SIMULATION AND ADAPTIVE CONTROL OF A ROBOT ARM

by

PAULINE ANNE MARCHAND

B.Eng., McGill University, 1980

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

Department of Mechanical Engineering

We accept this thesis as conforming

to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

June, 1985

© PAULINE ANNE MARCHAND, 1985
In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.
ABSTRACT

The equations of motion describing a robot's dynamics are coupled and nonlinear, making the design of an optimum controller difficult using classical techniques. In this work an explicit adaptive control law is proposed based on a discrete linear model for each link and on the minimization of a quadratic performance criterion. The system parameters are recursively estimated at each control step using least squares. A computer simulation of the resulting scheme is performed to evaluate the controller. The simulation model is based on the first three links of an existing robot, includes motor dynamics and treats the wrist assembly as a load mass. Simulated test paths requiring movement of the outer two links indicate that the controller adapts and that its behaviour is stable and convergent.
**TABLE OF CONTENTS**

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ABSTRACT</td>
<td>iii</td>
</tr>
<tr>
<td></td>
<td>TABLE OF CONTENTS</td>
<td>v</td>
</tr>
<tr>
<td></td>
<td>LIST OF TABLES</td>
<td>vi</td>
</tr>
<tr>
<td></td>
<td>LIST OF FIGURES</td>
<td>vii</td>
</tr>
<tr>
<td></td>
<td>NOMENCLATURE</td>
<td>xi</td>
</tr>
<tr>
<td></td>
<td>ACKNOWLEDGEMENTS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CHAPTER</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>INTRODUCTION</td>
<td></td>
</tr>
<tr>
<td>1.1</td>
<td>Preliminary Remarks</td>
<td>1</td>
</tr>
<tr>
<td>1.2</td>
<td>Literature Review</td>
<td>3</td>
</tr>
<tr>
<td>1.3</td>
<td>Scope and Objectives</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>ADAPTIVE CONTROL STRATEGY</td>
<td></td>
</tr>
<tr>
<td>2.1</td>
<td>Preliminary Remarks</td>
<td>14</td>
</tr>
<tr>
<td>2.2</td>
<td>Design of the Controller</td>
<td>15</td>
</tr>
<tr>
<td>2.2.1</td>
<td>Admissible Controls</td>
<td>16</td>
</tr>
<tr>
<td>2.2.2</td>
<td>Discrete Model Structure</td>
<td>16</td>
</tr>
<tr>
<td>2.2.3</td>
<td>Control Law</td>
<td>21</td>
</tr>
<tr>
<td>2.2.4</td>
<td>Identification Procedure</td>
<td>26</td>
</tr>
<tr>
<td>2.2.5</td>
<td>Replacement of True Parameters</td>
<td>29</td>
</tr>
<tr>
<td>3</td>
<td>THE SIMULATION MODEL</td>
<td></td>
</tr>
<tr>
<td>3.1</td>
<td>The RSI Robot</td>
<td>36</td>
</tr>
<tr>
<td>3.2</td>
<td>Kinematics</td>
<td>37</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Direct Kinematics</td>
<td>38</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Inverse Kinematics</td>
<td>42</td>
</tr>
<tr>
<td>3.2.3</td>
<td>Trajectory Planning</td>
<td>45</td>
</tr>
<tr>
<td>3.3</td>
<td>Dynamics</td>
<td>46</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Actuators</td>
<td>46</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Manipulator Dynamics</td>
<td>47</td>
</tr>
<tr>
<td>3.3.3</td>
<td>Simulation Equations</td>
<td>54</td>
</tr>
<tr>
<td>3.4</td>
<td>Summary</td>
<td>56</td>
</tr>
<tr>
<td>4</td>
<td>EVALUATION OF THE CONTROL STRATEGY</td>
<td></td>
</tr>
<tr>
<td>4.1</td>
<td>One Degree-of-Freedom Case</td>
<td>68</td>
</tr>
<tr>
<td>4.1.1</td>
<td>Results and Discussion</td>
<td>70</td>
</tr>
<tr>
<td>4.2</td>
<td>RSI Robot Simulation</td>
<td>73</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Results and Discussion</td>
<td>75</td>
</tr>
</tbody>
</table>
TABLE OF CONTENTS (Continued)

5 CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 Conclusions</td>
<td>140</td>
</tr>
<tr>
<td>5.2 Recommendations for Future Work</td>
<td>141</td>
</tr>
</tbody>
</table>

BIBLIOGRAPHY ......................................................... 144

APPENDIX

<table>
<thead>
<tr>
<th>Appendix</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>RECURSIVE LEAST SQUARES</td>
<td>148</td>
</tr>
<tr>
<td>II</td>
<td>THE A AND T MATRICES</td>
<td>155</td>
</tr>
<tr>
<td>III</td>
<td>THE PSEUDO INERTIA MATRIX ( J_4 )</td>
<td>157</td>
</tr>
<tr>
<td>IV</td>
<td>DERIVATION OF THE LOAD TERM ( L_4 )</td>
<td>159</td>
</tr>
<tr>
<td>V</td>
<td>( J_4 ) FOR THE RSI ROBOT</td>
<td>162</td>
</tr>
<tr>
<td>VI</td>
<td>SETTING UP THE INTEGRATION ROUTINE</td>
<td>164</td>
</tr>
</tbody>
</table>
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Parameter Values of the RSI Robot</td>
<td>32</td>
</tr>
<tr>
<td>II</td>
<td>Kinematic Parameters of the RSI Robot</td>
<td>33</td>
</tr>
<tr>
<td>III</td>
<td>Discrete Model Parameters and Coefficients for the Known Case</td>
<td>83</td>
</tr>
<tr>
<td>IV</td>
<td>Test Paths for the Single-Link Simulation</td>
<td>84</td>
</tr>
<tr>
<td>V</td>
<td>Run Parameters for the Single-Link Simulation</td>
<td>85</td>
</tr>
<tr>
<td>VI</td>
<td>Circle Trajectory - Parameters of the Simulation Runs</td>
<td>86</td>
</tr>
<tr>
<td>VII</td>
<td>Line at 45° - Parameters of the Simulation Runs</td>
<td>87</td>
</tr>
<tr>
<td>VIII</td>
<td>Horizontal Line - Parameters of the Simulation Runs</td>
<td>88</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Trajectory for a two-link arm</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>Schematic of robot control system</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>(a) joint control</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(b) endpoint control</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Robot coordinate frames</td>
<td>12</td>
</tr>
<tr>
<td>4</td>
<td>Schematic of the simulated system</td>
<td>13</td>
</tr>
<tr>
<td>5</td>
<td>Schematic of a MRAS</td>
<td>34</td>
</tr>
<tr>
<td>6</td>
<td>Schematic of a STR</td>
<td>35</td>
</tr>
<tr>
<td>7</td>
<td>The RSI robot</td>
<td>58</td>
</tr>
<tr>
<td>8</td>
<td>Position of a point relative to two frames</td>
<td>59</td>
</tr>
<tr>
<td>9</td>
<td>Link coordinate frames for the RSI robot</td>
<td>60</td>
</tr>
<tr>
<td>10</td>
<td>Vector definitions for the inverse kinematics</td>
<td>61</td>
</tr>
<tr>
<td>11</td>
<td>Workspace of links 2 and 3 of the RSI robot</td>
<td>62</td>
</tr>
<tr>
<td>12</td>
<td>Trapezoidal velocity law</td>
<td>63</td>
</tr>
<tr>
<td>13</td>
<td>Direction cosines for endpoint trajectory</td>
<td>64</td>
</tr>
<tr>
<td>14</td>
<td>Schematic of motor, drive train, and link</td>
<td>65</td>
</tr>
<tr>
<td>15</td>
<td>Schematic of single-degree-of-freedom case</td>
<td>89</td>
</tr>
<tr>
<td>16</td>
<td>Program flowchart - known case, one degree of freedom</td>
<td>90</td>
</tr>
<tr>
<td>17</td>
<td>Program flowchart - Adaptive case, one degree of freedom</td>
<td>91</td>
</tr>
<tr>
<td>18</td>
<td>Known control - effect of varying $\epsilon$, $T = 0.010$ s</td>
<td>92</td>
</tr>
<tr>
<td>19</td>
<td>Known control - effect of varying $\epsilon$, $T = 0.020$ s</td>
<td>94</td>
</tr>
<tr>
<td>Page</td>
<td>Title</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-------</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Effect of disturbance</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Feedforward in a proportional controller</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Adaptive control - no disturbance</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>Known vs. adaptive control, step inputs</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Known vs. adaptive control, ramp input</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>Known vs. adaptive control, trapezoid</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>Effect of zero initial estimates</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>Program flowchart - Adaptive case, RSI robot</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>Configuration for the Circle Path</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>Circle simulations - effect of loads</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>Circle simulations loads - effect of λ</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>Circle simulations - effect of initial estimates</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>Configuration for the 45° line</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>45° line simulations</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>Configuration for the horizontal line</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>Horizontal line simulations</td>
<td></td>
</tr>
</tbody>
</table>
NOMENCLATURE

Subscripts
i refers to the $i^{th}$ joint
d desired, disturbance
m motor
act actuator
l load

a, b continuous transfer function variables

$a_1, a_2, b_1, b_2$ discrete model parameters
$c_1, c_2, c_3, c_4, c_5$ control coefficients

g gravity
m mass
n gear ratio, order of system
$x_2$ location of centre of mass, link 2
$x_3$ location of centre of mass, link 3
$z_1$ location of mass, load

u control input
y system output
$y_d$ reference input
k sampling instant
q shift operator
t time
v velocity
a acceleration
T  sampling period
K  control input constant
K_T  motor constant
K_e  back emf constant
K  kinetic energy
L  Lagrangian
P  potential energy
R  armature resistance
I  performance criterion
D  coefficient in manipulator equations of motion
J  inertia
C  damping constant
W  weighting on the control

\(\lambda\)  forgetting factor
\(\varepsilon\)  pseudo-gain
\(\alpha\)  direction cosine
\(\beta\)  direction cosine
\(\theta\)  joint angle
\(\tau\)  torque

\(E[]\)  expected value
\(G(s)\)  continuous time transfer function
\(G(z)\)  discrete time transfer function
\(e(k)\)  white noise
\(v(k)\)  disturbance sequence
Vectors and matrices are in **bold** type

- \( \mathbf{x} \) cartesian position
- \( \mathbf{p}_k^{(k)} \) desired endpoint position
- \( \mathbf{r}_i \) position vector \( \mathbf{r} \) with respect to \( i^{\text{th}} \) coordinate frame
- \( \mathbf{u} \) control input
- \( \mathbf{y} \) system output (angular position)
- \( \mathbf{y}_d \) reference input (angular position)
- \( \mathbf{g} \) gravity
- \( \mathbf{J} \) pseudo-inertia matrix
- \( \mathbf{T} \) transformation matrix
- \( \mathbf{A} \) transformation matrix
- \( \mathbf{Y} \) output measurement vector
- \( \mathbf{\rho} \) centre of mass
- \( \mathbf{\theta} \) parameter vector
- \( \mathbf{\phi} \) measurement vector
- \( \mathbf{\Phi} \) information matrix

- **ARMA** autoregressive moving average
- **STR** self-tuning regulator
- **MRAS** model reference adaptive control
ACKNOWLEDGEMENTS

The author would like to express thanks to Mr. Alan Steeves for assistance with the VAX computer and to Ms. Jocelyn Hung for help with the plotting.

The guidance and encouragement from her supervisors, Professor Peter D. Lawrence and Professor Dale B. Cherchas, were greatly appreciated.

This research was supported by the Department of Electrical Engineering and the Department of Mechanical Engineering.
1.1 Preliminary Remarks

Present-day robots have limited capabilities. They are generally used for executing repetitive tasks. They have high weight-to-payload ratios and require powerful actuators. Because of the variable structure of the robot the controllers which regulate the joint movements are designed for the worst case. Thus, less than optimum performance is achieved at the other operating conditions. In this work an adaptive controller which can adjust to changes in the system dynamics is proposed, and its behaviour is studied by computer simulation.

An example of a change in the system dynamics as the robot moves is easily seen by considering Fig. 1 which illustrates a simple two-link arm, initially in position A, whose tip traces a straight-line path until position B is reached. The total inertia felt by the motor driving the first link will change because as the second link moves, the distance between its centre of mass and the joint of the first link changes.

The disturbances acting on each link also vary. Gravity loads depend on position. Velocity effects such as centripetal forces are present and are variable as well. Robots have been designed for slow speeds so that these disturbances are relatively small.

A schematic of a joint control system is shown in Fig. 2(a). Assuming that the desired joint angular position as a function of time, \( \theta_d(t) \) is known, the idea is to compare it to the measured joint position \( \theta(t) \) and form an error signal. Using knowledge of the system dynamics, a control law operating on the error is designed so that certain performance speci-
fications are met and disturbances, \( \tau_d \), are rejected satisfactorily. When the system dynamics change, the control law is no longer optimum. It would be advantageous if the coefficients of the controller could be varied on-line as the system parameters and the disturbances changed. These considerations have led to the study of more complex control laws, and motivate the present work.

A second motivation is that technological advances in sensor development and information processing are beginning to make more sophisticated control strategies feasible. Fig. 1 may be examined again. In order that the endpoint of the arm follow the straight line path, it is necessary that the movements of both links be precisely coordinated. The joint controllers each individually tend to drive the error between the desired angle and the measured angle of their link to zero. In order to obtain good endpoint positional accuracy, not only must the joint controllers be adequately designed, but flexibility in the links and backlash in the gear train must be absent. Rigid links imply heavier designs and hence powerful motors. Payloads must be light relative to the construction of the robot to prevent deflection of the links and to avoid introducing extra gravity disturbances which would disrupt the controllers. If, however, it was possible to measure the position of the endpiece and use that measurement for feedback, the above design limitations could be eliminated. This approach is termed endpoint control and is schematically illustrated in Fig. 2(b). One difficulty is that the feedback measurement is coded in terms of a fixed base reference frame whereas the control action takes place relative to joint space. This idea is illustrated in Fig. 3. Point P is uniquely defined by the vector pair \((x,y)\). By fixing a coordinate frame on each link, such as \((x_1,y_1)\) on link 1, it is possible to find the
relationship between \((x,y)\) and \((\theta_1, \theta_2)\). The mapping \(\theta + x\), however, is not unique. Fig. 3 shows the "elbow up" solution. Endpoint control strategies must deal with these issues. The intent of this study was to formulate a control scheme which might be suitable for extension to endpoint control, and would function adaptively at the joint level.

1.2 Literature Review

Several researchers have tackled the two problems mentioned above. There are basically two approaches in the design of these new advanced control strategies. The first approach is to obtain a more sophisticated model of the system. Using that information the required control inputs may be anticipated. The second approach presumes much less knowledge of the system. Instead, input-output data from the system is used to infer its structure and to provide adequate control. A short description of some of the work done in these areas follows, as it applies to robotics.

Whitney [1] proposed resolved rate control to control joint velocities obtained from hand or endpiece positions and velocities. His is a classic paper. Luh, Walker and Paul [2] propose a resolved acceleration technique to close the loop around the hand of the robot by pre-calculating the required torques given desired angular position, velocity and accelerations. This method requires a detailed model of the manipulator. Freund [3] has suggested nonlinear compensations to decouple the equations of motion of the manipulator. Golla, Garg, and Hughes [4] proposed linear multivariable control for assigning the closed loop poles of the system by state feedback. Cvetkovic and Vukobratovic [5] examine several two-stage control schemes where the control signal consists of a nominal pre-programmed value and a compensating term. Vukobratovic and Stokic [6]
study a few control strategies which take dynamics into account in different degrees. Hogan and Cotter [7] have suggested Cartesian impedance control for closing the loop around the hand.

In general the above techniques require a detailed model of the robot. Modelling errors would lead to decreased effectiveness of the control laws. A second disadvantage of these methods is that they typically require large amounts of calculation between the sampling instants when a control signal is downloaded to the joint actuator. Other researchers have taken the view of using an adaptive controller, which pre-supposes much less knowledge of the system. Lee [8] proposes that a pre-calculated nominal torque be supplemented by inputs calculated from identification of the perturbed system about the nominal trajectory. Balestrino et al. [9] proposed a model reference approach and design the control law using hyperstability theory. Only the bounds on the parameters need to be known. Further study of the model reference technique is done by Nicosia and Tomei [10], who investigate convergence. Recently, Kim and Shin [11] have described a method of averaged dynamics. Young [12] proposed sliding mode control based on the theory of variable structures systems, in which a similar requirement of knowledge of only the bounds of the parameters is needed.

The present work is based on that of Koivo and Guo [13] and motivated by that of Lobbezoo [14]. Lobbezoo showed that for the case of a two-degree-of-freedom robot, good performance was achieved by estimating the transformation matrix $\mathbf{x} + \mathbf{\Theta}$ rather than using the analytical formulation. He used recursive least squares to identify the entries of the transformation matrix. Koivo and Guo proposed that the robot system be modelled by an autoregressive moving average (ARMA), and the parameters
recursively identified using the least squares technique. A control law based on a quadratic performance criterion is derived and the estimated parameters are substituted into this law at each control step. Koivo chose a system model structure based on empirical tests. In this work an analysis of the continuous-time system representation is done and it is shown why a successful discrete model has the structure given. The same control law design is followed, but the controlled variable was chosen as angular position rather than angular velocity, since Koivo's method necessitated an ad-hoc adjustment for position errors. Angular position was chosen also because, if the controller was successful, as an avenue for future investigation, it was structured so that the transformation from base coordinates to joint coordinates could be included in the adaptation.

1.3 Objectives and Scope of the Study

The motivation behind the present study was presented in Section 1.1. The general objective was to investigate and test via computer simulation a control law for improved joint control. Formulation of the criteria for the design is discussed next. The choice of a control strategy depends on many factors and constraints, since performance must be specified within a complex framework of interacting and sometimes conflicting goals.

For example, performance can critically depend on an efficient trajectory planner. Each robot joint has limits of acceleration and velocity; a sophisticated trajectory planner ensures that the commanded sequence of $\theta$ results in $\dot{\theta}$ and $\ddot{\theta}$ well within these bounds, and also ensures smooth transitions between path segments. When these conditions are met, less rigorous demands are placed on the joint actuators and their controllers. The relationship between trajectory planning and controller
requirements is not studied here, but it is one illustration that performance of the robot depends on a host of factors.

However, it is possible to establish a framework for the design and five broad criteria were specified for the design of the controller. These are (1) the formulation would be digital, (2) an adaptive technique would be used, (3) the structure would permit extension to endpoint control, (4) the simulation model would be realistic, and (5) the controller would be geared to large motions (as opposed to adjustment or assembly). These five points are now discussed.

The great majority of the newer approaches to robot control are, on the theoretical level, treated using continuous time techniques. This is probably due to the relative newness of digital control and hence lack of familiarity with it. The behaviour of the system under consideration - the robot - is of course described by continuous-time equations. Control laws are designed from system models, and many of the proposed laws are by extension derived in continuous-time form [2,3,4,9,10]. However, it is implicit that the implementation of these control laws must be done digitally, i.e., the calculation of the control signals is done on a microcomputer and at a prescribed sampling rate the signals are downloaded to a digital-to-analog converter and on to the actuators of the joints. Issues such as the effect of sampling time and computational delays are impossible to analyze given the present formulation of these control laws. It seems to make more sense to use digital theory in the first place for the design of the controller. This is the approach chosen in this work.

The second criteria is adaptation, and it refers to the class of control laws from which the strategy studied in this work is chosen. As is pointed out in the literature review, there are basically two philosophies
behind the advanced control laws which have been studied for robots. One of the approaches is to improve the system model used in the design of the controller. Because the equations describing the system are coupled and non-linear, large numbers of calculations are typically required in the control law. The goal is to compensate using exact knowledge of the system, which in practice is not achievable. The second approach is to use some form of adaptive control, which presupposes much less knowledge of the system, and which offers a possibility for compensating for changing system parameters. The survey paper by Astrom [15] is a good overview of adaptive control, covering many of the issues in the field and listing many references. In this work an adaptive control strategy similar to the one proposed by Koivo [13] is studied.

The third objective which the controller had to meet was that its structure be chosen with an eye to future developments, so that sensory information from the environment could be added to it. In particular, incorporating Lobbezoo's adaptive transformations to attempt endpoint control was considered a possibility.

Computer simulation is a useful vehicle of experimentation as the first step in evaluating a controller. The costs and pitfalls of hardware implementation are avoided at the initial design stage. Simulation is used in this work to test the controller. However, an existing robot is used as the prototype for building the simulation model, rather than a fictitious robot of arbitrary characteristics. In addition, the actuators are included in the system model.

The last criterion is that the design be made for "large" motions. The design and performance of a joint controller is influenced by the type of task performed by the robot. One can classify operating modes into
three groups:
   i) large motions,
   ii) fine motions,
   iii) assembly tasks.

Assembly tasks require that compliance and force feedback be considered, and are not studied here. In fine motions, simplifying assumptions of slow speeds and linearizations about the operating point may be made which change the focus of the analysis. This study is geared to large motions, which include a few types. The task objective can be tracking of a path, object tracking (for example conveyor belt following) or simply moving from point A to B. Here some point-to-point motion is considered as well as path tracking. These criteria are to be met within the broad objective of improved joint control.

It is evident that attempts to study a six-degree-of-freedom robot without having acquired expertise on lower orders of magnitude would result in the major trends being obscured by secondary effects. The scope of the work is limited to three-degrees-of-freedom. Dynamically speaking, as far as the first three links are concerned, in many cases a good approximation is to treat the wrist assembly as a point mass. This is the approach taken here: the six-degree-of-freedom system is split into two subsystems. The lower subsystem, that is the first three links, is analyzed. The simulation is carried out for two links since it was felt that the distinguishing features of the control law can be illustrated by that case. For the chosen robot the dynamic effects of the second and third links upon the first are minor compared to the coupling between the second and third links.

The system to be studied is illustrated in schematic form in Fig. 4.
Fig. 1  TRAJECTORY FOR A TWO-LINK ARM
Fig. 2 SCHEMATIC OF ROBOT CONTROL SYSTEMS
a) joint control
b) endpoint control
Fig. 3  ROBOT ARM COORDINATE SYSTEMS
Fig. 4 SCHEMATIC OF THE SIMULATED SYSTEM.
Chapter 2 covers the design of the controller block. Trajectory planning, inverse kinematics, and the "camera" area are treated in 3.2, while the dynamics model of the robot is developed in 3.3. The simulation set-up and results are discussed in Chapter 4. Chapter 5 summarizes the conclusions which may be drawn from the investigation, and suggests directions for further study.
CHAPTER 2
ADAPTIVE CONTROL STRATEGY

2.1 Preliminary Remarks

A particular choice of control strategy is now formulated given the goals discussed in the Introduction. The objectives are to design, using digital techniques and in a realistic framework, an adaptive control law for joint control which could be reformulated to include sensory information from the environment, for example in the form of position feedback from the hand of the robot. The common six degree-of-freedom robot is separated into two three degree-of-freedom subsystems. One of these subsystems, consisting of the first three links, is examined. The proposed controller is tested by computer simulation of the dynamic system.

In order to use digital techniques to analyze a continuous-time system, it is necessary to sample the continuous time signals. This means that the continuous function \( f(t) \) is replaced by its value at discrete time instants. If the sampling period is \( T \), the function \( f(t) \) is represented by \( f(kT) \), where \( k \) is the sampling instant, \( k = \{ -1, 0, 1, \ldots \} \). For simplicity the notation will be \( f(k) \).

Referring again to Fig. 4, a closer look is taken at the adaptive controller block. The controller has two inputs: a discrete sequence of desired joint angle vectors \( \theta_d(k) \), and a discrete sequence of observed system output vectors \( \theta(k) \). Except as noted otherwise, in this chapter the reference input \( \theta_d(k) \) will be denoted by \( y_d(k) \) and similarly the measured output variable will be denoted \( y(k) \), in order to conform with standard notation in the controls literature. The controller has one output, the computed variable \( u(k) \). Ways of organizing the calculations in an adaptive framework are now examined.
There are basically two approaches: the Model Reference Adaptive System (MRAS) and the Self-Tuning Regulator (STR). Each arose independently but it has been shown that in some cases the methods are equivalent, for example Landau [16] and Egardt [17]. Fig. 5 shows a block diagram of a MRAS. The heavy lines form an ordinary feedback system. Performance specifications are implicitly specified by a reference model chosen by the designer. The ideal output $y_m$ of the reference model is compared to the output $y$ of the system, and the error is used to adjust the parameters of the controller. This outer loop is the adaptive part. This structure has been called direct because the controller is adjusted directly. The second approach, STR, is illustrated in Fig. 6. An ordinary feedback loop is present here, too, again identified by heavy lines. The other loop is the adjustment mechanism. Input-output data of the system are used to identify the system parameters. Then, the estimated system parameters are used in the controller design block to calculate the controller parameters. This is an explicit STR, because the system is identified explicitly. It is sometimes possible to organize the calculations such that the mapping from system identification + controller parameters is trivial; this is then an implicit STR.

2.2 Design of the Controller

The component parts of the design are now addressed. A STR approach was chosen for this study, closely following the work of Koivo and Guo [13], but in contrast to that work the controlled variable is chosen as position rather than velocity. Discussion is organized according to the framework described by Ljung and Soderstrom [18], who identify five items in their discussion on adaptive control. These are:
i) The choice of the set of admissible controls $u(k)$

ii) The choice of a model structure $M(\theta)$ to represent the true system

iii) The selection of a design method to determine the control law in terms of the model parameters $\theta$

iv) The selection of an identification procedure to determine the system parameters

v) The substitution of the estimated parameters into the control law

Attention is now turned to these issues.

2.2.1 Admissible Controls

It is assumed that the computation time required for the control law is a substantial portion of the sampling time $T$. Therefore $y(k)$ is not available for computing $u(k)$; there is a delay of one sampling period (this is the sort of thing that cannot be handled analytically in the continuous time case). The "best" predicted value of $y(k)$ can be used instead, based on measurements up to time $(k-1)$. The notation $y(k|k-d)$ denotes the best prediction of $y(k)$ given information up to time $(k-d)$. Admissible controls therefore consist of the reference signal $y_d$, past controls, and past output measurements.

2.2.2 Discrete Model Structure

A discrete model is required such that its structure is suitable both for control law determination and for identification of the system parameters. The continuous-time representation of the actuator/robot arm system dynamics is first examined in order to determine suitable discrete candidate structures.
The mathematical representation of the manipulator part of the system is given by a set of second-order, coupled, highly non-linear equations (see Chapter 3). It is assumed that the links of the manipulator have high structural rigidity such that deflection of the links may be ignored. For an n degree-of-freedom system the matrix form of the equations is

\[ D(\theta) \ddot{\theta} + C(\theta, \dot{\theta}) + G(\theta) + \tau_1 = \tau \] (2.1)

where \( \theta \) and \( \dot{\theta} \) are joint positions and velocities respectively, \( D(\theta) \) is an \( n \times n \) inertia matrix, \( \ddot{\theta} \) is an \( n \times 1 \) vector of joint accelerations, \( C(\theta, \dot{\theta}) \) is an \( n \times 1 \) vector of centripetal and Coriolis effects, and \( G(\theta) \) is an \( n \times 1 \) vector of gravitational forces. \( \tau \) is an \( n \times 1 \) input torque vector and \( \tau_1 \) is an \( n \times 1 \) vector of miscellaneous unmodelled disturbances such as loads and friction. This representation assumes there is no contact force at the tip of the manipulator.

As far as the joint actuators are concerned, Vukobratovic [6] suggests that the \( i^{th} \) actuator can be represented by a general state space model of the form

\[ \dot{x}_1 = A_1 x_1 + b_1 N_1 + f_1 \tau_1 \] (2.2)

where \( A_1, b_1, f_1 \) are constant matrices, \( N_1 \) is a saturation function, \( x_1 \) is the state vector of the \( i^{th} \) actuator of dimension \( m_i \); \( x_1 = [\theta_1, \dot{\theta}_1, \ldots, \dot{x}_1] \) where \( \dot{x}_1 \) is the \( (m_i - 2) \times 1 \) vector of remaining states. The saturation function accounts for the fact that actuators have power limitations.

It is later shown (Section 3.3) that for the robot used in the simulation the model for the actuators is of the form

\[ J_1 \ddot{\theta}_1 + C_1 \dot{\theta}_1 + \tau_1 = K_1 u_1 \] (2.3)
which can be written in the above form, $m_i = 2$.

It is possible to combine (2.1) and (2.3) into a state space form

$$\dot{x} = f(x(t), u(t), t)$$  \hspace{1cm} (2.4)

where $\dot{x} = \begin{pmatrix} \dot{\theta} \\ \dot{\theta} \end{pmatrix}$ is a $2 \times 1$ vector.

The $f(x(t), u(t), t)$ can be linearized using a Taylor series expansion and the resulting equation discretized, i.e. form

$$\delta \dot{x} = A \delta x + B \delta u$$  \hspace{1cm} (2.5)

Assuming zero-order hold sampling $T$ the discrete equivalent of (2.5) is

$$x(k+1) = \Phi x(k) + \Gamma u(k)$$

where

$$\Phi = I + A \psi = e^{AT}$$

$$\Gamma = \psi B = \int_0^T e^{At} \, dt \, B$$

$$\psi = \int_0^T e^{At} \, dt$$

$$= T(I + \frac{AT}{2!} + \frac{(AT)^2}{3!} + \ldots)$$

This method is computationally intensive.

For the purpose of designing an adaptive controller a different approach was taken. The gravity, Coriolis, centripetal and disturbing
effects are all treated as disturbances. The coupling is also assumed a disturbance such that the multivariable three degree-of-freedom system is reduced to three single-input/single output systems. A system model of the form

$$[J_i + D_{ii}] \ddot{\theta}_i + C_i \dot{\theta}_i + \tau_{d_i} = K_i u_i$$  \hspace{1cm} (2.7)$$

is proposed for each link and the discrete equivalent found, as follows.

Rewritten to simplify the notation

$$J\ddot{\theta} + C\dot{\theta} + \tau_d = Ku$$  \hspace{1cm} (2.8)$$

The disturbing torques $\tau_d$ are modelled as a time varying sequence $v(k)$ of unknown mean and variance. The continuous time remaining portion can be expressed in Laplace transform notation by

$$Y(s) = \frac{K}{s(Js + C)} U(s)$$

$$= G(s) U(s)$$

In standard form,

$$G(s) = b\left(\frac{a}{s(s+a)}\right)$$  \hspace{1cm} (2.9)$$

where $a = \frac{C}{J}$

$b = \frac{K}{C}$
Using a zero-order hold, the equivalent discrete transfer function is

\[ G(z) = \left( z^{-1} \right) z \left[ \frac{G(s)}{s} \right] \]

where \( z \{ \} \) denotes the z-transform. The result is, with a sampling time \( T \),

\[ G(z) = \frac{bz(e^{-aT+aT-1} + 1-e^{-aT-aTe^{-aT}})}{a(z-1)(z-e^{-aT})} \]

\[ = \frac{B(z)}{A(z)} \]  \hspace{1cm} (2.10)

With some manipulation, replacing \( z \) by the shift operator \( q \), where \( q^{-1}y(k) = y(k-1) \), the equation

\[ y(k) = \frac{B(z)}{A(z)} u(k) \]

can be expressed as

\[ y(k) + a_1 y(k-1) + a_2 y(k-2) = b_1 u(k-1) + b_2 u(k-2) \]  \hspace{1cm} (2.11)

where

\[ a_1 = -(1 + e^{-aT}) \]  \hspace{1cm} (2.12)

\[ a_2 = e^{-aT} \]  \hspace{1cm} (2.13)

\[ b_1 = b/a \left( e^{-aT} + aT - 1 \right) \]  \hspace{1cm} (2.14)

\[ b_2 = b/a \left( 1 - e^{-aT} - aT e^{-aT} \right) \]  \hspace{1cm} (2.15)

The disturbance \( v(k) \) can be modelled in more than one way. For example, it can be assumed that it is filtered white noise, i.e.
\[ v(k) = C(q^{-1}) e(k) \] (2.16)

with

\[ C(q^{-1}) = 1 + q^{-1} + q^{-2} + \ldots \]

Since it is expected that the disturbance would have non-zero mean over a finite time interval it is modelled as

\[ v(k) = h + e(k) \] (2.17)

where \( h \) is an offset term and \( e(k) \) is white noise. This method of subtracting a bias term out is described in Isermann [19]. Thus the complete discrete model of each link for the purpose of control design and parameter estimation is

\[ y(k) + a_1 y(k-1) + a_2 y(k-2) = b_1 u(k-1) + b_2 u(k-2) + h + e(k) \] (2.18)

The coefficients \( a_1, a_2, b_1, b_2, h \) must be estimated.

The treatment of the control law derivation and the estimation section is general - polynomials \( A(q^{-1}) \) and \( B(q^{-1}) \) are used instead of the above specific model. This was done so that modifications could have been made more easily in the simulation should the model (2.18) have proven to be inadequate.

2.2.3 Control Law

The chosen model structure for each link is

\[ A(q^{-1}) y(k) = B(q^{-1}) u(k) + h + e(k) \] (2.19)
with

\[ A(q^{-1}) = 1 + a_1 q^{-1} + a_2 q^{-2} + \ldots + a_m q^{-m} \]  \hspace{1cm} (2.20)

\[ B(q^{-1}) = b_1 q^{-1} + b_2 q^{-2} + \ldots + b_r q^{-r} \]  \hspace{1cm} (2.21)

h is a forcing or offset term and e(k) is white noise, (zero mean, uncorrelated, random). The proposed model for the robot system has \( m=2 \) and \( r=2 \).

Minimum variance control laws, based on a quadratic performance criterion

\[ I = E\{y(k)^2\} \]

have enjoyed some success in adaptive systems applied to the regulation of processes, where the goal is to keep the output constant at some reference value. See Astrom et al. [20] for references.

This provides some motivation for proposing a modified quadratic criterion based on the deviation of the output from the desired level and on the magnitude of the control effort:

\[ I = E\{[y_d(k+1) - y(k+1)]^2 + \epsilon u(k)^2|k-1]\} \]  \hspace{1cm} (2.22)

where \( E[f|k-1] \) denotes the expected value of the function \( f \) given measurements up to time \( (k-1) \), and \( \epsilon \) is a weighting factor. In other words the control effort at time \( k \), \( u(k) \), is chosen to minimize the positional error which will result due to that control one step ahead, but a penalty is attached to large control efforts. There is a delay of one sampling period because the influence of the control \( u(k) \) will only be felt at
y(k+1). This is reflected in the system model by the fact that the leading coefficient of the \( B(q^{-1}) \) polynomial is zero, \( b_0 = 0 \). Therefore a prediction of \( y(k+1) \) is needed.

The expectation is taken with respect to time \((k-1)\) because of the previously mentioned restriction that \( y(k) \) is not available for computing \( u(k) \). If the sampling time \( T \) was very long compared to the computational time to obtain \( u(k) \), one could assume that \( y(k) \) was available.

The problem is stated as follows: it is desired to minimize

\[
I = E\{[y_d(k+1) - y(k+1)]^2 + \varepsilon u^2(k)|k-1]\]

subject to the constraint

\[
A(q^{-1}) y(k) = B(q^{-1}) u(k) + h + e(k) \tag{2.23}
\]

Expanding \( I \) gives

\[
I = E\{y_d^2(k+1) - 2y_d(k+1) y(k+1) + y^2(k+1) + \varepsilon u^2(k)|k-1\}
= y_d^2(k+1) + \varepsilon u^2(k) - 2y_d(k+1) E\{y(k+1)|k-1\}
+ E\{y^2(k+1)|k-1\} \tag{2.24}
\]

where \( y_d \) may be taken outside the expectation operator because it is a known sequence, and similarly \( u(k) \) because it is calculated. From the constraint equation (2.23) \( y(k) \) may be written

\[
y(k) = -a_1 y(k-1) - \ldots - a_m y(k-m) + b_1 u(k-1) + \ldots \\
+ b_r u(k-r) + h + e(k) \tag{2.25}
\]
Multiplying (2.25) by the shift operator q gives

\[ y(k+1) = q \ y(k) \]

\[ = -a_1 y(k) - \ldots - a_m y(k-m+1) + b_1 u(k) + \ldots \]

\[ + b_r u(k-r+1) + h + e(k+1) \]

(2.26)

which is a prediction of \( y(k+1) \).

Denote \( R \) by

\[ R = [-a_2 y(k-1) - \ldots - a_m y(k-m+1) + b_2 u(k-1) + \ldots \]

\[ + b_r u(k-r+1) + h + e(k+1)] \]

(2.27)

such that \( R \) is a function only of past outputs and controls \( y(k-1), u(k-1), \)

\( i > 1 \) and of the \( e(k+1) \). Then

\[ y(k+1) = -a_1 y(k) + b_1 u(k) + R \]

\[ y^2(k+1) = (a_1 y(k))^2 + (b_1 u(k))^2 + R^2 \]

\[ + 2(b_1 Ru(k) - a_1 Ry(k) - a_1 b_1 y(k)u(k)) \]

Examining the last two terms of I which contain in Equation (2.24) the expressions

\[ E\{y(k+1)|k-1\} \]

and

\[ E\{y^2(k+1)|k-1\} \]
and comparing with equations (2.25) and (2.27) reveals that the expectations may be expressed by

i) past outputs \( y(k-1), y(k-2) \ldots \) and inputs \( u(k), u(k-1) \ldots \)

ii) noise terms \( e(k), e(k+1) \) for which by assumption \( E\{e\}=0 \)

iii) terms of the form \( E\{ey\} \) and \( E\{eu\} \), which are also zero since the noise is assumed independent of \( y \) and \( u \), and

iv) terms containing \( E\{e^2\} \), which is unknown, however this is immaterial since \( \partial I/\partial u(k) \) is the quantity of interest.

Performing the differentiation and setting the result equal to zero to determine the minimum gives

\[
\frac{\partial I}{\partial u(k)} = 0 + 2e(k) - 2y_d(k+1)b_1 + 2b_1u(k) + 2b_1R - 2a_1b_1y(k)
\]

\[
= 2[e(k) + b_1(R - a_1y(k) - y_d(k+1))]
\]

\[
= 2[e(k) + b_1(y(k+1) - y_d(k+1))]
\]

\[
= 0
\]

Thus the control law is

\[
e(k) + b_1\hat{y}(k+1|k-1) - y_d(k+1) = 0 \quad (2.28)
\]

where \( \hat{y}(k+1|k-1) \) denotes the predicted value of \( y(k+1) \) given information up
to time \((k-1)\). This value will be obtained by substituting the parameter estimates \(\hat{a}, \hat{b}, \hat{h}\) into (2.28) as explained later.

2.2.4 Identification Procedure

There are many methods for recursively identifying the parameters of an equation, for example, recursive least squares, extended least squares, recursive maximum likelihood, stochastic approximations, instrumental variables, to name a few. There are in addition several nonlinear techniques. Within each method there are variations or design factors to choose from. Another consideration for each method is the particular computational algorithm used. The same method implemented with different computational algorithms gives different performance [21]. An example is computer round-off error occurring due to poorly structured calculations. Thus choosing an estimator which will perform adequately is not easy.

It was decided to go with basic recursive least squares. This technique is known to be robust under many situations and well-conditioned algorithms were readily available. A derivation of the recursive least squares equations is given in Appendix I. A model of the form

\[
A(q^{-1})y(k) = B(q^{-1})u(k) + v(k)
\]

is postulated with polynomials

\[
A(q^{-1}) = 1 + a_1q^{-1} + a_2q^{-2} + \ldots + a_mq^{-m}
\]

\[
B(q^{-1}) = b_1q^{-1} + b_2q^{-2} + \ldots + b_rq^{-r}
\]
and \( v(k) \) some disturbance of unknown characters. Defining the following vectors

\[
\theta^T = [a_1 \ a_2 \ \ldots \ a_m \ b_1 \ b_2 \ \ldots \ b_r] \quad (2.29)
\]

\[
\psi^T(k) = [-y(k-1) - y(k-2) \ \ldots \ - y(k-m) \ u(k-1) \ \ldots \ u(k-r)] \quad (2.30)
\]

The system model may be written

\[
y(k) = \theta^T \psi(k) + v(k) \quad (2.31)
\]

A 'natural' estimate \( y(k) \) is

\[
\hat{y}(k) = \theta^T \psi(k) \quad (2.32)
\]

The least squares principle states that the best estimate \( \theta \) is the one that minimizes the square of the error \( \epsilon(k) = \hat{y}(k) - y(k) \), that is the criterion

\[
I(\theta) = \sum_{k=1}^{N} \epsilon^2(k) \quad (2.33)
\]

The result is

\[
\hat{\theta} = (\phi^T \phi)^{-1} \phi^T y \quad (2.34)
\]

provided the inverse exists. The information matrix \( \phi \) is
The matrix \((\phi^T\phi)^{-1}\) is known as the covariance matrix \(P(k)\). Singularity or near singularity of \(P(k)\) will cause the estimates to be inaccurate.

A recursive form of the equations may be derived. The result is

\[
\hat{\theta}(k) = \hat{\theta}(k-1) + K(k-1) [y(k) - \hat{\theta}^T(k-1) \psi(k)]
\]

\[
K(k-1) = P(k-1) \psi(k) [\lambda + \psi^T(k) P(k-1) \psi(k)]^{-1}
\]

\[
P(k) = \frac{[I - K(k-1) \psi^T(k)] P(k-1)}{\lambda}
\]

The scalar \(\lambda\) is a forgetting factor used when \(\theta(k)\) is time-varying, and it permits discounting of old data. If \(\theta\) is constant \(\lambda\) is set to 1.

In order to estimate the offset \(h\) the parameter vector \(\theta\) is augmented by \(h\) and the measurement vector \(\psi\) is augmented with a 1.

The recursive equations (2.35) are not well-conditioned from a numerical point of view [21]. Problems were encountered in applications, often because \(P(k)\) failed to remain non-negative definite. Various adhoc methods of circumventing the difficulties have been used, but the alternative approach of restructuring the covariance update using a square root type factorization has been shown to be superior. The covariance matrix is factored in a form \(UD1/2\) or \(UDU^T\) where \(U\) is upper triangular and \(D\) is diagonal. In order to avoid potential instabilities in the recursive least squares estimation portion of the
simulation, equation (2.35) are not used directly. Instead an algorithm presented by Bierman [21] is implemented (as a subroutine). It uses $U DU^T$ factorization.

2.2.5 Replacement of the True Parameters

The control law design is based on the certainty-equivalence principle: the controller is designed as if the system parameters were known, and then the estimated values are substituted into the control law. This concept is a special case of the Separation Theorem used in Linear Quadratic Gaussian (LQG) theory. The Separation Theorem is used when optimal control is used on state space models. It enables the design of the controller to be done separately from the estimation of the states, and it may be shown to be valid [22] if (1) the system can be described by linear equations with known parameters, (2) the control is optimized according to a quadratic performance criterion, and (3) the estimation is optimized for Gaussian disturbances, i.e. if the noise is white. In the case of adaptive control the system parameters are of course unknown, and the certainty-equivalence principle holds if the above three conditions are met, except that the parameters should be statistically independent. This condition is probably not satisfied for the case studied here, since there is probably some correlation between the estimated parameters. However, satisfactory results have been obtained by previous researchers investigating adaptive schemes based on certainty-equivalence, even if it is not rigorously justified on a theoretical level.

An estimate for $y(k)$ given measurements up to $(k-1)$ is
\[ y(k|k-l) = \theta^T(k-l) \psi(k) \]

\[ = -a_1 y(k-l) - a_2 y(h-2) + b_1 u(k-l) + b_2 u(k-2) + h \]  \hspace{1cm} (2.36)

Moving ahead one step, an estimate of \( y(k+1) \) given measurements up to \( (k-l) \) is

\[ y(k+1|k-l) = \theta^T(k-l) \psi(k+1) \]  \hspace{1cm} (2.37)

At time \( (k-l) \), \( \theta(k-l) \) is the best filtered estimate of \( \theta \) in the least squares sense. At time \( k \), in the absence of measurements, \( \theta(k-l) \) is the best prediction of \( \theta \) in the least squares sense. One may think that the uncertainty in \( P(k) \) is very high, and so \( K(k-l) \) goes to zero: no weight is given to new measurements because they are absent. So \( \theta(k-l) \) is the best prediction at time \( (k+d) \), and \( \psi(k+1) \) is given by

\[ \psi(k+1) = [y(k|k-l) y(k-l) u(k) u(k-l) 1] \]  \hspace{1cm} (2.38)

Substituting (2.36) into (2.38) and placing the result in (2.37) will give the estimate \( y(k+1|k-l) \). Then the control law (2.28) will give

\[ u(k) = W[y_d(k+1) - (c_1 y(k-l) + c_2 y(k-2) + c_3 u(k-l) + c_4 u(k-2) + c_5)] \]  \hspace{1cm} (2.39)

where
The variable \( W \) is a "weight" or gain and the \( c_j \) are control coefficients. Equation (2.39) gives the control \( u(k) \) at each sampling instant \( k \).

The choice of the adaptive control strategy has been described with respect to five points or topics. This division was made for convenience so that some sort of orderly framework could be followed. Evidently there is much interaction and overlap between the five areas. Many variations on the above themes are possible and concocting an adaptive control strategy is not a problem. However, analysis, especially of the overall scheme, is not easy. Two important considerations are overall stability of the system and convergence of the controller to the optimal one. A general theory is lacking although a few proofs exist for special cases, see for example Ljung and Soderstrom [18], Goodwin and Sin [23], Egardt [24], Narendra and Lin [25], Goodwin, Ramadge, and Caines [26]. Empirical insight may be gained through simulation, thus the discussion on stability and convergence is postponed until the simulation results are covered. Development of the simulation package is treated in the following chapter.

\[
W = \frac{\hat{b}_1}{a + \hat{b}_1} \quad (2.40)
\]

\[
c_1 = \frac{\Delta^2}{a_1 a_2} \quad (2.41)
\]

\[
c_2 = \Delta a_2 \quad (2.42)
\]

\[
c_3 = \Delta b_1 + b_2 \quad (2.43)
\]

\[
c_4 = \Delta a_2 \quad (2.44)
\]

\[
c_5 = \Delta (1 + \hat{a}_1) \quad (2.45)
\]
<table>
<thead>
<tr>
<th>Description</th>
<th>Symbol</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>mass of link 1</td>
<td>$m_1$</td>
<td>5.652 kg</td>
</tr>
<tr>
<td>mass of link 2</td>
<td>$m_2$</td>
<td>5.440 kg</td>
</tr>
<tr>
<td>mass of link 3</td>
<td>$m_3$</td>
<td>0.755 kg</td>
</tr>
<tr>
<td>centre of mass, link 2</td>
<td>$x_2$</td>
<td>-211 mm</td>
</tr>
<tr>
<td>centre of mass, link 3</td>
<td>$z_3$</td>
<td>71 mm</td>
</tr>
<tr>
<td>moment of inertia, link 1</td>
<td>$I_{1yy}$</td>
<td>$4.03 \times 10^{-2}$ kg-m$^2$</td>
</tr>
<tr>
<td>moment of inertia, link 2</td>
<td>$I_{2zz}$</td>
<td>$2.43 \times 10^{-1}$ kg-m$^2$</td>
</tr>
<tr>
<td>moment of inertia, link 3</td>
<td>$I_{3yy}$</td>
<td>$3.83 \times 10^{-3}$ kg-m$^2$</td>
</tr>
<tr>
<td>length of link 2</td>
<td>$a_2$</td>
<td>356 mm</td>
</tr>
<tr>
<td>armature resistance</td>
<td>$R$</td>
<td>0.75 ohm</td>
</tr>
<tr>
<td>motor constant</td>
<td>$K_T$</td>
<td>$4.308 \times 10^{-2}$ N-m/amp</td>
</tr>
<tr>
<td>moment of inertia, motors</td>
<td>$J_{act}$</td>
<td>$1.413 \times 10^{-4}$ kg-m$^2$</td>
</tr>
<tr>
<td>counter emf constant</td>
<td>$K_e$</td>
<td>$4.3 \times 10^{-2}$ volt-s/rad</td>
</tr>
<tr>
<td>gear ratio</td>
<td>$n$</td>
<td>400</td>
</tr>
</tbody>
</table>
### Table II
Kinematic Parameters for the RSI Robot

<table>
<thead>
<tr>
<th>JOINT</th>
<th>$a_1$</th>
<th>$d_1$</th>
<th>$\alpha_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>$d_1$</td>
<td>$\frac{\pi}{2}$</td>
</tr>
<tr>
<td>2</td>
<td>$a_2$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>$-\frac{\pi}{2}$</td>
</tr>
</tbody>
</table>

Notes: (i) $a_2$ is the length of link 2, which is 356 mm.

(ii) $d_1$ is the distance between the origin of frame 1 and the base frame 0, and may be chosen to be zero.
Fig. 5  BLOCK DIAGRAM OF MRAS.
Fig. 6  BLOCK DIAGRAM OF STR.
3.1 The RSI Robot

In this work the RT-3 Robot manufactured by Robotics System International (Sydney, B.C.) serves as prototype for the simulation. It is subsequently referred to as the RSI robot. This small general purpose robot was purchased in the fall of 1984 by the Department of Electrical Engineering at U.B.C. Fig. 7 illustrates its main characteristics. The RSI is a manipulator with six revolute links. It has no offsets, so the kinematical structure is simplified, and it has a spherical wrist, which is common in industrial manipulators (e.g. Puma 560) and which further facilitates the analysis. The joints are driven through a gear train by DC motors with armature control. Presently, there are six analogue PID servo boards, each controlling one joint of the robot. Each loop is tuned manually by adjusting two gain screws. Voltage control signals are downloaded from a small computer to the servo boards.

The main advantage in using this robot for the study is that its characteristics are known. Dimensions are available from a set of drawings; masses and principal moments of inertia for each link were obtained by Allen [27]; motor characteristics were measured from step response tests. These values are tabulated (for the first three links) in Table I. Since so much is known about this robot, one is fairly confident that a reasonable simulation model can be obtained once the theory has been applied to it.

The theoretical background necessary to implement a computer simulation of the RSI robot includes kinematics and dynamics. First, the
kinematics of the arm is described, and the importance of this area is discussed. Then, the equations of motion are derived. The assumptions made in developing the dynamics model are noted. These aspects are now examined in detail.

3.2 Kinematics

The direct kinematics problem is to find the position and orientation of the end effector, or tool, of the robot given the joint angles $\theta$. This is easy. The inverse kinematics problem is to determine the joint angles which will result in a desired tool orientation and position relative to fixed base coordinates. This is generally not easy. Both of these problems need to be solved for the RSI robot.

The direct kinematics analysis is needed because the difference between desired and actual endpiece position will provide a measure of error and hence performance of the controller. The desired endpiece position is of course known a priori. This information is downloaded from a higher level controller: the path planner. The actual endpiece position is either measured or calculated from measurements of joint angles. In an advanced real robot position might be measured by a vision system. However, most present-day robots use potentiometers or optical encoders at the joints and measure joint angles. Endpiece position may then be inferred. In this computer simulation of the manipulator, joint angles will be "measured" by numerically integrating the equations of motion. "Noise" may be added to simulate inaccuracies of a real system. Using the direct kinematics, the position of the endpiece is then calculated.

In order to relate positions on one link relative to another, a coordinate frame is attached to each link. This is described in greater
Transformation of vectors from one coordinate frame to another is done by a $4 \times 4$ homogeneous transformation matrix $T$. The $T$ matrices are used in the dynamics section to obtain the equations of motion of the manipulator. So, a kinematic analysis is necessary.

The inverse kinematics problem must be solved as well because an elementary trajectory planner is needed in the simulation. The user will input the parameters of a trajectory which the endpiece must follow. For a line, for example, the initial position, the slope, and the velocity of the endpiece is a set of inputs. The path planner will calculate the increment in position at each sampling instant $k$. Using inverse kinematics this vector of desired cartesian position is converted to a vector of the desired joint angles. Each angle is downloaded to its corresponding joint controller which sends (hopefully) the appropriate voltage signal $u(k)$ to the joint motor. The system is illustrated in Fig. 2(a). The loop around the hand is not closed; with feedback from the end effector, the system would look as in Fig. 2(b).

3.2.1 Direct Kinematics

Returning to kinematics, a means of relating positions on one link to a fixed frame or to another moving link is established. The principal points are covered here. Details and examples are found in Paul [28]. An alternative reference is Lee's paper [29], which gives a good exposition of robot arm kinematics.

Given a point $P$ and two coordinate frames $F_1$ and $F_2$ as shown in Fig. 8, the position of $P$ relative to $F_1$ is expressed as a vector $^{1}r$, whereas the position of $P$ relative to $F_2$ is expressed as $^{2}r$. A left superscript will denote which coordinate system the vector is referred to.
The vectors $\mathbf{r}$ and $\mathbf{r}'$ are related by

$$\mathbf{r} = \mathbf{d} + \mathbf{R}_2 \mathbf{r}'$$

(3.1)

where $\mathbf{R}_2$ is a rotation matrix and $\mathbf{d}$ is the distance between the origins of the two frames.

The transformation between $\mathbf{r}$ and $\mathbf{r}'$ may be effected with one matrix by using homogeneous coordinates. An extra element is added to vectors, so that given

$$\mathbf{r} = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

The vector $\mathbf{v}$ is defined

$$\mathbf{v} = \begin{bmatrix} x \\ y \\ z \\ \omega \end{bmatrix}$$

with $x' = \frac{x}{\omega}$, $y' = \frac{y}{\omega}$, $z' = \frac{z}{\omega}$. The element $\omega$ is commonly taken to be 1.

Now, transformations are expressed as

$$\mathbf{v} = \mathbf{A}_2 \mathbf{v}'$$

where

$$\mathbf{A}_2 = \begin{bmatrix} \mathbf{d} \\ \mathbf{R}_2 \\ \mathbf{d} \end{bmatrix}$$

(3.2)
The transformation matrices may be chained together to refer a vector through a series of coordinate frames, as follows

\[ \mathbf{v} = A_1 A_2 \ldots A_i \mathbf{v} \]

The base frame is denoted by "0" and superscripts usually omitted. Chains of \( A \) matrices have been called \( T \) transformation matrices,

\[ \mathbf{v} = T_i \mathbf{v} \] (3.3)

In order to relate positions and velocities of points on each link of the robot to a fixed based frame, a coordinate frame is attached to each link. Standard practice in robotics is to use the Denavit-Hartenberg convention [30], for establishing coordinate frames. It is applicable to a general linkage. The base frame is fixed and is numbered '0'. Link 1 is attached to the base by joint 1, link 2 is attached to link 1 by joint 2 and so on. For a rotary manipulator, the joint variable \( \theta_i \) describes rotation of joint \( i \) about the \( z_{i-1} \) axis. The \( x_i \) axis is chosen normal to \( z_i \) from the cross product \( z_{i-1} \times z_i \). The intersection is the origin. The \( y_i \) axis completes the right-hand rule, \( y_i = z_i \times x_i \). The following parameters describe the relationship between neighbouring links:

i) \( a_i \) is the distance between \( z_i \) and \( z_{i-1} \) measured along \( x_i \)

ii) \( d_i \) is the distance between \( x_i \) and \( x_{i-1} \) measured along \( z_{i-1} \)

iii) \( \alpha_i \) is the angle between \( z_{i-1} \) and \( z_i \) measured about \( x_i \), using the right-hand rule.
Some flexibility is afforded by choosing the above parameters in such a way as to align an axis in a desired direction. Paul [28] describes how to place coordinate frames on the robot links, but Lee [29] gives an algorithm which is easier to follow. In any case, for a rotary manipulator, the clearest exposition is Hollerbach's [31], from which the parameter descriptions $a_i^i$, $d_i^i$ and $a_i^i$ are taken. The result of applying the above procedure to the RSI robot is given in Fig. 9. The choice of coordinate frames is important, since the whole description of the robot's behaviour rests on it, from trajectories to dynamics equations. Since Hollerbach's paper contains the most efficient inverse kinematics solution developed to date (by Featherstone [32]), his choice of coordinate frames is followed. Table II gives the kinematic parameters for the RSI robot.

The next step is to obtain the $A_i$ matrices from the choice of coordinate frames. Using the above definitions, a vector $\mathbf{v}$ expressed in link $i$ coordinates may be re-expressed in terms of link $i-1$ coordinates as $\mathbf{v}^{i-1}$ by performing the following sequence of sub-transformations:

1) first, rotate by an angle $\theta_i$ about $z_{i-1}$ to bring the $x_i$ axis parallel to the $x_{i-1}$ axis
2) second, translate along $z_{i-1}$ a distance $d_i$ so that $x_i$ and $x_{i-1}$ coincide
3) third, translate along $x_i = x_{i-1}$ a distance $a_i$ to bring the two origins together
4) finally, rotate by an angle $\alpha_i$ about $x_i$ to bring the coordinate frames into coincidence.

The result is the general matrix $A_i$ given by
\[ A_i = \begin{bmatrix}
\cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\
\sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\
0 & \sin \alpha_i & \cos \alpha_i & d_i \\
0 & 0 & 0 & 1
\end{bmatrix} \]  

(3.4)

For the RSI robot, the resulting \( A_i \) matrices, \( i = 1, 2, 3 \) are given in Appendix II, along with the transformation matrices \( T \).

The direct kinematics have now been covered. In summary, given a position vector defined with respect to the \( i^{th} \) link coordinate frame, \( v^i \), this vector may be transformed to base coordinates given the angles \( \theta_i \), \( \ldots \), \( \theta_i \) and the matrix \( T_i \) by the simple multiplication

\[ v = T_i^1 v^i. \]

3.2.2 Inverse Kinematics

The inverse kinematics are now addressed. Featherstone's technique [32] as described by Hollerbach [31] is used for the first three links of the RSI robot. The method splits the solution at the wrist of the robot, so it is suitable in the present case.

The vector \( p_i \) is defined as the vector from the coordinate origin 0 to coordinate origin \( i \), and \( p^i_{i-1} \) is the vector from origin \( i-1 \) to origin \( i \). These definitions are illustrated in Fig. 10. The vector \( p_i \) is known, since it represents the desired cartesian position of the endpiece with respect to the fixed base frame. Since joints 2 and 3 move in a plane perpendicular to the \( x_0-y_0 \) plane, the first angle is easily found

\[ \theta_1 = \tan^{-1} \left( \frac{p_{iy}}{p_{ix}} \right) \]  

(3.5)
Note that a degeneracy occurs when $p_{4x} = p_{4y} = 0$, corresponding to a wrist position along the $z_0$ axis.

To find the next two angles, $p_4$ is expressed in link 1 coordinates, reducing the problem to a planar two-link one. Defining

$$P_w = P_4 - P_1$$

Then

$$P_w = P_4 - d_1z_0$$ \hspace{1cm} (3.6)

Expressed in link 1 coordinates

$$^1P_w = [r \quad p_{wz} \quad 0 \quad 1]^T$$ \hspace{1cm} (3.7)

where $r$ has been defined from

$$r^2 = p_{4x}^2 + p_{4y}^2$$ \hspace{1cm} (3.8)

By the cosine rule

$$\sin \theta_3 = \frac{a_2^2 + z_1^2 - (r^2 + p_{wz}^2)}{2a_2z_1}$$ \hspace{1cm} (3.9)

And

$$\cos \theta_3 = \pm \sqrt{1 - (\sin \theta_3)^2}$$ \hspace{1cm} (3.10)

where the + sign is elbow down and the - sign is elbow up. Then,
\[ \theta_3 = \tan^{-1} \left( \frac{\sin \theta_3}{\cos \theta_3} \right) \quad (3.11) \]

Finally, \( \theta_2 \) is found by expressing \( \mathbf{p_w} \) as

\[ \mathbf{p_w} = a_2 \mathbf{x_2} + \bar{z}_1 \mathbf{z}_3 \quad (3.12) \]

\[ \mathbf{1p_w} = \begin{bmatrix} a_2 \cos \theta_2 - \bar{z}_1 \sin(\theta_2 + \theta_3) \\ a_2 \sin \theta_2 + \bar{z}_1 \cos(\theta_2 + \theta_3) \\ 0 \\ 1 \end{bmatrix} \quad (3.13) \]

Solving simultaneously gives

\[ \sin \theta_2 = \frac{\mathbf{1p_wy}(a_2 - (\bar{z}_1 \sin \theta_3)) - \mathbf{1p_wx}(\bar{z}_1 \cos \theta_3)}{a_2^2 + \bar{z}_1^2 - 2a_2(\bar{z}_1 \sin \theta_3)} \quad (3.14) \]

\[ \cos \theta_2 = \frac{\mathbf{1pwx} + sin \theta_2(\bar{z}_1 \cos \theta_3)}{a_2 - (\bar{z}_1 \sin \theta_3)} \quad (3.15) \]

so that

\[ \theta_2 = \tan^{-1} \left( \frac{\sin \theta_2}{\cos \theta_2} \right) \quad (3.16) \]

Care must be exercised in the specification of the quadrants for the \( \theta_i \).

The inverse kinematics have been covered, so that given a pre-specified vector \( \mathbf{p}_w \) describing the desired wrist position, the joint angles \( \theta_1, \theta_2 \) and \( \theta_3 \) may be determined.
3.2.3 Trajectory Planning

Specification of a discrete time series \( p^*_4(k) \) is covered next. First, the workspace of the RSI robot was defined. From the joint limits, given in Table II, a projection of the workspace (in the \( x_0-z_0 \) plane) for links two and three is given by Fig. 11. The origin of the wrist frames, or endpiece of the robot in this case, may be positioned anywhere in this space. It was decided to specify straight line paths for testing the control law.

The initial position and the target position are specified. It is assumed that the arm starts from the rest and that a trapezoidal velocity law is followed in between, as illustrated in Fig. 12. The profile is completely specified by \( a, t_1, t_2, t_3 \). This type of velocity profile is sometimes called "bang-coast-bang", [33] since high initial accelerations are required to bring the endpiece to the travel velocity, and at the end of the motion high decelerations are applied to zero in and stop on the final position.

Using ideas discussed by Duncan [34], the line path is defined by direction cosines \( \alpha \) and \( \beta \) at point \( p^*_4(k) \), as shown in Fig. 13. The velocity \( v(k) \) is known from the velocity profile described above, so the distance travelled in time interval \( T \) (from time \( k \) to time \( k+1 \)) is

\[
\Delta s = v(k) T \quad (3.17)
\]

hence,

\[
p^*_X(k+1) = p^*_X(k) + \Delta s \cos \alpha \quad (3.18)
\]

\[
p^*_Z(k+1) = p^*_Z(k) + \Delta s \cos \beta \quad (3.19)
\]

This method of defining a path may be easily extended to 3-space by specifying another direction cosine.
3.3 Dynamics

The full equations of motion are required in order to simulate the real behaviour of the robot. They also served in choosing a simplified model of the dynamic system for the purpose of designing a controller. As far as the simulation is concerned we wish to relate the output signal of the controller, the voltage \( u(k) \), to the resulting motion of the links, the position \( y(k) \). The vectors \( u(k) \) and \( y(k) \) are conventional notation in control literature, which is why they are used. However, in the dynamics analysis which follows the measured output variable \( y(k) \) which is angular position, will be denoted by \( \theta(k) \) and by \( \theta \) in the continuous representation.

The system consists of the actuators, DC motors in this case, and the manipulation linkage. The current from the controller is applied to the motors, which produce a torque \( \tau_i \) at each joint \( i \). This torque is multiplied through a series of gears, so that a torque \( \tau_i \) is available at the output shaft to drive joint \( i \) to the desired position. The analysis is divided into the actuator dynamics, then the manipulator dynamics, and finally the relationship between the two.

3.2.1 Actuators

The actuators are DC motors with armature voltage control, in other words their speed is controlled by the armature voltage \( u \). Subscript \( i \) refers to the \( i^{th} \) joint and is omitted for now. A counter emf proportional to the armature speed \( \dot{\theta}_m \) is induced

\[
E_c = K \dot{\theta}_m \tag{3.20}
\]

The circuit equation for the armature is
\[ u - E_c = RI_a + \tau_a \frac{dI_a}{dt} \]  

(3.21)

where the time constant is \( \tau_a = \frac{L_a}{R_a} \). Since the inductance \( L_a \) is very small the last term is normally neglected.

The torque developed by the motor is proportional to the armature current and is given by

\[ \tau_m = K_T I_a \] 

(3.22)

Substituting from the preceding equations gives

\[ \tau_m = \frac{K_T}{R} u - \frac{K_T K_e}{R} \dot{\theta}_m \] 

(3.23)

which is an expression for the torque produced by the motor in terms of the control input \( u \), known constants \( K_T, K_e, R \) and the armature speed \( \dot{\theta}_m \).

3.3.2 Manipulator Dynamics

The equations of motion for the linkage are developed next. They are easily obtained by a straightforward though tedious application of Lagrange's equations. The general form of the equations for a manipulator is derived in Paul [28]. This derivation is sketched here. Then it is shown how the result is applied to the first three links of the RSI robot.

The Lagrangian \( L \) is written as

\[ L = K - P \] 

(3.24)

where \( K \) is the kinetic energy of the system and \( P \) the potential energy.

The equations of motion for an n-degree of freedom system are given by
where the \( q_i \) are any convenient generalized coordinates and \( Q_i \) is the generalized force. The equation

\[
\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = Q_i \quad i = 1, \ldots, n \quad (3.25)
\]

constitutes a definition of the generalized force, with \( dW \) being the work done by the external forces during a virtual displacement \( dq \).

For the case of a revolute manipulator, the generalized coordinates may be chosen as the joint angle displacements:

\[
q_i = \theta_i
\]

Then the generalized forces \( Q_i \) are simply the net torques \( \tau_i \) applied at each joint \( i \).

An expression for the kinetic energy is computed next. Consider a differential mass \( dm \) on link \( i \). Its position in link \( i \) coordinates is given by \( \mathbf{r}_i \). In base coordinates, its position is

\[
\mathbf{r} = \mathbf{T}_i \mathbf{r}_i \quad (3.27)
\]

and its velocity is
\frac{dr}{dt} = ( \sum_{j=1}^{i} \frac{\partial T_i}{\partial \theta_j} \delta_j ) \mathbf{r}_r \tag{3.28} \\

The velocity squared is

\left( \frac{dr}{dt} \right)^2 = \text{Trace} \left( \mathbf{r} \mathbf{r}^T \right) \tag{3.29}

Substituting from equation (3.29) gives

\left( \frac{dr}{dt} \right)^2 = \text{Trace} \left[ \sum_{j=1}^{i} \sum_{k=1}^{j} \frac{\partial T_i}{\partial \theta_j} \mathbf{r}_r \mathbf{r}_r^T \frac{\partial T_i}{\partial \theta_k} \delta_j \delta_k \right] \tag{3.30}

The kinetic energy of the particle of mass dm on link is then

\mathrm{d}K_i = \frac{1}{2} \mathrm{dm} \left( \frac{dr}{dt} \right)^2

= \frac{1}{2} \text{Trace} \left[ \sum_{j=1}^{i} \sum_{k=1}^{j} \frac{\partial T_i}{\partial \theta_j} \left( \mathbf{r}_r \mathrm{d}m \mathbf{r}_r^T \right) \frac{\partial T_i}{\partial \theta_k} \delta_j \delta_k \right] \tag{3.31}

The kinetic energy of the entire link is found by summing over the link

K_i = \int_{\text{link } i} \mathrm{d}K_i

= \frac{1}{2} \text{Trace} \left[ \sum_{j=1}^{i} \sum_{k=1}^{j} \frac{\partial T_i}{\partial \theta_j} \left( \int_{\text{link } i} \mathbf{r}_r \mathrm{d}m \mathbf{r}_r^T \right) \frac{\partial T_i}{\partial \theta_k} \delta_j \delta_k \right] \tag{3.32}
The expression in round brackets is pseudo inertia matrix \( J_1 \). See Appendix II for details.

The total kinetic energy is found by summing over the \( n \) links

\[
K = \sum_{i=1}^{n} K_i
\]

\[
= \frac{1}{2} \sum_{i=1}^{n} \text{Trace} \left[ \sum_{j=1}^{n} \sum_{k=1}^{n} \frac{\partial T_i}{\partial \theta_j} J_1 \delta_{jk} \delta_{ik} \right] \tag{3.33}
\]

The potential energy of the system is found next. If gravity is expressed as a vector \( \mathbf{g} \) and the centre of mass of link \( i \) is given by \( \mathbf{p} \) (both in base coordinates), the potential energy of link \( i \) is

\[
P_i = -m_i \mathbf{g} \cdot \mathbf{p}_i = -m_i \mathbf{g}^T \mathbf{p}_i
\]

where \( m_i \) is the mass of link \( i \). Alternatively \( P_i \) may be written

\[
P_i = -m_i \mathbf{g}^T T_i \mathbf{p}_i
\]

The total potential energy is therefore

\[
P = -\sum_{i=1}^{n} m_i \mathbf{g}^T T_i \mathbf{p}_i \tag{3.34}
\]

The next step in the derivation is to form the Lagrangian \( L = K - P \) and perform the required differentiations. This is not done here, as it does not add anything to the understanding of the equations. Details are
adequately covered in Paul's book [28]. The resulting general equation for an \( n \) degree of freedom manipulator is

\[
\sum_{j=1}^{n} \sum_{k=1}^{n} D_{ij} \dot{\theta}_j \dot{\theta}_k + D_i = \tau_i \quad i=1, \ldots, n \tag{3.35}
\]

where

\[
D_{ij} = \sum_{p=\text{max} \ i, j}^{n} \text{Trace} \left( \frac{\partial^2 P}{\partial \theta_j \partial \theta_i} \right) \tag{3.36}
\]

\[
D_{ijk} = \sum_{p=\text{max} \ i, j, k}^{n} \text{Trace} \left( \frac{\partial^2 P}{\partial \theta_j \partial \theta_i} \right) \tag{3.37}
\]

\[
D_i = -\sum_{p=1}^{n} m_p \left( \frac{\partial T}{\partial \theta_i} \right) P_p \tag{3.38}
\]

Terms of the form \( D_{ii} \) represent effective inertia at joint \( i \). The \( D_{ij} \) terms correspond to coupling inertia between joints \( i \) and \( j \). Terms of the form \( D_{ijj} \) give the centripetal forces at joint \( i \) due to velocities at joint \( j \), and terms of the form \( D_{ijk} \) given the Coriolis forces at \( i \) due to movement of \( j \) and \( k \). Finally, the \( D_i \) terms represent gravity loading.

Applying Equation (3.35) to the RSI robot is straightforward. However, since the wrist is being treated as a mass, the equation must be modified slightly and load term \( L_i \) will appear on the LHS of Equation (3.35). The analysis is essentially the same as the one just described, and specific details relating to the RSI are covered in Appendix IV.

Application of Equation (3.35) is outlined here. Using the kinematical definitions of section 3.2, one forms the transformation matrices \( T \).
Appendix II lists these matrices for the RSI robot. Then, the required matrix differentiations are performed. These terms are of the form $\frac{\partial T}{\partial \theta}$, $\frac{\partial^2 T}{\partial \theta^2}$, and $\frac{\partial^3 T}{\partial \theta^3}$. Using the pseudo inertia matrices $J_p$, the trace of each of the matrix products is found and terms collected. Appendix III gives the general form of $J$ whereas Appendix V gives the inertia matrices used for the RSI robot, as well as the center of mass vectors $\rho$ relative to link coordinates. It is assumed that the center of mass of each link lies along one of the link axes. This is not an unreasonable assumption given the construction of the RSI robot. The result is that the products of inertia are zero and all of the off-diagonal terms of the $J$ matrix cancel except for two. The effect is to considerably reduce the number of terms in the equations of motion, without sacrificing too much accuracy.

Assuming that link 1 is locked in position, the following equations are obtained:

$$
\tau_2 = D_2 + D_{22} \ddot{\theta}_2 + D_{23} \ddot{\theta}_3 + 2D_{223} \dot{\theta}_2 \dot{\theta}_3 + D_{233} \dot{\theta}_3^2 + L_2 \tag{3.39}
$$

$$
\tau_3 = D_3 + D_{33} \ddot{\theta}_3 + D_{32} \ddot{\theta}_2 + D_{322} \dot{\theta}_2^2 + L_3 \tag{3.40}
$$

where

$$
D_2 = g[a_2(m_2 + m_3)\cos \theta_2 + m_2 \bar{x}_2 \cos \theta_2 - m_3 \bar{z}_3 \sin (\theta_2 + \theta_3)]
$$

$$
D_{22} = I_{2zz} + a_2^2 m_2 + 2a_2 m_2 \bar{x}_2 + I_{3yy} + a_2^2 m_3 - 2m_3 \bar{z}_3 a_2 \sin \theta_3
$$
\[ D_{23} = I_{3yy} - a_2 m_3 \dot{z}_3 \sin \theta_3 \]

\[ D_{233} = -a_2 m_3 \dot{z}_3 \cos \theta_3 \]

\[ D_{233} = -a_2 m_3 \dot{z}_3 \cos \theta_3 \]

\[ D_3 = -m_3 g \dot{z}_3 \sin(\theta_2 + \theta_3) \]

\[ D_{33} = I_{3yy} \]

\[ D_{32} = I_{3yy} - a_2 m_3 \dot{z}_3 \sin \theta_3 \]

\[ D_{322} = a_2 m_3 \dot{z}_3 \cos \theta_3 \quad (3.41) \]

The structure of \( L_2 \) and \( L_3 \) is necessarily similar

\[ L_2 = L_2 + L_{22} \ddot{\theta}_2 + L_{23} \ddot{\theta}_3 + 2L_{223} \dot{\theta}_2 \dot{\theta}_3 + L_{233} \dot{\theta}_3^2 \quad (3.42) \]

\[ L_3 = L_3 + L_{33} \ddot{\theta}_3 + L_{32} \dot{\theta}_2 + L_{322} \dot{\theta}_2^2 \quad (3.43) \]

Denoting the load mass by \( m_1 \) and assuming that its centre of mass with respect to link 3 coordinates is \( \mathbf{\rho}_1^T = [0 \ 0 \ \overline{z}_1 \ 1] \), then

\[ L_2 = g[m_1 a_2 \cos \theta - m_1 \overline{z}_1 \sin(\theta_2 + \theta_3)] \]

\[ L_{22} = m_1 [a_2 + \overline{z}_1^2 - 2a_2 \overline{z}_1 \sin \theta_3)] \]
\[L_{23} = m_1 [z_1^2 - a_2 z_1 \sin \theta_3]\]

\[L_{223} = -m_1 a_2 z_1 \cos \theta_3\]

\[L_{233} = -m_1 a_2 z_1 \cos \theta_3\]

\[L_3 = -m_1 g z_1 \sin (\theta_2 + \theta_3)\]

\[L_{33} = m_1 z_1^2\]

\[L_{32} = m_1 [z_1^2 - a_2 z_1 \sin \theta_3]\]

\[L_{322} = m_1 a_2 z_1 \cos \theta_3\]

A final comment on the dynamics is that instead of using the
Lagrangian approach, the Newton–Euler method could have been used to
develop a simulation algorithm. The latter approach consists of a set of
forward–backward recursive equations which can be organized in an efficient
manner. The paper by Walker and Orin [35] evaluates four versions of the
algorithm. Although computational efficiency may have been gained by using
a Newton–Euler algorithm, it is felt that this issue is of lesser import­
ance in the present context. For one thing, only the motion of two and at
most three links is being simulated. On the other hand, the Lagrangian
formulation provides physical insight into the behaviour of the system from
a controls point of view.

3.3.3 Simulation Equations

The final step is to relate the torque \( \tau_{m_i} \) developed by the motor of
joint i, given in Equation (3.23) to the torque \( \tau_i \) available at the joint
output shaft (Equation 3.35). Consider Fig. 14, which is a schematic of motor, drive train, and link, and the corresponding free-body diagrams. Taking moments about the motor shaft and neglecting friction gives

\[ J_{\text{act}_i} \theta_{m_1} = T_{m_1} - F_1 r_1 \]  

(3.45)

where \( J_{\text{act}_i} \) is the inertia of the \( i^{th} \) actuator, \( F \) is the contact force and \( r \) the pinion radius. On the joint shaft the net torque, previously defined as \( \tau_i \), is simply

\[ \tau_i = n_i F_1 r_1 - C_1 \dot{\theta}_1 \]  

(3.46)

where \( n_i \) is the gear ratio and \( C_1 \dot{\theta}_1 \) is a friction term. Eliminating \( F \) and \( r \) from the equations gives

\[ n_i J_{\text{act}_i} \theta_{m_1} = n_i \tau_{m_1} - \tau_i - C_1 \dot{\theta}_1 \]  

(3.47)

Substituting from Equation (3.23) gives

\[ n_i J_{\text{act}_i} \theta_{m_1} = \frac{nK_e}{R} u_1 - \frac{nK_e}{R} \dot{\theta}_1 - \tau_i - C_1 \dot{\theta}_1 \]  

(3.48)

Motor speed and angular acceleration are related to joint angular velocity and acceleration by the gear ratio \( n \)

\[ \dot{\theta}_{m_1} = n \dot{\theta}_i \]  

\[ \ddot{\theta}_{m_1} = n \ddot{\theta}_i \]
and

\[
\dot{\theta}_m = n_i \dot{\theta}_i
\]

so that the following expression is obtained

\[
\frac{nK_T}{R} u_i = \tau_i + (n^2 J_{\text{act}}) \dot{\theta}_i + \frac{n^2 K_T e + C}{R} \dot{\theta}_i
\]

(3.49)

This equation forms the basis for the dynamical simulation, with \( \tau_i \) given by Equation (3.35).

In order to effect the simulation on computer the equations of motion (3.49) must be transformed into a set of first order equations. This is done in Appendix VI.

3.4 Summary

Together with the kinematics developed in section 3.2, the dynamics equations formulated here can be regarded as an experimental simulation package onto which different controllers could be connected. Details of the computer programs are covered in the following chapter. This chapter has developed the equations necessary to implement the algorithms.

A trajectory planning scheme has been formulated using direction cosines and initialization parameters. The output is a vector \( p_k(k) \), a discrete time series describing the desired position of the robot endpoint. The inverse kinematics have been described, such that \( p_k(k) \) may be transformed to the corresponding joint coordinates \( \theta_d(k) \). It has been shown how the actual joint position \( \theta(k) \) may be transformed, using direct kinematics, to a cartesian vector \( x(k) \) for comparison with \( p_k(k) \).
A dynamics model of the robot, including motors, has been developed. The resulting equations are second-order, coupled, and nonlinear. Flexibility of the links, backlash, and static friction are ignored. These equations may be numerically integrated so that given a control input $u(k)$ and a set of initial conditions $\theta(k)$ and $\dot{\theta}(k)$, the resulting angular positions and velocities one integration step further may be found, i.e. $\theta(k + t_{\text{step}})$, $\dot{\theta}(k + t_{\text{step}})$.

The analysis has been done with respect to an existing robot called the RSI, for which kinematical and dynamical data is available, enabling a realistic simulation to be performed. The simulation model can act as a testbed for various control strategies which use $\theta(k)$ and $\theta_d(k)$ to calculate $u(k)$. Evaluation of the adaptive controller is the topic of discussion of the next chapter.
Pages 58 and 59 numbering not used.
FIG. 7 THE RSI ROBOT
Fig. 8  POSITION OF POINT P RELATIVE TO TWO FRAMES
FIG. 9  RSI FRAMES
FIG. 10 VECTOR DEFINITIONS
FIG. 11 WORKSPACE
Fig. 12 TRAPEZOIDAL VELOCITY LAW

\[ a_o = \frac{v_c}{t_1} \]

\[ a_i = \frac{-v_c}{(t_3 - t_2)} \]
FIG. 13 DIRECTION COSINES
FIG. 14 DRIVE SCHEMATIC
EVALUATION OF THE CONTROL STRATEGY

4.1 One-Degree-of-Freedom Case

Since the behaviour of the robot system resulting from the action of the proposed control was unknown, and since bugs can easily slip into computer programs, it was decided to proceed in a modular fashion. The dynamics model of the RSI robot was put aside, and instead the adaptive control law is tested on a simple, one-degree-of-freedom, second order system of the form

\[ J\ddot{\theta} + C\dot{\theta} + \tau_d = Ku \]  

where \(\tau_d\) is some disturbance and \(J, C,\) and \(R\) are constant. A block diagram of the simulated system is shown in Fig. 15. In the full simulation discussed in the next section the dotted portion is one joint and its actuator, with \(\tau_d\) representing interaction from other links and gravity effects; the inertia \(J\) is variable because of the movements of the other links. For this initial study, however, \(J, C, K\) and \(\tau_d\) were assumed constant.

In fact, to simplify things further and to gain insight into the behaviour of the control law under ideal circumstances, it was first assumed that the system model parameters were known and the estimation portion of the control law was shut off. Thus the controller has fixed coefficients calculated according to the equations of Chapter 2. In this way it was possible to see if the control law works at all for the class of system models investigated here. Also, using this approach some knowledge of performance standards is obtained.
Once these initial tests were done and a working skeleton computer program was developed, complexity was added piece by piece. The first module to be added to the "single link" simulation was the estimator. This was an important step, since theoretical deficiencies were likely to show up at this stage. It was felt that if the adaptation worked for the one-degree-of-freedom, constant parameter case, there was a good chance that it would work for slowly varying parameters in the two-link case. Evidently, we were able to progress to this stage.

This section covers the simulations for the one-degree-of-freedom system under two situations:

(a) system model parameters are known and need not be estimated,
(b) system model parameters are unknown, but constant, and the least squares estimator must be used.

Equation (4.1) is put in a form suitable for integration by a subroutine package [36].

Using the equation for the transfer function (2.9), with \( T, a = C/J, \) and \( b = K/C, \) the model parameters may be calculated from Equations (2.12) to (2.15), and the control coefficients from Equations (2.40) to (2.45).

The values of \( J, C, \) and \( K \) were taken as the approximate values for the second link of the RSI arm with

\[
J = n^2 J_{act} + I_{zz} \\
C = \frac{n^2 K T}{R} \times 1.05 \\
K = \left(\frac{nT}{R}\right)_2
\]
The disturbance D was taken as the gravity load of a mass of 8 kg placed at 250 mm from the axis of rotation of the link. The resulting values for the model parameters and control coefficients are given in Table III for two sampling periods.

Five test paths were chosen to test the control law: two step inputs, two ramps, and one trapezoidal velocity law. The characteristics of the test reference inputs are given in Table IV.

Two programs were written. One simulates the known case, with no estimation, and is outlined in flowchart form in Fig. 16. The second simulates adaptive control with the estimation and design blocks of Fig. 15 tuned in. Fig. 17 gives a flowchart of that program.

Several runs were made and the salient variables are given in Table V. The first runs test the behaviour of the control law with the exact coefficients matched to the system model. Subsequent runs test the adaptive strategy and in some cases the response is compared to the ideal situation when the parameters are known. Results are now discussed.

### 4.1.2 Results and Discussion

#### Effect of Varying $\epsilon$

The first runs test the effect of tuning by using the pseudo-gain $\epsilon$ for the known case. Lower $\epsilon$ means less penalty on the control effort. The disturbance was set to zero, and the small step input was the reference signal.

For each of the sampling times it was attempted to find an $\epsilon$ giving similar shapes of underdamped response, critically damped response, and overdamped response.

Fig. 18a,b,c shows the result for $T = 0.010$ s and Fig. 19a,b,c for $T = 0.020$ s.
The range for $T = 0.010 \text{ s}$ is

$$\varepsilon = 0.01 \times 10^{-5} \text{ to } 0.10 \times 10^{-5}$$

The range for $T = 0.020 \text{ s}$ is

$$\varepsilon = 0.05 \times 10^{-5} \text{ to } 0.35 \times 10^{-5}$$

If it is considered that the control effort goes as $1/\varepsilon$ then a broader band is obtained for $T = 0.010 \text{ s}$ and the system is harder to tune. The critically damped values were chosen for subsequent runs.

It is seen from the response curves that the steady state error is zero, as expected since the discrete transfer function contains a $1/(z-1)$ factor (corresponding to $1/s$ in the continuous time case). Fig. 19d shows the effect of selecting $\varepsilon$ quite low: The control signal shows oscillations and the system overshoots.

**Effect of Disturbances**

Fig. 20a shows the response for the adaptive case given a step input, $T = 0.020 \text{ s}$. There is a steady state error of 0.013 radians, and the steady state control input settles at approximately the value of the disturbance. The same type of behaviour occurs at $T = 0.010 \text{ s}$. Thus, it seems that the adaptive controller is able to measure the offset, but the response is just like a proportional controller. Because the disturbance enters ahead of the natural integrating terms in the system it cannot be subtracted out. The best that the system can do is to feed-forward an equivalent torque, as shown in Fig. 20b. The control law
\[ \varepsilon u(k) + b_1(y(k+1) - y_d(k+1)) = 0 \]

in fact when rearranged, interpreting \( y(k+1) - y_d(k+1) \) as an error signal looks like a proportional controller.

When a disturbance of the same magnitude is added to the known control case the same steady state error occurs (Fig. 20c).

When the disturbance is removed from the adaptive case the response is just as good as in the case of known parameters (Fig. 22), at both sampling rates.

For a larger step input, Fig. 23 shows a comparison between the responses produced by a known controller and the ones produced by the adaptive one. A disturbance is present in the adaptive runs, and so a steady state error occurs. The rise time is slightly longer in the adaptive case.

Ramps

Two ramp inputs were simulated and results between known and adaptive control compared. Fig. 24a and b shows the results. There is a time lag in position, which decreases with larger gain (smaller \( \varepsilon \)). When there is no disturbance the steady state error goes to zero at the end of the ramp.

Trapezoidal Law

Fig. 25 illustrates this case: the response of both of the controllers is very similar, showing that adaptation is taking place and that the adaptive law appears to be stable and convergent.
Interestingly the parameter estimates did not converge to their true values, but the control coefficients converged to a fixed relationship between each other.

Effect of Initial Estimates

In the preceding runs non-zero initial values were used for the parameter estimates. Three runs were done with zero initial conditions, and with an initial covariance matrix of $P(0) = 100,000 I$. The adaptive controller was stable, Fig. 26a, however it became unstable at $P(0) = 10,000 I$, as illustrated by Fig. 26b. The controller saturated at 10 volts so the covariance matrix became singular, being locked in at those values of $u(k-1), u(k-2)$. When the upper limit on $u(k)$ was removed the controller was once again stable, as shown in Fig. 26c.

4.2 RSI Robot Simulation

Since the adaptive controller seemed to perform fairly well for the single-degree-of-freedom case, where an equation of motion with constant coefficients was used, the simulation was extended to the system model of the RSI robot. Here, the inertia terms vary, and the disturbances due to gravity, Coriolis, and centripetal effects vary, since the equations are coupled and nonlinear. It was conjectured that adaptation would occur, but the effect of the varying disturbances was uncertain. Their magnitude might make the controller worse; on the other hand, the fact that they are time-varying might inject enough excitation into the estimation and improve performance.

The detailed development of the control law and of the simulation model equations is done in the previous chapters. The segments which make
up the simulation programs are now briefly described. The set-up was made for simulating the first three links of the RSI robot, however during the simulations it is assumed that link one is locked in position. It was felt that the major features of the behaviour of the controller could be assessed by examining the response of two interacting links. The computer programs were in any case structured for a three-link simulation, and some subroutines include three links but $\theta_1$ and $\dot{\theta}_1$ are set to zero.

One of the segments of the simulation consists of the dynamics model of the RSI robot.

The equations of motion are, from Equation (3.49)

$$\left(\frac{nK_T}{R}\right)u_1 = \tau_1 + (n^2J_{\text{act}})\ddot{\theta}_1 + \left(\frac{n^2K_e}{R} + C\right)\dot{\theta}_1 \quad i = 1, 2, 3$$

The voltage $u$ is the input from the controller to the system. The quantities $K_T$, $K_e$, $R$ are motor constants; $n$ is the gear ratio; $J_{\text{act}}$ is the motor inertia; and $C$ is a friction factor. The joint rotation is $\theta$, and the torque $\tau$ at the joint shaft for links two and three is given by

$$\tau_2 = D_2 + D_{22}\ddot{\theta}_2 + D_{23}\ddot{\theta}_3 + 2D_{223}\dot{\theta}_2\dot{\theta}_3 + D_{233}\dot{\theta}_3^2 + L_2$$

$$\tau_3 = D_3 + D_{33}\ddot{\theta}_3 + D_{32}\ddot{\theta}_2 + D_{322}\dot{\theta}_2^2 + L_3$$

where the D quantities contain trigonometric functions and robot physical parameters, and $L_1$ accounts for a load at the end of the third link.

This dynamic model of the robot is transformed into a set of equivalent first order equations (Appendix VI). An integration package is used such that, following initialization, it may be recursively called, given $u(k)$,
the input from the controller, to produce the actual response $y_a(k+1) = y_a(k+1)$ one sampling time further. It is assumed that there is the equivalent of a zero-order hold in the system, so that $u(k)$ is constant for $t < kT < (t + kT)$ where $T$ is the sampling time, and $k$ the sampling instant. A package called DDE was used [36]. The integration step size was set to $T/20$.

The second segment is the control block. The control law was found to be
\[ \varepsilon u(k) + b_1[y(k+1|k-1) - y_d(k+1)] = 0 \]
where $\varepsilon$ is a pseudo-gain and $y_d(k+1)$ is the desired position one step ahead.

Substituting the prediction $y(k+1|k-1)$ given by Equation (2.37), the resulting control is given by Equation (2.39)
\[ u(k) = W[y_d(k+1) - (c_1y(k-1) + c_2y(k-2) + c_3u(k-1) + c_4u(k-2) + c_5)] \]
where the coefficients are given by Equations (2.40) to (2.45) in terms of $a_1, a_2, b_1, b_2, h$.

Limits of $\pm 10$ volts are placed on $u(k)$, to reflect the actual saturation limits of the RSI motors.

The model coefficients $a_1, a_2, b_1, b_2, h$ are estimated at each step using recursive least squares (Eq. 2.35) and these estimates used to update the control coefficients. The recursive least squares estimation is implemented as a subroutine and follows Bierman's factorization of the covariance matrix [21].

The effect of varying the forgetting factor $\lambda$, varying the initial
0 parameter estimates, and varying the initial value of the covariance matrix P were investigated.

The final segment of the simulation, which completes the diagram given in Fig. 4, consists of assorted kinematic operations. One is the transformation of the vector of actual angular positions $y(k)$ to the actual cartesian position $x(k)$, for comparison with desired cartesian position $p_0(k)$ of the endpiece of the RSI robot. The vector $p_0(k) = x_d(k)$ is user-defined. Given the initialization parameters consisting of direction cosines, initial position, initial velocity, and desired velocity profile, the trajectory planning algorithm outputs $p_0(k)$ at each time $k$. This desired endpiece position is transformed using the inverse kinematics subroutine into the vector of joint angles, $y_d(k)$. Timing of the program is organized so that $y_d(k+1)$ is available to calculate $u(k)$.

A flow chart of simulation program 'R' is given on Fig. 27. Similarities with the block schematic of Fig. 4 referred to earlier may be noted.

4.2.1 Results and Discussion

Three paths were chosen to investigate the behaviour of the controller. Results of the simulation are discussed in each case.

Circle

The first path simulated was a simple circle. Joint 3 is required to stay locked in position at $-45^\circ$, while joint 2 starts from rest and is commanded to move at $24^\circ/s$ counter clockwise. Fig. 28 illustrates the configuration. This case is essentially the same as the ramp input for the single-degree-of-freedom case described in the previous section. The
difference is that the inertia term is varying, that is, the coefficient of \( \theta_1 \), the angular acceleration. In addition, the disturbances are varying. The second link must reject gravity loads, which switch signs as the arm swings around. A load at the end of the third link will aggravate these effects. The third link, in order to remain stationary, must reject the \( D_{22}^2 \) disturbance as well as gravity loads.

Several runs were made under different conditions. The parameters for each run are given in Table VI. In each case the arm starts from rest. The quoted velocity of 24°/s corresponds to 15 seconds for one revolution and a tip velocity of about 20 cm/s. The endpoint is chosen as the origin of the wrist frames. Lower velocities were tried but are not included here because plotting would have required too many points and in any case the performance requirements were less stringent. The velocity of 24°/s for link two is probably close to maximum performance since the motor runs at over 80% capacity on the up legs and occasionally saturates (Fig. 29a).

Several runs were required for tuning the pseudo-gain \( \varepsilon \) which weights the control. Values of \( \varepsilon_2 = 1.2 \times 10^{-6} \) and \( \varepsilon_3 = 1.8 \times 10^{-5} \) were found to be satisfactory and were not varied from run to run. The forgetting factors were intially set to \( \lambda = 0.9 \) for both links. For the first few runs the initial parameter estimates \( \theta \) were chosen as the linearized values about the initial position. Covariance matrices of \( P_2(0) = 100 \) I and \( P_3(0) = 1000 \) I proved adequate in that case.

Effect of Magnitude of Disturbance

With the above conditions, the first simulation was performed by setting \( g = 0 \) and the load mass \( m_1 = 0 \). The load mass is actually the equivalent of the wrist assembly of the RSI robot, slightly over 2 kg.
Setting gravity to zero simulates a planar arm and reduces disturbances. Fig. 29b shows the path traced out by the endpoint: excellent positional control is achieved. The gap is a combination of lag and steady state error occurring at the end of the run. For reasons similar to the single-link case, the steady state error is not rejected, and again the controller behaves as a proportional controller, lacking integrating qualities. Fig. 29c is a plot of joint angle 3 vs time and shows that it is locked at 45° as required. Fig. 29d shows a plot of angle two and the control input vs time. As expected for a ramp input there is a lag in position (of 3° - 6°). The control signal is generally well-behaved.

A second simulation was performed, this time including gravity, but no load. The results are similar as far as tracking is concerned (Fig. 29e); however the torque requirements are more variable, requiring a higher $u_2(k)$ as the arm goes up and lower inputs as it goes down. The third link oscillates ever so slightly about its desired set point (Fig. 29f).

In the third run a load term was added and the above effects are more pronounced, as shown by Fig. 29g. The response, however is good (Fig. 29h).

Effect of Varying $\lambda$

The next two runs show the effect of varying the forgetting factor $\lambda$ in the estimation routine. As an extreme case it was set to .5 for both links. (The recommended range is $0.9 < \lambda < 1$). Recall that $\lambda$ close to 1 corresponds to very slowly varying parameters. Fig. 30a shows that the control $u_2(k)$ becomes jumpy and that the velocity becomes erratic. At the other extreme, $\lambda = 0.995$, the control is very smooth however the system is slow to respond around $t = 11$ seconds when the arm passes $\theta_2 = 270°$ and starts to fight gravity (Fig. 30b).
Effect of Varying Initial Estimates

The final three runs tested the effect of setting the initial parameter estimates to zero. In other words no knowledge of the system is assumed (except the order). Thus a very high covariance matrix is chosen, \( P(0) = 100,000 I \). From Fig. 31a, it is seen that link 3 droops slightly before adjusting back to 45° after about 2.5 seconds of adaptation. The second link similarly hunts around before finding out where it is supposed to be headed, as evidenced by the initial negative torque (Fig. 31b). These results indicate some degree of stability and convergence of the controller.

A large covariance matrix indicates a higher degree of uncertainty in the estimates. To illustrate a poor choice, in the following run the initial estimates were set to zero but \( P(0) \) was chosen fairly low - 100 I. The results is the response of Fig. 31c. A close look at the plots of joint angles vs time, however, reveals (Fig. 31d) that joint 2 converges to its proper controller. Joint 3 stays locked at \( u(k) = -10 \) volts for 9 seconds, however the adaptation seems to click on at that point when the controller output switches to +10 volts to bring back joint 3 to -45°. This type of behaviour is of course unacceptable in an actual system but it does indicate some robustness in convergence. It is speculated that if bounds on \( u(k) \) were not present, adaptation would take place a lot faster.

An alternative to powerful actuators is adjusting the adaptation speed via the forgetting factor \( \lambda \). A few values were tried for joint 3, since the response of joint 2 was acceptable. At \( \lambda_3 = 0.85 \), joint 3 initially drooped but re-adjusted to within one degree of its desired value after 2 seconds, even with \( P(0) = 100 \) I and initial \( \theta = [0] \).
Summary

The above results were encouraging. Good response and adaptation were achieved when the initial parameter estimates were chosen as the linearized values of the initial position, despite the presence of gravity and other disturbances. A forgetting factor close to 1 gives smoother control signals but the system is slow to respond to a disturbance. Zero initial estimates of the model parameters requires that care be exercised in choosing $\lambda$ and $P(0)$. However, convergent and stable behaviour was observed.

Line at 45°

The second test path involved motion of both links. The endpoint of the arm was required to trace a line at 45° in the $x_0-z_0$ plane of base coordinates. Fig. 32 illustrates the initial and final configuration of the robot. The arm starts at rest with links 2 and 3 out-stretched ($\theta_2=0$ and $\theta_3=-90°$). A choice is specified for either elbow up or elbow down tracing of the line. The endpoint is to follow a trapezoidal velocity law: acceleration to the cruise velocity in $t_1$ seconds, constant velocity up to time $t=t_2$ and then deceleration to bring the tip position to rest on the desired target point at time $t_3$. The run parameters are given in Table VII. Although the tip velocities may appear slow (5 cm/s or 15 cm/s depending on the run) the joint rate for link 3 becomes quite high and the motor saturates during a portion of the path. This situation illustrates the importance of (1) good trajectory planners which foresee limitations on joint rates in certain areas of the workspace, and (2) proper design of the kinematical structure of the robot so that the workspace is usable. Evidently the complications increase when more degrees of freedom are involved.
For each run, \( \lambda_2 = \lambda_3 = 0.9 \) and the gains are \( \varepsilon_2 = 1.2 \times 10^{-6} \) and 1.8\( \times 10^{-6} \).

Low Cruise Velocity

The first run includes gravity and a load mass, and is done under conservative conditions of acceleration of 2.5 cm/s\(^2\) to reach a cruise velocity of 5 cm/s. The initial conditions are substituted in the equations of motion to obtain initial model parameter estimates. The disturbance parameter \( h \) is set to zero. The resulting endpoint trace is shown in Fig. 33a, where a reference line is indicated as well for convenience. Maximum deviation from the desired position occurs at the beginning and is about 7 mm, decreasing steadily to 1 mm or less. Fig. 33b show joint angles vs time. The controller was left on for 2 seconds past the 10-second simulation time; steady state error decreases slightly.

In the second run conditions are the same except for higher initial acceleration of 5 cm/s\(^2\). The cruise velocity is still 5 cm/s. The deviation from the path is worse at the beginning (Fig. 33c) possibly due to the initial disturbance parameter \( h \) being set to zero. The motor for joint 3 saturates from \( t = .68 \) s to \( t = 1.84 \) s because the geometry requires high joint rates in that interval, as shown in Fig. 33d.

The same conditions were specified for the third run except that the elbow up specification was given instead of elbow down. Path following improves at the beginning, as shown in Fig. 33e. The explanation for this is that the initial motion of link 2 is up, rather than down, and link 3 does not have a tendency to be dragged down. Final positional accuracy is good.
High Cruise Velocity

In the fourth simulation run the cruise velocity is increased to 15 cm/s. The response shown in Fig. 33f is similar to that of the other runs. Comparing voltage levels, plotted in Fig. 33g, shows that faster variations are required at the higher velocity. The initial covariance matrix estimates were increased to 10,000 I, but little improvement occurs in the deviation pattern at the beginning. In the fifth run gravity and loading effects were removed, and much better performance results, as the plot of Fig. 33h indicates. Thus one may conclude that it takes some time for the controller to adjust to the disturbances.

Finally, a run was made with zero initial estimates for the model parameters and a high $P(0)$. Resulting trajectory of the endpoint is shown in Fig. 33i. Stable and adaptive behaviour of the controller once again occurs.

Summary

For the 45° line the two joint controllers succeed in achieving path tracking within 3 mm with the exception of the initial start-up period where errors as high as 7 mm occur. High joint rates in link 3 preclude tip velocities higher than 15 cm/s from being achieved. Even with initial model parameter estimates of zero, the controllers adapt. Steady state accuracy is within 0.5° for each joint.

Horizontal Line

The final test path is a horizontal line. The robot starts from rest with joint 2 at 60° and joint 3 at -120°, as shown in Fig. 34a. A trapezoidal velocity law is followed, as described for the previous line.
path. Final configuration is $\theta_2 = 96.5^\circ$ and $\theta_3 = -180^\circ$. Table VIII gives the run parameters. This path was more difficult to tune and at the faster velocity it was necessary to decrease $\epsilon$, which means that the control effort was penalized less than for either the circle or 45°-line case. Better performance resulted with higher initial covariances.

In the first run the planar, no load arm was simulated with $\lambda = .99$ and cruise velocity 6.3 cm/s. Maximum error is 4 mm. No definite trends were evident when either $\lambda$ or $P(0)$ were varied (Fig. 35a).

In the next run load and gravity were re-introduced by the tip velocity was reduced. Maximum error decreased to 2 mm, with $\lambda = .9$, as seen in Fig. 35b. Thus velocity disturbances seem to predominate for this path.

Varying $\epsilon$

The pseudo-gain $\epsilon$ was the next parameter which was varied. Decreasing it by a factor of ten improved performance significantly. The desired and actual paths are practically indistinguishable. It was possible to adjust $\epsilon$ for this path because the required range of control signals $u(k)$ is small. Hence, there is no danger of prolonged periods of actuator saturation. Two runs are shown here. The first, Fig. 35c was for $\lambda = 0.99$ and the other, Fig. 35d, was for $\lambda = 0.85$. The lower forgetting factor results in a slightly jagged trajectory but absolute error is as good as for $\lambda = 0.99$. Varying $P(0)$ had no significant effect either.

The final run described was the result of an oversight but it shows that the adaptive controller is extremely robust and gives strong indications that it is both stable and convergent. The simulation data files were initialized for $\theta_2 = 60^\circ$ and $\theta_3 = -120^\circ$, which corresponds to an elbow
up configuration. However, the elbow specification for the trajectory was given as "down" (Fig. 34b). The resulting endpoint movement is shown in Fig. 35e. It indicates that the arm whips down, and then returns to follow the specified path to within 0.1 mm.
Table III. Discrete Model Parameters and Coefficients for the Known Case

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Sampling Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.010 s</td>
</tr>
<tr>
<td>a₁</td>
<td>-1.8339</td>
</tr>
<tr>
<td>a₂</td>
<td>0.8339</td>
</tr>
<tr>
<td>b₁</td>
<td>4.736x10⁻⁵</td>
</tr>
<tr>
<td>b₂</td>
<td>4.458x10⁻⁵</td>
</tr>
<tr>
<td>c₁</td>
<td>2.5294</td>
</tr>
<tr>
<td>c₂</td>
<td>-1.5294</td>
</tr>
<tr>
<td>c₃</td>
<td>1.314x10⁻⁴</td>
</tr>
<tr>
<td>c₄</td>
<td>8.176x10⁻⁴</td>
</tr>
</tbody>
</table>

\[ \frac{K}{J} = 1.0055 \]
\[ \frac{C}{J} = 18.1589 \]
Table IV. Test Paths for the Single-Link Simulation

<table>
<thead>
<tr>
<th>Name</th>
<th>Time Interval [s]</th>
<th>Reference Input Function $y_d(t)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>[rad]</td>
</tr>
<tr>
<td>STEP 1</td>
<td>$0 &lt; t &lt; 5$</td>
<td>0.15</td>
</tr>
<tr>
<td>STEP 2</td>
<td>$0 &lt; t &lt; 5$</td>
<td>0.50</td>
</tr>
<tr>
<td>RAMP 1</td>
<td>$0 &lt; t &lt; 5$</td>
<td>$0.10t$</td>
</tr>
<tr>
<td>RAMP 2</td>
<td>$0 &lt; t &lt; 5$</td>
<td>$0.40t$</td>
</tr>
<tr>
<td>TRAPEZOID</td>
<td>$0 &lt; t &lt; 1.6$</td>
<td>$0.125t^2$</td>
</tr>
<tr>
<td></td>
<td>$1.6 &lt; t &lt; 3.6$</td>
<td>$0.32 + 0.4(t-1.6)$</td>
</tr>
<tr>
<td></td>
<td>$3.6 &lt; t &lt; 5.2$</td>
<td>$1.12 + 0.4(t-3.6)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$-0.125(t-3.6)^2$</td>
</tr>
<tr>
<td></td>
<td>$5.2 &lt; t &lt; 7.5$</td>
<td>1.44</td>
</tr>
</tbody>
</table>

Note: In all cases $y(t=0) = 0$
Table V. Run Parameters for the Single Degree of Freedom Simulation

<table>
<thead>
<tr>
<th>Run No.</th>
<th>$T_{samp}$</th>
<th>$\epsilon$</th>
<th>Path</th>
<th>D/K</th>
</tr>
</thead>
<tbody>
<tr>
<td>2,3,4</td>
<td>0.010</td>
<td>variable</td>
<td>STEP 1</td>
<td>0</td>
</tr>
<tr>
<td>5,6,7,13</td>
<td>0.020</td>
<td>variable</td>
<td>STEP 1</td>
<td>0</td>
</tr>
<tr>
<td>8,9</td>
<td>0.020</td>
<td>crit. damped</td>
<td>STEP 1</td>
<td>.859</td>
</tr>
<tr>
<td>10</td>
<td>0.010</td>
<td>crit. damped</td>
<td>STEP 1</td>
<td>.859</td>
</tr>
<tr>
<td>11</td>
<td>0.020</td>
<td>crit. damped</td>
<td>STEP 1</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>0.010</td>
<td>crit. damped</td>
<td>STEP 1</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>0.020</td>
<td>crit. damped</td>
<td>STEP 2</td>
<td>.859</td>
</tr>
<tr>
<td>15</td>
<td>0.020</td>
<td>crit. damped</td>
<td>RAMP 1</td>
<td>.859</td>
</tr>
<tr>
<td>16</td>
<td>0.020</td>
<td>crit. damped</td>
<td>RAMP 2</td>
<td>.859</td>
</tr>
<tr>
<td>17</td>
<td>0.020</td>
<td>crit. damped</td>
<td>TZOID</td>
<td>.859</td>
</tr>
</tbody>
</table>
Table VI. Circle Trajectory - Parameters of the Simulation Runs

<table>
<thead>
<tr>
<th>Run No.</th>
<th>( g ) ( [\text{m-s}^{-1}] )</th>
<th>( m_1 ) ( [\text{kg}] )</th>
<th>( \lambda_2 )</th>
<th>( \lambda_3 )</th>
<th>( P_2(0) )</th>
<th>( P_3(0) )</th>
<th>Initial Estimates</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>0</td>
<td>0</td>
<td>.9</td>
<td>.9</td>
<td>100</td>
<td>1,000</td>
<td>non-zero</td>
</tr>
<tr>
<td>22</td>
<td>9.81</td>
<td>0</td>
<td>.9</td>
<td>.9</td>
<td>100</td>
<td>1,000</td>
<td>non-zero</td>
</tr>
<tr>
<td>23</td>
<td>9.81</td>
<td>2.046</td>
<td>.9</td>
<td>.9</td>
<td>100</td>
<td>1,000</td>
<td>non-zero</td>
</tr>
<tr>
<td>28</td>
<td>9.81</td>
<td>2.046</td>
<td>.5</td>
<td>.5</td>
<td>1,000</td>
<td>1,000</td>
<td>non-zero</td>
</tr>
<tr>
<td>29</td>
<td>9.81</td>
<td>2.046</td>
<td>.995</td>
<td>.995</td>
<td>1,000</td>
<td>1,000</td>
<td>non-zero</td>
</tr>
<tr>
<td>30</td>
<td>9.81</td>
<td>2.046</td>
<td>.95</td>
<td>.95</td>
<td>100,000</td>
<td>100,000</td>
<td>0</td>
</tr>
<tr>
<td>31</td>
<td>9.81</td>
<td>2.046</td>
<td>.95</td>
<td>.95</td>
<td>100</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>32</td>
<td>9.81</td>
<td>2.046</td>
<td>.95</td>
<td>.95</td>
<td>100</td>
<td>100</td>
<td>0</td>
</tr>
</tbody>
</table>

In all runs:

gain:
- \( \epsilon_2 = .12 \times 10^{-5} \)
- \( \epsilon_3 = .18 \times 10^{-5} \)

initial velocity
- \( y_2 = 0 \)
- \( y_3 = 0 \)

initial position
- \( y_2 = 0 \)
- \( y_3 = -45^\circ \)

desired velocity
- \( y_2 = 0.42 \text{ rad/s} \)
- \( y_3 = 0 \)

time for one revolution = 15 s

sampling time \( t_{\text{samp}} = 0.020 \text{ s} \)

integration step \( t_{\text{step}} = 0.001 \text{ s} \)
### Table VII. Line at 45° – Parameters of the Simulation Runs

<table>
<thead>
<tr>
<th>Run No.</th>
<th>g [m/s&lt;sup&gt;2&lt;/sup&gt;]</th>
<th>m&lt;sub&gt;1&lt;/sub&gt; [kg]</th>
<th>P&lt;sub&gt;2&lt;/sub&gt;(0)</th>
<th>P&lt;sub&gt;3&lt;/sub&gt;(0)</th>
<th>Initial Estimates</th>
<th>a [m/s&lt;sup&gt;-2&lt;/sup&gt;]</th>
<th>v&lt;sub&gt;c&lt;/sub&gt; [m/s]</th>
<th>t&lt;sub&gt;1&lt;/sub&gt; t&lt;sub&gt;2&lt;/sub&gt; t&lt;sub&gt;3&lt;/sub&gt;</th>
<th>Elbow</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>9.81</td>
<td>2.046</td>
<td>100</td>
<td>1,000</td>
<td>non-zero</td>
<td>.025</td>
<td>.05</td>
<td>2 8 10</td>
<td>down</td>
</tr>
<tr>
<td>25</td>
<td>9.81</td>
<td>2.046</td>
<td>100</td>
<td>1,000</td>
<td>non-zero</td>
<td>.05</td>
<td>.05</td>
<td>1 6 7</td>
<td>down</td>
</tr>
<tr>
<td>26</td>
<td>9.81</td>
<td>2.046</td>
<td>100</td>
<td>1,000</td>
<td>non-zero</td>
<td>.05</td>
<td>.05</td>
<td>1 6 7</td>
<td>up</td>
</tr>
<tr>
<td>33</td>
<td>9.81</td>
<td>2.046</td>
<td>10,000</td>
<td>10,000</td>
<td>non-zero</td>
<td>.05</td>
<td>.15</td>
<td>3 4 7</td>
<td>down</td>
</tr>
<tr>
<td>34</td>
<td>0</td>
<td>0</td>
<td>10,000</td>
<td>10,000</td>
<td>non-zero</td>
<td>.05</td>
<td>.15</td>
<td>3 4 7</td>
<td>down</td>
</tr>
<tr>
<td>35</td>
<td>9.81</td>
<td>2.046</td>
<td>100,000</td>
<td>100,000</td>
<td>0</td>
<td>.05</td>
<td>.15</td>
<td>3 4 7</td>
<td>down</td>
</tr>
</tbody>
</table>

In all runs:

- **Gain**
  \[ \varepsilon_2 = 0.12 \times 10^{-5} \]
  \[ \varepsilon_3 = 0.18 \times 10^{-5} \]

- **Forgetting Factor**
  \[ \lambda_2 = 0.9 \]
  \[ \lambda_3 = 0.9 \]

- **Initial Velocity**
  \[ y_2 = 0 \]
  \[ y_3 = 0 \]

- **Initial Position**
  \[ y_2 = 0 \]
  \[ y_3 = -90\degree \]

- **Desired Velocity**
  \[ y_2 = 0.020 \text{ s} \]
  \[ y_3 = 0.001 \text{ s} \]

- **Sampling Time**
  \[ t_{\text{samp}} = 0.020 \text{ s} \]

- **Integration Step**
  \[ t_{\text{step}} = 0.001 \text{ s} \]

- **Direction Cosines**
  \[ \alpha = 180\degree \]
  \[ \beta = 90\degree \]
Table VIII. Horizontal Line – Parameters of the Simulation Runs

<table>
<thead>
<tr>
<th>Run No.</th>
<th>( g ) [m-s(^{-1})]</th>
<th>( m_1 ) [kg]</th>
<th>( p_{2,3}(0) )</th>
<th>( \varepsilon_2 )</th>
<th>( \varepsilon_3 )</th>
<th>elbow</th>
<th>( v_c ) [m-s(^{-1})]</th>
<th>( t_1 )</th>
<th>( t_2 )</th>
<th>( t_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>27</td>
<td>0</td>
<td>0</td>
<td>1,000</td>
<td>0.12</td>
<td>0.18</td>
<td>up</td>
<td>0.063</td>
<td>1</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>36</td>
<td>9.81</td>
<td>2.046</td>
<td>1,000</td>
<td>0.12</td>
<td>0.18</td>
<td>up</td>
<td>0.05</td>
<td>2</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>37</td>
<td>0</td>
<td>0</td>
<td>10,000</td>
<td>0.012</td>
<td>0.012</td>
<td>down</td>
<td>0.05</td>
<td>2</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>39</td>
<td>9.81</td>
<td>2.046</td>
<td>10,000</td>
<td>0.012</td>
<td>0.012</td>
<td>up</td>
<td>0.15</td>
<td>2</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>40</td>
<td>9.81</td>
<td>2.046</td>
<td>10,000</td>
<td>0.012</td>
<td>0.012</td>
<td>up</td>
<td>0.15</td>
<td>2</td>
<td>6</td>
<td>8</td>
</tr>
</tbody>
</table>

In all runs:

- initial velocity: \( y_2 = 0 \), \( y_3 = 0 \)
- initial position: \( y_2 = 60^\circ \), \( y_3 = -120^\circ \)
- direction cosines: \( \alpha = 180^\circ \), \( \beta = 90^\circ \)
- sampling time: \( t_{samp} = 0.020 \) s
- integration step: \( t_{step} = 0.001 \) s
- acceleration: \( a = 0.025 \) m/s\(^2\)
Fig. 15 SAMPLED DATA SYSTEM
a) sampling a continuous time system
b) equivalent discrete representation
Fig. 16  PROGRAM 'P' FLOWCHART.
FIG. 17  PROGRAM 'Q' FLOWCHART
FIG. 18a  Known Control - run 2
FIG. 18b  Known Control - run 3
FIG. 18c  Known Control - run 4
FIG. 19a  Known control  $T=0.020$ s
$\epsilon=0.20E-05$

**Diagram:**
- **Position (radians):**
  - $0.20$
  - $0.15$
  - $0.10$
  - $0.05$
  - $0.00$
  - $-0.05$
  - $0$
  - $1$
  - $2$
  - $3$
  - $4$
  - $5$

- **Control Effort (volts):**
  - $12$
  - $8$
  - $4$
  - $0$
  - $-4$
  - $-8$
  - $-12$
  - $0$
  - $1$
  - $2$
  - $3$
  - $4$
  - $5$
FIG. 19b  Known Control - run 6
FIG. 19c  Known Control - run 7
FIG. 19d  Known Control - run 13
FIG. 20  Adaptive Control - run 8
FIG. 20c  Known Control - run 9
FIG. 22  ADAPTIVE   D = 0
FIG. 23  Known vs Adaptive Control - run 14
FIG.24a  Known vs Adaptive Control - run 15
FIG. 24b  Known vs Adaptive Control - run 16
FIG. 25 Known vs Adaptive Control - run 17
FIG. 26a Adaptive Control
zero initial estimates, \( P=100,000 \)
FIG. 26c  Adaptive Control
initial estimates zero, $P=10000$
Fig. 27 Program 'R' Flowchart
FIG. 28  CIRCLE
FIG. 29b  Endpoint Trace - run 21
FIG. 29f  circle - joint 3

FIG. 29c  circle - joint 3
CONTROL EFFORT (volts)

TIME (sec)

Fig. 29d

Circle - Joint 2
FIG. 29e    Endpoint Trace - run 22
FIG. 29g
circle - joint 3

POSITION (radians)

TIME (sec)

-0.786
0
3
6
9
12
15

CONTROL EFFORT (volts)

TIME (sec)

-12
-8
-4
0
4
8
12

circle - joint 2
FIG. 29h  Endpoint Trace - run 23
FIG. 30b  circle - joint 2
FIG. 31a  circle - joint 3

FIG. 31b  circle - joint 2
FIG. 31C  Endpoint Trace - run 31
FIG. 31d  circle — joint 2

FIG. 31d  circle — joint 3
FIG. 32  45° line
FIG. 33a  Endpoint Trace - run 24
45 degree line - joint 2

FIG. 33b  line at 45 deg. - joint 3
FIG. 33c  Endpoint Trace - run 25
FIG. 33d  45 degree line - joint 3
FIG. 33e  Endpoint Trace - run 26
FIG. 33f  Endpoint Trace - run 33
FIG. 33g  45 degree line – joint 2

FIG. 33g  45 degree line – joint 3
FIG. 33h   Endpoint Trace - run 34
FIG. 33i  Endpoint Trace - run 35
FIG. 34  HORIZONTAL LINE
FIG. 35a  Endpoint Trace - run 27
FIG. 35b  Endpoint Trace - run 36
FIG. 35d    Endpoint Trace - run 40
FIG. 35e  Endpoint Trace - run 37
5.1 Conclusions

The following conclusions may be drawn:

i) A computer simulation package of a robot system has been developed which will enable a user to test various control strategies. The model is based on an existing robot. Static friction, backlash, and link flexibility are not modelled. Features of the simulator include:

(a) an elementary trajectory planner
(b) an inverse kinematics subroutine
(c) direct kinematics for endpoint movement monitoring
(d) the manipulator and actuator dynamics
(e) a controller, at present adaptive
(f) input files for varying dynamical parameters and initial conditions.

The simulation package has been structured for three links with load mass. It could be used as a test bed for evaluating different control laws.

ii) An explicit adaptive controller was derived from first principles. The controller is based on linear modelling of each link using an ARMA model, recursive least squares parameter estimation, and optimization of a quadratic performance index. In contrast to Koivo's [13] velocity controller, in which approximate compensation for position error is necessary, this work uses position control directly, such that these compensations are avoided.
iii) The control law is tested using the simulation package for two degrees-of-freedom and motion along three paths. Results indicate that the controller adapts and that it is stable and convergent. The control effort is well behaved, exhibiting no chatter or high frequency components.

iv) The results indicate that the type of control law proposed here could be a feasible approach to improved robot control.

5.2 Recommendations for Future Work

These may be divided into two groups: first, possible elaboration on the existing simulator; and second, extensions requiring more theoretical development.

i) Very little effort would be needed to extend the simulator to three degrees-of-freedom, consisting only of (1) obtaining and programming the equations of motion in the form of a subroutine callable by the integration package and (2) verifying the quadrant specifications for the inverse kinematics subroutine.

ii) A sensitivity analysis could be done, keeping the same control law but modifying the dynamical parameters of the robot.

iii) The simulation runs done in this work involved parameter estimation at each control step. An analysis could be done on the effect of estimating less often than is controlled. If performance does not degrade, this set-up is advantageous since the computational burden is reduced.
iv) An analysis of the number of computations necessary to implement the proposed control law was not carried out, and hence comparisons with other laws is not possible. As far as feasibility is concerned, computational burden is an important aspect, although advances in computer technology are making this less critical every year.

v) Since the offset or bias term does not seem to be effectively eliminated by the parameter h in all cases, the disturbances may need to be modelled in a different way, for example as filtered white noise

\[ v(k) = C(q^{-1}) e(k) \]

This approach would lead to parameter estimation using extended least squares, for which convergence results are available [18].

vi) The control law which results by optimizing the performance criterion

\[ I = \{(y_d(k+1) - y(k+1))^2 + \epsilon u^2(k)|(k-1)\} \]

does not contain any integrating terms. In practice, feedback must include an integrator, otherwise unacceptably large steady state errors result due to unmodelled effects such as friction. Adding an integrating term can be achieved by choosing a modified I containing a term of the form \( \gamma(u(k) - u(k-1))^2 \). Of course, complexity will increase and so will the number of calculations necessary to arrive at \( u(k) \).

vii) As an alternative to adding more terms to the performance criterion, a different structure of adaptive control could be examined. Instead of using adaptive control to control each joint directly the adaptation
would take place one hierarchical level higher. Classical analogue PID servo loops at each joint, after all, perform quite well once they are tuned to the operational characteristics of the particular task. Their limitation is that as the configurations of the task change performance degrades because the loop gains are no longer optimum. One can envisage a supervisory adaptive controller which would alter the gains of the feedback loops as operating conditions change. The problem, however, would have to be cast in a different form theoretically than is presented in this work.

viii) Extension to endpoint control is perhaps the most interesting aspect, but a multivariable approach is necessary. In other words the polynomials $A(q^{-1})$ and $B(q^{-1})$ are replaced by matrix polynomials. This approach is especially amenable to robots with spherical wrists since the inverse kinematics can be split at the wrist, so that (3×3) systems can be studied.


A derivation of the standard recursive least squares algorithm is included here for the sake of completeness. The least squares estimate is derived first; then a recursive form of the solution is derived.

Consider a system model defined by the equation

\[ y(k) = \theta^T \psi(k) + v(k) \]

where \( y(k) \) is the computed variable, \( v(k) \) is unknown noise, \( \theta \) is the vector of parameters to be estimated, \( \theta^T = [\theta_1 \theta_2 \ldots \theta_n] \), and \( \psi \) is a vector of measurements.

If an estimate of \( y(k) \) is denoted by \( \hat{y}(k) \) then the error is

\[ \epsilon(k) = y(k) - \hat{y}(k) \]

The principle of least squares says that the parameter vector \( \theta \) should be chosen in such a way as to minimize the criterion function

\[ I(\theta) = \sum_{k=1}^{N} \epsilon(k)^2 \]  \hspace{1cm} (1)
where $N$ is the number of measurements taken and $\omega(k)$ is a positive number allowing different weights to be attached to different observations.

Introducing the vectors

$$ Y^T(N) = [y_1 \ y_2 \ \cdots \ y_N] $$

$$ \varepsilon^T(N) = [\varepsilon_1 \ \varepsilon_2 \ \cdots \ \varepsilon_N] $$

and the matrix

$$ 
\phi(N) = \begin{bmatrix}
\phi^T(1) \\
\phi^T(2) \\
\vdots \\
\phi^T(N)
\end{bmatrix}
$$

and dropping the $N$ for the time being, the criterion function may be re-written

$$ I(\theta) = \varepsilon^T \varepsilon $$

(2)

Substituting

$$ \varepsilon = Y - \overline{Y} \ , \ \overline{Y} = \phi \theta $$

into (2) gives

$$ I(\theta) = (Y - \phi \theta)^T (Y - \phi \theta) $$

$$ = Y^T_Y - (\phi \theta)^T_Y - Y^{T} \phi \theta + (\phi \theta)^T \phi \theta $$

$$ = Y^T_Y - 2Y^{T} \phi \theta + \theta^{T} \phi^{T} \phi \theta $$
Performing the minimization with respect to \( \theta \) gives

\[
\frac{\partial L}{\partial \theta} = 0 - 2 \phi^T \Psi + 2 \phi^T \phi \theta
\]

Thus the best estimate \( \hat{\theta} \) in the least squares sense is given by

\[
\hat{\theta} = (\phi^T \phi)^{-1} \phi^T \Psi
\]

provided the inverse exists.

A recursive form of (3) is now found in order to include the \((N+1)^{\text{th}}\) observation in the estimate \( \hat{\theta} \). The argument \( N \) is reintroduced in the notation at this point. When a new measurement is obtained, the matrix \( \phi \) and the vector \( \Psi \) will be

\[
\phi^{(N+1)} = \begin{bmatrix} \phi^{(N)} \\ \phi^T \end{bmatrix}, \quad \Psi^{(N+1)} = \begin{bmatrix} \Psi^{(N)} \\ y^{(N+1)} \end{bmatrix}
\]

The estimate \( \hat{\theta}^{(N+1)} \) is, from (1)

\[
\hat{\theta}^{(N+1)} = [\phi^{T(N+1)} \phi^{(N+1)}]^{-1} \phi^{T(N+1)} \Psi^{(N+1)}
\]

noting that

\[
\phi^{T(N)} \phi^{(N)} = \sum_{k=1}^{N} \phi^{(k)} \phi^{T(k)}
\]

The above Equation (4) may be re-written
\[ \hat{\theta}(N+1) = [\phi^T(N) \phi(N) + \phi(N+1) \psi^T(N+1)]^{-1} \]
\[ \times [\phi^T(N) \mathbf{Y}(N) + \psi^T(N+1) \mathbf{y}(N+1)] \]
\[ = [\phi^T \phi + \psi \psi^T]^{-1} [\phi^T \mathbf{Y} + \psi \mathbf{y}] \] (5)

where the arguments have again been suppressed for convenience. Rewriting,

\[ \hat{\theta}(N+1) = (\phi^T \phi)^{-1} \phi^T \mathbf{Y} + [(\phi^T \phi + \psi \psi^T)^{-1} - (\phi^T \phi)^{-1}] \phi^T \mathbf{Y} \]
\[ + (\phi^T \phi + \psi \psi^T)^{-1} \psi \mathbf{y} \]

Recalling that

\[ \hat{\theta}(N) = (\phi^T \phi)^{-1} \phi^T \mathbf{Y} \]

and observing that

\[ [(\phi^T \phi + \psi \psi^T)^{-1} - (\phi^T \phi)^{-1}] \phi^T \mathbf{Y} \]
\[ = (\phi^T \phi + \psi \psi^T)^{-1} (\phi^T \phi - \phi^T \phi - \psi \psi^T)(\phi^T \phi)^{-1} \phi^T \mathbf{Y} \]
\[ = - (\phi^T \phi + \psi \psi^T)^{-1} (\phi^T \phi - \phi^T \phi - \psi \psi^T)(\phi^T \phi)^{-1} \phi^T \mathbf{Y} \]
\[ = - (\phi^T \phi + \psi \psi^T)^{-1} \psi \psi^T \hat{\theta} \]

Then (5) may be written
\[ \hat{\theta}(N+1) = \hat{\theta}(N) + K(N) \{ y(N+1) - \psi^T(N+1) \hat{\theta}(N) \} \]  

(6)

where

\[ K(N) = \left[ \phi^T(N+1) \phi(N+1) \right]^{-1} \psi(N+1) \]

In order to obtain a recursive form for \( K(N) \), the matrix \( P(N) \) is introduced

\[ P(N) = \left[ \phi^T(N) \phi(N) \right]^{-1} \]

(7)

Then

\[ P(N+1) = \left[ \phi^T \phi + \psi \psi^T \right]^{-1} \]

At this point the "matrix inversion lemma" is used.

**Lemma.** Let \( A, C, \) and \( (C^{-1} + DA^{-1}B) \) be non-singular square matrices. Then

\[ [A + BCD]^{-1} = A^{-1} - A^{-1}B \left[ C^{-1} + DA^{-1}B \right]^{-1} DA^{-1} \]

Taking

\[ A = \phi^T \phi, \quad B = \psi, \quad C = I, \quad D = \psi^T \]

\( P(N+1) \) may be written as
\[ P(N+1) = (\phi^T \phi)^{-1} - \phi^T \phi \psi [1 + P(N) \psi^T P(N) \psi -1] P(N) \psi^T \phi \psi^{-1} \]

\[ = P(N) - P(N) \psi [1 + \psi^T P(N) \psi -1] \psi^T P(N) \]

(8)

So,

\[ K(N) = P(N+1) \psi(N+1) \]

\[ = P(N) \psi(N+1) - \frac{P(N) \psi^T P(N) \psi(N+1)}{[1 + \psi^T P(N) \psi]} \]

\[ K(N) = P(N) \psi(N+1) [1 + \psi^T P(N) \psi]^{-1} \]

(9)

and,

\[ P(N+1) = P(N) - K(N) \psi^T(N+1) P(N) \]

\[ P(N+1) = [I - K(N) \psi^T(N+1)] P(N) \]

(10)

The recursive least squares estimation consists of Equations (6), (9) and (10).

A weighting factor may be included in the criterion function to account for time-varying parameters

\[ I(\theta) = \sum_{k=1}^{N} \omega(k) \phi(k)^2 \]

The weighting function chosen for this study was the common exponential forgetting factor
\[ \omega(k) = \lambda^{N-k} , \quad 0 < \lambda < 1 \]

In that case the recursive least squares equations become

\[
\hat{\theta}(k) = \hat{\theta}(k+1) + K(K-1) [y(k) - \psi^T(k) \hat{\theta}(k-1)]
\]

\[
K(K-1) = P(k-1) \psi(k) [\lambda + \psi^T(k) P(k-1) \psi(k)]^{-1}
\]

\[
P(k) = [I - K(k-1) \psi^T(k)] P(k-1) / \lambda
\]

The reader is referred to Lying and Soderstrom [18] who show how these modified equations are obtained.
APPENDIX II

THE A AND T MATRICES

\( \cos(\theta_1) \) is denoted \( C_1 \) and \( \sin(\theta_1) \) is denoted \( S_1 \).

Substituting values from Table II into the general expression \( A_1 \) given in section 3.2.1 gives

\[
A_1 = \begin{bmatrix}
C_1 & 0 & S_1 & 0 \\
S_1 & 0 & -C_1 & 0 \\
0 & 1 & 0 & d_1 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

\[
A_2 = \begin{bmatrix}
C_2 & -S_2 & 0 & a_2C_2 \\
S_2 & C_2 & 0 & a_2S_2 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

\[
A_3 = \begin{bmatrix}
C_3 & 0 & -S_3 & 0 \\
S_3 & 0 & C_3 & 0 \\
0 & -1 & 0 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

The transformation matrices \( T_i \), relating vectors expressed in coordinate frame i to base coordinates are easily found by multiplication

\[
T_1 = A_1
\]

\[
T_2 = \begin{bmatrix}
C_1C_2 & -C_1S_2 & S_1 & a_2C_1C_2 \\
S_1C_2 & -S_1S_2 & -C_1 & u_2S_1C_2 \\
S_2 & C_2 & 0 & a_2S_2+d_1 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]
These matrices are used to obtain the equations of motion using the Lagrangian approach.
APPENDIX III

THE PSEUDO-INERTIA MATRIX $J_i$

From equation (3.32), the pseudo-inertia matrix $J_i$ was defined

$$J_i = \int_{\text{link } i} \mathbf{x}_i \mathbf{x}_i^T \, dm$$

Expanding gives

$$J_i = \begin{bmatrix}
\int x^2 \, dm & \int x y \, dm & \int x z \, dm & \int x \, dm \\
\int y x \, dm & \int y^2 \, dm & \int y z \, dm & \int y \, dm \\
\int z x \, dm & \int z y \, dm & \int z^2 \, dm & \int z \, dm \\
\int \, dm & \int \, dm & \int \, dm & \int \, dm 
\end{bmatrix}$$

Defining

$$I_{xx} = \int (y^2 + z^2) \, dm$$

$$I_{yy} = \int (x^2 + z^2) \, dm$$

$$I_{zz} = \int (x^2 + y^2) \, dm$$

$$I_{xy} = \int xy \, dm$$

$$I_{xz} = \int xz \, dm$$

$$I_{yz} = \int yz \, dm$$
_mx = \int x \, dm

_my = \int y \, dm

_mz = \int z \, dm

Then it is possible to express the diagonal terms as

\[ I_x = \int (x^2) \, dm = \frac{-I_{xx} + I_{yy} + I_{zz}}{2} \]

\[ I_y = \int (y^2) \, dm = \frac{I_{xx} - I_{yy} + I_{zz}}{2} \]

\[ I_z = \int (z^2) \, dm = \frac{I_{xx} + I_{yy} - I_{zz}}{2} \]

and \( J_i \) may be written more concisely as

\[
J_i = \begin{pmatrix}
I_{ix} & I_{ixy} & I_{ixz} & m_i x_i \\
I_{ixy} & I_{iy} & I_{iyz} & m_i y_i \\
I_{ixz} & I_{iyz} & I_{iz} & m_i z_i \\
m_i x_i & m_i y_i & m_i z_i & m_i
\end{pmatrix}
\]
APPENDIX IV
DETERMINATION OF THE LOAD TERM $L_1$

The load term $L_1$ may account for the wrist assembly or for an
arbitrary mass placed at the end of the third link of the RSI robot. The
Lagrangian approach is applied in much the same way as it is for the
manipulator itself, the applicable equation being

$$L_1 = \frac{d}{dt} \left( \sum K \right) - \sum K_{\theta_i} + \sum P_{\theta_i}$$

The potential energy is found first. The position of the mass is described
by

$$3_{\rho_1}^T = [0 \ 0 \ z_1 \ 1]$$

or, in link 1 coordinates, a fixed frame in the simulation, by

$$1_{\rho_1} = A_2A_3^3_{\rho_1}$$

Using results from Appendix II and performing the multiplication gives

$$1_{\rho_1} = \begin{bmatrix}
-z_1 \sin(\theta_2 + \theta_3) + a_2 \cos \theta_2 \\
-z_1 \cos(\theta_2 + \theta_3) + a_2 \cos \theta_2 \\
0 \\
1
\end{bmatrix}$$
The gravity vector in link 1 coordinates is given by

\[ \mathbf{g}^T = [0 \ g \ 0 \ 1] \]

The potential energy \( P \) is then

\[ P = m_1 \mathbf{g}^T \mathbf{p}_1 \]

which gives

\[ P = m_1 g[a_2 \sin \theta_2 + \bar{z}_1 \cos(\theta_2 + \theta_3)] \]

Next, the kinetic energy is found. In base coordinates the position of the mass is

\[ \rho_1 = T_3 \mathbf{p}_1 \]

Then its velocity is

\[ \frac{d\rho_1}{dt} = \sum_{i=1}^{3} \frac{\partial T_3}{\partial \theta_i} \theta_i \mathbf{p}_1 \]

Since link 1 is locked in position, only the last two terms contribute. The square of the velocity is
\( \frac{d\rho_1}{dt} \) = Trace \( (\rho_1 \rho_1^T) \)

and the kinetic energy is

\[
K = \frac{1}{2} m_1 \left( \frac{d\rho_1}{dt} \right)^2
\]

giving the result

\[
K = \frac{1}{2} m_1 \left[ \dot{\theta}_2^2 \left( a_2^2 - 2a_2 z_1 \sin \theta_3 + \bar{z}_1^2 \right) + \bar{z}_1^2 \dot{\theta}_3^2 + 2(z_1^2 - a_2 z_1 \sin \theta_3) \dot{\theta}_2 \dot{\theta}_3 \right]
\]

Substituting \( K \) and \( P \) into Equation (3.25) and performing the required operations gives \( \mathcal{L}_2 \) and \( \mathcal{L}_3 \). The result is given in Equations (3.42) and (3.43).
APPENDIX V

J₁ FOR THE RSI ROBOT

From Appendix III, the general expression for J₁ is

\[
J₁ = \begin{bmatrix}
I_{ix} & I_{ixy} & I_{ixz} & m₁x₁ \\
I_{ixy} & I_{iy} & I_{iyz} & m₁y₁ \\
I_{ixz} & I_{iyz} & I_{iz} & m₁z₁ \\
m₁x₁ & m₁y₁ & m₁z₁ & m₁
\end{bmatrix}
\]

Referring to Fig. 9, it is assumed that the centre of mass of each link lies along one of the link axes, so that

\[
\rho₁^T = [0 \ -y₁ \ 0 \ 1]
\]

\[
\rho₂^T = [\bar{x}₂ \ 0 \ 0 \ 1]
\]

\[
\rho₃^T = [0 \ 0 \ \bar{z}₃ \ 1]
\]

Then the pseudo-inertia matrices Jᵢ for i = 1, 2, 3 are

\[
J₁ = \begin{bmatrix}
I_{1x} & 0 & 0 & 0 \\
0 & I_{1y} & 0 & m₁y₁ \\
0 & 0 & I_{1z} & 0 \\
m₁y₁ & 0 & 0 & m₁
\end{bmatrix}
\]
\[ J_2 = \begin{bmatrix}
I_{2x} & 0 & 0 & m_2 z_2 \\
0 & I_{2z} & 0 & 0 \\
0 & 0 & I_{3z} & 0 \\
m_2 z_2 & 0 & 0 & m_2
\end{bmatrix} \]

\[ J_3 = \begin{bmatrix}
I_{3x} & 0 & 0 & 0 \\
0 & I_{3y} & 0 & 0 \\
0 & 0 & I_{3z} & m_3 z_3 \\
0 & 0 & m_3 z_3 & m_3
\end{bmatrix} \]
APPENDIX VI

SETTING UP THE INTEGRATION SUBROUTINE

The full equations of motion are given in section 3.3.3. Identifying groups of terms to simplify the notation, we let

\[ T_2 = \left( \frac{nK_T}{R} \right)_2 \]
\[ T_3 = \left( \frac{nK_T}{R} \right)_3 \]
\[ C_2 = \left( C_\ell \frac{n^2K_6K_T}{R} \right)_2 \]
\[ C_3 = \left( C_\ell \frac{n^2K_6K_T}{R} \right)_3 \]
\[ F_1 = g[a_2(m_2 + m_3) + m_2\ddot{x}_2] \]
\[ F_2 = g m_3 \ddot{z}_3 \]
\[ F_3 = I_{2zz} + n^2J_{\text{act}2} + a_2^2m_2 + a_2m_3 + 2a_2m_2\ddot{x}_2 + I_{3yy} \]
\[ F_4 = m_3 \ddot{z}_3a_2 \]
\[ F_5 = I_{3yy} \]
\[ F_6 = I_{3yy} + n^2J_{\text{act}3} \]
\[ L_1 = g_m z_2 \]
\[ L_2 = g_m z_2 \]
\[ L_3 = (a_2^2 + z_2^2) m_g \]
\[ L_4 = m_g z_2 a_2 \]
\[ L_5 = m_g z_2^2 \]

Using the variables defined above we let

\[ A = D_2 + L_2 = (F_1 + L_1) \cos \theta_2 - (F_2 + L_2) \sin(\theta_2 + \theta_3) \]
\[ B = D_{22} + L_{22} = L_3 - L_3 - 2(F_4 + L_4) \sin \theta_3 \]
\[ C = D_{23} + L_{23} = D_{32} + L_{32} = (F_5 + L_5) - (F_4 + L_4) \sin \theta_3 \]
\[ E = -(D_{223} + L_{223}) = D_{322} + L_{322} = -(D_{233} + L_{233}) = (F_4 + L_4) \cos \theta_3 \]
Then the equations of motion may be re-written

\[ T_2 u_2 - C_2 \dot{\theta}_2 = A + B \ddot{\theta}_2 + C \dot{\theta}_3 - 2E \dot{\theta}_2 \dot{\theta}_3 - E \theta_3^2 \]

\[ T_3 u_3 - C_3 \dot{\theta}_3 = P + Q \ddot{\theta}_3 + C \theta_2 + E \theta_2^2 \]

Collecting \( \ddot{\theta}_2 \) and \( \ddot{\theta}_3 \) to one side

\[ \ddot{\theta}_3 = \frac{1}{C} [T_2 u_2 - C_2 \dot{\theta}_2 - A + 2E \dot{\theta}_2 \dot{\theta}_3 + E \theta_3^2] - \frac{B}{C} \ddot{\theta}_2 \]

\[ \ddot{\theta}_2 = \frac{1}{C} [T_3 u_3 - C_3 \dot{\theta}_3 - P - E \theta_2^2] - \frac{Q}{C} \ddot{\theta}_3 \]

These second-order equations are transformed to a set of first-order equations by letting

\[ x_1 = \dot{\theta}_2 \]
\[ x_2 = \dot{\theta}_3 \]
\[ x_3 = \theta_2 \]
\[ x_4 = \theta_3 \]
so we obtain

\[ \begin{align*}
  \dot{x}_1 &= \ddot{\theta}_2 \\
  \dot{x}_2 &= \ddot{\theta}_3 \\
  \dot{x}_3 &= x_1 \\
  \dot{x}_4 &= x_2
\end{align*} \]

Finally, substituting for \( \ddot{\theta}_2 \) and \( \ddot{\theta}_3 \) we obtain

\[ \begin{align*}
  \dot{x}_1 &= \left[ \frac{C}{C^2 - QB} \right] [T_3 u_3 - C_3 x_3 - P - \text{Ex}_1^2] \\
  & \quad - \left[ \frac{C}{C^2 - QB} \right] [T_2 u_2 - C_2 x_1 - A + 2\text{Ex}_1 x_2 + \text{Ex}_2^2] \\
  \dot{x}_2 &= \left[ \frac{C}{C^2 - QB} \right] [T_2 u_2 - C_2 x_1 - A + 2\text{Ex}_1 x_2 + \text{Ex}_2^2] \\
  & \quad - \left[ \frac{B}{C^2 - QB} \right] [T_3 u_3 - C_3 x_2 - P - \text{Ex}_1^2] \\
  \dot{x}_3 &= x_1 \\
  \dot{x}_4 &= x_2
\end{align*} \]

This is the form of the equations integrated by the DDE subroutine.