OPTIMAL TRAJECTORY PLANNING FOR HYDRAULIC MANIPULATORS WITH POWER LIMITATIONS

by

Maria-Magdalena Dilimot

A thesis

presented to the University of Manitoba

in fulfilment of the

thesis requirement for the degree of

Master of Science

in

Mechanical & Industrial Engineering

Winnipeg, Manitoba, Canada 1999

THE UNIVERSITY OF MANITOBA

FACULTY OF GRADUATE STUDIES

*****

COPYRIGHT PERMISSION PAGE

Optimal Trajectory Planning for Hydraulic Manipulators with Power Limitations

BY

Maria-Magdalena Dilimot

A Thesis/Practicum submitted to the Faculty of Graduate Studies of The University of Manitoba in partial fulfillment of the requirements of the degree of

MASTER OF SCIENCE

Abstract

Machine automation requires the robotic machine to be at least as productive as a manually operated machine. To increase robot productivity robot motion speed should be improved. A feasible approach to improving the motion speed is to minimize the motion time needed to perform a given task subject to actuator constraints.

This work addresses the problem of optimal trajectory planning for heavy-duty hydraulic manipulators. These manipulators have the following characteristics: they are powered by a single engine mounted on the machine and they are under-powered even during normal operations resulting in dynamic power redistribution to the actuators. For the hydraulic manipulators, the actuator characteristics are very significant and complex due to high nonliniarities in the hydraulic system and power coupling between the actuators.

The method developed in this thesis focuses on utilizing advantageously the actuator capabilities to minimize the time needed to move the manipulator end-effector along a specified path. To perform the search for the minimum motion time along the specified path, a downhill simplex technique is implemented. The method is applied to a Caterpillar 215B excavator-based log loader in a typical pick and place task.

The main contributions of this thesis are the incorporation of the complex actuator characteristics in the optimal trajectory planning and the implementation of an optimization algorithm (downhill simplex method), which shows effective results for solving the optimal trajectory planning problem.

Acknowledgments

I wish to express my gratitude to my advisor, Dr. N. Sepehri, whose encouragement and concern have made this study possible. I would also like to thank my advisory committee members, Dr. J. F. Peters, Dr. D. Strong and Dr. S. Balakrishnan, for their review and suggestions.

I would also like to acknowledge the financial support of the Institute of Robotics and Intelligent Systems (IRIS) and Precarn Associates Inc.

Table of Contents

ABSTRACT ..... **IV**

ACKNOWLEDGMENTS ..... **V**

LIST OF FIGURES ..... **VIII**

LIST OF TABLES ..... **XI**

1.INTRODUCTION **1**

1.1 MOTIVATION 1

1.2 OBJECTIVE AND SCOPE OF THIS WORK 4

2.RELEVANT BACKGROUND **6**

2.1 AN APPLICATION EXAMPLE 6

2.1.1 Linkage Dynamics *11*

2.1.2 Actuator Dynamics *13*

2.2 MOTION PLANNING FOR MANIPULATORS 16

2.2.1 Identification of Various Constraints *16*

2.2.2 Minimum-time motion planning *17*

3.MINIMUM-TIME TRAJECTORY PLANNING **26**

3.1 JOINT TRAJECTORIES GENERATION AND TIME SCALING 26

3.1.1 Formulation of spline function joint trajectories *26*

3.1.2 Constant time scaling *31*

3.2 TIME SCALING OF TRAJECTORIES TO SATISFY ACTUATOR CONSTRAINTS 32

3.2.1 Maximum achievable joint velocities *32*

3.2.2 Computation of time scaling factor *59*

3.3 OPTIMIZATION OF JOINT TRAJECTORIES 61

4.DEMONSTRATIVE RESULTS **69**

4.1. EXAMPLE 1 69

4.2. EXAMPLE 2 94

5.CONCLUSIONS **100**

5.1 ACHIEVEMENTS 100

5.2. FUTURE DEVELOPMENT 101

REFERENCES **102**

APPENDICES **106**

A. EXCAVATOR-BASED MANIPULATOR KINEMATICS 106

A.1 Forward kinematic *106*

A.2 Inverse kinematics *108*

B. SPLINE INTERPOLATION 110

C. APPLICATION OF DOWNHILL SIMPLEX METHOD 117

List of Figures

Fig. 1.1:Excavator-based machines (a) on wheels; (b) on tracks. 2
Fig. 2.1: Schematic of a typical excavator-based log-loader. 7
Fig. 2.2: Typical hydraulic actuator working with an open-center valve. 9
Fig. 2.3: Function of variable-displacement pumps in a Caterpillar 215B excavator-based machine. 10
Fig. 2.4: Typical actuator-arm mechanism. 12
Fig. 2.5: (a) Function $J(\theta)$ over full range for Boom/Stick motion; (b) Curve $X(\theta)$ for Boom/Stick 13
Fig. 3.1: Excavator-based machine without cross-overs. 33
Fig. 3.2: Scheme for computing scaling factor $k = k_1 k_2$ 36
Fig. 3.3: Pumps output flow versus summing pressure $P_1 + P_2$ 37
Fig. 3.4: Single-link actuation. 38
Fig. 3.5: Variation of orifice areas versus spool displacement for actuator. 39
Fig. 3.6: Algorithm for solving boom actuator equations. 41
Fig. 3.7: Boom position, velocity, acceleration profiles. 43
Fig. 3.8: (a) Boom torque; (b) boom hydraulic force. 44
Fig. 3.9: (a) Boom cylinder velocity; (b) Desired flow to the boom. 44
Fig. 3.10: Boom line pressures and boom spool displacement. 45
Fig. 3.11: (a) Summing pressure, $P_1 + P_2$; (b) Maximum available pump-rate. 45
Fig. 3.12: Two-link actuation with priority action 46
Fig. 3.13: Algorithm for solving swing and stick actuator equations. 48
Fig. 3.14: Swing and stick position, velocity and acceleration profiles. 51
Fig. 3.15: Swing, boom and stick actuator torque/forces. 52
Fig. 3.16: Swing and stick actuator velocities. 52
Fig. 3.17: Swing and stick spool displacements. 53
Fig. 3.18: Swing and stick line pressures 53

Fig. 3.19: (a) Supply pressures; (b) Summing pressure, $P_1 + P_2$; (c) Maximum available pump flow-rates 54
Fig. 3.20: Swing, boom and stick position, velocity and acceleration profiles. 55
Fig. 3.21: Swing, boom and stick actuator velocities. 56
Fig. 3.22: Swing, boom and stick spool displacements. 56
Fig. 3.23: (a) Supply pressures; (b) Summing pressure, $P_1 + P_2$; (c) Maximum available pump flow-rates 57
Fig. 3.24: Pump supply pressures. 58
Fig. 3.25: (a) Summing pressure, $P_1 + P_2$; (b) Maximum available pump flow-rates 58
Fig. 3.26: Scheme for determining time scaling factor λ 60
Fig. 3.27: Possible outcomes for a step in the downhill simplex method 62
Fig. 3.28: Optimization algorithm. 65
Fig. 4.1: Initial Cartesian path 70
Fig. 4.2: Convergence behavior of the optimization algorithm; 1st run. 75
Fig. 4.3: Stopping criteria: (a) Vector distance moved in each step; (b) Decrease in function value in each step; 1st run. 75
Fig. 4.4: Blow-up of Fig. 4.3; 1st run. 76
Fig. 4.5: Blow-up of Fig. 4.2; 1st run. 76
Fig. 4.6: Convergence behavior of the optimization algorithm; 2nd run. 77
Fig. 4.7: Cartesian path corresponding to the optimal solution. 80
Fig. 4.8: Joint position profiles: (a) initial solution; (b) optimal solution. 81
Fig. 4.9: Joint velocity profiles: (a) initial solution; (b) optimal solution. 81
Fig. 4.10: Joint acceleration profiles: (a) initial solution; (b) optimal solution. 82
Fig. 4.11: Swing, boom and stick spool displacements. 82
Fig. 4.12: Joint positions for four and eight knot points. 83
Fig. 4.13: Joint velocities for four and eight knot points. 83
Fig. 4.14: Joint accelerations for four and eight knot points. 84
Fig. 4.15: Comparison of joint velocities for no payload versus payload 84
Fig. 4.16: Comparison of joint accelerations for no payload versus payload. 85
Fig. 4.17: Swing, boom and stick velocities. 85
Fig. 4.18: Desired flow-rates to swing, boom and stick. 86

Fig. 4.19: Desired flow-rates: (a) to line 1 (boom), (b) to line 2 (swing and stick), versus maximum available pump flow. 86
Fig. 4.20: Flow-rates scaling factors. 87
Fig. 4.21: Scaled desired flow-rates: (a) to line 1 (boom); (b) to line 2 (swing and stick), versus available pump flow. 87
Fig. 4.22: Unscaled versus scaled swing, boom and stick velocities. 88
Fig. 4.23: (a) Summing pressure, $P_1 + P_2$; (b) Maximum available pump flow-rates. 88
Fig. 4.24: Swing, boom and stick velocities. 89
Fig. 4.25: Desired flow-rates to swing, boom and stick versus maximum available pump flow. 89
Fig. 4.26: Desired flow-rates: (a) to line 1 (boom), (b) to line 2 (swing and stick), versus maximum available pump flow. 90
Fig. 4.27: Flow-rates scaling factors. 90
Fig. 4.29: Unscaled versus scaled swing, boom and stick velocities 91
Fig. 4.30: Joint position profiles: (a) initial solution; (b) optimal solution. 92
Fig. 4.31: Joint velocity profiles: (a) initial solution; (b) optimal solution. 92
Fig. 4.32: Joint acceleration profiles: (a) initial solution; (b) optimal solution. 93
Fig. 4.33: Comparison of joint velocities for no payload versus payload. 93
Fig. 4.34: Straight line Cartesian path. 95
Fig. 4.35: Path in: (a) Cartesian space; (b) joint space. 96
Fig. 4.36: Joint position trajectories. 96
Fig. 4.37: Joint velocity trajectories. 97
Fig. 4.38: Cartesian position trajectories. 97
Fig. 4.39: Effect of increasing number of knot points on path error. 98
Fig. 4.40: Cartesian velocity trajectories. 98
Fig. 4.41: Comparison of optimal joint velocities for no payload versus payload. 99
Fig. A.1: Coordinate frames 107
Fig. B.1: n-points trajectory 110

List of Tables

Table 4.1: Effect on increasing the number of knot points on optimal solution..... 71
Table 4.2: Joint displacements corresponding to Cartesian knot points..... 72
Table 4.3: Initial simplex..... 73
Table A.1. Kinematic parameters of a 215 Caterpillar excavator-based machine.....106

Chapter 1

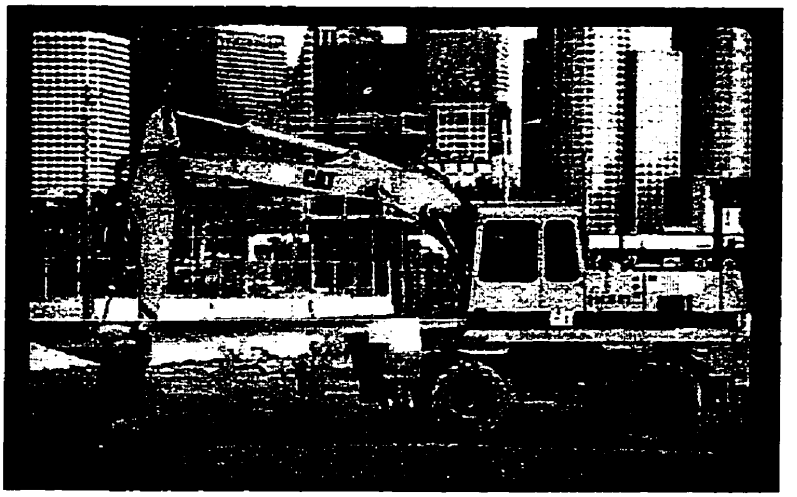
Introduction

1.1 Motivation

Many applications of robotic systems require the manipulators to operate from moving bases. A moving base - or mobile manipulator - has two important features. First, it has almost infinite workspace. Secondly, it may find many more applications, especially in unstructured, hostile environments.

Mobile manipulators are typically applied in primary industries such as construction, mining, and forestry. Figure 1.1 shows two examples of machines used in these industries. The common characteristics of mobile manipulators are: (i) they utilize hydraulic actuators which are powered by a single engine mounted on the machine and, (ii) they are under-powered even during normal operation resulting in dynamic power redistribution, i.e. the actuators usually request more power than the engine can supply (Krishna, 1998).

Robotics are now beginning to examine problems related to automating tasks in the areas of construction, mining, excavation and forestry. These tasks include mass excavation and continuous mining where a digging machine fills a bucket with material from a pile or a rock face, transports the bucket load to a waiting truck or conveyer belt, and dumps the load in the truck-bed/belt. These tasks are ideal candidates for automation



(a)

(b)

Fig. 1.1:Excavator-based machines (a) on wheels; (b) on tracks.

since they are repetitive and there exists room for enhancing productivity while decreasing production costs.

Automation can be a practical reality only if the robotic machinery is more productive than a manually operated one, while offering lower production cost and maintaining safe operation. In order to increase the productivity, the performance of the manipulator should be optimized. This can be done by planning the motion of the manipulator, in such a way that it can perform its task quickly (i.e., minimizing the motion time), with the minimum energy consumption, while the stability of the machine is maintained.

The speeds, which can be achieved by a manipulator during a given task, are limited by the capabilities of the manipulator. In order to use a manipulator at maximum efficiency, the optimal motion planning should be performed using the dynamic model of the manipulator and the more accurate the dynamic model is, the better the manipulator capabilities can be used. The dynamic model of a manipulator consists of the models of the linkage and the actuators driving the linkage. The linkage dynamics for a heavy-duty mobile hydraulic manipulator can be modeled using the Newton-Euler equations. A more difficult problem is to model the hydraulic dynamics, as the actuator model is very complex, coupled and nonlinear.

Early work on optimal motion planning was primary devoted to industrial manipulators driven by electric motors (see for example Bobrow *et al.,* 1985; Rajan, 1985). Those methods exploited the manipulator linkage dynamics to optimize an objective function, which was typically the execution time or a combination time-energy for a task. They did not consider the actuator dynamics and assumed a known torque limit

curve for each joint actuator. However, the torque limit curve is not easy to compute ahead of time for a heavy-duty hydraulic manipulator since the limit curve is a function of many variables due to coupled actuation in this type of manipulator.

Lin *et al.* (1983) developed a method to minimize the time required to move along a specified path, subject to the constraints on joint velocities, accelerations and jerks. The drawback of the method is that it uses constant bounds on joint velocities, accelerations and jerks. Therefore, it did not account for the actual capabilities of the manipulator actuators and actuators loading. The method can be improved for heavy-duty hydraulic manipulators, if the joint velocity bounds are determined considering the complex relation between the velocity and the actuator and linkage effects, such as loading, inter-actuator coupling and power limitations. Therefore, the velocity bounds should be determined using the dynamic model of the manipulator.

1.2 Objective and Scope of this Work

In this thesis an off-line method for planning the optimal trajectory for hydraulic manipulators is developed. The method aims to improve the manipulator motion speed by minimizing the execution time needed to perform a given task subject to the constraints imposed by the limited capabilities of the actuators. The actuator capabilities are limited due to the fact that they are powered by a limited power engine.

The approach taken is as follow. For a path specified by Cartesian points, the corresponding joint trajectories are generated using spline functions and the downhill simplex method is adopted to minimize the motion time. The time intervals between

adjacent points are the optimization variables. A "feasible solution converter" is developed to convert, during the search, an infeasible solution into a feasible one. The feasible solution converter, actually, determines whether a trajectory is dynamically realizable given the power limitations. If not, it changes the speed at which the manipulator follows the path so that the power limitations are not violated. By speed change is meant a constant scaling of the velocity profile so that the total execution time is scaled without changing the actual path through space. The constant scaling factor is determined considering the relation between the velocity and the actuator loading and power limitations.

The method developed in this thesis is applied to a Caterpillar 215B excavator-based log loader, which is a three-links manipulator with an additional moveable end-effector, namely the gripper. The first three links motion serves to control the position of the end-effector. The end-effector itself is not included in the studies performed, since only the position of the end-effector is of interest in this thesis.

The organization of this thesis is as follows. Chapter 2-Relevant Backgrounds-is divided in two parts. Part 1 presents a description of a heavy-duty mobile manipulator, its linkage dynamics and the dynamics of the actuators driving the links. Part 2 introduces a literature survey on optimal motion planning for manipulators. In Chapter 3, the optimal trajectory planning algorithm is developed. Chapter 4 presents results of the simulation studies. Conclusions are presented in Chapter 5.

Chapter 2

Relevant Background

This chapter first describes a typical heavy-duty mobile hydraulic manipulator used in primary industries. Its linkage dynamics and the dynamics of the hydraulic actuators driving the links are also described. Next, different optimal motion planning methods are surveyed and potentials and limitations of each method are outlined.

2.1 An Application Example

The heavy-duty mobile hydraulic manipulator, considered in this thesis, is a Caterpillar 215B excavator based machine (see Fig. 2.1). It is a three-degree-of freedom manipulator with an additional moveable end-effector, namely the gripper. The upper structure of the machine is rotated on the carriage by a hydraulic swing motor through a reduction gear. The other two main links, "boom" and "stick", are movable around their joints by hydraulic cylinders. The boom and stick together with the swing motion serve to control the position of the end-effector.

The power required to actuate the cylinders and the hydraulic motor is derived from an engine which through a gear train drives two hydraulic pumps. The output flow from the pumps is used to operate two separate hydraulic circuits that may be selectively interconnected by cross-overs valves. Pump 1 supplies the hydraulic fluid to the gripper valve and the boom valve. The two valves control the flow to the hydraulic actuators, i.e. a hydraulic motor for the gripper, a hydraulic cylinder for the boom. Pump 2 supplies the

hydraulic fluid to the swing valve, which controls the motor for rotating the Cab and to the stick valve to actuate the hydraulic cylinder for moving the stick.

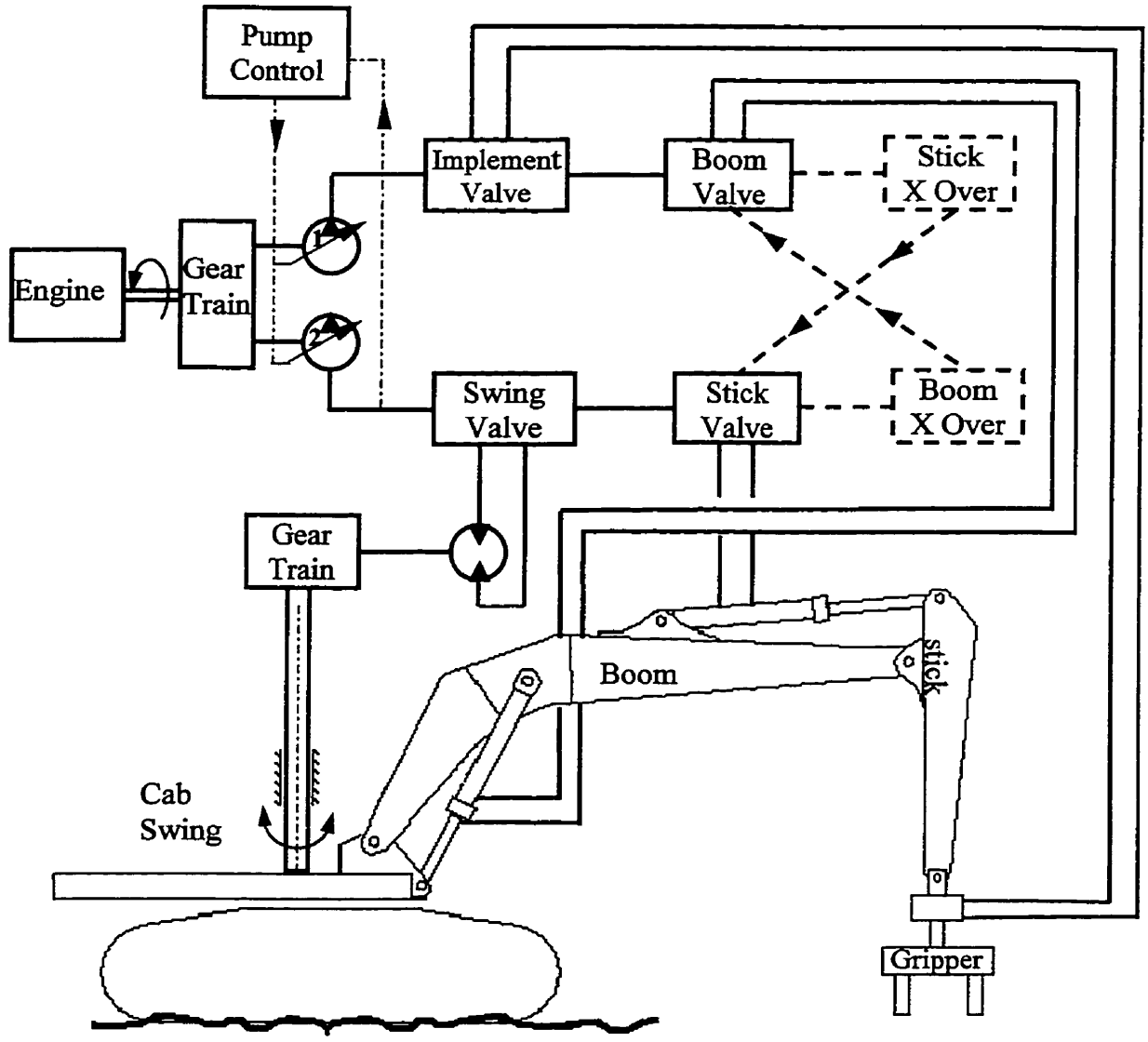


Fig. 2.1: Schematic of a typical excavator-based log-loader.

Stick movement controlled by the stick main valve, cannot be achieved if the swing main valve is fully open. If the latter is partly open, the stick can operate but at slower speed. The motion of the boom and the stick, in some models, are coupled via cross-over valves (shown by dashed lines in Fig. 2.1). The output from pump 2 may be shifted to facilitate movement of the boom as indicated by the cross-over and similarly the output of pump 1 may be shifted to apply fluid to stick valve via the stick cross-over valve depending on the demands of the two hydraulic circuits. This will allow a faster movement of one when the other is at less than full speed operation.

Pump 1 services on a priority basis, first the gripper valve, then the boom valve and finally the stick cross-over of a second hydraulic circuit. Pump 2 supplies the hydraulic flow on a priority basis first to the swing valve, then to the stick valve and to the boom cross-over.

When the total sum of the pressures in the implement circuits becomes high enough, the pumps reduce their outputs. This type of hydraulic circuit is known as load-sensing torque-limited circuit. The highest load is sensed and the output flow is changed to meet the maximum torque available from the engine.

The hydraulic system, described above, is an open-center system. In an open-center system the pumps do not reduce their output to zero. When no actuator flow is demanded, the pumps still output a non-zero flow. This "idle" flow goes to the tank through the center orifices. When the actuators are commanded to move, the tank orifices slowly close and are fully closed when maximum velocity is demanded. A typical open-center valve and a single-rod (asymmetric) cylinder are shown in Fig. 2.2.

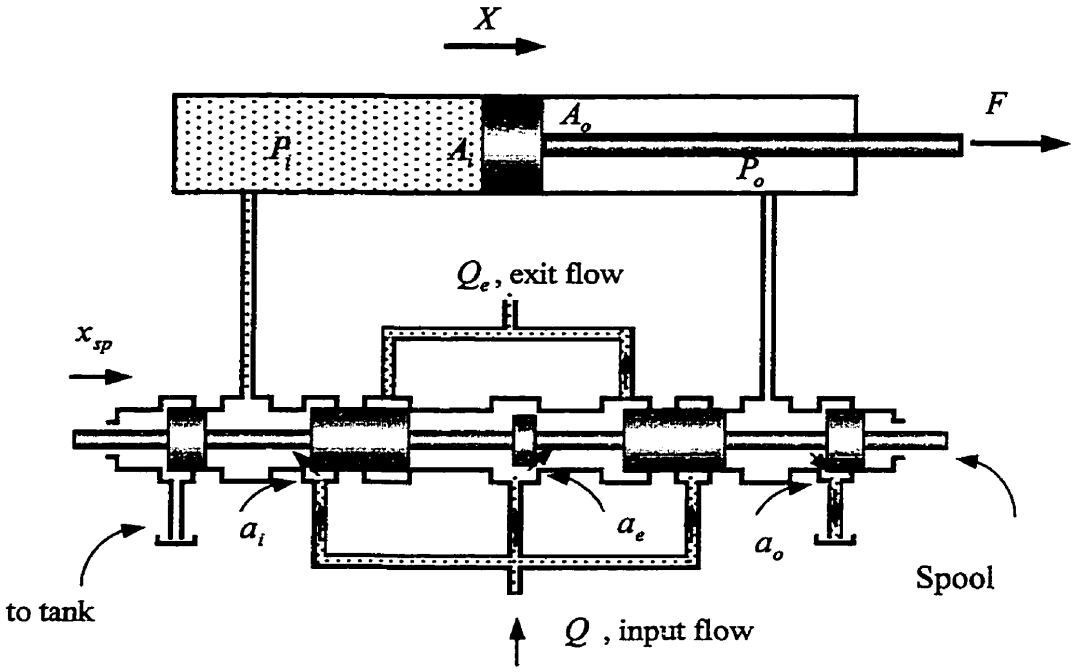


Fig. 2.2: Typical hydraulic actuator working with an open-center valve.

Note that the single-rod cylinders, used for the links (boom, stick) actuation, are characterized by limited linear motions. Since the joints are revolute, use of linear actuators result in joint angle limitations. This means that boom, stick motions are restricted to specific ranges. Kinematic analysis of the machine can be found in Appendix A.

A Diesel engine is used to turn the two axial-piston variable-displacement pumps. The two pumps have a common swash plate whose angle is controlled according to the summing pressure, $P_1 + P_2$. Consequently, these pumps have identical output flow or equivalently identical displacement coefficient, but different output pressures.

Figure 2.3 shows a typical pressure sum versus output curve for these pumps. According to the schematic diagram shown in the same figure, the output pressure of the pumps are sampled by two orifices. These pressures are applied to two small pistons that change the angle of the swash plate against three parallel springs. This mechanical feedback system serves to limit the power that is drawn from the engine, so that the pressure-flow curve is below the power limit curve.

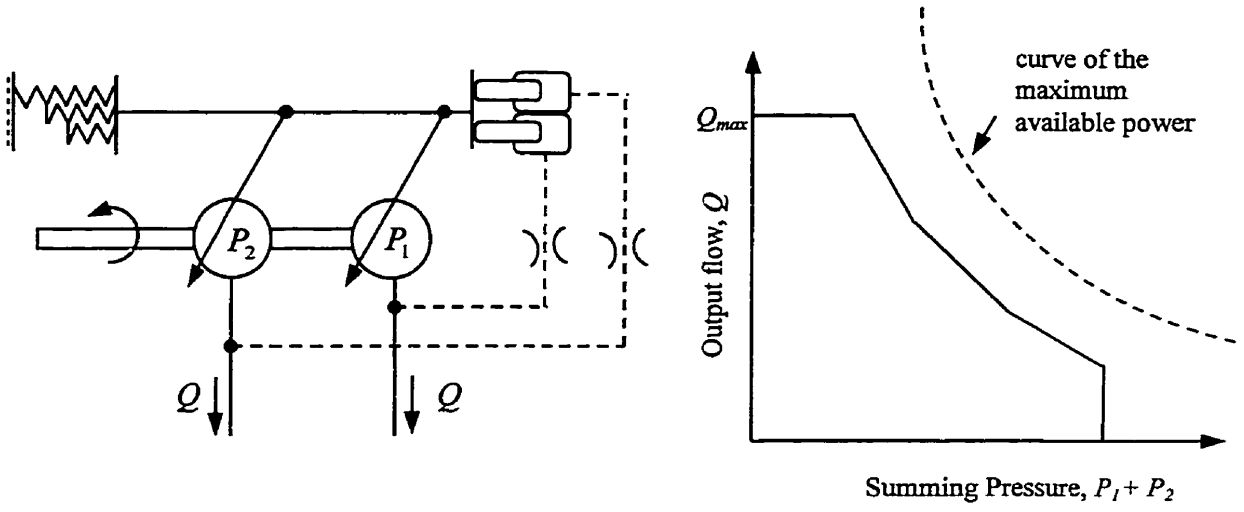


Fig. 2.3: Function of variable-displacement pumps in a Caterpillar 215B excavator-based machine.

The dynamic model of the hydraulic manipulator consists of the model of the linkage and the model of the actuators driving the manipulator joints. The linkage and actuator dynamics are described in the following.

2.1.1 Linkage Dynamics

The joint torques required to move the linkage are described by the dynamic equations of motion and depend on instantaneous joint position, velocity, acceleration, and the load that it is carrying. For an N-joint manipulator, the linkage dynamics can be compactly written as:

$$\boldsymbol{T}(t) = \boldsymbol{M}(\theta(t))\ddot{\theta}(t) + \boldsymbol{C}(\theta(t), \dot{\theta}(t)) + \boldsymbol{F}(\dot{\theta}(t)) + \boldsymbol{G}(\theta(t)) \tag{2.1}$$

where

$\boldsymbol{T}(t)$ is the $N \times 1$ vector of joint torques supplied by the actuators. $\boldsymbol{M}(\theta(t))$ is an $N \times N$ matrix, sometimes called the manipulator mass matrix. The vector $\boldsymbol{C}(\theta(t), \dot{\theta}(t))$ represents torques arising from centrifugal and Coriolis forces. The vector $\boldsymbol{F}(\dot{\theta}(t))$ represents torques due to friction acting at the joints. The vector $\boldsymbol{G}(\theta(t))$ represents torques due to gravity. $\theta(t)$ is the $N \times 1$ vector of joint displacements, with $\theta(t) = [\theta_1(t), \theta_2(t), ..., \theta_N(t)]^T$.

The relationship between the effective actuation force, F, applied on the link and the joint torque T, is obtained by applying the principle of virtual work.

$$T \cdot d\theta = F \cdot dX \tag{2.2}$$

where $d\theta$ and dX denote the incremental changes in joint displacement and piston displacement, respectively. Therefore, the force F can be calculated from the following equation:

$$F = \frac{T}{J(\theta)} \tag{2.3}$$

where the derivative function $J(.)$ is defined as follows:

$$J(\theta) = \frac{dX}{d\theta} \tag{2.4}$$

The joint displacement, θ, and the piston displacement, X, are related by geometrical configuration. With respect to the Fig. 2.4 the numerical value of $J(\theta)$ can be evaluated as follows:

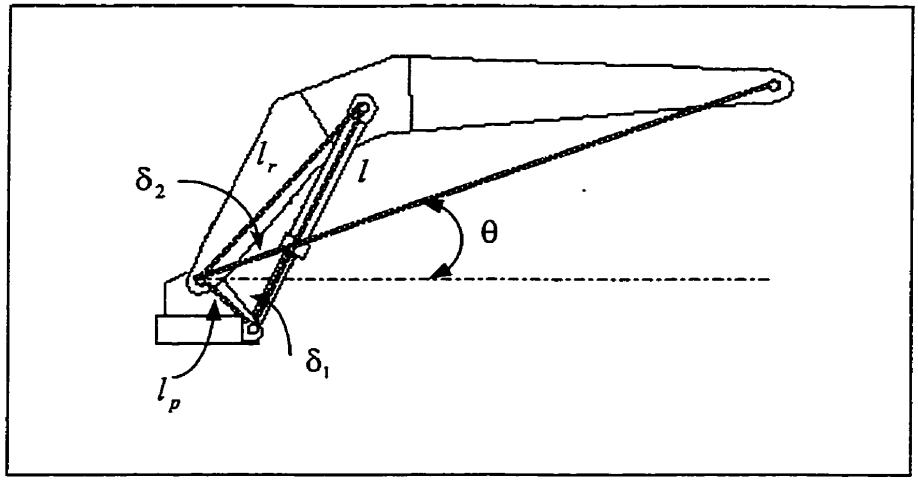


Fig. 2.4: Typical actuator-arm mechanism.

$$l^2 = l_p^2 + l_r^2 + 2l_p l_r \cos(\theta + \delta) \tag{2.5}$$

where $\delta = \delta_1 + \delta_2 = const.$ l_p, l_r, δ_1 and δ_2 are the kinematic parameters of the actuator-linkage mechanism and they are shown in Fig. 2.4. Taking the derivative of (2.5) yields:

$$2l\frac{dl}{dt} = -2l_p l_r \sin(\theta + \delta)\frac{d\theta}{dt} \tag{2.6}$$

$$\frac{dl}{dt} = \dot{X} = \frac{-l_p l_r \sin(\theta + \delta)}{\sqrt{l_p^2 + l_r^2 + 2l_p l_r \cos(\theta + \delta)}}\dot{\theta} \tag{2.7}$$

$$J(\theta) = \frac{-l_p l_r \sin(\theta + \delta)}{\sqrt{l_p^2 + l_r^2 + 2 l_p l_r \cos(\theta + \delta)}} \tag{2.8}$$

Figure 2.5a shows the function $J(\theta)$ over full joint angular range for boom and stick. Figure 2.5b shows the relationship between joint displacement, θ, and the piston displacement, X for boom and stick. As it is seen the relationship is almost linear.

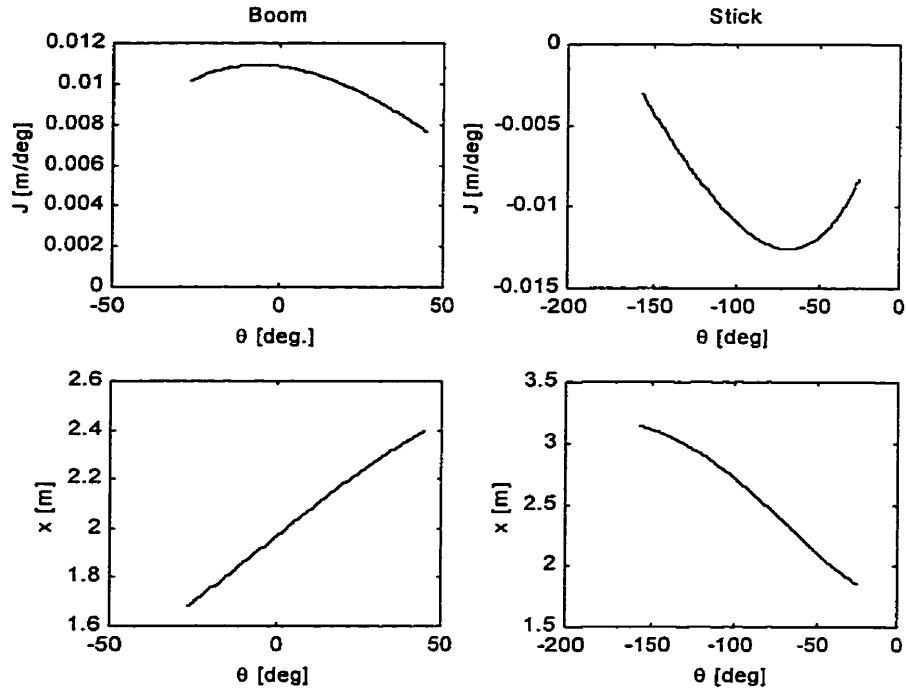


Fig. 2.5: (a) Function $J(\theta)$ over full range for Boom/Stick motion; (b) Curve $X(\theta)$ for Boom/Stick

2.1.2 Actuator Dynamics

The term "actuator dynamics" requires a clarification for hydraulic machines. Used in the context of traditional serial chain electric driven robots, it refers to the dynamics of all the elements responsible for moving the robot links, i.e., the motors and the gears. The hydraulic actuator dynamics, on the other hand refers to the dynamics of the entire hydraulic system responsible for moving the joints, i.e, the engine that supplies the

power, the hydraulic pumps, the valves used to regulate the fluid flow and the hydraulic cylinders and motors.

The motion of the manipulator presented here is governed by the coupled swing, boom and stick link dynamics and the actuator dynamics driving the links. The gripper link and its actuator are not considered, since only the positioning of the end-effector is of interest in this thesis.

The hydraulic actuator models include orifice flow equations, fluid compressibility equations, as well as the force balance equations for the cylinders. The orifice flow equations governing flow and pressure drop across an orifice is

$$Q = ka\sqrt{\Delta P} \tag{2.10}$$

where Q is the flow rate through an orifice, k is the orifice coefficient of discharge (a constant), a is the orifice area and ΔP is the pressure drop across the orifice. The valve orifice area a is a nonlinear function of spool displacement.

The fluid compressibility equation, which captures the dynamic of the hydraulic fluid is

$$\dot{P} = \frac{1}{C}(Q - Q'), \qquad C = \frac{V}{\beta} \tag{2.11}$$

where P is the pressure in a control volume, C is the hydraulic compliance of the flexible hoses connecting the valve to the actuator, β is the bulk modulus of the oil and V is the volume of oil in the control volume. Q' is related to the cylinder velocity $\dot{X}$:

$$Q' = \dot{X}A \tag{2.13}$$

where A is the piston area.

During the steady-state response Eq. (2.11) becomes

$$Q = Q' = \dot{X}A \tag{2.14}$$

The force balance equation for a cylinder is

$$m\ddot{X} = P_i A_i - P_o A_o - F - d\dot{X} - F_c \tag{2.15}$$

where m is the mass of the cylinder rod; P_i and P_o are the pressures in the two cylinder chambers; A_i and A_o are the piston areas on the two sides of the actuator; d is the viscous damping of the cylinder; F_c is the Coulomb friction of the cylinder

For a steady-state actuator response and neglecting the effect of the friction, the expression for the force F applied on the link becomes

$$F = F_a = P_i A_i - P_o A_o \tag{2.16}$$

F_a is the net force on the cylinder piston.

For the Cab swing actuator the governing equations are:

$$Q = D_m \dot{\theta}_m \tag{2.18}$$

$$\tau = D_m (P_1 - P_2) \tag{2.19}$$

where D_m is the volumetric displacement of the hydraulic motor and $\dot{\theta}_m$ is the motor shaft angular velocity. Q and τ are motor flow and torque, respectively. In practice, these equations may not be exact due to two sources of losses, leakage flow and internal friction. The torque τ generated by the hydraulic motor is transmitted through a gear train to rotate the upper structure (i.e, the Cab swing). The torque T, from motor referred to the output is:

$$T = n \cdot \tau \tag{2.20}$$

where n is gear reduction from motor to output shaft .The swing rotational velocity $\dot{\theta}_M$ is

$$\dot{\theta}_M = \frac{\dot{\theta}_m}{n} \tag{2.21}$$

2.2 Motion planning for manipulators

As the terms path and trajectory are used quite extensively in this thesis, it is appropriate to define them for clarity. The term "path" refers to a continuous curve in the Cartesian space connecting an initial and a final configuration. A "trajectory" is a continuous curve in state space connecting initial and final states.

2.2.1 Identification of Various Constraints

The most important constraints are derived from physical limitations of the robot system. They arise from the limited capabilities of the arm, such as maximum speed for each joint and limited force/torque output from each actuator. These are the ***actuator*** and ***linkage constraints***, which include the effect of loading. There are also, constraints imposed by the user, such as maximum allowed deviations from a pre-specified path at the tool tip, as in automated welding operations. This is called the ***path constraint***.

Another constraint is the ***collision constraint,*** which restricts the manipulator movement due to the potential collision with obstacles in the workspace. Collision or obstacle avoidance is an important part of motion planning, especially for industrial manipulators, which operate in restricted spaces. In the presence of fixed and stationary obstacles in the workspace, the path is given by the path planner. This path typically contains sharp corners and is not the minimum time path. An algorithm can be used to

find the minimum time path which lies close to the given one and which avoids the obstacles (Rajan, 1985).

For the mobile manipulators an additional constraint, namely the ***stability constraint***, should be considered. The stability constraint restricts the manipulator movement in order to maintain the stability of the machine.

2.2.2 Minimum-time motion planning

Computer-controlled robots are beginning to have a major impact on contemporary automation and manufacturing systems due to their potential for increasing productivity. To increase robot productivity robot motion speed should be improved. The robots speed and hence their productivity, are limited mainly by the capability of their actuators. Increasing actuator size and power is not a suitable solution due to increased inertia of the actuators themselves, and because of the increased cost and power consumption of larger actuators. A more feasible approach is to minimize the motion time needed to perform a given task subject to the actuator constraints.

At high speeds of robot motion a dynamic control strategy is required. With the dynamic control strategy, the instantaneous torque, velocity and position are controlled simultaneously. At lower speeds of robot motion, a kinematic control strategy, which controls only the position and, indirectly, the average velocity can be applied. A kinematic control strategy performs well at lower speeds of robot motion, while at high speeds, it works poorly. The reason is that at high speeds the hypothesis of static equilibrium at the points under consideration is less valid.

The control for manipulators is a very difficult problem due to manipulator nonlinear and coupled dynamics, physical constraints and the danger of colliding with other objects in the workspace. The control problem becomes even more complicated for hydraulic manipulators. For this type of manipulator additional nonlinearities are present in the actuator dynamics due to nonlinear pressure/flow relationships in the hydraulic valves and nonlinearity in the orifice due to dead zones.

Because of the difficulty with the control problem for the manipulators, the overall problem can be divided into two stages.

(1) offline optimal trajectory planning, which results in the prescription of the position, velocity and acceleration of each joint as a function of time along a given path (and open-loop optimal control trajectory).

(2) online path tracking, which tries to minimize the deviation of the actual trajectories from the desired ones using a feedback-control algorithm (e.g., linear quadratic regulator control law).

Over the last decade, a number of researchers have focused on solving the control problem for the manipulators. In their approaches they attempted to determine the control torques applied to each joint that drove a manipulator from a given initial state to a given final state in minimum time, subject to control torque constraints. This is the general point-to-point minimum time control (MTC) problem.

The minimum time control problem can be divided into two categories in terms of different constraints on motion: (1) the MTC problem with unconstrained motion paths between two endpoints; and (2) the MTC problem with constrained motion paths between

two endpoints, i.e., intermediate configurations of the robot arm are given so that the manipulator follows a specified path.

Minimum time control problem with unconstrained path

Roughly speaking, the approaches to solving the minimum time control problem with unconstrained motion path can be divided into two groups: the standard control theoretical approaches (e.g., employ Pontryagin's Minimum Principle) and the nonstandard approximation approaches, such as exhaustive search technique from artificial intelligence and nonlinear parameter optimization approaches (Chen and Desrochers, 1994).

With the manipulator dynamic equations and joint torque/ force constraints in suitable forms, the minimum time control problem can be addressed as an actual standard control problem. The usual method of solving such a problem is to employ Pontryagin's minimum principle. When the Pontryagin's principle is applied to this problem, it leads to a set of $4N$ coupled differential equations with a two point boundary value problem (N is the number of degrees of freedom). If the differential equations are linear, then the two-point boundary value problem can be solved numerically. However, the differential equations obtained when the Pontryagin's principle is applied to the robot arms are nonlinear. A set of coupled nonlinear differential equations with the two-point boundary value problem is not very tractable computationally. One way to overcome this problem is to linearize the differential equations. However, this is a drastic approximation, since

linerization of the dynamics of a robot arm leads to significant errors and therefore is unsatisfactory (Rajan, 1985).

The difficulties with the control theory approach to the minimum time control problem led researchers to apply an artificial intelligence approach to this problem. They discretized the state space and then performed an exhaustive search looking for the minimum time trajectory. The search was greatly simplified using different techniques which reduced the search space (Rajan, 1985; Shiller and Dobowsky, 1991). For example, Rajan proposed an approach with the following steps:

- Characterize the path in some manner
- Given a path, determine the minimum time trajectory subject to the actuator torque constraints
- Search among all possible paths to find the minimum time path, i.e., the path is varied until the path, which gives the shortest time among all possible paths is found.

Rajan characterized the paths using splines. The minimum-time trajectory was computed using the approach by Bobrow *et al.* (1985), and a gradient descent optimization technique was used to vary the spline parameters in the search for the minimum time path.

Minimum time control problem with constrained path

Bobrow *et al.* (1985) developed an algorithm to find the minimum time optimal motion for manipulators driven by electrical motors. The algorithm used both the full nonlinear dynamic characteristic of the manipulator and the constraints imposed by its actuators. The basic idea behind their work is that a pre-specified constrained motion path leads to

an overall motion with one degree-of-freedom (DOF) only, expressed by the path parameter $r(t)$. Thus, using the parameterization along the given path, the original dynamic equations of an N -degree-of-freedom manipulator can be transformed into a set of N nonlinear differential equations in terms of the path parameter. Then, applying the constraints on the torques to the parameterized equations, a set of constraints, or bounds on the second derivative of the path parameter $r(t)$ with respect to time (i.e., the pseudo-accceleration of the path) is obtained. The path parameter $r(t)$ and its first derivative $\dot{r}(t)$ (i.e., the pseudo-velocity of the path) are taken as the state vector and the pseudo-acceleration is thought of as the control variable. The problem then becomes a minimum time control problem for a double integrator system with state-dependent control constraints. The main idea of the solution is to select the control variable that produces the largest pseudo-velocity profile such that, at each point on the path, the pseudo-velocity is not greater than the maximum velocity at which the constraint on the motion will not be violated. The solution is constructed in terms of switching curve in the phase plane ($r-\dot{r}$ plane). The control torques for individual joints can be determined by a set of N -parameterized equations of the robot dynamics once the path parameter $r(t)$, the pseudo-velocity $\dot{r}(t)$ and the pseudo-acceleration $\ddot{r}(t)$ are obtained. Therefore, the switching curve indirectly provides a graphical representation of the feedback control law for MTC problem. However, the resulting bang-bang strategies are physically unrealizable due to the typical discontinuities at the switching times and the non-negligible actuator dynamics. In addition, such strategies are undesirable due to the structural vibrations induced by the control discontinuities, and due to the damage to the

permanent magnets of the electric motors caused by the abrupt changes in the motor current (Shiller, 1994).

The approaches mentioned above focused on manipulators driven by electrical motors. They attempt to exploit the complete robot linkage dynamics model and simplified actuator models to find the minimum time motion. A simplified actuator model does not include the dynamics of the actuators. For example, the complete mathematical model of an electric DC motor is:

$$\boldsymbol{V} = \boldsymbol{Ri} + \boldsymbol{L}\frac{\boldsymbol{di}}{\boldsymbol{dt}} + \boldsymbol{K}_e\dot{\theta} \tag{2.22}$$

$$\boldsymbol{T} = \boldsymbol{K}_t\boldsymbol{i} \tag{2.23}$$

where $\boldsymbol{V}$ is a vector of applied voltages, $\boldsymbol{i}$ is a vector of the motor currents, and $\boldsymbol{R}, \boldsymbol{L}, \boldsymbol{K}_e, \boldsymbol{K}_t$, are diagonal matrices with the motor parameters: resistance, inductance, back EMF constant, and torque constant, respectively. Therefore, in a simplified actuator model Eq. (2.22) becomes:

$$\boldsymbol{V} = \boldsymbol{Ri} + \boldsymbol{K}_e\dot{\theta} \tag{2.24}$$

However, the actuator dynamics are not negligible, and therefore some practical difficulties have been encountered using these approaches. For example, the actuator torques required for a typical optimal trajectory are discontinuous functions of time. In practice, due to the delays caused by the actuator dynamics, it is impossible to produce these torques exactly. Furthermore, the transient response of the neglected actuator dynamics might cause tracking errors in typical path following applications (Tarn *et al.*, 1991). This is even more so in applications requiring high accuracy at high speeds (Youcef-Toumi and Kuo, 1987). Some of these practical difficulties can be addressed by

modeling the actuator dynamics (see for example Tarkiainen and Shiller, 1993). This, however, increases the dimensionality of the model and the required computational effort.

The methods in the literature that allow simplified actuator models assume a known torque limit curve for each joint actuator (see for example Bobrow *et al.*, 1985). This torque limit curve is a function of the velocity only, i.e., the actuator torques are all independent of one another. However, the torque limit curve is not easy to compute ahead of time for a heavy-duty hydraulic manipulator, since the limit curve is a function of many variables due to coupled actuation in this type of manipulator. For instance, the force limit for the stick actuator on the excavator-based machine is dependent on a number of swing circuit variables, in addition the stick circuit variables. Therefore, minimum time motion planning for hydraulic robots requires a different approach from those seen in the literature.

Lin *et al.* (1983) developed a method to find the minimum time required to move along a pre-specified path. The method was not concerned with finding the optimal control to execute the trajectory computed. In their method cubic spline functions were used for generating the joint trajectories. The resulting spline functions were expressed in terms of time intervals between adjacent knots. The downhill simplex method was adopted to schedule the time intervals between each pair of the adjacent knots such that the total motion time was minimized subject to the constraints on joint velocities, accelerations and jerks. Therefore, with this method an optimal trajectory was computed off-line and the control torque remained to be computed on-line. The drawback of the method is that it uses fixed joint velocities, accelerations and jerks limits.

Since the fixed limits upon the joint velocities and accelerations, must cover all possible configurations of the robot, they may be too conservative. For example, an optimal path for a manipulator has to be generated on the basis of the maximum velocity and acceleration allowed under the worst (global) conditions. This implies that the motion planning has to be made with the global least upper bounds of all possible manipulator's velocities and accelerations. Therefore, fixed joint velocities, accelerations and jerks constraints do not account for the actual capabilities of the manipulator actuators and the actual dynamic load upon them. They only consider some rough estimates of the allowable joint velocities and accelerations. Therefore, the estimates of the minimum time may be rough.

In this thesis the method developed by Lin is improved in that instead of fixed values for joints maximum velocities, variable values are computed using the dynamic model of the manipulator.

For the heavy-duty hydraulic manipulators under investigation there is a very complex relationship between joint velocities and dynamics. Each joint velocity is related to the flow directed to its corresponding actuator as:

$$Q_d = \dot{\theta}_d \cdot J(\theta) \cdot A \tag{2.25}$$

where Q_d is the desired flow to the actuator; $\dot{\theta}_d$ is the desired joint velocity. Therefore, given the desired joint velocities the flow rates to the hydraulic actuators required for achieving these velocities can be determined. Further, it should be checked whether the desired flow rates can be achieved. The computation of the flow distributions to the actuators is complicated due to the coupled actuation in the hydraulic system.

Due to the hydraulic circuit interdependency, the flow from limited power variable-displacement pumps is supplied to the valves on a priority basis, i.e., the valve closest to the pump gets all the flow it requires, and the remaining flow is distributed among the rest of the valves. On the other hand, when multiple actuators request flow simultaneously, the power demanded exceeds the power output of the engine and the pumps reduce their flow rates to keep the engine from getting overloaded. The pumps flow-rate changes according to the pump output pressure, which is load dependent. Thus, the joint velocities may not be achievable as a result of these constraints and the joint velocity limits need to be calculated and updated. For example, the maximum joint velocity for swing is limited to the pump flow, which itself is variable and load dependent. The maximum velocity that the stick could achieve depends on the remaining flow that has not been consumed by the swing. Therefore, the maximum achievable joint velocities are variable for hydraulic machines and depend on the required number of simultaneous joint motions and loading.

In the following chapter, Lin's method is extended in that the joint velocity bounds are determined using the linkage model and a simplified model of the actuators driving the links. Determining these bounds any joint velocity profile can be scaled to make full use of the actuator capabilities. Lin's method computes only the optimal trajectory (position, velocity and acceleration of each joint as a function of time); the method to be presented computes the optimal trajectory and open-loop control trajectory.

Chapter 3

Minimum-Time Trajectory Planning

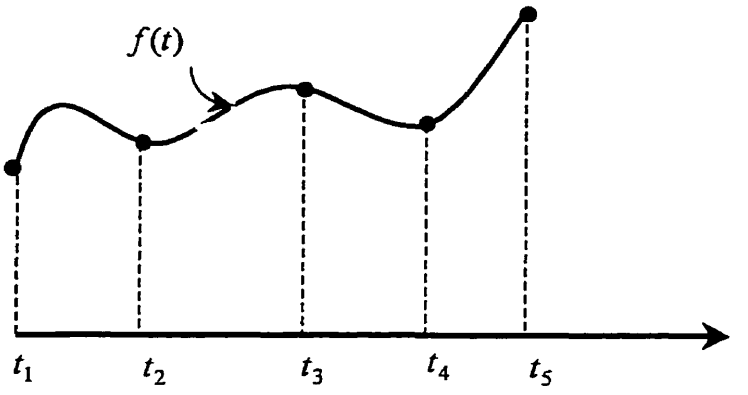
The method developed here is particularly suited to finding the minimum time for a specified path. The work is focused on planning manipulator motions, which advantageously use the actuator capabilities to optimize the motion time needed to perform a task for which the path is specified. The optimization is performed with respect to the combination of actuators and linkage effects. This is especially important for hydraulic manipulators where the actuator effects are very significant.

To perform the search for the optimal solution, a simple downhill simplex algorithm is implemented. The algorithm does not require gradient information; it only uses function evaluations to move towards the optimum solution. This is important since gradient information is not readily available.

3.1 Joint trajectories generation and time scaling

3.1.1 Formulation of spline function joint trajectories

In this section joint trajectories are generated using spline functions. A spline function is a way of passing a smooth curve through a given set of points in space (hereafter these points are referred to as knots). In mathematical terms, a smooth curve is continuously differentiable.



A cubic spline $f(t)$ is a continuous function that coincides, in each partial interval (the interval between adjacent knot points), with a cubic polynomial, and at every knot point the spline $f(t)$ itself and its first and second derivatives are continuous. If at each knot, both the slope and curvature of the cubic pieces to either side will match, the curve will be twice continuously differentiable. If the slopes of the two polynomial segments to either side of each knot will match at that knot, but their curvatures will not, the curve will be continuously differentiable, but in general not twice continuously differentiable. Spline functions have been found to provide the shortest path that passes through the knots while satisfying the continuity constraints. A rigorous proof of this property may be found in Ahlberg *et al.* (1967), where it is shown that cubic splines possess the "best approximation property" which minimizes $[\int f''(t)^2 dt]^{1/2}$ along the curve, thereby minimizing the norm value of curvature.

The procedure for generating joint trajectories is as follows. The path is specified by a sequence of Cartesian knot points $E_1, E_2, \ldots, E_n$. These n Cartesian knot points are converted into n knot points in joint space using inverse kinematics. For each joint, the data specifying the trajectory are position, velocity and acceleration at the initial and the

final points and the positions or joint displacements at intermediate knot points. The trajectory can be generated using polynomials between the knot points. These polynomials are splined together by requiring that the position, velocities and accelerations are continuous at the junction points. This is necessary for a manipulator to achieve a smooth motion. Therefore, the sequence of these polynomials splined to each other describes the desired trajectory that will generate a smooth motion of the manipulator. The trajectory planning is performed in joint space because the limits on the manipulator's performance are expressed primarily in this space.

The order of the polynomials is decided considering the trajectory specifications. Each intermediate knot point imposes four constraints: two position constraints, as each of the splines is required to pass through the knot points, and two constraints to guarantee continuity on velocity and acceleration. Four constraints can be met by a third-order polynomial. The first and last segments need to satisfy five constraints, three at start- and end-points, since the velocity and acceleration at these points should be zero, and two from the nearest intermediate points. Five constraints are satisfied using a fourth-order polynomial. Thus, to develop a trajectory through n knot points n-1 polynomials are required: *a fourth-order* polynomial for the *first segment*, *n-3 third-order* polynomials for the *intermediate segments*, and *a fourth-order* polynomial for the *final segment*.

Thus, to construct the joint trajectories, n Cartesian knots are first transformed into joint vectors $[\theta_{11},\theta_{21},...,\theta_{N1}]$, $[\theta_{12},\theta_{22},...,\theta_{N2}],...,[\theta_{1n},\theta_{2n},...,\theta_{Nn}]$, where θ_{ji} is the displacement of joint j at knot i corresponding to E_i. In this section the procedure of constructing joint trajectories deals with one joint at a time. The joint number j is not necessary to be specified, and hence θ_{ji} is replaced by θ_i for simplicity.

Assuming that the joint knot points are arranged as $\theta_1, \theta_2, ..., \theta_n$, corresponding to the instants $t_1, t_2, ..., t_n$, respectively. The problem is to find a set of polynomials to join these knot points. Intermediate segments require a third-order polynomial of the form:

$$\Theta(t) = B_4 t^3 + B_3 t^2 + B_2 t + B_1 \tag{3.1}$$

where $\Theta(t)$ represents the position of the joint as a function of a parametric variable t. The first and final segments require fourth-order spline segments of the form:

$$\Theta(t) = B_5 t^4 + B_4 t^3 + B_3 t^2 + B_2 t + B_1 \tag{3.2}$$

The spline coefficients are determined function of the intervals h_1, $h_2, ..., h_{n-1}$, where $h_i = t_{i+1} - t_i$, and the slope of the curve at the intermediate points where two polynomials are splined. Using the constraints at the intermediate points, a set of equations is generated that determines the slopes of the curve at the intermediate points. The slopes are determined in terms of time intervals h_i's and the given values of joint displacements. The calculation of the spline coefficients defining each segment of the trajectory is shown in Appendix B.

Calculating the spline coefficients, the entire joint trajectory $\theta(t)$ is determined. Note that t is not a physical time, it is a parameter, which varies from $t=0$, at the initial configuration, to $t = t_f$ at the final configuration.

A new trajectory $\theta^*(t)$ will be defined such that $\theta^*(t) = \theta(r)$, where $r = r(t)$ is a monotonically increasing function of time, with $r(0) = 0$ and $r(T_f) = t_f$ for some $T_f > 0$. Thus one needs to determine the function $r(t)$, which describes how fast the arm

moves along the path. Function $r(t)$ must increase monotonically because time cannot reverse itself, and $r(0) = 0$, because the movement must start at the same point.

Using chain rule one obtains:

$$\dot{\theta}^*(t) = \theta'(r)\dot{r}(t) \tag{3.3}$$

where the dot notation is used for time derivatives and $\theta'(r) = d\theta(r)/dr$. Similarly,

$$\ddot{\theta}^*(t) = \theta''(r)\dot{r}(t)^2 + \theta'(r)\ddot{r}(t) \tag{3.4}$$

One method to optimize the motion for a specified path is to determine the function $r(t)$ such that the time expired in going from the initial configuration to the final configuration is minimized.

A faster way to get there from here is either to "run" at maximum velocity wherever possible or else accelerate as much as possible and then decelerate as much as possible (bang-bang theorem). In the algorithm developed by Bobrow *et al.* (1985), Eqs. (3.3) and (3.4) were substituted into the dynamic equation (2.1) and the torque constraints expressed for all N joints:

$$T^i_{\min}(\theta,\dot{\theta}) < T_i < T^i_{\max}(\theta,\dot{\theta}) \tag{3.5}$$

are transformed into a set of lower and upper bounds on the pseudo-acceleration $\ddot{r}(t)$, i.e, for all N joints, $\max_i LB_i \le \ddot{r} \le \min_i UB_i$, or $GLB(r,\dot{r}) \le \ddot{r} \le LUB(r,\dot{r})$. The path parameter $r(t)$ and its first derivative $\dot{r}(t)$ are taken as the state vector and the pseudo-acceleration is thought of as the control variable.

The difference between Bobrow *et al.'s* algorithm and the methods, which use constant bounds on the acceleration, can be seen in terms of the equations above. Assume that the parameter r is arc length in Cartesian space. Then $\dot{r}$ is the speed and $\ddot{r}$ the

acceleration along the geometric path. Therefore, one would have $GLB(r,\dot{r}) \le \ddot{r}_{\min} \le \ddot{r} \le \ddot{r}_{\max} \le LUB(r,\dot{r})$, where $\ddot{r}_{\min}$ and $\ddot{r}_{\max}$ are constants. The methods, which use constant bounds on the acceleration, then, restrict the acceleration more than is really necessary. Likewise, constant bounds on the velocity will also be more restrictive than necessary.

Bobrow *et al.'s* algorithm is not directly applicable to hydraulic manipulators since the actuator constraints cannot be expressed as simple torque limit curves. However, the algorithm to be presented is based on the same idea of determining the bounds on the velocity-for the hydraulic manipulator considered here-using the dynamic characteristics of the manipulator and the constraints imposed by its actuators.

3.1.2 Constant time scaling

Hollerbach (1984), suggested for function $r(t)$ the following expression:

$$r(t) = ct \tag{3.6}$$

for some constant $c > 0$. If $c > 1$ the movement speed ups; if $c < 1$ the movement slows down. Equation (3.3) then becomes:

$$\dot{\theta}^*(t) = c\theta'(ct) \tag{3.7}$$

The above expression for $r(t)$ does not give the minimum motion time; it just gives the time necessary to execute the motion with allowable speeds of movements. By allowable speed it is meant that the trajectory is stretched or compressed uniformly to fit the appropriate duration without changing the path or the velocity profile shape. Constant scaling of velocity is a simple but important method of bringing a trajectory within actuator constraints (Hollerbach, 1984).

The following section presents the algorithm, which computes the scaling value c. The algorithm determines allowable speeds of movements for the manipulator, so that the actuator constraints are not violated. For the hydraulic manipulators under consideration these constraints are due to the hydraulic power limitations.

3.2 Time scaling of trajectories to satisfy actuator constraints

3.2.1 Maximum achievable joint velocities

The hydraulic system considered here consists of two main lines (see Fig. 3.1). One line consists of boom main valve, the other line consists of swing and stick main valves. In this simplified mode of operation boom and stick cross-overs are not included.

Depending upon the maximum available pump flow, Sepehri *et al.* (1990) has proposed an algorithm which determines whether the joint velocities specified for a motion are achievable or not. Basically the algorithm checks the desired flow rates against two constraints to assure that the system can deliver the required power.

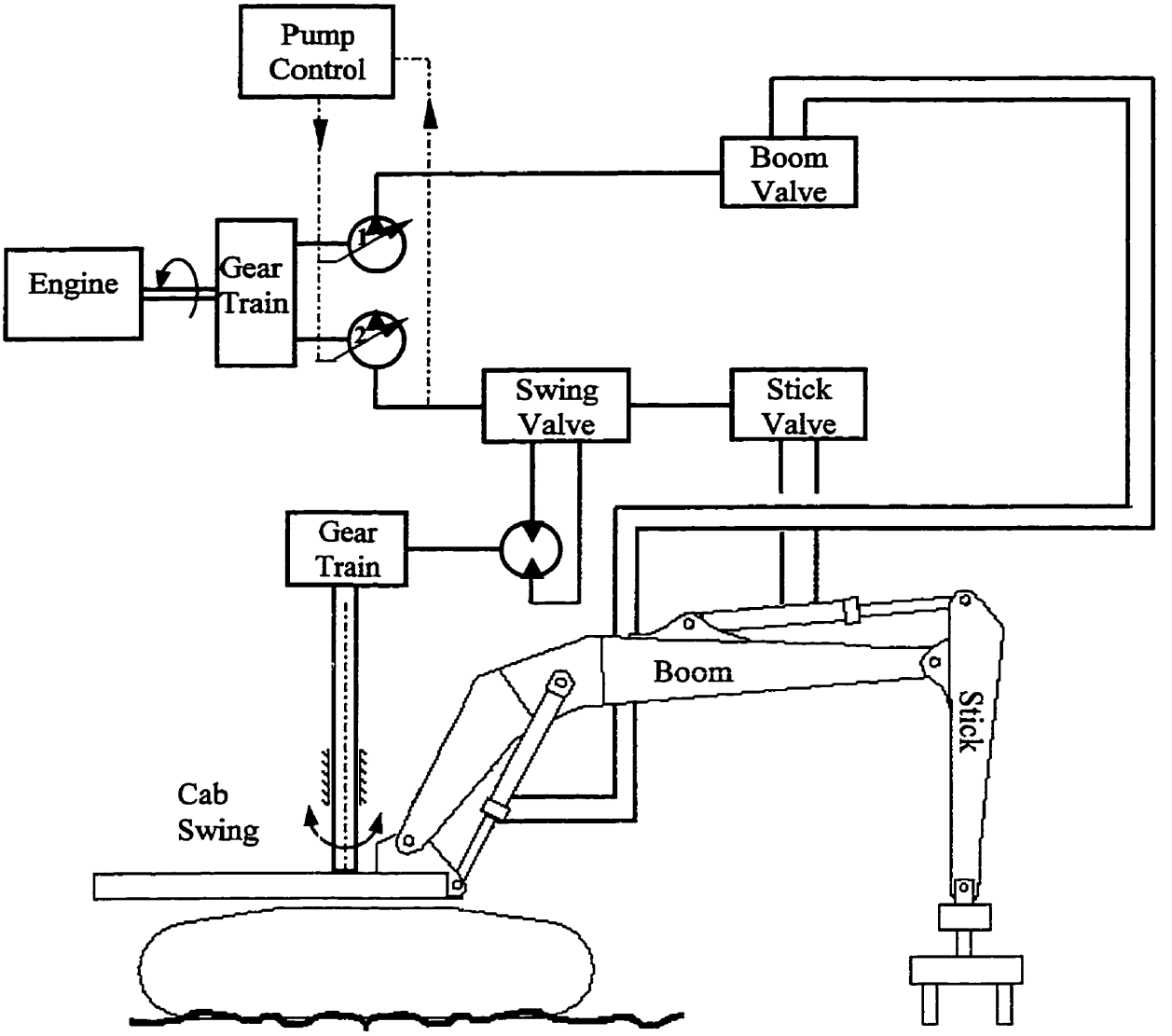


Fig. 3.1: Excavator-based machine without cross-overs.

1. *Maximum Availability Constraints*

When the desired flow to any of the actuators exceeds the maximum available flow to the respective actuators, the flow rates to all the actuators are reduced on the basis of a scaling factor k_1. Thus, the scaling factor k_1, should satisfy the following constraints namely:

$$Q_{Bo} \leq Q \tag{3.8}$$

$$Q_{Sw} \leq Q \tag{3.9}$$

$$Q_{St} \leq Q \tag{3.10}$$

where Q_{Bo}, Q_{Sw}, Q_{st} are the desired flows to the boom, swing and stick actuators; Q is the maximum available flow from the pumps. Any violation of the above constraints will require modifying the fluid flows by proportionally scaling down all flow-rates, on the basis of the scaling factor k_1. k_1 is the smallest scaling factor as determined after examining all the above constraints. If there is no violation of the constraints, the scaling factor k_1 is set at 1; i.e., there is no modification imposed on the flows.

2. *Interconnection Constraints*

For the excavator-based machine shown in Fig. 3.1 some actuators have higher priority than others. For example, the swing valve takes priority over the stick valve. In this case, a second scaling factor k_2, in which such priorities are considered, may also be applied to the flows to reduce the relative speeds of all actuators. Therefore, the flow-rates modified by scaling factor k_1, should satisfy the following constraint as well:

$$Q - Q_{sw} \geq Q_{st} \tag{3.11}$$

If the priority constraint is not violated, the scaling factor k_2 becomes unity. A total scaling factor k is then obtained by combining both k_1 and k_2.

Thus, the above procedure provides a scaling factor k. $k=1$ means the desired joint velocities are achievable. $k<1$ means the desired joint velocities are not achievable and should be scaled down. Multiplication of k with the desired joint velocities will give the maximum achievable joint velocities. The scheme given in Fig. 3.2 shows the steps taken for computing the scaling factor k.

The algorithm presented requires computation of the flow available from the pumps. The pumps change their output flow according to the sum of the pressures in the implement circuits ($P_1 + P_2$), so that the pressure-flow curve is below the engine power limit curve. Therefore, based on the required supply pressures P_1, P_2, the flow, which can be delivered by the pumps, is determined from the pressure-flow limit curve.

The following section describes the algorithm, which computes the required supply pressures and therefore the maximum available flow from the pumps, using a model of the manipulator. The model of the manipulator consists of the model of the linkage and the model of the actuators driving the manipulator joints.

Maximum available flow from the pumps

The solution for the response of even the simplified hydraulic system, shown in Fig. 3.1, involves the simultaneous solution of orifice equations, compressibility equations and force balance equations for each cylinder. An important approximation is made in this approach, i.e, the actuator response is approximated by the *steady-state* actuator response.

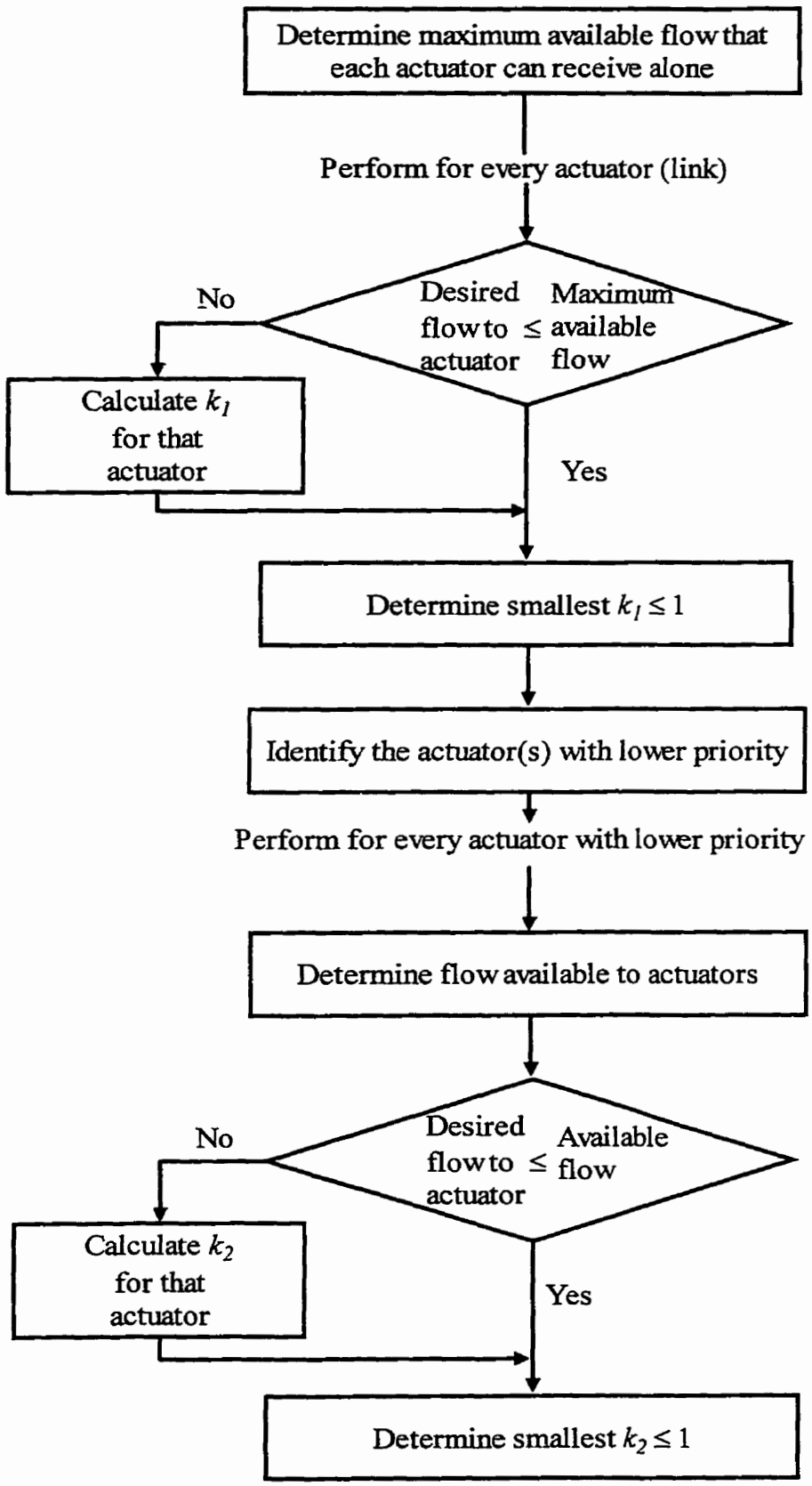


Fig. 3.2: Scheme for computing scaling factor $k = k_1 k_2$.

A steady-state solution would not include the fluid dynamics, Eq. (2.11). However, the cylinder forces are not restricted to steady-state forces, they are computed using the dynamic model of the linkage. This approximation is made since a solution for the complete actuator dynamics requires specification of an additional set of variables $(\dot{P}, C)$, which greatly increases the dimensionality of the model and the required computational effort.

The linkage model is used to compute the required actuation forces applied on each link for a given trajectory. The actuation forces are used in the actuator model to obtain the pump pressures P_1, P_2. With the summing pressure $(P_1 + P_2)$, the maximum output flow from the pumps is determined from the pressure-flow limit curve. The pressure-flow limit curve is shown in Fig. 3.3. The remainder of this section presents the

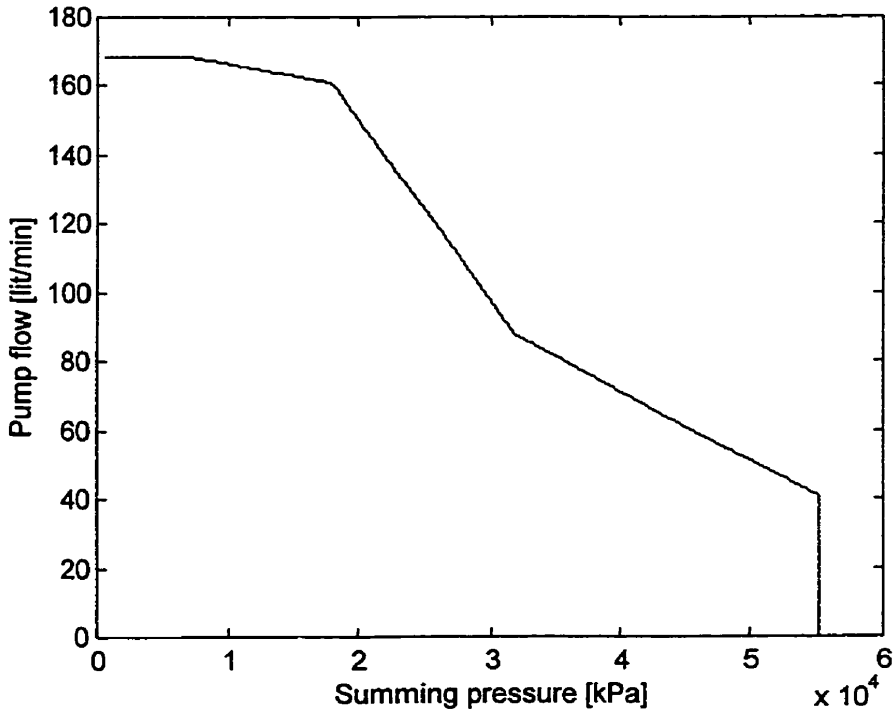


Fig. 3.3: Pumps output flow versus summing pressure $P_1 + P_2$.

details of these computations. The maximum output flow is, first, determined for one-link actuation, i.e, boom actuation. Next, it is determined for two-links actuation with priority action, i.e., swing and stick actuation. Finally, the maximum output flow is computed for three-links actuation.

a) One-link actuation

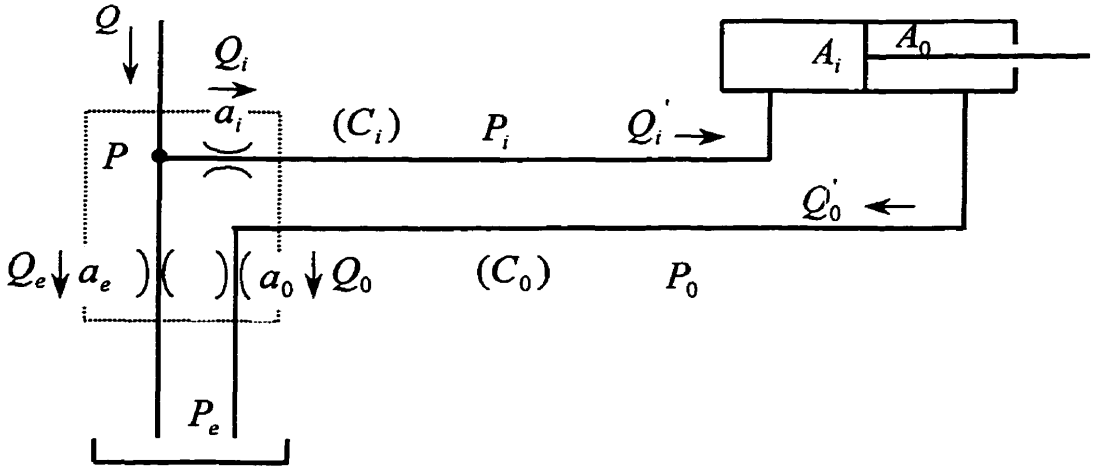


Fig. 3.4: Single-link actuation.

The actuator is connected to an open center main valve through compliant hoses. The valve monitors the flow to (Q_i) and from (Q_o) the actuator. The supply pressure (P_1) is provided by the pump, which is run by the engine. The orifice areas a_i, a_o, a_e are controlled by the displacement in the main valve spool (x_{sp}). Thus, a single spool position can represent all the orifice area variables for a joint (see Fig 3.5). When the spool is in a neutral position, orifices a_i and a_o are closed and the pump flow returns to the tank through the open orifice, a_e, and at atmospheric pressure (P_e). Q_e is the pump flow to the tank.

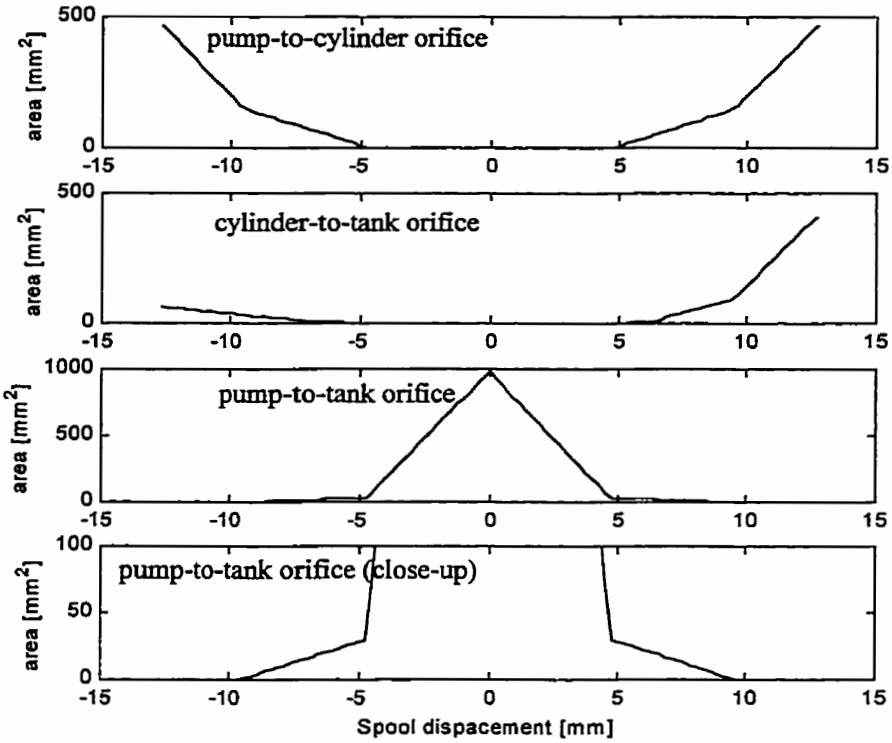


Fig. 3.5: Variation of orifice areas versus spool displacement for actuator.

With reference to Fig. 3.4, the equations describing the actuator model during steady state response can be written as follows:

(To the actuator):

$$Q = Q_i + Q_e \tag{3.12}$$

$$Q_i = ka_i\sqrt{P_1 - P_i} \tag{3.13}$$

$$Q_i = \dot{X}A_i \tag{3.14}$$

$$Q_e = ka_e\sqrt{P_1 - P_e} \tag{3.15}$$

(Out of the actuator):

$$Q_o = ka_o\sqrt{P_o - P_e} \tag{3.16}$$

$$Q_o = \dot{X} A_o \tag{3.17}$$

The force generated by the actuator is:

$$F = P_i A_i - P_o A_o \tag{3.18}$$

The equations describing the hydraulic system are nonlinear and an iterative method is applied to solve them. The flow chart, given in Fig. 3.6, shows the steps taken for solving the hydraulic system equations, in order to determine the maximum available pump flow, Q. For the joint position θ, velocity $\dot{\theta}$, acceleration $\ddot{\theta}$, the actuation force F and the cylinder velocity $\dot{X}$ can be determined using Eq. (2.3), Eq. (2.7) respectively.

At the beginning of the iterative procedure the spool displacement is initialized, $x_{sp} = 0$.

$$a_i, a_o, a_e = f_m(x_{sp}) \qquad m = i, o, e \tag{3.19}$$

Equating (3.16) with (3.17) the output line pressure is determined as:

$$P_o = \left(\frac{\dot{X} A_o}{k a_o} \right)^2 + P_e \tag{3.20}$$

The input line pressure is determined from Eq. (3.18) as:

$$P_i = \frac{F + P_o A_o}{A_i} \tag{3.21}$$

Equating (3.13) with (3.14) the supply pressure P_1 is determined as:

$$P_1 = \left(\frac{\dot{X} A_i}{k a_i} \right)^2 + P_i \tag{3.22}$$

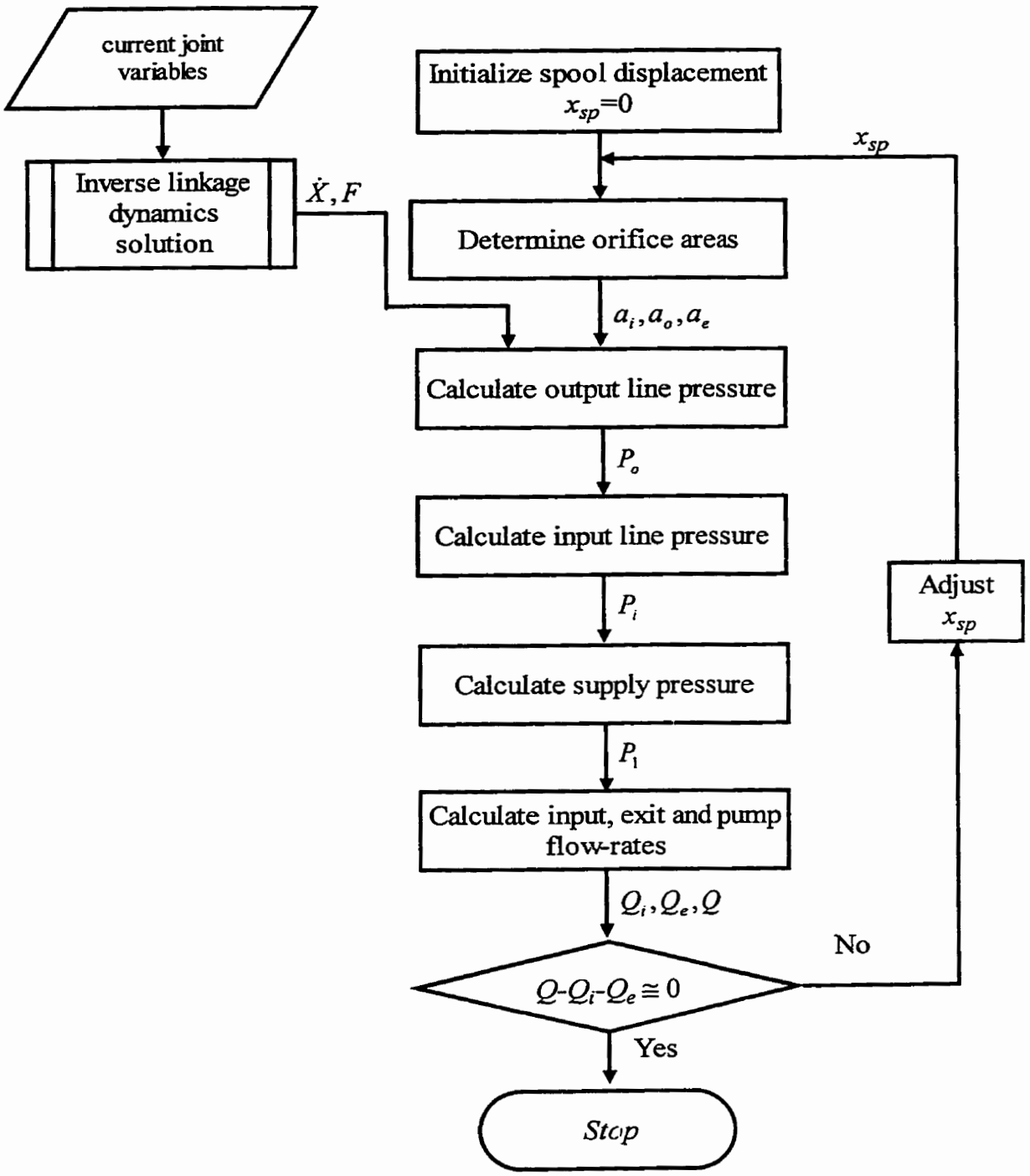


Fig. 3.6: Algorithm for solving boom actuator equations.

The pump flow changes according to the sum of the supply pressures P_1, P_2 as shown in Fig. 3.3. Supply pressure P_2 is considered for this case constant and equal to the tank pressure. Therefore, the available pump flow, Q is determined as:

$$Q = f(P_1 + P_2) \tag{3.23}$$

The spool displacement is incremented, until Eq. (3.12) is satisfied:

Therefore, for the spool displacement x_{sp}, for the actuation force F and desired actuator velocity $\dot{X}$, the maximum flow from the pump Q can be determined for a steady-state condition.

To show the actuator response for one-link actuation (i.e boom), the following manipulator motion is simulated: the manipulator is moved from a start position of (boom=-27^0) to a goal position of (boom=45^0). The simulation results are shown in Figs. 3.7 to 3.11.

Given a start position and goal position the joint trajectories are generated using trigonometric series of the following form (Hornick and Ravani, 1986):

$$\theta(t) = \theta_i + (\theta_f - \theta_i)(t/T) + \sum_{n=1}^{k} \frac{a_n T}{n^2 \pi^2} \sin \frac{n\pi t}{T} \tag{3.24}$$

where k is chosen equal to 4. Joint trajectories are generated using the above trigonometric series only in this section, in which the main purpose is to show how the hydraulic system equations are solved. They have been chosen for their simplicity.

The joint trajectories are shown in Fig. 3.7. For the joint position $\theta(t)$, velocity $\dot{\theta}(t)$, acceleration $\ddot{\theta}(t)$, the required joint torque can be computed using linkage dynamics (2.1). Figure 3.8 shows the required joint torque for the given trajectory. The force applied on the link by the actuator is shown in Fig. 3.8b.

To achieve the joint velocity $\dot{\theta}$, the cylinder piston should be moved by the fluid with a velocity $\dot{X}$. The cylinder velocity profile is shown in Fig. 3.9a. The flow-rate required to achieve this velocity is shown in Fig. 3.9b. When the spool is displaced, orifices a_i and a_o open and the orifice a_e start closing. This results is a rise in the supply pressure due to restricting the pump flow, until it exceeds the line pressure (P_l) allowing fluid flow into the actuator. When the supply pressure becomes high enough, the pump reduces its output flow to prevent the engine from stalling. Fig. 3.11b shows that the pump flow reduces as the required supply pressure increases, as shown in 3.11a. The line pressures P_i, P_o are shown in Fig. 3.10.

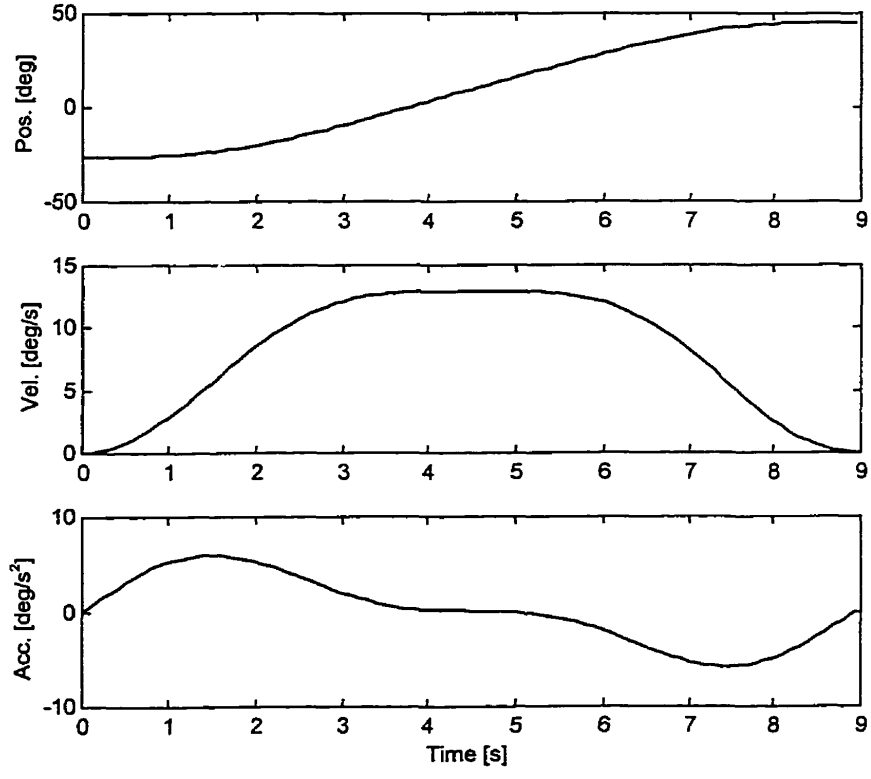


Fig. 3.7: Boom position, velocity, acceleration profiles.

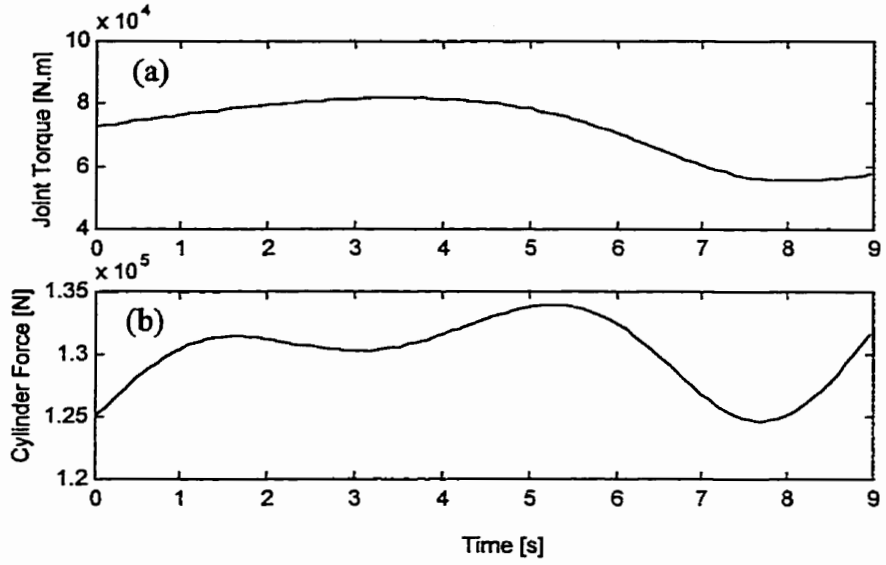


Fig. 3.8: (a) Boom torque; (b) boom hydraulic force.

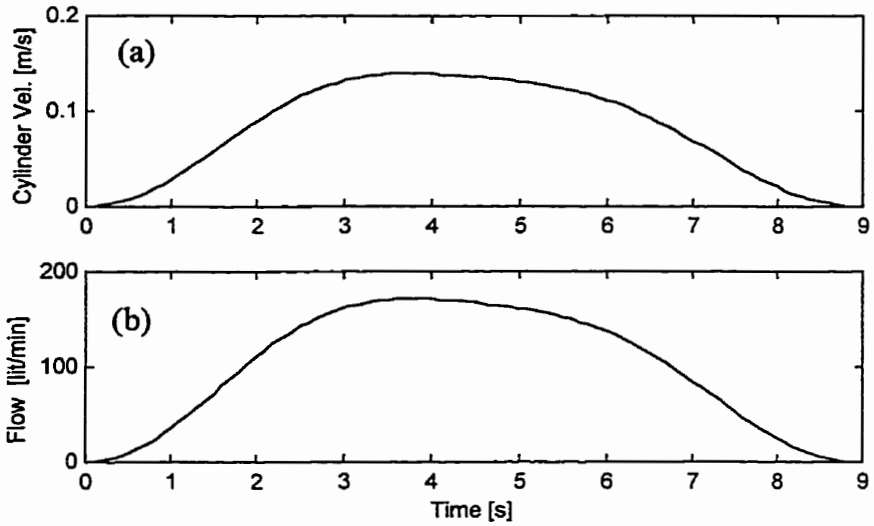


Fig. 3.9: (a) Boom cylinder velocity; (b) Desired flow to the boom.

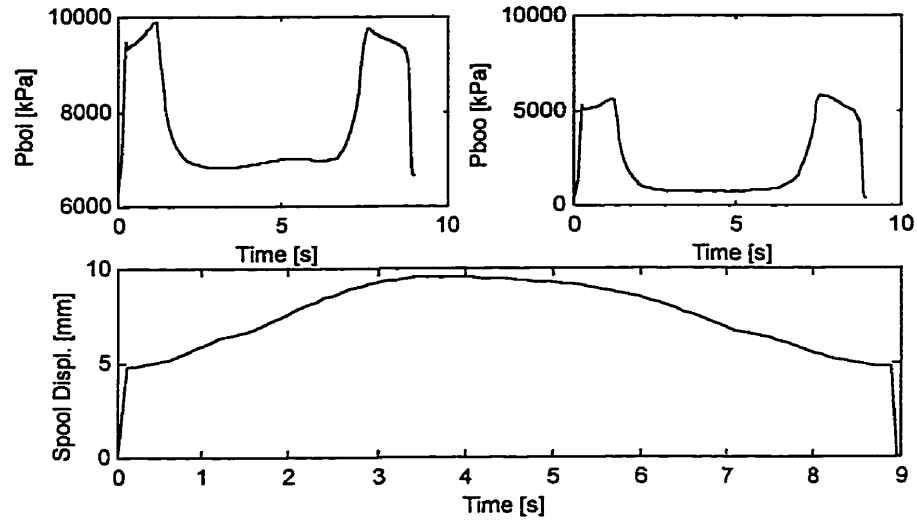


Fig. 3.10: Boom line pressures and boom spool displacement.

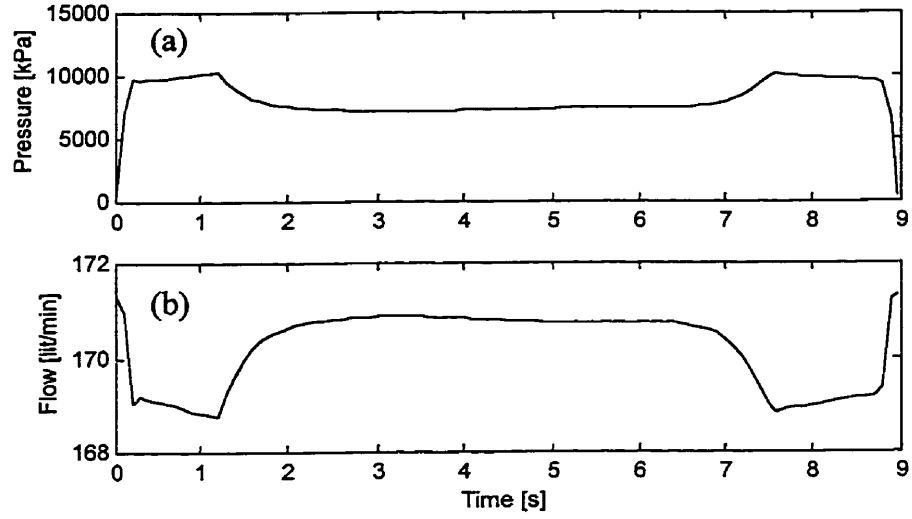


Fig. 3.11: (a) Summing pressure, $P_1 + P_2$; (b) Maximum available pump flow-rate.

b) Two-link actuation with priority action

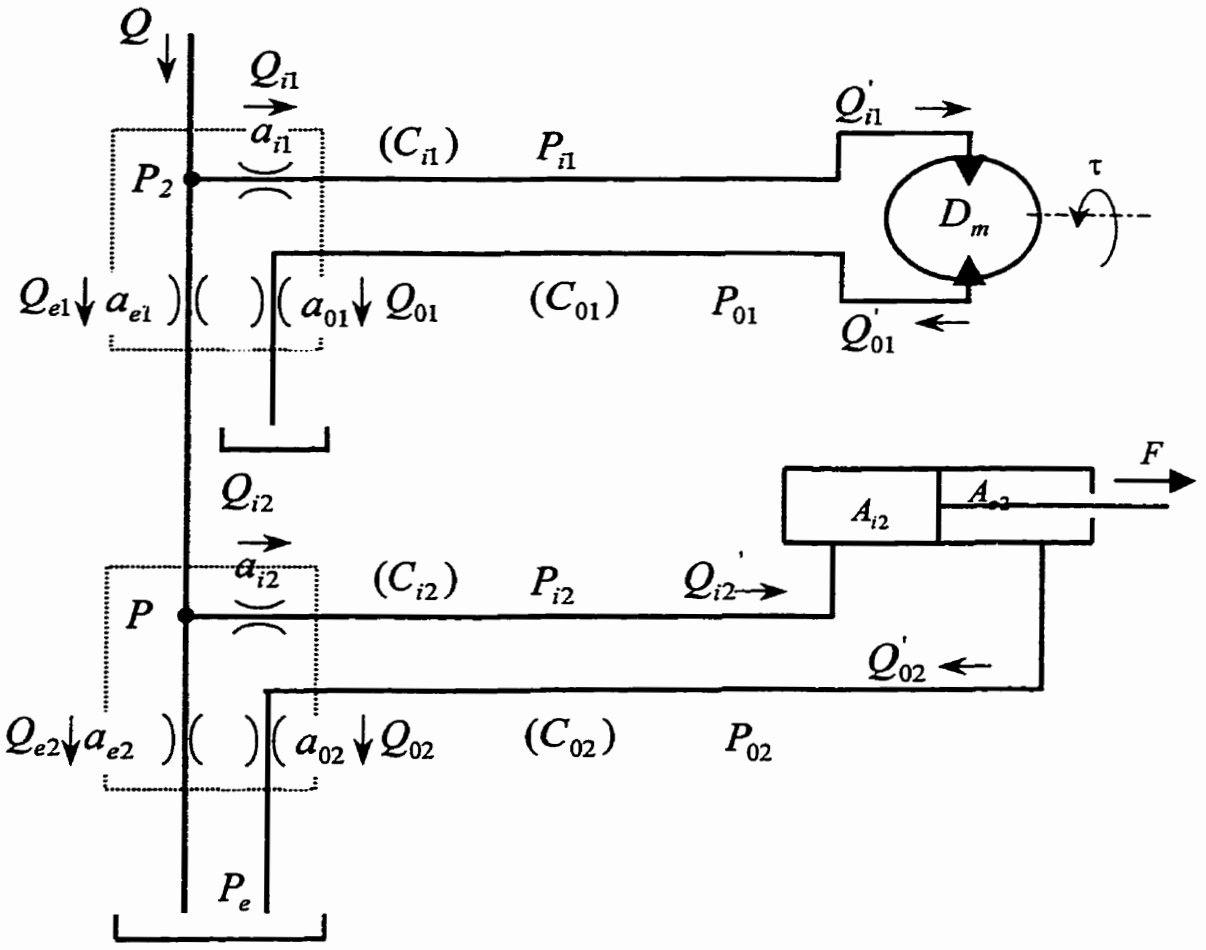


Fig. 3.12: Two-link actuation with priority action.

Figure 3.12 shows two open-center valves connected in series, which control the flow to two hydraulic actuators (a motor and a cylinder). This hydraulic system is similar to the swing and stick actuation system. The pump supplies the hydraulic flow on a priority basis, first to the swing valve, then to the stick valve. The equation describing the hydraulic system can be written as follows:

(To the actuator 1):

$$Q = Q_{i1} + Q_{e1} \tag{3.25}$$

$$Q_{i1} = ka_{i1}\sqrt{P_2 - P_{i1}} \tag{3.26}$$

$$Q'_{i1} = Q_{i1} = n \cdot \dot{\theta} \cdot D_m \tag{3.27}$$

$$Q_{e1} = ka_{e1}\sqrt{P_2 - P} \tag{3.28}$$

(Out of the actuator 1):

$$Q_{o1} = ka_{o1}\sqrt{P_{o1} - P_e} \tag{3.29}$$

$$Q'_{o1} = Q_{o1} = n \cdot \dot{\theta} \cdot D_m \tag{3.30}$$

(To the actuator 2):

$$Q_{e1} = Q_{i2} + Q_{e2} \tag{3.31}$$

$$Q_{i2} = ka_{i2}\sqrt{P - P_{i2}} \tag{3.32}$$

$$Q_{i2} = \dot{X} \cdot A_i \tag{3.33}$$

$$Q_{e2} = ka_{e2}\sqrt{P - P_e} \tag{3.34}$$

(Out of the actuator 2):

$$Q_{o2} = ka_{o2}\sqrt{P_{o2} - P_e} \tag{3.35}$$

$$Q_{o2} = \dot{X} \cdot A_o \tag{3.36}$$

The torque and force generated by the actuator 1 and actuator 2 respectively are:

$$\tau = (P_{i1} - P_{o1}) \cdot D_m = \frac{T}{n} \tag{3.37}$$

$$F = P_{i2}A_i - P_{o2}A_o \tag{3.38}$$

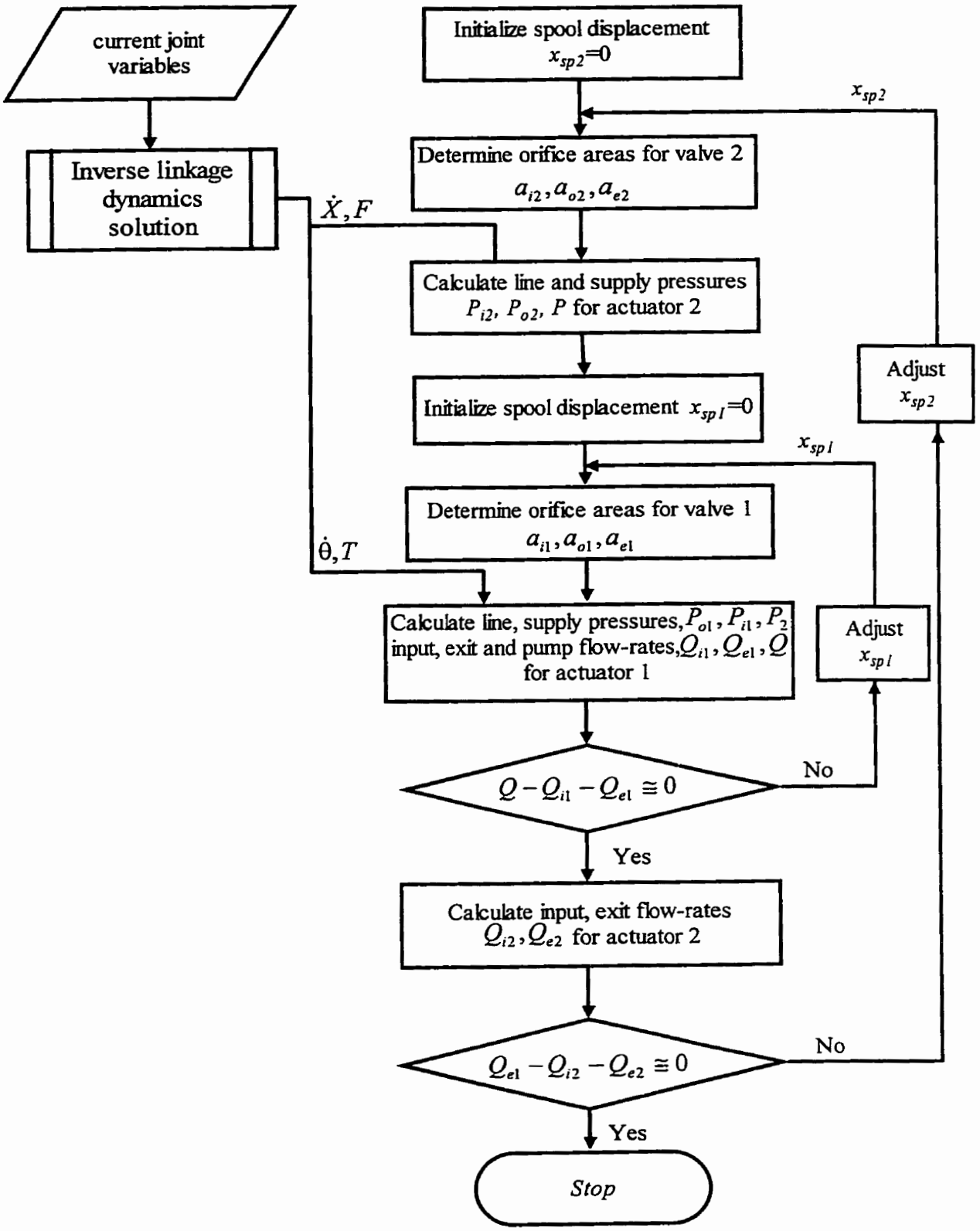


Fig. 3.13: Algorithm for solving swing and stick actuator equations.

The steps taken for solving the hydraulic system equations are described in the flow diagram shown in Fig. 3.13. With reference to this figure the steps are described in the following.

Given the current joint position θ, velocity $\dot{\theta}$, acceleration $\ddot{\theta}$, the actuation torque T is computed using linkage dynamics equation (2.1). The actuation force F is computed using Eq. (2.3).

At the beginning of the iterative procedure, the spool displacement for valve 2 is initialized, $x_{sp2} = 0$.

$$a_{i2}, a_{o2}, a_{e2} = f_s(x_{sp2}) \qquad s = i2, o2, e2 \tag{3.39}$$

Equating (3.35) with (3.36) the output line pressure for actuator 2 is determined as:

$$P_{o2} = \left(\frac{\dot{X}A_o}{ka_{o2}}\right)^2 + P_e \tag{3.40}$$

The input line pressure for actuator 2 is determined from Eq. (3.38) as:

$$P_{i2} = \frac{F + P_{o2}A_o}{A_i} \tag{3.41}$$

Equating Eqs. (3.32) with (3.33) the required pressure to valve 2, P, is computed as:

$$P = \left(\frac{\dot{X}A_i}{ka_{i2}}\right)^2 + P_{i2} \tag{3.42}$$

The spool displacement for valve 1 is initialized, $x_{sp1} = 0$.

$$a_{i1}, a_{o1}, a_{e1} = f_p(x_{sp1}) \qquad p = i1, o1, e1 \tag{3.43}$$

Equating (3.29) with (3.30) the output line pressure for actuator 1 is determined as:

$$P_{o1} = \left(\frac{n\dot{\theta}D_m}{ka_{o1}}\right)^2 + P_e \tag{3.44}$$

The input line pressure for actuator 1 is determined from Eq. (3.37) as:

$$P_{i1} = \frac{\tau}{D_m} + P_{o1} \tag{3.45}$$

Equating Eqs. (3.26) with (3.27) the supply pressure to valve 1, P_2, is computed as:

$$P_2 = \left(\frac{n\dot{\theta} D_m}{k a_{i1}} \right)^2 + P_{i1} \tag{3.46}$$

Then, pump flow Q is determined as a function of the summing pressure $P_1 + P_2$, where P_1 is considered constant and equal to the tank pressure. The input flow Q_{i1} is determined with Eq. (3.26) and the exit flow Q_{e1} is computed using Eq. (3.28). The spool displacement x_{sp1} is incremented until Eq. (3.25) is satisfied. Next, the input flow Q_{i2} is computed with (3.32) and the exit flow Q_{e2} with (3.34). The spool displacement x_{sp2} is incremented until Eq. (3.31) is satisfied.

Therefore, for the spool displacements x_{sp1}, x_{sp2}, the actuation force/torque F, T and actuator velocities $\dot{X}, \dot{\theta}$, the maximum flow Q can be determined for a steady-state condition.

To show the actuator response for two-link actuation with priority action (i.e., swing and stick), the following manipulator motion is simulated: the manipulator is moved from a start position of (swing=0°; stick=-85°) to a goal position of (swing=120°; stick=-25°). Joint trajectories for this motion are generated using truncated trigonometric series and they are shown in Fig. 3.14. Plots of swing actuation torque and stick actuation force are shown in Fig. 3.15.

Figure 3.17 shows that even for providing the flow required to realize a slow motion of 10 deg/s, the stick valve has to be fully opened (i.e. the stick spool is displaced

until it reaches its maximum value). This is due to the fact that the swing actuator draws a greater fraction of the total flow and only the rest of it is delivered to the stick actuator.

Figure 3.19a shows that the required supply pressure is not large. For this manipulator motion the power demand does not exceed the capacity of the engine and therefore, there is no need for the pumps to reduce their flow. Fig. 3.19c shows that the maximum available flow is constant for this motion.

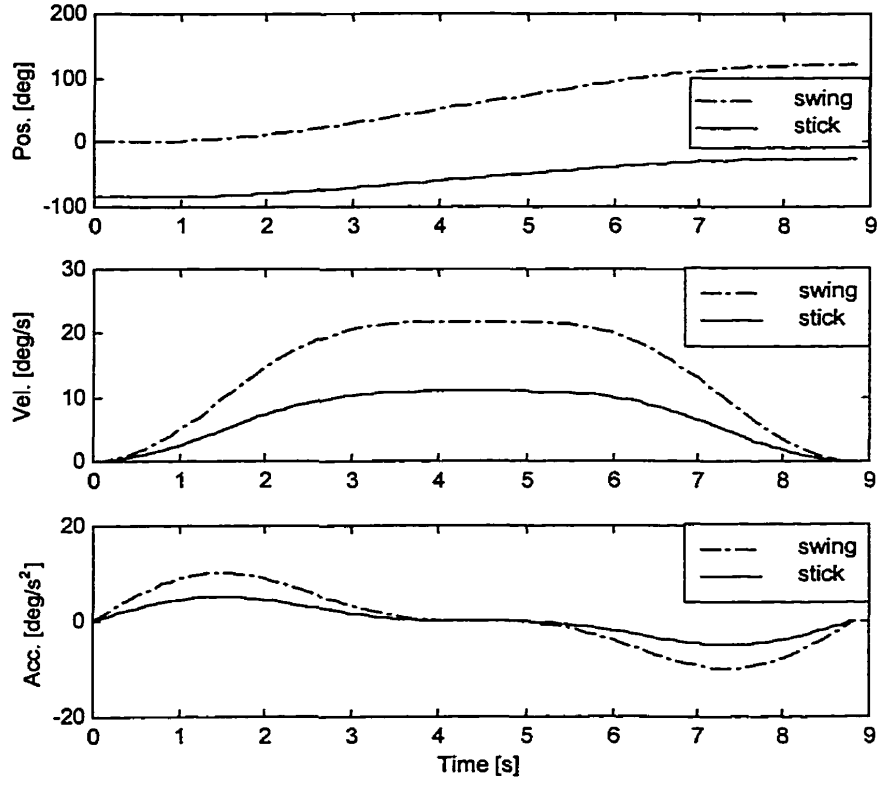


Fig. 3.14: Swing and stick position, velocity and acceleration profiles.

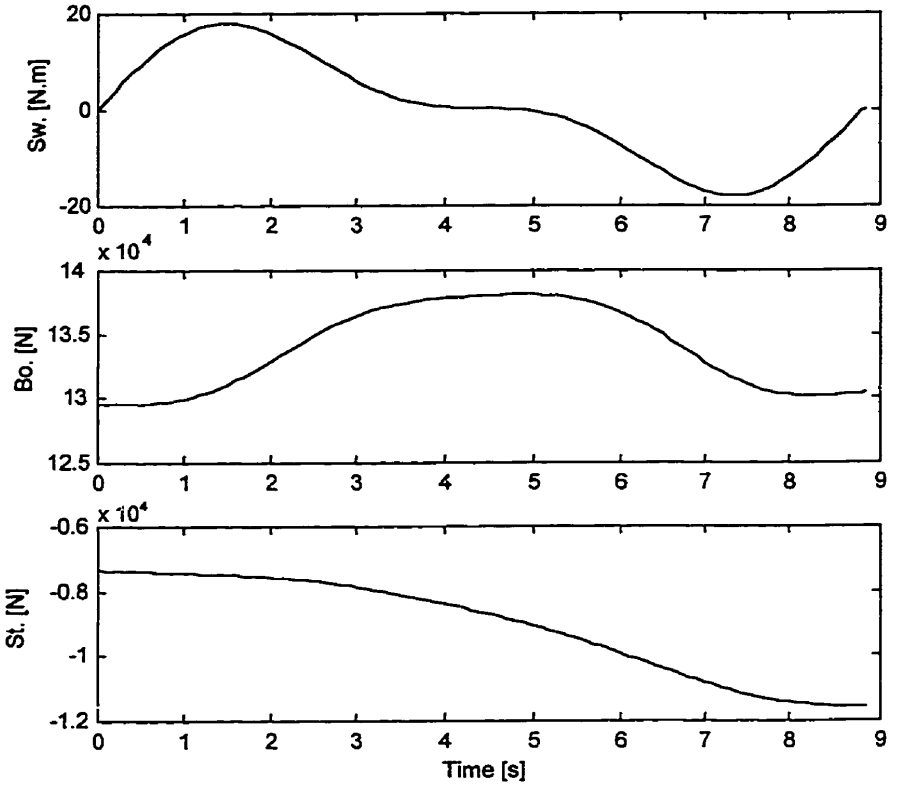


Fig. 3.15: Swing, boom and stick actuator torque/forces.

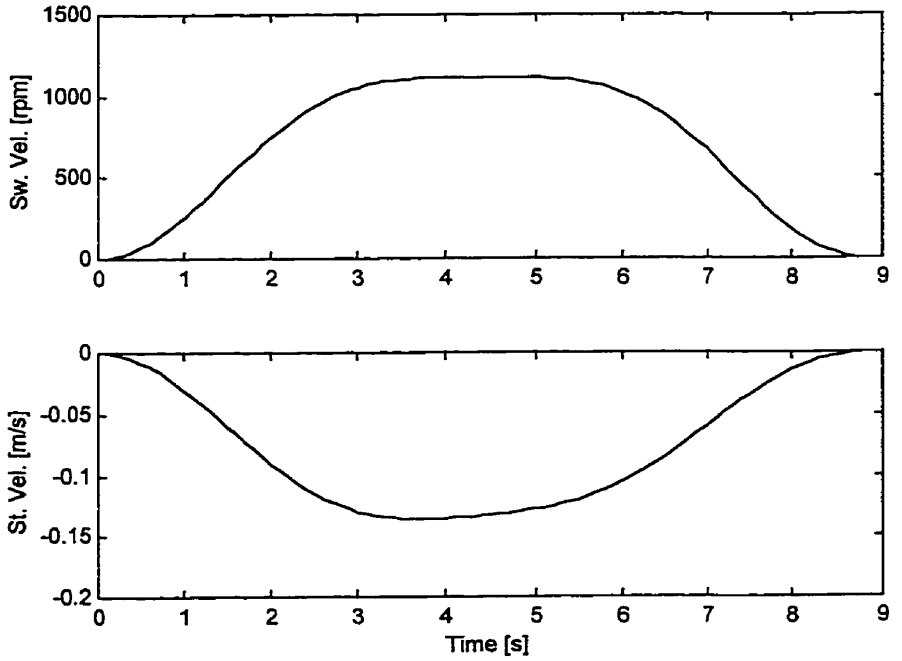


Fig. 3.16: Swing and stick actuator velocities.

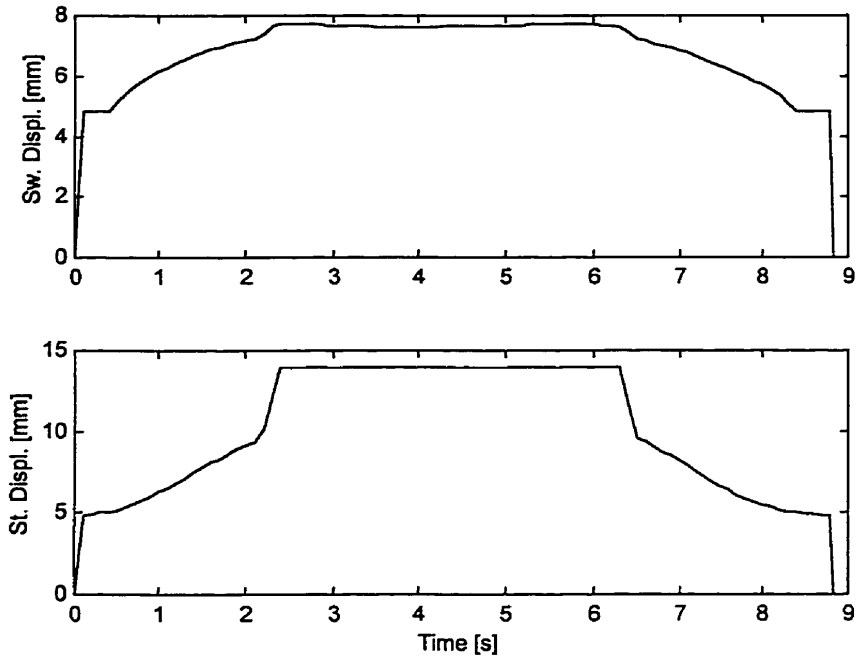


Fig. 3.17: Swing and stick spool displacements.

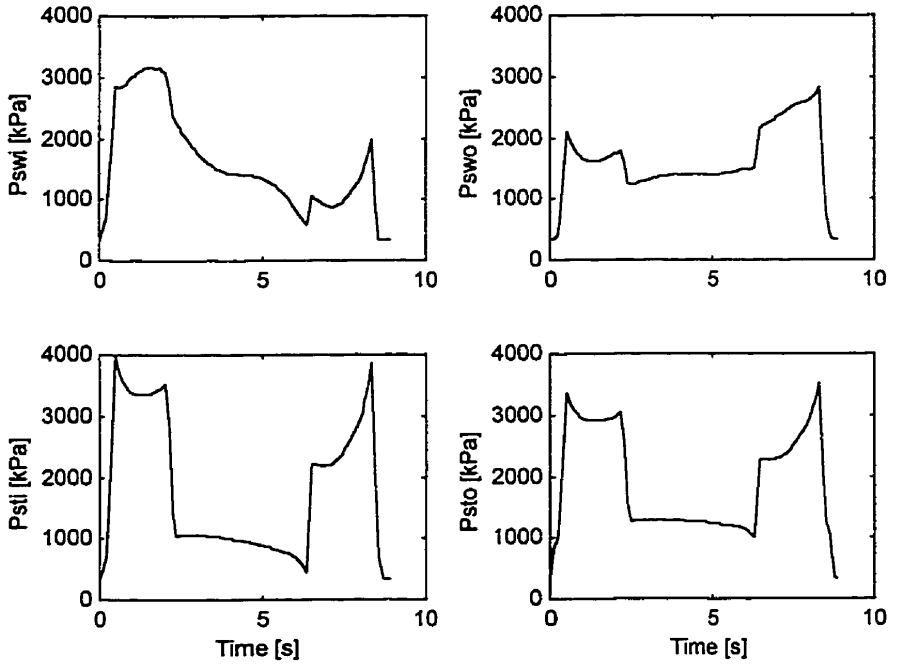


Fig. 3.18: Swing and stick line pressures.

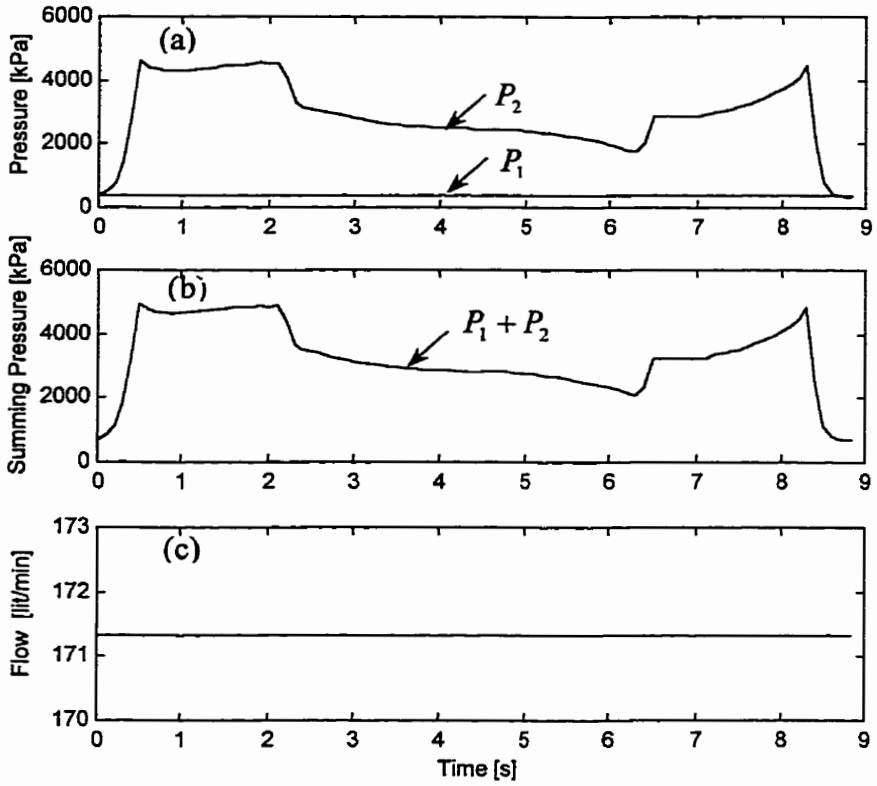


Fig. 3.19: (a) Supply pressures; (b) Summing pressure, $P_1 + P_2$; (c) Maximum available pump flow-rates.

c) Three-links actuation

For three-links actuation the algorithms shown for single link actuation and two-links actuation are combined. However, a slight modification in the two algorithms is imposed. For each step in the algorithm the hydraulic system variables are determined based on the flow rate determined in the previous step. Therefore for a time t_n the supply pressures $P_1(t_n)$ and $P_2(t_n)$ are determined using flow rate $Q(t_{n-1})$. For these supply pressures, the corresponding flow-rate $Q(t_n)$ is determined, considering the pressure-flow limit curve.

The following test shows the actuator response for a manipulator motion, which involves the movement of all links. The manipulator has to move from a start position of (swing=0°; boom=-27°; stick=-85°) to a goal position of (swing=120°; boom=45°; stick=-25°).

Figure 3.20 shows the joint trajectories generated using truncated trigonometric series. The actuator velocity profiles are shown in Fig. 3.21. The spool displacements required to achieve the desired trajectories for swing, boom and stick are shown in Fig. 3.22. Figure 3.23 shows plots of the supply pressures P_1, P_2 and the maximum available pump flow Q.

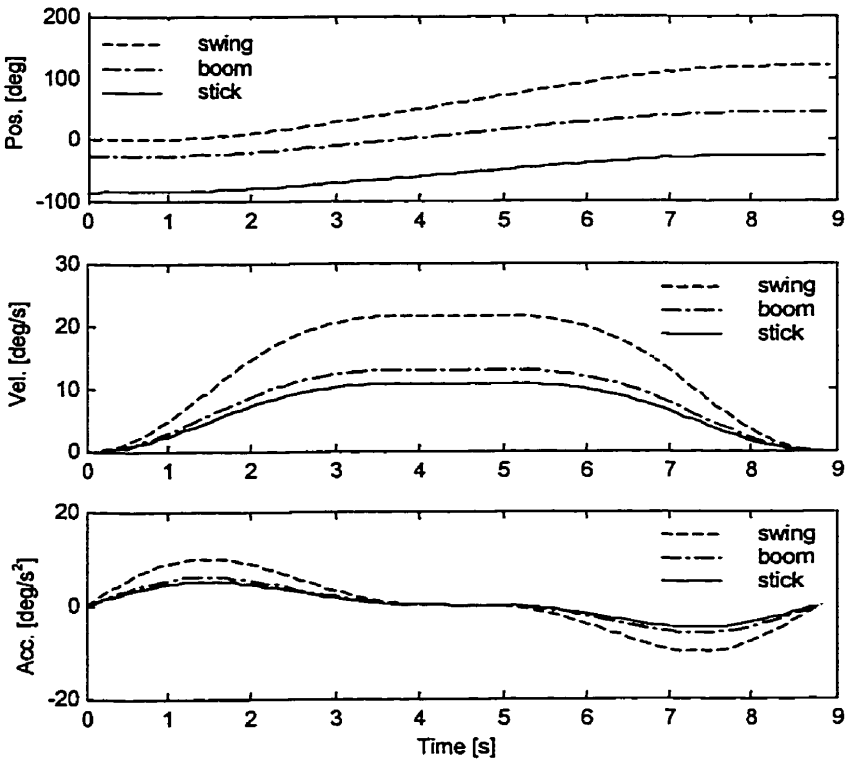


Fig. 3.20: Swing, boom and stick position, velocity and acceleration profiles.

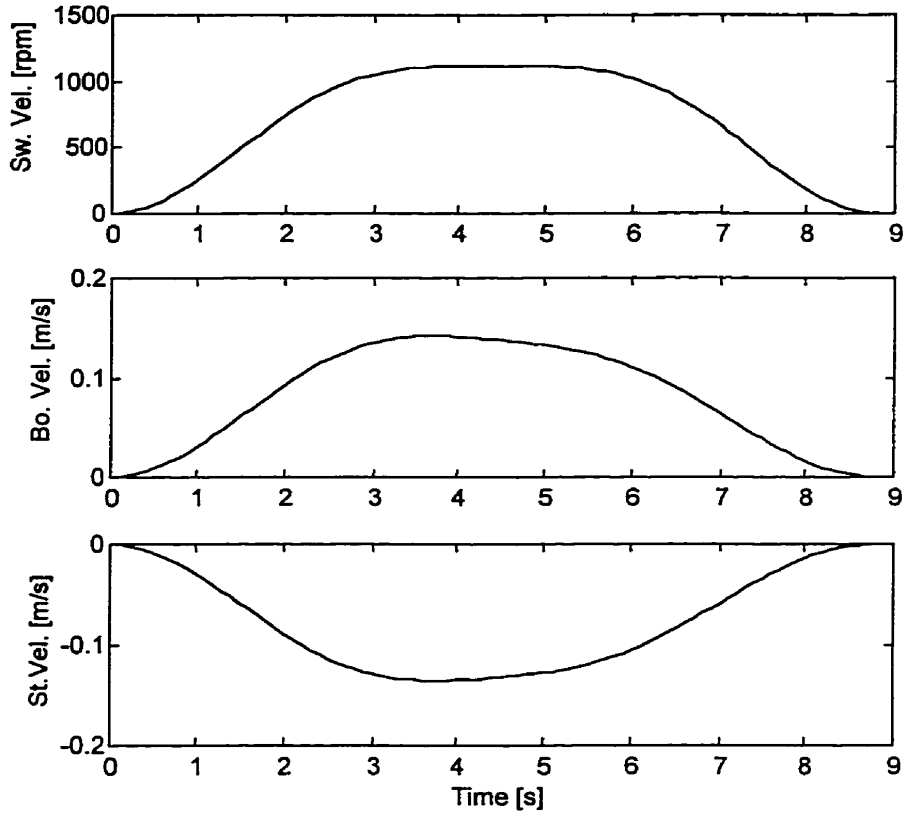


Fig. 3.21: Swing, boom and stick actuator velocities.

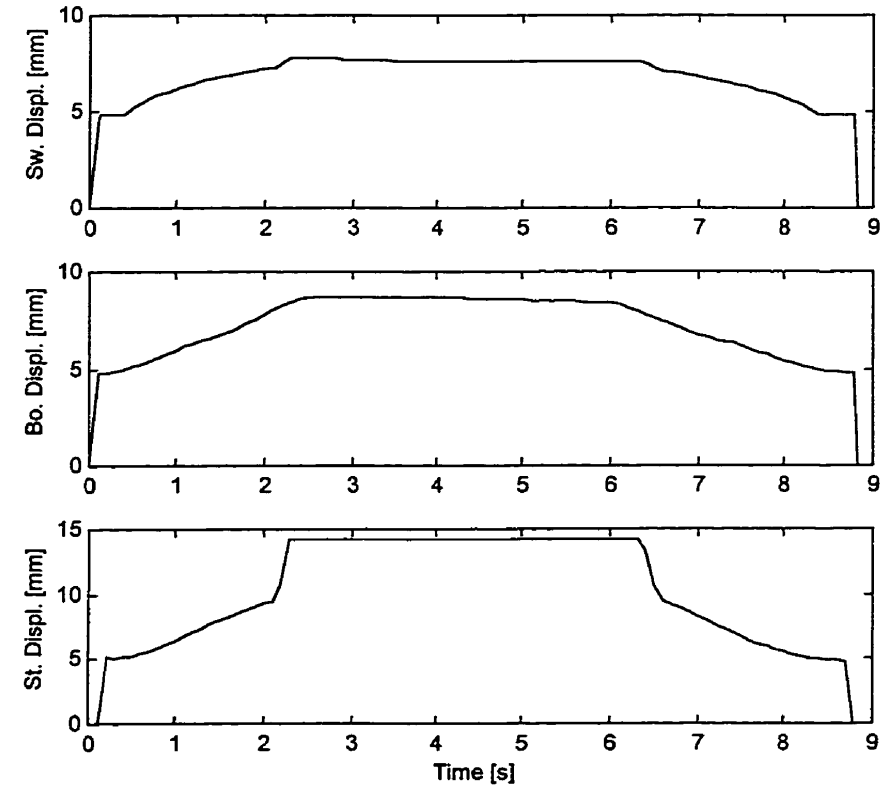


Fig. 3.22: Swing, boom and stick spool displacements.

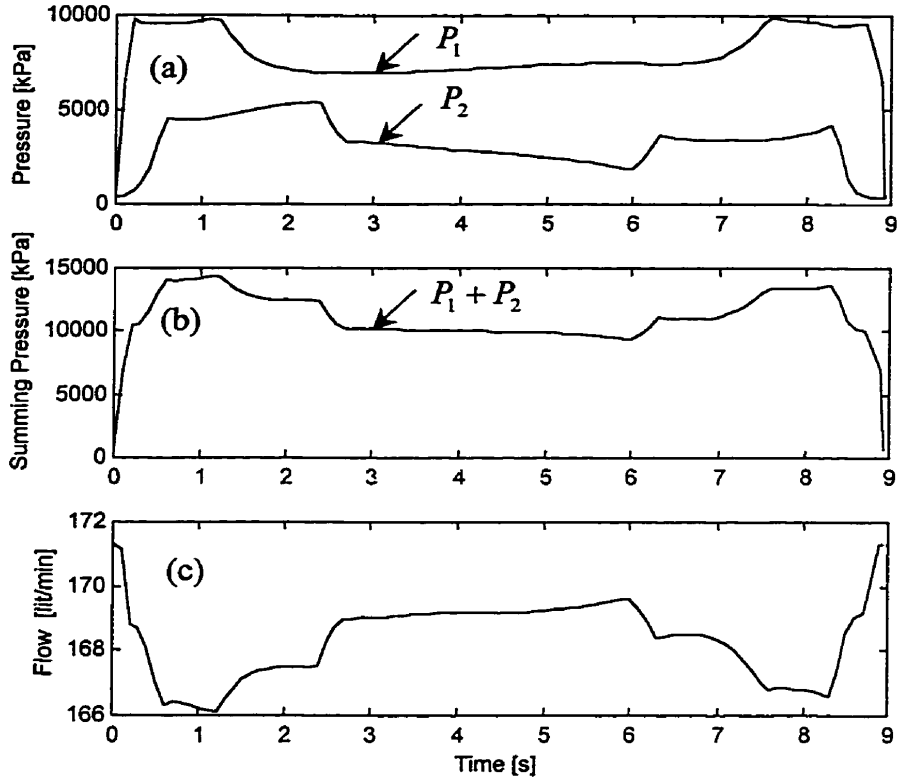


Fig. 3.23: (a) Supply pressures; (b) Summing pressure, $P_1 + P_2$; (c) Maximum available pump flow-rates.

The following tests show the effect of the payload on the output flow from the pumps. The tests are performed for three different cases: (1) the manipulator does not carry a payload; (2) the manipulator carries a payload of 500 kg; (3) the manipulator carries a payload of 1000 kg. It can been seen from Fig. 25a that as the load increases the hydraulic power demand increases and the pumps reduce their output as shown in Fig. 3.25b.

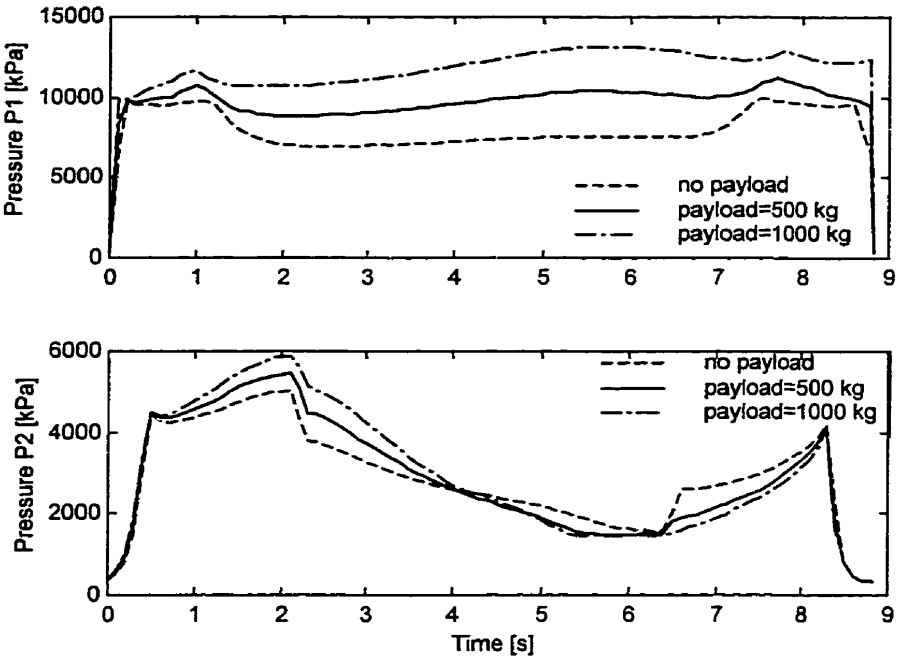


Fig. 3.24: Pump supplies pressures.

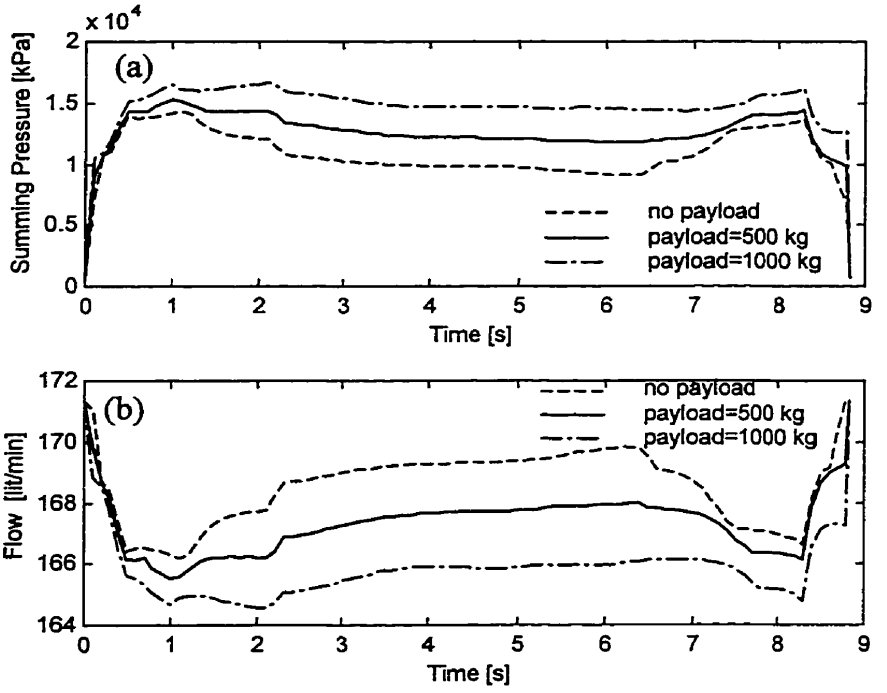


Fig. 3.25: (a) Summing pressure, $P_1 + P_2$; (b) Maximum available pump flow-rates

The maximum flow-rates, which can be delivered from the pumps without exceeding the power limitation, can be computed using the linkage dynamic model and the actuator model for a steady-state situation. The results obtained show that the output flow from the pumps is load dependent. The iterative algorithms developed for solving the actuator model computes also the spool displacements required to achieve the desired trajectories. Once the open-loop spool displacement has been determined, the corresponding open-loop input voltage can be found, i.e. the relationship between the input voltage and the spool displacement is modeled as a first-order differential equation. This open-loop input voltage should drive the manipulator along the desired path if the dynamic model is accurate. However, this is not a realistic assumption because of the approximation made (i.e., the actuator response is approximated by the *steady-state* actuator response), unmodeled disturbances, etc. For this reason it is necessary to apply the open-loop voltage in conjunction with feedback control.

Depending upon the maximum available flow from the pumps, the algorithm presented here determines the flow distribution to each actuator and therefore the scaling factor for joint velocities. A scaling factor k is computed for each time t. The velocity scaling factor c for the entire trajectory is equal to $\min(k(t))$. Therefore, the time scaling factor λ is equal to $1/c$.

3.2.2 Computation of time scaling factor

The steps taken for computing λ are shown in Fig 3.26. For the given Cartesian knot points $E_1, E_2, \ldots, E_n$, the joint displacements $\theta_{j1}, \theta_{j2}, \ldots, \theta_{jn}$ ($j = 1,2,\ldots,N$, where N is the total number of joints) are calculated using inverse kinematics. Given the time intervals,

$h_1, h_2, ..., h_{n-1}$, and the joint displacements, the joint trajectories $\theta_j(t), \dot{\theta}_j(t), \ddot{\theta}_j(t)$ are generated for each joint j using spline functions. For these joint trajectories the maximum flow-rate which can be delivered by the pumps $Q(t)$ ($0 \le t \le T_f$ and $T_f = h_1 + h_2 + ... h_{n-1}$) is determined. Next, the velocity scaling factor, $k(t)$, is determined. Finally, the time scaling factor λ is determined as $\lambda = 1 / \min(k(t))$.

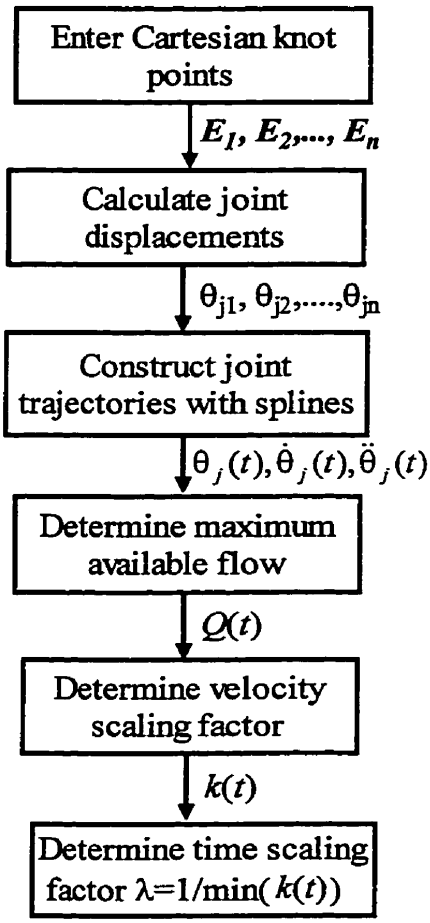


Fig. 3.26: Scheme for determining time scaling factor λ.

3.3 Optimization of joint trajectories

The downhill simplex method is used to find the minimum motion time for a given path. The method is based on an initial design of $p+1$ points, where p is the number of variables to be determined. A $p+1$ geometric figure in a p-dimensional space is called a simplex. The corners of the simplex are called vertices. The downhill simplex algorithm determines which vertex has the least favorite objective function and tries to replace it with a new vertex having a better value of the objective function. The best vertex, found during the search, is used to construct a new simplex for the next search. As a result, the flexible simplex is expected to move closer to the optimal solution, step-by-step. The algorithm determines the search direction by utilizing $p+1$ points in the variable space and following two basic rules: expand in a direction of more favorable conditions, or contract if a move was taken in a direction of less favorable conditions.

Figure 3.27 shows possible outcomes for a step in the downhill simplex method. The simplex at the beginning of the step is shown. The simplex at the end of the step can be any one of: (1) a reflection away from the worst point, (2) a reflection and expansion away from the worst point, (3) a contraction along one dimension from the worst point (a) or from the reflection point (b), or (4) a contraction along all dimensions towards the best point. An illustration of the downhill simplex method operations is given in Appendix C.

Here the objective of the optimization is to adjust the time intervals between adjacent knot points. For a path specified by n knot points, there are $n-1$ time intervals to be adjusted. Downhill simplex method requires three essential elements:

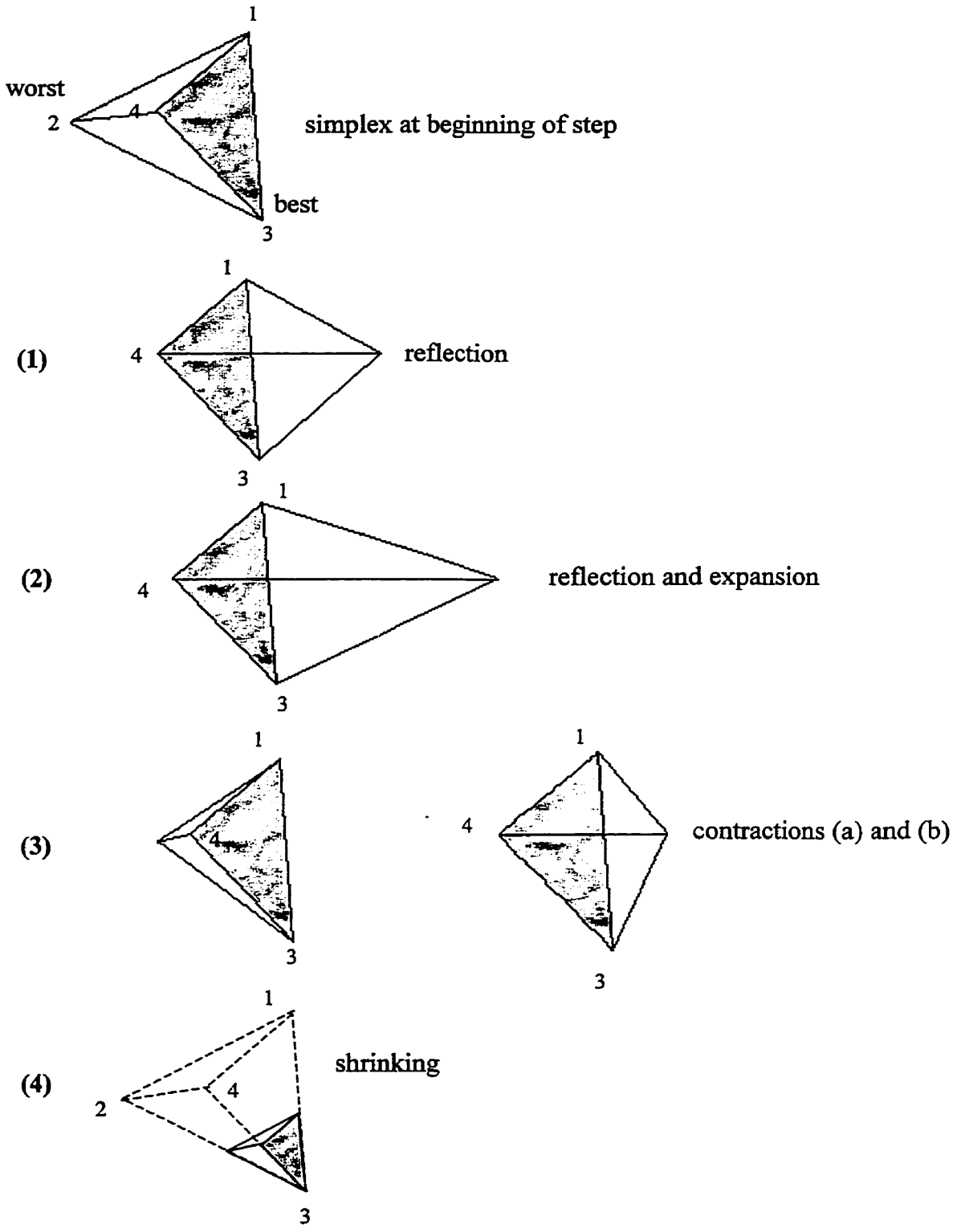


Fig. 3.27: Possible outcomes for a step in the downhill simplex method.

1. Search space: the search space for the optimization is the space of the time intervals $[h_1, h_2, ..., h_{n-1}]$ between adjacent knot points.
2. Objective function: it is represented by the sum of the time intervals $\sum_{i=1}^{n-1} h_i$. The ultimate goal is to minimize this objective function.
3. Stopping criteria: an optimization process terminates when the optimization objective is reached or when the responses cannot be improved further. For the downhill simplex method presented here, the stopping criteria are: (1) the vector distance moved in the terminating step must be smaller than a preselected tolerance ε_1; (2) the decrease in the function value in the terminating step must be smaller than a preselected tolerance ε_2.

Either of the above criteria might be deceived by a single anomalous step that, for one reason or another, failed to get anywhere. Therefore, it is frequently a good idea to *restart* the optimization algorithm at a point where it claims to have found a minimum (Press *et al.*, 1992).

The method must start with n points, defining an initial simplex. A point (vertex) V is defined as the vector of time intervals between knots, $[h_1, h_2, ..., h_{n-1}]$. The objective function for V is represented by $F(V) = h_1 + h_2 + ... h_{n-1}$. The n vertices V_i, $(i = 1,2...n)$ are selected to form the initial simplex as:

$$V_i = V_0 + \mu \cdot e_i \tag{3.47}$$

where e_i's are *(n-1)* unit vectors, and the scalar μ is a constant which might be chosen so as to equalize, as far as possible, the quantities $|F(V_0 + \mu \cdot e_i) - F(V_0)|$. (Walsh, 1975).

Thus, to select these n vertices, one should select the first vertex V_0 and decide for the value of the constant μ. V_0 is selected as the lower bound of time intervals. Let $\theta_{j1}, \theta_{j2} ..., \theta_{jn}$ denote the displacement sequence of joint j. The lower bound of the vector of time intervals V_0 is estimated as:

$$V_0 = (\max_j \frac{|\theta_{j2} - \theta_{j1}|}{v_{max,j}}, \max_j \frac{|\theta_{j3} - \theta_{j2}|}{v_{max,j}}, ... \max_j \frac{|\theta_{j,n} - \theta_{j,n-1}|}{v_{max,j}}) \quad (3.48)$$

Where v_{max} is the highest joint velocity determined by the manipulator's physical limitations:

$$v_{max} = \frac{\dot{X}_{max}}{\min(J(\theta))} \quad (3.49)$$

If the maximum flow-rate from the pumps Q_{max} is used by each actuator alone then:

$$\dot{X}_{max} = \frac{Q_{max}}{A_i} \quad (3.50)$$

For the restart of the optimization algorithm at a point where it claims to have found a minimum, n-1 of n vertices of the simplex are reinitialized by Eq. (3.47) with V_0 being the vertex of the claimed minimum.

A flow diagram for implementing the downhill simplex method is shown in Fig. 3.28. With reference to this figure the notations used and the steps taken are explained in the following. The procedure for defining first simplex has been already explained. For this simplex the vertices are ranked in order: B, N_w, W, where B is the vertex with the lowest function value, N_w is the vertex with the second highest function value and W is the vertex with the highest function value. Operations for searching a better vertex and

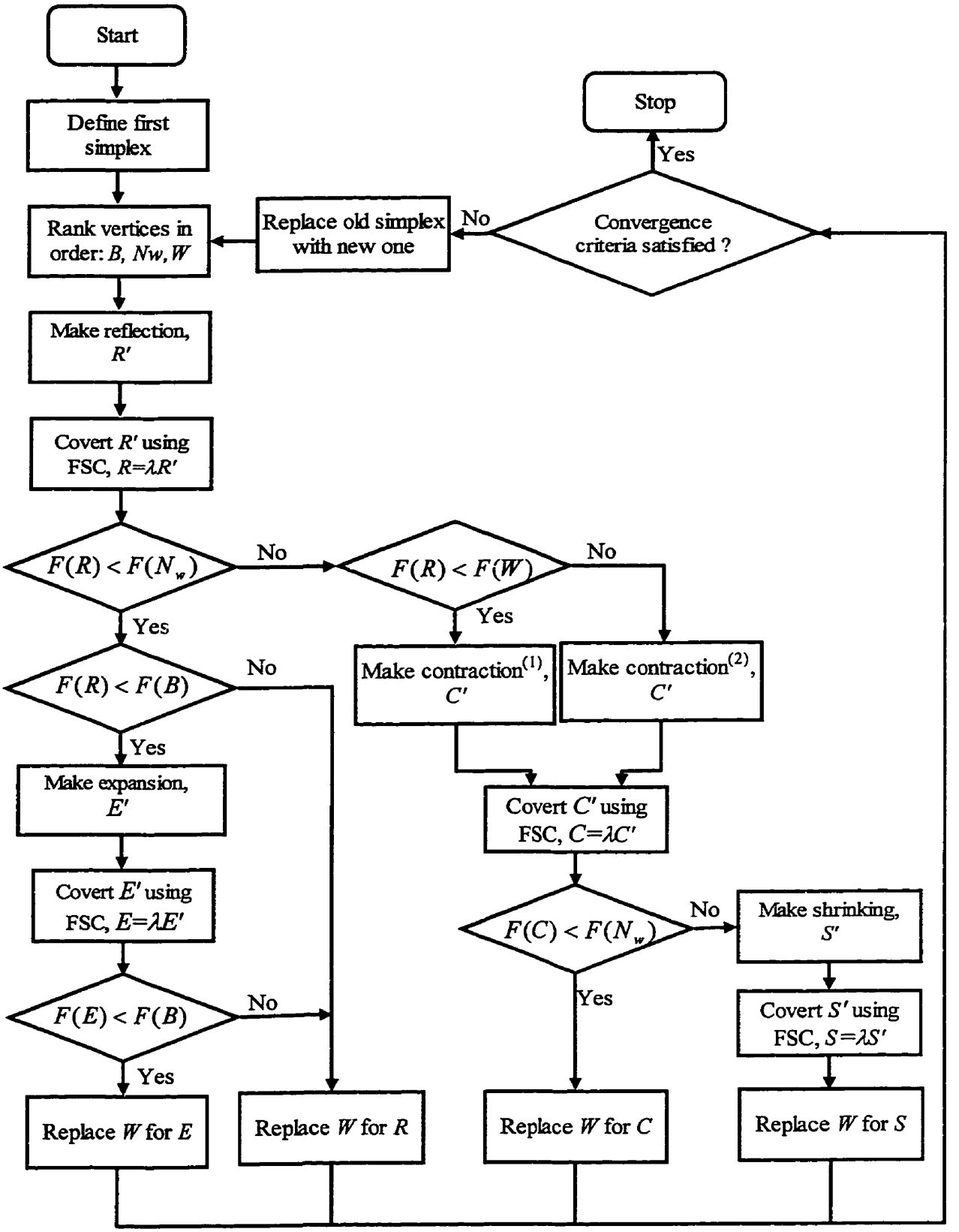


Fig. 3.28: Optimization algorithm.

for reducing the size of the simplex include reflection, expansion, contraction and shrinking. They are defined as follows:

(1) Reflection:

$$\boldsymbol{R}' = \overline{\boldsymbol{C}} + \alpha(\overline{\boldsymbol{C}} - \boldsymbol{W}) \tag{3.51}$$

where α is the reflection factor, $\overline{\boldsymbol{C}}$ is the centroid of all vertices except $\boldsymbol{W}$, i.e.,

$$\overline{\boldsymbol{C}} = \frac{1}{n-1}\left[\sum_{i=1}^{n} \boldsymbol{V}_i - \boldsymbol{W}\right] \tag{3.52}$$

Each vertex obtained by the search is a vector of time intervals. With these time intervals and the joint displacements, the joint trajectories can be determined. However, these trajectories might not be dynamically realizable given the actuator constraints. In this case, these time intervals should be adjusted to bring the trajectories within the actuator constraints. Therefore, in the general downhill simplex algorithm should be added a procedure which converts an infeasible vertex to a feasible one. The procedure is called Feasible Solution Converter and it is as follows:

1) Determine time scaling factor λ.

2) Replace time intervals $(h_1, h_2 \ldots h_{n-1})$ by $(\lambda h_1, \lambda h_2 \ldots \lambda h_{n-1})$.

Hence, using the Feasible Solution Converter procedure, the infeasible vertex $\boldsymbol{R}'$ is converted to the feasible vertex $\boldsymbol{R} = \lambda \boldsymbol{R}'$.

Another aspect particular to the problem under investigation is the nature of the elements of the vertex, i.e., all the elements are time intervals, therefore, it is necessary to assure that all of them are positive. For this, the reflection factor α should be properly determined. α is determined as suggested by Lin *et al.* (1983). At first, α is set to 1. If

any element of $\boldsymbol{R}'$ is negative α will be changed to a smaller value. For $\alpha = 1$, $\boldsymbol{R}'$ is obtained as:

$$\boldsymbol{R}' = 2\overline{\boldsymbol{C}} - \boldsymbol{W} = [2h_1^{\overline{C}} - h_1^W, 2h_2^{\overline{C}} - h_2^W, ..., 2h_{n-1}^{\overline{C}} - h_{n-1}^W] \tag{3.53}$$

If $2h_i^{\overline{C}} - h_i^W \leq 0$ for some i, then α should be reduced. From Eq. (3.51) one obtains

$$\boldsymbol{R}' = [h_1^{\overline{C}} + \alpha(h_1^{\overline{C}} - h_i^W), h_2^{\overline{C}} + \alpha(h_2^{\overline{C}} - h_2^W), ..., h_{n-1}^{\overline{C}} + \alpha(h_{n-1}^{\overline{C}} - h_{n-1}^W)] \tag{3.54}$$

Therefore,

$h_i^{R'} = h_i^{\overline{C}} + \alpha(h_i^{\overline{C}} - h_i^W)$ for some i.

Consequently, α should be less than $\dfrac{h_i^{\overline{C}}}{(h_i^W - h_i^{\overline{C}})}$ to make $h_i^{R'}$ positive. Based on the above discussions, α is determined as:

$$\alpha = \begin{cases} 1 & \text{if } 2h_i^{\overline{C}} - h_i^W > 0 \quad \text{for all } i \\ \delta_1 = \min\left\{\dfrac{h_i^{\overline{C}}}{(h_i^W - h_i^{\overline{C}})}\right\} & \text{if } 2h_i^{\overline{C}} - h_i^W > 0 \quad \text{for some } i \end{cases} \tag{3.55}$$

where $0 < \delta_1 < 1$.

(2) Expansion:

$$\boldsymbol{E}' = \overline{\boldsymbol{C}} + \gamma(\boldsymbol{R} - \overline{\boldsymbol{C}}) \tag{3.56}$$

To keep all elements of $\boldsymbol{E}'$ as positive, the expansion coefficient γ can be determined as:

$$\gamma = \begin{cases} 2 & \text{if } 2h_i^{R'} - h_i^{\overline{C}} > 0 \quad \text{for all } i \\ \delta_2 = \min\left\{\dfrac{h_i^{\overline{C}}}{(h_i^{\overline{C}} - h_i^{R'})}\right\} & \text{if } 2h_i^{R'} - h_i^{\overline{C}} > 0 \quad \text{for some } i \end{cases} \tag{3.57}$$

where $0 < \delta_2 < 1$

(3) Contraction:

Here there are two cases to consider. Contraction[a] find $\boldsymbol{C}'$ as

$$\boldsymbol{C}' = \overline{\boldsymbol{C}} + \beta(\boldsymbol{R} - \overline{\boldsymbol{C}}) \tag{3.58}$$

Contraction[b] find $\boldsymbol{C}'$ as

$$\boldsymbol{C}' = \overline{\boldsymbol{C}} + \beta(\boldsymbol{W} - \overline{\boldsymbol{C}}) \tag{3.59}$$

(4) Shrinking:

This operation reduces the size of the simplex by halving the distances from $\boldsymbol{B}$.

$$\boldsymbol{V}_i = \boldsymbol{B} + 0.5(\boldsymbol{V}_i - \boldsymbol{B}) \qquad i = 1,2,...,n \tag{3.60}$$

Chapter 4

Demonstrative Results

In this chapter the optimization algorithm described in Chapter 3 is employed to determine the minimum time trajectories for a specified path. The path is specified by a sequence of knot points. The control variables for the optimization are the time intervals between adjacent knot points and the objective function is the total motion time, therefore the sum of these time intervals. The stopping criteria for the optimization algorithm are: (1) the vector distance moved in the terminating step must be smaller than a tolerance ε_1, and (2) the decrease in the function value in the terminating step must be smaller than a tolerance ε_2. The tolerances $\varepsilon_1, \varepsilon_2$, were chosen, for all the tests performed, equal to 10^{-4}.

4.1. Example 1

In this example a typical pick and place task was considered. The task required the end-effector to move along the three-dimensional path shown in Fig. 4.1. As it can be seen from the figure four knot points initially specified the path. A set of tests was performed to investigate the effect of increasing the number of knot points along the path, in order to pick the "best" number of knot points, i.e., the number of knot points for which the optimization algorithm finds the best solution. The optimization was performed for four, six, eight and ten knot points along the path, therefore for three, five, seven and nine control variables.

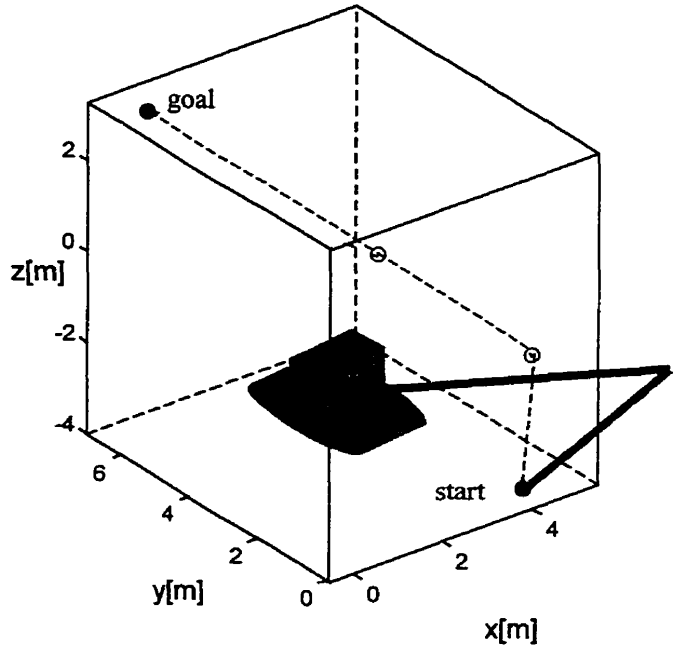


Fig. 4.1: Initial Cartesian path.

From the results shown in Table 4.1 it can be seen that the minimum function value (minimum motion time), found by the optimization algorithm, decreased with the number of the control variable (time intervals). However, for more than seven control variables the optimization algorithm did not converge. Actually, it is known that downhill simplex method works particularly well if the number of variables is not large (i.e. five or six, with reference to Walsh, 1975). This effect might be due to the fact that adding dimensions to the simplex causes more local minima and searching among the local minima becomes more complicated as the dimensionality increases.

On the other hand, increasing the number of control variables increases the computation time. The optimization algorithm is computationally expensive due to the feasible solution converter procedure, particularly the calculation of the pump flow

trajectory curve, which requires the iterative solutions for the hydraulic manipulator model. The minimum objective function values in Table 4.1 were obtained for two runs. The first run consists of starting the optimization algorithm from a starting point and letting it run until it reaches the final solution. The downhill simplex method searches for a better solution until it reaches a point where the solution can not be further improved, in the other words when the stopping criteria are satisfied. The second run consists of restarting the optimization algorithm at the solution obtained for the first run and letting it run until it reaches the final solution.

Table 4.1: Effect on increasing the number of knot points on optimal solution.

No. of control variables	1st run		2nd run	
	starting point	final point	starting point	final point
3	12.98 s	12.50 s	12.50 s	11.30 s
5	13.04 s	10.44 s	10.44 s	10.15 s
7	13.71 s	9.74 s	9.74 s	8.98 s
9	13.70 s	13.70 s	-	-

The above results indicate that for seven control variables, which means eight knot points along the path, the minimum motion time, found by the optimization algorithm, has the smallest value for all three cases. Therefore, the best number of knot points along the path is eight knot points.

Optimal trajectory planning for a path specified by eight knot points

In the following are shown the results of the optimization performed for a path specified by eight knot points. These results are obtained by running the optimization algorithm for two cases:

- Case 1- the end-effector moves along a specified path with no payload.
- Case 2-the end-effector moves along a specified path carrying a 500 kg payload

Manipulator with no payload

Eight knot points specified the Cartesian path of the end-effector. By means of inverse Jacobian, joint displacements were computed for these knots as shown in Table 4.2.

Table 4.2: Joint displacements corresponding to Cartesian knot points.

	Swing [deg]	Boom [deg]	Stick [deg]
1	0	-25	-111.5
2	5.7	-17	-106
3	10	-10	-96.3
4	20	-3	-81
5	50	11.5	-78.5
6	70	23.5	-60.2
7	80	34.5	-51.6
8	90	40	-42

The optimal solution was obtained by running the optimization algorithm twice. The results are explained in detail in the following.

First run

To start the search for the minimum motion time, the initial simplex was defined using Eq. (3.47). The value for the constant μ was selected as 0.05.

The first vertex was determined computing the lower bound of the vector of the time intervals with Eq. (3.48)

$$V_0' = [0.43 \quad 0.42 \quad 0.67 \quad 0.75 \quad 0.81 \quad 0.6 \quad 0.43]$$

The feasible solution converter procedure was applied to convert each infeasible vertex of the initial simplex, V_i', to a feasible one, so that the actuator constraints were satisfied. The vertices of the initial simplex are shown in Table 4.3.

Table 4.3: Initial simplex.

	V_0	V_1	V_2	V_3	V_4	V_5	V_6	V_7
1	1.44	1.51	1.44	1.44	1.44	1.44	1.44	1.44
2	1.40	1.40	1.47	1.40	1.40	1.40	1.40	1.40
3	2.23	2.23	2.23	2.34	2.23	2.23	2.23	2.23
4	2.51	2.51	2.51	2.51	2.63	2.51	2.51	2.51
5	2.70	2.70	2.70	2.70	2.70	2.83	2.70	2.70
6	1.99	1.99	1.99	1.99	1.99	1.99	2.09	1.99
7	1.43	1.43	1.43	1.43	1.43	1.43	1.43	1.51

For the first run, the algorithm performed 1156 function evaluations of $\sum_{i=1}^{n-1} h_i$ to reach the final solution. The motion time at the starting point of the optimization algorithm was 13.71 s and the final value for the motion time was 9.74 s. The optimal solution found for the first run was:

$$V = [1.14 \quad 0.71 \quad 2.05 \quad 1.54 \quad 1.82 \quad 0.89 \quad 1.58].$$

Comparing vectors of time intervals shown in Table 4.3 to the optimal solution V, it is seen that all time intervals were adjusted simultaneously to achieve a shorter motion time.

Second run

For the second run the first vertex V_0 of the initial simplex was reinitialized with the value of the solution found for the first run V.

$$V_0 = [1.14 \quad 0.71 \quad 2.05 \quad 1.54 \quad 1.82 \quad 0.89 \quad 1.58].$$

The final value for the motion time obtained for the second run was 8.98 s and the vector V of time intervals was:

$$V = [1.13 \quad 0.73 \quad 1.52 \quad 1.28 \quad 1.85 \quad 0.82 \quad 1.64].$$

Fig. 4.2 shows the algorithm convergence to the optimal solution. For each step in the optimization algorithm a tentative solution was obtained. The figure shows the minimum objective function value for each solution found during the search.

Fig. 4.3a shows how the algorithm, through successive expansions and contractions of the simplex, made its way until it encountered a minimum (at least a local one). Fig. 4.3b indicates the decrease in the function value in each step. When the vector distance moved in one step and also the decrease in the function value become smaller than the specified tolerances, it means that the optimization can not find a better solution than the one already at the hand, and the optimization terminates. Very small tolerances were chosen for the stopping criteria, i.e., $\varepsilon_1, \varepsilon_2$ were equal to 10^{-4}. Fig. 4.5 shows that for almost 200 steps, as the simplex became very small (see Fig. 4.4), the solution could not be improved significantly. To improve computation time, an alternative to using very small tolerances might be to use larger tolerances for the first run and smaller ones for the second one.

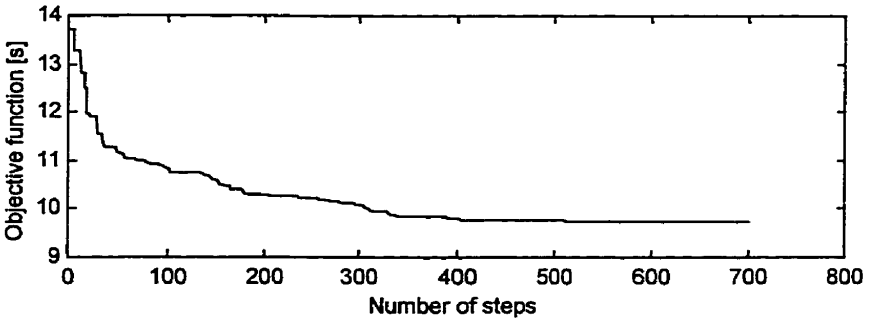


Fig. 4.2: Convergence behavior of the optimization algorithm; 1st run.

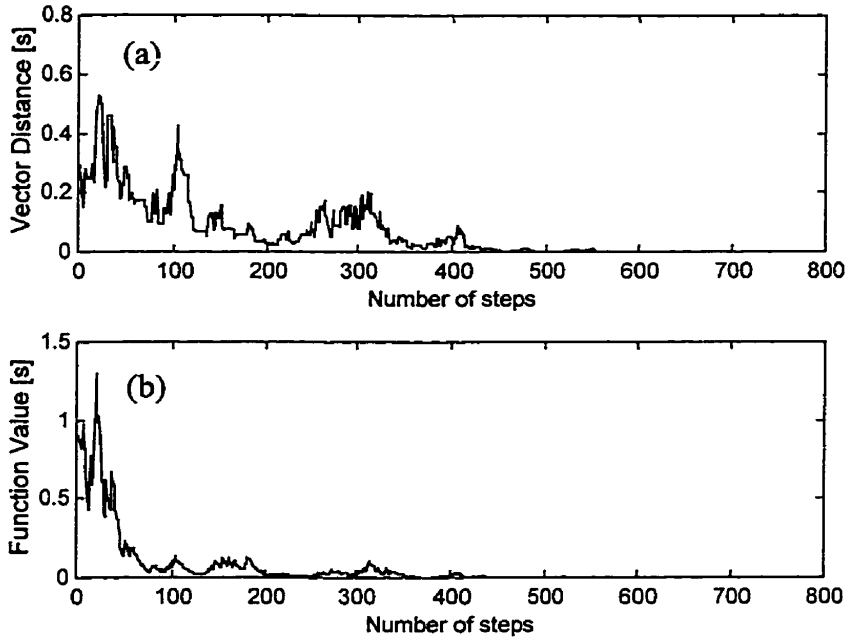


Fig. 4.3: Stopping criteria: (a) Vector distance moved in each step; (b) Decrease in function value in each step; 1st run.

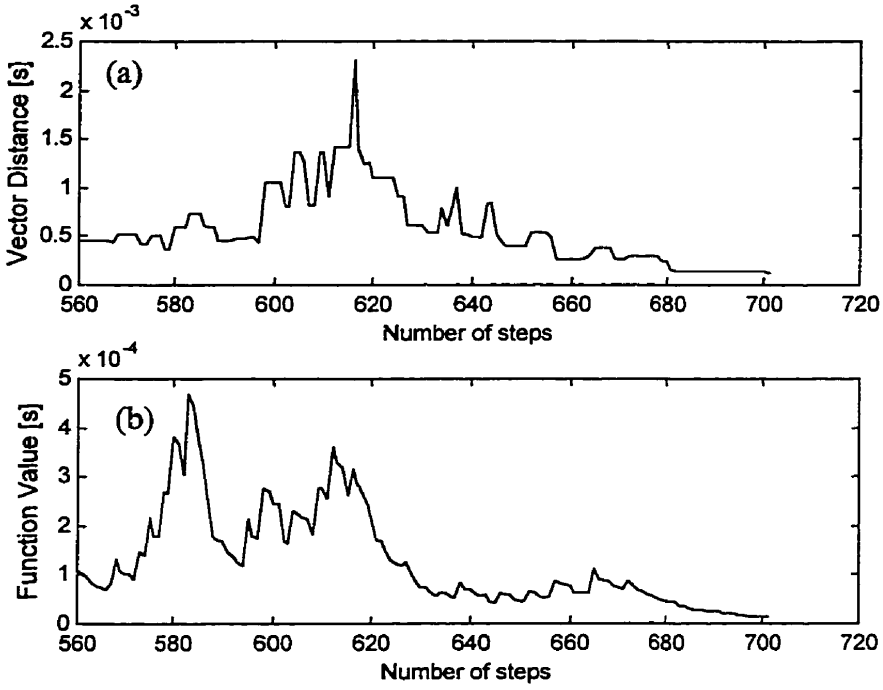


Fig. 4.4: Blow-up of Fig. 4.3; 1st run.

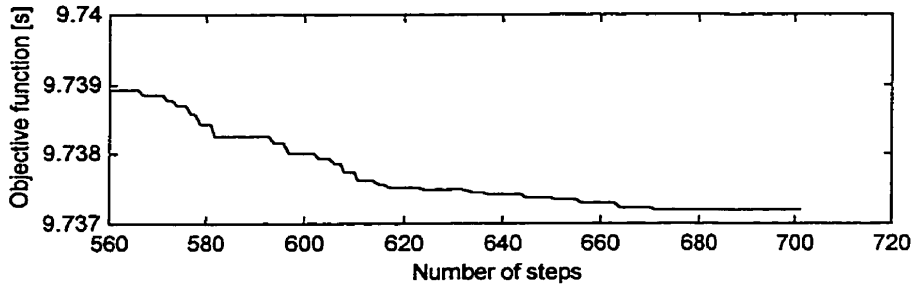


Fig. 4.5: Blow-up of Fig. 4.2; 1st run.

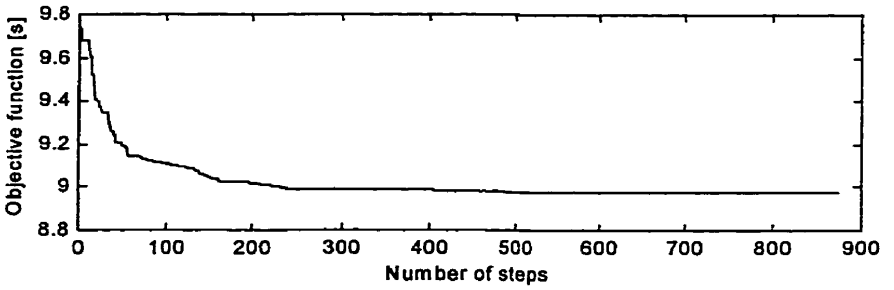


Fig. 4.6: Convergence behavior of the optimization algorithm; 2nd run.

Fig. 4.7 shows the Cartesian path corresponding to the optimal solution. It can be seen that the path is smooth everywhere, including at knot points. Figures 4.8 to 4.10 show a comparison of the trajectories corresponding to the optimal solution to the trajectories obtained at the starting point of the optimization. In Figs. 4.8b, 4.9b and 4.10b are showed the values that the manipulator must follow to achieve a minimum motion time. It can be seen from Fig. 4.10 that a high acceleration brings up the velocity in a shorter time interval so that the motion time for the manipulator is shortened. Figure 4.11 shows the spool displacements required to achieving the optimal trajectories shown in Fig. 4.8b to 4.10b.

Figures 4.12 to 4.14 show the optimal trajectories generated with four knot points and eight knot points respectively. For four knot points small joint accelerations for the end segments of the trajectories resulted in small velocities and therefore the minimum motion time found by the optimization algorithm was larger for four knot trajectories.

Manipulator with payload

A similar test to the previous one was performed for the manipulator carrying a payload. The end-effector moved along the specified path carrying a 500 kg payload. The resulting minimum motion time for this test was 9.11 s, comparable to 14.53 s the motion time for the starting point of the optimization algorithm.

Fig. 4.15 shows the velocity profiles in the two cases, Case (1)-the manipulator does not carry a payload, Case (2)-the manipulator carries a 500 kg payload. For Case (2) it can be seen a decrease in joint velocities. In Chapter 3 it had been shown the effect of the payload on the output flow from the pumps (see Fig. 3.25). Increasing the payload results in a decrease of the output flow from the pumps. This is due to the fact that increasing the load, the power demand increases and might exceed the capacity of the engine. To prevent such a situation, the pumps reduce their flow to the actuators. As less flow is delivered to the actuators, smaller actuator velocities can be achieved and therefore smaller joint velocities.

The optimization algorithm checks each solution obtained by the search for feasibility. In the following an example of conversion of an infeasible solution, for which the actuator constraints are not satisfied, into a feasible one is presented.

For the desired joint velocity trajectories shown in Fig. 4.17, the desired flow-rate to each hydraulic actuator was determined. Fig. 4.18 shows plots of the desired flow-rates to swing, boom and stick. From Fig. 4.19, it can be seen that the desired flow-rates to both hydraulic circuits exceed the available flow from the pumps. Therefore, the desired flow-rates should be modified. The modification was performed using the algorithm

described in Section 3.2.1. The desired flow-rates were checked against the maximum availability and interconnection constraints and two scaling factors k_1, k_2, respectively, were obtained. With k_1 and k_2 a factor k is determined as $k = k_1 k_2$. Figure 4.20 shows plots of the scaling factors $k_1(t)$, $k_2(t)$ and $k(t)$. The desired flow-rates are scaled by a constant scaling factor $c = \min(k(t))$ (see Fig. 4.21). The time scaling factor λ is determined as $1/c$. The scaled joint velocities are shown in Fig. 4.22.

A new test was performed to investigate what would happen if the end-effector was to follow the optimal trajectory, computed with no payload, carrying a 5000 kg payload. With reference to Fig. 4.23, a large increase in the summing pressure and a large decrease in the pump flows is noticed as a result of carrying the 5000 kg payload. Thus, the optimal trajectory computed for the no-load case becomes infeasible. This can be seen from Figs. 4.25 and 4.26 where the required flows exceed the available flow from the pumps. The conversion to a feasible solution is shown in Figs. 4.27 to 4.29. Fig 4.29 shows the resulting feasible velocity trajectories. These are achievable trajectories but may not be optimal ones. Therefore, the optimization algorithm should be employed again to compute minimum-time trajectories.

Similar to the previous tests the optimization algorithm was run twice to obtain the optimal solution for the case with 5000 kg payload. The minimum motion time, found by the optimization algorithm, was 14 s and the vector V of time intervals was:

$$V = [2.2154 \quad 1.3104 \quad 1.9775 \quad 2.2245 \quad 2.5875 \quad 1.5115 \quad 2.1815].$$

The resulting optimal position, velocity, acceleration trajectories are shown in Figs. 4.30 to 4.32. They are compared with the position, velocity, and acceleration trajectories at the starting point of the optimization.

Fig. 4.33 shows the optimal velocity profiles for the case with no payload and for the case with 5000 kg payload. A decrease in joint velocities is observed for the case with 5000 kg payload, which is expected.

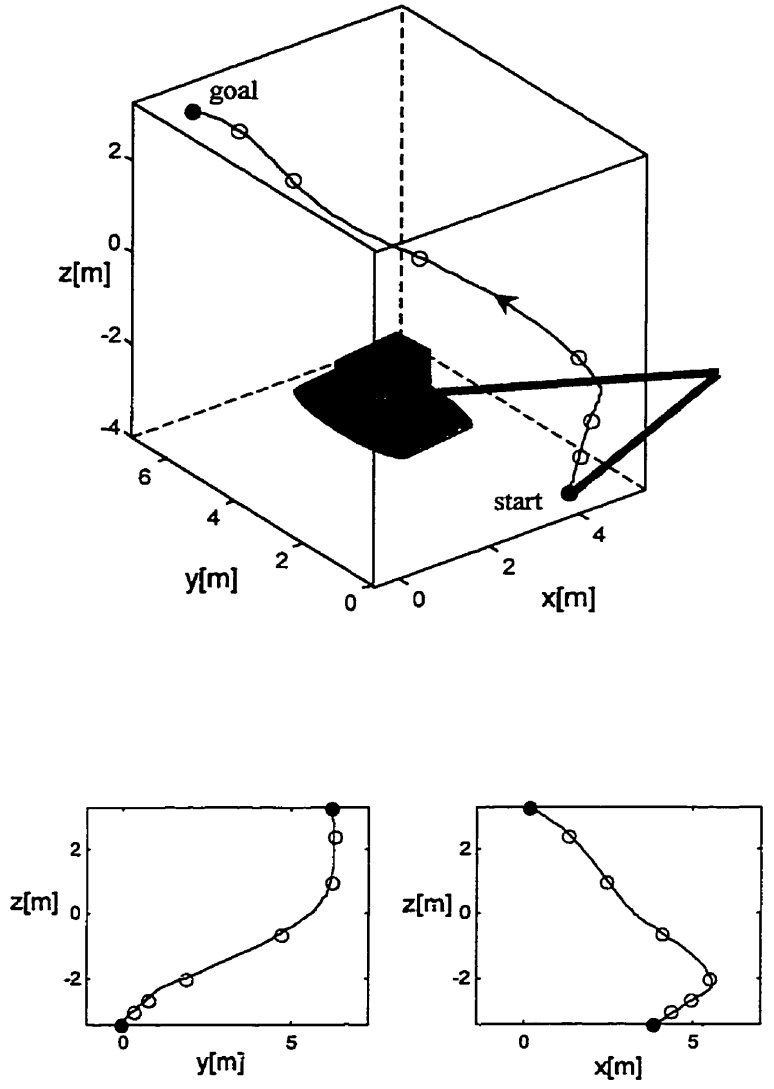


Fig. 4.7: Cartesian path corresponding to the optimal solution.

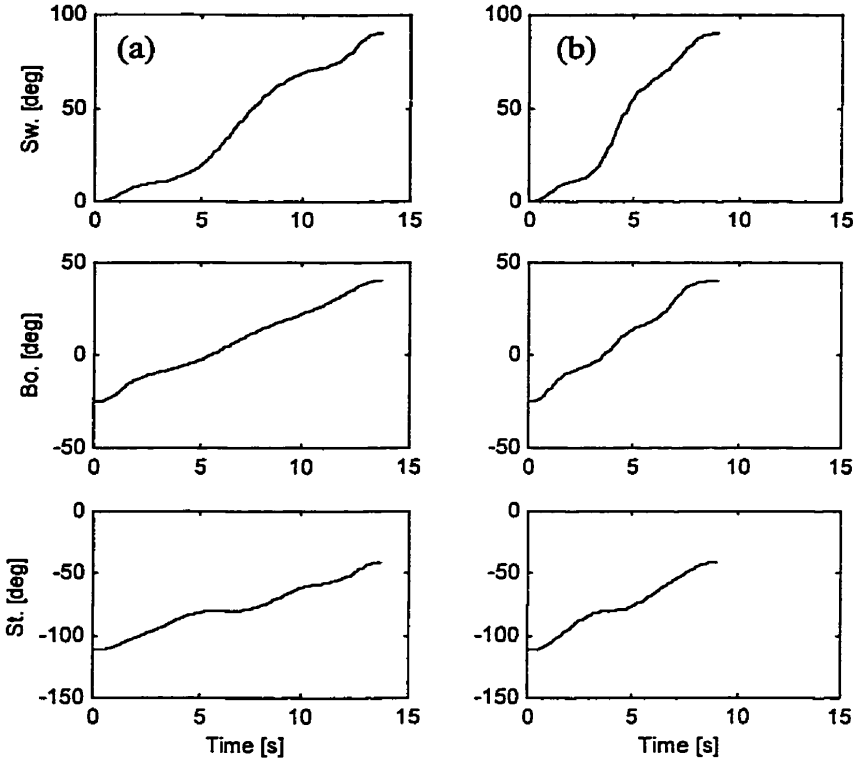


Fig. 4.8: Joint position profiles: (a) initial solution; (b) optimal solution.

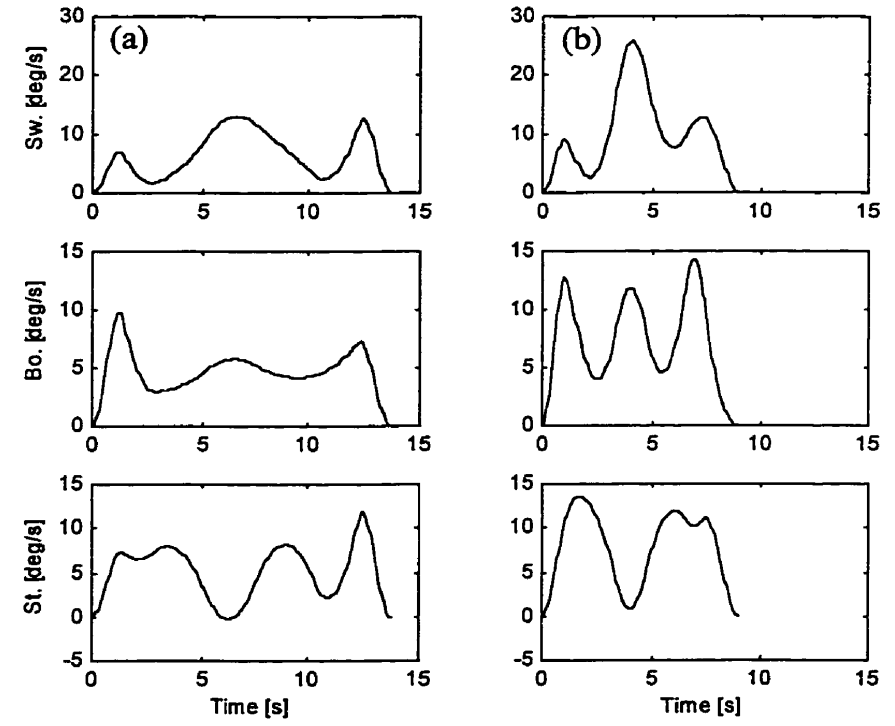


Fig. 4.9: Joint velocity profiles: (a) initial solution; (b) optimal solution.

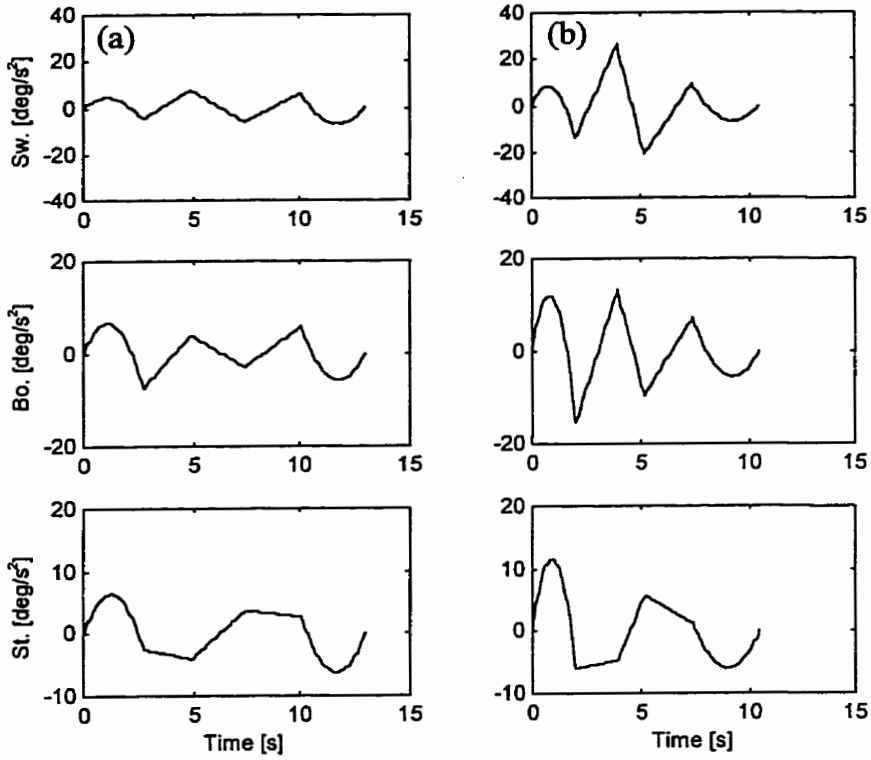


Fig. 4.10: Joint acceleration profiles: (a) initial solution; (b) optimal solution.

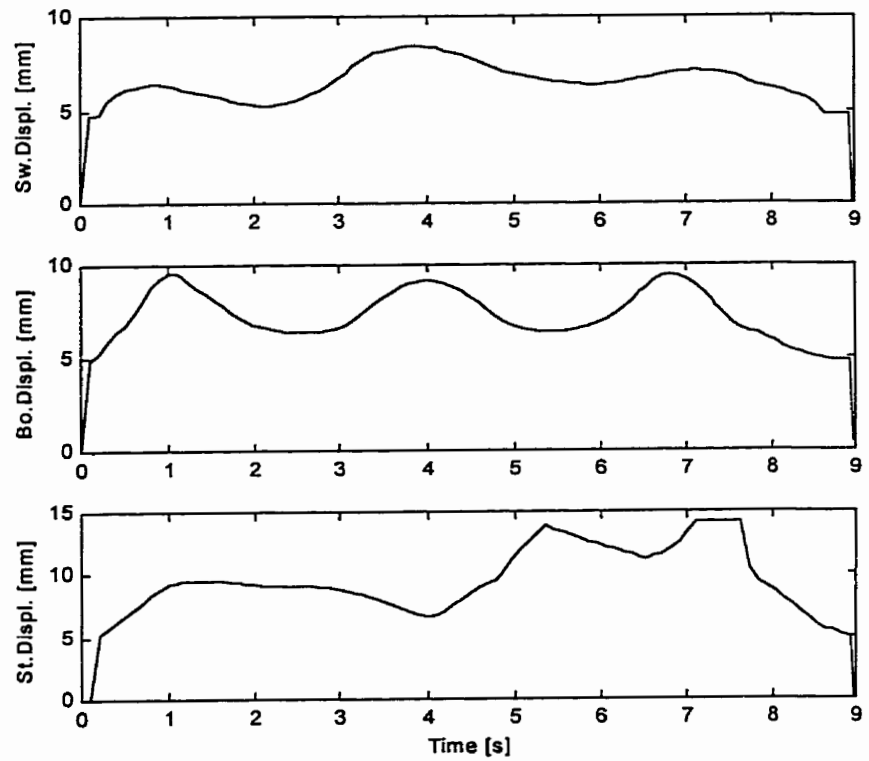


Fig. 4.11: Swing, boom and stick spool displacements.

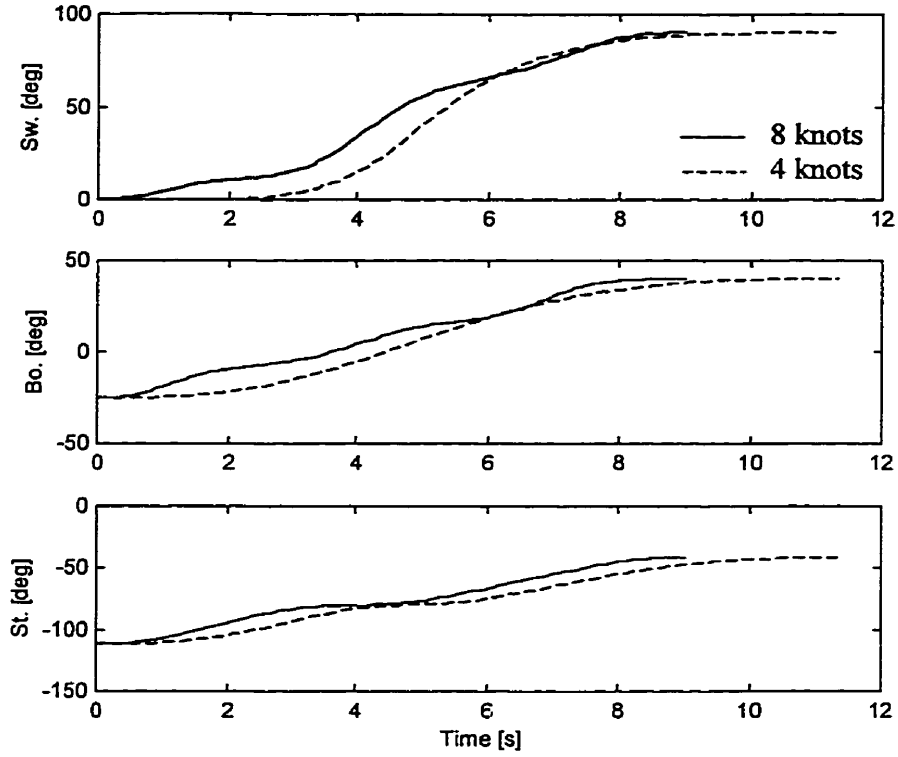


Fig. 4.12: Joint positions for four and eight knot points.

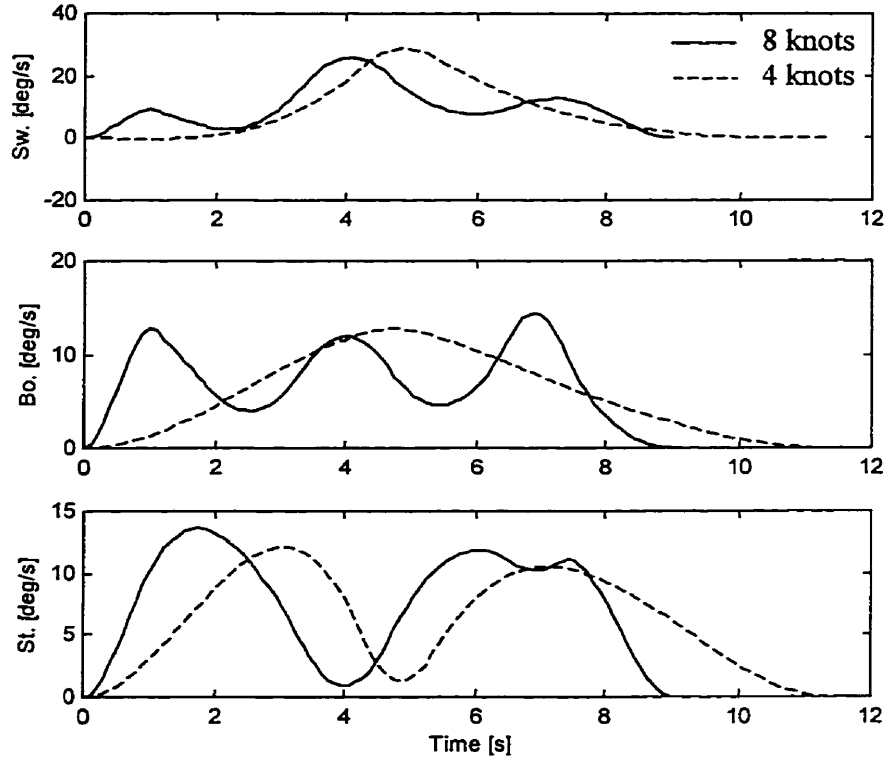


Fig. 4.13: Joint velocities for four and eight knot points.

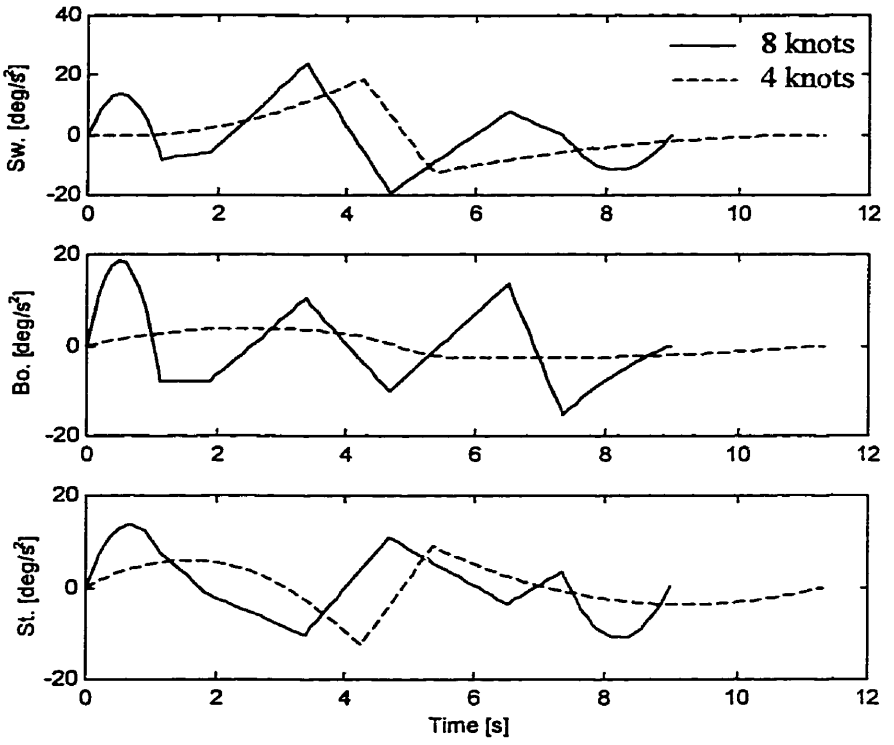


Fig. 4.14: Joint accelerations for four and eight knot points.

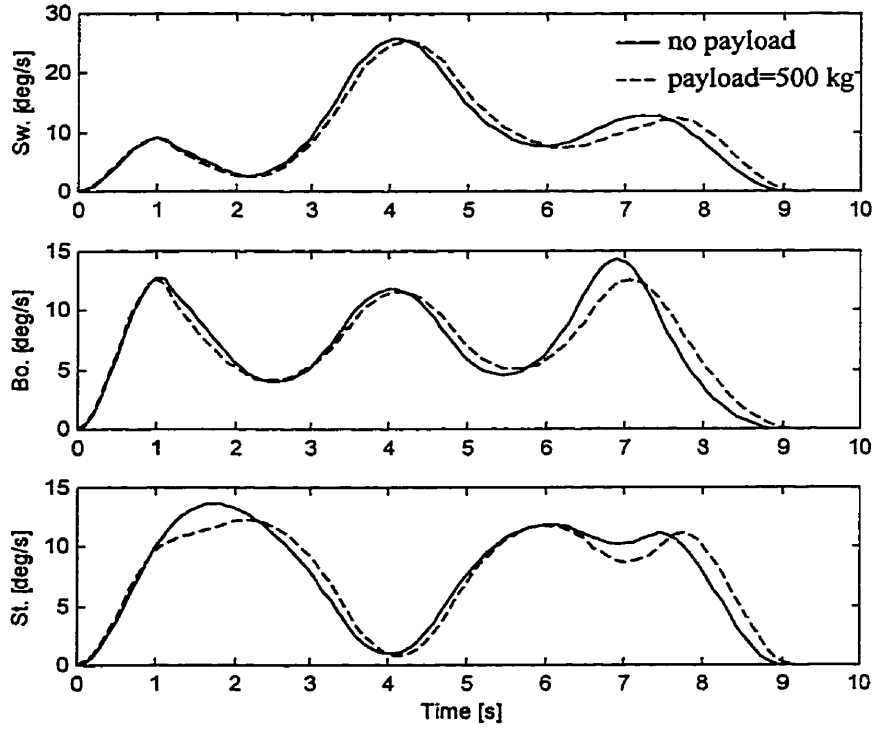


Fig. 4.15: Comparison of joint velocities for no payload versus payload.

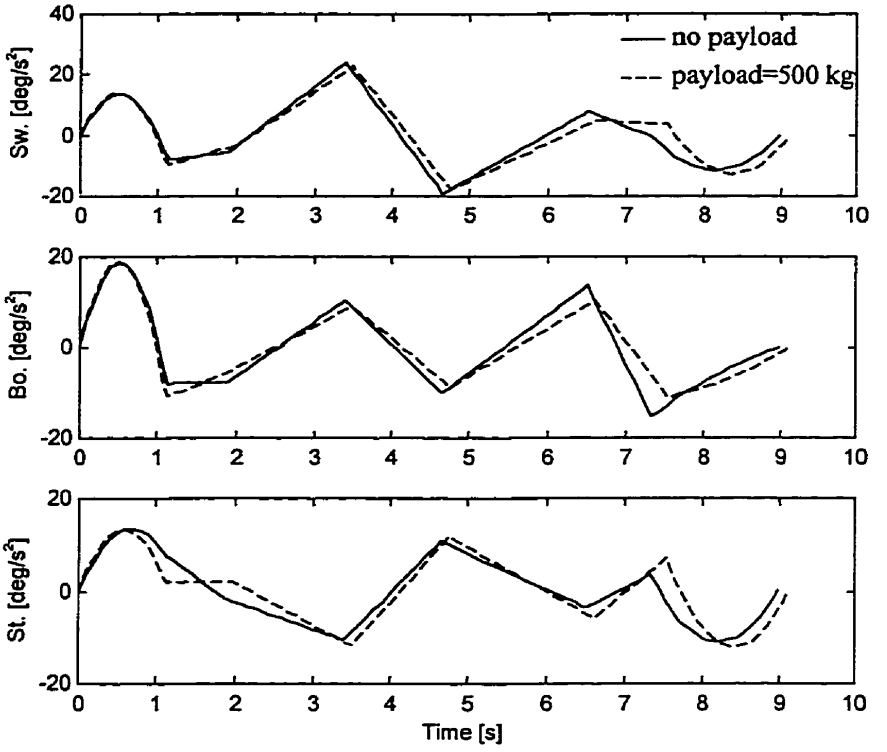


Fig. 4.16: Comparison of joint accelerations for no payload versus payload.

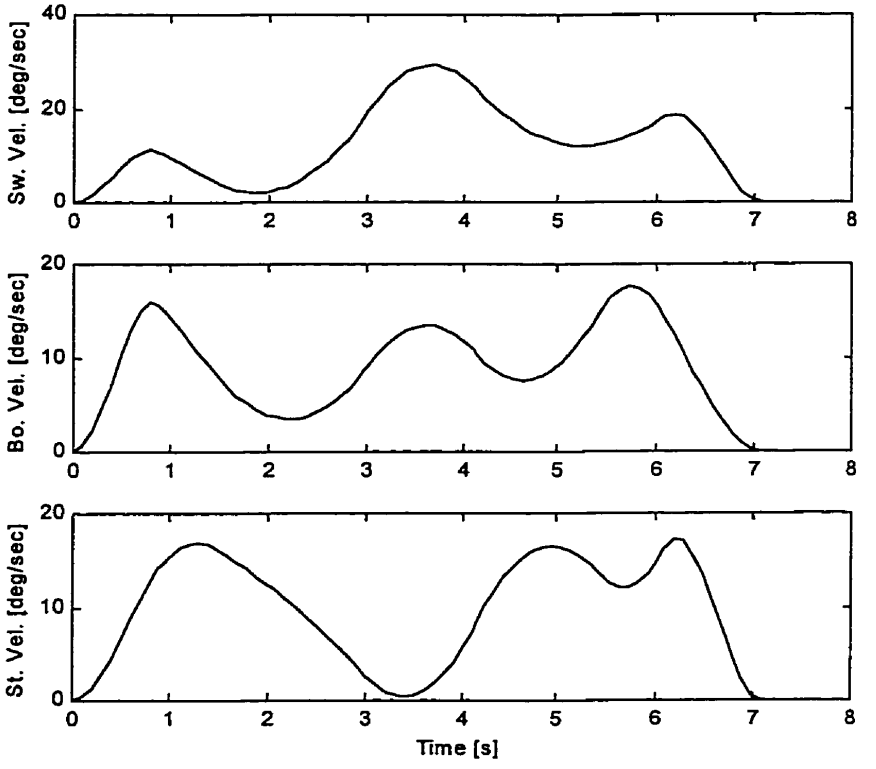


Fig. 4.17: Swing, boom and stick velocities.

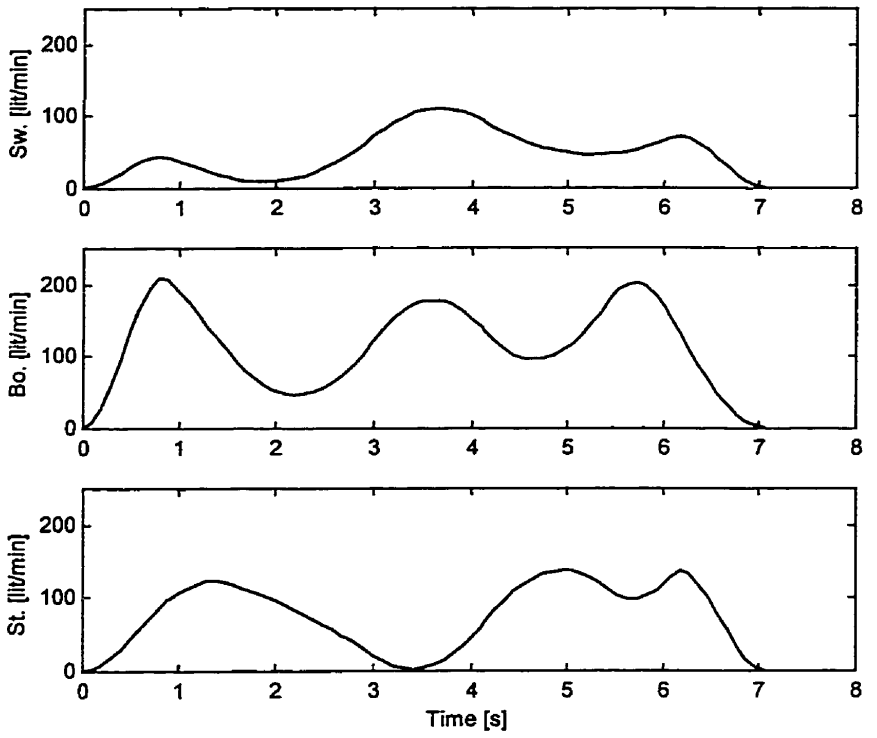


Fig. 4.18: Desired flow-rates to swing, boom and stick.

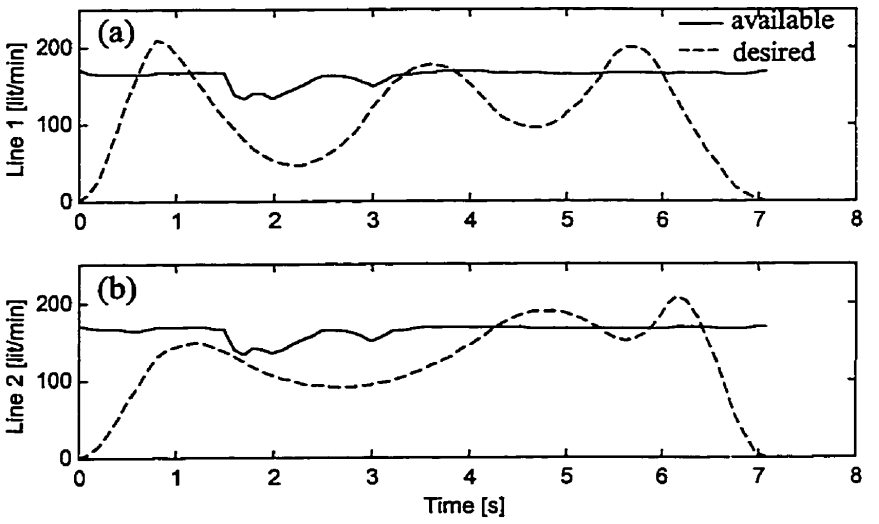


Fig. 4.19: Desired flow-rates: (a) to line 1 (boom), (b) to line 2 (swing and stick), versus maximum available pump flow.

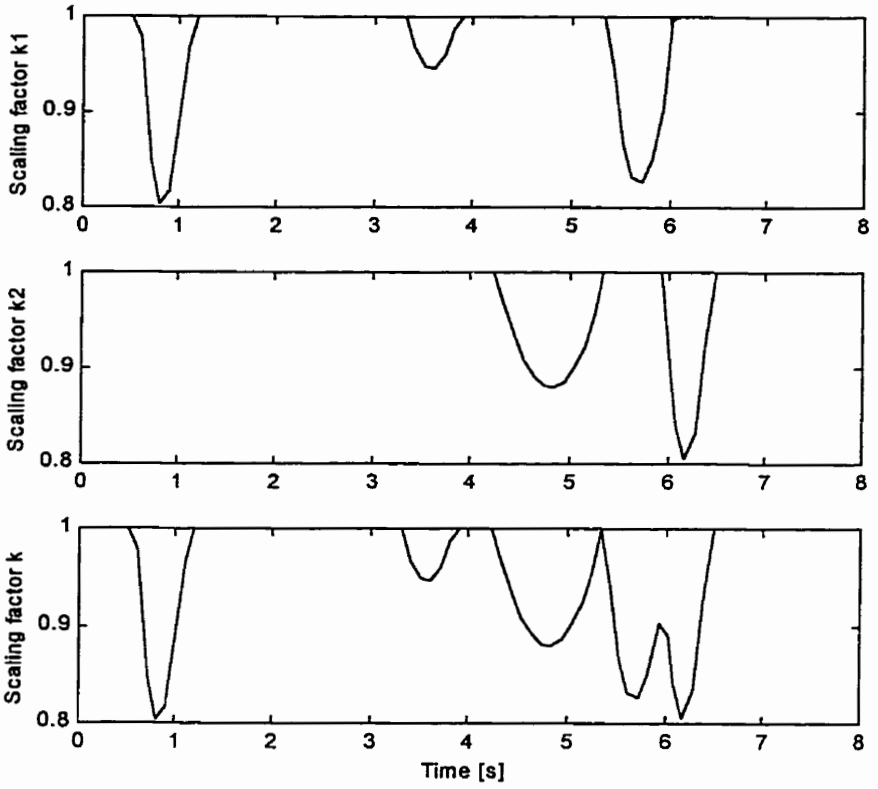


Fig. 4.20: Flow-rates scaling factors.

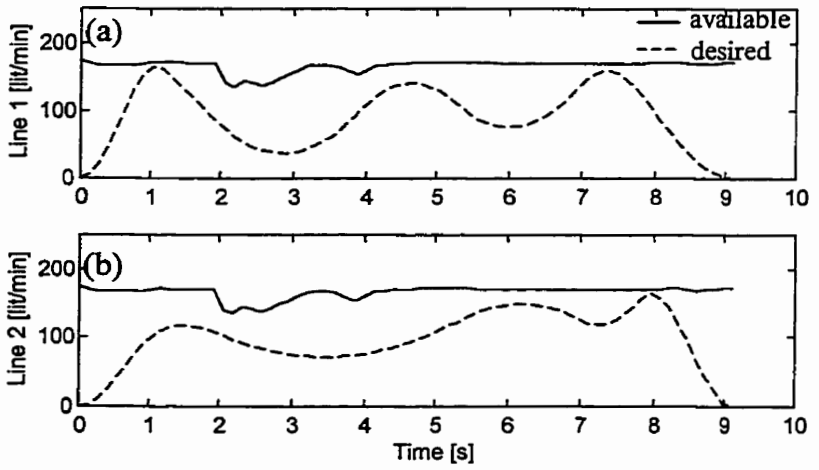


Fig. 4.21: Scaled desired flow-rates: (a) to line 1 (boom); (b) to line 2 (swing and stick), versus available pump flow.

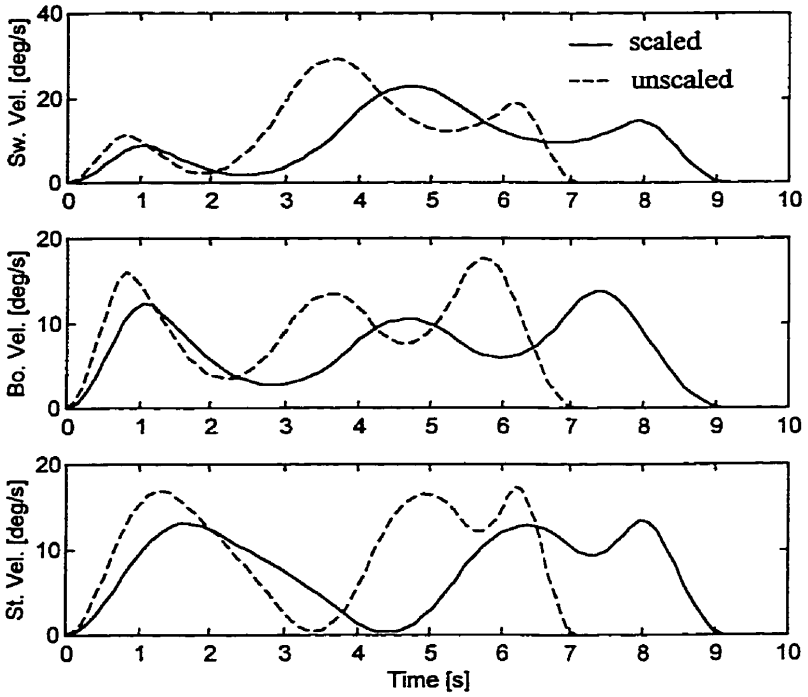


Fig. 4.22: Unscaled versus scaled swing, boom and stick velocities.

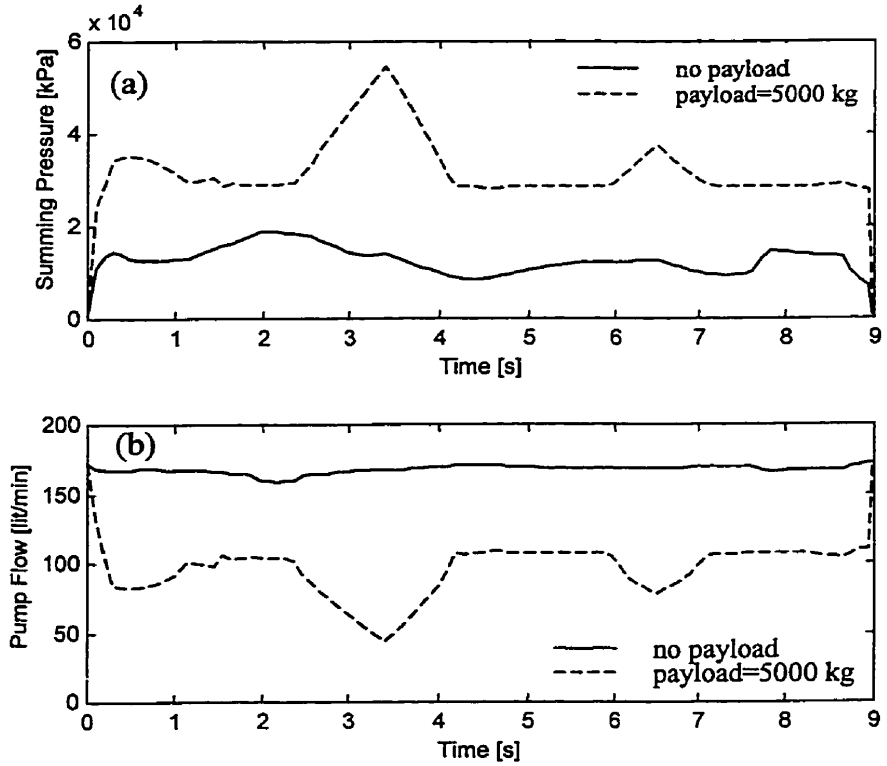


Fig. 4.23: (a) Summing pressure, $P_1 + P_2$; (b) Maximum available pump flow-rates.

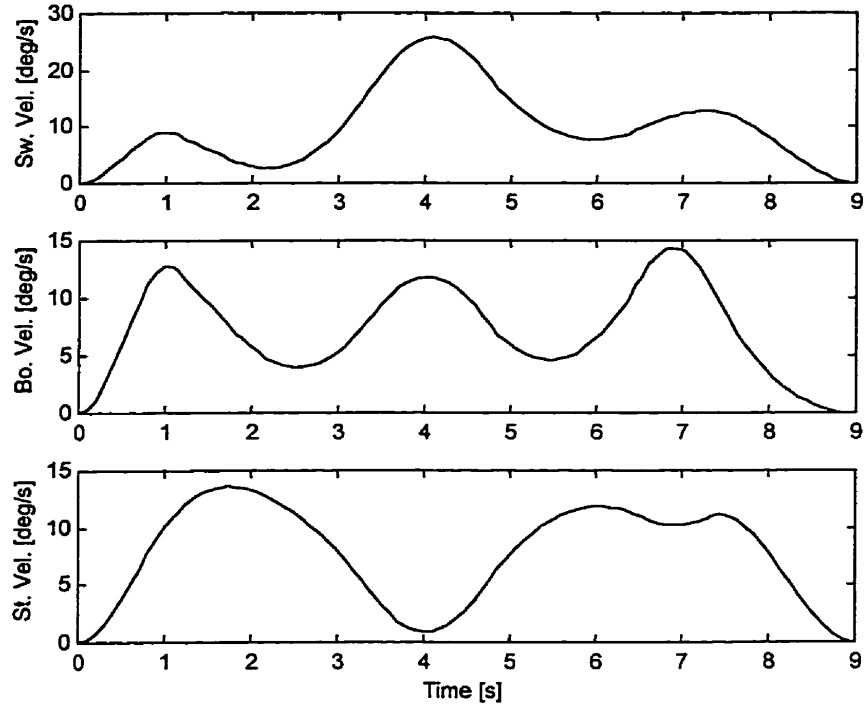


Fig. 4.24: Swing, boom and stick velocities.

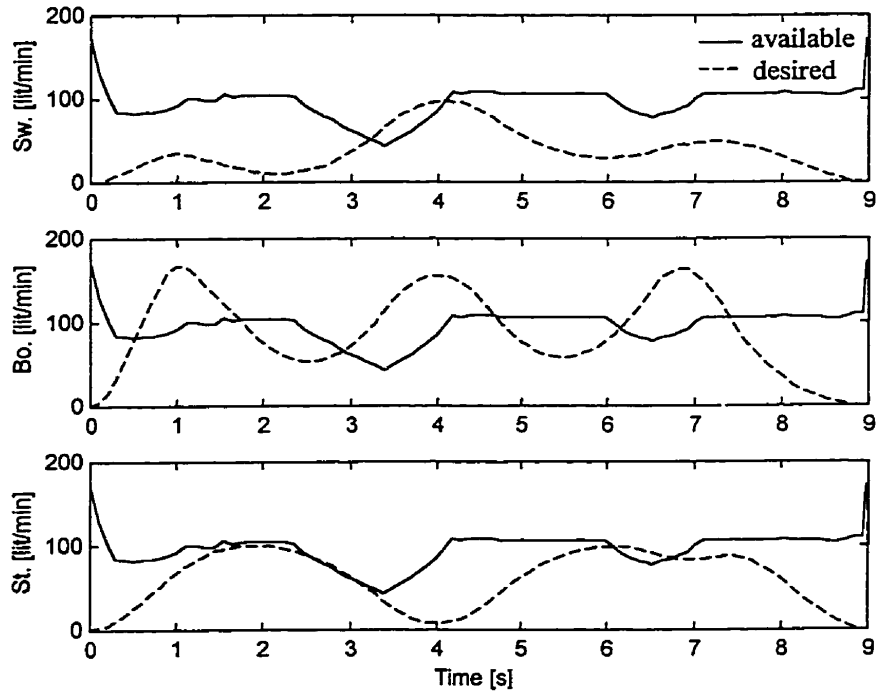


Fig. 4.25: Desired flow-rates to swing, boom and stick versus maximum available pump flow.

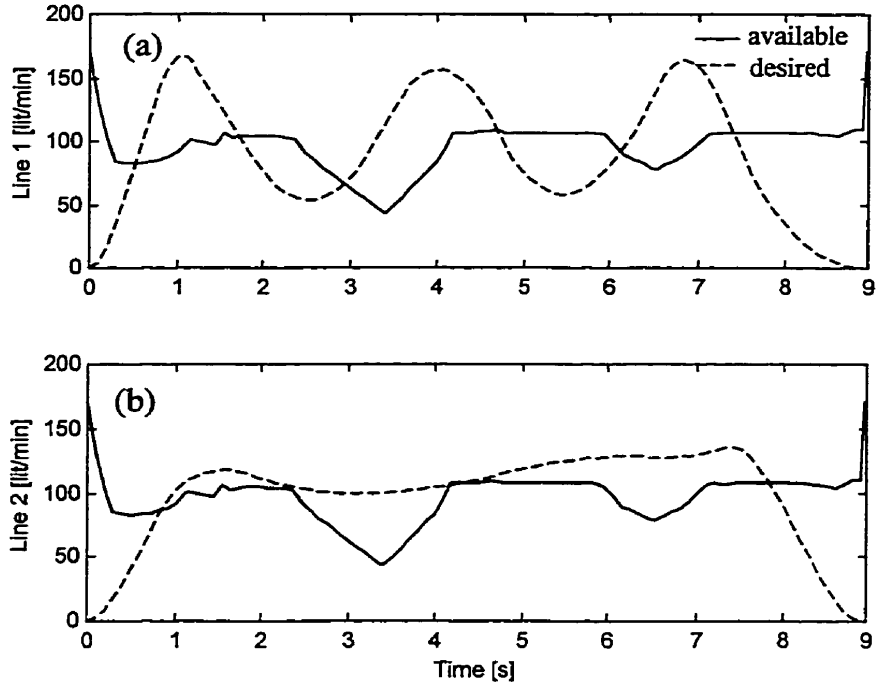


Fig. 4.26: Desired flow-rates: (a) to line 1 (boom), (b) to line 2 (swing and stick), versus maximum available pump flow.

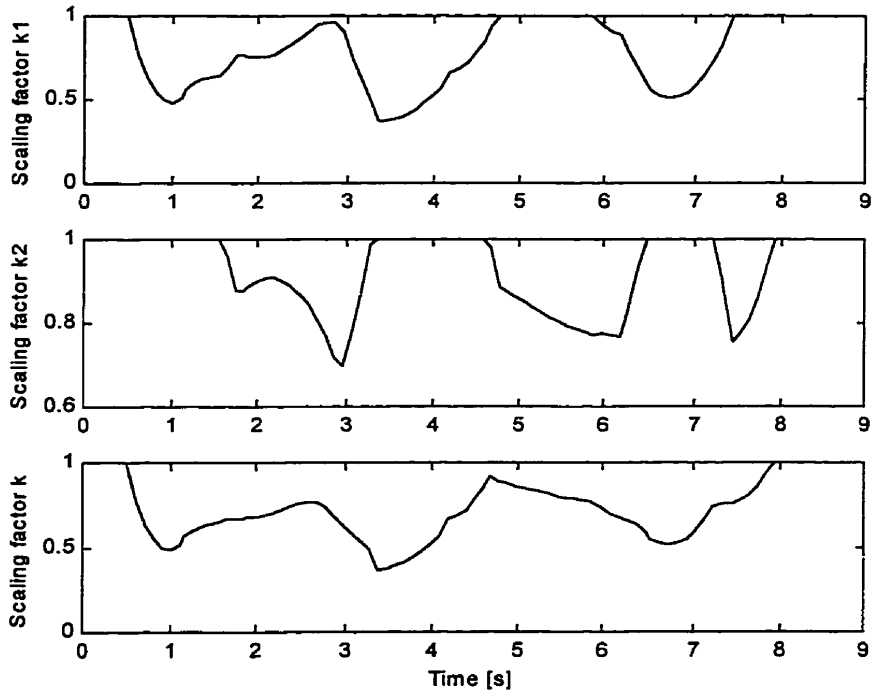


Fig. 4.27: Flow-rates scaling factors.

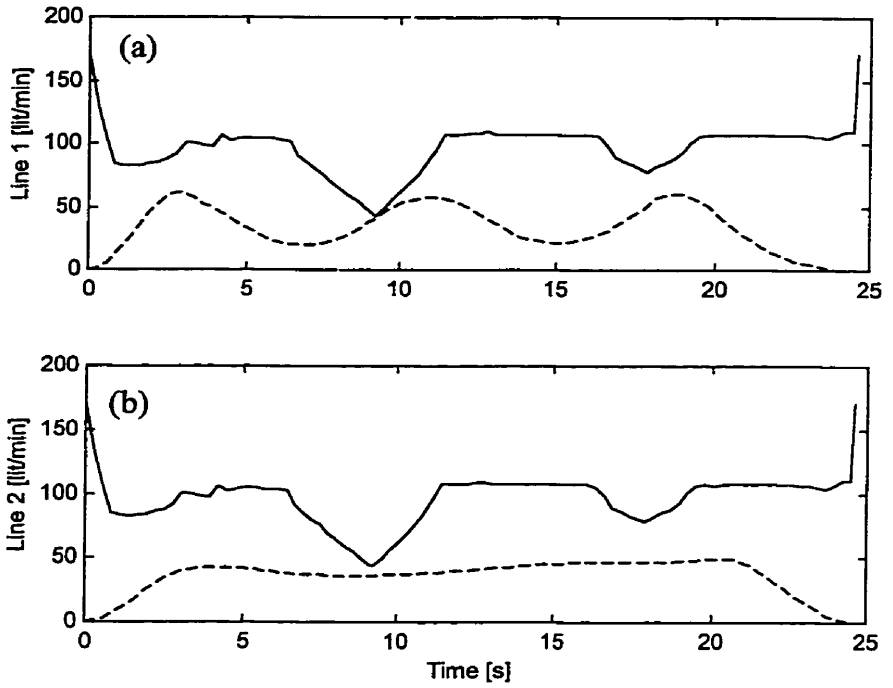
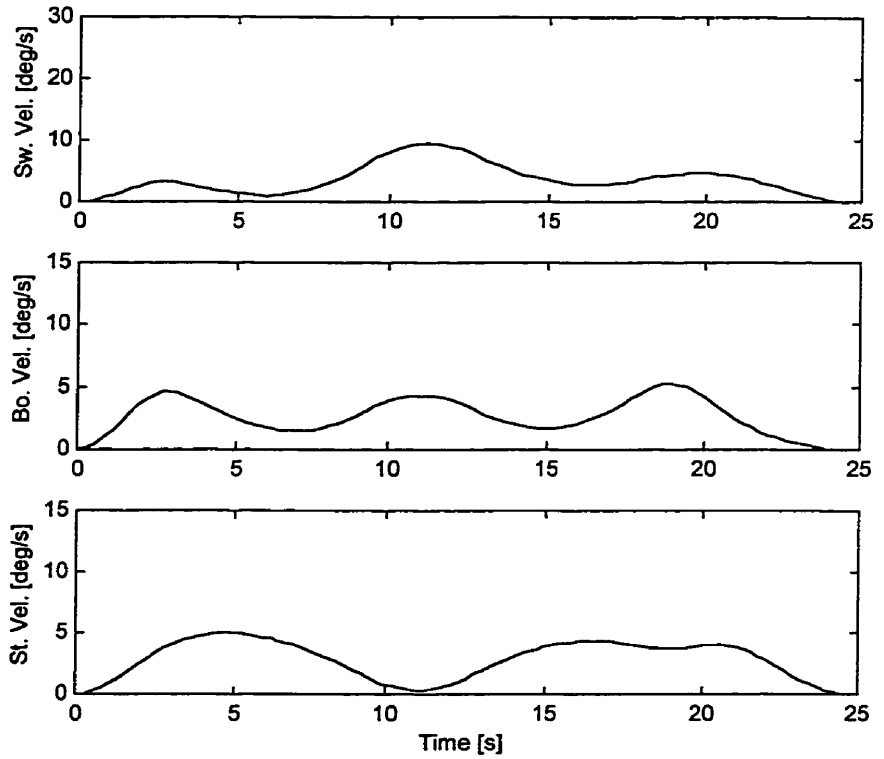


Fig. 4.28: Scaled desired flow-rates: (a) to line 1 (boom); (b) to line 2 (swing and stick), versus available pump flow.



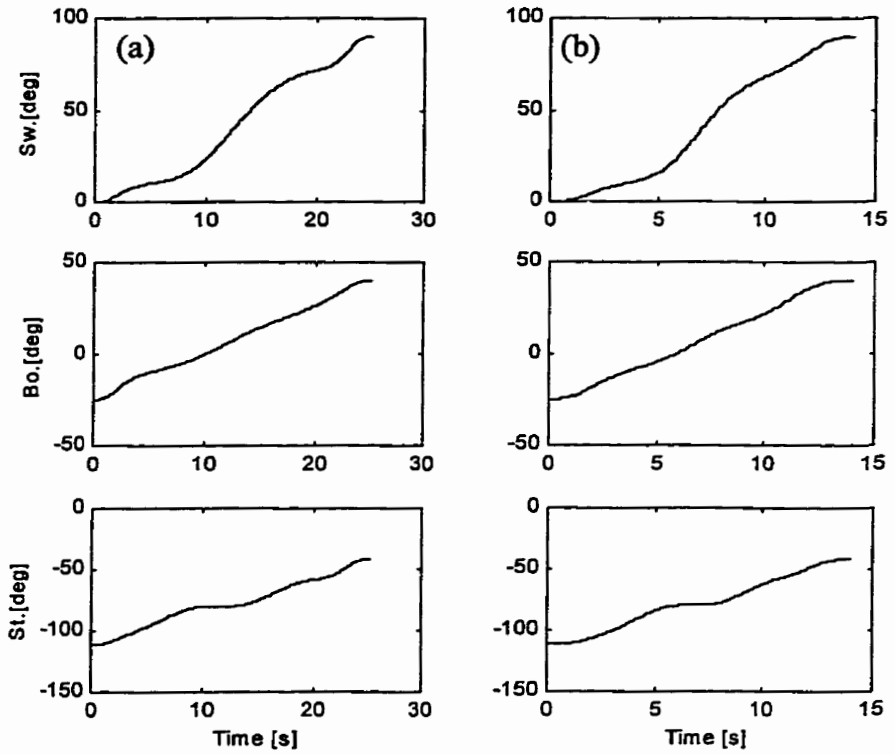


Fig. 4.30: Joint position profiles: (a) initial solution; (b) optimal solution.

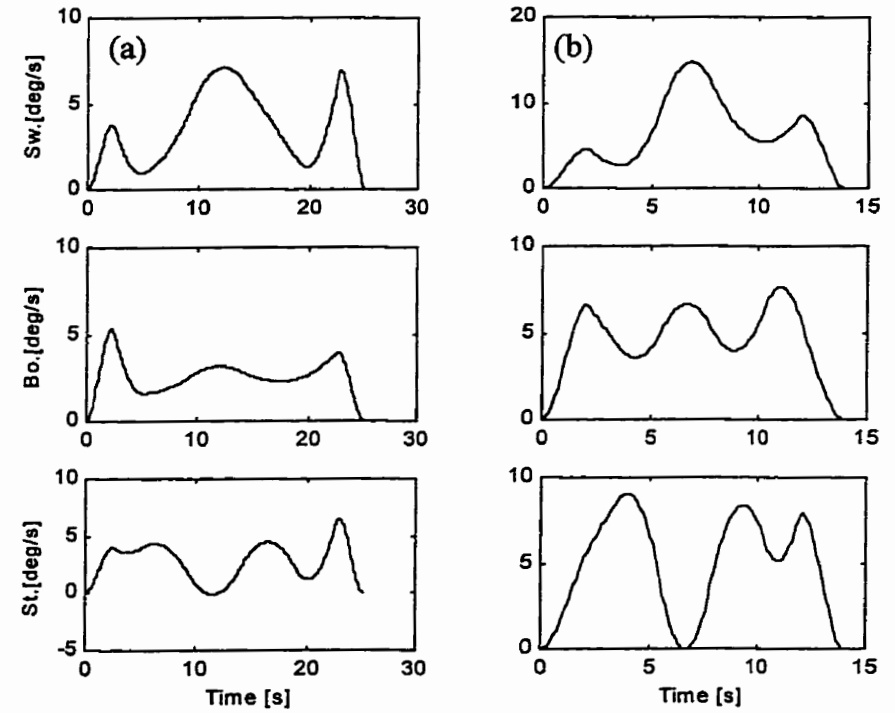


Fig. 4.31: Joint velocity profiles: (a) initial solution; (b) optimal solution.

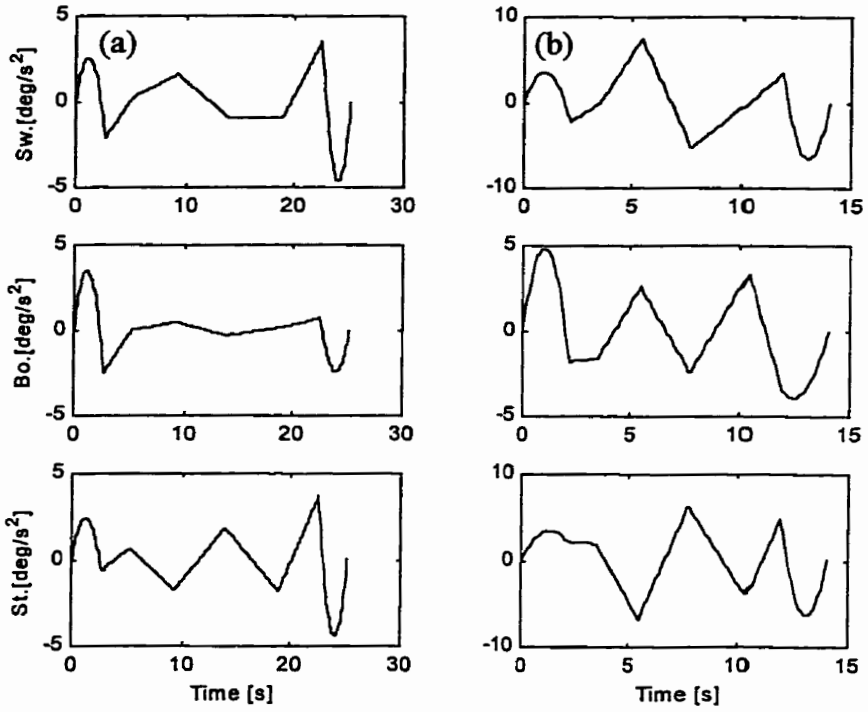


Fig. 4.32: Joint acceleration profiles: (a) initial solution; (b) optimal solution.

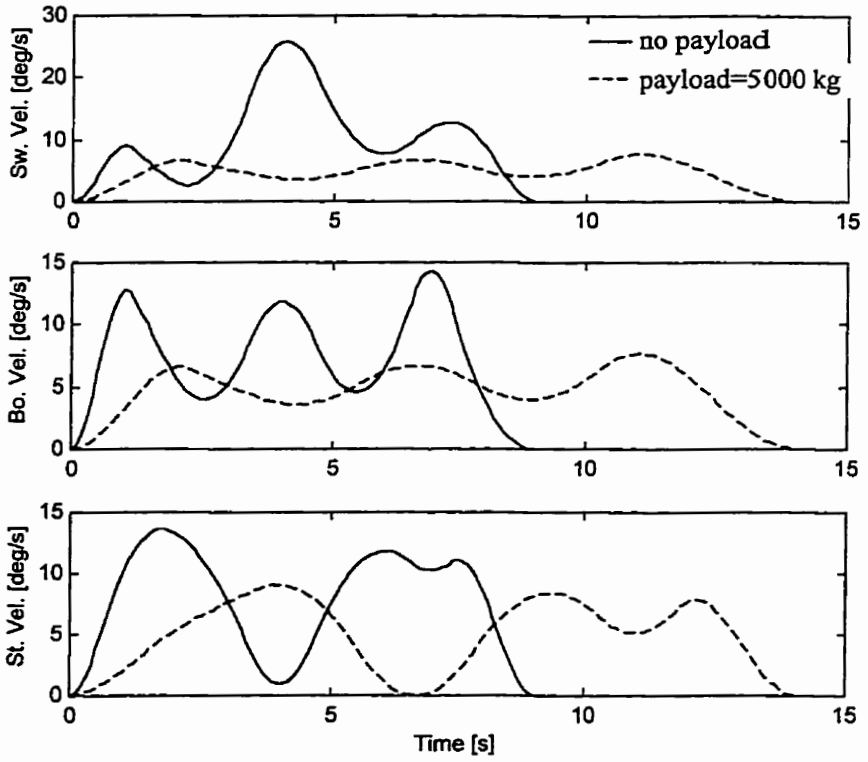


Fig. 4.33: Comparison of joint velocities for no payload versus payload.

4.2. Example 2

This example presents a test performed to investigate the effect of increasing the number of knot points on the error between the desired path and the approximated path obtained using spline functions in joint space.

In this test the manipulator task was to move along a straight-line from a start position to a goal position. The start position of the end-effector in Cartesian coordinate was represented by $[x_s, y_s, z_s]$=[4.352 m, -0.12 m, -1.757 m] and the goal position by $[x_g, y_g, z_g]$=[5.8359 m, -0.12 m, 2.4445 m] as shown in Fig. 4.34. These Cartesian end-points correspond to the joint space points [swing=0 deg, boom=-4.4121 deg., stick=-126.67 deg.] and [swing=0 deg., boom=40.58 deg., stick=-71.9 deg.]. Note that the given task required only boom and stick motion. Initially, two knot points were introduced between the start and goal positions. Fig. 4.35 shows that if the knot points specifying the desired path in the Cartesian space are located at equidistant intervals, the corresponding points in the joint space are not equally spaced and the path connecting these points is not a straight line. Given the knot points shown in Fig. 4.35b, the joint trajectories were generated using spline functions and the optimization algorithm was employed to optimize the joint trajectories. The optimization algorithm was run twice with the same values for the stopping tolerances, $\varepsilon_1 = \varepsilon_2 = 10^{-4}$ and constant $\mu = 0.05$, as in Example 1.

The same test was applied for eight knot points located at equidistant intervals along the path. Figures 4.36 and 4.37 show the optimized position and velocity joint trajectories obtained for four knot points along the path and eight knot points respectively. It can be seen that for eight knot points trajectories the motion time is

shorter than for four knot points trajectories and therefore higher joint velocities are obtained.

The foregoing trajectories were then mapped (pointwise) into the Cartesian space to obtain $x(t), y(t), z(t)$, which represent the desired trajectories in Cartesian space (see Fig. 4.38). Figure 4.39 shows that increasing the number of knot points from four to eight almost eliminates the error between the desired path and the approximated one. Therefore, to reduce the error between the desired path and the approximated one, additional knot points should be specified along the desired path. Figure 4.40 indicates smaller jumps in the Cartesian velocities for eight knot points. Therefore, the result for eight knot points improved even in terms of Cartesian velocities. Figure 4.41 shows the optimal velocity profiles for the case with no payload and for the case with 5000 kg payload.

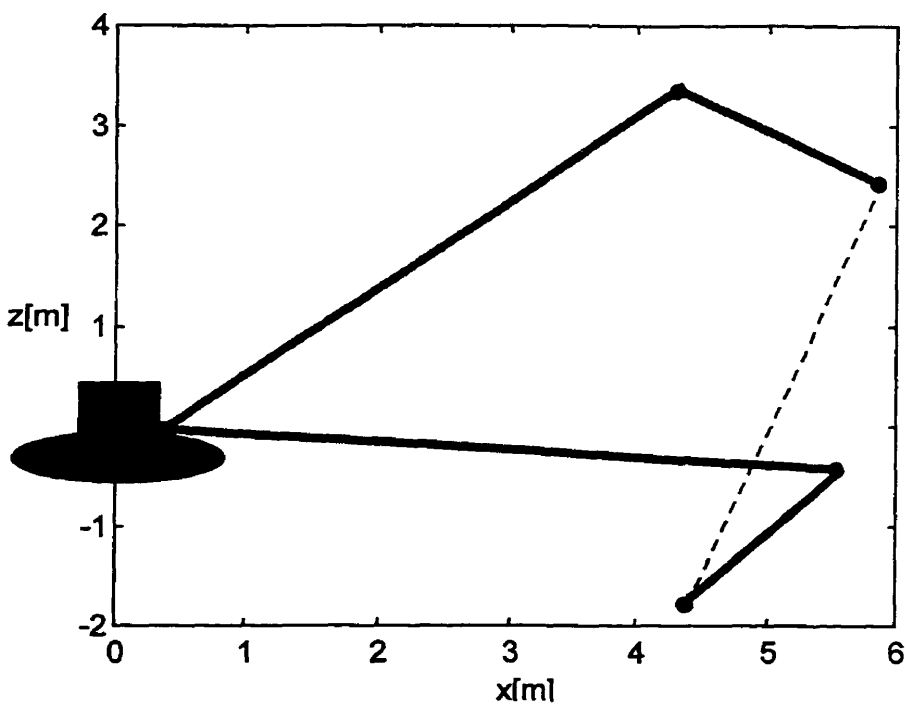


Fig. 4.34: Straight line Cartesian path.

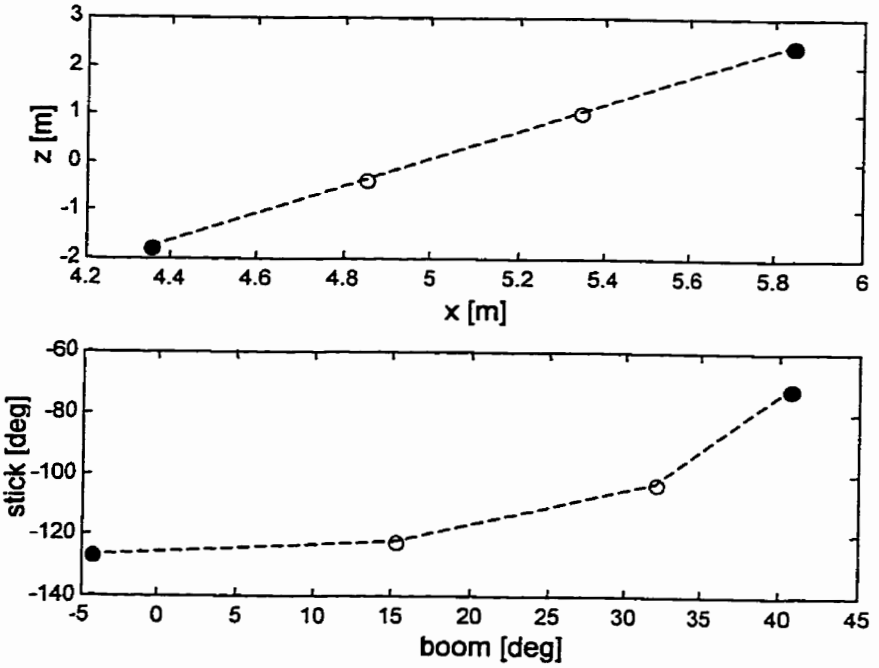


Fig. 4.35: Path in: (a) Cartesian space; (b) joint space.

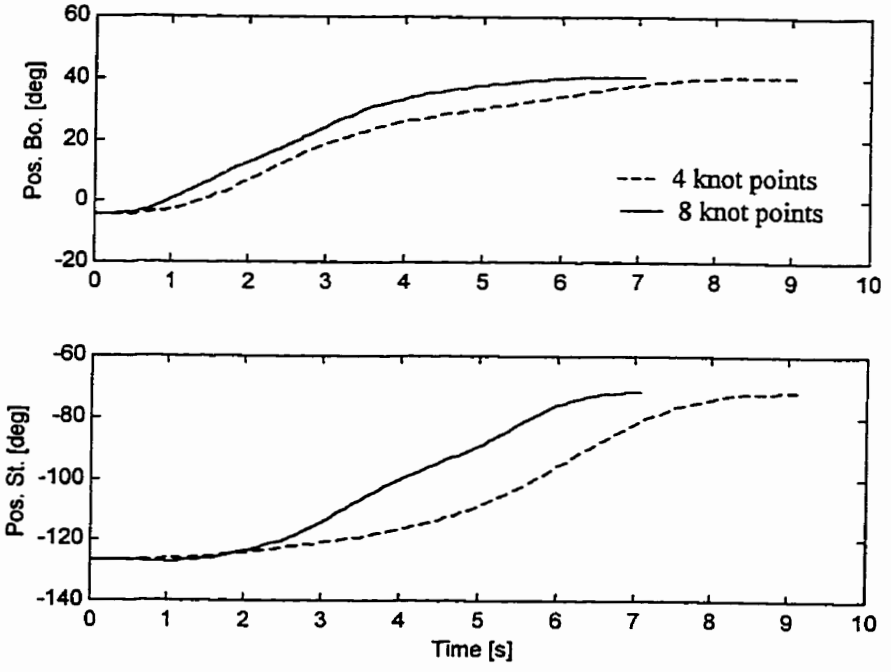


Fig. 4.36: Joint position trajectories.

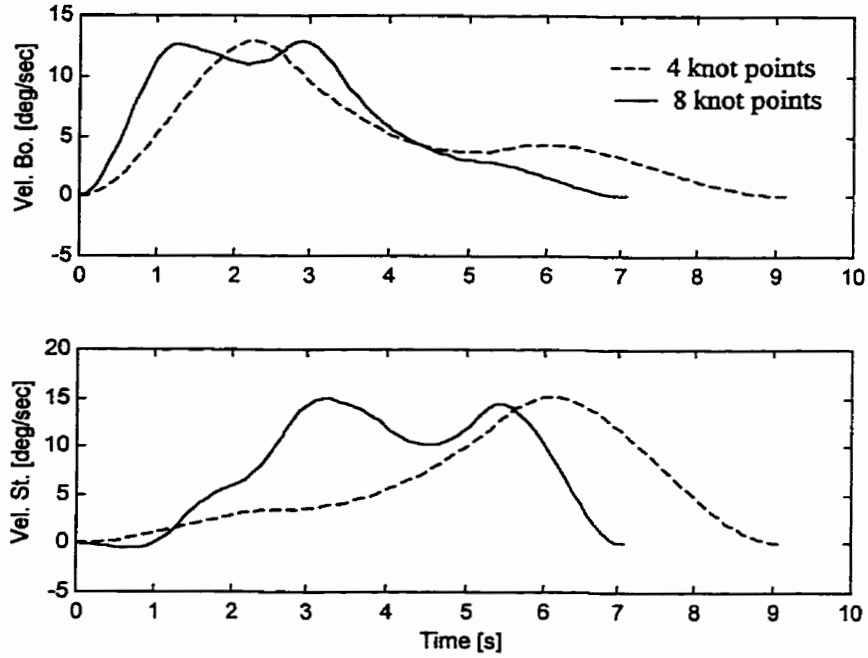


Fig. 4.37: Joint velocity trajectories.

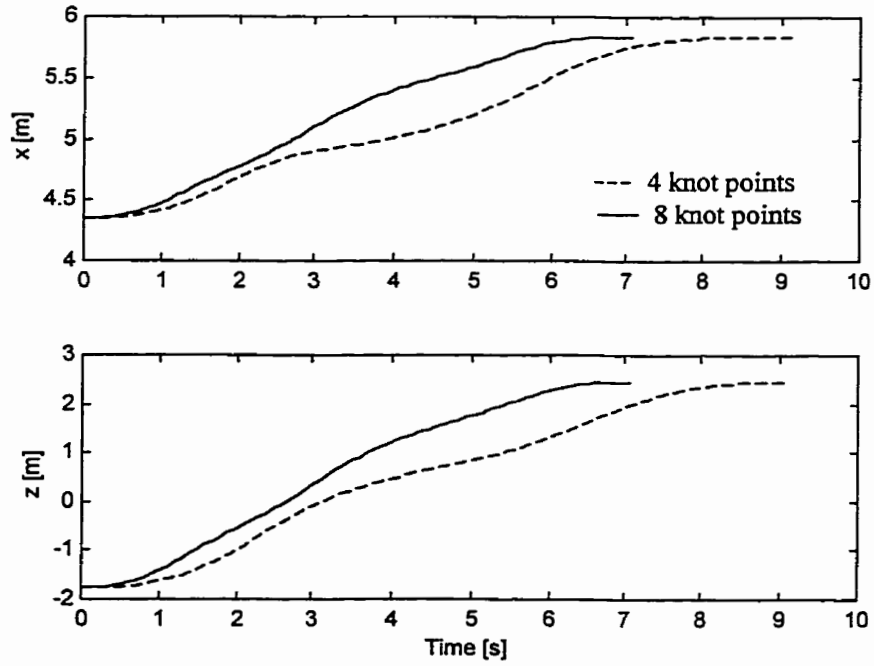


Fig. 4.38: Cartesian position trajectories.

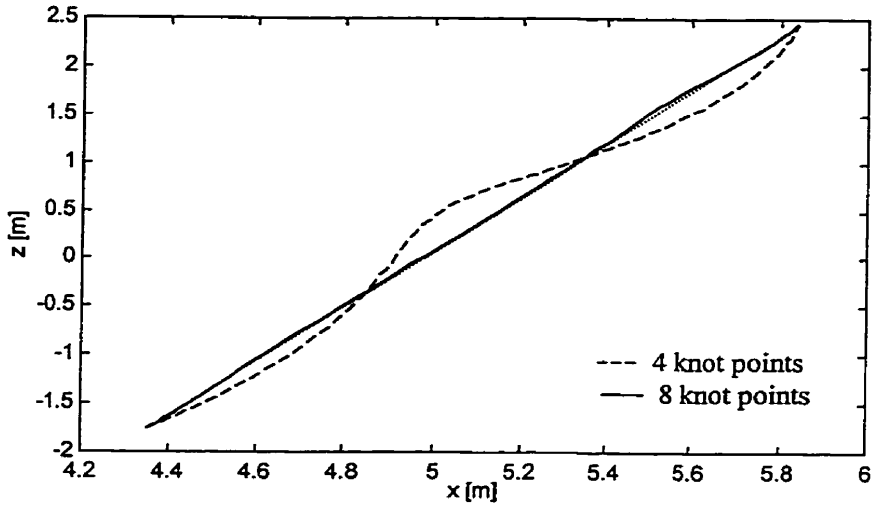


Fig. 4.39: Effect of increasing number of knot points on path error.

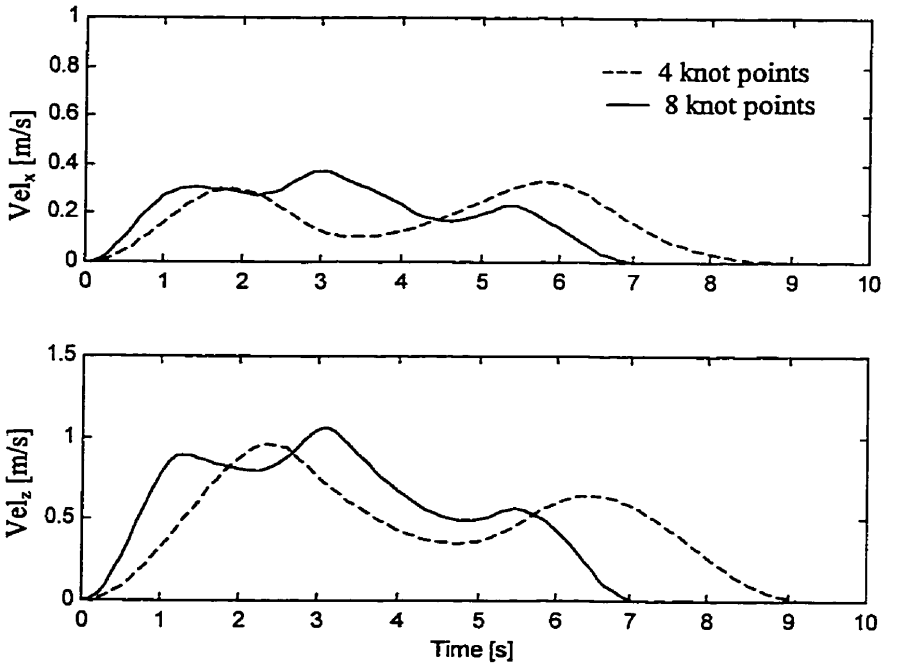


Fig. 4.40: Cartesian velocity trajectories.

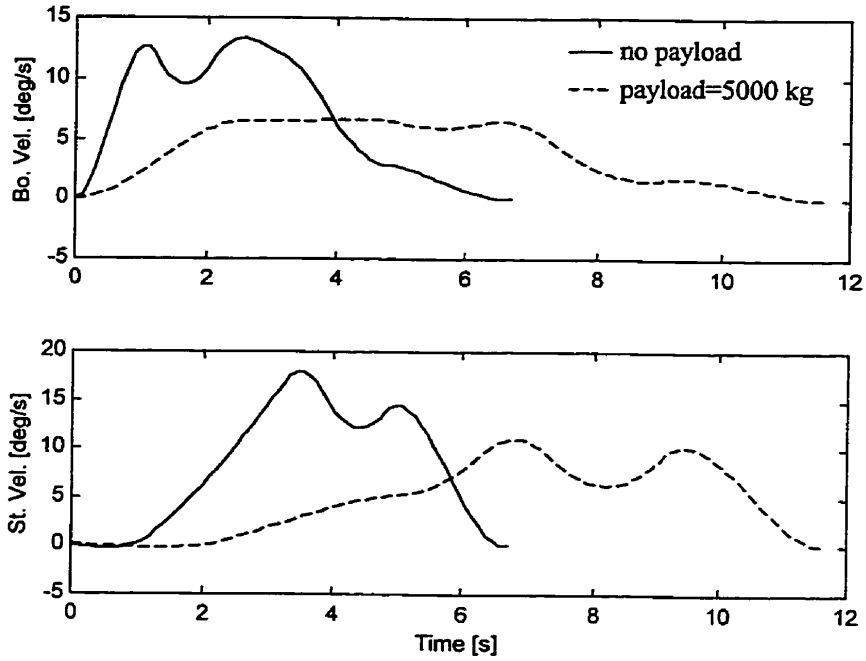


Fig. 4.41: Comparison of optimal joint velocities for no payload versus payload.

Chapter 5

Conclusions

5.1 Achievements

In this thesis, the optimal motion planning along a specified path was performed for a hydraulic manipulator. The path specified by knot points was mapped from Cartesian space to joint space using inverse kinematics and joint trajectories were generated using spline functions. The resulting joint trajectories were smooth everywhere inclusive at knot points (i.e. position, velocity, acceleration were continuous over the duration of the motion).

Downhill simplex method was adopted to optimize the joint trajectories. The search space for the optimization was the space of the time intervals between adjacent knot points. The objective of optimization was to adjust the time intervals subject to constraints imposed by the limited capabilities of the actuators, so that the total motion time was minimized. The actuator capabilities are limited due to the fact that they are powered by a limited power engine.

Each solution obtained by the search was evaluated for feasibility. A solution was feasible, if the corresponding trajectory was dynamically realizable given the actuator constraints. If it was not, the trajectory was modified by a constant scaling of the joint velocity profile, so that the total movement duration was scaled without changing the actual path through space. The bounds on joint velocities were computed with respect to

the combination of actuator and linkage effects, such as loading, inter-actuator coupling and power limitation.

The main contributions of this work are: (1) the incorporation of complex and coupled hydraulic actuation in optimal trajectory planning for heavy-duty hydraulic manipulators and, (2) utilization of an optimization algorithm, downhill simplex method, that has been shown to be effective in solving the optimal trajectory planning problem addressed here. Although the downhill simplex method does not guarantee that the global minimum can be reached, it improves the chances of reaching the global minimum by restarting the optimization at the claimed minimum.

5.2. Future Development

This work could be extended to the global optimal motion computation. A method, such as workspace discretization, can be used to generate all possible paths between the start and goal position and the method developed here can be used to test each one for the time optimality. The one with the lowest time is obviously the global optimal.

Another possible development could be the incorporation of the machine stability constraint in optimal motion planning. Ghasempoor (1994) defined a measure of stability margin, which included the effect of dynamic forces/moments arising from the manipulator motion. For the purpose of optimal motion planning with machine stability constraint, the motion should be computed, so that these forces/moments will not be larger than the machine can handle and cause the machine to tip-over.

References

Ahlberg J. H., Nilson E. N. and Walsh J. L., "The Theory of Splines and Their Applications", Academic Press, New York and London, 1967.

Bobrow, J.E., Dubowsky, S., and Gibson, J.S., "Time-Optimal Control of Robotic Manipulators along Specified Paths", *International Journal of Robotics Research*, Vol.4, No. 3, 1985, pp. 3-17.

Chen, Y. and Desrochers, A. A., " On the Structure of the Minimum-Time Control Law for Robotic Manipulators", *Progress in Robotics and Intelligent Systems*, Vol. 1, edited by Zobrist, G. W., Ho, C. Y., Ablex Publishing Corporation Norwood, New Jersey, 1994, pp. 136-181.

Craig, J., "Adaptive Control of Mechanical Manipulators", Addison-Wesley Publishing Company, New York, 1988.

Dennis, J. E. Jr. and Woods, D. J., "New Computing Environments: Microcomputers in Large-Scale Computing", edited by A. Wouk, SIAM, 1987, pp. 116-122.

Ghasempoor-Nobandgany, A., "A Measure of Stability for Mobile Manipulators with Application to Heavy-Duty Hydraulic Machines", M.Sc. Thesis, Dept. of Mechanical Engineering, The University of Manitoba, 1994.

Hollerbach, J. M., "Dynamic Scaling of Manipulator Trajectories", *ASME Journal of Dynamic Systems, Measurement, and Control,* Vol. 106, 1984, pp 102-106.

Hornick, M. L. and Ravani, B., "Computer-Aided Off-Line Planning and Programming of Robot Motion", *International Journal of Robotics Research,* Vol. 4, No. 4, 1986, pp. 18-31.

Krishna, M, "Optimal Motion Generation for Hydraulic Robots", Ph.D. thesis proposal, Carnegie Mellon University, March 1998, http://www.cs.cmu.edu/~mkrishna.

Lin, C.S., Chang, P.R. and Luh, J.Y.S., "Formulation and optimization of cubic polynomial joint trajectories for mechanical manipulators", *Proc. IEEE Conference on Decision and Control*, Orlando, FL, 1983, pp. 1066-1073.

Luh, J.Y.S. and Lin, C. S., "Optimum path planning for mechanical manipulators", *ASME Journal of Dynamic Systems, Measurement, and Control,* Vol. 102, 1981, pp. 142-151.

Luh, J.Y.S., "An Anatomy of Industrial Robots and Their Controls", *IEEE Transactions on Automatic Control,* Vol. AC-28, No. 2, 1983, pp. 133-153.

Luh, J.Y.S., Walker, M.W. and Paul, R.P.C., "On-line Computational Scheme for Mechanical Manipulators", *ASME Journal of Dynamics Systems, Measurement, and Control*, Vol. 102, 1980, pp. 69-76.

Medanic, J., Yuan and M., Medanic, B., "Robust Multivariable Non-Linear Control of a Two Link Excavator: Part I", *Proc. of the 36th Conference on Decision and Control*, San Diego, California, 1997, pp. 4231-4236.

Merritt, H.E., "Hydraulic Control Systems", Wiley, New York, 1967.

Press, W. H., Teukolsky, S.A., Vetterling, W. T and Flannery, B.P., "Numerical Recipes in C: The Art of Scientific Computing", Second Edition, Cambridge University Press, Cambridge, 1992.

Rajan, V.T., "Minimum Time Trajectory Planning", *Proc. IEEE Conference on Robotics and Automation*, St. Louis, MO, 1985, pp. 759-764.

Ranky, P.G and Ho, C.Y, "Robot Modelling: Control and Applications with Software", Springer-Verlag, Berlin, 1985.

Rao, S.S., "Optimization Theory and Applications", John Wiley & Sons Ltd., New York, 1984, pp. 292-300.

Sepehri, N, Frenette, R and Lawrence, P.D., "Proportional Hydraulic Control", US Patent No. 5,167,121 (allowed Dec. 1, 1992, Date U.S. filled Jun. 25,1991, Application No. 720,378).

Sepehri, N., "Dynamic Simulation and Control of Teleoperated Heavy-Duty Hydraulic Manipulators", PhD thesis, Dept. of Mechanical Engineering, University of British Columbia, 1990.

Shiller, Z. and Dubowsky, S., "On Computing the Global Time-Optimal Motions of Robotic Manipulators in the Presence of Obstacles", *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 6, 1991, pp. 785-797

Shiller, Z. and Dubowsky, S., "On the Optimal Control of Robotic Manipulators with Actuator and End-Effector Constraints", *Proc. IEEE Conference on Robotics and Automation*, St. Louis, MO, 1985, pp. 614-620.

Shin, K.,G.and McKay, N. D., "Minimum-Time Control of Robotic Manipulators with Geometric Path Constraints", *IEEE Transactions on Automatic Control*, Vol. AC-30, No. 6, 1985, pp. 531-541.

Singh, N., Zghal, H., Sepehri, N., Balakrishnan, S. and Lawrence, P.D., "Coordinated-Motion Control of Heavy-Duty Industrial Machines with Redundancy", *Robotica*, Vol. 13, 1995, pp. 623-633.

Stokic, D. M., Vukobratovic, M. K and Lekovic D. B., "Simulation of Robots in Flexible Manufacturing Cells", *Robotics and Computer-Integrated Manufacturing*, Vol. 8, No. 1, 1991, pp. 1-8.

Tafazoli, S., "Identification of Frictional Effects and Structural Dynamics for Improved Control of Hydraulic Manipulators," PhD thesis, Electrical & Computer Engineering Dept., University of British Columbia, 1997.

Tarkiainen, M. and Shiller, Z., "Time Optimal Motions of Manipulators with Actuator Dynamics", *Proc. IEEE Conference on Robotics and Automation*, Atlanta, Georgia, 1993, pp. 725-730.

Tarn T.J., Bejczy, A.K., Yun, X., and Li, Z., "Effect of Motor Dynamics on nonlinear Feed-back Robot Arm Control," *IEEE Trans. on Robotics and Automation,* Vol. 7, pp. 114-122.

Walsh, G. R., "Method of Optimization", John Wiley & Sons Ltd., London, 1975

Wapenhans, H., Holzl, J., Steinle, J. and Pfeiffer, F., " Optimal Trajectory Planning with Application to Industrial Robots", *International Journal of Advanced Manufacturing Technology*, Springer-Verlag London Limited, Vol. 9, No. 1, 1994, pp. 49-55.

Youcef-Toumi, K., and Kuo, A.T.Y., "High Speed Trajectory Control of A Direct-Drive Manipulator," *Proc. of 26th Conference on Decision and Control*, Los Angeles, CA, pp. 2202-2208.

Appendices

A. Excavator-Based Manipulator Kinematics

In this section, the forward and inverse kinematics equations are derived for the 215B Caterpillar excavator-based machine.

A.1 Forward kinematics

Problem: *Given the joint angles (θ_1, θ_2, θ_3) calculate position and orientation of the end effector with respect to the base frame.*

The Denavit-Hartenberg parameters for the coordinate frames associated with all joints are listed in Table A.1. The coordinate frames associated with all joints are shown in Fig. A1.

Table A.1: Kinematic parameters of 215B Caterpillar excavator-based machine.

Joint	Link	α_i (deg.)	a_i (m)	d_i (m)	θ_i (deg.)
1	Swing	90	0.31	0.12	θ_1(unbounded)
2	Boom	0	5.19	0	θ_2 (-27 to 45)
3	Stick	0	1.8	0	θ_3 (-157 to -25)

An intermediate coordinate frame $\{x_{1a}, y_{1a}, z_{1a}\}$ is used to account for the fact that link 1 is not collinear with the x_0 axis. The homogeneous transformations have been computed by Singh, (1995) as following:

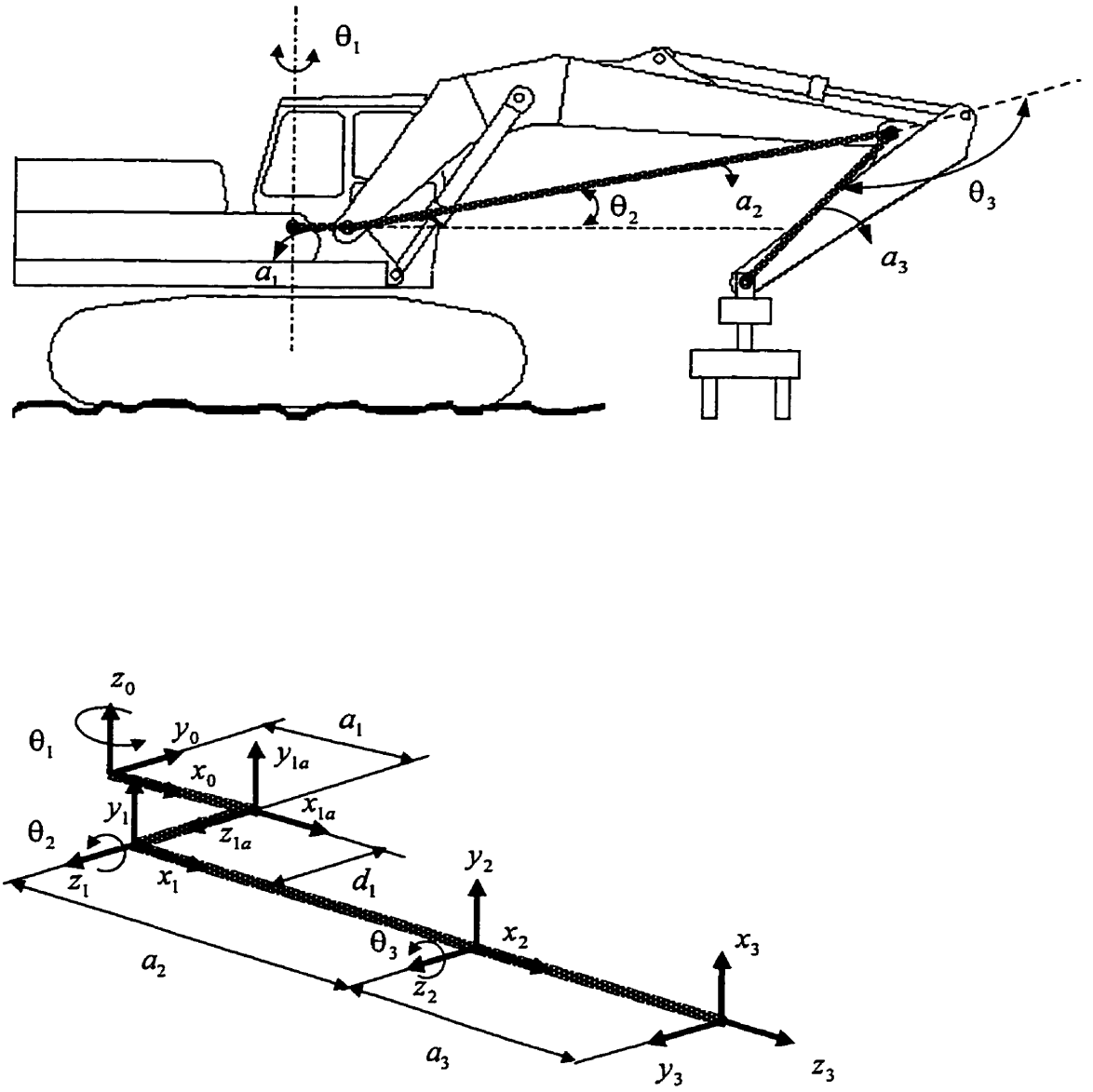


Fig. A.1: Coordinate frames.

$$A_{1a}^{0}=\begin{bmatrix} c_1 & 0 & s_1 & a_1c_1 \\ s_1 & 0 & -c_1 & a_1s_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad A_{1}^{1a}=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_{1}^{0}=A_{1a}^{0}A_{1}^{1a}=\begin{bmatrix} c_1 & 0 & s_1 & a_1c_1+d_1s_1 \\ s_1 & 0 & -c_1 & a_1s_1-d_1c_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_{2}^{1}=\begin{bmatrix} c_2 & -s_2 & 0 & a_2c_2 \\ s_2 & c_2 & 0 & a_2s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad A_{3}^{2}=\begin{bmatrix} c_3 & -s_3 & 0 & a_3c_3 \\ s_3 & c_3 & 0 & a_3s_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{(A.1)}$$

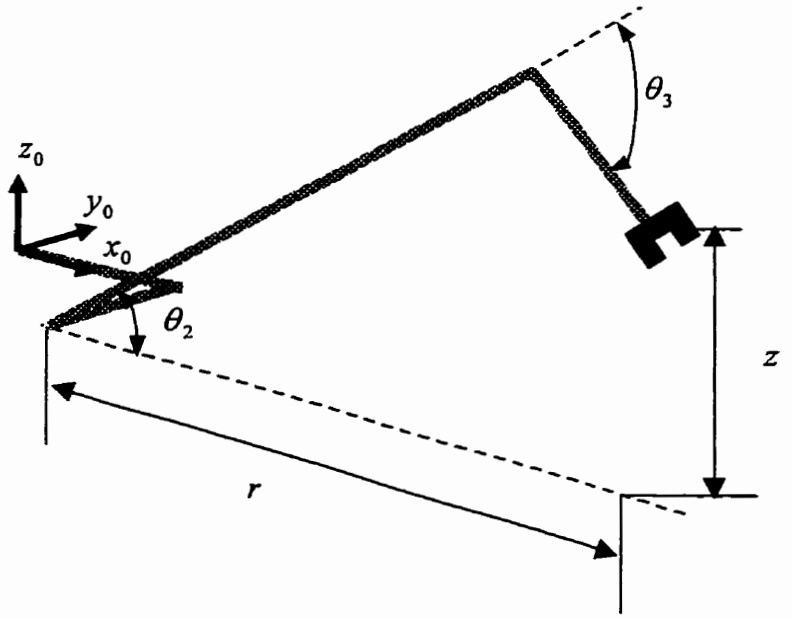
The position of the end-effector, in Cartesian coordinate is the last column of matrix resulting from multiplying matrices A_1, A_2, A_3:

$$\begin{aligned} x &= \cos(\theta_1)\left[a_1 + a_2\cos(\theta_2) + a_3\cos(\theta_2+\theta_3)\right] + d_1\sin(\theta_1) \\ y &= \sin(\theta_1)\left[a_1 + a_2\cos(\theta_2) + a_3\cos(\theta_2+\theta_3)\right] - d_1\cos(\theta_1) \\ z &= a_2\sin(\theta_2) + a_3\sin(\theta_2+\theta_3) \end{aligned} \qquad \text{(A.2)}$$

A.2 Inverse kinematics

Problem: *Given the position of the end-effector in Cartesian coordinates (x,y,z), find the joint angles (θ_1, θ_2, θ_3).*

The joint angles are calculated through geometric relations as below:



$$\theta_1 = \tan\frac{(r+a_1)y + d_1 x}{(r+a_1)x - d_1 y}$$

$$\theta_2 = \tan^{-1}\left(\frac{z}{r}\right) + \cos^{-1}\left[\frac{a_2^2 + l^2 - a_3^2}{2a_2 l}\right]$$

$$\theta_3 = -\pi + \cos^{-1}\left[\frac{a_2^2 + a_3^2 - l^2}{2a_2 a_3}\right] \qquad \text{(A.6)}$$

where :

$$r = \sqrt{x^2 + y^2 - d_1^2} - a_1$$

$$l = \sqrt{r^2 + z^2}$$

One should note that, due to the joint angle constraints, "elbow up" is the only possible configuration. Therefore, the problem of inverse kinematics has a unique solution.

B. Spline Interpolation

The results in this appendix are derived following the method of Ranky and Ho (1985).

A sequence of knot points is given for a single joint as $\theta_1, \theta_2, ..., \theta_n$, corresponding to the instants $t_1, t_2, ..., t_n$, respectively. The planning of the entire joint trajectory can be divided into three parts: (1) the start segment which connects θ_1 and θ_2; (2) the intermediate segments which connect $\theta_2, \theta_3, ..., \theta_{n-1}$ together and (3) the final segment which connects θ_{n-1} and θ_n.

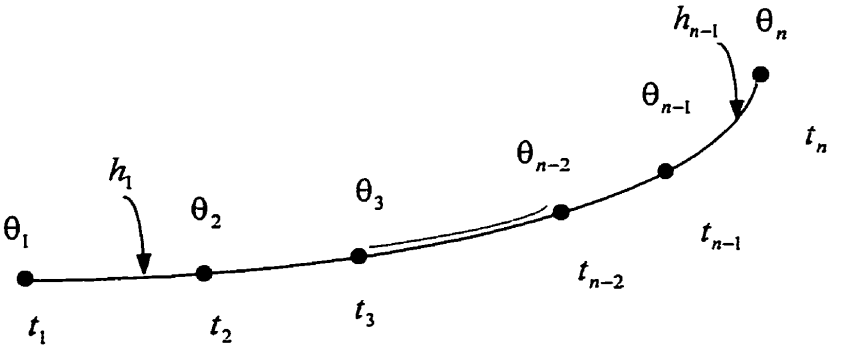


Fig. B.1: n-points trajectory.

The intermediate segments

The equation of the spline segment for two intermediate knot points θ_k and θ_{k+1} ($2<k<n-2$) of the n-points trajectory consisting of $n-1$ spline segments can be written for a single joint as:

$$\Theta(t) = B_4 t^3 + B_3 t^2 + B_2 t + B_1 \qquad \text{(B.1)}$$

Where $\Theta(t)$ represents the position of the joint as a function of parametric variable t. Letting t for this segment run from $0 \rightarrow h_k$, where $h_k = t_{k+1} - t_k$, the boundary conditions can be expressed as:

$$\begin{aligned} \Theta(0) &= \theta_k & \Theta(h_k) &= \theta_{k+1} \\ \Theta'(0) &= \theta'_k & \Theta'(h_k) &= \theta'_{k+1} \end{aligned} \tag{B.2}$$

Where θ'_k and θ'_{k+1} represent the velocities of the joint at the points θ_k and θ_{k+1} respectively. Substituting Eq. (B.2) into Eq. (B.1) the coefficients for the intermediate segments can be expressed:

$$\begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \frac{-3}{h^2_k} & \frac{3}{h^2_k} & \frac{-2}{h_k} & \frac{-1}{h_k} \\ \frac{2}{h^3_k} & \frac{-2}{h^3_k} & \frac{1}{h^2_k} & \frac{1}{h^2_k} \end{bmatrix} \begin{bmatrix} \theta_k \\ \theta_{k+1} \\ \theta'_k \\ \theta'_{k+1} \end{bmatrix} \tag{B.3}$$

In order to calculate the spline coefficients, one must first know the values of h_k, θ'_k and θ'_{k+1}. The parametric intervals h_k can be determined using (3.48); the velocities θ'_k and θ'_{k+1} can be determined using the continuity constraint on the acceleration at the knot points. Given θ_k, θ_{k+1} and θ_{k+2} $(2 \le k \le n\text{-}3)$ with two spline segments connecting them with parameters $0 \le t \le h_k$ and $0 \le t \le h_{k+1}$, the acceleration at the end of the first segment is:

$$\Theta^{*}(h_k) = 2B_3 + 6B_4 h_k = \frac{6}{h_k}\left[\frac{2(\theta_k - \theta_{k+1})}{h_k} + \theta'_k + \theta'_{k+1}\right] + \frac{2}{h_k}\left[\frac{3(\theta_{k+1} - \theta_k)}{h_k} - 2\theta'_k - \theta'_{k+1}\right] \quad \text{(B.4)}$$

The acceleration at the beginning of the second segment is:

$$\Theta^{*}(0) = 2B_3 = \frac{2}{h_{k+1}}\left[\frac{3(\theta_{k+2} - \theta_{k+1})}{h_{k+1}} - 2\theta'_{k+1} - \theta'_{k+2}\right] \quad \text{(B.5)}$$

The continuity constraint on acceleration requires Eqs. (B.4) and (B.5) to be equal. Equating them and rearranging terms gives:

$$h_{k+1}\theta'_k + 2(h_{k+1} + h_k)\theta'_{k+1} + h_k\theta'_{k+2} = \frac{3}{h_k h_{k+1}}\left[h_k^2(\theta_{k+2} - \theta_{k+1}) + h_{k+1}^2(\theta_{k+1} - \theta_k)\right] \quad \text{(B.6)}$$

Expressing Eq. (B.6) in matrix form for $2 \le k \le n\text{-}3$ yields:

$$\begin{bmatrix} h_3 & 2(h_2+h_3) & h_2 & 0 & 0 \\ 0 & h_4 & 2(h_3+h_4) & h_3 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & h_{n-2}\cdot & 2(h_{n-3}+h_{n-2}) & h_{n-3} \end{bmatrix} \begin{bmatrix} \theta'_2 \\ \theta'_3 \\ \cdot \\ \cdot \\ \cdot \\ \theta'_{n-1} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{3}{h_2 h_3}[h_2^2(\theta_4-\theta_3)+h_3^2(\theta_3-\theta_2)] \\ \frac{3}{h_3 h_4}[h_3^2(\theta_5-\theta_4)+h_4^2(\theta_4-\theta_3)] \\ \cdot \\ \cdot \\ \cdot \\ \frac{3}{h_{n-3}h_{n-2}}[h_{n-3}^2(\theta_{n-1}-\theta_{n-2})+h_{n-2}^2(\theta_{n-2}-\theta_{n-3})] \end{bmatrix} \tag{B.7}$$

Or symbolically, $[m][\theta'] = [a]$, where $[m]$ is an $(n\text{-}4)\times(n\text{-}2)$ matrix, $[\theta']$ is an $(n\text{-}2)\times 1$ vector, and $[a]$ is an $(n\text{-}4)\times 1$ vector.

The first and last segments

For the first and last segments, the additional constraints $\theta'_1 = \theta''_1 = \theta'_n = \theta''_n = 0$ require a fourth-order spline segment of the following form:

$$\Theta(t) = B_5 t^4 + B_4 t^3 + B_3 t^2 + B_2 t + B_1 \tag{B.8}$$

For the first segment, letting $0 \le t \le h_1$ the boundary conditions are expressed:

$$\begin{aligned} &\Theta(0) = \theta_1 && \Theta(h_1) = \theta_2 \\ &\Theta'(0) = 0 && \Theta'(h_1) = \theta'_2 \\ &\Theta''(0) = 0 \end{aligned} \tag{B.9}$$

Substituting (B.9) into (B.8), the spline coefficients for the first segment are expressed:

$B_1=\theta_1 \qquad B_2=0 \qquad B_3=0$

$$B_4 = \frac{4}{h_1^3}(\theta_2 - \theta_1) - \frac{1}{h_1^2}\theta'_2$$

$$B_5 = \frac{3}{h_1^4}(\theta_1 - \theta_2) + \frac{1}{h_1^3}\theta'_2$$

For the last segment, let $0 \le t \le h_{n-1}$ and have the following boundary conditions:

$$\begin{aligned} &\Theta(0) = \theta_{n-1} && \Theta'(h_{n-1}) = 0 \\ &\Theta'(0) = \theta'_{n-1} && \Theta''(h_{n-1}) = 0 \\ &\Theta(h_{n-1}) = \theta_n \end{aligned} \qquad \text{(B.10)}$$

Substituting (B.10) into (B.8), the spline coefficients for the last segment are expressed:

$B_1=\theta_{n-1}$

$B_2=\theta'_{n-1}$

$$B_3 = \frac{1}{h_{n-1}^2}(6\theta_n - 6\theta_{n-1} - 3\theta'_{n-1} h_{n-1})$$

$$B_4 = \frac{1}{h_{n-1}^3}(-8\theta_n + 8\theta_{n-1} + 3\theta'_{n-1} h_{n-1})$$

$$B_5 = \frac{1}{h_{n-1}^4}(3\theta_n - 3\theta_{n-1} - \theta'_{n-1} h_{n-1})$$

The equation (B.7) for finding $[\theta']$ must be redefine since 4th order segments are included at the beginning and end. Finding the acceleration at the end of the first segment yields:

$$\theta''(h_1) = 6B_4 h_1 + 12B_5 h_1^2 = \frac{6}{h_1^2}[4(\theta_2 - \theta_1) - h_1\theta'_2] + \frac{12}{h_1^2}[3(\theta_1 - \theta_2) + h_1\theta'_2] \quad \text{(B.11)}$$

Equating Eq. (B.11) with Eq. (B.5) yields:

$$\left(\frac{2}{h_2}+\frac{3}{h_1}\right)\theta'_2+\frac{1}{h_2}\theta'_3=\frac{3}{h_2^2}(\theta_3-\theta_2)+\frac{6}{h_1^2}(\theta_2-\theta_1) \tag{B.12}$$

Performing the same operations for the beginning of the last segment and equating it with the acceleration found for the end of the previous intermediate segment yields:

$$\left(\frac{2}{h_{n-2}}+\frac{3}{h_{n-1}}\right)\theta'_{n-1}+\frac{1}{h_{n-2}}\theta'_{n-2}=\frac{3}{h_{n-2}^2}(\theta_{n-1}-\theta_{n-2})+\frac{6}{h_{n-1}^2}(\theta_n-\theta_{n-1}) \tag{B.13}$$

From Eqs. (B.12), (B.13) and (B.7) the matrix equation for solving for n-2 unknown velocities can be expressed as [M][θ']=[A], where [M] is $(n\text{-}2)\times(n\text{-}2)$ matrix, [θ'] is $(n\text{-}2)\times 1$ vector, and [A] is $(n\text{-}2)\times 1$ vector in the form:

$$\begin{bmatrix} M_{22} & M_{23} & 0 & \cdots & & 0 \\ h_3 & 2(h_2+h_3) & h_2 & 0 & \cdots & \vdots \\ 0 & & & & & \\ \vdots & & & & & 0 \\ \vdots & & & h_{n-2} & 2(h_{n-3}+h_{n-2}) & h_{n-3} \\ 0 & \cdots & & 0 & M_{n-1,n-2} & M_{n-1,n-1} \end{bmatrix} \begin{bmatrix} \theta'_2 \\ \theta'_3 \\ \vdots \\ \\ \theta'_{n-1} \end{bmatrix}$$

$$= \begin{bmatrix} a_2 \\ \frac{3}{h_2h_3}[h_2^2(\theta_4-\theta_3)+h_3^2(\theta_3-\theta_2)] \\ \vdots \\ \frac{3}{h_{n-3}h_{n-2}}[h_{n-3}^2(\theta_{n-1}-\theta_{n-2})+h_{n-2}^2(\theta_{n-2}-\theta_{n-3})] \\ a_{n-1} \end{bmatrix} \tag{B.14}$$

where :

$$M_{22} = \frac{2}{h_2} + \frac{3}{h_1} \qquad M_{23} = \frac{1}{h_2}$$

$$M_{n-1,n-2} = \frac{1}{h_{n-2}} \qquad M_{n-1,n-1} = \frac{2}{h_{n-2}} + \frac{3}{h_{n-1}}$$

$$a_2 = \frac{3}{h_2^2}(\theta_3 - \theta_2) + \frac{6}{h_1^2}(\theta_2 - \theta_1)$$

$$a_{n-1} = \frac{3}{h_{n-2}^2}(\theta_{n-1} - \theta_{n-2}) + \frac{6}{h_{n-1}^2}(\theta_n - \theta_{n-1})$$

Solving Eq. (B.14) for $[\theta']$ allows us to find spline functions for a single joint. The same procedure is applied for each joint, using the same values of h_k. Thus a path through n points for a robot with N joints will consists of $(n\text{-}1)\times N$ unique spline functions.

C. Application of Downhill Simplex Method

The following example serves to illustrate the downhill simplex method operations, namely, reflection, expansion and contraction. This example has been adopted from the reference by Rao, (1984). Note that the stopping criteria used in this thesis is based on Matlab M-file: Fmins.m and it is different from the one suggested by Rao. The stopping criteria used in this thesis is formulated as: (1) the vector distance moved in the terminating step must be smaller than a tolerance ε_1, and (2) the decrease in the function value in the terminating step must be smaller than a tolerance ε_2. The stopping criteria in the example shown here is: the standard deviation of the objective function at the $n+1$ vertices of the current simplex must be smaller than a tolerance ε.

Problem Statement:

Minimize $F(x_1, x_2) = x_1 - x_2 + 2x_1^2 + 2x_1x_2 + x_2^2$. The points (vertices) defining the initial simplex are taken as:

$$V_1 = \begin{Bmatrix} 4.0 \\ 4.0 \end{Bmatrix}, V_2 = \begin{Bmatrix} 5.0 \\ 4.0 \end{Bmatrix} \text{ and } V_3 = \begin{Bmatrix} 4.0 \\ 5.0 \end{Bmatrix},$$

and $\alpha = 1.0$, $\beta = 0.5$, $\gamma = 2.0$. For convergence, take the value of ε as 0.2.

Solution

Iteration 1

1. The function value at each vertices of the current simplex is given by:

$$F_1 = F(V_1) = 4.0 - 4.0 + 2(16.0) + 2(16.0) + 16.0 = 80.0$$

$$F_2 = F(V_2) = 5.0 - 4.0 + 2(25.0) + 2(20.0) + 16.0 = 107.0$$

$$F_3 = F(\boldsymbol{V}_3) = 4.0 - 5.0 + 2(16.0) + 2(20.0) + 25.0 = 96.0$$

$$\therefore \boldsymbol{W} = \boldsymbol{V}_3 = \begin{Bmatrix} 5.0 \\ 4.0 \end{Bmatrix},\ F(\boldsymbol{W}) = 107.0,\ \boldsymbol{B} = \boldsymbol{V}_1 = \begin{Bmatrix} 4.0 \\ 4.0 \end{Bmatrix} \text{ and } F(\boldsymbol{B}) = 80.0.$$

2. The centroid $\overline{\boldsymbol{C}}$ is obtained as

$$\overline{\boldsymbol{C}} = \frac{1}{2}(\boldsymbol{V}_1 + \boldsymbol{V}_2) = \begin{Bmatrix} 4.0 \\ 4.5 \end{Bmatrix} \text{ with } F(\overline{\boldsymbol{C}}) = 87.75$$

3. The reflection point is found as

$$\boldsymbol{R} = 2\overline{\boldsymbol{C}} - \boldsymbol{W} = \begin{Bmatrix} 3.0 \\ 5.0 \end{Bmatrix} \text{ and } F(\boldsymbol{R}) = 71.0$$

4. As $F(\boldsymbol{R}) < F(\boldsymbol{B})$, $\boldsymbol{E}$ is as

$$\boldsymbol{E} = 2\boldsymbol{R} - \overline{\boldsymbol{C}} = \begin{Bmatrix} 2.0 \\ 5.5 \end{Bmatrix} \text{ and } F(\boldsymbol{E}) = 56.75$$

5. Since $F(\boldsymbol{E}) < F(\boldsymbol{B})$, $\boldsymbol{W}$ is replaced by $\boldsymbol{E}$ and the vertices of the new simplex are obtained as

$$\boldsymbol{V}_1 = \begin{Bmatrix} 4.0 \\ 4.0 \end{Bmatrix},\ \boldsymbol{V}_2 = \begin{Bmatrix} 2.0 \\ 5.5 \end{Bmatrix} \text{ and } \boldsymbol{V}_3 = \begin{Bmatrix} 4.0 \\ 5.0 \end{Bmatrix}$$

6. To test for convergence, the standard deviation of the function at the $n+1$ vertices of the current simplex is computed as

$$\boldsymbol{Q} = \left\{ \frac{(80.0 - 87.75)^2 + (56.75 - 87.75)^2 + (96.0 - 87.75)^2}{3} \right\}^{1/2} = 19.6$$

As this quantity is not smaller than ε, we go to the next iteration.

The next two iterations are similar to the Iteration 1.

Iteration 4

1. The vertices of the current simplex are

$$\boldsymbol{V}_1 = \begin{Bmatrix} -3.5 \\ 6.625 \end{Bmatrix}, \boldsymbol{V}_2 = \begin{Bmatrix} 2.0 \\ 5.5 \end{Bmatrix} \text{ and } \boldsymbol{V}_3 = \begin{Bmatrix} 1.0 \\ 4.25 \end{Bmatrix},$$

$$F(\boldsymbol{V}_1) = 11.89, \qquad F(\boldsymbol{V}_2) = 56.75, \qquad F(\boldsymbol{V}_3) = 25.3125,$$

$$\therefore \boldsymbol{W} = \boldsymbol{V}_2 = \begin{Bmatrix} 2.0 \\ 5.5 \end{Bmatrix} \text{ and } \boldsymbol{B} = \boldsymbol{V}_1 = \begin{Bmatrix} -3.5 \\ 6.625 \end{Bmatrix}$$

2. The centroid $\overline{\boldsymbol{C}}$ is obtained as

$$\overline{\boldsymbol{C}} = \frac{1}{2}(\boldsymbol{V}_1 + \boldsymbol{V}_2) = \begin{Bmatrix} -1.25 \\ 5.4375 \end{Bmatrix} \text{ with } F(\overline{\boldsymbol{C}}) = 12.4037.$$

3. The reflection point is found as

$$\boldsymbol{R} = 2\overline{\boldsymbol{C}} - \boldsymbol{W} = \begin{Bmatrix} -4.5 \\ 5.375 \end{Bmatrix} \text{ and } F(\boldsymbol{R}) = 11.14.$$

4. As $F(\boldsymbol{R}) < F(\boldsymbol{B})$, $\boldsymbol{E}$ is found as

$$\boldsymbol{E} = 2\boldsymbol{R} - \overline{\boldsymbol{C}} = \begin{Bmatrix} -7.75 \\ 5.3125 \end{Bmatrix} \text{ and } F(\boldsymbol{E}) = 52.9525$$

5. Since $F(\boldsymbol{E}) > F(\boldsymbol{B})$, the new vertices are obtained as

$$\boldsymbol{V}_1 = \begin{Bmatrix} -3.5 \\ 6.625 \end{Bmatrix}, \boldsymbol{V}_2 = \begin{Bmatrix} -4.5 \\ 5.375 \end{Bmatrix} \text{ and } \boldsymbol{V}_3 = \begin{Bmatrix} 1.0 \\ 4.22 \end{Bmatrix}$$

6. For testing convergence, $\boldsymbol{Q}$ is computed as

$$\boldsymbol{Q} = \left\{ \frac{(11.89 - 12.40375)^2 + (11.14 - 12.40375)^2 + (25.3125 - 12.40375)^2}{3} \right\}^{1/2} = 7.5$$

As $\boldsymbol{Q} > \varepsilon$, we go to the next iteration.

Iteration 5

1. As

$$\boldsymbol{V}_1 = \begin{Bmatrix} -3.5 \\ 6.625 \end{Bmatrix}, \boldsymbol{V}_2 = \begin{Bmatrix} -4.5 \\ 5.375 \end{Bmatrix} \text{ and } \boldsymbol{V}_3 = \begin{Bmatrix} 1.0 \\ 4.22 \end{Bmatrix}$$

$$F(\boldsymbol{V}_1) = 11.89, \qquad F(\boldsymbol{V}_2) = 11.14, \qquad F(\boldsymbol{V}_3) = 25.3125,$$

$$\therefore \boldsymbol{W} = \boldsymbol{V}_3 = \begin{Bmatrix} 1.0 \\ 4.25 \end{Bmatrix} \text{ and } \boldsymbol{B} = \boldsymbol{V}_1 = \begin{Bmatrix} -4.5 \\ 5.375 \end{Bmatrix}$$

2. The centroid $\overline{\boldsymbol{C}}$ is obtained as

$$\overline{\boldsymbol{C}} = \frac{1}{2}(\boldsymbol{V}_1 + \boldsymbol{V}_2) = \begin{Bmatrix} -4.0 \\ 6.0 \end{Bmatrix} \text{with } F(\overline{\boldsymbol{C}}) = 10.0..$$

3. The reflection point is found as

$$\boldsymbol{R} = 2\overline{\boldsymbol{C}} - \boldsymbol{W} = \begin{Bmatrix} \text{-9.0} \\ 7.75 \end{Bmatrix} \text{ and } F(\boldsymbol{R}) = 65.8125.$$

4. As $F(\boldsymbol{R}) > F(\boldsymbol{B})$ and $F(\boldsymbol{R}) > F(\boldsymbol{W})$, $\boldsymbol{C}$ is found by contraction as

$$\boldsymbol{C} = 0.5\boldsymbol{W} + 0.5\overline{\boldsymbol{C}} = \begin{Bmatrix} -1.5 \\ 5.125 \end{Bmatrix} \text{ and } F(\boldsymbol{C}) = 8.75$$

5. As $F(\boldsymbol{C}) > F(\boldsymbol{W})$, the vertices of the new simplex are found as

$$\boldsymbol{V}_1 = \begin{Bmatrix} -3.5 \\ 6.625 \end{Bmatrix}, \boldsymbol{V}_2 = \begin{Bmatrix} -4.5 \\ 5.375 \end{Bmatrix} \text{ and } \boldsymbol{V}_3 = \begin{Bmatrix} -1.5 \\ 5.125 \end{Bmatrix}$$

6. For testing convergence, $\boldsymbol{Q}$ is computed as

$$\boldsymbol{Q} = \left\{ \frac{(11.89 - 10.0)^2 + (11.14 - 10.0)^2 + (8.75 - 10.0)^2}{3} \right\}^{1/2} = 1.466$$

As $\boldsymbol{Q} > \varepsilon$, we go to the next iteration.

This procedure can be continued until the specified convergence is satisfied.