A SOLUTION TECHNIQUE FOR THE MINIMUM-TIME CONTROL PROBLEM OF AN R-THETA MANIPULATOR

by

ANUP SHEETY

B.S., Wichita State University, 1985

A MASTERS THESIS

submitted in partial fulfillment of the requirements for the degree MASTER OF SCIENCE

Department of Mechanical Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1987

Approved by:

[Signature]

Major Professor
# TABLE OF CONTENTS

LIST OF TABLES ................................................................. ii
LIST OF ILLUSTRATIONS ......................................................... iii
ACKNOWLEDGEMENTS .............................................................. iv

Chapter
I. INTRODUCTION .................................................................. 1
II. THE MINIMUM TIME CONTROL PROBLEM .............................. 7
III. MATHEMATICAL MODEL ...................................................... 15
   KINEMATIC MODEL ....................................................... 16
   DYNAMIC MODEL .......................................................... 19
IV. THE FINITE ELEMENT METHOD .......................................... 33
V. RESULTS AND RECOMMENDATIONS .................................... 63

LIST OF REFERENCES ............................................................. 77
APPENDICES ......................................................................... 79
   APPENDIX I ..................................................................... 79
   APPENDIX II ................................................................. 82
LIST OF TABLES

5.1 Discrete time solutions for the minimum final time .... 67
5.2 Effect of grid density on final time for Case 1 .... 67
# LIST OF ILLUSTRATIONS

<table>
<thead>
<tr>
<th>Section</th>
<th>Illustration Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>The R-Theta Manipulator</td>
<td>17</td>
</tr>
<tr>
<td>3.2</td>
<td>Kinematic Model</td>
<td>18</td>
</tr>
<tr>
<td>3.3</td>
<td>One-dimensional illustration of two possible types of minima with inequality constraints</td>
<td>27</td>
</tr>
<tr>
<td>4.1</td>
<td>Natural Coordinates for First Order, One Dimensional Finite Element</td>
<td>37</td>
</tr>
<tr>
<td>4.2</td>
<td>Element Matrix Equation</td>
<td>49</td>
</tr>
<tr>
<td>4.3</td>
<td>Element Jacobian Equation</td>
<td>52</td>
</tr>
<tr>
<td>5.1</td>
<td>Joint 1 Variable</td>
<td>69</td>
</tr>
<tr>
<td>5.2</td>
<td>Joint 2 Variable</td>
<td>70</td>
</tr>
<tr>
<td>5.3</td>
<td>Lambda1</td>
<td>71</td>
</tr>
<tr>
<td>5.4</td>
<td>Lambda2</td>
<td>72</td>
</tr>
<tr>
<td>5.5</td>
<td>Joint 1 Variable</td>
<td>73</td>
</tr>
<tr>
<td>5.6</td>
<td>Joint 2 Variable</td>
<td>74</td>
</tr>
<tr>
<td>5.7</td>
<td>Lambda1</td>
<td>75</td>
</tr>
<tr>
<td>5.8</td>
<td>Lambda2</td>
<td>76</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A.1</td>
<td>Flow Chart of Finite Element Program</td>
<td>82</td>
</tr>
</tbody>
</table>
ACKNOWLEDGEMENTS

I would like to thank the following people for their contributions to this thesis: Dr. Warren N. White, Jr., whose guidance and enthusiasm made this paper possible, Dr. J. Garth Thompson for his guidance in the optimal control theory concepts and for serving on my committee, Dr. Prakash Krishnaswami for his help with the minimization techniques and for serving on my committee, Dr. Ruth A. Dyer for serving on my committee, the Department of Mechanical Engineering, Kansas State University, and the Center of Excellence for Research in Computer-Controlled Automation, College of Engineering, Kansas State University and finally, my friends for their encouragement and support.
American industry, as a whole, and the automobile industry in particular, has been investing heavily in recent years to modernize manufacturing facilities, to lessen manufacturing costs, and to improve quality. Emphasis has been placed on teachable automation in an effort to ensure that new equipment will be flexible enough to accommodate future products with minimal additional investment. The primary teachable components of flexible manufacturing systems are robots, and the primary component of a robotic system that allows it to be teachable is computer control.

However, computer control of robots is not well refined and robots do not perform up to their physical capabilities. Higher performance robots would be valuable in manufacturing. For example, many applications of robots in factories can be justified economically only if implemented with a faster robot than state-of-the-art control permits. Other applications could be implemented with fewer faster robots, which would result in considerable investment savings. Another performance limitation, besides speed, is the maximum load bearing capacity of commercially available robots.
These performance limitations can be traced to the manner in which existing robots are controlled. State-of-the-art control schemes are based solely on stability requirements. The control law is designed to produce stability in axis position and then the fact of stability itself is used to induce motion by iteratively changing the position reference. This form of control gives rise to inaccuracy at high speeds and to position overshoots. Consequently robot designers have restricted the peak speed and acceleration of their products so that accuracy and overshoot can be limited to acceptable levels. This is a performance limitation due to the control laws and not due to the capabilities of the machine. Therefore, to control the robot for high performance, the true physical performance limitations must first be established. The limitations are then based on constraints and not merely on the control [1].

A real life example will illustrate this difference in performance levels. The standard PUMA arm equipment could allow base motor speeds of up to 144 rad/s, and currently, the limit set in the Puma controller is 89.9 rad/s [1]. A 60% improvement is possible if the full potential of the motor can be tapped.

The question of what is minimum time control can be answered with a commonly used analogy. If a person were travelling in a car and wished to get to the next intersection in the shortest possible time, what would he do? He would push the accelerator to the floor for a certain amount of time and then releasing the accelerator apply maximum braking (switch controls) for some other period of time to come to a stop. If the control switched too late, the car would slide into the
intersection. If it switched too soon, the car would stop short of the intersection. Thus, it is obvious that the switching time is critical in obtaining the desired final position and velocity. This is the basic idea in the minimum time optimal control of a robot manipulator or articulated mechanism. This type of control is also called bang-bang control [2], because the control is always on the control boundary, in one direction or the other.

The minimum time, optimal control problem for a robotic manipulator can be divided into two main classes, minimum time control along a specified path and minimum time control with no path constraints. Task oriented problems and obstacle avoidance planning fall into the former class. The second class can be subdivided into two categories, that of problems where the complete nonlinear minimum time system is considered and the true solution is sought, and of problems where approximations (usually linearizations) are made on the nonlinear system and the near-minimum time control is investigated. Extensive work has been done with problems belonging to the first class [3]–[7]. Problems dealing with the near-minimum time control have also been quite extensively investigated in recent years. In 1971 Kahn and Roth published a paper on the near-minimum time control of open loop articulated kinematic chains [8]. They developed a suboptimal feedback control by linearizing the equations of motion for a three degree of freedom manipulator. Approximations were made for the effects of gravity loads and angular velocities in the nonlinear dynamic equations. The suboptimal control was obtained by decoupling the system into three double integrators and deriving the equations for the switching curves.
of the transformed system. The response time of the suboptimal control was compared to that of the optimal control which was obtained by an iterative technique. Wen and Desrochers [9] investigated two control strategies for suboptimal control, the method of averaging dynamics (AD) and the method of linear equivalence (LE). The first method is used when a time-fuel suboptimal solution is required. The latter uses exact linearization where the dynamic equations are written for a reduced system of decoupled double integrators. The LE method is found to be superior to the AD method in obtaining a smaller final time. However both methods need a very good model of the system since the nonlinear part of the system has to be evaluated repetitively. Sato, Shimojima and Kitamura [10] obtained switching times of the control variables for a two degree of freedom manipulator by approximating the velocity of a DC servomotor. They found that when the driving force was operating at saturation it was necessary to make additional approximations on the angular velocities. Kao, Sinha and Mahalanabis [11] developed an algorithm for the near-minimum time control of a three link robotic manipulator. They linearized the dynamic equations by expanding them in a Taylor series and neglecting the higher order terms. The poles of the linearized closed loop system were placed in the z-plane so as to permit minimum time response without violating the actuator torque constraints. This is a digital algorithm that can be implemented using microprocessors.

However very little work has been done in the area of the complete problem, the problem of determining the minimum time optimal control history for a system with no linearizations or approximations. Kahn and
Roth [8] obtained the minimum time optimal control by an iterative technique. Guesses were made on the unknown constants at the final time $\lambda(t_f)$, and the dynamic equations were integrated backwards in time to the initial time to give the states $x(t_o)$ and the constants $\lambda(t_o)$. If $x(t_o)$ is not sufficiently close to the specified initial state $x_o$ then a new set of variables $\lambda(t_f)$ are chosen and the integration is repeated. The iteration is continued till $x(t_o)$ is sufficiently close to $x_o$. Then the constants $\lambda(t_o)$ and $x_o$ are substituted into the dynamic equations and the optimal control is obtained by integrating the equations forward to the final time.

The purpose of this work is to develop an alternate method of solving the complete minimum time problem, to examine the deviation of the discrete time solution (finite element method) from the continuous case, and to explore the feasibility of a real-time minimum time controller.

In Chapter 2, the minimum time control problem is stated. The basic concepts of control theory, as well as some variational calculus principles, utilized in the problem formulation, are presented.

In Chapter 3, the mathematical model of the r-theta manipulator (a two degree of motion manipulator) is developed. The dynamic equations, derived using the Lagrangian formulation, are used in the control algorithm for the minimum time simulation of the manipulator.

In Chapter 4, the finite-element solution technique for the minimum time problem is developed.
In Chapter 5, the simulation results are presented. The finite element method is found to converge to the solution with reasonable initial guesses on unknown parameters. When this method is used in conjunction with a grid search method to start the algorithm, it converges quite rapidly to the true solution. The discrete time solution compares favorably with the continuous case, and as the grid density of the finite element mesh is increased the accuracy of the solution is improved.

Some of the limitations of the technique, as well as recommendations on areas for further investigations are also presented.
CHAPTER 2

THE MINIMUM TIME CONTROL PROBLEM

Optimal Control Theory

The objective of optimal control theory is to determine the control signals that will cause a process to satisfy the physical constraints and at the same time minimize (or maximize) some performance criterion.

In order to evaluate the performance of a system quantitatively, the designer has to select a performance index or cost function $J$. An optimal control is defined as one that minimizes (or maximizes) the performance index.

In the general case, it will be assumed that the performance of a system is evaluated by a measure of the form [12]

$$ J = h(x(t_f), t_f) + \int_{t_o}^{t_f} g(x(t), u(t), t) \, dt \quad (2.1) $$

where $t_o$ is the initial time, $t_f$ is the final time, and $h$ and $g$ are scalar functions. The final time $t_f$ may be specified or free depending on the problem statement. For the minimum time problem

$$ J = \int_{t_o}^{t_f} 1 \, dt \quad (2.2) $$
where \( t_f \) is unspecified.

Throughout this paper bold face characters will represent vectors. For example \( x(t) \) and \( u(t) \) are the state and control vectors.

The optimal control problem is to find an admissible control which causes the system described by the set of first order ordinary differential equations

\[
\dot{x}(t) = a(x(t), u(t), t)
\]  

(2.3)

to follow an admissible trajectory \( x^*(t) \) that minimizes (or maximizes) the performance index \( J \). The quantity \( u^*(t) \) which minimizes \( J \) is called the optimal control and \( x^*(t) \) an optimal trajectory. The elements of equation (2.3) are called state equations and involve the state variables \( x(t) \) and the controls \( u(t) \). A more formal definition of state variables is provided in Chapter 3.

A control history which satisfies the control constraints during the entire time interval \( [t_0, t_f] \) and achieves the desired final state \( x(t_f) \) is called an admissible control. A state trajectory which satisfies the state variable constraints, both the differential equation constraints as well as the boundary constraints, during the entire time interval \( [t_0, t_f] \) is called an admissible trajectory.

**Variational Formulation**

Variational calculus is a branch of mathematics that is very useful in solving optimization problems. The performance index \( J \) is a functional. A functional is a function of a function and/or functions.
For example, if

\[ x_1 = f_1(q_1, q_2) \]  
\[ x_2 = f_2(q_1, q_2) \]

and

where \( q_1, q_2 \) are independent variables and \( f_1, f_2 \) are scalar functions

then the quantity

\[ J = g(x_1, x_2) \]

is a functional where \( g \) is a scalar function.

Variation of a Functional

The variation of a functional plays the same role in determining extreme values (maximum or minimum) of a functional as does the differential in finding maxima or minima of functions. The differential, \( df \), of a function, \( f \), of variables \( q_1, q_2, \ldots, q_n \) is given by the relation

\[ df = \sum_{i=1}^{n} \frac{\partial f}{\partial q_i} dq_i \]

Similarly, the variation, \( \delta J \), of a functional, \( J \), of functions \( x_1, x_2, \ldots, x_n \) is given by the relation

\[ \delta J = \sum_{i=1}^{n} \frac{\partial J}{\partial x_i} \delta x_i \]

Fundamental Theorem of the Calculus of Variations

The fundamental theorem states that the variation must be zero on an extremal (maximum or minimum) curve, provided there are no bounds imposed on the curves.
In other words

\[ \delta J(x^*, \delta x) = 0 \]  \hspace{1cm} (2.7)

for all admissible \( \delta x \). By admissible \( \delta x \), it is meant that \( x + \delta x \) must be of some class \( \Omega \) to which \( x \) belongs. For example, if \( \Omega \) is the class of continuous functions, \( x \) and \( \delta x \) must both be continuous. In this case \( \Omega \) comprises all the state histories which satisfy (2.3).

**Constrained Minimization of Functionals**

So far, functionals involving the state vector \( x(t) \) have been discussed and it has been assumed that the components of \( x(t) \) are independent. This is usually not the case in control problems where the state trajectory is determined by the control \( u(t) \) and the state equations. Therefore it is necessary to consider functionals of \( n+m \) functions, \( x(t) \) and \( u(t) \), but only \( m \) of the functions are independent - the controls. The next step is to derive the necessary conditions for extremals of constrained systems. The Lagrangian multiplier method will be used.

**The Lagrangian Multiplier Method for a System with Differential Equation Constraints**

The objective is to find the necessary conditions for functions \( x^*(t) \) and \( u^*(t) \) to be extremals for a functional

\[ J(x,u) = \int_{t_0}^{t_f} g(x(t), u(t), t) \, dt \]  \hspace{1cm} (2.8)

where \( x^*(t) \) is the state vector of order \( n \) and \( u^*(t) \) is the control vector of order \( m \). These vectors must also satisfy equation (2.3), the
differential equation constraints on the states. To include these constraints the augmented functional is formed. The augmented functional is defined as

\[
J_a(x, u, \lambda) \triangleq \int_0^T \left[ g(x(t), u(t), t) + \lambda^T(t) [a(x(t), u(t), t) - \dot{x}(t)] \right] dt \tag{2.9}
\]

where \( \lambda_i(t) \) \( i = 1, 2, \ldots, n \) are the Lagrangian multipliers whose values are to be determined. When the constraints are satisfied, the augmented functional, \( J_a \), equals the functional, \( J \), for any \( \lambda(t) \).

The quantity \( g_a(x(t), \dot{x}(t), u(t), \lambda(t), t) \) can be defined as

\[
g_a(x(t), \dot{x}(t), u(t), \lambda(t), t) \triangleq g(x(t), u(t), t)
\]

so that

\[
J_a(x(t), \dot{x}(t), u(t), \lambda(t), t) = \int_0^T \left[ g_a(x(t), \dot{x}(t), u(t), \lambda(t), t) \right] dt. \tag{2.11}
\]

The variation of the functional \( J_a \), \( \delta J_a \), after integrating by parts and simplifying is

\[
\delta J_a = \left[ \frac{\partial g_a(x(t_f), \dot{x}(t_f), u(t_f), \lambda(t_f), t_f)}{\partial \dot{x}} \right]^{T_f}_{t_f} \delta t_f + \left[ g_a(x(t_f), \dot{x}(t_f), u(t_f), \lambda(t_f), t_f) \right]^{T}_t \dot{x}(t_f) \delta t_f
\]

\[
+ \int_0^T \left[ \left[ \frac{\partial g_a(x(t), \dot{x}(t), u(t), \lambda(t), t)}{\partial x} \right]^{T_t}_{t_o} \delta t + \frac{d}{dt} \left[ \frac{\partial g_a(x(t), \dot{x}(t), u(t), \lambda(t), t)}{\partial \dot{x}} \right] \delta x(t) \right].
\]

-11-
The necessary conditions can be derived from the above equation by applying the fundamental theorem. However it is more convenient to use another functional, the Hamiltonian, which can be defined as [12]

$$H(x(t), u(t), \lambda(t), t) = g(x(t), u(t), t) + \lambda^T(t)[a(x(t), u(t), t)]$$

(2.13)

where \(a(x(t), u(t), t)\) is the right hand side of equation (2.3).

For the minimum time problem, the Hamiltonian can be written as

$$H = 1 + \lambda^T(t)[a(x(t), u(t), t)].$$

(2.14)

For an extremal curve the fundamental theorem gives us the condition

$$\delta J_a(x^*(t), u^*(t), \lambda^*(t), t) = 0.$$  

(2.15)

The superscript * signifies the extremal or optimal value.

The above equation gives us the necessary but not the sufficient conditions for optimal control which are

$$\dot{x}(t) = \frac{\partial H}{\partial \lambda}(x^*(t), u^*(t), \lambda^*(t), t),$$

(2.16)

$$\lambda(t) = -\frac{\partial H}{\partial x}(x^*(t), u^*(t), \lambda^*(t), t),$$

(2.17)

$$\frac{\partial H}{\partial u}(x^*(t), u^*(t), \lambda^*(t), t) = 0,$$

(2.18)

and

$$0 = [-\lambda^*(t_f)]^T \delta x_f + [H(x^*(t_f), u^*(t_f), \lambda^*(t_f), t_f)] \delta t_f.$$  

(2.19)
Equation (2.16) constitutes the \( n \) state equations. Equation (2.17) constitutes the \( n \) co-state equations. Equation (2.18) constitutes the \( m \) optimality conditions. Equation (2.19) constitutes the boundary condition equation. The conditions above are not sufficient to solve the minimum time problem because constraints on the controls are required to solve the problem. If the controls are unconstrained then the optimal control will be infinite torque or infinite force and the minimum time will be zero. The control constraints are defined in Chapter 3. For the minimum time problem the final time, \( t_f \), is free to vary, but the final state \( x_f \) is fixed. Therefore

\[
\delta x_f = 0. \tag{2.20}
\]

The boundary condition equation reduces to

\[
H(x^*(t_f), u^*(t_f), \lambda^*(t_f), t_f) = 0. \tag{2.21}
\]

This equation is also called the transversality equation.

So far it has been assumed that the admissible controls and states are not constrained by any boundaries, however, in realistic systems such constraints do commonly occur. Physically realizable controls generally have magnitude limitations. Actuators in robot joints have a maximum torque output beyond which they saturate. The generalization of the fundamental theorem to include the effects of the control boundary constraints leads to Pontryagin's minimum principle.

Pontryagin's minimum principle states that an optimal control must minimize the Hamiltonian, i.e.

\[
H(x^*(t), u(t), \lambda^*(t), t) \leq H(x^*(t), u(t), \lambda^*(t), t) \tag{2.22}
\]
for any $t \in [t_0, t_f]$ and for all admissible controls.

The conditions for minimum time control, equations (2.16), (2.17), (2.18), (2.21) and (2.22) are utilized in the continuous time simulation and in the discrete time simulation of a robotic manipulator in Chapters 3 and 4, respectively.
CHAPTER 3

MATHEMATICAL MODEL

The Concept of State

The concept of state occupies a central position in modern control theory. It is a complete summary of the status of the system at a particular point in time. Knowledge of the state at some initial time $t_0$, plus knowledge of the system inputs after $t_0$, allows the determination of the state at some later time $t_1$. At any fixed time the state of a system can be described by the values of a set of variables $x_i$, $i = 1, 2, \ldots, n$ where $n$ is the order of the system. These variables are called the state variables.

The Mathematical Model

An important part of any control problem is modelling the process. The objective is to obtain the simplest mathematical model that adequately predicts the response of the physical system to all anticipated inputs. The $r$-theta manipulator belongs to the class of systems that can be described by ordinary differential equations in state variable form. Thus if $x_1(t), x_2(t), \ldots, x_n(t)$ are the state variables of the process at time $t$ and $u_1(t), u_2(t), \ldots, u_m(t)$ are the
control inputs to the process at time \( t \), then the system may be described by \( n \) first order differential equations, such as

\[
\begin{align*}
\dot{x}_1(t) &= a_1(x_1(t), x_2(t), \ldots, x_n(t), u_1(t), u_2(t), \ldots u_m(t)) \\
\dot{x}_2(t) &= a_2(x_1(t), x_2(t), \ldots, x_n(t), u_1(t), u_2(t), \ldots u_m(t)) \\
&\quad \vdots \\
\dot{x}_n(t) &= a_n(x_1(t), x_2(t), \ldots, x_n(t), u_1(t), u_2(t), \ldots u_m(t))
\end{align*}
\]

The state vector \( x(t) \) of the system is defined as

\[
x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix}
\]

and the control vector is defined as

\[
m(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_m(t) \end{bmatrix}
\]

The state equations in vector form are

\[
\dot{x}(t) = a(x(t), m(t), t).
\] (3.4)

**Kinematic Model**

The two degrees-of-freedom robotic manipulator on which the minimum time control is performed is called an \( r \)-theta manipulator. A schematic of the manipulator is shown in Figure 3.1. The kinematic model is illustrated in Figure 3.2.
The r-theta manipulator has two joints. Joint 1 is revolute and joint 2 is prismatic. The joint variables are θ and r (or D₂ in the Denavit-Hartenberg representation [13] used in Figure 3.2), respectively. The torque on joint 1 is T and the force on joint 2 is F. The plane of motion of the manipulator is parallel to the ground and hence the gravitational force does not enter into the dynamic equations.

Dynamic Model

The equations of motion are nonlinear and coupled. In order to simplify the equations, and hence the simulation, the following assumptions are made.

1. The mass of the payload is much greater than the mass of the links and actuators.

2. The payload m is treated as a point mass.

The first assumption allows the r-theta links to be treated as massless kinematic linkages. The second assumption simplifies the inertia terms.

The dynamic equations are derived from Lagrange's equation of motion. If not all the forces acting on the system are derivable from a potential, then Lagrange's equations can be written in the form [14]

\[
\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_j} \right) - \frac{\partial L}{\partial q_j} = Q_j
\]  

(3.5)

where L is the Lagrangian, q_j represents the generalized coordinates, and Q_j represents the forces not arising from a potential.
If KE is the kinetic energy and PE the potential energy then the Lagrangian is defined as [14]

\[ L = KE - PE \]  \hspace{1cm} (3.6)

Summing up the kinetic energy of the manipulator gives

\[ KE = \frac{1}{2} m(\dot{r})^2 + \frac{1}{2} m(\dot{\theta})^2 \]  \hspace{1cm} (3.7)

where \( m \) is the mass of payload, \( \theta \) is the joint 1 variable, \( \dot{\theta} \) is the time derivative of \( \theta \), namely \( \frac{d}{dt}(\theta) \), \( r \) is the joint 2 variable, and \( \dot{r} \) is the time derivative of \( r \), namely \( \frac{d}{dt}(r) \). The potential energy of the manipulator is

\[ PE = 0 \]  \hspace{1cm} (3.8)

The quantities \( r \) and \( \theta \) are explicit functions of time, \( t \). Throughout this chapter the independent variable \( t \) is omitted from the notation of the explicit functions of \( t \) for brevity. Also, throughout this chapter the superscript ' will indicate the first differential with respect to time \( \frac{d}{dt} \) and the superscript '' will indicate the second differential with respect to time \( \frac{d^2}{dt^2} \).

The forces at the joints are

\[ Q_1 = T \]  \hspace{1cm} (3.9)

and

\[ Q_2 = F \]  \hspace{1cm} (3.10)
where $T$ is the torque at joint 1 and $F$ is the force at joint 2. Substituting equations (3.7) and (3.8) in equation (3.6) gives the Lagrangian

$$L = \frac{1}{2} m (\dot{r}^2 + r^2 \dot{\theta}^2) . \tag{3.11}$$

For the $r$-theta manipulator, the independent generalized coordinates are $r$ and $\theta$. The Lagrangian equations for the $r$-theta manipulator are of the form

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{r}} \right) - \frac{\partial L}{\partial r} = F, \tag{3.12}$$

and

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}} \right) - \frac{\partial L}{\partial \theta} = T . \tag{3.13}$$

From equation (3.11) the expressions for the various derivative terms of equations (3.12) and (3.13) are

$$\frac{\partial L}{\partial r} = m r \dot{\theta}^2 , \tag{3.14}$$

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{r}} \right) = \frac{d}{dt} (m \ddot{r}) = m \ddot{r} , \tag{3.15}$$

$$\frac{\partial L}{\partial \theta} = 0 , \tag{3.16}$$

and

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}} \right) = \frac{d}{dt} (m r^2 \dot{\theta}) = m r^2 \ddot{\theta} + 2mr \dot{r} \dot{\theta} . \tag{3.17}$$

Substituting equations (3.14) and (3.15) into equation (3.12) gives

$$m \ddot{r} - m r \dot{\theta}^2 = F . \tag{3.18}$$

Substituting equations (3.16) and (3.17) into equation (3.13) gives
\[ mr^2 \dddot{\theta} + 2mr\dot{\theta} = T. \quad (3.19) \]

The controls are defined as
\[ u_1 = \frac{F}{F_{\text{max}}} \quad (3.20) \]
and
\[ u_2 = \frac{T}{T_{\text{max}}} \quad (3.21) \]
where \( u_1 \) is the control at joint 2, \( u_2 \) is the control at joint 1, \( T \) is the actual torque applied at joint 1, \( T_{\text{max}} \) is the maximum torque that can be applied at joint 1, \( F \) is the actual force applied at joint 2, and \( F_{\text{max}} \) is the maximum force that can be applied at joint 2. The controls \( u_1 \) and \( u_2 \) are explicit functions of \( t \).

Substituting equations (3.21) and (3.20) into equations (3.19) and (3.18), respectively, gives
\[ mr^2 \dddot{\theta} + 2mr\dot{\theta} - T_{\text{max}} u_2 = 0 \quad (3.22) \]
and
\[ mr^2 \dddot{\theta} - mr\dot{\theta}^2 - F_{\text{max}} u_1 = 0. \quad (3.23) \]

The state variables \( x_1, x_2, x_3, \) and \( x_4 \) are now introduced. They are defined as
\[ x_1 = r, \quad (3.24.1) \]
\[ x_2 = \dot{r}, \quad (3.24.2) \]
\[ x_3 = \theta, \quad (3.24.3) \]
and
\[ x_4 = \dot{\theta}. \quad (3.24.4) \]
From equations (3.22), (3.23) and (3.24) the state equations can be written as

\[
\begin{align*}
\dot{x}_1 &= x_2, \\
\dot{x}_2 &= x_1 x_4^2 + \frac{F_{\text{max}}}{m} u_1, \\
\dot{x}_3 &= x_4,
\end{align*}
\]

and

\[
\dot{x}_4 = \frac{2 x_2 x_4^2}{x_1} + \frac{T_{\text{max}}}{m x_1^2} u_2
\]

where \(x_1, x_2, x_3, x_4\) are explicit functions of \(t\).

In this paper the performance index \(J_a\) is formulated in 2 ways which are:

1. The First Order Formulation where the performance index \(J\) is augmented with first order ordinary differential equation state constraints (refer to equation (2.9)). This formulation is used in the continuous time simulation (numerically integrated using the Runge-Kutta method).

2. The Second Order Formulation where the performance index \(J\) is augmented with second order ordinary differential equation state constraints. This formulation is used in the discrete time simulation (using Finite Element methods).

The First Order Formulation

From equations (2.9) and (3.25) the augmented performance index \(J_a\) can be written as
\[ J_a(x, \dot{x}, u, \lambda) = \int_{t_0}^{t_f} \left[ 1 + \lambda_1^1 [x_2 - \dot{x}_1] + \frac{1}{2} [x_1^2 + \frac{F_{\max}}{m} u_1 - \dot{x}_2] \right. \]

\[ + \lambda_3^1 [x_4 - \dot{x}_3] + \lambda_4^1 \left[ -\frac{2x_2x_4}{x_1} + \frac{T_{\max}}{x_1^2} u_2 - \dot{x}_4 \right] \right] dt \quad (3.26) \]

where the superscript 1 on the \( \lambda \)'s signifies first order formulation and

\[ \lambda_1, \lambda_2, \lambda_3, \lambda_4 \]

where \( \lambda_1, \lambda_2, \lambda_3, \lambda_4 \) are the Lagrangian multipliers which are explicit functions of \( t \).

From equations (2.14) and (3.25) the Hamiltonian for the \( r \)-theta manipulator can be defined as

\[ H(x, u, \lambda) = 1 + \lambda_1^1 x_2 + \lambda_2^1 \left[ \frac{x_1^2}{x_1^2} + \frac{F_{\max}}{m} u_1 \right] + \lambda_3^1 x_4 \]

\[ + \lambda_4^1 \left[ -\frac{2x_2x_4}{x_1} + \frac{T_{\max}}{x_1^2} u_2 \right]. \quad (3.27) \]

Substituting equation (3.27) in equation (2.17) gives the co-state equations for the \( r \)-theta manipulator as

\[ \lambda_1 = -\frac{1}{\lambda_2} x_2^2 - \lambda_4^1 \left[ \frac{2x_2x_4}{x_1^2} - \frac{2T_{\max}}{x_1^3} u_2 \right], \quad (3.28) \]

\[ \lambda_2 = -\lambda_1 - \lambda_4^1 \left[ \frac{-2x_4}{x_1} \right], \quad (3.29) \]

\[ \lambda_3 = 0, \quad (3.30) \]
Substituting equation (3.27) into equation (2.21) gives the transversality condition for the r-theta manipulator as

\[ 0 = 1 + \lambda_1 x_2 + \lambda_2 \left[ x_1 x_4 + \frac{F_{\text{max}}}{m} u_1 \right] + \lambda_3 x_4 + \lambda_4 \left[ \frac{-2x_2 x_4}{x_1} + \frac{T_{\text{max}}}{m x_1^2} u_2 \right] \]  

(3.32)

at \( t = t_f \).

For the simulation example it is assumed that the manipulator starts from rest and come to a stop at the final state. Therefore, the state variables

\[ x_2(t_f) = 0 \]  

(3.33)

\[ x_4(t_f) = 0 . \]  

(3.34)

Substituting equations (3.33) and (3.34) into equation (3.32) gives

\[ 1 + \lambda_2 \left[ \frac{F_{\text{max}}}{m} u_1 \right] + \lambda_4 \left[ \frac{T_{\text{max}}}{m x_1^2} u_2 \right] = 0 . \]  

(3.35)

The Second Order Formulation

The augmented functional \( J_a \) is defined in terms of the second order differential equation constraints (3.22) and (3.23) as
\[ J_a(x, \dot{x}, u, \lambda, t) = \int_{t_0}^{t_f} [1 + \lambda_1 [F_{\text{max}} u_1 - m\ddot{r} + m\dot{\theta}^2] + \lambda_2 [T_{\text{max}} u_2 - \frac{d}{dt}(m r^2 \dot{\theta})] \, dt \]  

(3.36)

where the \( \lambda \)'s with no superscripts signify the second order formulation.

Taking the variation \( J_a \) and applying the fundamental theorem gives the two multiplier equations and the transversality equation (refer to Appendix I). The multiplier equations are the second order differential equations in the Lagrangian multipliers. They are given by the following relations

\[-m \dddot{\lambda}_1 + m\lambda_1 \dot{\theta}^2 + \lambda_2 2m r \dot{\theta} = 0 \]  

(3.37)

and

\[2m[\lambda_1 (\dddot{r} + r \dddot{\dot{\theta}}) + \lambda_1 r \dot{\theta}] + \lambda_2 m r^2 + \lambda_2 2m r \dot{\dot{r}} = 0.\]  

(3.38)

When the velocities at the final state are zero the transversality equation is

\[1 + \lambda_1 [F_{\text{max}} u_1 ] + \lambda_2 [T_{\text{max}} u_2 ] = 0 \]  

(3.39)

which is equivalent to the first order formulation (equation (3.35)).

The Optimality Conditions for a Problem with Inequality Constraints

Figure 3.3 [2] provides one-dimensional illustrations of two possible types of minima with inequality constraints. It is required to minimize a functional \( I(u) \) subject to the inequality constraints

\[ f(u) \leq 0 \]  

(3.40)
Figure 3.3. One-dimensional illustration of two possible types of minima with inequality constraints.
where in general $f$ and $u$ are vectors of different dimensions. The constraints can be appended to the functional $I(u)$ to give the augmented functional $I_a$

$$I_a (u, \mu) = I(u) + \mu f(u)$$

(3.41)

where $\mu$ is the vector of the inequality constraint multipliers. When these constraints are satisfied

$$I_a (u, \mu) = I(u) .$$

(3.42)

There are two cases for the optimal value of $u$, $u^*$, which are

$$f(u^*) < 0$$

(3.43)

and

$$f(u^*) = 0 .$$

(3.44)

In the former case $\mu = 0$ so that equation (3.42) is satisfied. In the latter case consider small perturbations about $u^*$. If $I(u^*)$ is a minimum, then

$$\delta I = \frac{\partial I}{\partial u} \delta u \geq 0$$

(3.45)

for all admissible values of $\delta u$, which must also satisfy

$$\delta f = \frac{\partial f}{\partial u} \delta u \leq 0 .$$

(3.46)

For equations (3.45) and (3.46) to be true they must be of opposite sign which indicates

$$\text{sgn} \left( \frac{\partial I}{\partial u} \right) = -\text{sgn} \left( \frac{\partial f}{\partial u} \right)$$

(3.47)

or

$$\frac{\partial I}{\partial u} = 0$$

(3.48)

where the signum function, sgn, is the sign of the argument and
where \( \frac{\partial I}{\partial u} \) is the negative linear combination of \( \frac{\partial f_1}{\partial u} \), \( \frac{\partial f_2}{\partial u} \), ..., \( \frac{\partial f_m}{\partial u} \)

where \( m \) is the number of constraints. Equations (3.47) and (3.48) can be combined to give

\[
\frac{\partial I}{\partial u} + \mu \frac{\partial f}{\partial u} = 0 \quad (3.49)
\]

for \( \mu \geq 0 \). \( (3.50) \)

Therefore the necessary conditions for minimizing \( I(u) \) are

\[
\frac{\partial I}{\partial u} = 0 \quad (3.51)
\]

and

\[
f(u) \leq 0 \quad (3.52)
\]

subject to the conditions

\[
\mu \geq 0 \quad (3.53.1)
\]

for

\[
f(u) = 0 \quad (3.53.2)
\]

and

\[
\mu = 0 \quad (3.53.3)
\]

for

\[
f(u) < 0. \quad (3.53.4)
\]

For minimum time problems bang-bang control is used, that is

\[
u = \pm u_{\text{max}} \quad (3.54)
\]

where \( u \) is the control vector and \( u_{\text{max}} \) is the maximum control magnitude.

For the \( r \)-theta manipulator

\[
u_{\text{max}} = 1. \quad (3.55)
\]

From equations (3.52),(3.54) and (3.55) the inequality control constraints for the \( r \)-theta manipulator are

\[
f_1(u_1) = u_1 - 1, \quad (3.56.1)
\]
\[ f_2(u_1) = -u_1 - 1 , \quad (3.56.2) \]
\[ f_3(u_2) = u_2 - 1 , \quad (3.56.3) \]

and
\[ f_4(u_2) = -u_2 - 1 . \quad (3.56.4) \]

The functional \( I \) for the first order formulation is equal to \( H \) in equation (3.27)

\[
I(x,u,\lambda) = 1 + \lambda_1 x_2 + \lambda_2 \left[ x_1^2 + \frac{F_{\text{max}}}{m} u_1 \right] + \lambda_3 x_4
\]
\[ + \frac{1}{\lambda_4} \left[ -\frac{2x_2^2}{x_1} + \frac{T_{\text{max}}}{m} u_1 \right]. \quad (3.57) \]

From equations (3.41) and (3.57) \( I_a \) is

\[
I_a(x,u,\lambda,\mu,t) = I + \mu_1 (u_1 - 1) + \mu_2 (-u_1 - 1)
\]
\[ + \mu_3 (u_2 - 1) + \mu_4 (-u_2 - 1) . \quad (3.58) \]

Substituting equation (3.58) into equation (3.51) gives

\[
\frac{\partial I_a}{\partial u_1} = \frac{1}{\lambda_2} \frac{F_{\text{max}}}{m} + \mu_1 - \mu_2 = 0 \quad (3.59)
\]

and

\[
\frac{\partial I_a}{\partial u_2} = \frac{1}{\lambda_4} \frac{T_{\text{max}}}{m} \frac{2}{x_1^2} + \mu_3 - \mu_4 = 0 \quad (3.60)
\]

subject to the conditions (3.52), and (3.53). From equation (3.59)

if
\[ u_1 = 1 \quad (3.61.1) \]

then
\[ \mu_1 \geq 0 \quad (3.61.2) \]
and \[ \mu_2 = 0 . \] (3.61.3)

Therefore \[ \lambda_2 \leq 0 \] (3.61.4)
in order for equation (3.59) to be true. If \[ u_1 = -1 \] (3.62.1)
then \[ \mu_1 = 0 \] (3.62.2)
and \[ \mu_2 \geq 0 . \] (3.62.3)

Therefore \[ \lambda_2 \geq 0 \] (3.62.4)
in order for equation (3.59) to be true. From equations (3.61) and (3.62) the control \( u_1 \) can be defined as

\[ u_1 = - \frac{1}{|\lambda_2|} \frac{1}{\lambda_2} = -\text{sgn}(\lambda_2) \] (3.63)

for \[ \lambda_2 \neq 0 . \] (3.64)
The signum function, \( \text{sgn} \), takes on a value which is equal to the sign of the argument. Similarly from equation (3.60) the control \( u_2 \) is defined as

\[ u_2 = - \frac{1}{|\lambda_4|} \frac{1}{\lambda_4} = -\text{sgn}(\lambda_4) \] (3.65)
When $\lambda_2$ or $\lambda_4$ is zero the respective control can move away from the constraint boundary. However for the $r$-theta manipulator $\lambda_2$ and $\lambda_4$ are at zero for only an instant and therefore the effect of control at those instances is not significant.

For the second order formulation the control $u_1$ is defined as

$$u_1 = -\frac{|\lambda_1|}{\lambda_1} = -\text{sgn}(\lambda_1)$$  \hspace{1cm} (3.67)

for $\lambda_1 \neq 0$ \hspace{1cm} (3.68)

and the control $u_2$ is defined as

$$u_2 = -\frac{|\lambda_2|}{\lambda_2} = -\text{sgn}(\lambda_2)$$  \hspace{1cm} (3.69)

for $\lambda_2 \neq 0$. \hspace{1cm} (3.70)

These optimality conditions are used in the continuous time and discrete time simulations.
CHAPTER 4

THE FINITE ELEMENT METHOD

Introduction

In Chapter 3 the mathematical model of the manipulator was formulated. The state equations, the multiplier equations, the transversality equation, and the optimality conditions were derived for the r-theta manipulator by both the first and the second order formulations. The first order formulation is used in the continuous time simulation where the set of eight first order differential equations are numerically integrated using the Runge-Kutta method. The guesses on the unknown parameters (the Lagrangian multipliers at the initial time $t_0$ and the final time $t_f$) are iterated upon using conventional minimization techniques like the conjugate gradient method [15], the Quasi-Newton method [16], and the Newton-Raphson method [17]. However none of the methods converge to the solution if the initial guesses on the unknown parameters are not sufficiently close to the optimal values.

A more robust method of solving the minimum time problem, a two point boundary value problem (TPBVP), is needed. Both the optimal control problem and the finite element method can be developed from variational principles. The finite element method has been successfully applied to a wide range of nonlinear problems as well as to the TPBVP.
Therefore, this method was applied to the solution of the minimum time problem with the objective of investigating the advantages and drawbacks of a discrete time simulation as compared to the continuous case. A flow chart of the finite element program is included in Appendix II.

**Finite Element Analysis**

The finite element method is a numerical analysis technique for obtaining approximate solutions to a wide range of engineering problems. In nonlinear problems, as in this particular case, closed form solutions are not available, so it is necessary to obtain approximate numerical solutions. Two of the more commonly used methods are the finite difference and the finite element methods [18]. For some problems, especially problems with irregular geometry or unusual boundary conditions, the finite element method is superior to the finite difference method.

The finite element method takes a continuum problem and discretizes the solution region into a finite number of elements. By expressing the unknown solution within each element in terms of assumed approximating functions called interpolation functions, the infinite number of unknowns in terms of the Taylor series expansion is reduced to a finite number.

One of the advantages of this method is the ability to formulate the properties of the individual elements, before putting them together to represent the entire problem. In effect, a complex problem is reduced to considering a series of greatly simplified problems. Another advantage is the variety of ways in which the problem can be formulated.
These include the direct approach (from physical laws), the variational approach, the weighted residual approach, and the energy balance approach [18]. In this paper the variational approach was used in determining the element properties.

There are 5 basic steps to the finite element method. These are:

1. Discretization of the continuum.
2. Selection of the interpolation function.
3. Determination of the element properties.
4. Assembly of element properties to obtain system equations.
5. Application of boundary conditions and solution of system unknowns.

Discretization of the Continuum

The first step is the discretization of the solution region into elements. The range of the independent variable, time $t$, from the initial state to the final state is discretized into elements of uniform length $\Delta t$. These elements are connected to adjoining elements by sharing common nodes. The element length, $\Delta t$, varies with change in grid density or the final time. The element unknowns are the position coordinates $(r, \theta)$ and the Lagrangian multipliers $(\lambda_1, \lambda_2)$ at each node and the length of element ($\Delta t$). The performance index $J_a$ will be expressed in terms of the approximations so that the continuous time
problem of minimizing $J_a$ over the time interval $[t_o, t_f]$ is reduced to one of minimizing $J_a$ for each element in the time domain.

Discretizing equation (3.35) gives

$$J_a = \sum_{i=1}^{n} \int_{(i-1)\Delta t}^{i\Delta t} \left( 1 + \lambda_1 [F_{\text{max}} u_1 + mr\dot{\theta} - m\ddot{\theta}] \right)$$
$$+ \lambda_2 [T_{\text{max}} u_2 - \frac{d}{dt} (mr^2 \dot{\theta})] \, dt$$

(4.1)

where $n$ is the number of elements and $\Delta t$ is the length of each element and $\sum_{i=1}^{n}$ is the summation over the elements 1 to $n$.

Selection of the Interpolation Function

The next step is to assign nodes to each element (points in time) and choose the interpolation function to represent the variation of the unknown variables over the elements. The state variables $r$, $\dot{r}$, $\Theta$, $\dot{\Theta}$ and the multipliers and their time derivatives $\lambda_1$, $\lambda_2$, $\dot{\lambda}_1$, $\dot{\lambda}_2$ will be represented by linear interpolation functions of the form

$$x(t) = N_1(t)x_1 + N_2(t)x_2,$$

(4.2)

and

$$\dot{x}(t) = \frac{x_2 - x_1}{\Delta t}$$

(4.3)

where $x_1$, $x_2$ are the values of the given unknowns $x(t)$ at nodes 1 and 2 of each element and the natural coordinates $N_1$, $N_2$ vary as shown in Figure 4.1
Figure 4.1: Natural Coordinates for First Order, One Dimensional Finite Element
The natural coordinates are defined by the relations

\[ N_1 = 1 - \frac{t}{\Delta t} \]  
\[ N_2 = \frac{t}{\Delta t} \]  

and

\[ f^{\Delta t}(N_1) \, dt = f^{\Delta t}(N_2) \, dt = \frac{\Delta t}{2} \]  
\[ f^{\Delta t}(N_1 N_2) \, dt = \frac{\Delta t}{6} \]  

and

\[ f^{\Delta t}(N_1^2) \, dt = f^{\Delta t}(N_2^2) \, dt = \frac{\Delta t}{3} \]  

The relations (4.6), (4.7) and (4.8) will be extensively used in the development of the element equations.

From equations (4.2) and (4.3) the unknown variables are defined as

\[ r(t) = N_1 r_1 + N_2 r_2 \]  
\[ \Theta(t) = N_1 \Theta_1 + N_2 \Theta_2 \]  
\[ \lambda_1(t) = N_1 \lambda_{11} + N_2 \lambda_{12} \]  
\[ \lambda_2(t) = N_1 \lambda_{21} + N_2 \lambda_{22} \]  
\[ \dot{r}(t) = \frac{r_2 - r_1}{\Delta t} \]  
\[ \dot{\Theta}(t) = \frac{\Theta_2 - \Theta_1}{\Delta t} \]  
\[ \dot{\lambda}_1(t) = \frac{\lambda_{12} - \lambda_{11}}{\Delta t} \]
and

$$\lambda_2(t) = \frac{\lambda_{22} - \lambda_{12}}{\Delta t}$$  \hspace{1cm} (4.9.8)

where \( \lambda_{11} \) and \( \lambda_{12} \) are the values of \( \lambda_1(t) \) at nodes 1 and 2, respectively, of each element while \( \lambda_{21} \) and \( \lambda_{22} \) are the values of \( \lambda_2(t) \) at the respective node designated by the second subscript.

Determination of the Element Properties

The variational approach will be used in the formulation of the element properties. The equations for the minimization of the performance index \( J_a \) over a single element will be derived in this section. In the next section the elements will be assembled to give the equations for the minimization of \( J_a \) over the entire time period \([t_0, t_f]\).

\( J_a \) in equation (4.1) is simplified by integrating by parts the second derivative terms in the equation. This gives rise to two boundary terms. For interior elements the boundary terms cancel with those from the adjoining elements. For exterior elements these terms go to zero because \( \dot{r}(0), \dot{\theta}(0), \dot{r}(t_f) \) and \( \dot{\theta}(t_f) \) are specified as zero in the boundary conditions. Considering equation (4.1) for a single element and substituting equations (4.6) – (4.9) into (4.1) gives
\[ J_a = \Delta t + \frac{m}{6\Delta t} \begin{bmatrix} \lambda_{11} & \lambda_{12} \\ \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 1 & 2 \\ \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ \end{bmatrix} \begin{bmatrix} \theta_1 & \theta_2 \\ -1 & 1 \\ \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \end{bmatrix} \] 

\[ + \frac{m}{6\Delta t} \begin{bmatrix} r_1 & r_2 \\ 1 & 2 \\ \end{bmatrix} \begin{bmatrix} \lambda_{21} & \lambda_{22} \\ -1 & 1 \\ \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \end{bmatrix} \] 

\[ + \frac{m}{\Delta t} \begin{bmatrix} \lambda_{11} & \lambda_{12} \\ -1 & 1 \\ \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ \end{bmatrix} \] 

\[ + \int_0^{\Delta t} (\lambda_1 F_{\text{max}} u_1 + \lambda_2 T_{\text{max}} u_2) \, dt \] 

(4.10)

For terms containing the controls two situations will be considered.

**No switching of controls**

If no switch (change of sign) occurs within the element, then

\[ \text{sgn}(\lambda_{11}) = \text{sgn}(\lambda_{12}) \] 

(4.11.1)

and

\[ \text{sgn}(\lambda_{21}) = \text{sgn}(\lambda_{22}) \] 

(4.11.2)

are true. Substituting equations (3.63), (4.6) and (4.9.3) into the first integral term in equation (4.10) gives

\[ \int_0^{\Delta t} (\lambda_1 F_{\text{max}} u_1) \, dt = - \frac{F_{\text{max}}}{2} \Delta t \left( |\lambda_{11}| + |\lambda_{12}| \right) . \] 

(4.13)

Similarly the second integral term in equation (4.10) is found to be

\[ \int_0^{\Delta t} (\lambda_2 T_{\text{max}} u_2) \, dt = - \frac{T_{\text{max}}}{2} \Delta t \left( |\lambda_{21}| + |\lambda_{22}| \right) . \] 

(4.14)
Switching of controls

If a switch in $u_1$ occurs within the element then

$$\text{sgn}(\lambda_{11}) = - \text{sgn}(\lambda_{12}) .$$  \hspace{1cm} (4.15)

Let $t_s$ be the time to switch from the start of the element in which the switching occurs. Then

$$\lambda_1(t_s) = 0$$  \hspace{1cm} (4.16)

or

$$N_1(t_s)\lambda_{11} + N_2(t_s)\lambda_{12} = 0$$  \hspace{1cm} (4.17)

or

$$\frac{t_s}{\Delta t} \lambda_{11} + \frac{t_s}{\Delta t} \lambda_{12} = 0 .$$  \hspace{1cm} (4.18)

Rearranging the above equation yields

$$t_s = \frac{\Delta t \lambda_{11}}{\lambda_{11} - \lambda_{12}} .$$  \hspace{1cm} (4.19)

Therefore, for a switch

$$\int_0^{\Delta t} \{ \lambda_{1Fmax} u_1 \} dt = - \int_0^{t_s} \{ \lambda_{1Fmax} \text{sgn}(\lambda_{11}) \} dt$$

$$- \int_{t_s}^{\Delta t} \{ \lambda_{1Fmax} \text{sgn}(\lambda_{12}) \} dt .$$  \hspace{1cm} (4.20)

Substituting equation (4.19) into (4.20) above and simplifying gives

$$\int_0^{\Delta t} \{ \lambda_{1Fmax} u_1 \} dt = - \frac{\text{Fmax}}{2} \text{sgn}(\lambda_{11}) \Delta t \frac{\lambda_{11}^2 + \lambda_{12}^2}{(\lambda_{11} - \lambda_{12})} .$$  \hspace{1cm} (4.21)

For switch in $u_2$ within the element

$$\text{sgn}(\lambda_{21}) = - \text{sgn}(\lambda_{22}) .$$  \hspace{1cm} (4.22)
Following the same steps as in the case of $u_1$ switch gives

$$t_s = \frac{\Delta t \lambda_{21}}{\lambda_{21} - \lambda_{22}}$$  \hspace{1cm} (4.23)

and

$$\int_0^{\Delta t} [\lambda_2 T_{\text{max}} u_2] \, dt = -\frac{T_{\text{max}}}{2} \operatorname{sgn}(\lambda_{21}) \Delta t \left(\frac{\lambda_{21}^2 + \lambda_{22}^2}{\lambda_{21} - \lambda_{22}}\right). \hspace{1cm} (4.24)$$

**Element equations from a Variational principle**

The finite element solution to the problem involves picking the values of $\phi_i$ (consisting of $r$, $\Theta$, $\lambda_1$, $\lambda_2$) where $i$ goes from 1 to $p$ and $p$ is equal to four times the number of nodes, and element length $\Delta t$, so as to make the functional $J_a(\phi, \Delta t)$ stationary. To make $J_a(\phi, \Delta t)$ stationary with respect to $\phi_i$ and $\Delta t$ the fundamental theorem of variational calculus requires that

$$\delta J_a(\phi, \Delta t) = \frac{\partial J_a}{\partial \phi_i} \delta \phi_i + \sum_{i=1}^{p} \frac{\partial J_a}{\partial \Delta t} \delta \Delta t = 0. \hspace{1cm} (4.25)$$

Since the $\delta \phi_i$'s and $\Delta t$ are independent, equation (4.25) can hold only if

$$\frac{\partial J_a}{\partial \phi_i} = 0 \hspace{1cm} (4.26.1)$$

and

$$\frac{\partial J_a}{\partial \Delta t} = 0. \hspace{1cm} (4.26.2)$$

Therefore, $J_a$ in equation (4.10) is differentiated with respect to the nodal unknowns, $r_1$, $r_2$, $\Theta_1$, $\Theta_2$, $\lambda_{11}$, $\lambda_{12}$, $\lambda_{21}$, $\lambda_{22}$, and with respect to
time $\Delta t$, to get the element equations. Differentiating $J_a$ with respect to $r_1$, $r_2$ gives

$$\frac{\partial J_a}{\partial \begin{bmatrix} r_1 \\ r_2 \end{bmatrix}} = \frac{m}{6\Delta t} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} + \frac{m}{3\Delta t} \begin{bmatrix} \lambda_{21} & \lambda_{22} \end{bmatrix} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} + \frac{m}{\Delta t} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \lambda_{11} \\ \lambda_{12} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$ \hfill (4.27)

Let

$$cm_{13} = \frac{m}{6\Delta t} (\theta_1 - \theta_2)^2,$$ \hfill (4.28)

$$cm_{11} = \frac{m}{3\Delta t} (\theta_1 - \theta_2)(\lambda_{21} - \lambda_{22}),$$ \hfill (4.29)

and

$$cm_{31} = \frac{m}{\Delta t}.$$ \hfill (4.30)

Substituting equations (4.28), (4.29), (4.30) into (4.27) gives
\[ \frac{\partial J_a}{\partial r_1} = \text{cm}13 \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} \lambda_{11} \\ \lambda_{12} \end{bmatrix} + \text{cm}11 \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} \]
\[ + \text{cm}31 \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \lambda_{11} \\ \lambda_{12} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \]

(4.31)

Let
\[ \text{cm}22 = \frac{m}{3\Delta t} (\lambda_{11} [2r_1 + r_2] + \lambda_{12} [r_1 + 2r_2]) , \]  

(4.32)

and
\[ \text{cm}24 = \frac{m}{3\Delta t} (r_1^2 + r_1 r_2 + r_2^2) . \]  

(4.33)

Differentiating \( J_a \) with respect to \( \theta_1 \) and \( \theta_2 \) plus substituting \( \text{cm}22 \) and \( \text{cm}24 \) into the expression gives

\[ \frac{\partial J_a}{\partial \theta_1} = \text{cm}22 \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} + \text{cm}24 \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \lambda_{21} \\ \lambda_{22} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} . \]  

(4.34)

Differentiating \( J_a \) with respect to \( \lambda_{11} \) and \( \lambda_{12} \) and substituting \( \text{cm}13 \) and \( \text{cm}31 \) into the expression gives
For the last term of equation (4.35) there exist two situations depending upon the occurrence of a switch in \( \lambda \). If a switch does not occur then \( \lambda \) does not change sign. Differentiating equation (4.13) with respect to \( \lambda \), \( \lambda \) gives

\[
\frac{\partial}{\partial \lambda} \int_0^{\lambda_{12}} \lambda_{11} \, d\lambda = \int_0^{\lambda_{12}} \lambda_{11} \, d\lambda = \lambda_{12} \lambda_{11}
\]

For the last term of equation (4.35) there exist two situations depending upon the occurrence of a switch in \( \lambda \). If a switch does not occur then \( \lambda \) does not change sign. Differentiating equation (4.13) with respect to \( \lambda \), \( \lambda \) gives

\[
\frac{\partial}{\partial \lambda} \int_0^{\lambda_{12}} \lambda_{11} \, d\lambda = \int_0^{\lambda_{12}} \lambda_{11} \, d\lambda = \lambda_{12} \lambda_{11}
\]

If a switch occurs then \( \lambda \) will pass through zero. Differentiating equation (4.13) with respect to \( \lambda \) and \( \lambda \) gives

\[
\frac{\partial}{\partial \lambda} \int_0^{\lambda_{12}} \lambda_{11} \, d\lambda = \int_0^{\lambda_{12}} \lambda_{11} \, d\lambda = \lambda_{12} \lambda_{11}
\]
and
\[
\frac{\partial}{\partial \lambda_{12}} \int_0^{\Delta t} \left( \lambda_{12} \max u \right) dt = -\frac{\max}{2} \Delta t \quad \text{sgn}(\lambda_{12})
\]
\[
\frac{\lambda_{12}}{\lambda_{11} - \lambda_{12}} \left[ (\lambda_{11}^2 + \lambda_{12}^2) + \frac{\lambda_{12}^2}{2(\lambda_{11} - \lambda_{12})^2} \right].
\]

Differentiating \( J_a \) with respect to \( \lambda_{21}, \lambda_{22} \) and substituting \( \text{cm24} \) into the expression gives

\[
\begin{align*}
\frac{\partial J_a}{\partial \lambda_{21}} &= \text{cm24} \begin{bmatrix} 1 & -1 & \theta_1 \\ -1 & 1 & \theta_2 \end{bmatrix} \\
+ \frac{\partial}{\partial \lambda_{21}} \int_0^{\Delta t} \left( \lambda_{21} \max u \right) dt &= \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \quad (4.33)
\end{align*}
\]

For the last term in equation (4.33), there exist two situations depending upon the occurrence of a switch in \( \lambda_2 \). For the situation of no switch, differentiating equation (4.14) with respect to \( \lambda_{21} \) and \( \lambda_{22} \) gives

\[
\frac{\partial}{\partial \lambda_{21}} \int_0^{\Delta t} \left( \lambda_{21} \max u \right) dt = -\max \frac{\Delta t}{2} \quad \text{sgn}(\lambda_{21}) \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \quad (4.39)
\]

For the situation where a switch occurs, differentiating equation (4.24) with respect to \( \lambda_{21} \) and \( \lambda_{22} \) gives
\[
\frac{\partial}{\partial \lambda_{21}} \int_0^{\Delta t} (\lambda_2 T_{\text{max}} u_2) \, dt = - \frac{T_{\text{max}}}{2} \Delta t \, \text{sgn}(\lambda_{21})
\]

\[
\frac{\lambda_{21}}{\lambda_{21} - \lambda_{22}} - \frac{(\lambda_{21}^2 + \lambda_{22}^2)}{2(\lambda_{21} - \lambda_{22})^2}
\]

(4.40.1)

and

\[
\frac{\partial}{\partial \lambda_{22}} \int_0^{\Delta t} (\lambda_2 T_{\text{max}} u_2) \, dt = - \frac{T_{\text{max}}}{2} \Delta t \, \text{sgn}(\lambda_{21})
\]

\[
\frac{\lambda_{22}}{\lambda_{21} - \lambda_{22}} + \frac{(\lambda_{21}^2 + \lambda_{22}^2)}{2(\lambda_{21} - \lambda_{22})^2}
\]

(4.40.2)

Derivation of the transversality equation.

Differentiating \( J_a \) in equation (4.10) with respect to time \( \Delta t \) and simplifying gives

\[
\frac{\partial J_a}{\partial \Delta t} = 1 - \frac{1}{\Delta t} \left[ \frac{c m 22}{m} \frac{3 \Delta t}{m} + \frac{cm 24}{m^2} \frac{3 \Delta t}{m} + \frac{cm 11 + cm 31 (r_1 - r_2)(\lambda_{11} - \lambda_{12})}{m} \right]
\]

\[
+ \frac{\partial}{\partial \Delta t} \int_0^{\Delta t} (\lambda_1 F_{\text{max}} u_1 + \lambda_2 T_{\text{max}} u_2) \, dt.
\]

(4.41)

There are two situations for the integral terms. If no switch occurs, then

\[
\frac{\partial}{\partial \Delta t} \int_0^{\Delta t} (\lambda_1 F_{\text{max}} u_1) \, dt = - \frac{F_{\text{max}}}{2} (| \lambda_{11} | + | \lambda_{12} |)
\]

(4.42)

and

\[
\frac{\partial}{\partial \Delta t} \int_0^{\Delta t} (\lambda_2 T_{\text{max}} u_2) \, dt = - \frac{T_{\text{max}}}{2} (| \lambda_{21} | + | \lambda_{22} |).
\]

(4.43)

If a switch on \( u_1 \) occurs, then
\[
\frac{\partial}{\partial \Delta t} \int_0^{\Delta t} \left( \lambda_1 F_{\text{max}} u_1 \right) \, dt = -\frac{F_{\text{max}}}{2} \frac{(\lambda_{11}^2 + \lambda_{12}^2)}{\lambda_{11}^2 - \lambda_{12}^2} \cdot \text{(4.44)}
\]

If a switch on \( u_2 \) occurs, then

\[
\frac{\partial}{\partial \Delta t} \int_0^{\Delta t} \left( \lambda_2 T_{\text{max}} u_2 \right) \, dt = -\frac{T_{\text{max}}}{2} \frac{(\lambda_{21}^2 + \lambda_{22}^2)}{(\lambda_{21}^2 - \lambda_{22}^2)} \cdot \text{(4.45)}
\]

The nine element equations derived above, (4.31), (4.34), (4.35), (4.38) and (4.41) are assembled to give the 9 x 9 element matrix \([A]\) as shown in Figure 4.2. The nodal unknowns, \( r_1, r_2, \theta_1, \theta_2, \lambda_{11}, \lambda_{12}, \lambda_{21}, \lambda_{22}, \) and \( \Delta t \) constitute the vector \( x \). The element submatrices, \( M(i,j) \), are two by two matrices for values of \( i \) and \( j \) ranging from one to four. The submatrices \( M(i,5) \) are two by one vectors while the submatrices \( M(5,i) \) are one by two row vectors. The submatrix \( M(5,5) \) is a scalar.

The element equation can be stated as

\[
[A] x = f(x) = 0 . \text{(4.46)}
\]

To solve the system of nonlinear equations the Newton-Raphson method will be used. The nonlinear equations are linearized by using a Taylor series expansion about the true solution \( f(x^*) \). Guesses on the unknown variables can be expressed as

\[
x_i = x^* + \Delta x_i \text{,} \quad i = 1, \ldots, n \text{.} \text{(4.47)}
\]

where \( i \) is the iteration number and \( \Delta x_i \) is the vector of deviations from the solution vector \( x^* \).
\[
\begin{array}{c|c|c|c|c|c}
M(1,1) & M(1,2) & M(1,3) & M(1,4) & M(1,5) & R_1 \\
M(2,1) & M(2,2) & M(2,3) & M(2,4) & M(2,5) & R_2 \\
M(3,1) & M(3,2) & M(3,3) & M(3,4) & M(3,5) & \theta_1 \\
M(4,1) & M(4,2) & M(4,3) & M(4,4) & M(4,5) & \theta_2 \\
M(5,1) & M(5,2) & M(5,3) & M(5,4) & M(5,5) & L_{11} \\
& & & & & L_{12} \\
& & & & & L_{21} \\
& & & & & L_{22} \\
& & & & & DT \\
\end{array}
\]

\[ AX = \Theta \]

**Figure 4.2: Element Matrix Equation**
Considering the first two terms of the Taylor series expansion gives

\[
f(x_1 + \Delta x_1, x_2 + \Delta x_2, \ldots, x_p + \Delta x_p) = f(x_1^*, x_2^*, \ldots, x_p^*) + \frac{\partial f}{\partial x_1} \Delta x_1 + \frac{\partial f}{\partial x_2} \Delta x_2 + \ldots + \frac{\partial f}{\partial x_p} \Delta x_p
\]

(4.48)

where \( f \) is a vector of order \( p \) for a system of \( p \) equations. From equation (4.46)

\[
f(x) = 0
\]

(4.49)

Therefore, equation (4.48) reduces to

\[
f(x) = \frac{\partial f}{\partial x} \Delta x
\]

(4.50)

The Jacobian \([ J ]\) is defined as

\[
[J] = \frac{\partial f}{\partial x} = \begin{bmatrix}
\frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \ldots & \frac{\partial f_1}{\partial x_p} \\
\frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \ldots & \frac{\partial f_2}{\partial x_p} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial f_p}{\partial x_1} & \frac{\partial f_p}{\partial x_2} & \ldots & \frac{\partial f_p}{\partial x_p}
\end{bmatrix}
\]

(4.51)

Therefore equation (4.50) can be written as

\[
f(x) = [ J ] \Delta x
\]

(4.52)
Equation (4.50) is solved iteratively and the guesses are updated each iteration until the convergence criterion

\[ \text{CONV} = \left[ \sum_{i=1}^{n} \frac{(\Delta x_i)^2}{x_i^2} \right]^{0.5} < 1.0 \times 10^{-10} \]  

(4.53)

is satisfied.

Evaluation of the Jacobian

The Jacobian is a 9 x 9 matrix that is defined by several sub-matrices as shown in Figure 4.3. These sub-matrices are presented in the following development.

From equations (4.46) and (4.51)

\[ J(1,1) = \frac{\partial^2 J}{\partial x_1 \partial x_2} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \]  

(4.54)

and

\[ J(1,2) = \frac{\partial^2 J}{\partial x_1 \partial \theta_2} = \begin{bmatrix} (1,3) & -(1,3) \\ (2,3) & -(2,3) \end{bmatrix} \]  

(4.55.1)

where

\[ c_{j12} = \frac{m}{3 \Delta t} , \]  

(4.55.2)

\[ (1,3) = (\lambda_{21} - \lambda_{22})(2r_1 + r_2) + (\theta_1 - \theta_2)(2\lambda_{11} + \lambda_{12}) , \]  

(4.55.3)

and

\[ (2,3) = (\lambda_{21} - \lambda_{22})(r_1 + 2r_2) + (\theta_1 - \theta_2)(\lambda_{11} + 2\lambda_{12}) . \]  

(4.55.4)
\[ \begin{bmatrix}
J(1,1) & J(1,2) & J(1,3) & J(1,4) & J(1,5) \\
J(2,1) & J(2,2) & J(2,3) & J(2,4) & J(2,5) \\
J(3,1) & J(3,2) & J(3,3) & J(3,4) & J(3,5) \\
J(4,1) & J(4,2) & J(4,3) & J(4,4) & J(4,5) \\
J(5,1) & J(5,2) & J(5,3) & J(5,4) & J(5,5)
\end{bmatrix}
\begin{bmatrix}
\text{DR}_1 \\
\text{DR}_2 \\
\text{DH}_1 \\
\text{DH}_2 \\
\text{DL}_{11} \\
\text{DL}_{12} \\
\text{DL}_{21} \\
\text{DL}_{22} \\
\text{DDT}
\end{bmatrix} = 0
\]

**Figure 4.3 Element Jacobian Equation**
The \((1,3)\) and \((1,4)\) submatrices are

\[
J(1,3) = \frac{\partial^2 J_a}{\partial \lambda_{11} \partial r_1} = \text{cm13} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} + \text{cm31} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}
\] (4.56)

and

\[
J(1,4) = \frac{\partial^2 J_a}{\partial \lambda_{21} \partial r_1} = \text{cj14} \begin{bmatrix} 1 & 7 \\ 2 & 7 \end{bmatrix} (1,7) - (1,7)
\] (4.57.1)

where

\[
\text{cj14} = \frac{m}{3\Delta t} (\theta_1 - \theta_2)
\] (4.57.2)

(1,7) = \((2r_1 + r_2)\),

(4.57.3)

and

(2,7) = \((r_1 + 2r_2)\).

(4.57.4)

The \((1,5)\) submatrix is

\[
J(1,5) = \frac{\partial^2 J_a}{\partial \Delta t \partial r_1} = -\frac{1}{\Delta t} \left[ \text{cm11} \begin{bmatrix} 2r_1 + r_2 \\ r_1 + 2r_2 \end{bmatrix} + \text{cm31} \begin{bmatrix} 2\lambda_{11} + \lambda_{12} \\ \lambda_{11} + 2\lambda_{12} \end{bmatrix} + \text{cm31} \begin{bmatrix} \lambda_{11} - \lambda_{12} \\ -\lambda_{11} + \lambda_{12} \end{bmatrix} \right].
\] (4.58)

Since the Jacobian is symmetric we have the condition

\[
J(2,1) = \frac{\partial^2 J_a}{\partial r_1 \partial \theta_1} = \begin{bmatrix} J(1,2) \end{bmatrix}^T.
\] (4.59)
The other terms from the second row of submatrices in the Jacobian are

\[
J(2,2) = \frac{\partial^2 J_a}{\partial \theta_1^2} = cm22 \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}, \quad (4.60)
\]

\[
J(2,3) = \frac{\partial^2 J_a}{\partial \theta_1 \partial \lambda_{12}} = cj14 \begin{bmatrix} (1,7) & (2,7) \\ -(1,7) & -(2,7) \end{bmatrix}, \quad (4.61)
\]

\[
J(2,4) = \frac{\partial^2 J_a}{\partial \theta_2^2} = cm24 \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}, \quad (4.62)
\]

and \( J(2,5) = \frac{\partial^2 J_a}{\partial \theta_1 \partial \Delta t + \Delta t \partial \theta_2} = \frac{1}{\Delta t} \left[ cm22 \begin{bmatrix} \theta_1 - \theta_2 \\ -\theta_1 + \theta_2 \end{bmatrix} + cm24 \begin{bmatrix} \lambda_{21} - \lambda_{22} \\ -\lambda_{21} + \lambda_{22} \end{bmatrix} \right], \quad (4.63) \)

By symmetry we have

\[
J(3,1) = \frac{\partial^2 J_a}{\partial \lambda_{11}^2} = \left[ J(1,3) \right]^T, \quad (4.64)
\]
and \[ J(3,2) = \frac{\partial^2 J}{\partial \theta_1 \partial \theta_2} = [ J(2,3) ]^T. \] \[ (4.65) \]

There are two separate expressions for \( J(3,3) \) given by

\[
J(3,3) = \frac{\partial^2 J}{\partial \lambda_{11}^2}.
\] \[ (4.66.1) \]

For no switch on \( u_1 \) we have

\[
J(3,3) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.
\] \[ (4.66.2) \]

For switch on \( u_1 \) we get the result

\[
J(3,3) = c_{j33} \begin{bmatrix} 2 \lambda_{12}^2 & \lambda_{11} \lambda_{12} \\ \lambda_{11} \lambda_{12} & 2 \lambda_{11}^2 \end{bmatrix}
\] \[ (4.66.3) \]

where

\[
c_{j33} = -\frac{2 F_{\text{max}} \Delta t}{(\lambda_{11} - \lambda_{12})^3} \text{sgn}(\lambda_{11})
\] \[ (4.66.4) \]

The last two terms in row 3 are the submatrices
where the quantities (5,9) and (6,9) for no switch on \( u_1 \) are given by

\[
(5,9) = (2cm13 + cm31)r_1 + (cm13 - cm31)r_2 - \frac{F_{\text{max}}}{2} \Delta t \, \text{sgn}(\lambda_{11})
\]  

(4.68.2)

and

\[
(6,9) = (cm13 - cm31)r_1 + (2cm13 + cm31)r_2 - \frac{F_{\text{max}}}{2} \Delta t \, \text{sgn}(\lambda_{11})
\]  

(4.68.3)

while for switch on \( u_1 \) we get

\[
(5,9) = (2cm13 + cm31)r_1 + (cm13 - cm31)r_2 - \frac{F_{\text{max}}}{2} \Delta t
\]

\[
\text{sgn}(\lambda_{11})\left[\frac{\lambda_{11}}{(\lambda_{11} - \lambda_{12})} - \frac{(\lambda_{11}^2 + \lambda_{12}^2)}{2(\lambda_{11} - \lambda_{12})^2}\right]
\]

(4.68.4)

\[
(6,9) = (cm13 - cm31)r_1 + (2cm13 + cm31)r_2 - \frac{F_{\text{max}}}{2} \Delta t
\]

\[
\text{sgn}(\lambda_{11})\left[\frac{\lambda_{12}}{(\lambda_{11} - \lambda_{12})} + \frac{\lambda_{11}^2 + \lambda_{12}^2}{2(\lambda_{11} - \lambda_{12})^2}\right]
\]  

(4.68.5)
For the fourth row of submatrices we have by symmetry

\[
J(4,1) = \frac{\partial^2 J}{\partial r_1 \partial r_2} = [ J(1,4) ]^T ,
\]

(4.69)

\[
J(4,2) = \frac{\partial^2 J}{\partial \theta_1 \partial \lambda_{21}} = [ J(2,4) ]^T ,
\]

(4.70)

and

\[
J(4,3) = \frac{\partial^2 J}{\partial \lambda_{11} \partial \lambda_{22}} = [ J(3,4) ]^T .
\]

(4.71)

There are two separate expressions for \( J(4,4) \) given by

\[
J(4,4) = \frac{\partial^2 J}{\partial \lambda_{21} \partial \lambda_{22}} .
\]

(4.72.1)

For no switch on \( u_2 \) we have

\[
J(4,4) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} .
\]

(4.72.2)

For switch on \( u_2 \) we have the result
\[ J(4,4) = c_{j44} \begin{bmatrix} \lambda_{22} & -\lambda_{21}\lambda_{22} \\ -\lambda_{21}\lambda_{22} & \lambda_{21}^2 \end{bmatrix} \] (4.72.3)

where

\[ c_{j44} = -\frac{2}{\Delta t} \frac{T_{\text{max}}}{(\lambda_{21} - \lambda_{22})^3} \text{sgn}(\lambda_{21}) \] (4.72.4)

The last term in the fourth row of submatrices is

\[ J(4,5) = \frac{\partial^2 J_a}{\partial \Delta t \partial \lambda_{21}} = -\frac{1}{\Delta t} \begin{bmatrix} (7,9) \\ (8,9) \end{bmatrix} \] (4.73.1)

where if no switch on \( u_2 \) occurs, the quantities (7,9) and (8,9) are given by

\[ (7,9) = \text{cm}24 \Theta_1 - \text{cm}24 \Theta_2 - \frac{T_{\text{max}}}{2} \Delta t \text{sgn}(\lambda_{21}) \] (4.73.2)

and

\[ (8,9) = -\text{cm}24 \Theta_1 + \text{cm}24 \Theta_2 - \frac{T_{\text{max}}}{2} \Delta t \text{sgn}(\lambda_{21}) \] (4.73.3)

If a switch occurs on \( u_2 \) then (7,9) and (8,9) are given by

\[ (7,9) = \text{cm}24 \Theta_1 - \text{cm}24 \Theta_2 - \frac{T_{\text{max}}}{2} \Delta t \text{sgn}(\lambda_{21}) \]

\[ \begin{bmatrix} \frac{\lambda_{21}}{(\lambda_{21} - \lambda_{22})} & -\frac{\lambda_{21}^2 + \lambda_{22}^2}{2(\lambda_{21} - \lambda_{22})^2} \end{bmatrix} \] (4.74.4)
and

\[
(8,9) = - \text{cm}24 \: \Theta_1 + \text{cm}24 \: \Theta_2 - \frac{T_{\text{max}}}{2} \: \Delta t \: \text{sgn}(\lambda_{21})
\]

\[
\frac{\lambda_{22}}{(\lambda_{21} - \lambda_{22})} + \frac{\lambda_{21}^2 + \lambda_{22}^2}{2(\lambda_{21} - \lambda_{22})^2}
\]  \quad (4.74.5)

For the last row of submatrices we have by symmetry

\[
J(5,1) = \frac{\partial^2 J_a}{\partial \frac{r_1}{r_2} \partial \Delta t} = [ J(1,5) ]^T, \quad (4.75)
\]

\[
J(5,2) = \frac{\partial^2 J_a}{\partial \Theta_1 \partial \Delta t} = [ J(2,5) ]^T, \quad (4.76)
\]

\[
J(5,3) = \frac{\partial^2 J_a}{\partial \lambda_{11} \partial \Delta t} = [ J(3,5) ]^T, \quad (4.77)
\]

and

\[
J(5,4) = \frac{\partial^2 J_a}{\partial \lambda_{21} \partial \Delta t} = [ J(4,5) ]^T. \quad (4.78)
\]

The last term in the fifth row of the submatrices is given by

\[
J(5,5) = \frac{\partial^2 J_a}{\partial^2 \Delta t} = \frac{2}{\Delta t} \left[ \frac{3 \Delta t}{m} \: \text{cm}22 \: \text{cm}13 + \frac{3 \Delta t}{m} \: \text{cm}24 \: \text{cm}11
\]

\[
+ \text{cm}31 (r_1 - r_2)(\lambda_{11} - \lambda_{12}) \right] . \quad (4.79)
\]
Assembly of element properties
to obtain system equations

To solve for the unknowns in the whole solution region it is
necessary to assemble or combine the element matrix equations to form
the global equations governing the behavior over the entire problem
domain. The basis for the assembly procedure stems from the fact that
at a node where elements are interconnected, the values of the unknowns
are the same for each element sharing that node. The assembly is
performed by two routines. These are:

1. Node: Converts the nodal unknowns in the local (element)
numbering scheme to the nodal unknowns in the global (system)
numbering scheme (refer to Appendix II).

2. Build: Combines all the element unknowns to form the system
unknowns (refer to Appendix II).

Application of boundary conditions
and solution of system unknowns

Before the system equations can be solved, they must be modified
to account for the boundary conditions of the problem, otherwise the
system matrix will be singular. The application of the boundary
conditions is performed in routine Solve (refer to Appendix II) where in
addition to the boundary conditions on the states the transversality
equation is applied on the Lagrangian multipliers \( \lambda_1 \) and \( \lambda_2 \) at both the
initial and final time.
Application of transversality equation as boundary conditions

Substituting equations (3.67) and (3.69) into the transversality equation (3.39) gives the equation

\[ f(x) = F_{\text{max}} \text{sgn}(\lambda_1)\lambda_1 + T_{\text{max}} \text{sgn}(\lambda_2)\lambda_2 - 1 = 0. \]  \hspace{1cm} (4.80)

To apply equation (4.80) in the Newton-Raphson iteration it should be of the form

\[ \frac{\partial f}{\partial x} \Delta x = 0. \] \hspace{1cm} (4.81)

Differentiating equation (4.80) with respect to \( \lambda_1 \) gives

\[ \frac{\partial f}{\partial \lambda_1} = F_{\text{max}} \text{sgn}(\lambda_1). \] \hspace{1cm} (4.82)

Differentiating equation (4.80) with respect to \( \lambda_2 \) gives

\[ \frac{\partial f}{\partial \lambda_2} = T_{\text{max}} \text{sgn}(\lambda_2). \] \hspace{1cm} (4.83)

Substituting equations (4.82) and (4.83) into equation (4.81) gives

\[ F_{\text{max}} \text{sgn}(\lambda_1)\Delta \lambda_1 + T_{\text{max}} \text{sgn}(\lambda_2)\Delta \lambda_2 = f(x_i) \] \hspace{1cm} (4.84)

where \( f(x_i) \) is equal to zero. The right side vector elements \( f(x_i) \) at \( t_0 \) and \( t_f \) are modified in the MATRIX subroutine while the left hand side of equation (4.84) is implemented in the SOLVE subroutine (refer Appendix II).

In chapter 5 the results from the continuous time and the discrete time simulations are presented. The discrete case is compared to the
continuous case. The effect of varying the grid density, on the final time is examined. Recommendations for further study are also presented.
CHAPTER 5

RESULTS AND RECOMMENDATIONS

Introduction
In this paper two methods of determining the minimum time control for the r-theta manipulator, the continuous time method and the discrete time method were investigated. Two formulations, the first order and the second order, were used in the continuous time method. In the discrete time method the second order formulation was used. The simulations were written in Fortran 77 and implemented on an IBM 370 mainframe as well as a HARRIS H-800 computer. The reason for using two computer systems was the availability of different minimization routines on the two systems.

Continuous Time Method
Four IMSL routines [18] on the IBM 370 were used in the continuous time simulation. These include DVERK (a differential equation solver), LEQTIF (a linear equations solver) and two minimization routines, ZXHIN (a Quasi-Newton Method), and ZXCGR (a conjugate-gradient method).

The continuous time problem required a large number of iterations in order to get good initial guesses on the unknowns (initial values on $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ and final time $t_f$). A combination of the conjugate-gradient method and Newton's method was used. However, if the
initial guesses were not close to the optimum values the routines diverged. This difficulty is caused by the state and multiplier equations being very sensitive to the values of the Lagrangian multipliers. If the guesses are such that they do not cause switches in the multipliers then the Jacobian becomes singular. It is then necessary to come up with guesses on the multipliers at the initial time that bring about the correct number of switches in order for the minimization routine to converge.

The minimum time problem is solved for three cases.

Case 1

\[ r(t_o) = 1.0, \]  
\[ r(t_f) = 1.0, \]  
\[ \theta(t_o) = 0.0, \]  
and
\[ \theta(t_f) = 1.5708. \]  

Case 2

\[ r(t_o) = 1.5, \]  
\[ r(t_f) = 1.5, \]  
\[ \theta(t_o) = 0.0, \]  
and
\[ \theta(t_f) = 1.208. \]  

Case 3

\[ r(t_o) = 1.2, \]  
\[ r(t_f) = 1.2, \]  
\[ \theta(t_o) = 0.0, \]  

-64-
and \[ \Theta(t_f) = 1.795 \]. 

The velocities \( \dot{r} \) and \( \dot{\Theta} \) at both \( t_0 \) and \( t_f \) are set to zero.

Results for the continuous time simulation for case 1 are shown in Figures 5.1 - 5.4.

Figure 5.1 shows the trajectory of the first joint variable \( \Theta \) and its first and second time derivatives. These variables are indicated by \( T, TD, \) and \( TDD \) respectively, in the figure.

Figure 5.2 shows the trajectory of the second joint variable \( r \) and its first and second time derivatives. These variables are indicated by \( R, RD, \) and \( RDD \) respectively, in the figure.

Figure 5.3 illustrates the trajectory of the Lagrangian multiplier \( \lambda_1 \), and its first and second time derivatives. These variables are indicated by \( L1, LD1, \) and \( LDD1 \) respectively, in the figure.

Figure 5.4 illustrates the trajectory of the Lagrangian multiplier \( \lambda_2 \), and its first and second time derivatives. These variables are indicated by \( L2, LD2, \) and \( LDD2 \) respectively, in the figure.

All the variables are seen to exhibit either even or odd symmetry about \( t_f/2 \). The second derivative curves are not smooth at points indicating the switchings of the bang bang controls.

Discrete Time Method

Three routines were used in the discrete time simulation. These include LEQT1F on the IBM 370, and two routines from Sandia Laboratories, MINA (a grid search minimization technique), and ODE (an
integration routine) which are available on the HARRIS. The finite element method also uses a combination of techniques. A grid search method is used to give a reasonably good guess on the final time \( t_f \). This guess is then used in the finite element program to start the iterations on the nodal unknowns. The program then iterates until the convergence criterion has been satisfied. The guesses on the multipliers have to be of correct sign and must constitute a symmetrical path. The guesses on the joint variables \( r \) and \( \Theta \) must also constitute a symmetrical path. However the guesses on the magnitude of the multipliers can be quite far off, sometimes over a 100 percent. With a reasonable guess on the time \( t_f \), convergence is achieved quite rapidly.

Results for the discrete time solution for the three cases are given in Table 5.1. The number of elements used for the three cases is twenty one. The effect of varying the grid density in the discrete time solution of Case 1 are presented in Table 5.2. Results for the first case are illustrated in Figures 5.5 - 5.8.

Figure 5.5 shows the trajectories of \( \Theta \), and its first and second derivatives. Figure 5.6 shows the trajectories of \( r \), and its first and second derivatives. Figure 5.7 illustrates the trajectories of \( \lambda_1 \), and its first and second derivatives. Figure 5.8 illustrates the trajectories of \( \lambda_2 \), and its first and second derivatives.

Conclusions and Recommendations

Comparing the two simulations it is seen that the discrete time simulation agrees very closely with the continuous case, except in the
TABLE 5.1
DISCRETE TIME SOLUTIONS FOR THE MINIMUM FINAL TIME

<table>
<thead>
<tr>
<th>CASE</th>
<th>r(t₀) (ft)</th>
<th>r(tₙ) (ft)</th>
<th>θ(t₀) (rad)</th>
<th>θ(tₙ) (rad)</th>
<th>Final Time(tₙ) (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.0</td>
<td>1.0</td>
<td>0.0</td>
<td>1.5708</td>
<td>1.963</td>
</tr>
<tr>
<td>2</td>
<td>1.5</td>
<td>1.5</td>
<td>0.0</td>
<td>1.208</td>
<td>2.499</td>
</tr>
<tr>
<td>3</td>
<td>1.2</td>
<td>1.2</td>
<td>0.0</td>
<td>1.795</td>
<td>2.350</td>
</tr>
</tbody>
</table>

TABLE 5.2
EFFECT OF GRID DENSITY ON FINAL TIME FOR CASE 1

<table>
<thead>
<tr>
<th>SIMULATION</th>
<th>NUMBER OF ELEMENTS</th>
<th>FINAL TIME (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>continuous time</td>
<td>21</td>
<td>1.9644</td>
</tr>
<tr>
<td>discrete time</td>
<td>42</td>
<td>1.9629</td>
</tr>
<tr>
<td></td>
<td>84</td>
<td>1.9640</td>
</tr>
</tbody>
</table>

second derivatives where the discrete time curves are smoother than the corresponding continuous curves. This is due to the linear interpolation function used in the formulation of the element equations. A closer agreement is obtained when the grid density is increased (refer Table 5.2). However this also introduces corresponding increases in storage space requirements.

The finite element method achieved the objective of applying a discrete time method to the solution of the minimum time control. The transversality equation that was enforced at both ends was critical in heading the iterations in the right direction. This equation is not only true at the initial and final time but also at the internal nodes.
The transversality equation at the internal nodes includes some velocity terms in addition to the terms in the equation at the initial and final time. An area of further investigation could be the application of the transversality equation at the internal nodes. The use of higher order interpolation functions for the formulation of the element equations can also be investigated. Since a linear interpolation function was used in this paper, it does not possess first derivative continuity.

This thesis has investigated the application of finite element methods to the solution of the minimum time problem. The deviation of the discrete time solution from the continuous time solution has been investigated and found to be reasonable. Some recommendations for further investigation in the area have been presented.
CONTINUOUS TIME SIMULATION

Figure 5.1 Joint 1 Variable
CONTINUOUS TIME SIMULATION

FIGURE 5.2 JOINT 2 VARIABLE
CONTINUOUS TIME SIMULATION

Figure 5.3 Lambda1
DISCRETE TIME SIMULATION

FIGURE 5.7 LAMBDA1
DISCRETE TIME SIMULATION

FIGURE 5.8 LAMBDA2
LIST OF REFERENCES


-77-


APPENDIX I

DERIVATION OF THE MULTIPLIER EQUATIONS FOR THE SECOND ORDER FORMULATION

From equation (3.35) the performance index is defined as

\[ J_a(x, \dot{x}, u, \lambda, t) = \int_{t_0}^{t_f} \left[ 1 + \lambda_1[F_{\text{max}} u_1 - mr + mr^2 \theta \frac{\ddot{x}}{\dot{x}}] + \lambda_2[T_{\text{max}} u_2 - \frac{d}{dt}(mr^2 \theta)] \right] dt. \] (a.1)

The variation of \( J_a \) with respect to \( r \) and \( \theta \) is

\[ \delta J_a = dt + \int_{t_0}^{t_f} \left[ -\lambda_1 m \dot{\theta} + m \lambda_1 \dot{\theta}^2 \dot{\theta} + \lambda_1 mr \dot{\theta} + \lambda_2 mr \dot{\theta}^2 \right] dt. \] (a.2)

Integrating by parts and rearranging gives

\[ \delta J_a = dt + \left[ -\lambda_1 m \dot{\theta} + m \lambda_1 \dot{\theta}^2 \dot{\theta} + \lambda_1 mr \dot{\theta} + \lambda_2 mr \dot{\theta}^2 \right] \bigg|_{t_0}^{t_f} + \int_{t_0}^{t_f} \left[ \delta r \left( -m \lambda_1 + \lambda_1 \dot{\theta}^2 + \lambda_2 mr \right) \right] dt. \] (a.3)

\[ \delta \theta \left( \frac{d}{dt} \left( 2mr \lambda_1 \right) - \frac{d}{dt} \left( \lambda_2 mr^2 \right) \right) \] dt.

-79-
For an extremal curve

\[ \delta J_a = 0 . \]  

(a.4)

Setting the integrand in (a.3) equal to zero gives the multiplier equations

\[ -m_1 \ddot{\lambda}_1 + m_1 \dot{\lambda}_1 \dot{\theta}^2 + \dot{\lambda}_2 m_2 \theta \dot{\theta} = 0 \]  

(a.5)

and

\[ \frac{d}{dt} (2mr\lambda_1) + \frac{d}{dt} (\lambda_2 m_2^2) = 0 . \]  

(a.6)

Therefore, equation (a.4) reduces to

\[ \delta J = dt_f + \left( -\lambda_1 m \delta \dot{t} + \lambda_1 m \delta r + m \theta (\delta \theta) \lambda_1 \right. \]

\[ - \lambda_2 m_2 r (\delta r) \dot{\theta} - \lambda_2 m_2 \theta \delta \dot{\theta} + \lambda_2 m_2 \delta \theta \right) \bigg|_{t=t_0}^{t=t_f} \]  

(a.7)

The variation of a variable \( x \) at the final time is given by the relation

\[ \delta x(t_f) = \delta x_f - \dot{x} dt_f \]  

(a.8)

where \( \delta x_f \) is the variation of the final \( x \), and \( \dot{x} \) is the slope of \( x \) at time \( t_f \). Using equation (3.8) in (3.7) gives

\[ \delta J_a = dt_f + (\lambda_1 m \ddot{r} - \lambda_1 m \dot{r} - m \theta^2 \lambda_1 \]

\[ + \lambda_2 m_2 \dot{r} \dot{\theta} + \lambda_2 m_2 \dot{\theta}^2 \theta - \lambda_2 m_2 \dot{\theta}^2 \theta) \bigg|_{t=t_0}^{t=t_f} \]

\[ - (-\lambda_1 m \delta \dot{r} + \lambda_1 m \delta r + m \theta \delta \theta \lambda_1) \]

\[ - \lambda_2 m_2 \theta \delta r - \lambda_2 m_2 \theta \delta \theta + \lambda_2 m_2 \theta \delta \theta \bigg|_{t=t_0}^{t=t_f} \]  

(a.9)

From equations (3.21) and (3.22)
\[ F_{\text{max}} \lambda_1 u_1 = \lambda_1 m r - mr\dot{\theta} \lambda_1 \]  \hspace{1cm} (a.10)

and

\[ T_{\text{max}} \lambda_2 u_2 = \lambda_2 m r^2 \dddot{\theta} + \lambda_2 m 2r\dot{r}\dot{\theta} \]  \hspace{1cm} (a.11)

The initial and final velocities \((\dot{r}, \dot{\theta})\) are zero in the example and the initial and final states are specified. Substituting equations (a.10) and (a.11) into (a.9) together with the boundary conditions gives

\[ 1 + F_{\text{max}} \lambda_1 u_1 + T_{\text{max}} \lambda_2 u_2 = 0 \]  \hspace{1cm} (a.12)

The above equation is the transversality equation for the second order formulation.
APPENDIX II

FLOW CHART AND SUBROUTINES USED IN FINITE ELEMENT PROGRAM

START
↓
INITIALIZE VARIABLES
↓
MAKE GUESSES ON NODAL UNKNOWNS
↓
SUBROUTINE NODE
↓
START ITERATIONS
↓
SUBROUTINE SWITCH
↓
SUBROUTINE BUILD ←→ SUBROUTINE MATRIX
↓
SUBROUTINE SOLVE
↓
NO ← CONVERGED?
↓
YES
↓
STOP

Figure A.1 Flow chart of Finite Element Program

-82-
The MAIN program initializes the variables, defines the initial guesses on the unknowns, calls the various subroutines and controls the iterations.

Subroutine NODE is used to convert the local nodal unknowns into the global (or system) unknowns.

SUBROUTINE NODE(NT,NTABLE,MAXELM)

INTEGER NTABLE(MAXELM,8),NT,MAXELM,NELM,A1,A2,A3,A4,A5,A6,A7,A8

NELM = element number
NT = Number of elements
NTABLE = Table containing the global numbers for the element unknowns
MAXELM = Maximum number of elements

NELM=0
DO 10 I=1,NT
   A1=4*(I-1)+1
   A2=A1+4
   A3=4*(I-1)+2
   A4=A3+4
   A5=4*(I-1)+3
10 CONTINUE
A6=A5+4
A7=4*(I-1)+4
A8=A7+4

NELM=NELM+1

NTABLE(NELM,1)=A1
NTABLE(NELM,2)=A2
NTABLE(NELM,3)=A3
NTABLE(NELM,4)=A4
NTABLE(NELM,5)=A5
NTABLE(NELM,6)=A6
NTABLE(NELM,7)=A7
NTABLE(NELM,8)=A8

CONTINUE

RETURN

END

Subroutine SWITCH is used to determine if a switch of either lambda1 or lambda2 or both, has occurred within the element. The signs on the lambdas at the two nodes of the element are compared. If the signs are different, a switch has occurred within the element.
Subroutine \texttt{MATRIX} is used to set up the element matrix equations as well as the element Jacobian. The relations developed in Chapter 4 are used in this routine.

Subroutine \texttt{BUILD} is used to assemble the element Jacobian into the global Jacobian.

\begin{verbatim}
SUBROUTINE BUILD(NT, NTABLE, ITABLE, JAC, RSV, SGNL1, SGNL2, TS1, TS2, GJ, M
*AXRG, RSVG, MAXELM, MAT, R, THETA, LB1, LB2, MAXN, DT)

REAL*8 JAC(9,9), RSV(9), SGNL1, SGNL2, TS1(MAXELM), TS2(MAXELM)
REAL*8 GJ(MAXRG, MAXRG), RSVG(MAXRG), MAT(9,9)
REAL*8 R(MAXN), THETA(MAXN), LB1(MAXN), LB2(MAXN), DT
INTEGER NT, NTABLE(MAXELM,8), ITABLE(8), I1, I2

GJ is the global jacobian.
RSVG is the global right side vector
NT is the number of elements

ARRAYS ARE INITIALIZED

DO 70 I=1,4*NT+5
   DO 80 J=1,4*NT+5
      GJ(I,J)=0.0
   80 CONTINUE
\end{verbatim}
RSVG(I) = 0.0

70 CONTINUE

C C THE GLOBAL JACOBIAN AND GLOBAL RIGHT SIDE VECTOR ARE ASSEMBLED

C DO 10 I1 = 1, NT

DO 20 J = 1, 8
     ITABLE(J) = NTABLE(I1, J)
20 CONTINUE

C C THE ELEMENT MATRIX, JACOBIAN AND RIGHT SIDE VECTOR ARE OBTAINED

C CALL MATRX(R, MAXN, THETA, LB1, LB2, DT, NT, SGNL1, SGNL2, TS1, TS2, MAXELM, M
     *AT, RSV, JAC, I1)

C The element jacobians and the element right side vectors are
C assembled into the global Jacobian and the global right side
C vector here.

DO 30 I2 = 1, 8

DO 40 J = 1, 8
     GJ(ITABLE(I2), ITABLE(J)) = JAC(I2, J) + GJ(ITABLE(I2), ITABLE(J))
40 CONTINUE

GJ(ITABLE(I2), 4*NT+5) = JAC(I2, 9) + GJ(ITABLE(I2), 4*NT+5)
RSVG(ITABLE(I2)) = RSV(I2) + RSVG(ITABLE(I2))
30 CONTINUE
DO 95 J=1,8

   GJ (4*NT+5, ITABLE(J)) = JAC(9, J) + GJ (4*NT+5, ITABLE(J))

95 CONTINUE

   GJ (4*NT+5, 4*NT+5) = JAC(9, 9) + GJ (4*NT+5, 4*NT+5)
   RSVG(4*NT+5) = RSV(9) + RSVG(4*NT+5)

C

10 CONTINUE

RETURN

END

C

SUBROUTINE SOLV(GJ, RSVG, MAXRG, NT, MRK, WK, R, THETA, LB1, LB2, MAXN, CONV,
*DT, NTABLE, MAXELM, ITABLE, K)

REAL*8 GJ(MAXRG, MAXRG), RSVG(MAXRG), WK(MRK), PI, CONVN, CONVD
REAL*8 R(MAXN), THETA(MAXN), LB1(MAXN), LB2(MAXN), DT, CONV, A, B, C
INTEGER M1, N2, IA, IDGT, IER, NTABLE(MAXELM, 8), ITABLE(8), K

GJ is the global Jacobian

RSVG is the global right side vector

NT is the number of elements.

R is the Joint 2 variable
Theta is the joint 1 variable

LB1 is lambda1

LB2 is lambda2

DT is the length of element

CONV is the convergence criteria

CONVN= 0.0
CONVD= 0.0
CONV= 0.0

BOUNDARY CONDITIONS ARE SPECIFIED

DO 60 I=1,3
   DO 60 J=1,4*NT+5
      GJ(I,J)= 0.0
      GJ(4*NT+I,J)= 0.0
60      CONTINUE

DO 65 I=1,2
   RSVG(I)= 0.0
   RSVG(4*NT+I)= 0.0
65      CONTINUE

DO 68 I=1,2
   GJ(I,I)= 1.0
   GJ(4*NT+I,4*NT+I)= 1.0
68      CONTINUE
TRANSVERSALITY EQUATION APPLIED AT THE INITIAL AND FINAL TIME

 known data

 Fmax = 1
 Tmax = 1
 m = 1

 at t = t₀  sgn(LB₁) = 1
 sgn(LB₂) = -1
 at t = tₜ  sgn(LB₁) = 1
 sgn(LB₂) = 1

 Substituting the above values into equation (4.83) gives

 GJ(3,3) = 1.0
 GJ(3,4) = -1.0
 GJ(4*NT+3,4*NT+3) = 1.0
 GJ(4*NT+3,4*NT+4) = 1.0

 GLOBAL JACOBIAN IS SOLVED

 M₁=1
 N₂=4*NT+5
 IA=MAXRG
 IDGT=0

 CALL LEQT1F(GJ,M₁,N₂,IA,RSVG,IDGT,WK,IER)
C UPDATE NODAL VARIABLES

C

DO 20 I3=1,NT+1
   R(I3) = R(I3) - RSVG(4*(I3-1)+1)
   THETA(I3) = THETA(I3) - RSVG(4*(I3-1)+2)
   LB1(I3) = LB1(I3) - RSVG(4*(I3-1)+3)
   LB2(I3) = LB2(I3) - RSVG(4*(I3-1)+4)
20 CONTINUE
   DT = DT - RSVG(4*NT+5)
C
C CONVERGENCE CRITERION IS COMPUTED. Refer equation (4.53)
C
C
DO 30 I=1,NT+1
   CONVN= CONVN+(RSVG(4*(I-1)+1)**2)
   CONVD= CONVD+(R(I)**2)
C
   CONVN= CONVN+(RSVG(4*(I-1)+2)**2)
   CONVD= CONVD+(THETA(I)**2)
C
   CONVN= CONVN+(RSVG(4*(I-1)+3)**2)
   CONVD= CONVD+(LB1(I)**2)
C
   CONVN= CONVN+(RSVG(4*(I-1)+4)**2)
   CONVD= CONVD+(LB2(I)**2)
C
-90-
30 CONTINUE

C

CONVN = CONVN + (RSVG(NT+5)**2)
CONVD = CONVD + (DT**2)

C

CONV = CONVN / CONVD
CONV = DSQRT(CONV)

RETURN

END

C
A SOLUTION TECHNIQUE FOR THE MINIMUM-TIME CONTROL PROBLEM OF AN R-THETA MANIPULATOR

by

ANUP SHETTY

B.S., Wichita State University, 1985

AN ABSTRACT OF A MASTER'S THESIS

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Mechanical Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1987
ABSTRACT

This paper investigates the minimum time control of a two degree of freedom manipulator subject to control magnitude constraints. Two methods of solution, a continuous time method and a discrete time method are applied, and their relative merits are examined.

The mathematical model for the r-θ manipulator, a two degree of freedom manipulator operating in the horizontal x-y plane is developed.

The necessary conditions for the minimum time control of the manipulator are presented. Two formulations, the first and the second order formulations, are used.

The finite element method is developed for the discrete time simulation of the time optimal control of the manipulator. A combination of a grid search technique and Newton-Raphson iteration on the finite element equations is used to obtain the minimum time with the state and control histories. The discrete time solution is compared to the continuous time solution. The results of the computer simulations are presented, as well as recommendations for further study.