINEERING
UNIVERSITY OF OTTAWA

OTTAWA-CARLETON INSTITUTE FOR
MECHANICAL AND AERONAUTICAL ENGINNEERING

ISBN 0-315-60082-9

UNIVERSITÉ D'OTTAWA
UNIVERSITY OF OTTAWA

# Abstract

Robot arms are used within their work space to execute a variety of physical tasks like pick and place, or weld along a contour. From a robot control perspective, these tasks can be simply viewed as tasks of trajectory planning of the robot end effector. The path of the robot end effector is generally prescribed as a number of spatial points through which the end effector has to pass. A large number of approaches for the planning of the trajectory of a robot has been developed. These approaches do not generally allow adequate control over the acceleration profile of the trajectory. Furthermore, among those approaches only a few generate trajectories that are time optimal, and only for very simplistic robots.

This work deals with yet another method for trajectory planning. The method allows full control over the acceleration profile so as to minimize the jerk at the beginning and end of the motion. The method also allows the utilization of the axes drive motors to their full capability in order to obtain a quasi-optimum trajectory that passes through all the points defining the end effector path.

Simulation of the method for a three degrees of freedom revolute arm manipulator has been carried out. Results and discussion are presented in this study.

# Acknowledgements

The author wishes to express his gratitude to Dr. A. Fahim for his continuous encouragement, and patient guidance throughout the research.
Special thanks to Nabila Ouanane, my wife, and to the Tunisian students at the University of Ottawa for their moral support and encouragement during the study.
The financial support received from the Tunisian government through the University Mission of Tunisia in Washington is gratefully acknowledged.

# Contents

# List of Figures

# Nomenclature

| | | |
|---|---|---|
| $a'$ | = | Visiting point |
| $a_{ij}$ | = | Polynomial coefficient |
| $a_i$ | = | Length of link i |
| ${}^i a_i^*$ | = | Acceleration of center of mass of link i |
| ${}^i A_i$ | = | Rotation matrix |
| $(a, b, c, d, e)$ | = | Proposed function parameters |
| b' | = | Visiting point |
| $c_i$ | = | $\cos(\theta_i)$ |
| $C$ | = | Curvature equation |
| $E$ | = | End effector transformation |
| $g$ | = | Acceleration due to gravity |
| $H$ | = | manipulator inertia matrix |
| $H_{ij}$ | = | element ij of the H matrix |
| h | = | vector of the coriolis and centrifugal terms |
| $h_i$ | = | element i the h vector |
| $I_{ix}$ | = | inertia of link i w.r.to its center of mass about the x axis |
| $I_{iy}$ | = | inertia of link i w.r.to its center of mass about the y axis |
| $I_{iz}$ | = | inertia of link i w.r.to its center of mass about the z axis |
| $i$ | = | joint reference |
| $\vec{i}$ | = | unit vector along x coordinate |
| $j$ | = | segment number reference |
| $\vec{j}$ | = | unit vector along y coordinate |
| $k$ | = | subtrajectory number reference |
| $\vec{k}$ | = | unit vector along z coordinate |
| $K_i$ | = | Kinetic energy of link i |
| L | = | Lagrangian |
| $q_{ijk}$ | = | generalized coordinate |
| $\dot{q}_{ijk}$ | = | generalized speed |
| $\ddot{q}_{ijk}$ | = | generalized acceleration |
| $\dddot{q}_{ijk}$ | = | generalized jerk |
| $Q_i$ | = | Generalized force |
| $\vec{r}(s)$ | = | vector as a function of parameter s |
| $R$ | = | Position vector |

| | | |
|---|---|---|
| $s$ | = | distance parameter |
| $s_i$ | = | $\sin(\theta_i)$ |
| $T$ | = | normalized time |
| $t$ | = | real time |
| $t_{ijk}$ | = | real time at the end of the $k^{th}$ subtrajectory for joint i |
| $t_j$ | = | real time for segment j |
| $\vec{T}$ | = | tangential vector |
| $U$ | = | Potential energy |
| $V_i^*$ | = | Velovity of center of mass |
| ${}^iV_i$ | = | Linear velocity of frame i |
| $x_i$ | = | coordinate along x axis |
| $x_k(s)$ | = | parametric function of s |
| $y_i$ | = | coordinate along y axis |
| $y_k(s)$ | = | parametric function of s |
| $z_i$ | = | coordinate along z axis |
| $z_k(s)$ | = | parametric function of s |
| $\alpha_{max}$ | = | maximum angular acceleration |
| $\kappa(s)$ | = | curvature of curve |
| $\mu$ | = | mean value (parameter) |
| ${}^i\omega_i$ | = | angular velocity of frame i |
| $\rho(s)$ | = | radius of curvature |
| $\sigma$ | = | standard deviation (parameter) |
| $\tau_i$ | = | driving torque |
| $\theta_s$ | = | starting angular displacement |
| $\dot{\theta}_s$ | = | starting angular velocity |
| $\ddot{\theta}_s$ | = | starting angular acceleration |
| $\theta_e$ | = | final angular displacement |
| $\dot{\theta}_e$ | = | final angular velocity |
| $\ddot{\theta}_e$ | = | final angular acceleration |

# Chapter 1

# INTRODUCTION

The performance and reliability of industrial robots has improved steadily since their broad introduction on the manufacturing floor. Robot applications today are as diversified and sophisticated as production techniques are themselves. Many modern robot applications require the robot to execute a specific task such as to transfer parts from one station to another, assembly of parts, machining or weld along contours, etc.

To accomplish its task, the robot end effector has to follow a certain trajectory. Even though there is a large number of approaches for trajectory planning of the robot arm, high positioning accuracy, minimum time trajectory or optimal motion control, are areas that still require developing.

Trajectory planning may be defined as the process of converting a description of a desired motion to a trajectory defining the time sequence of intermediate configurations of the arm between an initial position or orientation and final position or orientation. As the trajectory is executed, the tip of the end effector traces a curve in space and changes its orientation. The space curve traced by the end effector is called the path of the trajectory.

Trajectory planning consists mainly of defining the manipulator motion between a source point and destination point with respect to a reference frame. Generally, two common approaches are used to describe and plan trajectories. In the first approach, the programmer implicitly describes the path and trajectory followed by the arm, by specifying a set of constraints on its position, velocity and acceleration at a number of points along the desired path. The trajectory planner chooses one class of parameterized trajectories that satisfies the constraints. In the second approach a path is described explicitly by an analytic function, usually in Cartesian coordinates, and the objective is to determine a trajectory whose corresponding path closely approximates the prescribed one. In both approaches, the description of the task is given in Cartesian space, since it's easy to visualize paths and the corresponding changes in Cartesian space.

The spatial duality of trajectory planning requires that some characteristics are either related to joint space or Cartesian space. These characteristics such as the visibility of Cartesian path to prevent the robot arm from colliding with objects in its environment, and some physical constraints such as the joints limits and the working volume of the robot arm yield the planning of the trajectory more difficult.

The output of trajectory planning is a sequence of arm configurations that forms the input to the arm's feedback control system. The control approach used for industrial robots is to consider each joint as a simple servomechanism. For these type of robots, the system controller is designed based on the critical configuration of the robot arm, this results in the joints actuators being driven, the vast majority

of the time, below their maximum available capacity. To overcome this problem the trajectory has to be planed taking into account the effect of manipulator dynamics. This approach will be dealt with in the second part of this work.

This work deals with yet another method for trajectory planning. The trajectory control in joint space is treated here by means of imposed acceleration profiles for each transition segment. The acceleration is selected by taking an approximation of the Gauss function, after two integrations the equation of motion governing the trajectory is found. The method allows full control over the acceleration profile so as to minimize the jerk at the beginning and end of the motion. Using the maximum allowable torques or forces of the joints actuators a quasi-optimum trajectory that passes through all the points defining the end effector path is obtained.

To start the analysis a brief introduction to trajectory planning and the techniques that has to be considered is presented and reviewed in Chapter 2.

In Chapter 3, two method for trajectory specification and generation for robot arms are presented. The first method is a parameterized class of function, here a second order polynomial combined with a cosine function, to generate a specific trajectory. The second is an approach used to compute a trajectory that causes the end effector of the manipulator to follow a Cartesian straight line path between two given end points.

Chapter 4 deals with the design of the motion trajectory based on the theory developed. The general algorithm for the method is presented and discussed.

In Chapter 5, a novel approach for quasi-optimum trajectory motion is presented. The effect of manipulator dynamics on the design of a specific path is discussed

and an algorithm for the path generation is presented.

Chapter 6 deals with the implementation of different proposed algorithms. A robot motion simulation for a typical task is presented.

Finally, conclusions are drawn based on the analysis as well as recommendations for further research are presented.

# Chapter 2

LITERATURE REVIEW

## 2.1 Introduction

Training a robot to perform a specific task is one of the most important class of problems in the area of robotics. The study to generate a specific task requires the analysis of several aspects such as the kinematics of the robot, its dynamics, trajectory planning and the motion control and coordination.

In this chapter the discussion will be restricted only to trajectory planning and control. The criteria to analyse in order to achieve an ideal trajectory taking into account the physical constraints of the robot manipulator will be defined first. Several authors have discussed these criteria, among these are Shahinpoor[52], Lee et Al [28], and Paul[41].

The spatial duality of trajectory planning requires that some characteristics are either related to joint space or Cartesian space. Some of these criteria that are related to the Cartesian path are: a) The feasibility of the Cartesian path. Since most of the industrial robots work in a crowded work space, it is important to plan a trajectory that results in a feasible Cartesian path, so that the robot arm does not collide with object in its environment. This requires that the trajectory

planner has to specify and predict the motion to be executed by the robot manipulator, and not just be concerned with the source and the destination of the motion. b) The second criteria is the accuracy of the Cartesian path. According to the practical requirements of some applications like arc welding along contour, the execution of a high precision trajectory is vital to the success of the task. c) The continuity of the Cartesian path, which represent a physical position constraint, to allow a feasible solution for the trajectory. d) Finally it should be possible to determine efficiently whether a proposed Cartesian trajectory requires the arm to move to a point outside its workspace or move with a velocity or acceleration that is physically impossible.

All of the above criteria are described in Cartesian space, however the motion of the manipulator arm stems from the motion of different joint actuators defined in joint space. To plan a trajectory one has to take into consideration some criteria stated at joint level, such as: the maximum available and allowable velocity and acceleration that are necessary to ensure that the trajectory is feasible and executable. Furthermore one has to take into account the continuity of the position, velocity and acceleration. Some authors have suggested that the rate of the change of the acceleration, called "jerk", should also be smooth and moderate to prevent the vibrational excitation of the robot manipulator structure.

Generally, it is necessary to design a trajectory that would be able to cater for a variety of tasks. The algorithm must be simple to implement and easy to use to facilitate the task of the operator. An important aspect of such algorithm is the time control of the robot arm. Usually the robot arm works within a work-cell. In

order to coordinate the motion of different units of the cell, the algorithm should provide the possibility of defining the time between points along the path. In the case where the time is not prescribed, the method should allow a time minimization so as to increase the productivity.

A review of literature dealing with the generation of trajectories using kinematic and dynamic approaches in both Cartesian and joint space will be carried out. Literature dealing with techniques used to find the optimal motion control of the robot manipulator will also be reviewed.

## 2.2 Trajectory generation in joint space

In robotics, trajectory planning defines the process of converting a description of the desired motion to a time sequence of intermediate configurations of the robot arm between the source point and destination point. Usually these points are defined in Cartesian space, and are transformed in joint space using the inverse kinematic approach.

The existing algorithms for trajectory generation can be classified into three classes. In the first class, a polynomial of high degree is used to describe the entire trajectory. In the second class, the trajectory is divided into smaller subtrajectories. These are further divided into segments, and a unique function is used to describe each segment. Again, in the third class the trajectory is divided into subtrajectories. A composite (or multi-function) is used to describe each subtrajectory.

### 2.2.1 High degree polynomial trajectories

It is always possible to satisfy an arbitrary number of physical constraints (including collision avoidance problem) by a polynomial trajectory, Brady et Al[4]. This approach is not practical because of a large number of drawbacks. Some of these drawbacks follow:

a) It is difficult to check if a polynomial trajectory of degree n violates physical constrains during the robot arm movement. The difficulty of checking increase with n. Such constraints include checking of collisions with objects in the workspace.

b) A polynomial of degree n has n distinct roots, and as such may cross an arbitrary straight line up to n times. This leads to a wavy motion from one visiting point to another.

c) The numerical accuracy to which a polynomial can be computed decrease as the degree of the polynomial increases.

### 2.2.2 Decomposed trajectories

To correct for the drawbacks of using a high degree polynomials, the trajectory may be decomposed into subtrajectories bounded by knot points. Analytical function, usually polynomials of low degree, are used for the subtrajectories. These function are smoothly joint together at knot points to cover the total trajectory. There are several types of polynomial functions that can be used to describe a given subtrajectory. A linear function is the simplest of such polynomials. Such a simple function, however results in only the position constraints being satisfied. A cubic curve may be used, its four coefficients can be used to satisfy the four con-

straints of position and velocity at the knot points. In order to satisfy the position, velocity and acceleration constraints at both knot points five degree polynomial has to be used.

Generally a $3^{rd}$, $4^{th}$ and $5^{th}$ degree polynomials are used to approximate the three segments of motion trajectory, namely; lift-off, cruise and set-down. According to Lin et Al[30], this approach generates a smooth subtrajectory with minimum discrepancy.

The joint approximated Cartesian path generated by this approach deviates from the desired path. Furthermore optimization and computation cannot be achieved in real time. Lin and Chang[31] proposed an algorithm based on curve fitting, which allows a real time computation. This method does not take into consideration the physical limits of the robot manipulator and the resulting trajectory is not optimized.

To optimize the trajectory, Lin et Al[30] proposed an algorithm to optimize the time between each two points taking into consideration the velocity, acceleration and jerk constraints at joint level. This method require long computation time.

### 2.2.3 Multi-function for each subtrajectory

In comparison with the above method, the multi-function approach is a more local solution. Since the governing function for the subtrajectory are generated based on the adjacent subtrajectory points. The most known of these is the linear interpolation method presented by Paul[41], and modified by Taylor[58].

## Linear joint interpolation

The general motion scheme consist of moving the manipulator at constant velocity in either joint space coordinates or Cartesian coordinates, from one point to the other. As each position is approached, the motion to the next position is determined in order to evaluate the change in velocity. This change in velocity is established by a $5^{th}$ order polynomial function of time. Due to the symmetry imposed on the transition a $4^{th}$ order polynomial is used. Based on the maximum desired acceleration, a transition time is determined and a transition to the next path segment is initiated at one half the transition time before the end of the subtrajectory.

Linear joint interpolation does not force the end effector of the manipulator to pass by the visiting point. The end effector would pass by a knot point only if it is allowed to stop a the these point.

The algorithm proposed by Taylor[58] is a compromise between the efficiency of implementation and the accuracy of following a linear Cartesian path. The planning is done by an efficient recursive algorithm which generates only enough intermediate points to guarantee that the deviation from a straight line path stays within prespecified error bound. Taylor[58] assume that the maximum deviation occur at the middle of the segment, and hence the generated points are forced at the middle of segments.

In the case of linear interpolation, the robot end effector pass by the knot points only when it is allowed to stop there. The generated points reduce the discrepancy of the segment, and results in a very slow motion near the knot points. As the

number of these intermediate points increase the deviation from the desired path is reduced.

## 2.3 Trajectory generation in Cartesian space

Trajectory generation in Cartesian space are not frequently used due to the difficulty of satisfying the joint constraints imposed by the physical structure of the manipulator. Several approaches are used to generate the Cartesian trajectory, and to transform the joint constraints into Cartesian constraints.

To generate a desired path in Cartesian space, current methods consist of finding the inverse kinematic problem so that the information fed to the controller are at the joint level. The role of the controller is to coordinate the joint motion to achieve the desired Cartesian path.

Based on the experimental limits of the cartesian velocity and acceleration, Paul[42] propose a linear interpolation in Cartesian space. Between successive visiting points the robot motion is coordinated using a constant linear velocity which results in a linear Cartesian path. To satisfy the continuity at the intermediate points the linear segments are joined together by a $5^{th}$ order polynomial segment. Luh and Lin[37] proposed to transform the velocity and acceleration limits from joint space to Cartesian space. The algorithm proposed consist of using a linear Cartesian interpolation between successive visiting points and optimize the trajectory taking into account the constraints mapped from joint space to the Cartesian space. It is difficult to transform the joint constraints into Cartesian constraints since the relationship between the two spaces are not linear and also the dynamic equations are coupled and non linear.

## 2.4 Optimal motion control of a robot manipulator

Up to now trajectories that require the end effector to pass by a number of points without stopping can only be preplanned off-line, where the position,velocity and acceleration of the joints are specified at each instant of time. The robot controller coordinates the motion of the joints based on these information. The information fed to the controller from the trajectory planner does not take into account the dynamics of the manipulator. The dynamics of the manipulator is more related to the control law of the system, and the kinematics of motion are essential for the computation of the dynamics.

Several authors such as Yamamoto and Mohri[64], Kahn and Roth[19] conclude that an explicit computation of the minimum time control is impossible. Kahn and Roth[19] proposed a linear approximation (linearization of manipulator dynamics by Taylor expansion) for optimal control. This leads to a quasi-optimal solution, and only suitable for simple, one or two degree, manipulators. Yamamoto and Mohri[64] proposed an algorithm for planning of quasi-minimum time trajectories, where the optimal input torque/force is of a bang-bang type. The approximation to the dynamics is carried out by linearizing the equation of motion. This theory stipulate that a bang-bang control is the optimal control system for a linear mechanism.

From an application standpoint, robot manipulators are mainly used for repeatable tasks. In this case planning a trajectory off-line is possible and can be optimized to increase the productivity performance. The performance index can be time, energy consumption or the like.

This section will highlight methods used for planning optimal motion trajectories using dynamic constraints. These algorithm may be grouped into two categories:

a) Using performance indices for generating an optimal time trajectory, where it is assumed that the path is given in parameterized form, and the solution is found based on the maximum principle[64].

b) Discretization of the state space using grid method.

### 2.4.1 Optimal control theory

Optimal control theory leads to a mathematic problem which can be solved by the calculus of variation (Courant and Hilbert 1937). This consist of solving Euler-Lagrange's equation which leads to minimizing an arbitrary cost function depending on a specified or desired performance index. For robot studies time or energy consumption are generally used. Due to the fact that the mathematical formulation of the problem lead to complex equations to solve, most of the proposed algorithm assume that the Cartesian path is given as a function of time.

Bobrow, J. et al[4] presented a method to generate a minimum time trajectory. He proposed to drive one joint, using bang-bang scheme, at its maximum allowable capacity, and that other joints velocities be modulated to coordinate the motion. The approach is suitable for generating quasi-optimal time trajectories for two and three links manipulator. Dubowsky and Shiller[9] proposed an extension for the above method for multi-degree manipulator (six degree). They found that it is extremely difficult to develop a minimum time trajectory for six degree manipulator based on optimal control theory.

Vukobratovic and Kircansk[59] developed an algorithm to minimize the energy

consumption along a Cartesian path by optimizing the velocity along the trajectory.

Schimtt et Al[51] formulated the optimal motion programming as a problem in calculus of variation using Raleigh-Ritz technique to approximate the cost function associated with the energy consumption. The optimized parameters are the joint motion parameters of the manipulator. The developed technique is applicable only for two or three degree manipulator.

In conclusion most of the proposed methods on optimal control are only suitable for one or two degree of freedom manipulators. This due to the fact that the resulting formulation is mathematically complex to solve, as result of the non-linearity of the dynamic equations, and the different constraints imposed by the state space.

### 2.4.2 Discretization of the state space

The following methods optimize the time by varying the Cartesian path and allowing the limiting joint to reach its maximum speed. Lin and Chang[32] proposed to optimize a certain performance index using a composite cubic polynomial functions trajectory. They increase the number of cubic functions until there is no change in the minimization procedure. The method proposed is based on a random walk method. As the number of sampling points increase the resulting path converge to the desired. The method is not totally efficient since there is no technique imposed to accelerate the convergency of the algorithm.

Shin and Mckay[55] proposed the dynamic programming approach to optimize the time of travel. First they approximate the geometric path by a parameterized

curve to reduce the dimensionality of the problem into two state variable (position and velocity). To apply dynamic programming they divide the phase plane (position and velocity) into a discrete grid and the cost of going from one point on the grid to the next is calculated. Since the specification of the geometric path affect the time optimization problem, the authors proposed another algorithm to approximate the geometric path that result in a minimum time trajectory between two points. A combination of both methods results in an optimal time trajectory. From the above study it can be concluded that two factors has to be considered to find a minimum time trajectory. First the choice of the geometric path in Cartesian space and second the dynamics of the manipulator. Other factors like the effect of the manipulator structure and the manipulator positioning at starting time on the time optimality has been studied by Scheinman and Roth[50].

# Chapter 3

## TRAJECTORY SPECIFICATION AND GENERATION FOR ROBOT ARMS

### 3.1 Introduction

In order to execute a useful tasks, a robot arm must move from an admissible initial position/orientation to an admissible final desired position/orientation. This requires that the robot arm's configurations at both the initial and final position must be known before planning the motion trajectory. Generally, there may exist an infinite number of trajectories between the two given positions. Some of these trajectories are infeasible because of motion constraints arising from arm geometry, work space limitations, and obstacles. Brady et al [4] discusses different methods for generating trajectories based on acceleration functions taking the form of cosine, quintic, and the sum of sine and linear function. They observed that a " critically damped " trajectory consisting of a sum of decaying exponentials of the form $e^{-\omega_d t}$, can be chosen to approximate the behavior of a damped second order oscillator.

This chapter is divided into two parts. The first part consist of presenting two methods for computing trajectories in joint space. The first method uses an expo-

nential approximation of Gauss function, and the second method uses a parameterized class of function in the form of a second order polynomial combined with cosine function. The second part deals with the approach to generate a Cartesian path that results in a minimum deviation from straight line path between two end points.

## 3.2 Trajectory Generation in the Joint Space Coordinate System

The study presented in this section aims at developing a trajectory whose acceleration curve can be easily modified to control the jerk at motion transition. Acceleration curves based on two mathematical functions are studied.

### 3.2.1 Acceleration curve based on Gauss function

The first approach uses an acceleration curve with the shape of Gauss function as shown in Figure 3.1, and is given by:

$$f(t) = \ddot{\theta}(t) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}(\frac{t-\mu}{\sigma})^2} \tag{3.1}$$

and the derivative, used as jerk is given by:

$$f'(t) = \dddot{\theta}(t) = -\frac{t-\mu}{\sqrt{2\pi}\sigma^3} e^{-\frac{1}{2}(\frac{t-\mu}{\sigma})^2} \tag{3.2}$$

With reference to Figure 3.1 the effect of the parameters $\mu$ and $\sigma$ on the behavior of the Gauss function are as follow:

- As $\sigma$ increases the curve moves down and becomes more flat, and as $\sigma$ decreases the curve moves upward and becomes more narrow and sharp.

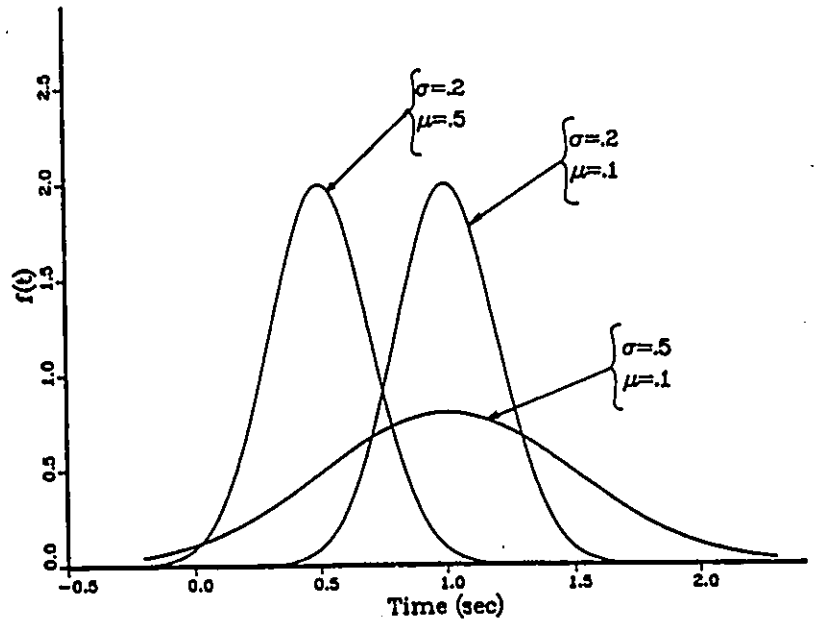



Figure 3.1: Gauss function

- As $\mu$ varies, the maximum value of the function is shifted by the magnitude of $\mu$.

Assuming that the acceleration profile has the form of the Gauss function, the trajectory can be obtained by integrating Equation 3.1 twice. The Gauss function has the form of $e^{-au^2}$ and can not be integrated directly. The function, however, can be first approximated using the Maclaurin series, and then the integration can be performed.

Equation 3.1 can be approximated as follows:

$$f(t) = \frac{1}{\sqrt{2\pi}\sigma} \sum_{n=1}^{\infty} (\frac{-1}{2})^n \frac{1}{n!} (\frac{t-\mu}{\sigma})^{2n} \tag{3.3}$$

The number of terms in the approximating Maclaurin series has to be properly specified in order to ensure that the error between the approximated polynomial function and the original Gauss function is acceptable. It should be noted here that a large order polynomial suffer from a number of drawbacks like large execution

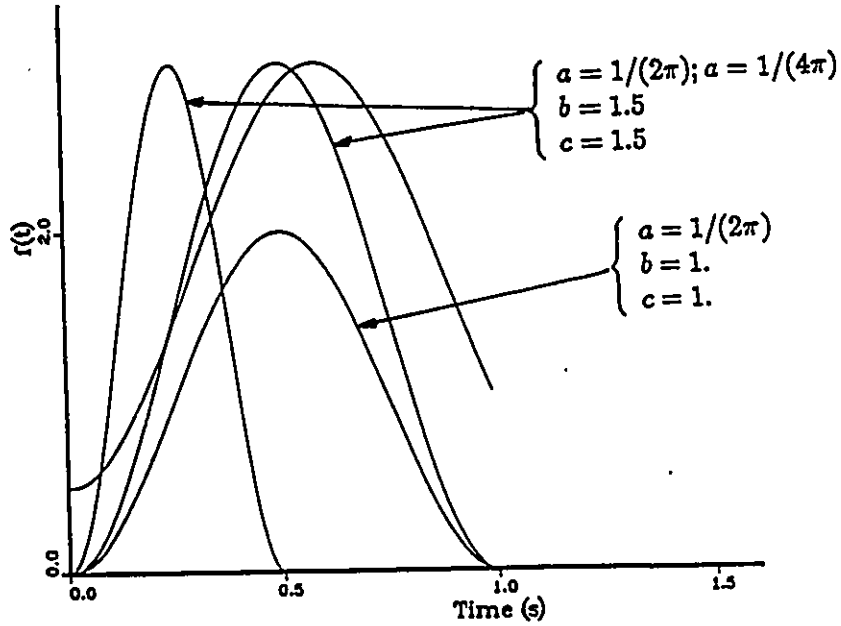



Figure 3.2: The cycloid function

time for trajectory generation, and degeneracy problems. The resulting position, velocity and acceleration using Gauss function as proposed acceleration profile are shown in Figure 3.3, and the development of the equations are given in Appendix A.

Another function that has a shape similar to the Gauss function and that can be manipulated through its different parameters is a member of the cycloid family, the curvate cycloid. The shape of this cycloid is given by:

$$f(t) = \ddot{\theta}(t) = c - b\cos(\frac{t}{a}) \tag{3.4}$$

and the derivative, called the jerk is given by:

$$f'(t) = \dddot{\theta}(t) = b/a\sin(\frac{t}{a}) \tag{3.5}$$

With reference to Figure 3.2, the effect of the parameter "a","b" and "c" on the shape of the curve given by Equation(3.2) are as follow:

- As "a" varies the period over which the cycloid is defined varies with a resulting variation in the curve width.
- the peak magnitude of the curve varies with changes in the scale factor "b".
- "c" is an offset and can be used to determine the initial jerk required.

These parameters allow the acceleration curve to be shaped in order to control the movement time, satisfy the constraints imposed by the initial and final configurations of the arm, and satisfy the physical constraints such as the maximum allowable velocity, acceleration, and jerk.

The velocity and position of a joint with an acceleration curve given by Equation 3.4 is obtained by integration as follows:

$$\dot{\theta}(t) = ct - ba\sin(\frac{t}{a}) + d \tag{3.6}$$

and

$$\theta(t) = \frac{c}{2}t^2 + dt + ba^2\cos(\frac{t}{a}) + e \tag{3.7}$$

Where d and e are the constant of the integration.

The variation in position, velocity and acceleration of the proposed function are plotted with other types of previous approaches as shown in Figure 3.3.

## 3.3 Trajectory Generation in Cartesian Space

The curve traced by the tip of the end effector in space while performing a task is called end effector path. The path may be defined as a function of a single parameter s, the distance along the curve. The distance parameter is generally

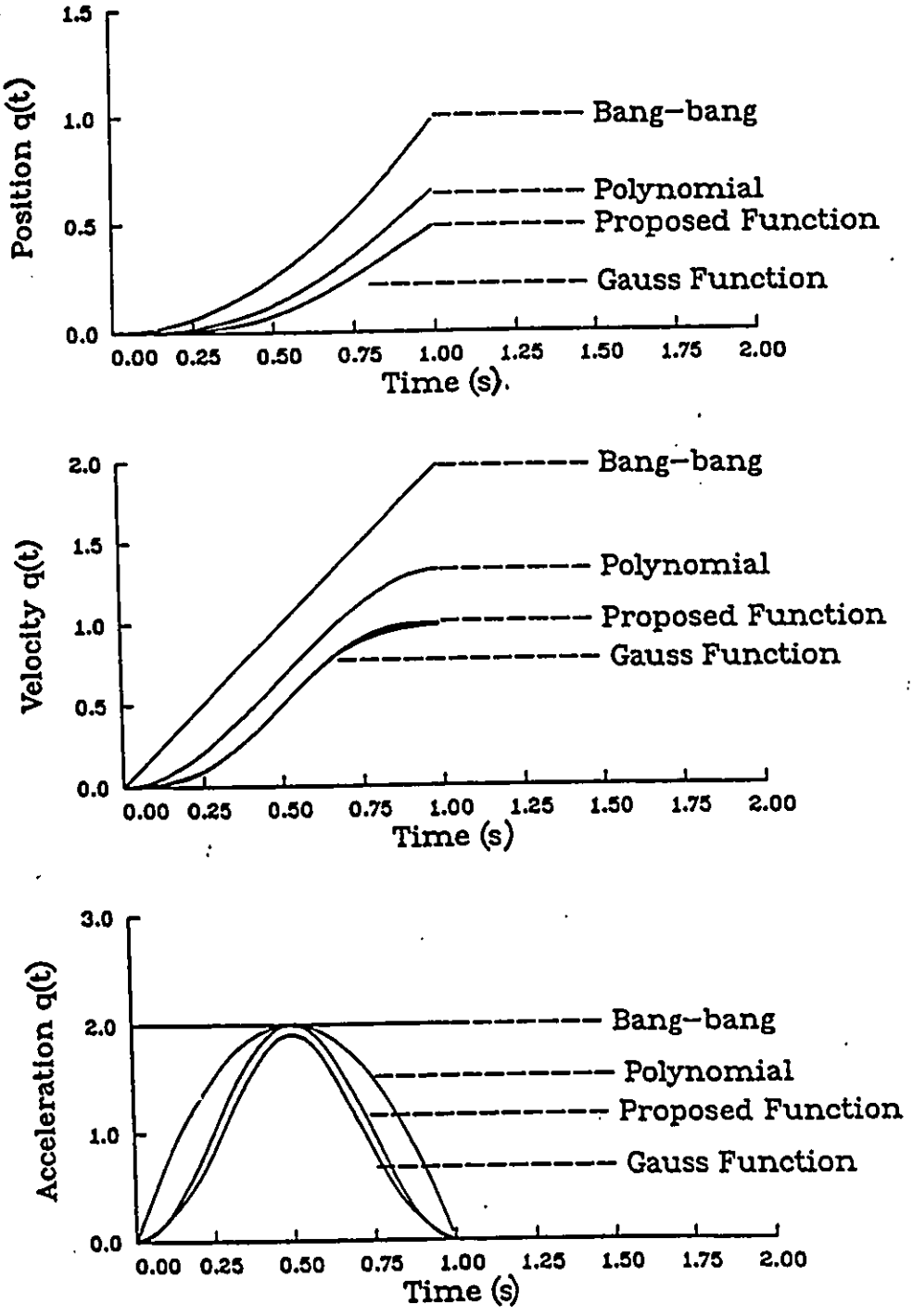



Figure 3.3: Position, velocity and acceleration of the proposed function

normalized so that $s = 0$ at the source configuration, and $s = 1$ at the destination. For trajectory generation time is used as the independent variable.

In general there are two types of robot motions used, rectilinear motion and curvilinear motion.

a) In rectilinear motion the robot end effector is commanded to move between points defined along its path in straight line segment. The directions of the velocity and the acceleration are along straight lines. Due to the dynamics of the robot arm, the end effector must stop at each of the points defined along the path in order to execute a straight line motion. Each of the straight line segments is divided into three zones; an acceleration zone, a constant velocity zone and deceleration zone.

b) Curvilinear motion is a motion along curved path as shown in Figure 3.4.b. The velocity is a vector along the tangent to the curve at a given point. The acceleration along the path is composed of two components, a tangential acceleration and a centrifugal one.

The vast majority of pick and place applications use rectilinear paths, and require trajectories in which the end effector is commanded to pass through visiting points in between the source and the destination. The visiting points may be required to veer the robot away from obstacles along its path. As the number of the visiting points increase the shape of the path joining them becomes more complex. In order to minimize the travel time the end effector is not allowed to stop at the visiting points. The end effector will have to deviate from the rectilinear path due to the robot dynamics. In collision avoidance problem the deviation from the

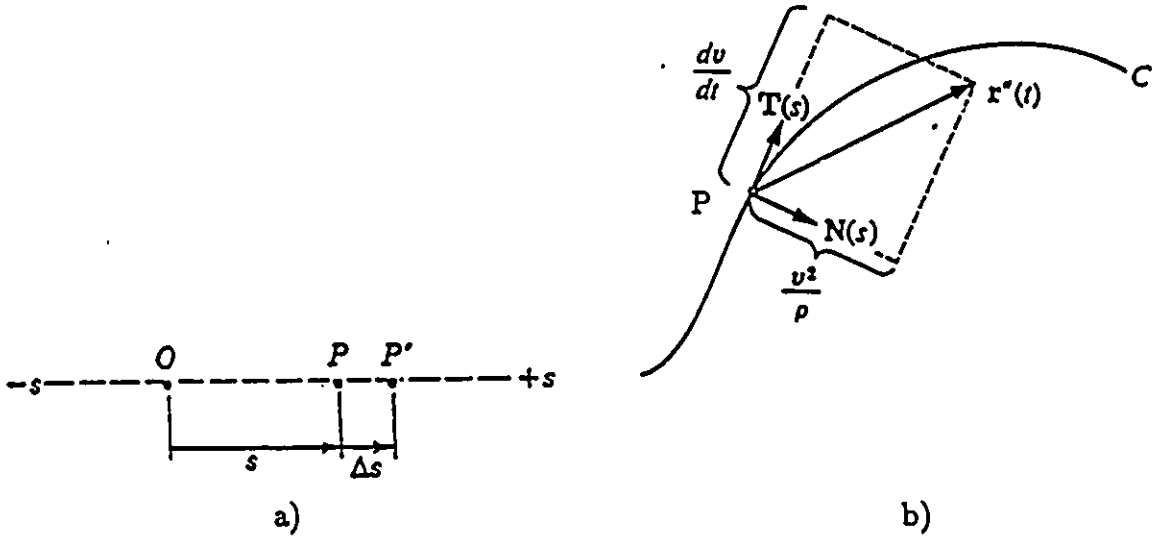



Figure 3.4: Two different types of motion, a) rectilinear motion, b) curvilinear motion.

rectilinear path has to be carefully evaluated to avoid collision.

Another approach to this problem is to use a single high order polynomial joining all the visiting points, and avoiding obstacles. The drawback of this approach is that once this high order polynomial trajectory is specified to satisfy end conditions, and to pass through a given number of points, any addition or removal of via point requires a new computation of the trajectory. Further, a high order polynomial is more complex and time consuming to compute.

The technique used to formulate the Cartesian path that passes through a given number of n via points without stopping at each one, and with a minimum curvature so that the path traced does not deviates much from straight line segments follows:

The path from the source to the destination points is first decomposed into

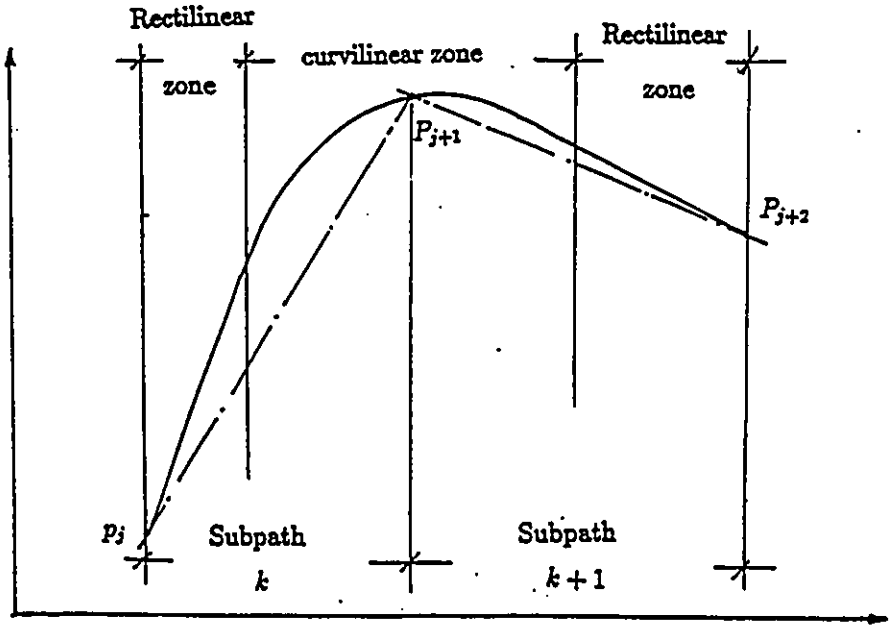



Figure 3.5: Decomposition of the robot path.

subpaths. Each subpath is further divided into two zones a rectilinear zone and a curvilinear zone as shown in Figure 3.5.

The rectilinear zone is described by a parametric equation function of the distance parameter s as follows:

$$\begin{cases} x_k(s) = x_j + (x_{j+1} - x_j)s \\ y_k(s) = y_j + (y_{j+1} - y_j)s \\ z_k(s) = z_j + (z_{j+1} - z_j)s \end{cases} \tag{3.8}$$

Where "i" referes to the knot points number. $(j = 1, 2....n)$ and "k" is the number of the subpaths. $(k = 1, 2, .....n-1)$.

With reference to Figure 3.5, the transition from one zone to the next is continuous in function, and in first and second derivatives. Furthermore, the curvilinear zone has to have a minimum curvature in order to reduce the discrepancy from

the straight line.

To generate the equation governing the curvilinear zone, first one has to start by describing the general form of the curves.

Any point on a curve can be represented with respect to a convenient reference by a vector $\vec{r}(s)$ as a function of the length parameter s.

$$\vec{r}(s) = x(s)\vec{i} + y(s)\vec{j} + z(s)\vec{k} \tag{3.9}$$

With reference to Figure 3.4.b, the unit tangent vector to the curve $C$ at point P is given by:

$$\vec{T}(s) = \frac{dr}{ds} = \frac{dx(s)}{ds}\vec{i} + \frac{dy(s)}{ds}\vec{j} + \frac{dz(s)}{ds}\vec{k} \tag{3.10}$$

The curvature of the curve at a point P is given by the rate of the change of the tangent to the length as follows:

$$\kappa(s) = \left| \frac{d\vec{T}}{ds} \right| \tag{3.11}$$

and its radius of curvature is given by:

$$\rho(s) = \frac{1}{\kappa(s)} \tag{3.12}$$

Where $\rho(s)$ is held in the direction of the normal $\vec{N}$.

Note that for straight line the curvature is equal to zero since the rate of change of the tangent is equal to zero, it follows that its radius of curvature tend to infinity.

Now suppose that s is a function of time, the unit tangent vector given by Equation 3.10 can be rewritten as follows:

$$\vec{T}(s) = \frac{dr}{dt}\frac{dt}{ds} = \frac{dr/dt}{ds/dt} \tag{3.13}$$

where $dt/ds$ must be positive, since both s and t increase. Consequently,

$$\vec{T}(s) = \frac{\vec{\dot{r}}}{|\,\dot{r}\,|} \tag{3.14}$$

where $\dot{r} = ds/dt$.

The curvature can be expressed as a function of t as follows:

$$\kappa(s) = \frac{|\,\vec{T}/dt\,|}{|\,\dot{r}\,|} \tag{3.15}$$

Differentiating Equation 3.14 and substitute the result into Equation 3.15 gives:

$$\kappa(t) = \frac{|\,d^2\vec{r}/dt^2\,|}{|\,\dot{r}\,|^2} \tag{3.16}$$

The numerator in Equation 3.16:

$$\frac{d^2\vec{r}}{dt^2} = \frac{d^2x}{dt^2}\vec{i} + \frac{d^2y}{dt^2}\vec{j} + \frac{d^2z}{d^2t}\vec{k} \tag{3.17}$$

is the tangential acceleration.

Thus in order to obtain the smallest possible curvature Equation 3.17 should be minimized. Asume that the function governing the curvilinear zone is a polynomial of degree n represented by the following parametric equation:

$$\begin{cases} x_k(s) = \sum_{m=0}^{N} a_m s^m \\ y_k(s) = \sum_{m=0}^{N} b_m s^m \\ z_k(s) = \sum_{m=0}^{N} c_m s^m \end{cases} \tag{3.18}$$

The resulting path given by the above system of equations has to satisfy the following constraints:

1. Containts the via points.

2. Has minimum curvature (minimizing the Equation 3.17).

3. Ensure the continuity of the rectilinear motion.

## 3.4 Summary

The generation of the motion trajectory in Cartesian space or in joint space is discussed. To achieve a controllable and feasible trajectory, a cycloid function of time is proposed as an acceleration profile in joint space. For Cartesian space, the basic idea to accomplish a smooth and optimal curve with minimum curvature is presented. The computation of the parameters for the equations governing the motion trajectories either in joint space or Cartesian space are presented in the coming chapters.

# Chapter 4

## TRAJECTORY PLANNING

### 4.1 Introduction

In this chapter, the trajectory function developed in the last chapter for joint space, is used to plan the motion trajectory. The first and second section deal respectively with the joint interpolated subtrajectories and the computation of the different coefficients of the functions governing the subtrajectory segments.

### 4.2 Joint Interpolated Subtrajectories

The approach used here to plan a joint interpolated motion utilizes the function given by Equation 3.4 for the transition segment at the beginning and end of the subtrajectories. A linear function in time is used for the constant velocity segment. For each subtrajectory the transition segments may take the form of an acceleration, a deceleration, or may disappear entirely depending on the adjacent subtrajectories. A hypothetical subtrajectory shown in Figure 4.1 is composed of three types of motion, an acceleration in the first segment, constant velocity motion in the second segment, and a deceleration in the third segment. The three segments are linked together by appropriate boundary conditions to ensure

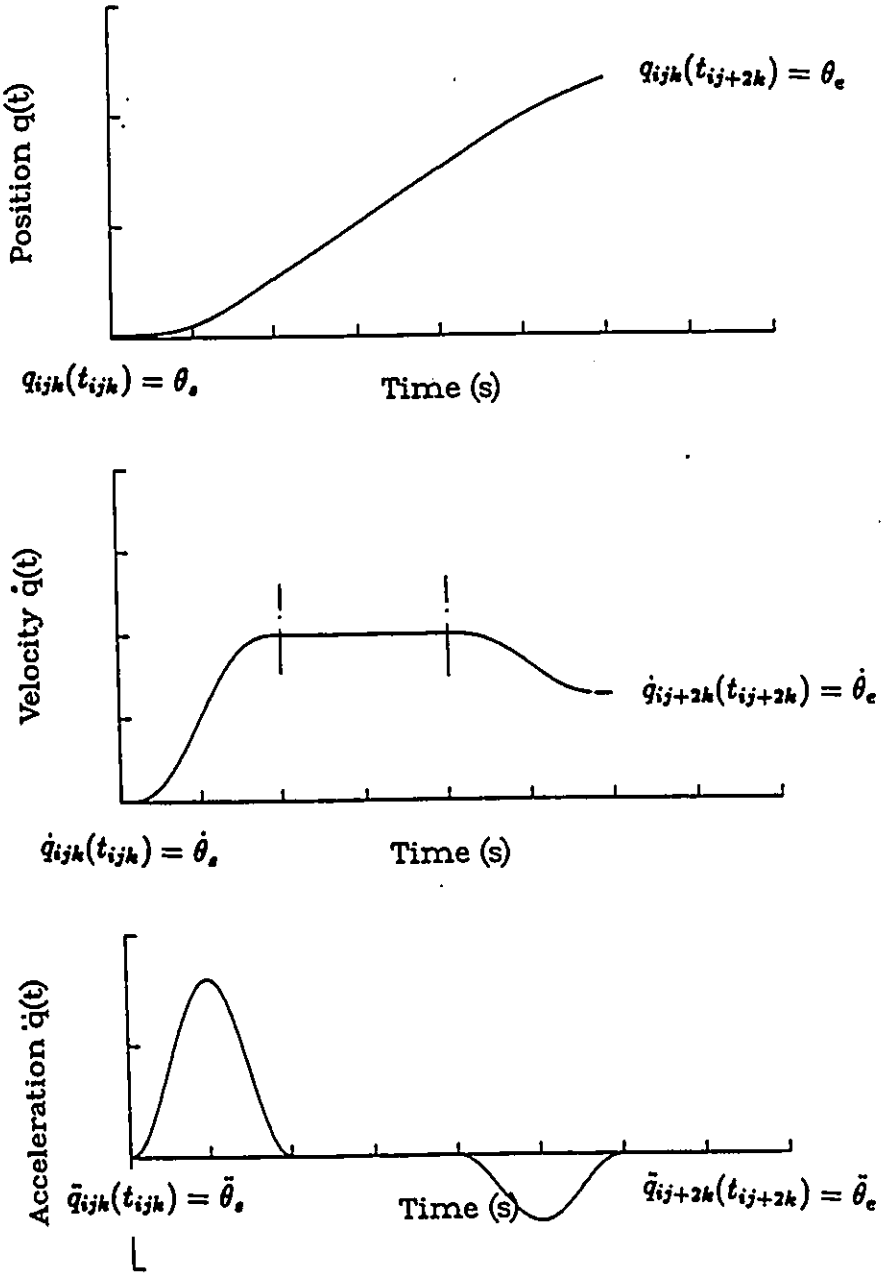



Figure 4.1: Types of motion of a hypothetical subtrajectory.

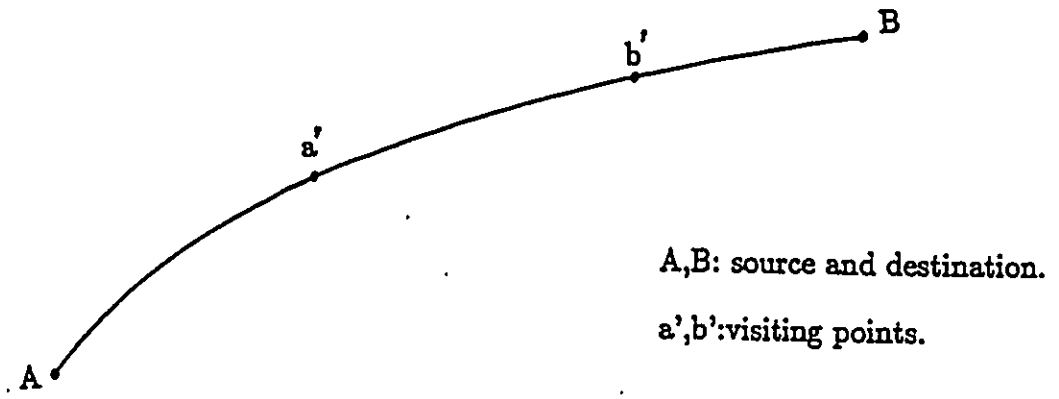



Figure 4.2: The definition of visiting points

continuity of position, velocity and acceleration. When planning a robot end effector trajectory a set of visiting points are defined in Cartesian space that would cause the robot arm to veer from objects along its path and avoid collision. These points are transformed into the joint space and become part of the set of the knot points that bound the subtrajectories. As an example, assume the trajectory shown in Figure 4.2. Points "A"and "B" are the source and destination of the end effector and "a"and "b" are the visiting points. Several approaches may be used to generate the motion from the source to the destination.

a) Each subtrajectory is composed of two segments, a transition segment followed by a constant velocity one as shown in Figure 4.3.

b) Each visiting point is stradled by a transition segment that adjusts the velocity between sets of visiting points. The transition segments are linked by constant velocity segments. This approach is described by Paul[42] and shown in

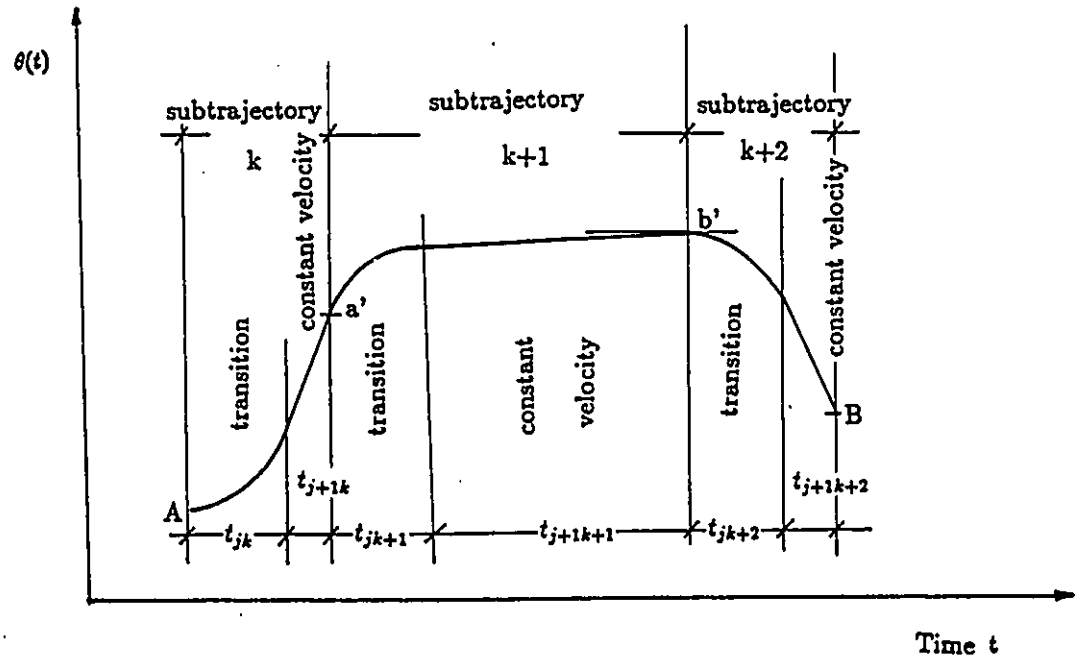



Figure 4.3: Path transition for approach "a".

Figure 4.4.

c) Each subtrajectory is composed of three segments, namely; lift-off, cruise and set down.

In the present study the first approach will be used to generate the motion for a given subtrajectory.

The conditions that the set of joint subtrajectory functions must satisfy are as follow:

Let "i" refers to the joint, "j" refers to the segment number and "k" to the subtrajectory number.

a) For transition segment:

1. Initial position $q_{ijk}(t_j) = \theta_{a'}$.

2. Initial velocity $\dot{q}_{ijk}(t_j) = \dot{\theta}_{a'}$.

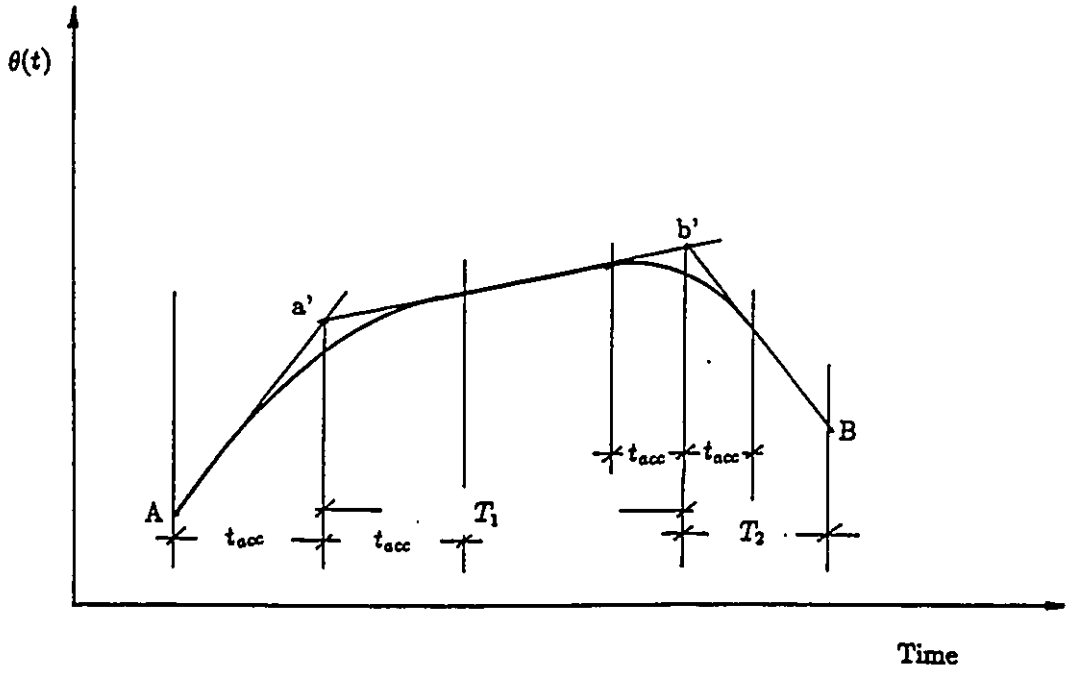



Figure 4.4: Path transition for approach "b", after Paul[42].

3. Initial acceleration $\ddot{q}_{ijk}(t_j) = \ddot{\theta}_{a'}$.

b) For constant velocity segment:

1. Continuity in position at $t_j$ (i.e $q_{ijk}(t_j) = q_{ij+1k}(t_{j+1})$).

2. Continuity in velocity at $t_j$ (i.e $\dot{q}_{ijk}(t_j) = \dot{q}_{ij+1k}(t_{j+1})$).

3. Continuity in acceleration at $t_j$ (i.e $\ddot{q}_{ijk}(t_j) = \ddot{q}_{ij+1k}(t_{j+1})$).

These conditions must be satisfied for all subtrajectories from k=1 to k=n-1.

At this time its convenient to introduce a normalized time to treat the equations of each subtrajectory segment.

To start definig first the following variables:

T: normalized time, $T \in [0, 1]$.

t: real time in second.

$t_{ijk}$: real time at the end of the $j^{th}$ subtrajectory segment of the $i^{th}$ joint .

$t_j = t_{ijk} - t_{ij-1k}$: real time for the $j^{th}$ segment.

Therefore the normalized time is given by:

$$T = \frac{t - t_{ij-1k}}{t_{ijk} - t_{ij-1k}} \tag{4.1}$$

Where $t \in [t_{ij+1k}, t_{ijk}]$ and $T \in [0, 1]$.

### 4.2.1 Transition Segment

The transition segment is bounded by two constant velocity segments as shown in Figure 4.3, which impose a zero acceleration at the start and the end of the segment. In addition to this boundary condition, constraints on the maximum allowable velocity and acceleration have to be imposed when deriving the coefficients of the transition segment function.These constraints are necessary to prevent planning a trajectory that would require the robot joint to accelerate beyond the torque capability of its drive.

With reference to Figure 4.1 the conditions that this set of joint segment function must satisfy are the following:

$$q_{ijk}(t=0) = \theta_s \qquad q_{ijk}(t=1) = \theta_e$$

$$\dot{q}_{ijk}(t=0) = \dot{\theta}_s \qquad \dot{q}_{ijk}(t=1) = \dot{\theta}_e$$

$$\ddot{q}_{ijk}(t=0) = \ddot{\theta}_s \qquad \ddot{q}_{ijk}(t=1) = \ddot{\theta}_e$$

The subscripts s and e refer to start and end of the transition segment.

It is clear that the coefficients depend on the subscript. Without loss of generality, the subscript "i" and "k" will be omitted to simplify the presentation of the

equations. The first derivative of Equation 3.7 with respect to real time t can be written as:

$$\dot{q}_j = \frac{dq_i(T)}{dt} = \frac{dq_j(T)}{dT} \cdot \frac{dT}{dt} \tag{4.2}$$

where $j = 1, 2, ...n - 1$.

From Equation (4.1)

$$\frac{dT}{dt} = \frac{1}{t_{j+1k} - t_{jk}} = \frac{1}{t_j}$$

Substitute for $dT/dt$ into Equation 4.2 and substituting the result in Equation 3.6 gives:

$$\dot{q}_j(T) = \frac{1}{t_j}\left[ct - ba\sin(\frac{T}{a}) + d\right] \tag{4.3}$$

Similarly, the acceleration equation resulting from Equation 3.4 is given by:

$$\ddot{q}_j(T) = \frac{1}{t_j^2}\left[c - b\cos(\frac{T}{a})\right] \tag{4.4}$$

and also the equation of third derivative "jerk" is given by:

$$\dddot{q}_j(T) = \frac{1}{t_j^3}\left[\frac{b}{a}\sin(\frac{T}{a})\right] \tag{4.5}$$

Applying the boundary conditions stated above to the transition segment, the coefficients of Equations 3.7, 4.3 and 4.4 are as follow:

$$\cos(\frac{1}{a}) = 1. - 2.\frac{\ddot{\theta}_s}{\alpha_{max} - \ddot{\theta}_s} \tag{4.6}$$

$$b = c = t_j(\dot{\theta}_e - \dot{\theta}_s) \tag{4.7}$$

$$d = t_j\dot{\theta}_s \tag{4.8}$$

$$e = \theta_s - ba^2 \tag{4.9}$$

Where $\alpha_{max}$ is the maximum allowable acceleration and $t_j$ is the transition time given by:

$$t_j = 2.\frac{\dot{\theta}_s - \dot{\theta}_e}{\alpha_{max}} \tag{4.10}$$

### 4.2.2 Constant Velocity Segment

For constant velocity segment the accelerations at both boundaries are equal to zero. This results in a first order polynomial function of time given by:

$$q_{ij+1k}(t) = a_{1j+1}t + a_{0j+1} \tag{4.11}$$

For a continuis motion, one has to ensure the continuity in position and velocity. It follows that the coefficients of the above polynomial are given by:

$$a_{1j+1} = \dot{\theta}_e T_{j+1} \tag{4.12}$$

$$a_{0j+1} = d + \frac{c}{2} + \dot{\theta}_s \tag{4.13}$$

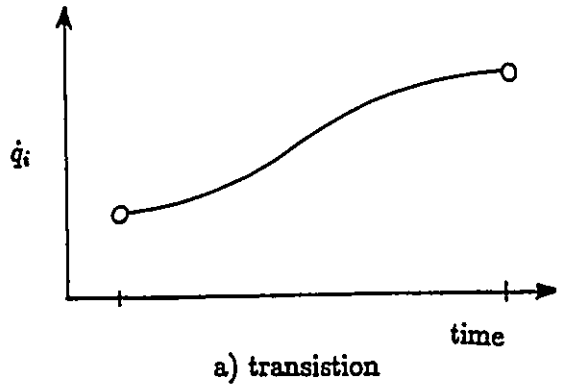
### 4.2.3 Motion Coordination for the Robot Joints

In general there are three types of motion to cover a subtrajectory: One transition segment, one transition segment followed by a linear segment, or two transition segments and one constant velocity segment as illustrated in Figure 4.5.

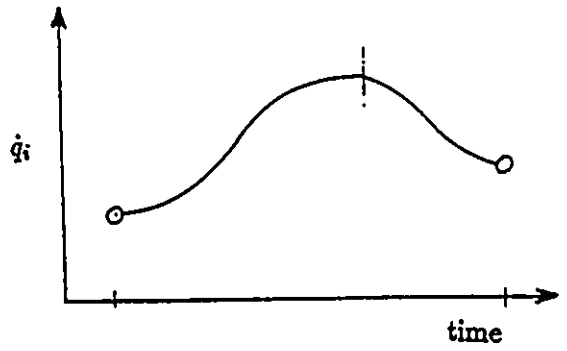
Since all the robot joint must pass by the visiting points simultaneously. The minimum travel time between two visiting points is equal to the largest time taken by the different joints, thus

$$T_{i,min} = max[\sum_{j=1}^{3} T_{ijk}]_{i=1}^{n}$$

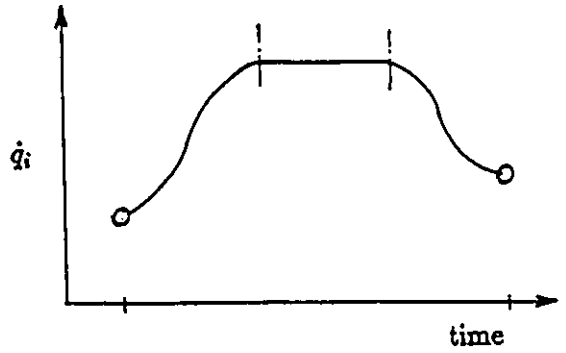
where "i" is the joint with the longest travel time. Joints other than "i", will have to be slowed down and forced to cover the subtrajectory at the same time as the joint "i", so that they would all converge to the same point at the same time. The approach used to slow these joints down consists of accelerating the joint at the maximum available acceleration then travel at a velocity below the maximum allowable.




a) transistion




b) two transitions




c) two transitions + constant velocity

Figure 4.5: Different types of robot motion for constrained joint.

# Chapter 5

## MANIPULATOR DYNAMICS FOR PATH DESIGN

### 5.1 Introduction

The position and orientation of the robot end effector can be described in terms of the position and orientation of a coordinate frame attached to the end effector with respect to an inertial reference coordinate frame. This treatment is called the kinematics of robot manipulation.

Kinematics is one of the most important areas of the robot engineering because robotics manipulation can be achieved only by the motion of the robot hand and associated parts. Also it relates the motion in joint space to the Cartesian space which is very important since the robots are servoed in joint space. The first section of this chapter deals with kinematics. The second section deals with the governing equations pertaining to the dynamics of robotic manipulators. Essentially, dynamic equations are needed for the dynamic control of robotic manipulator, such as the torques or forces required to drive the members of the manipulator. In the second section the physical model for the robotic dynamics is obtained, and used in the third section to design the path in Cartesian space.

## 5.2 Kinematics

### 5.2.1 Description of the manipulator

The robot to be considered has three revolute joints arranged as shown in Figure 5.1. The joint axis for this manipulator are derived from the Denavit-Hartenberg specification[18]. With reference to Figure 5.1 The rotation axis $z_i$ corresponds to joint angle $\theta_{i+1}$ between links i and $i+1$. The internal coordinate system for link i is completed by defining $x_i$ from the cross product $z_{i-1} \times z_i$, which also locates the origin, and $y_i = z_i \times x_i$. Successive coordinate system are related by the following three parameters:

1. $a_i$ is the distance between $z_{i-1}$ and $z_i$ measured along $x_i$
2. $d_i$ is the distance between $x_{i-1}$ and $x_i$ measured along $z_{i-1}$
3. $\alpha_i$ is the angle between $z_{i-1}$ and $z_i$ measured in a right-hand sense about $x_i$.

For coordinate system three, the rotation axis $z_3$ is desired to point at the wrist to facilitate the decomposition of the position of the manipulator arms into the wrist position and the end effector position[17].

Once the link coordinate frame have been assigned to the manipulator, the homogeneous transformation between links can be expressed as function of the joint actuator coordinate. These transformations are referred to as $A_i$ matrices, where the subscript refers to the transformation between the ith link in terms of the (i-1)th link. The product of the $A_i$ matrices from 1 to 3 is called $T_3$ and describes the end of a 3 degrees of freedom manipulator in terms of its base.

Table 5.1: Link parameters for rotary arm of Figure 5.1

| Links | Variable | a | d | $\alpha$ |
|---|---|---|---|---|
| 1 | $\theta_1$ | 0. | $d_1$ | +90. |
| 2 | $\theta_2$ | $a_2$ | 0. | 0. |
| 3 | $\theta_3$ | 0. | $d_3$ | +90. |

$$T = T_3 * E \tag{5.1}$$

E describes the transformation of the end effector with respect to $T_3$. The rotation matrix $A_i$ that transforms point expressed in link i coordinates to link i-1 coordinates is:

$$^{i-1}A_i = \begin{bmatrix} c_i & -s_i c_{\alpha_i} & s_i s_{\alpha_i} \\ s_i & c_i c_{\alpha_i} & -c_i s_{\alpha_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} \end{bmatrix} \tag{5.2}$$

Where the abbreviations $s_i = sin\theta_i$, $c_i = \theta_i$ are used.

The various constant link parameters for the rotary arm of Figure 5.1 are given in table 5.1. It follows that the corresponding joint transformation matrices for each coordinate system are:

$$^{0}A_1 = \begin{bmatrix} c_1 & 0 & s_1 \\ s_1 & 0 & -c_1 \\ 0 & 1 & 0 \end{bmatrix}$$

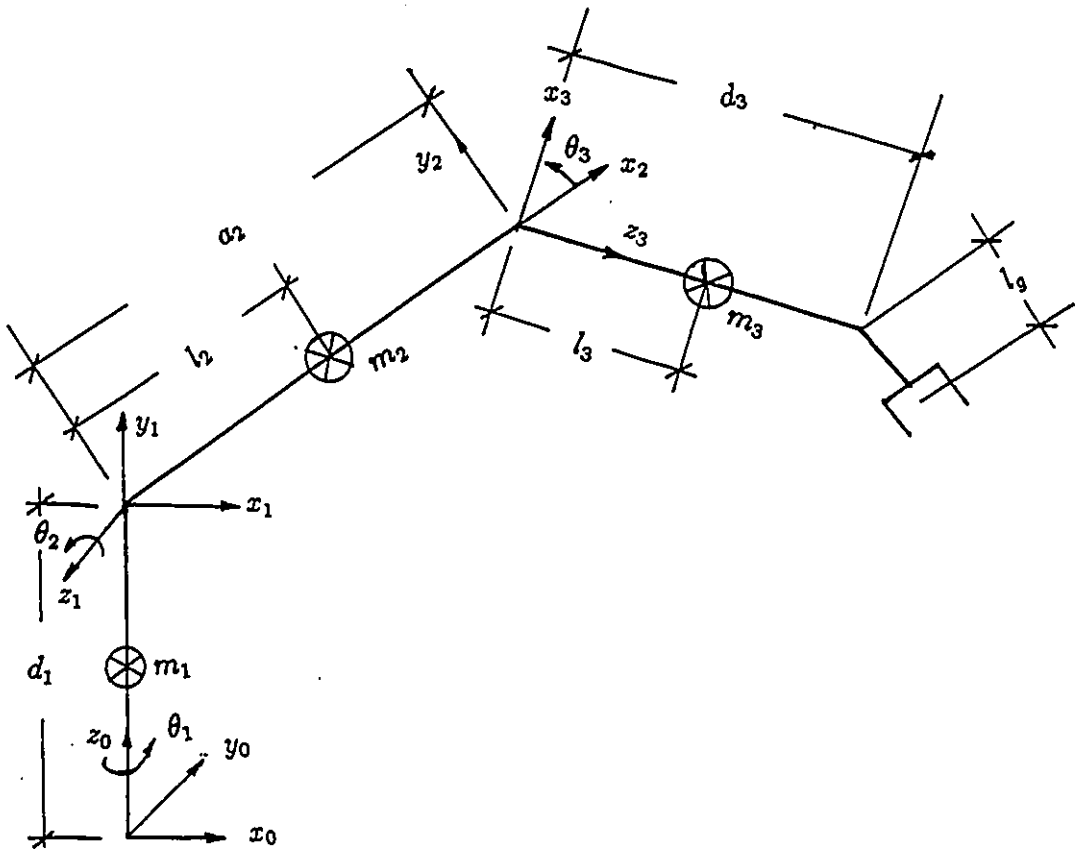



Figure 5.1: Coordinate frames for 3R manipulator

$$ {}^{1}A_2 = \begin{bmatrix} c_2 & -s_2 & 0 \\ s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{2}A_3 = \begin{bmatrix} c_3 & 0 & s_3 \\ s_3 & 0 & -c_3 \\ 0 & 1 & 0 \end{bmatrix} $$

### 5.2.2 Forward kinematic positions

Given the robot characteristics and the joint space coordinate, one can buildup the transformation from the base coordinate to the end effector by rotating successively about the joint and translating along the links until the end effector is reached. With reference to Figure 5.2 the position vector R is calculated as follow:

$$ R = W_1 + P \tag{5.3} $$

Where $W_1$ is given by:

$$ W_1 = (-s\theta_1 W_{2z}, c\theta_1 W_{2z}, d_1 + W_{2xy}) \tag{5.4} $$

and $W_2$ is given by:

$$ W_{2xy} = a_2 s\theta_2 + a_3 c(\theta_2 + \theta_3) \tag{5.5} $$

$$ W_{2z} = a_2 c\theta_2 + a_3 s(\theta_2 + \theta_3) \tag{5.6} $$

$W_2$ is the position of the wrist with respect to the frame ( shoulder) as shown in Figure 5.2 and is given by:

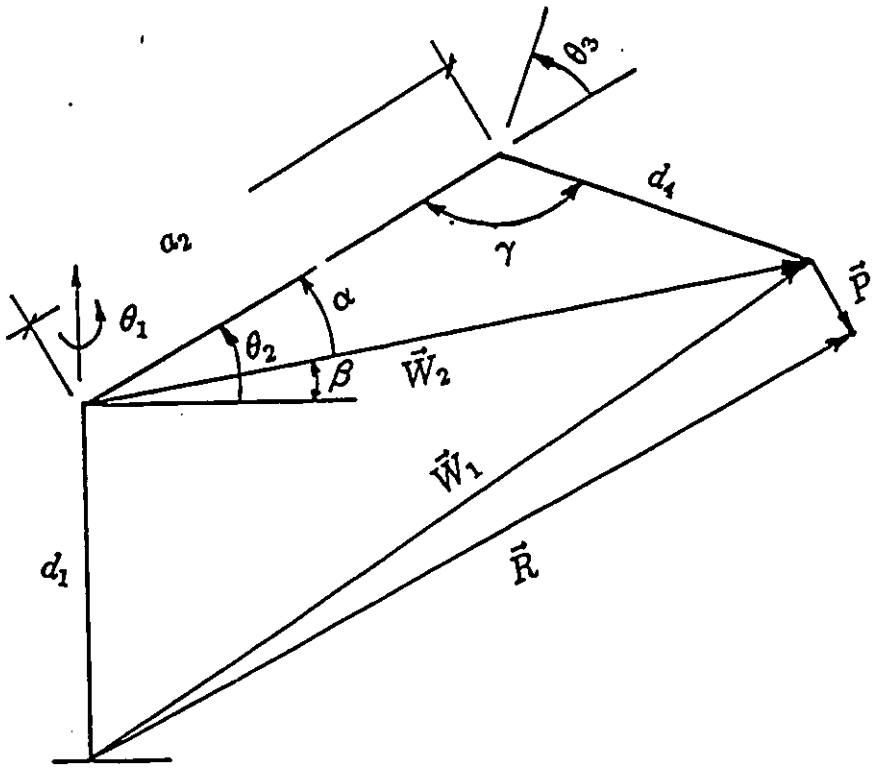



Figure 5.2: Vector definition between various coordinates of the rotary manipulator.

$$W_2 = W_1 - d_1 k_0 \tag{5.7}$$

### 5.2.3 Inverse kinematic positions

For the inverse kinematics, given the robot charateristics and the task space, one can find the joint space coordinates. Joint 1 is directly found from $W_1$, since joint 2 and 3 act in a plane that does not change the projection of the wrist into the $x_0$, $y_0$ plane.

$$\theta_1 = tan^{-1}(\frac{W_{1y}}{W_{1x}}) \tag{5.8}$$

A degeneracy occurs when the wrist lies on the $z_0$ axis, where $W_{1x} = W_{1y} = 0$ and $\theta_1$ can assume any value.

The next two angles are easily found using the trigonometric relationship for the triangle ($a_2$, $d_4$, $W_2$) of Figure 5.2 and as follow:

$$\gamma = cos^{-1}(\frac{a_2 + d_4^2 - W_2^2}{2.a_2 d_4}) \tag{5.9}$$

Which leads to:

$$\theta_3 = \gamma + \pi/2 \tag{5.10}$$

Similarly the joint angle $\theta_2$ it is given by:

$$\alpha = cos^{-1}(\frac{a_2 - d_4^2 + W_2^2}{2.a_2 d_4}) \tag{5.11}$$

which gives:

$$\theta_2 = tan^{-1}(\frac{\sqrt{W_{2x}^2 + W_{2y}^2}}{W_{2z}}) + \alpha \tag{5.12}$$

### 5.2.4 Forward kinematic velocities

Using the generalized speeds $\dot{\theta}_i$ i=1,..3 implicitly one can write the different angular velocities as follows:

$$ {}^{1}\omega_1 = \dot{\theta}_1^1 A_0^0 k_0 = \begin{bmatrix} c_1 & -s_1 & 0 \\ 0 & 0 & 1 \\ s_1 & -c_1 & 0 \end{bmatrix} \begin{pmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{pmatrix} = \begin{pmatrix} 0 \\ \dot{\theta}_1 \\ 0 \end{pmatrix} \tag{5.13} $$

$$ {}^{2}\omega_2 = {}^{2} A_1({}^{1}\omega_1 + \dot{\theta}_2^1 k_1) \tag{5.14} $$

$$ = \begin{bmatrix} c_2 & s_2 & 0 \\ -s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \left\{ \begin{pmatrix} 0 \\ \dot{\theta}_1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \dot{\theta}_2 \end{pmatrix} \right\} = \begin{pmatrix} c_2\dot{\theta}_1 \\ c_2\dot{\theta}_1 \\ \dot{\theta}_2 \end{pmatrix} $$

$$ {}^{3}\omega_3 = {}^{3} A_2({}^{2}\omega_2 + \dot{\theta}_3^2 k_2) \tag{5.15} $$

$$ = \begin{bmatrix} c_3 & s_3 & 0 \\ 0 & 0 & 1 \\ s_3 & -c_3 & 0 \end{bmatrix} \left\{ \begin{pmatrix} c_2\dot{\theta}_1 \\ c_2\dot{\theta}_1 \\ \dot{\theta}_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \dot{\theta}_3 \end{pmatrix} \right\} = \begin{pmatrix} s_{23}\dot{\theta}_1 \\ \dot{\theta}_2 + \dot{\theta}_3 \\ -c_{23}\dot{\theta}_1 \end{pmatrix} $$

The linear velocity of the frame i in base frame are as follow:

$$ {}^{0}V_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \tag{5.16} $$

$$ {}^{0}V_2 = {}^{0} A_1^1 A_2^2 V_2 = {}^{0} A_1^1 A_2({}^{2}\omega_2 \times a_2^2 i_2) \tag{5.17} $$

$$ = \begin{pmatrix} -c_1 s_2 a_2\dot{\theta}_2 - s_1 c_2 a_2\dot{\theta}_1 \\ -s_1 s_2 a_2\dot{\theta}_2 + c_1 c_2 a_2\dot{\theta}_1 \\ c_2 a_2\dot{\theta}_2 \end{pmatrix} $$

$$ {}^{0}V_3 = {}^{0} A_1^1 A_2^2 A_3^3 V_3 = {}^{0} A_1^1 A_2^2 A_3({}^{3}\omega_3 \times d_4^3 k_3) \tag{5.18.a} $$

Let:

$$\begin{aligned} Z_1 &= d_4 c_{23} \\ Z_2 &= d_4 s_{23} \\ Z_3 &= d_4 s_{23} + c_2 a_2 \\ Z_4 &= d_4 c_{23} - s_2 a_2 \end{aligned}$$

$$^0V_3 = \begin{pmatrix} -Z_3 s_1 \dot{\theta}1 + Z_4 c_1 \dot{\theta}_2 + Z_1 c_1 \dot{\theta}_3 \\ Z_3 c_1 \dot{\theta}1 + Z_4 s_1 \dot{\theta}_2 + Z_1 s_1 \dot{\theta}_3 \\ Z_3 \dot{\theta}_2 + Z_2 \dot{\theta}_3 \end{pmatrix} \tag{5.18.b}$$

The linear velocity of the center of mass $V_i^*$ is given in the same manner as the linear velovity of the frames with the exception that the distance of the center of mass of each link is taken into account, (i.e change $a_2$ in $^2V_2$ with $l_2$ and $d_4$ in $^3V_3$ with $l_3$).

### 5.2.5 Forward kinematic accelerations

Using the generalized angular accelerations $\ddot{\theta}_i$ $i = 1,..3$ implicitly one can write the different angular accelerations as follows:

$$^1\dot{\omega}_1 = \ddot{\theta}_1^1 A_0^0 k_0 = \begin{bmatrix} c_1 & -s_1 & 0 \\ 0 & 0 & 1 \\ s_1 & -c_1 & 0 \end{bmatrix} \begin{pmatrix} 0 \\ 0 \\ \ddot{\theta}_1 \end{pmatrix} = \begin{pmatrix} 0 \\ \ddot{\theta}_1 \\ 0 \end{pmatrix} \tag{5.19}$$

$$^1\dot{\omega}_2 = ^1\dot{\omega}_1 + \ddot{\theta}_1^1 k_1 + ^1\omega_1 \times \dot{\theta}_2^1 k_1 \tag{5.20}$$

$$^2\dot{\omega}_2 = ^2A_1^1 \dot{\omega}_2 \tag{5.21}$$

$$^2\dot{\omega}_2 = ^2\dot{\omega}_2 + \ddot{\theta}_2^2 k_2 + ^2\omega_2 \times \dot{\theta}_2^2 k_2 \tag{5.22}$$

$$^3\dot{\omega}_3 = ^3 A_2^2 \dot{\omega}_3 \tag{5.23}$$

The linear acceleration of the frame i with respect to frame i-1 and of the center of masses are as follow:

$$^1a_1^* = 0. \tag{5.24}$$

$$^2a_2^* = ^2 \dot{\omega}_2 \times l_2^2 i_2 + ^2 \omega_2 \times (^2\omega_2 \times l_2^2 i_2) \tag{5.25}$$

$$^2a_{2,3} = ^2 \dot{\omega}_2 \times a_2^2 i_2 + ^2 \omega_2 \times (^2\omega_2 \times a_2^2 i_2) \tag{5.26}$$

$$^3a_{2,3} = ^3 A_2^2 a_{2,3} \tag{5.27}$$

$$^3a_3^* = ^3 a_{2,3} + ^3 \dot{\omega}_3 \times d_4^3 k_3 + ^3 \omega_3 \times (^3\omega_3 \times d_4^3 k_3) \tag{5.28}$$

## 5.3 Dynamic analysis

The dynamic equations developed here are based on Lagrangian formulation, which is also known as work-energy method. The dynamic behavior of a system is described in terms of the work done by all the external forces acting on the system and by the total energy stored in it.

### 5.3.1 Theoretical background

Consider a robot arm composed of N movable rigid body which are attached to a fixed frame. The kinetic energy of segment i, $T_i$ is the sum of its energy of translation with respect to its mass center and the kinetic energy of rotation about this mass center. For segment i, we have:

$$K_i = \frac{1}{2} tr(V_i M_i V_i^T) + \frac{1}{2} \omega_i^T I_i \omega_i \tag{5.29}$$

in which $M_i$ and $I_i$ are the mass and the inertia tensors, and $V_i$ and $\omega_i$ are the translation and rotation angular velocities respectively.

The total kinetic energy in the system is given by:

$$K = \sum_i^N K_i \tag{5.30}$$

The potential enrgy is a function of the generalized coordinates $\theta_i$ only, which can be written as:

$$U = \sum_i^N F_i(\theta_1, ...., \theta_N) \tag{5.31}$$

The Lagrangian L is defined as:

$$L = K - U \tag{5.32}$$

and is a function of the generalized coordinates $q = (q_1, ...., q_N)^T$ and $\dot{q} = (\dot{q}_1, ...., \dot{q}_N)^T$.

According to Lagrange dynamics, the equation of motion can be written as:

$$\frac{d}{dt}(\frac{\delta L}{\delta \dot{q}_i}) - (\frac{\delta L}{\delta q_i}) = Q_i \tag{5.33}$$

Where $Q_i$ is the work of all the external forces acting on the system.

The general form of $Q_i$ is as follow:

$$\begin{aligned} Q_i = & \sum_{j=1}^{N} \sum_{k=1}^{j} A_{ijk}(\theta_1, ...\theta_j)\ddot{\theta}_k \\ & + \sum_{j=1}^{N} \sum_{k=1}^{j} \sum_{i=1}^{j} \delta A_{ijk}(\theta_1, ...., \theta_j)/\delta\dot{\theta}_i\dot{\theta}_k\dot{\theta}_i \\ & - \sum_{j=1}^{N} \delta F_j(\theta_1, ...., \theta_i)/\delta\theta_i \end{aligned} \tag{5.34}$$

There are two intrinsic problems involved in the above equation regarding the manipulator dynamics, complexity and non linearity.

### 5.3.2 Dynamic equations of 3 R manipulator

The robot used in the analysis is of the serial drive type. Figure 5.1 shows a schematic representation of the manipulator. Each link of the manipulator is assumed to have an inertia given by:

$$I_i = \begin{bmatrix} I_{ix} & 0 & 0 \\ 0 & I_{iy} & 0 \\ 0 & 0 & I_{iz} \end{bmatrix} \tag{5.35}$$

where $I_i$ is the moment of inertia along the principle axis that are parallel to the direction of frame i.

All the elements in system are assumed to be rigid and that the effect of friction and backlash are negligible. By applying Equations 5.29 and 5.31 the total kinetic and potential energy in the system can be written as:

$$K = \sum_i^N K_i = \tag{5.36}$$

$$\tfrac{1}{2}m_2 l_2^2(c_2^2\dot{\theta}_1^2 + \dot{\theta}_2^2)+$$

$$\tfrac{1}{2}m_3(l_3^2\dot{\theta}_3^2 + (a_2^2 + l_3^2 + 2l_3a_2s_3)\dot{\theta}_2^2+$$

$$(l_3s_{23} + a_2c_2)\dot{\theta}_1^2 + 2(l_3^2 + a_2l_2s_3)\dot{\theta}_2\dot{\theta}_3)+$$

$$\tfrac{1}{2}\{(I_{1y} + I_{2x}s_2^2 + I_{2y}c_2^2 + I_{3x}s_{23}^2 + I_{3z}c_{23}^2)\dot{\theta}_1^2+$$

$$(I_{2z} + I_{2y})\dot{\theta}_2^2 + I_{3y}\dot{\theta}_3^2 + 2I_{3y}\dot{\theta}_2\dot{\theta}_3\}$$

The potential energy is as follows:

$$U = m_2l_2gs_2 + m_3g(l_3s_{23} + a_2s_2) \tag{5.37}$$

Substituting Equations 5.36 and 5.37 into Equation 5.33, the equations of motion associated with the generalized coordinates are given by:

$$\tau_i = \frac{d}{dt}(\frac{\delta k}{\delta\dot{\theta}_i}) - \frac{\delta k}{\delta\theta_i} + \frac{\delta U}{\delta\theta_i} \tag{5.38}$$

where $\tau_i$ is the driving torque developed by the actuators of joint i.

The resulting equations of motion can be presented in matrix form as follow:

$$\tau = H(\theta)\ddot{\theta} + h(\theta,\dot{\theta}) + g(\theta) \tag{5.39}$$

where $H(\theta)$ is $3\times3$ inertia matrix representing the inertia moments of the actuator and the mechanical structure of the manipulator. The elements of the matrix are:

$$H_{11} = I_{1y} + I_{2x}s_2^2 + I_{2y}c_2^2 + I_{3x}s_{23}^2 + I_{3y}s_{23}^2 \quad (5.40)$$
$$I_{3z}c_{23}^2 + m_2l_2^2c_2^2 + m_3((l_3s_{23} + a_2c_2)^2$$

$$H_{12} = 0 \quad (5.41)$$

$$H_{13} = 0 \quad (5.42)$$

$$H_{21} = 0 \quad (5.43)$$

$$H_{22} = I_{3z} + I_{3y} + m_2l_2^2 + m_3(a_2^2 + l_3^2 + 2l_3a_2s_3) \quad (5.44)$$

$$H_{23} = m_3(l_3^2 + a_2l_3s_3) + I_{3y} \quad (5.45)$$

$$H_{31} = 0 \quad (5.46)$$

$$H_{32} = m_3(l_3^2 + a_2l_3s_3) + I_{3y} \quad (5.47)$$

$$H_{33} = m_3l_3^2 + I_{3y} \quad (5.48)$$

$H(\theta, \dot{\theta})$ is a $1 \times 3$ vector representing Corriolis and centrifugal forces. The elements of this vector are:

$$h_1 = [-m_2 l_2^2 s_2 c_2 + 2m_3(l_3 s_{23} + a_2 c_2)(l_3 c_{23} - a_2 s_2) \tag{5.49}$$
$$+(I_{2x} - I_{2y})s_2 c_2 + (I_{3x} + I_{3y} - I_{3z})s_{23} c_{23}]\dot{\theta}_1 \dot{\theta}_2$$
$$+ [2m_3(l_3 s_{23} + a_2 c_2) l_3 c_{23} + (I_{3x} + I_{3y} + I_{3z} s_{23} c_{23}]\, \dot{\theta}_1 \dot{\theta}_3$$

$$h_2 = [-m_2 l_2^2 s_2 c_2 + m_3(l_3 s_{23} + a_2 c_2)(l_3 c_{23} - a_2 s_2) \tag{5.50}$$
$$+(I_{2x} - I_{2y})s_2 c_2 + (I_{3x} + I_{3y} - I_{3z})s_{23} c_{23}]\dot{\theta}_1^2$$
$$+m_3 a_2 l_3 c_3 \dot{\theta}_3^2 + 2m_3 a_2 l_3 c_3 \dot{\theta}_1 \dot{\theta}_3$$

$$h_3 = [m_3 l_3 a_2 c_3 + m_3(l_3 s_{23} + a_2 c_2)(l_3 c_{23} + \tag{5.51}$$
$$+(I_{3x} - I_{3z})s_{23} c_{23}]\dot{\theta}_1^2 + +m_3 a_2 l_3 c_3 \dot{\theta}_2^2$$

and $g(\theta)$ is a $1 \times 3$ vector representing the effect of gravity on each link. The elements of this vector are:

$$g_1 = 0 \tag{5.52}$$

$$g_2 = m_2 l_2 c_2 g + m_3 g(a_2 c_2 + l_3 c_{23}) \tag{5.53}$$

$$g_3 = m_3 g l_3 c_{23} \tag{5.54}$$

## 5.4 Path design of robot manipulator

When planning a specific task an approach that may be used to generate a robot end effector trajectory is to compute the necessary torque to drive the joint along the desired path. Several techniques to compute this torque are available (Paul

1972; Markiewicz 1973). A block diagram for a computed torque technique is shown in Figure 5.3.

Most of the computed torque techniques do not take into consideration the capability of the joint actuator to deliver torques in order to minimize the time of the task. In this section a technique that can be used to minimize the time of travel along path taking into account the maximum torques, is presented.

### 5.4.1 Mathematic modeling of the path

To minimize the time of travel between two visiting points the path of the end effector must have a minimum curvature. The simplest such path is a straight line. To travel along a straight line and still pass by the visiting points the end effector must stop at these points, and as a result the execution time increases. In this thesis the approach used is to use a linear path followed by a curvature to reorient the end effector to the next linear part if the three points are not linear, as shown in the Figure 5.4. There are several types of curvatures that can be fit between the two linear parts. An arc of circle is the simplest and is used in this study. The minimum radius of the circle that can be fit is computed by the following equation which resulting from Equation 2.14:

$$\rho(t) = \frac{1}{\kappa(t)} = \frac{|\dot{r}|^3}{|\dot{r} \times \ddot{r}|} \tag{5.55}$$

To have a minimum radius of curvature, the numerator has to be minimum .This can be acheived by reducing the velocity as the tip travels along the curved part. Alternatively, the denominator of Equation 5.55 can be increased. This can be

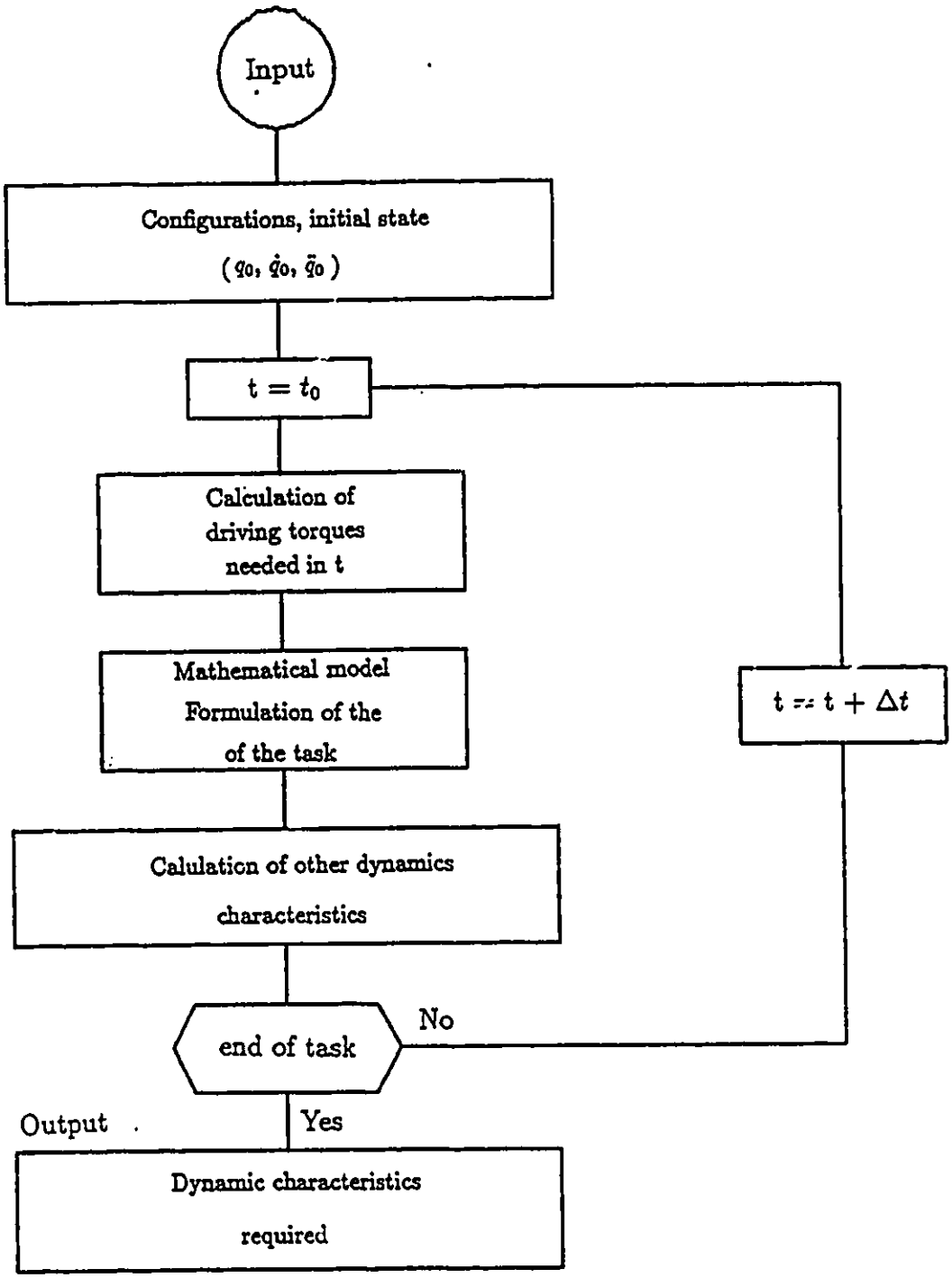



Figure 5.3: Block scheme of the algorithm for specified task

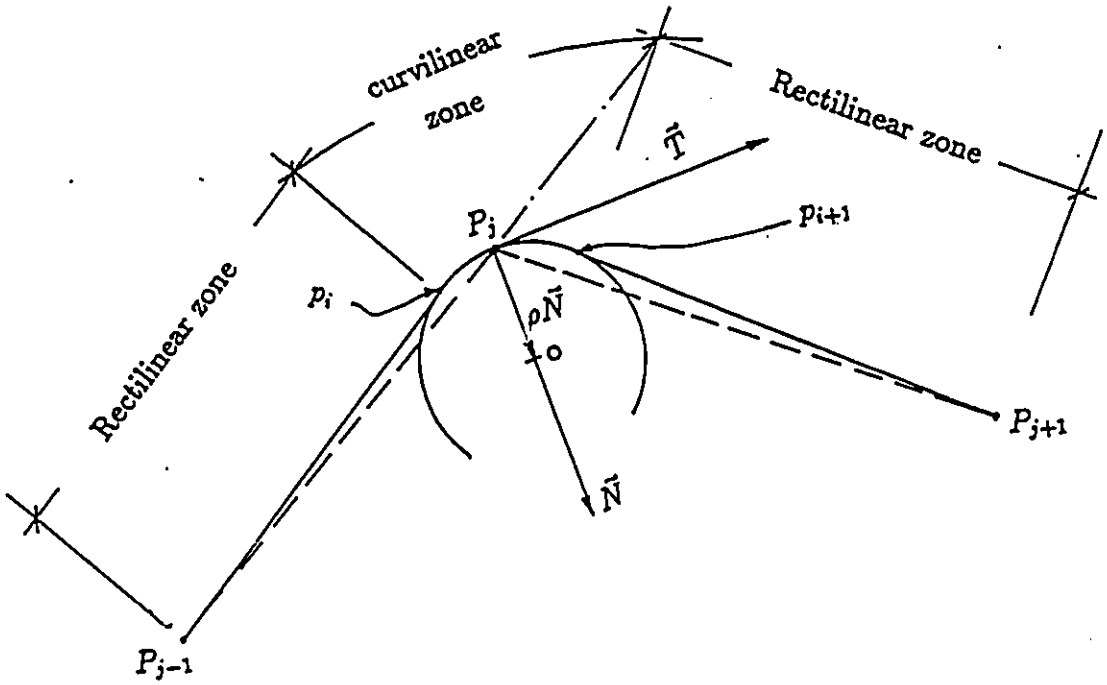



Figure 5.4: Decomposition of path for given 3 visiting points

achieved by increasing the acceleration $\ddot{r}$. To maximumize the travelling speed. the objective function (radius of curvature) is subjected to some constraints on the acceleration. From the previous section one can impose the different torque equations as a constraints.

To find the orientation of the arc of circle, the velocity direction of the tip of the end effector is given by the tangential vector resulting from the sum of the two slopes of the straight lines joining the three visiting points, see Figure 5.4. The unit tangent vector is given by:

$$\vec{T} = \frac{\vec{P_{j-1}P_j} + \vec{P_jP_{j+1}}}{\mid \vec{P_{j-1}P_j} + \vec{P_jP_{j+1}} \mid} \tag{5.56}$$

The velocity at the tip has the following direction:

$$\vec{T} = \frac{\dot{\vec{r}}}{\mid \dot{r} \mid} = \frac{\dot{x}\vec{i} + \dot{y}\vec{j} + \dot{z}\vec{k}}{\sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2}} \tag{5.57}$$

The above Equations 5.56 and 5.57 must mach in all directions ($\vec{i}$, $\vec{j}$ and $\vec{k}$).

In the case where it is desired to move the end effector along the curved part of the path with a constant velocity, an additional constraint has to established as follows:

$$s(t) = \int_{t_0}^{t} \sqrt{(\frac{dx}{dt})^2 + (\frac{dy}{dt})^2(\frac{dz}{dt})^2}dt \tag{5.58}$$

where at $t_0$ is the time of starting of the motion along the curved part. Differentiating Equation 5.58 with respect to time as follows:

$$\frac{ds}{dt} = \sqrt{(\frac{dx}{dt})^2 + (\frac{dy}{dt})^2(\frac{dz}{dt})^2} \tag{5.59}$$

Differentiating one more time to get the acceleration along the curvature, gives:

$$\frac{d^2s}{dt^2} = \frac{\frac{dx}{dt}\frac{d^2x}{dt^2} + \frac{dy}{dt}\frac{d^2y}{dt^2} + \frac{dz}{dt}\frac{d^2z}{dt^2}}{\sqrt{(\frac{dx}{dt})^2 + (\frac{dy}{dt})^2(\frac{dz}{dt})^2}} \tag{5.60}$$

Since the end effector moves with constant velocity along the curvature, $d^2s/dt^2$ has to be equal to zero, which results in the following equation:

$$\frac{dx}{dt}\frac{d^2x}{dt^2} + \frac{dy}{dt}\frac{d^2y}{dt^2} + \frac{dz}{dt}\frac{d^2z}{dt^2} = 0 \tag{5.61}$$

using vector notation $\dot{r}$ and $\ddot{r}$ from chapter 2 one can write:

$$\dot{r}.\ddot{r} = 0 \tag{5.62}$$

This means that the tangential component of $\ddot{r}$ is equal to zero.

Finally arranging the above constraints, the problem becomes an optimization of the radius of curvature subject to some physical constraints and can be stated as follows:

*Minimize* $\rho(\dot{\theta}_i, \ddot{\theta}_i)$; *i=1,2,3.*

$(\dot{\theta}_i, \ddot{\theta}_i) \in R^2$

Subject to:

*Torques constraints*

$$H(\theta_i)\ddot{\theta}_i + h(\theta_i, \dot{\theta}_i) + g(\theta_i) \leq \tau_i$$

*Velocity direction constraints*

$$\dot{x}^2 - T_x^2(\dot{x}^2 + \dot{y}^2 + \dot{z}^2) = 0$$

$$\dot{y}^2 - T_y^2(\dot{x}^2 + \dot{y}^2 + \dot{z}^2) = 0$$

$$\dot{z}^2 - T_z^2(\dot{x}^2 + \dot{y}^2 + \dot{z}^2) = 0$$

*and assume constant velocity*

$$\dot{x}\ddot{x} + \dot{y}\ddot{y} + \dot{z}\ddot{z} = 0$$

### 5.4.2 Cartesian interpolated subpath

Once the radius of curvature is found from the last section, one can proceed to state the equation of the curved part and compute the points of intersection of the circular section with the linear part.

With reference to Figure 5.4 the intersection points are given by the points that result from the tangent lines to the circle passing through a knot point, and starting at the two adjacent knot points.

The center of the circle is given by:

$$\vec{P_jO} = \rho\vec{N} \Longrightarrow O(o_x, o_y, o_z) \tag{5.63}$$

Where:

$$\vec{N} = \frac{P_j\vec{P}_{j-1} + P_j\vec{P}_{j+1}}{\mid P_j\vec{P}_{j-1} + P_j\vec{P}_{j+1} \mid} \tag{5.64}$$

Since the circle lies on the plane that passes through the three visiting points, one has first to write the equation of the plane as follows:

$$\mathcal{P}: \; ax + by + cz + d = 0 \tag{5.65}$$

where

$$(a, b, c)^T = P_j\vec{P}_{j-1} \times P_j\vec{P}_{j+1} \tag{5.66}$$

Now the point $P_l(x, y, z)$ of the intersection is given by solving the following system equations:

$$\begin{cases} (x - o_x)^2 + (y - O_y)^2 + (z - O_z)^2 = \rho^2 \\ ax + by + cz + d = 0 \\ (x - x_{j-1})^2 + (y - y_{j-1})^2 + (z - z_{j-1})^2 = \mid \vec{P_{j-1}O} \mid^2 - \rho^2 \end{cases} \tag{5.67}$$

Similarly for point $P_{l+1}$, with $P_{j+1}$ used instead of $P_{j-1}$.

Finally the curvature equation $\mathcal{C}$ is given by:

$$\mathcal{C}: \begin{cases} (x - o_x)^2 + (y - O_y)^2 + (z - O_z)^2 = \rho^2 \\ ax + by + cz + d = 0 \end{cases} \tag{5.68}$$

# Chapter 6

ROBOT MOTION SIMULATION

## 6.1 Introduction

The method proposed in this study is used to plan a trajectory and simulate numerically and graphically a typical task. The manipulator used to perform the task is a serial drive with three degrees of freedom as shown in Figure 5. The links characteristics are given in appendix B.

The results of the trajectory obtained from the imposed acceleration function are compared to those of a trajectory composed of a 4-1-4 polynomial function, Paul[42]. The polynomial function is chosen for several reason, its simplicity to implement, its mathematic modeling is comparable to the proposed algorithm and its extensivly used for industrial robot tasks such as spray and arc welding along contours. The results of these algorithms are presented in section 6.3.

The last section deals with results of the algorithm when generating a quasi-optimum path in Cartesian space.

Table 6.1: The values of the via points

| Point # | $P_x$ (m) | $P_y$ (m) | $P_z$ (m) | Vel. (m/s) | Acc. ($m/s^2$) |
|---|---|---|---|---|---|
| 1 | 1.02 | .59 | .97 | 0. | 0. |
| 2 | .46 | .55 | 1.44 | - | - |
| 3 | .45 | .38 | 1.45 | 0. | 0. |

## 6.2 Task to be performed

Generally, manipulators are attached a to fixed base at one end and carry a terminal device or tool at the other. The time-varying actuator commands are intended to cause this terminal device to follow a given trajectory through space. The trajectory in either joint space or Cartesian space is generated based on the concept of defining visiting points which the robot arm has to pass by along its path.

In this present simulation, the task to be executed is portrayed graphically in Figure 6.1. The end of the hand starts from the source $P_1$ with an initial velocity and acceleration equal to zero. It passes through point $P_2$ so that it does not collide with the static obstacle, and finally reaches the destination point $P_3$ with zero velocity and acceleration.

The points are defined as given in Table 6.1 in Cartesian space. The task is simulated by a Fortran program. The results are presented in the following sections.

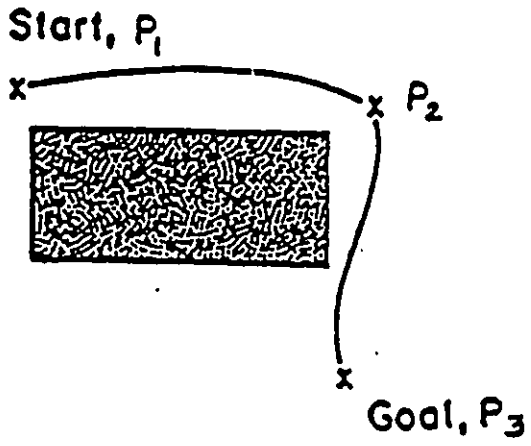



Figure 6.1: The use of via points to prevent the arm colliding with object in the work space

## 6.3 Simulation results

### 6.3.1 Trajectory control at joint level

Trajectory control by means of polynomial function

Without loss of generality the discussion will be limited to a single variable, the generalized position q. For each pair of consecutive positions a tri-polynomial 4-1-4 is used Paul [42]. Across the boundary between successive segments, continuity is imposed upon position, velocity and acceleration. The resulting polynomial function describing the trajectory is given by:

$$q = a_4 t^4 + a_3 t^3 + a_2 t^2 + a_1 t + a_0 \tag{6.1}$$

The derivatives of this function, the velocity and acceleration are given by:

$$\dot{q} = 4a_4 t^3 + 3a_3 t^2 + 2a_2 t + a_1 \tag{6.2}$$

$$\ddot{q} = 12a_4 t^2 + 6a_3 t + 2 \tag{6.3}$$

and the third derivative, the "jerk", is given by:

$$\dddot{q} = 24a_4 t + 6a_3 \tag{6.4}$$

With reference to Figure 4.4, and applying the boundary conditions, the functions specifying position, velocity, acceleration and jerk are given respectively by Paul [42] as follow:

$$q(t) = \left[(\Delta C \frac{t_{acc}}{T_1} + \Delta B)(2 - h)h^2 - 2\Delta B\right] h + B + \Delta B \tag{6.5}$$

Table 6.2: Practical values for $t_{acc}$, $\omega_{max}$ and $\alpha_{max}$

| $t_{acc}$ | $\omega_{i,max}$ | $\alpha_{i,max}$ |
|---|---|---|
| .5sec | 1. rad/s | 1. rad/$s^2$ |

$$\dot{q}(t) = \left[(\Delta C \frac{t_{acc}}{T_1} + \Delta B)(1.5 - h)2h^2 - \Delta B\right] \frac{1}{t_{acc}} \tag{6.6}$$

$$\ddot{q}(t) = \left[(\Delta C \frac{t_{acc}}{T_1} + \Delta B)(1. - h)3h\right] \frac{1}{t_{acc}^2} \tag{6.7}$$

$$\dddot{q}(t) = \left[(\Delta C \frac{t_{acc}}{T_1} + \Delta B)(1. - 2h)3\right] \frac{1}{2t_{acc}^3} \tag{6.8}$$

In implementing the above equations for the algorithm proposed by Paul, one has to know the transition time "$t_{acc}$" and the maximum velocities of the joints. Generally these values are determined experimentally at the critical robot configurations.

An example have been worked out with this technique to execute the above task, given the values of $t_{acc}$ and $\omega_{max}$ (maximum angular velocity) stated in Table 6.2. The resulting profiles for the position, velocity, acceleration and jerk are displayed in Figures 6.2 to 6.5.

From the above examples one can conclude that as the time for the transition decreases, the value of the jerk increase dramatically, which results in the excitation of the dynamic structure of the robot manipulator. An additional constraint

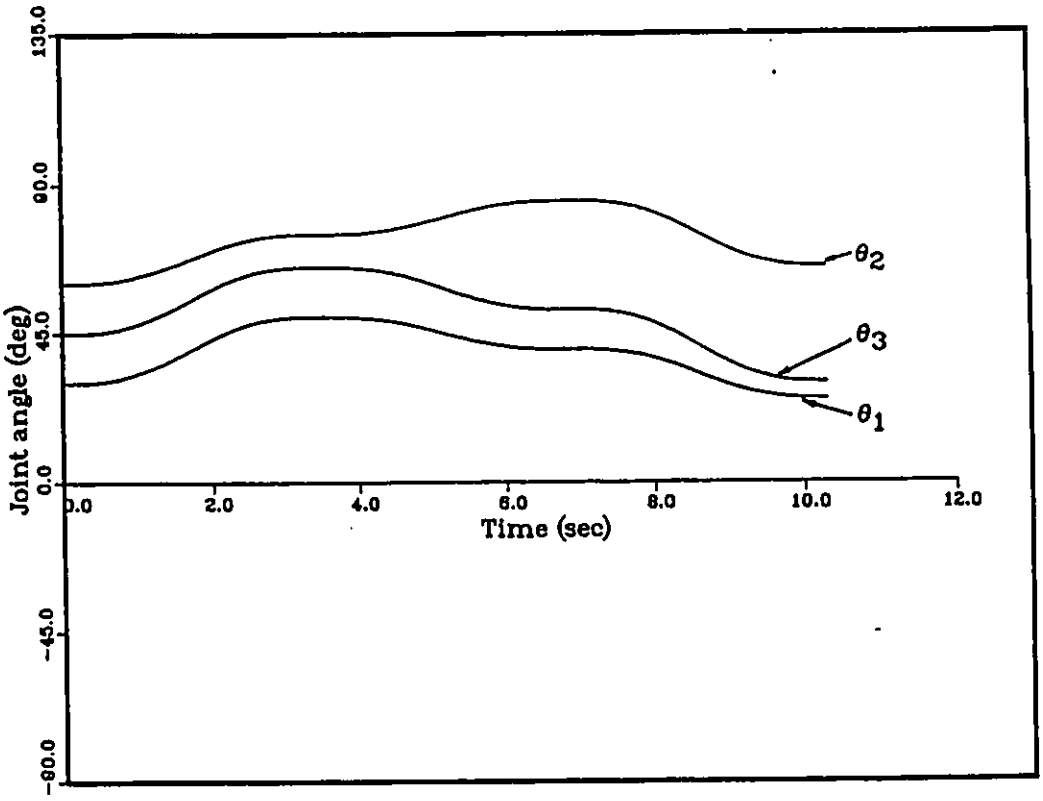



Figure 6.2: The positions by means of polynomial function for joint 1, 2 and 3 with $t_{acc}$=.5s

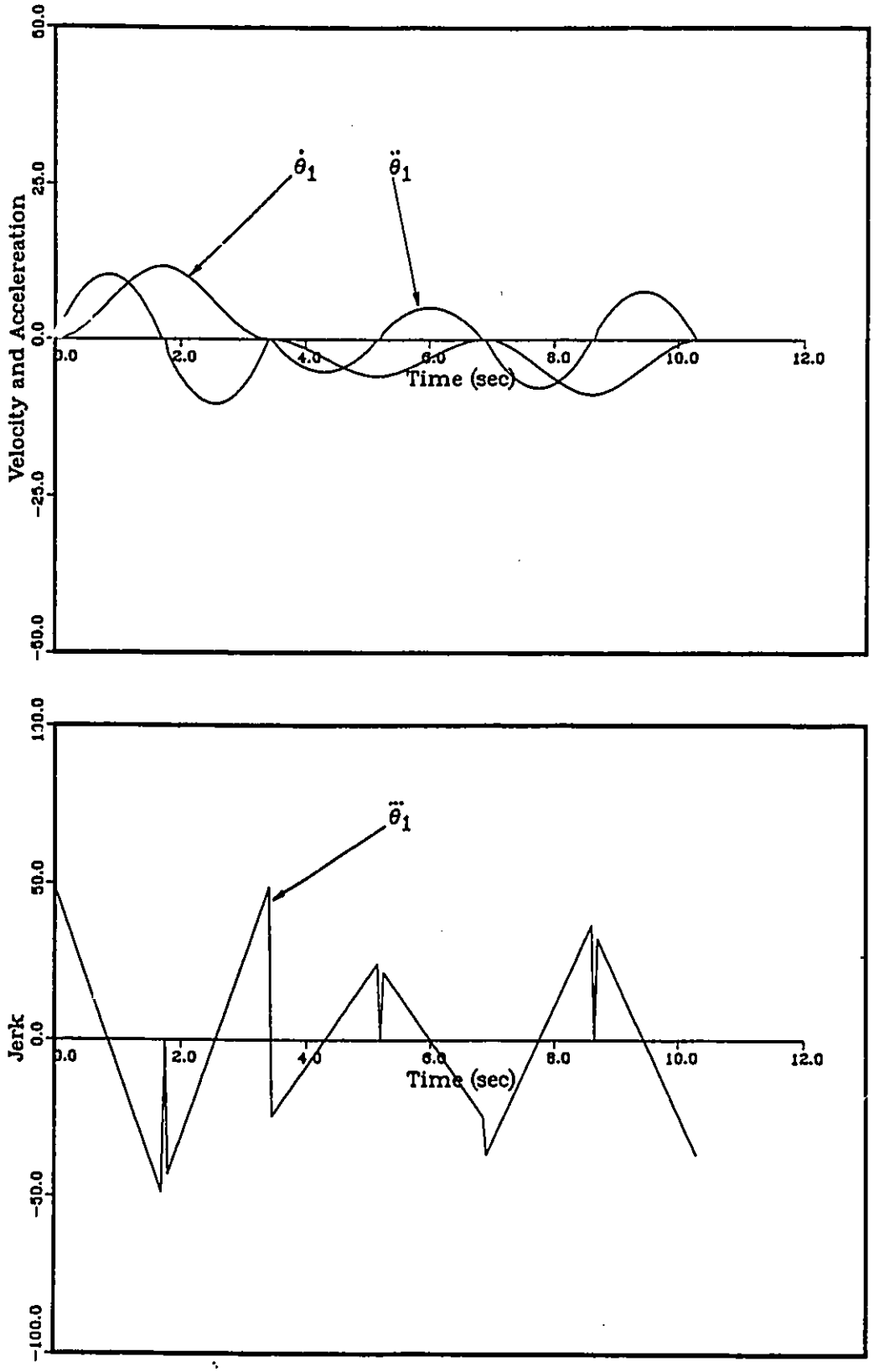



Figure 6.3: The velocity, acceleration and jerk by means of polynomial function of joint 1 with $t_{acc}$=.5s

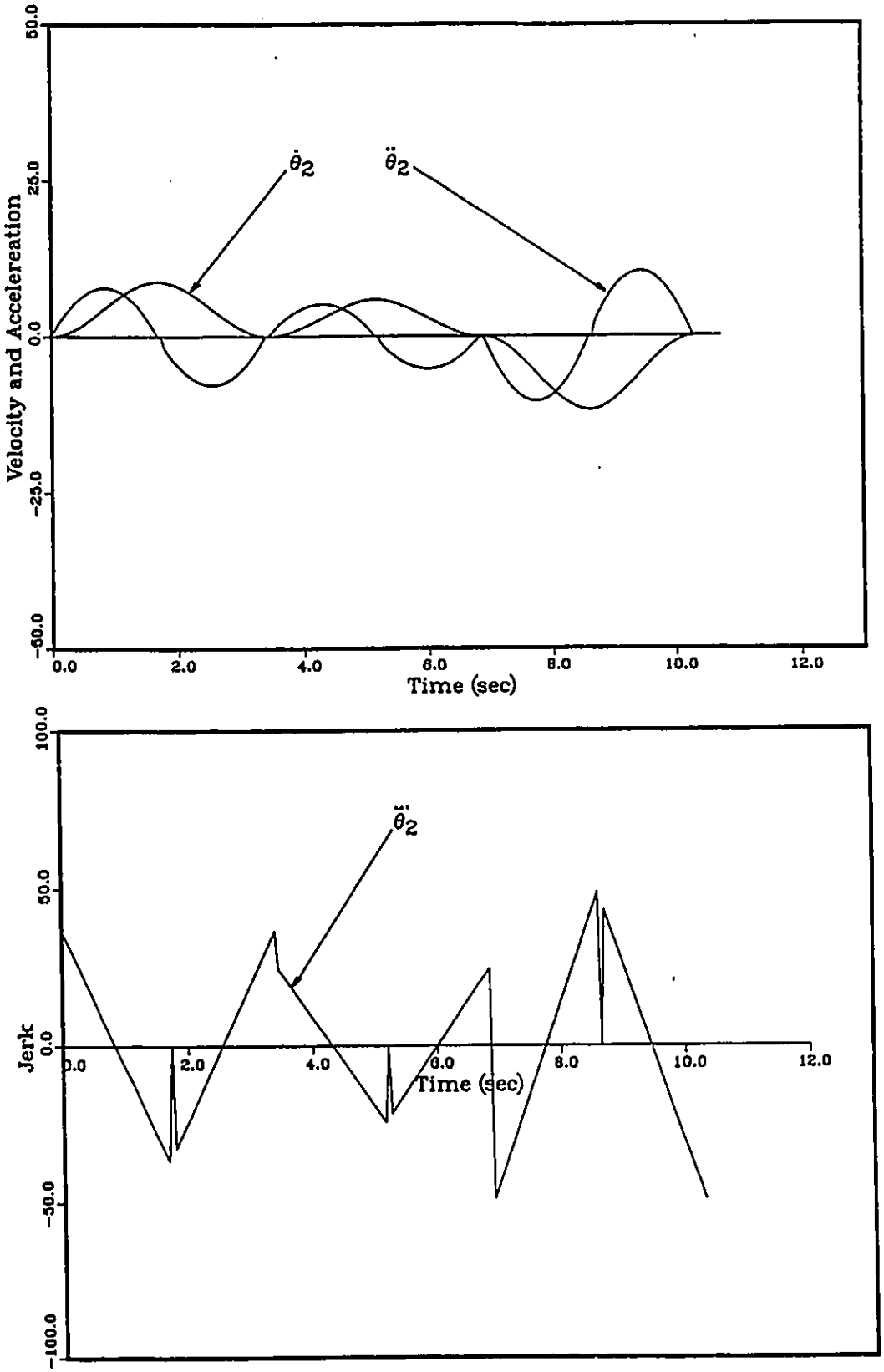



Figure 6.4: The velocity, acceleration and jerk by means of polynomial function for joint 2 with $t_{acc}$=.5s

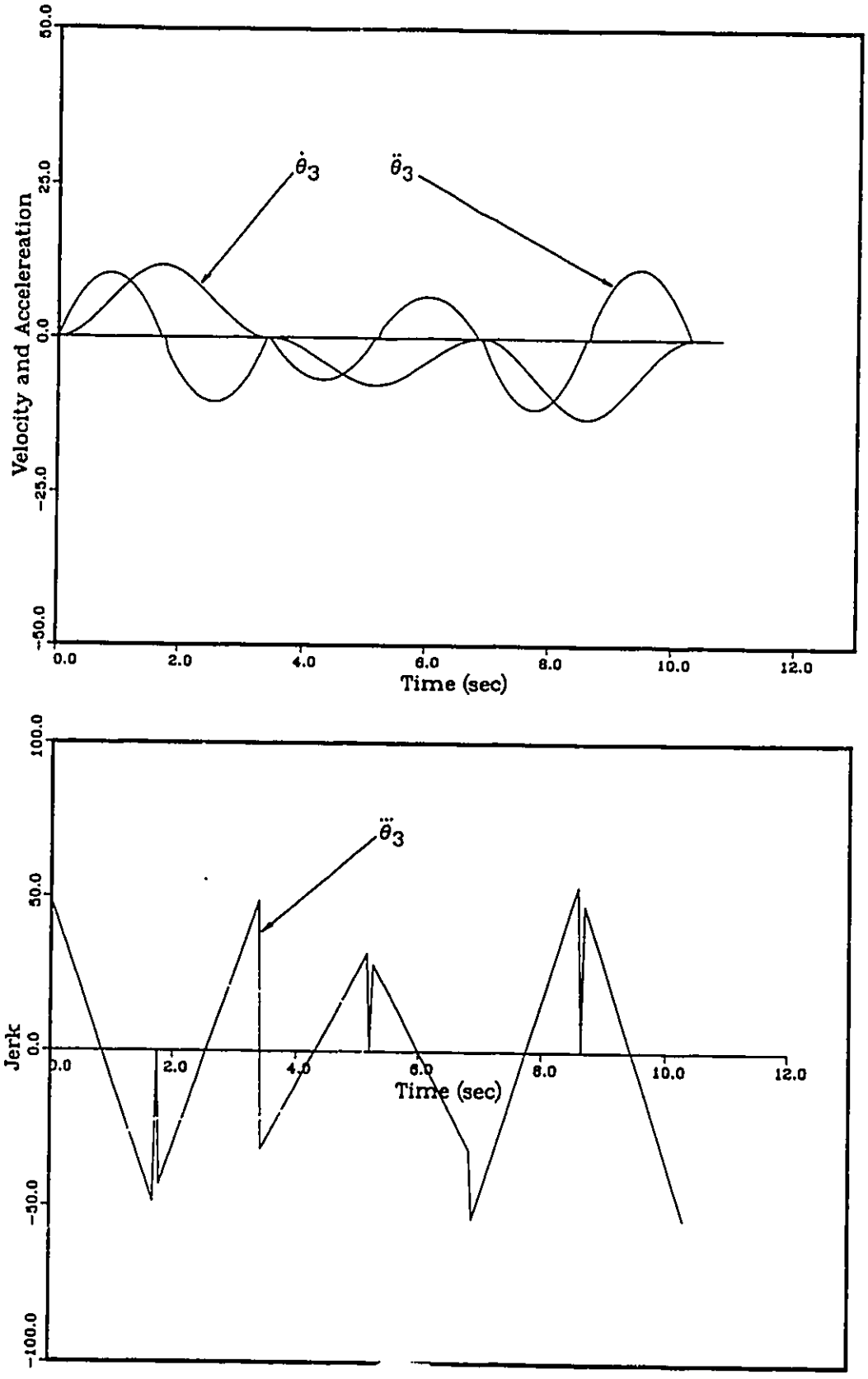



Figure 6.5: The velocity, acceleration and jerk by means of polynomial function for joint 3 with $t_{acc}$=.5s

in the trajectory leads to a recomputation of all the coefficients of the polynomial function governing the trajectory. The results become very impredictable, especially if the trajectory is required to be continuous up to the third derivative. This results in a polynomial of degree seven. As the degree of the polynomial increases the trajectory has the tendency to under-shoot or over-shoot the desired position. Consequently the calculated trajectory can easily exceed the joint capabilities.

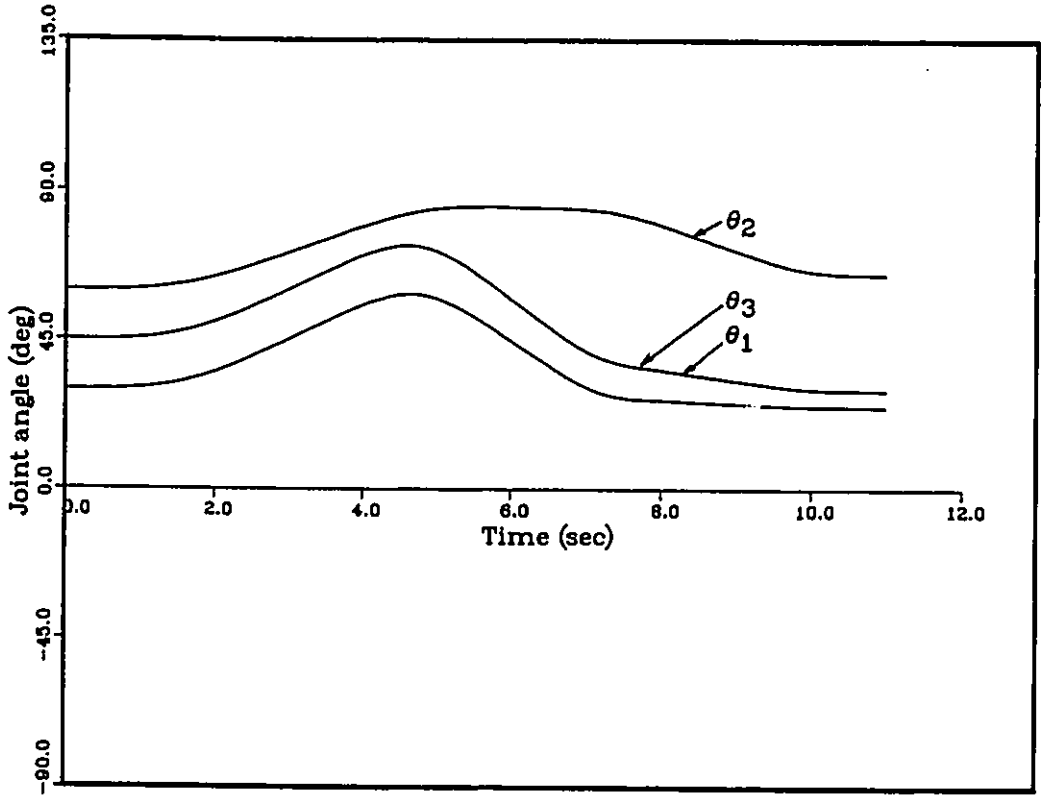



Figure 6.6: The positions by means of imposed acceleration profile for joint 1, 2 and 3.

## Trajectory control by means of imposed acceleration profile

The task portrayed in Figure 6.1 is planned using the proposed algorithm with the same values of $\omega_{i,max}$ and $\alpha_{i,max}$ stated in Table 6.2.

The resulting profiles for the position, velocity, acceleration and jerk are displayed in Figures 6.6 to 6.9.

Figures 6.6 to 6.9 show that the proposed trajectory planning method ensures the continuity between successive segment up to jerk. This in contrast to the tri-polynomial method as evident from Figures 6.2 to 6.5. With reference to Figures

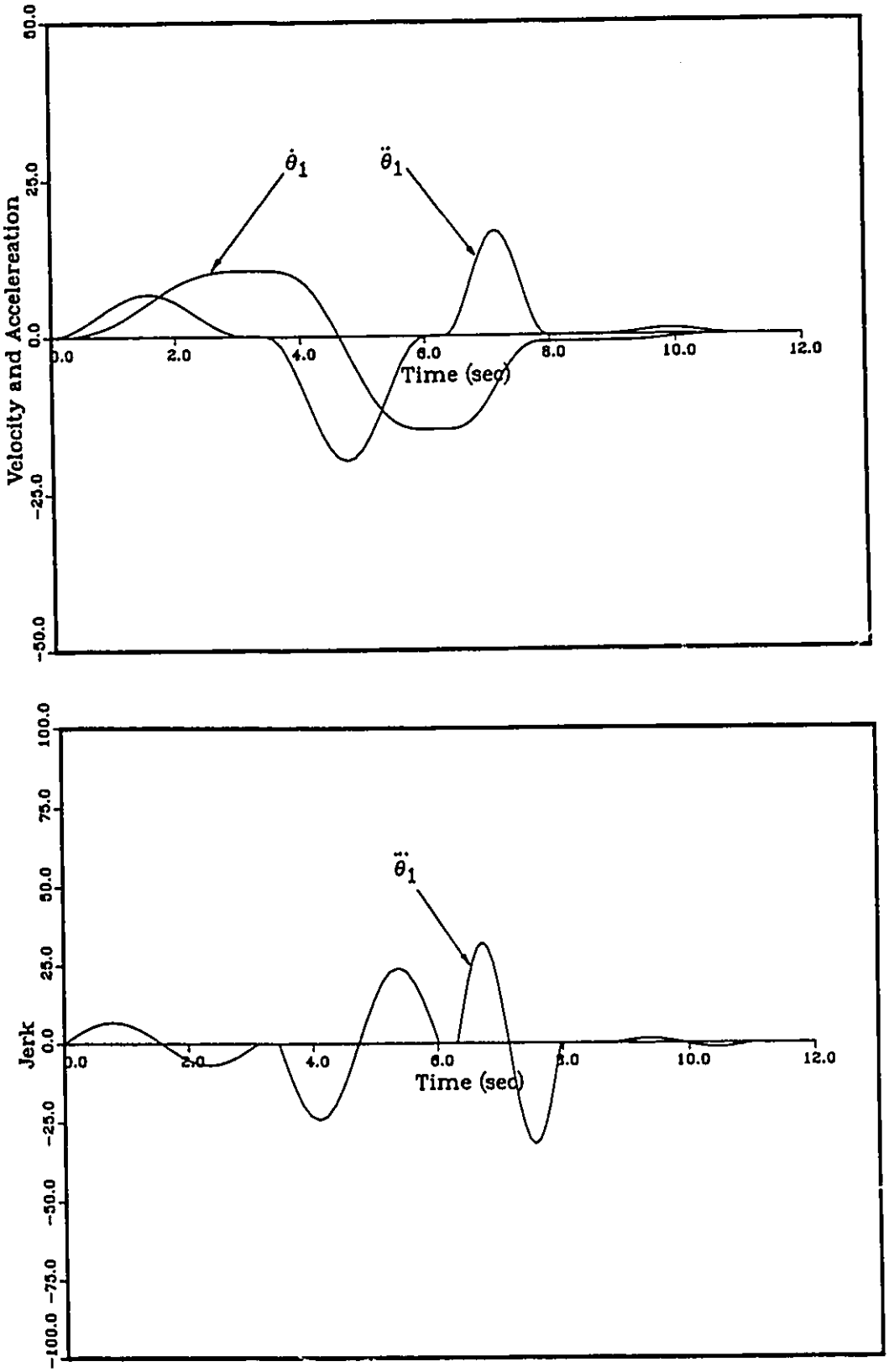



Figure 6.7: The velocity, acceleration and jerk by means of imposed acceleration profile for joint 1.

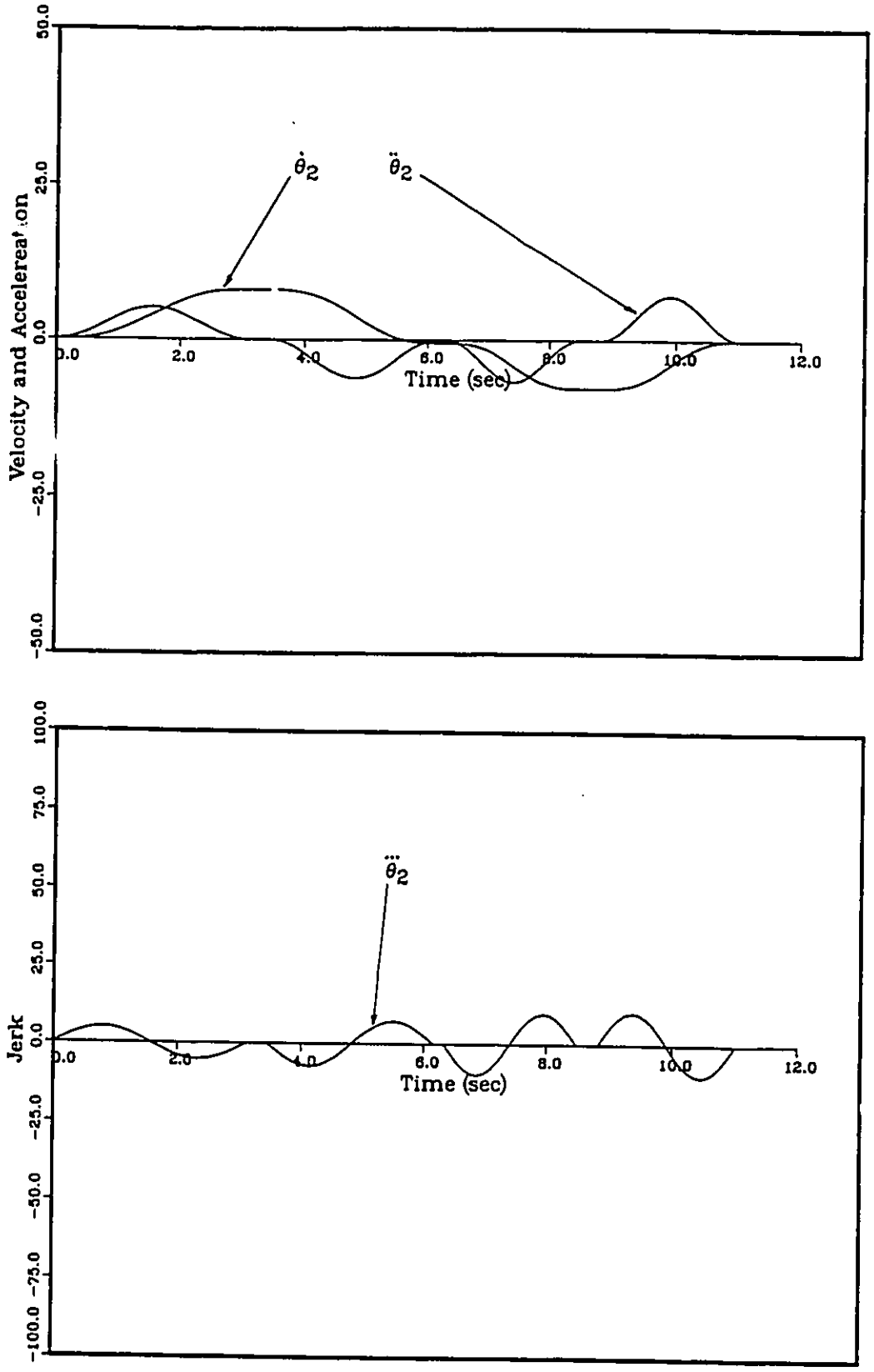



Figure 6.8: The velocity, acceleration and jerk by means of imposed acceleration profile for joint 2.

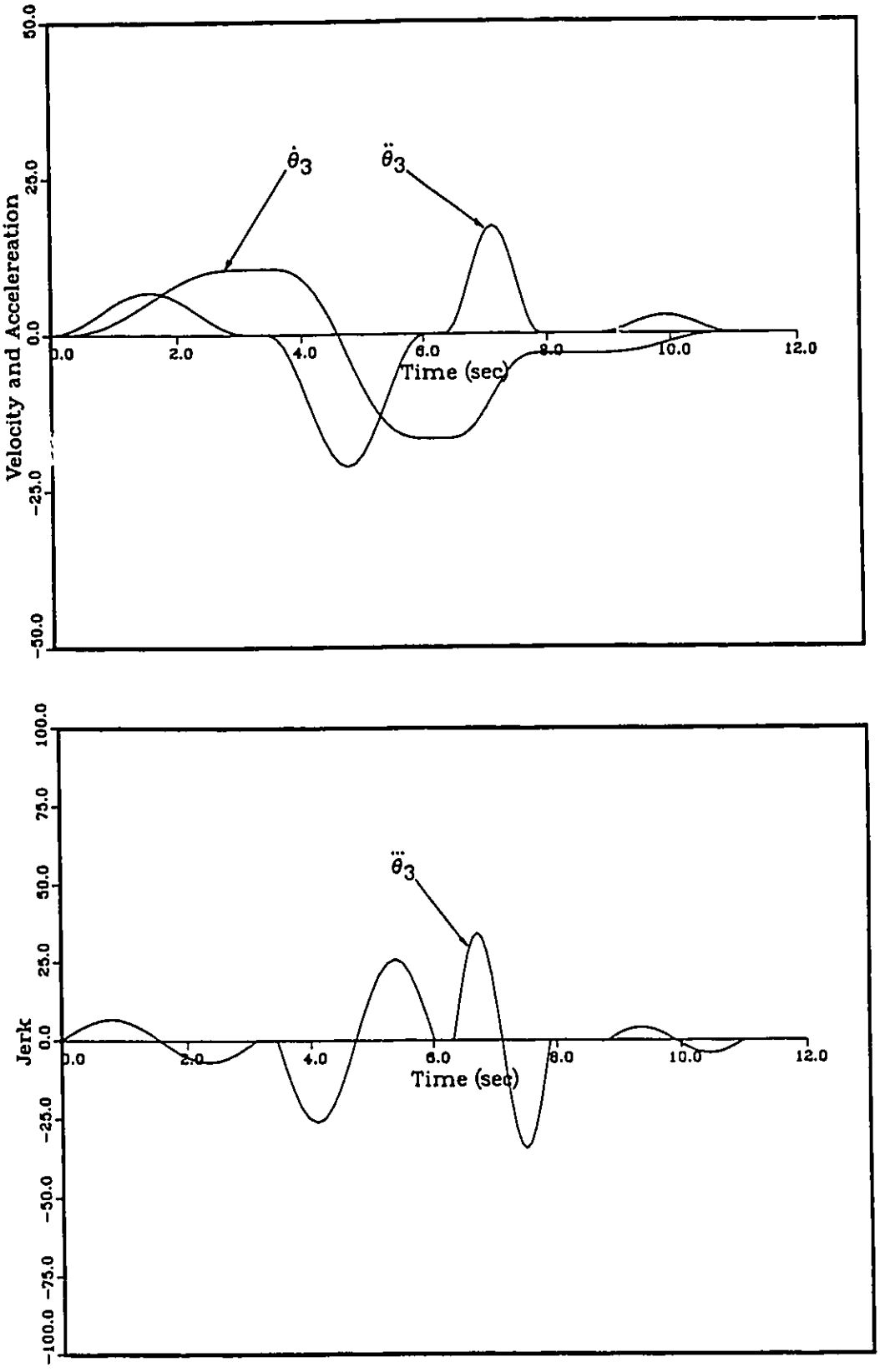



Figure 6.9: The velocity, acceleration and jerk by means of imposed acceleration profile for joint 3.

6.7 to 6.9 the values of the jerk at the beginning and the end of the transition segment are minimum (in this case equal to zero). This leads to a smooth motion with minimal dynamic excitation of the robot structure.

The proposed acceleration profile can cater for different starting and ending values for the acceleration. The trajectory equation for a polynomial interpolation would be of fifth order polynomial if the starting differs from the ending acceleratio-

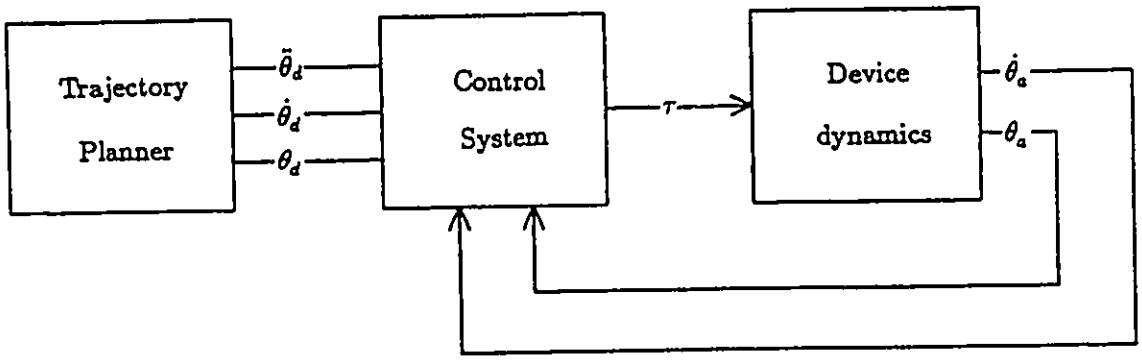



Figure 6.10: Block diagram for manipulator system

## 6.4 Path design in cartesian space

Generally the robot arm is required to move along a specified trajectory in Cartesian space. The trajectory is preplanned off-line where the position, velocity and acceleration of each joint are specified at each time. The robot controller coordinates the move of all the joints based on these informations. Figure 6.10 shows the block diagram of a manipulator system controller.

The informations fed to the controller ($\ddot{\theta}_d$, $\dot{\theta}_d$ and $\theta_d$) from the trajectory planner are generally very conservative so as to ensure the feasibility of the task. This means that the joint actuators are underutilized and the trajectory is far from being optimal.

The proposed method developed in Chapter 5 attempts to remedy this drawback. This method consists of preplanning the trajectory, taking into consideration the

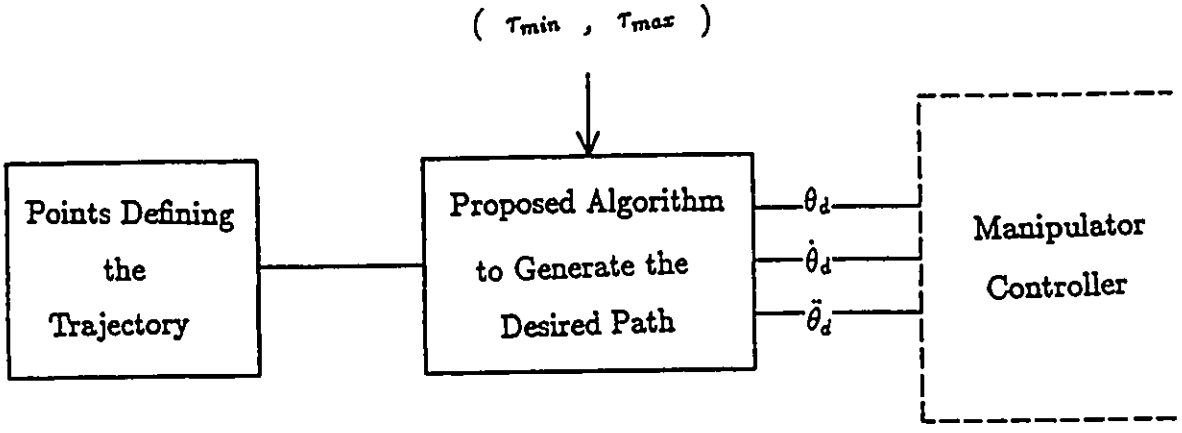



Figure 6.11: Modified block diagram for manipulator system

maximum available torques from the joint actuators. Also the designed path is minimized in terms of curvature so that the resulting path does not deviate from a straight line. The resulting block diagram is as shown in Figure 6.11.

### 6.4.1 Simulation

The end effector is asked to move from a source point to a destination point via a visiting point without stopping at the via point. The designed path is as prescribed in the last chapter. It is composed of two linear parts and a curved part. The curve is an arc of a circle having a minimum radius and being fitted between the two linear parts. The sesulting path does not violate the maximum capability of the joint actuators.

Table 6.3: The values of the visiting points

| Point # | $P_x$ (m) | $P_y$ (m) | $P_z$ (m) |
|---|---|---|---|
| 1 | .2 | .8 | .5 |
| 2 | .5 | 1.0 | 1.0 |
| 3 | .8 | 1.2 | .6 |

Figure 6.12 and 6.13 show the resulting designed path for the three visiting points given in Table 6.3.

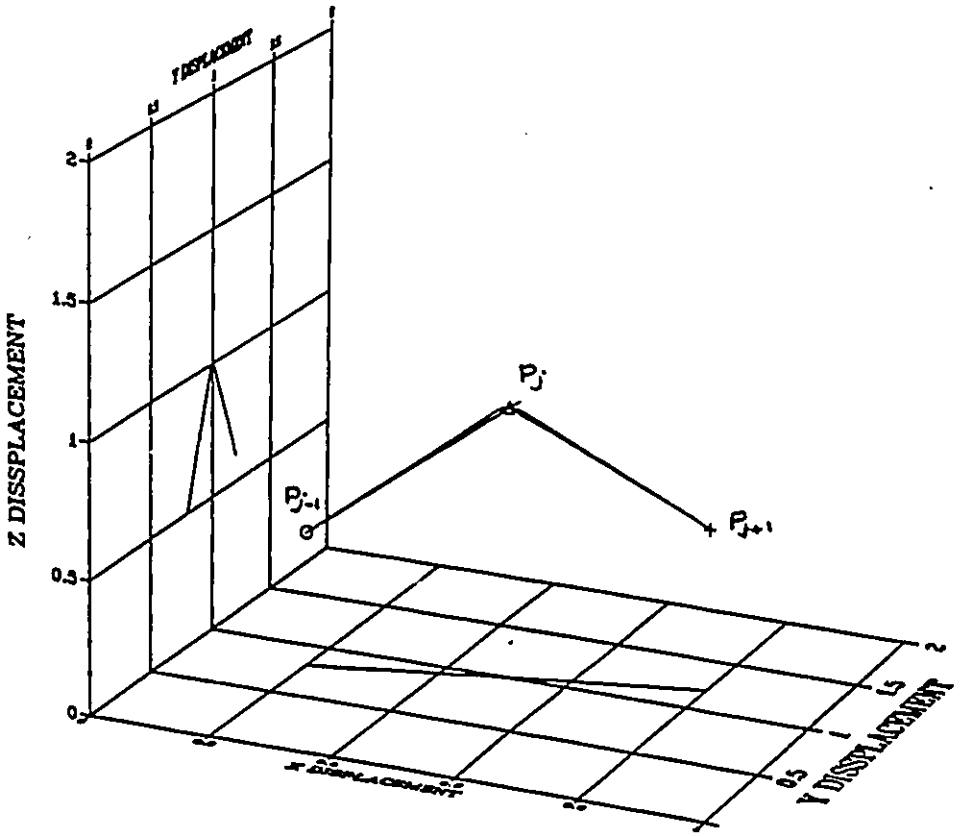



Figure 6.12: 3-D diagram for the resulting path.

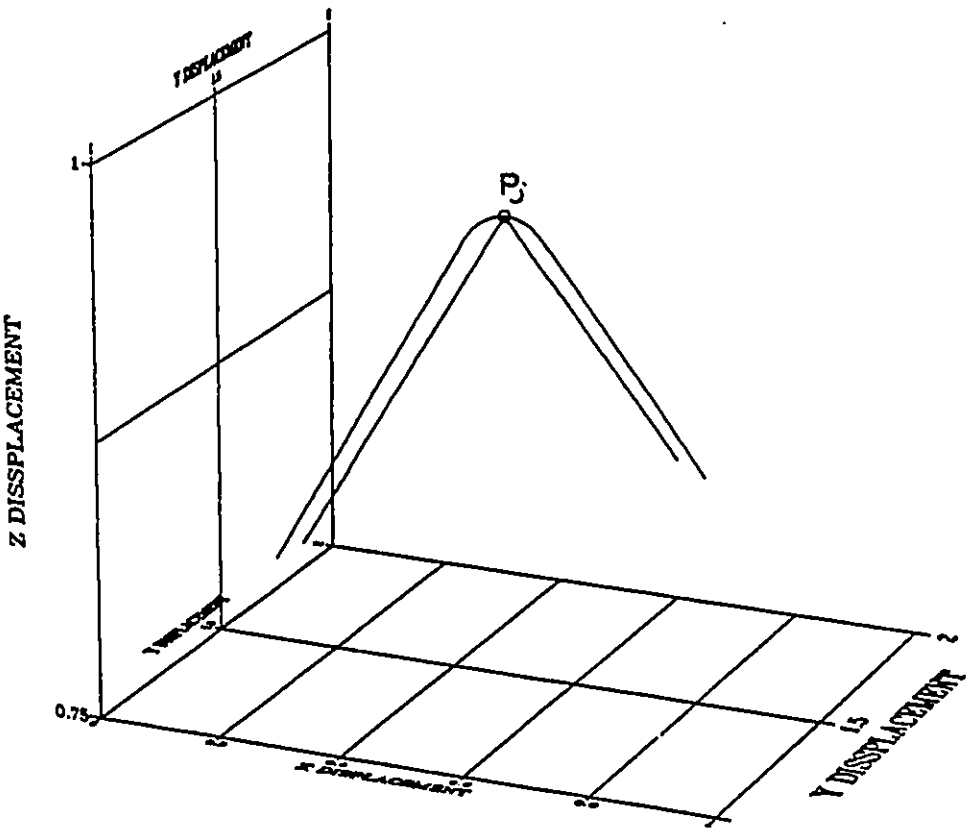



Figure 6.13: Magnification of the curvature part.

## 6.5 Discussion

The trajectory generation by means of the proposed acceleration profile offers the following advantages over the polynomial based method.

a) The resulting trajectory is computationally more efficient since for a polynomial, to satisfy the continuity up to the jerk one has to compute eight coefficients for a given transition segment as given in Appendix C. The proposed trajectory by means of imposed acceleration profile results in less coefficients to compute (given in Chapter 4) then the trajectory obtained by means of polynomial functions.

b) The proposed algorithm generates a trajectory which passes through all points along the path with minimum deviation from straight line paths between these points.

c) The end effector does not need to stop at the points along the trajectory to change direction. Change in direction via points is accomplished by minimum radius of curvature with maximum torque constraints.

d) generated trajectory is near time optimal.

# Chapter 7

## CONCLUSIONS AND RECOMMENDATIONS

This thesis deals with the trajectory control and generation for given robot manipulator tasks. In the first part of the work an efficient and accurate method for interpolation along the robot trajectory is proposed. The method is based on an approximation of the Gauss function as a shape for the desired acceleration profile. The motion trajectory is composed of a combined second order and cosine function in time for the joint acceleration segment, followed by a first order polynomial for the constant velocity segment. The method allows a full control over the acceleration profile, and the continuity of derivatives up to "jerk" without additional computation overhead.

In comparison with the currently available methods such as the polynomial function, the developed technique is found to be computationally more efficient. The variation of the jerk is very smooth and moderated with respect to time and leads to fast motion without exciting the dynamic structure of the manipulator. The problem of robot control is very complex because of the nonlinearity and couplings in robot dynamics. Usually it is solved by a two-stage optimization, the first stage is called trajectory planning and the second stage is called trajectory tracking or

control. The trajectory tracker is responsible for making the robot's actual position, velocity and acceleration match the desired values provided to the tracker by trajectory planner. The informations fed to the tracker are not always feasible and executable specially in changing the directions and the velocities. The proposed algorithm is a novel method for controlling the different types of acceleration at the tip of the end effector. Also, the generated path takes into account the maximum available torque from the joint actuators which results in a near time-optimal trajectory.

The end effector need not stop at the points along the trajectory to change direction. Change in direction via the intermediate points is accomplished by a minimum radius of curvature with maximum torque constraints. The resulting designed path passes through all points along the path with minimum deviation from straight line path between these points.

This work deals only with a particular case of continuous motion control, but it can be extended to more general case of on-line point-to-point control motion trajectory using a dynamic approach.

# Bibliography


[1] ASADA, H. and J. E. SLOTINE. *"Robot analysis and Control"*, USA, John Wiley and Sons, 1986, 266 p.

[2] BENHARIB, B., et al. *"Optimal continuous path planning for Seven-Degrees-of-Freedom Robots"*, Journal of Engineering for Industry, vol. 108, august 1986, p. 213-218.

[3] BOBROW, J.E., et al. *"On The Optimal Control of Robotic Maninpulators with Actuator Constraints"*, Proceeding of American Control Conference, june 1983, p. 782-787.

[4] BRADY, M., et al. *"Basics of Robot Motion Planning and Control"*, Cambridge, Massachusetts, USA, MIT press.

[5] BRAIDANT V. and Geradin M. *" Optimum Path Planning of Robot Arms"*, Robotica, vol. 5, 1987, p. 323-331.

[6] CHAND, Sujeet, and Keith L. DOTY. *"On-line Polynomial Trajectories for Robot Manipulators"*, The International Journal of Robotics Research, vol. 4, no. 2, summer 1985, p. 38-48.

[7] DENAVIT, J., and HARTENBERG, R.S. *"A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices"*, Journal of Applied Mechanics, 1955, p. 215-221.

[8] DUBOWSKY, S., and D.T. DESFORGES. *"The application of Model- Referenced Adaptive Control to Robotic Manipulators"*, Journal of Dynamic Systems, Measurements, and Control (ASME), vol. 101, september 1979, p. 193-200.

[9] DUBOWSKY, S., and Z. SHILLER. *" Optimal Dynamic Trajectories for Robotic Manipulators"*, Theory and Practice of Robots and Manipulators, edition A. Morecki and G. Bianchi, Proceeding Romansy 1984, Cambridge, USA, MIT press, 1985, p. 133-144.

[10] FIEDRICH P. and RAINER .J. *"A Concept for Manipulator Trajectory Planning"*, International Journal of Robotics and Automation, vol. RA.3, no. 2, April 1987.

[11] FEATHERSTONE, R. *"Position and Velocity Transformations Between Robot End-Effector Coordinates and Joint Angles"*, The International Journal of Robotics Research, vol. 2, no. 2, summer 1983, p. 35-45.

[12] GEERING, Hans P., et al. *"Time-Optimal Motions of Robots in Assembly Tasks"*, IEEE Transactions on Automatic Control, vol. AC-31, no. 6, june 1986, p. 512-518.

[13] GRIEBEL U. and Higger R.A. *"Analizing Robot Motion Performance"*, International Journal of Robotics and Automation, vol. 1, no. 3, 1986, p. 71-76.

[14] GROOVER, P.M. *"Automation, Production Systems, and Computer-Integrated Manufacturing"*, USA, Prentice-Hall, 1987, 808 p.

[15] HANAFI, A., et al. *"Optimal Trajectory Control of Robotic Manipulators"*, Mechanism and Machine Theory, vol. 19, no. 2, 1984, p. 267-273.

[16] HOLLERBACH, Jhon M. *"A Recursive Lagrangian Formulation of Manipulator Dynamics and a Comparative study of Dynamics Formulation Complexity"*, IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-10, no. 11, november 1980, p. 730-736.

[17] HOLLERBACH, Jhon M. *"Dynamic Scaling of Manipulator Trajectories"*, Journal of Dynamic Systems, Measurement, and Control (ASME), vol. 106, march 1984, p. 102-106.

[18] HOLLERBACH, Jhon M. *"Wrist-Partitioned, Inverse Kinematic Accelerations and Manipulator Dynamics"*, The International Journal of Robotics Researsh, vol. 2, no. 4, winter 1983, p. 61-76.

[19] KAHN, M.E., and B. ROTH. *"The Near-Minimum-Time Control of Open-Loop Articulated Kinematic Chains"*, Journal of Dynamic Systems, Measurement, and Control (ASME), september 1971, p. 164-172.

[20] KANE, Thomas R., and David A. Levinson. *"The Use of Kane's Dynamical Equations in Robotics"*, The International Journal of Robotics Research, vol. 2, no. 3, fall 1983, p. 3-21.

[21] KANT, Kamal, and Steven W. ZUCKER. *"Toward Efficient Trajectory Planning: The Path-Velocity Decomposition"*, The International Journal of Robotics Reseach, vol. 5, no. 3, fall 1986, p. 72-89.

[22] KIM, Byung Kook, and Kang G. SHIN. *"Minimum-Time Path Planning for Robot Arms and Their Dynamics"*, IEEE Transactions on Systems , Man, and Cybernetics, vol. SMC-15, no. 2, march/april 1985, p. 213-223.

[23] KIM, Byung Kook, and Kang G. SHIN. *"Suboptimal Control of Industrial Manipulators with a Weighted Minimum Time-Fuel Criterion"*, IEEE Transactions on Automatic Control, vol. AC-30, no. 1, january 1985, p. 1-10.

[24] KOHLI, D., and J. SPANOS. *"Workspace Analysis of Mechanical Manipulators Using Polynomial Discriminants"*, Journal of Mechanisms, Transmissions, and Automation in Design, vol. 107, june 1985, p. 209-215.

[25] KRUT'KO, P.D., and YE.P. POPOV. *"Kinematic Algorithms for Manipulating Robot Movement Control"*, Engineering Cybernetic, vol. 17, no. 4, 1981, p. 65-75.

[26] KRUT'KO, P.D., and YE.P. POPOV. *"Motion Control of the Manipulating Robots Based on Second-Order Kinematic Algorithms"*, Engineering Cybernetic, vol. 19, no. 6, 1982, p. 89-97.

[27] LEE, C.S. George. *"Robot Arm Kinematics, Dynamics, and control"*, Computer, vol. 15, no. 12, december 1982, p. 62-80.

[28] LEE, C.S. George, et al. *"Tutorial on Robotics"*, s.l., Computer Society press (IEEE), s.d., 275 p.

[29] LEE, C.S. George, and M.J. CHUNG, *"An Adaptive Control Strategy for Mechanical Manipulators"*, IEEE Transactions on Automatic Control, p. 243-249.

[30] LIN, Chun-Shin, et al. *"Formulation and Optimazation of Cubic Polynomial Joint Trajectories for Industrial Robots"*, IEEE Transactions on Automatic Control, vol. AC-28, no. 12, december 1983, p. 1066-1073.

[31] LIN, Chun-Shin, and Po-Rong CHANG. *"Joint Trajectories of Mechanical Manipulators for Cartesian Path Approximation"*, IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-13, no. 6, november/december 1983, p. 1094-1102.

[32] LIN, Chun-Shin, and Po-Rong CHANG. *"Approximate Optimum Paths of Robot Manipulators Under Realistic Physical Constraints"*, Proceeding IEEE International Conference Robotics and Automation, march 1985, p. 737-742.

[33] LOZANO-PEREZ, Tomas. *"Spatial Planning: A Configuration Space Approach"*, IEEE Transactions on Computers, vol. C-32, no. 2, february 1983, p. 108-120.

[34] LUH, John Y.S. *"Resolved-Acceleration Control of Mechanical Manipulators"*, IEEE Transactions on Automatic Control, vol. AC-25, no. 3, june 1980, p. 468-474.

[35] LUH, John Y.S., et al. *"On-line Computational Scheme for Mechanical Manipulators"*, Journal of Dynamic Systems, Measurement, and Control (ASME), vol. 102, 1980, p. 69-76.

[36] LUH, John Y.S., and Yuan-Fang ZHENG. *"Computation of Input Generalized Forces for Robots with Closed Kinematic Chain Mechanisms"*, IEEE Journal of Robotics and Automation, vol. RA-1, no. 2, june 1985, p. 95-103.

[37] LUH, John Y.S., and C.S. LIN. *"Approximate Joint Trajectories For Control of Industrial Robots Along Cartesian Paths"*, IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-14, no. 3, may/june 1984, p. 444-450.

[38] LUH, John Y.S., and C.S. LIN. *"Optimum Path Planning for Mechanical Manipulators"*, Journal of Dynamic Systems, Measurement, and Control (ASME), vol. 102, june 1981, p. 142-151.

[39] PAUL, B. *"Analytical Dynamics of Mechanisms - A Computer Oreinted Overview"*, Mechanism and Machine Theory, vol. 10, 1975, p. 481-507.

[40] PAUL, Richard P. and B.E. SHIMANO. *"Compliance and Control"*, Proceeding of the Joint Automatic Control Conference, 1976, p. 694-699.

[41] PAUL, Richard P. *"Manipulator Cartesian Path Control"*, IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-9, no. 11, november 1979, p. 702-711.

[42] PAUL, Richard P. *"Robot Manipulators: mathematics, programming, and control"*, USA, MIT press, 1981, 279 p.

[43] PAUL, Richard P., et al *"Kinematic Control Equations for Simple Manipulators"*, IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-11, no. 6, june 1981, p. 449-455.

[44] PAUL, Richard P., et al *"Differential Kinematic Control Equations for Simple Manipulators"*, IEEE Transactions on Systems, Man, Cybernetics, vol. SMC-11, no. 6, june 1981, p. 456-460.

[45] PAUL, Richard P., and Charles N. Stevenson. *"Kinematics of Robot Wrists"*, The Internatinal of Robotics Research, vol. 2, no. 1, spring 1983, p. 31-38.

[46] RAIBERT, M.H., and J.J. CRAIG. *"Hybrid Position/Force Control of Manipulators"*, Journal of Dynamic Systems, Measurement, and Control (ASME), vol. 102, june 1981, p. 126-133.

[47] RAJAN, V.T. *"Planning of Minimum-Time Trajectories for Robot Arms"*, Proceeding IEEE International Conference Robotics and Automation, march 1985, p. 759-764.

[48] SAHAR, Gideon, and Jhon M. HOLLERBACH. *"Planning of Minimum-Time Trajectories for Robot Arms"*, Proceeding IEEE International Conference Robotics and Automatiion, march 1985, p. 751-758.

[49] SAHAR, Gideon, and Jhon M. HOLLERBACH. *"Planning of Minimum-Time Trajectories for Robot Arms"*, The International Journal of Robotics Research, vol. 5, no. 3, fall 1986.

[50] SCHEINMAN, V., and B. ROTH. *"On the Optimal Selection and Placement of Manipulators"*, Theory and Practice of Robotics and Manipulators, edition A. Morecki and G. Bianchi, Proceeding Romansy 1984, Cambridge, USA. MIT press, 1985, p. 39-46.

[51] SCHMITT, D., et al. *"Optimal Motion Programming of Robot Manipulators"*, Journal of Mechanisms, Transmissions, and Automation in Design, vol. 107, june 1985, p. 239-244.

[52] SHAHINPOOR, M. *"A robot engineering textbook"*, New York, Harper and Row edition, 1987, 480 p.

[53] SHILLER, Z., and S. DUBOWSKY. *"On the Optimal Control of Robotic Manipulators with Actuator and End-Effeector Contraints"*, Proceeding IEEE Conference Robotics and Automation, 1985, p. 614-620.

[54] SHIN, Kang G., and Neil D. McKAY. *"A Dynamic Programming Approach to Trajectory Planning of Robotic Manipulators"*, IEEE Transactions on Automatic Control, vol. AC-31, no. 6, june1986, p. 491-500.

[55] SHIN, Kang G., and Neil D. McKAY. *"Minimum-Time Control of Robotic Manipulators with Geometric Path Constraints"*, IEEE Transactions on Automatic Control, vol. AC-30, no. 6, june1985, p. 531-541.

[56] SHIN, Kang G., and Neil D. McKAY. *"Selection of Near-Minimum Time Geometric Paths for Robotic Manipulators"*, IEEE Transactions on Automatic Control, vol. AC-31, no. 6, june1986, p. 501-511.

[57] SILVER, William M. *"On the Equivalence of Lagrangian and Newton-Euler Dynamics for Manipulators"*, The International Journal of Robotics Research, vol. 1, no. 2, summer 1982, p. 60-70.

[58] TAYLOR, Russell H. *"Planning and Execution of straight Line Manipulator Trajectories"*, IBM Journal of Research and Development, vol. 23, july 1979, p. 424-436.

[59] VUKOBRATOVIC, M., and M. KIRCANSKI. *"A Method for Optimal Synthesis of Manipulation Robot Trajectories"*, Journal of Dynamic Systems, Measurements and Control (ASME), vol. 104, june 1982, p. 188-193.

[60] WANG, J.T., and R. L. HUSTON. *"Kane's Equations with Undetermined Multipliers-Application to Constrained Multibody Systems"*, Journal of Applied Mechanics, vol. 54, june 1987, p. 424-429.

[61] WHITNEY Daniel E. *"Force Feedback Control of Manipulator Fine Motions"*, Journal of Dynamic Systems, Measurements and Control (ASME), june 1977, p. 91-97.

[62] WHITNEY Daniel E. *"The Mathematics of Coordinated Control of Prosthetic Arms and Manipulators"*, Journal of Dynamic Systems, Measurements and Control , december 1972, p. 303-309.

[63] WU, Chi-Haur, and Richard P. PAUL. *"Resolved Motion Force Control of Robot Manipulator"*, IEEE Transactions on Systems, Man and Cybernetics , vol. SMC-12, no. 3, june 1982, p. 266-275.

[64] YAMAMOTO M. and Mohri A. *"Planning of Quasi-Minimum Time Trajectories of Robot Manipulators (Generation of Bang-Bang Control)"*, Robotica, vol. 7, 1989, p. 43-47.

# Appendix A

## Develpment of the Trajectory Equations Based on Gauss Function

This appendix describes the steps and the trajectory equations obtained based on the approximation of the Gauss function by Maclaurin series.

The shape of the acceleration is given by Equation 3.1, and using Maclaurin series approximation, it follows:

$$q(T) = \frac{1}{\sigma\sqrt{2\pi}} \sum_{n=0}^{n} \frac{(-1)^n}{n!2^n} (\frac{t-\mu}{\sigma})^{2n} + C_0 \qquad (A.1)$$

The acceleration rersults in a polynomial of degree n.

Using normalized time as stated in Chapter 4, and integrating Equation A.1 by term-wise integration, the position and velocity are obtained as follow:

the position:

$$q_j(T) = \frac{1}{\sigma\sqrt{2\pi}} \{ \sum_{n=0}^{N} \frac{(-1)^n \sigma^2}{n!2^n(2n+1)(2n+2)} (\frac{T-\mu}{\sigma})^{2n+2} \} + .5C_0T^2 + C_1T + C_2 \quad (A.2)$$

the velocity:

$$\dot{q}_j(T) = \frac{1}{t_j\sigma\sqrt{2\pi}} \{ \sum_{n=0}^{N} \frac{(-1)^n \sigma}{n!2^n(2n+1)} (\frac{T-\mu}{\sigma})^{2n+1} \} + \tau(C_0T + C_1) \qquad (A.3)$$

Where $C_1$ and $C_2$ are constants of integration, the acceleration as a function of normilized time is given by:

$$\ddot{q}_j(t) = \frac{1}{t_j^2\sigma\sqrt{2\pi}}\{\sum_{n=0}^{n}\frac{(-1)^n}{n!2^n}(\frac{T-\mu}{\sigma})^{2n}\} + C_0 \tag{A.4}$$

Applying the boundaries conditions for the transition time given in Chapter 4, the coefficients of the polynomial are obtained as follow:

$$C_0 = -\frac{1}{\sigma\sqrt{2\pi}}\{\sum_{n=0}^{N}\frac{(-1)^n}{n!2^n}(\frac{\mu}{\sigma})^{2n}\} + \ddot{\theta}_e t_j^2 \tag{A.5}$$

$$C_1 = -\frac{1}{\sigma\sqrt{2\pi}}\{\sum_{n=0}^{N}\frac{(-1)^n\sigma}{n!2^n(2n+1)}(\frac{-\mu}{\sigma})^{2n+1}\} + \dot{theta}_e t_j \tag{A.6}$$

$$C_2 = -\frac{1}{\sigma\sqrt{2\pi}}\{\sum_{n=0}^{N}\frac{(-1)^n\sigma^2}{n!2^n(2n+1)(2n+2)}(\frac{\mu}{\sigma})^{2n+2}\} + \theta_e \tag{A.7}$$

To find the parameter $\mu$ and $\sigma$, one has to solve the following equations simultaneously.

$$\begin{aligned}&\tfrac{1}{t_j^2\sigma\sqrt{2\pi}}\{\textstyle\sum_{n=0}^{N}\frac{(-1)^n\sigma^2}{n!2^n(2n+1)(2n+2)}[(\frac{1-\mu}{\sigma})^{2n+2} - (\frac{\mu}{\sigma})^{2n+2}]\}\\ &-\tfrac{1}{t_j^2\sigma\sqrt{2\pi}}\{\textstyle\sum_{n=0}^{N}\frac{(-1)^n\sigma}{n!2^n(2n+1)}(\frac{-\mu}{\sigma})^{2n+1} - \frac{1}{t_j^2\sigma\sqrt{2\pi}}\{\sum_{n=0}^{N}\frac{(-1)^n}{n!2^n}(\frac{\mu}{\sigma})^{2n}+\\ &(\theta_e-\theta_s) + t_j(\dot{\theta}_s - \dot{\theta}_e) - t_j^2\ddot{\theta}_e = 0.\end{aligned} \tag{A.8}$$

and

$$\begin{aligned}&\tfrac{1}{t_j\sigma\sqrt{2\pi}}\{\textstyle\sum_{n=0}^{N}\frac{(-1)^n\sigma}{n!2^n(2n+1)}[(\frac{1-\mu}{\sigma})^{2n+1} + (\frac{\mu}{\sigma})^{2n+1}]\}\\ &-\tfrac{1}{t_j\sigma\sqrt{2\pi}}\{\textstyle\sum_{n=0}^{N}\frac{(-1)^n}{n!2^n}(\frac{\mu}{\sigma})^{2n} + (\dot{\theta}_s - \dot{\theta}_e) + t_j\ddot{\theta}_e = 0.\end{aligned} \tag{A.9}$$

The resulting position, velovity and acceleration are portrayed in Figure 3.3. This approach is withdrawn for the following reasons:

a) To approximate the Gauss function by Maclaurin series within a certain range of error, n should be of the order of 9, and this results in a polynomial of degree 18. This leads to a difficulty to compute since a polynomial of high degree suffer from several drawbacks.

b) The parameters $\mu$ and $\sigma$ are found by solving Equations A.8 and A.9 using trial and error method since the equations are non linear. This results in a high computational time.

# Appendix B

Physical Characteristics of the Simulated Robot

The following characteristics are approximated from a real model of robot manipulator. Table B.1 gives the lenths characteristics, Table B.2 gives the masses and inertia of each link, and Table B.3 gives the joints limit for the simulated robot. These values are used in the simulation through out this thesis.

Table B.1: Lengths characteristics of the simulatex robot.

| Link # | $l_i$ (m) | $a_i$ (m) | $d_i$ (m) |
|---|---|---|---|
| 1 | - | - | .5 |
| 2 | .5 | .9 | - |
| 3 | .5 | - | .7 |

Table B.2: Masses and Inertia of the simulated robot.

| Link # | $m_i$ (m) | $I_{ix}$ (kg.$m^2$) | $I_{iy}$ ($kg.m^2$) | $I_{iz}$ ($kg.m^2$) |
|---|---|---|---|---|
| 1 | 40. | 1.5 | .1 | 1.5 |
| 2 | 16. | .12 | .12 | .01 |
| 3 | 12. | .12 | .12 | .01 |

Table B.3: Joint limits of the simulated robot.

| Position (deg.) | Velocity (rad/s) | acceleration ($rad/s^2$) | Torque (N.m) |
|---|---|---|---|
| $0. \leq \theta_1 \leq 90.$ | $-1. \leq \dot{\theta}_1 \leq 1.$ | $-1. \leq \ddot{\theta}_1 \leq 1.$ | $-150. \leq \tau_1 \leq 150.$ |
| $180. \leq \theta_2 \leq 360.$ | $-1. \leq \dot{\theta}_2 \leq 1.$ | $-1. \leq \ddot{\theta}_2 \leq 1.$ | $-200. \leq \tau_2 \leq 200.$ |
| $-180. \leq \theta_3 \leq 360.$ | $-1. \leq \dot{\theta}_3 \leq 1.$ | $-1. \leq \ddot{\theta}_3 \leq 1.$ | $-200. \leq \tau_3 \leq 200.$ |

# Appendix C

## Solution of a Seven Order Polynomial

To satisfy the continuity up to the third derivative "jerk" for a transition segment as shown in Figure C.1. A polynomial of seven degrees is required.

The polynomial function is given by:

$$q(t) = a_7 t^7 + a_6 t^6 + a_5 t^5 + a_4 t^4 + a_3 t^3 + a_2 t^2 + a_1 t + a_0 \qquad \text{(C.1)}$$

Imposing the following constraints:

$q(t_s) = A$ $\qquad$ $q(t_e) = C$

$\dot{q}(t_s) = B$ $\qquad$ $\dot{q}(t_e) = D$

$\ddot{q}(t_s) = 0.$ $\qquad$ $\ddot{q}(t_e) = 0.$

$\dddot{q}(t_s) = 0.$ $\qquad$ $\dddot{q}(t_e) = 0.$

The coefficients are obtained as follow:

$$a_0 = \left[16(A+B) + 11(C-D)T\right]/32 \qquad \text{(C.2)}$$

$$a_1 = -\left[35(A-B) + 19(C+D)T\right]/(32T) \qquad \text{(C.3)}$$

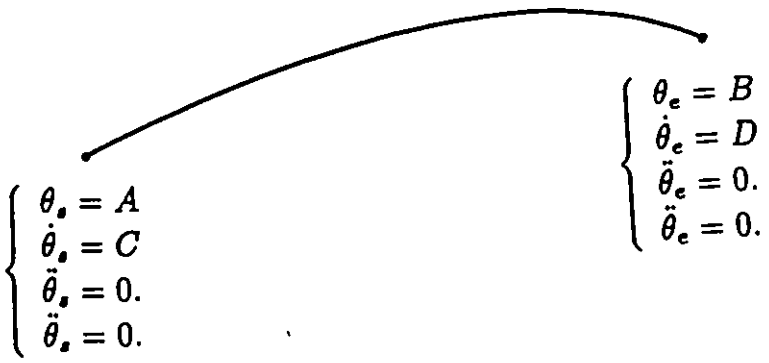



Figure C.1: Typical boundaries conditions for a transition segment.

$$a_2 = -15\,(C - D)\,/(32T) \tag{C.4}$$

$$a_3 = 35\,[(A - B) + (C + D)T]\,/(32T^3) \tag{C.5}$$

$$a_4 = 5\,(C - D)\,/(32T^3) \tag{C.6}$$

$$a_5 = -21\,[(A - B) + (C + D)T]\,/(32T^5) \tag{C.7}$$

$$a_6 = -\,(C - D)\,/(32T^5) \tag{C.8}$$

$$a_7 = 5\,[(A - B) + (C + D)T]\,/(32T^5) \tag{C.9}$$

Where The values of "B" and "T" are:

$$B = A + T(C + D) \tag{C.10}$$

$$T = 15(C - D)/(16\alpha_{max}) \tag{C.11}$$