Université d'Ottawa · University of Ottawa

Impedance Control of a Dual-Arm Robot

by

© Rahim Jassemi-Zargani

A dissertation submitted in partial fulfilment
of the requirements for the degree of
Doctor of Philosophy
in
Mechanical Engineering

Ottawa-Carleton Institute for
Mechanical and Aeronautical Engineering
(OCIMAE)
University of Ottawa
Ottawa, Ontario
Canada, K1N 5N6

Date: December 1997

0-612-28348-8

To my wife and daughters
Shirley, Sara and Sabrina

Abstract

Intensive research efforts are currently directed toward special robotic applications. One of the largest international projects with a focus on this research is the Space Station Freedom. Canada's contribution to this project is a Mobile Servicing System (MSS). This system will be used for assembly and maintenance of the space station. The MSS will consist of a long single arm with a dual-arm robot attached to its end. They require a hierarchical controller to coordinate the arms while executing such tasks as trajectory generation, arm-arm collision avoidance, arm-obstacle collision avoidance, and the manipulation of solid and flexible body payloads.

This thesis presents the research work for building a functional hardware and software model of a test-bed dual-arm robot with flexible object handling capabilities using impedance control. It addresses issues such as input-output linearization, perturbation observer, sensor fusion, and provides the experimental results of a dual-arm manipulator using the proposed techniques. To date, no work exists which shows that the application of a dual-arm robot handling flexible payloads in space applications is possible.

Acknowledgement

I wish to express my deep appreciation to all those who helped me in completing this research.

I am particularly grateful to my supervisor Doctor Dan Necsulescu, and my Ph.D. committee members Dr. Fahim and Dr. Sasiadek, for their guidance, reviews, criticisms, and insightful suggestions, throughout the development of this thesis.

I am also grateful to Mr. Bill Graham of System Software Engineering and Dr. Serdar Kalaycioglu of the Canadian Space Agency, for their guidance and financial support throughout my research.

Special thanks to: Kazuo Kiguchi who helped me in the design and building of the dual-arm robot, Peter Krueger, a visiting student from Germany, for his help in speeding up the process of building the dual-arm robot, and all the technicians in our shop for making the dual-arm robot a reality.

Thanks also to: Jean de Carufel and Bumsoo Kim, for their technical expertise, guidance, and moral support throughout my research.

I would also like to thank Wayne Baudais for taking the time from his busy schedule to proofread this thesis.

Finally, I would like particularly to thank my wife Shirley, whose continuous support and patience throughout my Ph.D. program have enabled me to produce this thesis. Also, I would like to thank my kids Sara and Sabrina for their patience and understanding and letting their daddy work on his thesis.

Table of Contents

List of Figures

Appendix J

List of Tables

List of Symbols

MSS	Mobile Servicing System
SPDM	Special Purpose Dexterous Manipulator
dSPACE	digital Signal Processing and Control Engineering
EKF	Extended Kalman Filter
a( ), b( )	Coefficient of Butterworth filter
f_C	Cutoff frequency of the analog filter
f_{CLK}	Clock frequency of the analog filter
S_i	$\sin(\theta_i)$
C_i	$\cos(\theta_i)$
S_{ij}	$\sin(\theta_i + \theta_j)$
C_{ij}	$\cos(\theta_i + \theta_j)$
l_1, l_2	Length of link one and two of the robot arm
θ_i	Angular position of the robot joints
$\dot{\theta}_*$	Angular velocity of the robot joints
$\ddot{\theta}_*$	Angular acceleration of the robot joints
τ_i	Angular torque of each joint of the robot
$M(\theta)$	Inertial term(robot) in joint space
$V(\theta, \dot{\theta})$	Centrifugal and Coriolis terms(robot) in joint space
$B(\theta)$	Coriolis term(robot) in joint space
$C(\theta)$	Centrifugal term (robot) in joint space
$G(\theta)$	Gravity term (robot) in joint space
$F(\dot{\theta})$	Friction term (robot)
f_{ext}, f	External forces
$J(\theta)$, Φ_θ	Jacobian matrix

$J_e(\theta)$	Jacobian matrix respect to end-effector coordinate
$J_b(\theta)$	Jacobian matrix respect to base coordinate
λ	Lagrangian multipliers
RLSM	Recursive Least Square Method
SRCUA	Square Root Covariance Update Algorithm
I	Initial position
D	Desired position
X_e	End-effector coordinate
X_b	Base coordinate
$M_x(\theta)$	Inertial term (robot) in Cartesian space
$V_x(\theta, \dot{\theta})$	Centrifugal and Coriolis terms (robot) in Cartesian space
T_{i-1}^{i}	Transformation matrix
$\delta\omega$	Virtual work
^{R}	All variables with this superscript are related to the Right arm
^{L}	All variables with this superscript are related to the Left arm
Right	Represents the right arm of the dual-arm robot
Left	Represent the left arm of the dual-arm robot
f_W, f_O, f_R	Reaction forces on payload
m	Mass of object
r	Position coordinate of the mass
L_C	Unstretched length of payload (elastic object)
L, L_O	Distance between the two end-effectors of a dual-arm robot
d	Deflection of elastic payload
X	Cartesian position of end-effector
$\dot{X}$	Cartesian velocity of end-effector
$\ddot{X}$	Cartesian acceleration of end-effector
KIN	Kinematic equations
k_p	Stiffness of spring for impedance between two end-effectors of dual-arm robot
b_p	Damper coefficient for impedance between two end-effectors of dual-arm robot

T	Sampling time
w	Noise signal
V	Jerk (joint angular)

1.0 INTRODUCTION

1.1 Motivation

The mobile servicing system for the Space Station Freedom includes a special purpose dexterous manipulator. This SPDM is a dual-arm robot, providing manipulative capability over a wide range of tasks. To increase operational automation, (providing the general dexterity level usually associated with that of an astronaut) the technological enhancements to the SPDM require development.

Manipulators, in practice, are generally provided with a conventional Proportional-Integrated-Derivative position controller. However, with current advances in microprocessor, computing, and signal processing technologies, the application of more complex controllers into robotic applications has become more feasible. This is particularly true where dual-arm robotic control is required. Areas such as integrated signal feedbacks from sensors to controller, kinematics and dynamics computations, and arm coordination, benefit greatly from these technologies.

Impedance Control has proved itself a strong candidate for real time trajectory generation and obstacle avoidance in various robotic applications. This control technique can also be used in controlling a dual-arm robot. In Phase 1, the Impedance Controller will direct each arm, individually, toward the object to be grasped. During Phase 2, the two arms will grasp the object creating a closed loop mechanism. At this point, kinematic and dynamic constraints can be transformed into virtual impedances between the two arms. Also, while external forces are small, maximum torque and elastic force application to the payload can be simulated. In Phase 3, each arm will be attracted to the desired point independently by maintaining correct impedance between the two arms.

To date, no publication exists which clearly displays that dual-arm robotic handling of flexible payloads in space is possible. This motivates one to apply Impedance Control in dual-arm robot handling elastic objects in space or non-space applications.

1.2 Research Goals

When a dual-arm robot handles an object, various issues have to be addressed. These issues include coordination, manoeuvring, object handling, kinematics, dynamics, the closed loop chain problem, and the internal force and torque reactions between the end-effectors and the payload. Since this thesis uses the impedance control method, such issues as robustness and efficiency must also be addressed. Disadvantages of computed torque control schemes include such factors as unknown dynamics, friction, and coupled inertial terms. These problems must also be addressed when dealing with a dual-arm robot.

The research goals of this thesis are as follows:

a) build a test-bed dual-arm robot that validates the proposed new techniques.

b) develop an impedance controller that provides enhanced efficiency and robustness.

c) coordinate dual-arm robotic motion in areas of trajectory generation, obstacle avoidance, and flexible body object handling.

1.3 Summary of the results

The experimental setup with all its software gives the user a strong capability to move from simulation to experimentation with a minimum of effort. This system provides ready to use sensors such as integrated resolver and accelerometer, etc. The experimental setup allows the testing of various control methodologies.

The impedance controller has been provided with extensions including a perturbation observer, sensor fusion, and a predictor. These extensions have been verified both analytically and experimentally. The results show the behaviour of a dual-arm robot while handling a flexible payload. Issues of dual-arm robotic coordination including trajectory generation and obstacle avoidance, have also been tested and documented in this thesis.

Contributions:

The following contributions were made during this research work:

- Dual-arm robotic coordination using impedance control was tested for trajectory generation, obstacle avoidance, and flexible object handling. Simulation and experimentation work was carried out to illustrate these issues.
- Impedance control was tested and improved in areas of robustness and efficiency. The addition of a perturbation observer significantly improves the robustness of the controller. These improvements were tested both analytically and experimentally.
- Improvement of the perturbation observers' behaviour and effectiveness was achieved by integrating accelerometers into the system and prediction of an acceleration signal for longer sampling time. These accelerometers improved signal feedback from the observer and increased the observers' bandwidth.
- Observer signals were improved by using sensor fusion of accelerometers and resolvers. An Extended Kalman Filter was used as the sensor fusion algorithm. This improved the observer signal bandwidth. The extended Kalman filter for this work was developed based on kinematic and dynamic models of the system.

2.0 LITERATURE REVIEW

2.1 Dual-Arm Robot

During recent years several researchers worked on different topics in multi arms systems. Because of their complexity, problems such as control, trajectory generation, object handling and obstacle avoidance attracted attention. Multi arms motion control requires a higher level of controller complexity compared to the single-arm robot motion control because of the need to continuously coordinate the motion of the two arms[1]. Various combinations of position and/or force control of the multi arms motion have been reported in the literature.

Luh and Zheng proposed a leader/follower coordination strategy with no force feedback [2,3]. Kinematics constraints for the closed kinematic chain are used for the determination of the follower motion given the leader motion in terms of positions, velocities and accelerations. One rigid body and two articulated rigid bodies are used as loads. The dynamic dependence of the leader and follower input generalized forces are also derived. The major difficulty in using these results to coordinate the motion comes from the lack of contact force feedback. Consequently, such a coordination scheme would be successful only in the case of no servoing and modelling errors.

Ozguner et al developed an approach for cancelling the cross-coupling terms of the dual-arm dynamics which reduces the control problem to controlling independent joint servomotors[4]. This is achieved by a sliding mode, discontinuous controller with gains obtained for a given bound on the norm of cross-coupling interactions. Simulations are performed for free and contact motion of the two arms with no load. No comments are made concerning the evaluation of the norm of cross-coupling interactions and on the operation for various loads.

Swern and Tricamo proposed a modification of the position controllers of two arms intended to compensate the constraint forces applied by the arms on the load[5]. This is achieved by correcting the reference positions based on the end-effector's measured forces/torques and the load stiffness matrix. A flexible rod and a beam are used as loads in simulations and a recursive identification scheme is used for the load stiffness matrix. While this approach uses the actual load stiffness matrix, Kazerooni and Tsai propose an artificial compliance for a similar control scheme[6].

Ahmad and Guo analyzed the effect of joint flexibility on the steady-state error for arms with position/force controllers[7].

Hybrid coordination schemes

Some dual-arm motion hybrid coordination schemes were developed based on the work space force, velocity and position vectors defined as symmetric functions with regard to their joint-space counterparts for the two arms.

Uchiyama and Dauchez proposed a hybrid control scheme for a chosen set of relative and absolute force and velocity in the workspace[8]. Kopf and Yabuta compared experimentally the leader/follower and hybrid control schemes[9].

In the former control scheme, the leader is position controlled and the follower is forced controlled, while absolute position of the centre of the object and the relative(internal) force applied on the object are controlled symmetrically for the two arms. The leader/follower scheme has performed poorly. This can be explained by the lag in the reaction of the follower. The hybrid scheme has had restricted use because of the requirement of the symmetric functions mentioned above.

Suh and Shin generalized the leader/follower scheme by allowing the follower grasping position to change when reaching a configuration singularity, or for obstacle avoidance[10].

Necsulescu and Jassemi applied impedance controller to controlling dual-arm robots for object handling, trajectory generation and obstacle avoidance[11-13].

Shih, Sadler and Gruver introduced a method for planning collision-free trajectories by treating two coordinating robots as a single redundant system. In this method a two-parameter space is created from the intersection of the robots and a trajectory is planned by considering collision-free path in the parameters space as a primary constant and by minimizing an objective function in joint space as a secondary goal. An artificial potential field has also been used for path planning in the parameter space. This method applies to flexible assembly cells equipped with multiple Scara robots[14].

Yoshida, Kurazume and Umetani introduced a new method to control two arms of a free flying robot. One arm traces and given path, while the other arm works both to keep the satellite attitude and to optimize the total operational torque of the system on the basis of a redundancy resolution technique. They also formulated the generalized Jacobian matrix (GJM), to guarantee proper motion control of the free-flying manipulators[15]. Tao and Luh introduced two different model-reference adaptive control schemes, position and force model-reference adaptive controls[16]. Uchiyama and Yamashita worked on adaptive load sharing of two coordinative manipulators that hold a common object searching a robust holding[17].

Yun, Kumar, Sarkar and Paljug studied the control of contact between the end-effectors and the object which is characterized by nonholonomic constraints, i.e., they studied the properties of mechanical systems subject to rolling and sliding constraints. They show that mechanical systems with nonholonomic constraints are not input-state linearizable. Also, they characterized the zero dynamics and derived a nonlinear feedback for the input-output linearization for systems with only position variables, in the output equation of the system[18]. Yao, Gao, Chan and Cheng developed variable structure control(VSC) method for motion, internal force and constrained force control of two manipulators grasping a common constrained object. This control scheme guarantees the system with prescribed qualities in the sliding mode and during the reaching transient regime[19]. Chu and Elmaraghy searched a real-time approach to solve the trajectory planning problem using the updated feedback of joint angles, for determining the step and direction of the next motion. This approach incorporates an efficient collision avoidance strategy to allow decisions to be made during execution. Robot arm linkages are approximated as cylinders and projected on a two-dimensional plane. By this simplification an effective interference checking algorithm is developed to allow decisions to be made in real-time[20].

Derventzis and Davison worked on robust motion/force control of cooperative multi arm systems. Their results show that the closed loop system possesses excellent robustness properties with respect to variations in the arm and load dynamics, and that position and internal force control can be carried out using minimal or even no model information[21]. Chiacchio, Chiaverini and Siciliano developed three schemes for control of multiple arm systems which manipulate a common object. The three schemes are: object task space variables, joint space variables and a combination of object and joint space variables. Their results show that the first and the third schemes operate successfully under imperfect modelling[22]. Hsu and Su showed experimentally the results of the control of two three-DOF-robots to steering the payload and in tracking a preplanned trajectory while controlling the interaction forces and the load distribution among the manipulators[23]. Liu and Goldenberg proposed a hybrid impedance control method in task space which splits into two forces and position controlled spaces. The dynamic behaviour is improved by choosing the desired inertia and damping coefficient of the force control. Impedance control is used in the position controlled subspace. This scheme is equivalent to tracking a desired acceleration trajectory which is generated in real-time[24]. Hemmami worked on the kinematic control of a dual arm robot. The algorithm suggested applies to motion planning and the control of two-arm robotic workstations[25].

Zheng and Chen show two methods for bending a flexible beam[26]. The first method allows the relative position of the two manipulators to be altered, but no relative orientation. The second method allows both position and the orientation to be altered. They conclude that the second method shows better performance, because of reducing the force and moment at the end-effectors. The bending angle is introduced as an independent variable. The trajectory planning is based on bending angle and minimizing force and moment on end-effectors.

Dellinger and Anderson developed an algorism for two manipulators to handle the interactive forces when the object contains a single joint with one or more degree of freedom[27].

Paljug and Yen present experimental results for two arms in performing the task of manipulating a large object by means of an enveloping[28]. Williams and Khatib propose a model to characterize internal forces and moments during multi-grasp manipulation[29]. This approach is based on the construction of a physical model, called the virtual linkage, which is a closed chain mechanism that represents the object being manipulated. Therefore, forces and moments applied at the grasp points of this linkage cause joint forces and torques at its actuators. The virtual linkage becomes a locked structure when these actuators oppose these forces and torques. Inertial forces in the object are then characterized by these forces and torques.

Xi, Tran and Bejczy developed new planning and control scheme for multi-robot coordination[30]. This planning is based on the combination of general task space with the nonlinear feedback technique. They design a hybrid position controller. They, also consider a robot dynamic in force control to improve a distributed computing architecture to implement this scheme in a parallel computation.

This literature review shows that experimental results in handling flexible (especially vibrating) load by dual-arm robot were not published. Also, collision avoidance issue for dual-arm robot and experimental study of dynamic compensation of dual-arm robot were not sufficiently analyzed.

2.2 Recent Improvements in Impedance Control:

Background:

The implementation of the artificial impedance approach was initially based on using a two-part control scheme[31] or a partitioned control law [32], which contains a linear servo control law in Cartesian space and a nonlinear state feedback compensator.

The decoupling method for nonlinear systems has been formulated by Freund and has been illustrated for a joint decoupling case[33]. An interesting implementation of the two-part control scheme consists of generating the joint trajectory by tracking a "false target" point in the phase space of each decoupling joint[34]. Generally, joint space decoupling has been more extensively analyzed than Cartesian space decoupling [31,35,36].

The artificial impedance approach and its continuum mechanical counterpart, artificial potential field approach, are special cases of Cartesian control in which, not only the errors are controlled in Cartesian space, but the robot arm is reduced for the linear controller to a decoupled Cartesian system of independent inertial systems. Khatib formulated the problem in the operational space of the robot end-effector[37,38]. A variety of similar terms (task space, end-effector space, hand space, operational space) refer to Cartesian space. Khatib's results are similar to those incorporated in a Cartesian two-part control scheme, except they were developed originally in a continuum mechanics (Artificial Potential Field) formulation. Hogan et al developed further Artificial Potential Field approach and Artificial Impedance Approach using also a Cartesian two-part control law [39-41,91]. An inverse kinematics scheme for Cartesian space control has been proposed without the coupling considerations[42]. The implementation issues of a Cartesian two-part control law are mainly the difficulties in achieving the Cartesian decoupling of a highly nonlinear robot arm.

Recent progress:

The implementation of a Cartesian two-part control law requires extensive computations of the highly nonlinear terms $M_x(\theta)$ and $V_x(\theta,\dot{\theta})$ besides the kinematics computations present in all Cartesian control schemes. Compared to Cartesian decoupling schemes, joint decoupling schemes are computationally less intensive.

Necsulescu and Jassemi developed a three-part control scheme to reduce the computation intensity of the two-part control scheme[43]. Since in this controller the dynamic terms compensation is done in joint space the computational time reduced.

During recent years several researchers worked on improving the computed torque control method by different type of parameter identification to model the unknown dynamics or develop more robust controllers to overcome some of these unknown dynamics effects. One of the disadvantages of computed torque is the inverse dynamics which can be prohibitive for real time computation, because of time consumption. There is active research to improve this problem.

Inverse dynamic compensation is one of the most time consuming part of any computed torque controllers. S. Komada and K. Okinishi [44] used a first order lag filter to estimate unknown dynamics terms. Papers by T.C. Hsia[45] and K. Youcef-Toumi[46] present a time-delay method for compensation terms. Hsia[45] suggested a simple approach, which basically compares the input and output torque of robot and, for short time delay this error will be added to the next torque input. This is feasible as long as the sampling time is very small. Since there is no torque sensor and the dynamic models are not very accurate to compute output torque, this idea is more useful to compare the computed input and measured output accelerations.

Robust controllers were designed to overcome some of the unknown dynamics terms. Adaptation for an Impedance controller is proposed by W.S. Lu.[47]. Also, an integral term for improving the robustness of controller was considered by G.J. Liu and A.A. Goldenberg[48].

These contributions referred to joint position controller. Necsulescu and Jassemi added an observer to an Impedance controller (Cartesian position controller) to replace the analytical inverse dynamic computation[49].

Also, for better performance of impedance controller the gains of control law can be changed during the motion of the robot while avoiding the obstacles or reaching the target. Jassemi and Necsulescu developed an impedance controller with attractive and repulsive force field variable gains[50]. These gains can vary exponentially.

2.3 Sensor Fusion

The contributions to the integration of multiple sensors into a robotic system are surveyed in [51]. A hieratical integration of multiple sensors into an existing system is proposed and evaluated. The methods for sensory data integration are not analyzed. A methodological analysis of Kalman filter and set-based approaches for sensor data fusion is presented in [52]. Experimental evaluation leads to the conclusion that the performance of each approach is problem dependent. In the methodological analysis of paper [52] and in the books[53] and [54], the data fused refer mainly to complex static(geometric) environmental data while the data fusion problem in this report refers to less complex but time varying(kinematic) data. Data fusion for such state estimation for dynamic systems using an extended Kalman Filter is presented in [55-57]. The approaches proposed in these papers are suitable for the estimation of the time varying joint accelerations by sensor fusion of data from joint(servomotor) sensors and inertial sensors (accelerometers). Several other approaches were proposed for sensor data fusion. Geometric (static) features parameters obtained using a set membership approach to the propagation of uncertain sensory data[58]. The correspondence between observed landmark and a pre-loaded map is verified by an interval based model to account for the uncertainly of sensors data [59].

Novel approaches are proposed in [60] and [61]. Maximum entropy method is applied to the construction of priors end likelihoods from sensors data for a geometric(static) world model [60]. Data fusion using fuzzy logic is applied to a features extraction problem for a static scene [61].

The literature review shows that at present time the estimation of kinematic state variables of a dynamic model using data fusion from various sensors has been approached in an elaborate form only using Extended Kalman Filter(EKF) method. This method is proposed for the data fusion for state estimation of manipulators considered in this thesis.

2.4 Recent Progress in Impedance Control and Dual-Arm Robot(1996-1997)

Bonitz and Hsia [62,63] introduce a robust internal force-based impedance control scheme for coordinating manipulators. This scheme enforces a relationship between the velocity of each manipulator and the internal force on the manipulated objects and requires no knowledge of the object dynamic model. The nonlinear dynamic terms of the robot are compensated by a robust auxiliary controller which is insensitive to robot-model uncertainty and payload variation. They also, presented the effects of computational delays in coordinating multiple-arm manipulating a rigid object using robust internal force-based impedance control[64]. They showed that there exists a lower bound on the inertia matrix vis-a-vis the manipulator's estimated Cartesian endpoint inertia which is independent of the sampling period and is valid under the rigid-body model assumption for the manipulators.

Kraus and McCarragher[65] coordinate the motion of the dual-arm through kinematic redundancy resolution at differential kinematic level. They proposed the use of load flexibility to accommodate for position errors that would result in the build-up of inertial forces in a rigid-load case. They also, proposed a solution based on hybrid position/force approach to coordinate two arms for manipulation of a flexible beam[66]. The choice of controlled variables in each direction is determined by minimum-effort optimality criteria. The control of forces exploits the material's elasticity, which is an impedance for the manipulators. The force controller then behaves as an admittance, in the form of an accommodation(inverse damping) law.

Bonitz and Hsia[67] present a simple method to calibrate a multi-manipulator robotic system. This algorithm reduces the endpoint positioning errors by an order of magnitude using an internal force-based impedance controller and precisely machined calibration plates.

3.0 DUAL-ARM MANIPULATOR

3.1 Manipulator Hardware

3.1.1 Conceptual Design

SPDM robot makes the best test-bed. Unfortunately, not everyone has access to them and they are made of flexible links. Since this work, at this stage, is for a robot with rigid links, and to proof the concept, a simple dual-arm robot will be satisfactory. Especially when the new ideas are

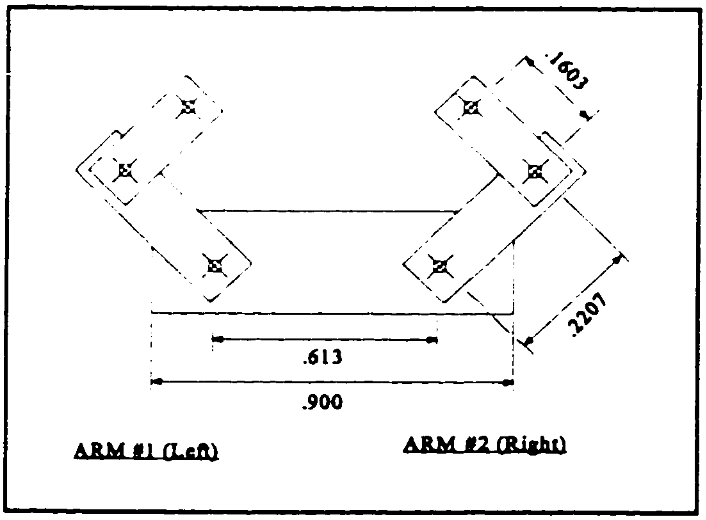

Figure 3-1 Dual-arm planar robot.

going to be tested and they have to be tried on simple robots. Therefore, two two-degree-of-freedom planar robots were considered. These robots have a typical SCARA configuration design ("shoulder, elbow"). Each joint is a revolute joint, and since the robot only moves in planar motion, gravity will not affect the dynamics of the robot. This simple model of cooperative robot arms was selected to test some fundamental issues and control schemes. Each arm can alsc be used for experimental purposes as a single arm. All the links were designed to have a high natural frequency and light weight, and therefore, behave as rigid bodies.

3.1.2 Mechanical Design

The mechanical design of these robots is kept very simple for easy calculation of such factors as inertia terms and kinematics. Each link has been designed for high rigidity. Aluminum was the material used, to obtain low weight. A schematic diagram [figure (3-1)] shows the overall view of both robots. As shown in this diagram, each arm is build from two links and two direct drive motors. For detailed drawings and frequency calculations of each link, refer to Appendix B. The frequency of 'link one' is 270 Hz and that of 'link two' is 876 Hz. This shows that both links behave as rigid bodies for a large variety of experiments.

Direct drive motors were chosen for simplification and to reduce the number of problems that result from using a geared motor. These problems include such factors as, friction in gears, backlash, and no back-drivability.

3.1.3 Grippers

No real grippers have been used. Two pins are used [figure (3-2)] instead of grippers. The payload was connected with these two pins.

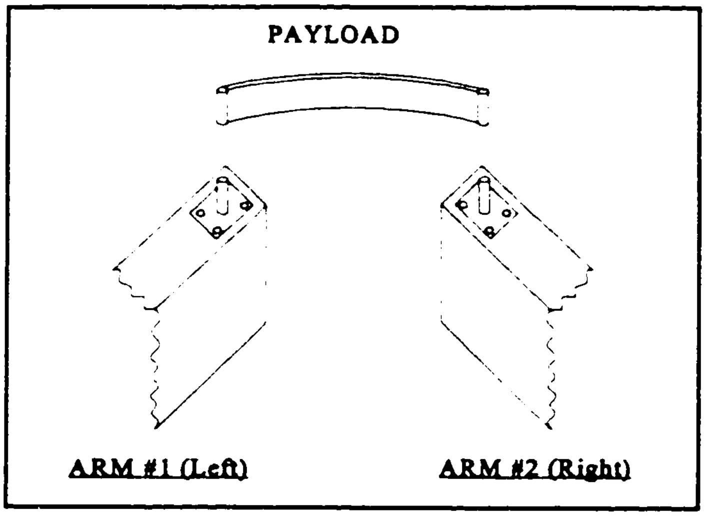

Figure 3-2 End-effectors pins and the payload

3.1.4 Payloads

An elastic string was used for the dual-arm robot to handle. This string has a low stiffness constant, and therefore, if the distance between end-effectors is less than the length of string, the string would not have any reaction forces on the end-effectors, except the weight of the string. There would, however, be internal forces acting on the end-effectors if the robots pulled the string from both sides, and these forces would affect the dynamics of the robots.

3.1.5 Specifications

Specifications of all the motors, sensors and other equipment are in Appendix D. These specifications will help in understanding the experimental setup in greater detail.

3.2 Computing System (hardware)

The schematic diagram [figure (3-3)] shows the setup of hardware, robots, computing systems, sensors and their linkages. All computing is done in dSPACE, which is a Digital Signal Processor(DSP). dSPACE includes a Processor, D/A, A/D, and encoder boards. It runs on a TMS320C30 processor chip. dSPACE is connected to the personal computer(486DX33) by an expansion bus. All motor drivers were serially connected to the computer by a Daisy chain. This serial connection is necessary for setting up the motors before starting the application.

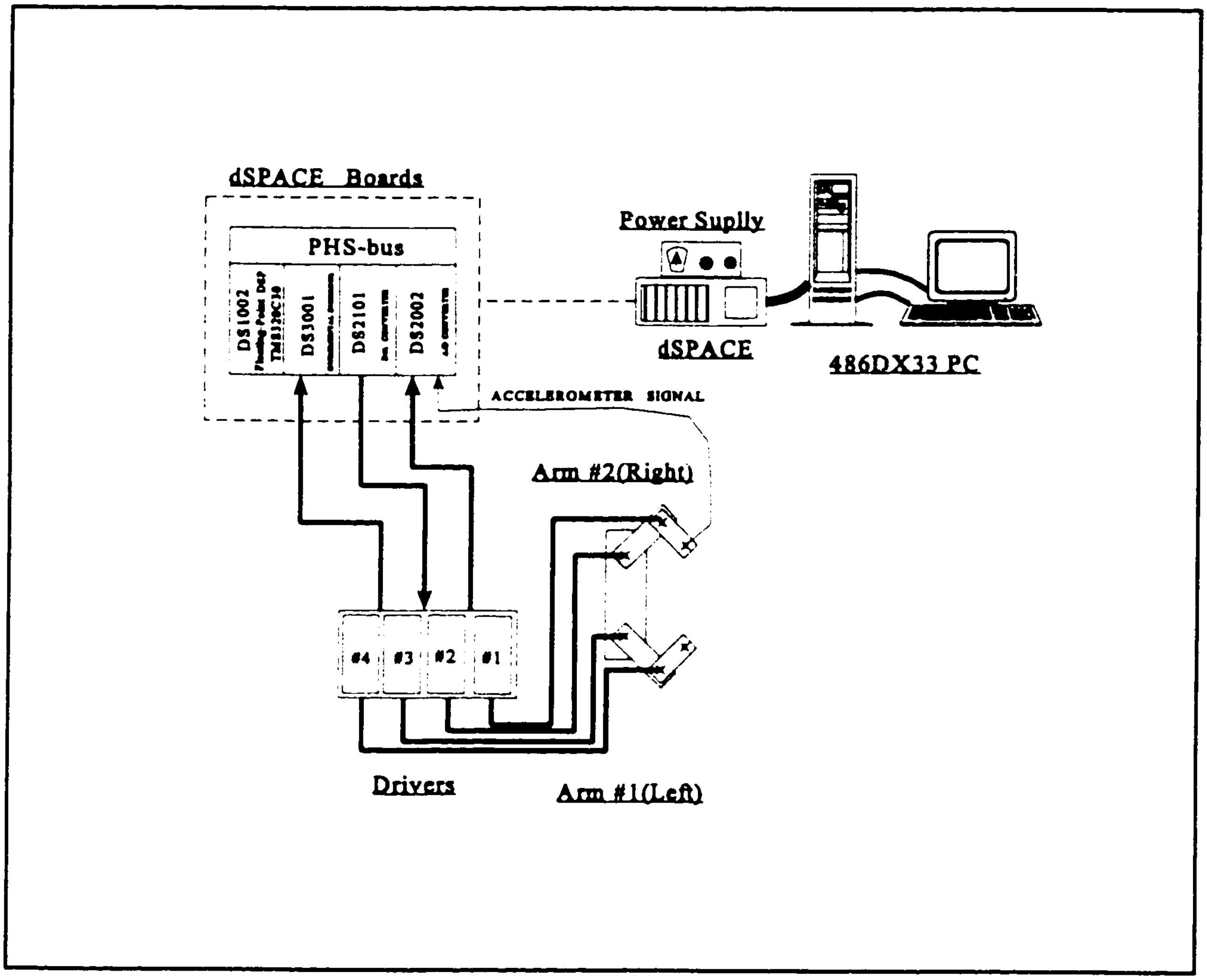

Figure 3-3 Computing system and dual-arm robot

3.2.1 Digital Signal Processor

The hardware of the dSPACE contains the following cards:

Name	Description	Qty.
DS1002	Processor Board	1
DS2002	MUX ADC Board	1
DS2101	DAC Board	2
DS3001	Incremental Encoder Board	2

Table 3-1 List of Boards in dSPACE

3.2.2 Processor Board (DS1002):

A Texas Instruments TMS320C30 (FLoating-point Digital Signal Processor [DSP]) is installed on the DS1002 board. This processor provides a fast instruction cycle for numerical algorithms. The DS1002 is connected to an IBM compatible 486DX33 by a PHS-bus. An architectural overview can be seen in [figure (F-1)]. The processor speed for executing a single-cycle instruction is 60 nano seconds or 16.7 MIPS(a million instructions per second). Greater detail concerning this board can be found in References [68-69].

3.2.3 MUX ADC Board (DS2002):

This board is designed for high-speed variable digital controllers. Specifications of the DS2002 are as follows:

- two separate 16 bit A/D converters with 16 multiplexed inputs each
- 5 μs conversion time for 16-bit resolution

An architectural overview of this board can be seen in [figure (F-2)].

3.2.4 DAC Board (DS2101):

This board is designed for high speed variable digital controllers. The specifications are as follows:

- 5 fully parallel 12 bit voltage output D/A converters
- 3 μs full scale settling time to .01%

Two DS2101 boards are used in this application. Each board is dedicated to a single arm. An architectural overview of this board is shown in [figure (F-3)].

3.2.5 Incremental Encoder Board (DS3001):

This board is designed for high speed variable digital controllers. The specifications are as follows:

- 5 fully parallel 24-bit encoder interface channels
- Fourfold pulse multiplication
- differential (RS422) or single ended (TTL) encoder inputs
- Digital noise pulse filters for the phase lines

Two DS3001 boards are used in this application. Each board is dedicated to a single arm. An architectural overview of this board is shown in [figure (F-4)].

3.3 Software

3.3.1 Computing system software

3.3.1.1 SED30:

This is a setup software. It is used to configure all boards, as improper use by the user, and generate the setup file. IMPEX software usually creates setup files automatically. In the event a user does not wish to use IMPEX, the SED30 should be used to create the setup file[69].

3.3.1.2 MON30:

This is a utility program that loads and evaluates the setup file and object code modules. MON30 then downloads the object module into the DSP. It allows control of the TMS320C30 DSP operation by use of reset and holds signals. It also allows access to the peripheral board registers though the PHS-bus and permits initialization of the I/O channels[69].

3.3.1.3 TRACE30:

This software can, in real time, trace any object code variable from the DS1002 digital signal processing board. It also allows the graphical display of parameters and signals in DS1002 memory, (represented as either single-precision float or integer variables). Trace30 requires two input files. The first file is a memory map file (.TRC) and the second is a linker map file (. MAP). The (.MAP) file contains the addresses of all variables to be traced[70].

3.3.1.4 IMPEX:

This is an application expert, which supports the implementation of high-speed variable linear-invariant controllers. Application issues, such as, discretization and quantizational effects, A/D and D/A conversion delays, computational factors and structure, can be studied by the user by using IMPEX. An example of how to design a simple linear controller can be found in Appendix C.

This software can also be interfaced with MatrixX or Matlab software. Basically, a controller can be designed with MatrixX or Matlab and then be imported into IMPEX. Once imported it can be discretized and used to generate a C code structure. This code should be compiled with the compiler provided by dSPACE. Reference [71] provides further information if needed. The only disadvantage of the IMPEX code generator is its inability to generate nonlinear codes. The user can, however, generate the code for a linear system and then modify the code directly, entering all the nonlinear terms.

3.3.2 User Software

As mentioned in the previous section, the user can use either MatrixX or Matlab software to design a simple linear controller. IMPEX can then generate a C code for the controller. The user should then add all nonlinear terms to the code. This code should then be compiled and linked with the dSPACE compiler and the linker. The resultant will be an object code module that can be down loaded to dSPACE by the MON30 software.

Users also have the option of directly programming the controller without using IMPEX. The only important difference is that the user must have a proper knowledge of all the addresses and programs. If the user does not currently possess such knowledge, it is suggested that they use the IMPEX software. A sample procedure for using MatrixX to design a simple controller will be found in Appendix C. Sample C programs for a single and a dual-arm robot will be found in Appendix G.

3.4 Sensors and Filters

3.4.1 Joint Angle Sensors

The direct drive motors are equipped with high resolution resolvers. Joint one("shoulder") of each robot is driven by about 100 N.m direct drive motor[72]. This motor has a resolver with a resolution of 614400 pulses per revolution. Joint two ("elbow") of each robot is driven by about 10 N.m motor, which has a resolver giving a resolution of 409600 pulses per revolution[72]. Angular velocity and acceleration can be numerically calculated, because of the high resolution of the resolvers. However, due to noise in acceleration calculations, a lowpass digital filter is necessary. A second order Butterworth filter has been selected to cut off high frequency noises. A 5 to 25 Hz cutoff frequency can be selected to give a clear signal.

3.4.2 Accelerometers

Two types of accelerometers have been used, one from ACCESS and one from NOVA[73]. These accelerometers are installed near the end-effectors, one for each (x,y) axis. Accelerometers measure the Cartesian acceleration of the end-effector. This signal can be used to calculate joint acceleration.

Accelerometers also need a digital filter to reduce noise. Because of aliasing, an analog filter is used before digitizing the analog signal. A fourth order Butterworth analog filter, as shown in a figure (3-4), was used for this purpose. After digitization, the acceleration signal is further filtered for noise using a second order Butterworth digital filter with a cutoff frequency of 5 to 25 Hz.

3.4.3 Digital Filters

A second order Butterworth digital filter is used to filter noise signals digitally, such as those associated with acceleration and the observer. The general format of a Butterworth filter is as follows:

$$H(z) = \frac{y(z)}{x(z)} = \frac{\sum b_i z^{-i}}{1 + \sum a_j z^{-j}} \tag{3-1}$$

therefore the filter is given by:

$$y(nT) = \sum b_i x(nT - iT) - \sum a_j y(nT - jT) \tag{3-2}$$

For a second order Butterworth filter the relationship is given by:

$$y(n) = b(1)x(n) + b(2)x(n-1) + b(3)x(n-2) - a(2)y(n-1) - a(3)y(n-2) \tag{3-3}$$

Since different filters with different cutoff frequencies are used in the program, the following table will give the coefficients for different filters with a different bandwidth, when the sampling frequency is 1000 Hz. This table is not valid if the sampling frequency changes.

Coefficient	5 Hz	10 Hz	15 Hz	20 Hz	25 Hz
a(2)	-1.95557824	-1.91119707	-1.86689228	-1.82269495	-1.77863178
a(3)	0.95654368	0.91497583	0.87521455	0.83718165	0.80080265
b(1)	0.00024136	0.00094469	0.00208057	0.00362168	0.00554272
b(2)	0.00048272	0.00188938	0.00416113	0.00724336	0.01108543
b(3)	0.00024136	0.00094469	0.00208057	0.00362168	0.00554272

Table 3-2 Coefficients of a second order Butterworth filter(1000 Hz sampling frequency)

3.4.4 Analog Filters

A fourth order Butterworth analog filter has been used as anti-aliasing filter for accelerometer signals. Figure (3-4), shows a schematic diagram of the filter. The schematic includes a buffer to prevent the accelerometer and the filter from overloading. Reference [74] provides more detail information on buffer and filter used in the figure (3-4). This filter is used to prefilter signals to limit the high frequency components before sampling.

Aliasing occurs when the sampling frequency is less than twice the signal frequency. In this dissertation the sampling frequency of reading analog signals is chosen 1000 Hz. The maximum accelerometer frequencies are 200 Hz for NOVA accelerometer and 500 Hz for ACCESS accelerometer. For these conditions a 120-Hz cutoff frequency filter has been selected. This allows experiments to be conducted using different sampling frequencies between 1000 Hz and 240 Hz while retaining the accelerometer signals below 120 Hz. Sampling frequencies lower than 240 Hz would cause aliasing to occur. To prevent such aliasing, the analog filter should have a cutoff frequency lower than 120 Hz.

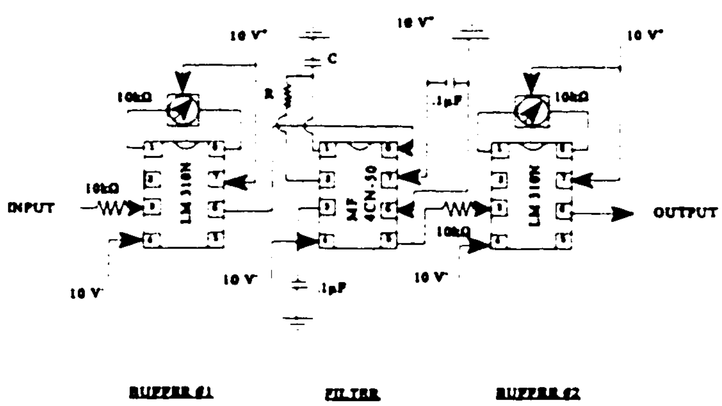

Figure 3-4 4th order Butterworth analog filter with buffers.

This filter can be set to the desired cutoff frequency (f_C) by tuning the filter's clock frequency (f_{CLK}). Varying the values of the resistor(R) and the capacitor(C) can set the filter's clock frequency or self clocking. The ratio of the f_{CLK} to the low-pass f_C is 50 to 1, with accuracy of +/- .3%[74].

The relation between f_{CLK} and RC for V_{CC}=10 volts is:

$$f_{CLK} = \frac{1}{1.69RC} \tag{3-4}$$

also:

$$f_{CLK} = 50 f_C \tag{3-5}$$

therefore for a 120-Hz cutoff frequency, R and C can be calculated as follows:

$f_{CLK} = 50 * 120 = 6000$ Hz

such that for C= .1 μf

$$R = \frac{1}{1.69 f_{CLK} C} = 986\ \Omega \tag{3-6}$$

Therefore, the value R=1000 Ω was selected. This gives a cutoff frequency of about 118 Hz.

4.0 KINEMATIC AND DYNAMIC MODEL OF MANIPULATORS

4.1 Single Arm Manipulator

4.1.1 Kinematic model

A kinematic model of each arm can be found in most robotics books. This work uses Denavit-Hartenberg conventions[32] to develop all kinematic models. Development of the transformation matrix, inverse kinematics, and Jacobian matrix is given in Appendix A. The results are as follows:

Transformation matrix (from eq. A-3):

$$T_3^0 = \begin{bmatrix} C_{12} & -S_{12} & 0 & l_1C_1+l_2C_{12} \\ S_{12} & C_{12} & 0 & l_1S_1+l_2S_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \textbf{(4-1)}$$

Forward kinematics (from eq. A-4):

$$\begin{aligned} x &= l_1 C_1 + l_2 C_{12} \\ y &= l_1 S_1 + l_2 S_{12} \\ z &= 0 \end{aligned} \tag{4-2}$$

Jacobian matrix (from eq. A-13):

$$J(\theta) = \begin{bmatrix} \dfrac{\partial x}{\partial \theta_1} & \dfrac{\partial x}{\partial \theta_2} \\ \dfrac{\partial y}{\partial \theta_1} & \dfrac{\partial y}{\partial \theta_2} \\ \dfrac{\partial \alpha_z}{\partial \theta_1} & \dfrac{\partial \alpha_z}{\partial \theta_2} \end{bmatrix} = \begin{bmatrix} -l_1 S_1 - l_2 S_{12} & -l_2 S_{12} \\ l_1 C_1 + l_2 C_{12} & l_2 C_{12} \\ 1 & 1 \end{bmatrix} \tag{4-3}$$

where:

θ_i	angular position of the robot joints(i=1,2)
S_i	sin (θ_i)
C_i	cos (θ_i)
S_{ij}	sin $(\theta_i + \theta_j)$
C_{ij}	cos $(\theta_i + \theta_j)$
l_1, l_2	lengths of link one and two of the robot arm
α_z	$\theta_1 + \theta_2$

The inverse kinematics equations are provided in the appendix A section A.1.1.

4.1.2 Dynamic Model

There are different ways to design a robot controller. Generally robot dynamic models are considered for the design of the controllers. Computed-torque controllers are a good example of using dynamic-based controllers. These controllers use dynamic models to linearize nonlinear robot dynamics in such a way that one can apply 'a linear controller' to control the robot.

Newtonian and Lagrangian methods lead to the following model of a robot arm:

$$\tau = M(\theta)\,\ddot{\theta} + V(\theta,\dot{\theta}) + G(\theta) + F(\dot{\theta}) + J^T f_{ext} \qquad \textbf{(4-4)}$$

Equation (4-4) can also be written in the following format (eq. A-24):

$$\tau = M(\theta)\,\ddot{\theta} + L(\theta)[\dot{\theta}\dot{\theta}] + C(\theta)[\dot{\theta}^2] + G(\theta) + F(\dot{\theta}) + J^T f_{ext} \qquad \textbf{(4-5)}$$

where, each term is described as follows:

Inertial term:

$$M(\theta) \rightarrow [n \times n]$$

$$M(\theta) = \begin{bmatrix} I_1 + l_1^2 m_1 + a_1^2 m_1 + I_2 + a_2^2 m_2 + 2l_1 a_2 c_2 m_2 & I_2 + a_2^2 m_2 + a_2 l_1 m_2 C_2 \\ I_2 + a_2^2 m_2 + l_1 a_2 C_2 m_2 & I_2 + a_2^2 m_2 \end{bmatrix} \qquad \textbf{(4-6)}$$

Applied Torques:

$$\tau \rightarrow [n \times 1]$$

$$\tau = [\tau_1 \quad \tau_2]^T \qquad \textbf{(4-7)}$$

Centrifugal and Coriolis terms:

$$V(\theta,\dot{\theta}) \rightarrow [n \times 1]$$

$$V(\theta,\dot{\theta}) = \begin{bmatrix} -l_1 a_2 S_2 m_2 (\dot{\theta}_2^2 + 2\dot{\theta}_1 \dot{\theta}_2) \\ l_1 a_2 S_2 m_2 \dot{\theta}_1^2 \end{bmatrix} \qquad \textbf{(4-8)}$$

Centrifugal and Coriolis terms can be written individually as shown in equations(4-9) and (4-10).

Coriolis term:

$$\begin{cases} B(\theta) \rightarrow [n \times n(n-1)/2] \\ [\dot{\theta}\dot{\theta}] = [\dot{\theta}_1\dot{\theta}_2 \quad \dot{\theta}_1\dot{\theta}_3 \;......\dot{\theta}_{n-1}\dot{\theta}_n]^T \rightarrow [n(n-1)/2 \times 1] \end{cases}$$

$$\begin{cases} B(\theta) = -2l_1 a_2 S_2 m_2 \\ [\dot{\theta}\dot{\theta}] = \dot{\theta}_1\dot{\theta}_2 \end{cases} \tag{4-9}$$

Centrifugal term:

$$\begin{cases} C(\theta) \rightarrow [n \times n] \\ [\dot{\theta}^2] = [\dot{\theta}_1^2 \quad \dot{\theta}_2^2 \;.....\; \dot{\theta}_n^2]^T \rightarrow [n \times 1] \end{cases}$$

$$\begin{cases} C(\theta) = \begin{bmatrix} 0 & -l_1 a_2 S_2 m_2 \\ l_1 a_2 S_2 m_2 & 0 \end{bmatrix} \\ [\dot{\theta}^2] = [\dot{\theta}_1^2 \quad \dot{\theta}_2^2]^T \end{cases} \tag{4-10}$$

Gravity term:

$$G(\theta) = [n \times 1]$$

$$G(\theta) = 0 \quad (\textit{planar robot})$$

Friction term:

$$
\begin{cases}
F \rightarrow [n \times n] \\
(\dot{\theta}) = sgn(\dot{\theta}) \rightarrow [n \times 1]
\end{cases}
$$

$$
\begin{cases}
F = \begin{bmatrix} F_1 & 0 \\ 0 & F_2 \end{bmatrix} \\
(\dot{\theta}) = [sgn(\dot{\theta}_1) \quad sgn(\dot{\theta}_2)]^T
\end{cases}
\tag{4-11}
$$

External forces:

$$
\begin{cases}
J^T(\theta) \rightarrow [n \times n] \\
f_{ext} \rightarrow [6 \times 1]
\end{cases}
$$

$$
\begin{cases}
J^T(\theta) = \begin{bmatrix} l_1 S_2 & l_2 + l_1 C_2 & 1 \\ 0 & l_2 & 1 \end{bmatrix} \\
f_{ext} = \begin{bmatrix} f_x \\ f_y \\ \tau_z \end{bmatrix}
\end{cases}
\tag{4-12}
$$

where: J .... Jacobian Matrix

Since external forces are applied to the end-effector and are measured in local end-effector coordinates, the Jacobian term is defined with respect to the end-effector. If the forces are transferred to a base coordinate, the Jacobian given by the equation (A-15) will be used.

4.1.3 Robot Parameter Identification

Parameter identification has always caused problems in dynamic control systems. Many algorithms have been proposed for solving these problems. A Kalman algorithm for the discrete linear estimation problem has produced major improvements for aerospace applications, but some numerical accuracy and stability problems remain to be solved.

One such approach is the U-D(covariance matrix factors: U upper triangular and D diagonal) factorization of the Kalman update algorithm defined in reference[75]. This algorithm is considered a square root type, since $UD^{1/2}$ is a covariance square root. The U-D algorithm has the accuracy characteristic of a square root algorithm(Square Root Covariance Update Algorithm(SRCUA)), and can also be used in real time(appendix E). This algorithm is as efficient as a Kalman algorithm. For a comparison purpose, Recursive Least Square Method(RLSM) was selected as an alternative method since this method is one of the popular methods among the researchers. Therefore, the identifications of the robot parameters were investigated based on two methods:

- Recursive Least Square Method (RLSM)
- Square Root Covariance Update Algorithm (SRCUA)

Experimental results are shown for both methods in Appendix H. Figures (H-1) to (H-8) show experimental results using the RLSM method and figures (H-9) thru (H-16) show results based on the SRCUA method. Since the SRCUA method converged faster and produced more stable estimations, it was selected as the better method. But, for better comparison of which parameters are more accurate, Impedance control based on either parameters should be applied on robot arm. The trajectory which is closest to a straight line it represents the better parameters.

The advantages of SRCUA are:

- computational savings
- reduction in computer storage requirements

- numerical accuracy
- numerical stability

A detailed explanation of the SRCUA algorithm and its associated computer code will be found in Appendix E.

Robot model:

The general format of a robot dynamic equation for unconstrained motion is given by:

$$\tau = M(\theta)\ \ddot{\theta} + V(\theta,\dot{\theta}) + G(\theta) + F(\dot{\theta}) \qquad \textbf{(4-13)}$$

For estimation of the constant parameters in robot dynamic terms, the equation (4-13) can be written in the following format:

$$\tau = \varphi^T \Phi \qquad \textbf{(4-14)}$$

Reference to equations (4-6 to 4-13) allows equation (4-14) for a two DOF planar robot to be written in the two following formats:

Method #1:

This method is based on following equations, (4-6) to (4-13), and (4-14).

A 2x4 matrix format gives:

$$\varphi^T = \begin{bmatrix} \ddot{\theta}_1 & \ddot{\theta}_2 & 2C_2\ddot{\theta}_1 + C_2\ddot{\theta}_2 - 2S_2\dot{\theta}_1\dot{\theta}_2 - S_2\dot{\theta}_2^2 & sgn(\dot{\theta}_1) \\ 0 & \ddot{\theta}_1 + \ddot{\theta}_2 & C_2\ddot{\theta}_1 + S_2\dot{\theta}_1^2 & sgn(\dot{\theta}_2) \end{bmatrix} \qquad \textbf{(4-15)}$$

$$\tau = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \qquad \textbf{(4-16)}$$

$$\Phi = [\Phi_1 \quad \Phi_2 \quad \Phi_3 \quad \Phi_4]^T$$
$$= [I_1 + I_2 + l_1^2 m_1 + a_1^2 m_1 + a_2^2 m_2 \quad I_2 + a_2^2 m_2 \quad l_1 a_2 m_2 \quad F_1]^T \tag{4-17}$$

where Φ_1 ... Φ_4 are four functions of actual robot parameters which they will be identified.

Initial values for Φ can be estimated from information in table (D-3) and equation (4-17), as follows:

Φ_1 =.833, Φ_2 =.057, Φ_3 =.062 and Φ_4 =3.4 (for joint two Φ_4 =.1).

Two sets of results are produced based on each joint's torque.

The first results were based on joint one torque (figures H-9 and H-10):

Φ = [.6866 .0325 .0095 3.271]

The second sets of results were based on joint two torque(figures H-11 and H-12):

Φ = [0 .0552 .0931 .1387]

Method #2:

This method is based on following equations, (4-5) to (4-12), and (4-14). For better identification, equation (4-13) can be expanded to the following 2x6 matrix format.

$$\varphi^T = \begin{bmatrix} \ddot\theta_1 & \ddot\theta_1 + \ddot\theta_2 & C_2(2\ddot\theta_1 + \ddot\theta_2) & -2S_2\dot\theta_1\dot\theta_2 & -S_2\dot\theta_2^2 & sgn(\dot\theta_1) \\ 0 & \ddot\theta_1 + \ddot\theta_2 & C_2\ddot\theta_1 & 0 & S_2\dot\theta_1^2 & sgn(\dot\theta_2) \end{bmatrix} \tag{4-18}$$

$$\Phi = [\Phi_1 \quad \Phi_2 \quad \Phi_3 \quad \Phi_4 \quad \Phi_5 \quad \Phi_6]^T$$
$$= [I_1 + l_1^2 m_1 + a_1^2 m_1 \quad I_2 + a_2^2 m_2 \quad a_2 l_1 m_2 \quad l_1 a_2 m_2 \quad l_1 a_2 m_2 \quad F_1]^T \tag{4-19}$$

where, the initial values for Φ based on first and second joints were found to be:

First joint : $\Phi = [\ .776 \quad .0572 \quad .0619 \quad .0619 \ .0619 \ 3.4]$

Second joint: $\Phi = [\ 0 \quad .0519 \quad .0661 \quad 0 \ .0661 \ .2\ ]$

The final results produced were as follows:

First joint: $\Phi = [\ .6613 \quad .0466 \quad .0073 \quad .1978 \ .0578 \ 3.37]$
(Figures H-13 and H-14)

Second joint $\Phi = [\ 0 \quad .054 \quad .0934 \quad 0 \ .0614 \ .1585\ ]$
(Figures H-15 and H-16)

The choice of method is based on testing experimentally the effect of dynamic compensation using the results of each method. Exact parameters would give a straight line trajectory. Figure (4-1) visually displays' end-effector trajectories using the identification results for the parameters Φ of initial estimation, Method #1 and Method #2. Trajectories, in cases of dynamic compensation, using Method #1 [curve (b)] and Method #2 [curve (c)] results are closer to a straight line trajectory than the one using initial dynamic parameters[curve (a)]. The initial dynamic parameters are calculated from the data in table D-3. Method #2, results, [curve (c)] provide a much closer to a straight line trajectory.

It was therefore concluded that the expanded equation based on [2X6] matrix φ^T(Method#2) would result in a better estimation of parameters

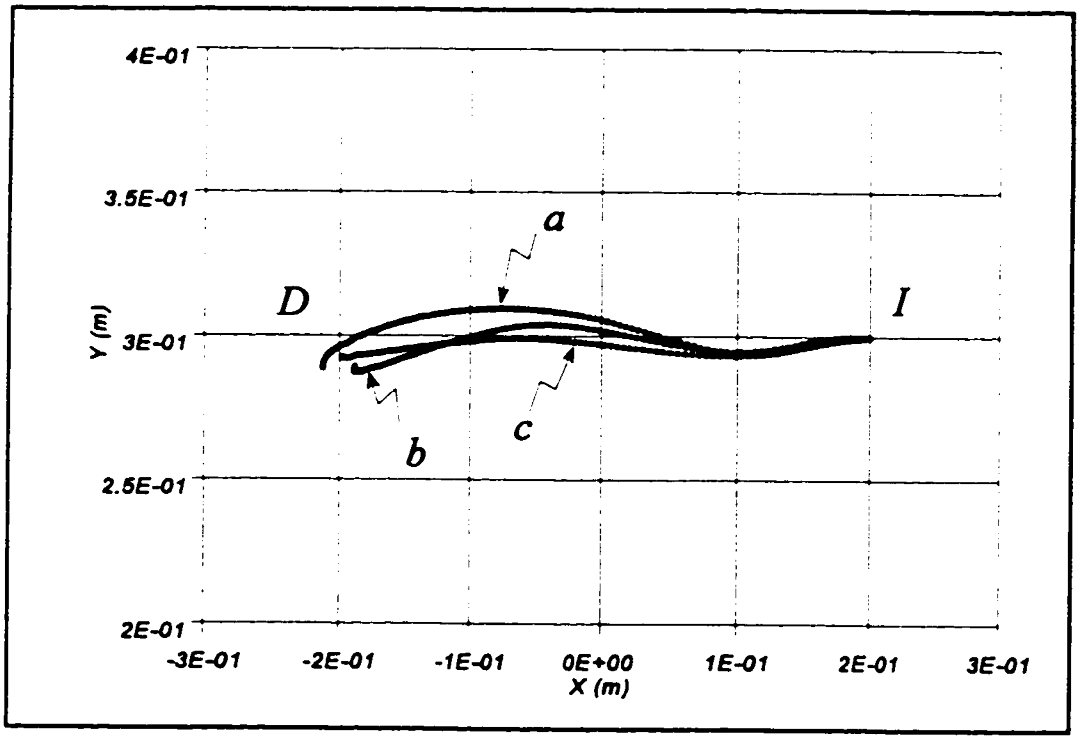

Figure 4-1 Experimental results of trajectory generation by end-effector under these three sets of parameters. Curve (a) represents full dynamic compensation using initial dynamic parameters, curve (b) represents the results of dynamic compensation using Method#1 parameters and curve (c) represents results of dynamic compensation using Method#2 parameters.

4.1.4 External Forces on a Single Arm Manipulator

External forces applied on the end-effector are due to different reasons, such as contact motion and object handling. These forces can be represented in two different coordinate systems,

1) with respect to end-effector coordinates X_e
2) with respect to base coordinate X_b

The n x 1 vector of external forces given by equation (4-12) is:

$$f = [f_x \, f_y \, f_z \, \tau_x \, \tau_y \, \tau_z]^T \tag{4-20}$$

Transformation of forces between the end-effector and joint frames are obtained as follows:

Virtual work in joint space:

$$\delta\omega = \tau^T \Delta\theta \tag{4-21}$$

where:

τ^T = is n-vector of arm control torques/forces

$\Delta\theta$ = differential changes in joint $[\Delta\theta_1, \ldots\ldots \Delta\theta_n]^T$

which, in a force coordinate frame, the virtual work is:

$$\delta\omega = f^T \Delta y \tag{4-22}$$

where:

$$\Delta y = J \Delta\theta \tag{4-23}$$

J is the Jacobian

- From equations (4-21 to 4-23):

$$\delta\omega = \tau^T \Delta\theta = f^T J \Delta\theta \tag{4-24}$$

$$\tau = J^T(\theta) f \tag{4-25}$$

where, f , is the Cartesian generalized force:

$$f = [f_c \quad \tau_c]^T \tag{4-26}$$

and where: $f_c = [f_x f_y f_z]^T$

$\tau_c = [\tau_x \; \tau_y \; \tau_z]^T$

If the generalized force (f) is within the end-effector coordinates, then this will be represented as (f_e), and can be transferred into a base coordinate.

From equation (A-19):

$$f_b^b = R_e^b f_e^e$$

$$n_i^i = R_{i+1}^i n_{i+1}^{i+1} + P_{i+1}^i \times R_{i+1}^i f_{i+1}^{i+1} \qquad \textbf{(4-27)}$$

$where$: $i = b \ldots e$

b= base coordinate

e= end-effector coordinate

The Jacobian matrices with respect to the end-effector (J_e) and the base (J_b) coordinates from equations (4-12,A-15) are:

$$J_e(q) = \begin{bmatrix} l_1 S_2 & 0 \\ l_2 + l_1 C_2 & l_2 \\ 1 & 1 \end{bmatrix}$$

$$\textbf{(4-28)}$$

$$J_b(q) = \begin{bmatrix} -l_1 S_1 - l_2 S_{12} & -l_2 S_{12} \\ l_1 C_1 + l_2 C_{12} & l_2 C_{12} \\ 1 & 1 \end{bmatrix}$$

4.2 Dual-Arm Manipulator

4.2.1 Dynamic Model

Dual-arm robot motion is analyzed in relation to two distinct phases: the approach phase (when each arm is moving independently toward a desired position) and the coordination phase (when the two arms are transporting an object to a desired position).

Motion in the approach phase is coordinated by introducing (artificial) attractive impedances between each arm and a desired position, and artificial repulsive impedances between the arms and obstacles detected on their trajectory paths.

Motion in the second phase, when the two arms are holding an object, represents the motion of a closed loop mechanism. The artificial impedance approach to motion coordination, in this case, is based on designating one arm as a leader arm and the other as a follower arm. Safe object movement then becomes an issue, because materials and holding requirements impose kinematic and dynamic constraints at the contact points of the object and the two arms. Using the artificial impedance approach, these constraints are transformed into virtual impedances between the two arms, (as long as the external forces are small), which can simulate maximum elastic forces and torques applied on the payload.

This approach permits the motion of the two arms to be coordinated, in the same fashion as in the case of free motion, by introducing suitable artificial impedances between the arms, the desired positions, and the obstacles.

Since each robot arm is controlled separately, the leader and follower concept cannot be used, except when handling a solid (rigid) object. The dynamic model for an individual robot is similar to that of a single arm robot as described in equation(4-13).
In order to prevent confusion in equation notation, the following symbols will be used to distinguish between the two robots. L = left robot arm and R = right robot arm.

Dynamic model for Left robot:

$$\tau^L = M(\theta^L)\ \ddot{\theta}^L + v(\theta^L,\dot{\theta}^L) + G(\theta^L) + F(\dot{\theta}^L) + J^{T^L} f_{ext}^L \qquad \textbf{(4-29)}$$

Dynamic model for Right robot:

$$\tau^R = M(\theta^R)\ \ddot{\theta}^R + v(\theta^R,\dot{\theta}^R) + G(\theta^R) + F(\dot{\theta}^R) + J^{T^R} f_{ext}^R \qquad \textbf{(4-30)}$$

where:

f_{ext}^L represents external forces applied on the Left end-effector

and f_{ext}^R represents external forces applied on the Right end-effector

The external forces referred to contain all reaction forces. These reaction forces include those generated by repulsive forces, contacts, payload, and the other robot. Later sections in this chapter provide a more detailed description of the external forces under discussion.

4.2.2 Coordinate System for the Payload

Two planar robots have been considered, as displayed in the figure (4-2), with the following coordinate system:

X^L Left position coordinate

X^R Right position coordinate

Position vectors can be defined as follows:

The vector from X^L to X^R:

$$X_b^{LR} = X_L^R - X^L \tag{4-31}$$

The vector from X^R to X^L:

$$X_b^{RL} = X_R^L - X^R \tag{4-32}$$

where: (X_L^R = Right arm coordinate(R) express in Left arm coordinate(L)

X_R^L = Left arm coordinate(L) express in Right arm coordinate(R))

Angles between the payload and the base coordinate are:

$$\begin{aligned} &\textit{Left}: \quad \theta_b^L = atan2(X_{b_y}^{LR}, X_{b_x}^{LR}) \\ &\textit{Right}: \quad \theta_b^R = atan2(X_{b_y}^{RL}, X_{b_x}^{RL}) \end{aligned} \tag{4-33}$$

Rotation matrices for transferring the object coordinate from the base to object coordinate are:

Left:

$$R_b^{o\,L} = \begin{bmatrix} C\theta_b^L & -S\theta_b^L & 0 \\ S\theta_b^L & C\theta_b^L & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad R_b^{b\,L} = \begin{bmatrix} C\theta_b^L & S\theta_b^L & 0 \\ -S\theta_b^L & C\theta_b^L & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4-34}$$

Right:

$$R_b^{o\,R} = \begin{bmatrix} C\theta_b^R & -S\theta_b^R & 0 \\ S\theta_b^R & C\theta_b^R & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad R_o^{b\,R} = \begin{bmatrix} C\theta_b^R & S\theta_b^R & 0 \\ -S\theta_b^R & C\theta_b^R & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4-35}$$

Object positioning with respect to the object coordinate is given by:

Left:

$$X_o^L = R_b^{o^L} X_b^{LR}$$

$$X_o^L = \begin{bmatrix} C\theta_b^L & -S\theta_b^L \\ S\theta_b^L & C\theta_b^L \end{bmatrix} \begin{bmatrix} X_{b_x}^{LR} \\ X_{b_y}^{LR} \end{bmatrix} \tag{4-36}$$

Right:

$$X_o^R = R_b^{o^R} X_b^{RL}$$

$$X_o^R = \begin{bmatrix} C\theta_b^R & -S\theta_b^R \\ S\theta_b^R & C\theta_b^R \end{bmatrix} \begin{bmatrix} X_{b_x}^{RL} \\ X_{b_y}^{RL} \end{bmatrix} \tag{4-37}$$

Angles between the end-effector and object coordinate are:

$$\begin{aligned} &\textit{Left}: && \theta^L = \theta_b^L - (q_1^L + q_2^L) \\ &\textit{Right}: && \theta^R = \theta_b^R - (q_1^R + q_2^R) \end{aligned} \tag{4-38}$$

The rotation matrix between the end-effector and object coordinate is:

Left:

$$R_o^{e^L} = \begin{bmatrix} C\theta^L & -S\theta^L & 0 \\ S\theta^L & C\theta^L & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad R_e^{o^L} = \begin{bmatrix} C\theta^L & S\theta^L & 0 \\ -S\theta^L & C\theta^L & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4-39}$$

Right:

$$R_o^{e^R} = \begin{bmatrix} C\theta^R & -S\theta^R & 0 \\ S\theta^R & C\theta^R & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad R_e^{o^R} = \begin{bmatrix} C\theta^R & S\theta^R & 0 \\ -S\theta^R & C\theta^R & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4-40}$$

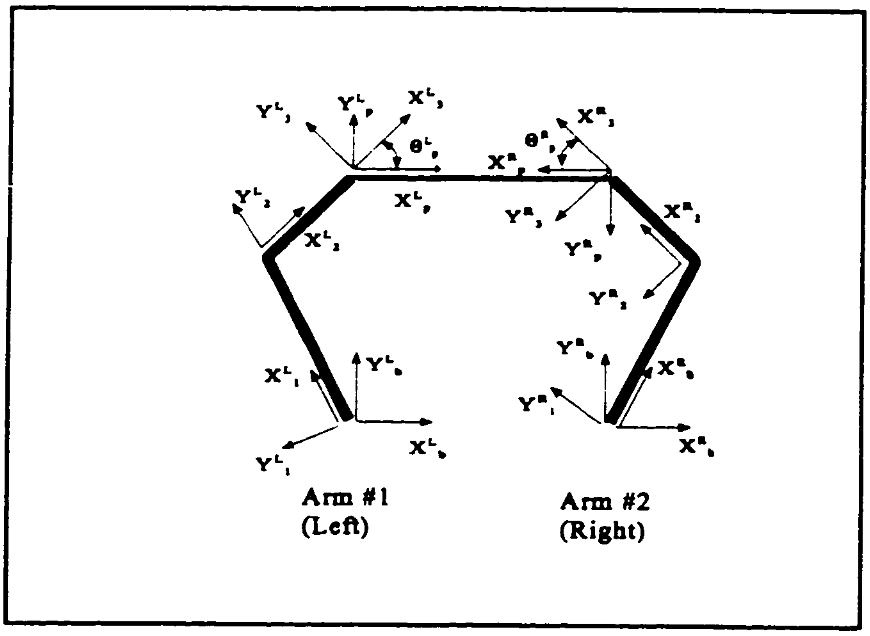

Figure 4-2 Coordinate system of dual-arm robot

4.2.3 Forces Applied on the Payload

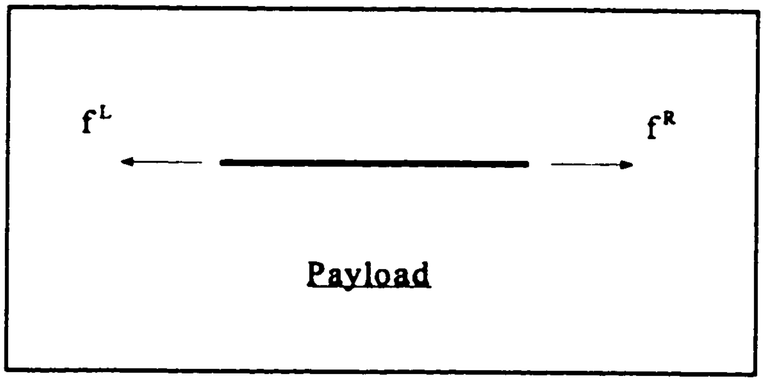

Figure 4-3 Forces applied on the end-effectors

Resultant forces (f^L, f^R), applied to the payload, have the following components:

$f_w=(f_w^L \; , f_w^R)$ forces due to the payload weight, which produce a bending moment and generate forces on the end-effector(approximation $f_w^L=f_w^R=\frac{mg}{2}$ where m is mass of the payload),

$f_a=(f_a^L \; , f_a^R)$ inertial forces, such as Newton law $f=ma=m\ddot{r}$ (where: r = absolute position of the centre of mass), to maintain a consistency with other forces, this force has to be transferred to object coordinates by using rotation matrix R_o^b (b= base coordinate, o = object coordinate).

$f_o = (f_o^L, f_o^R)$ the reaction forces resulting from the elastic deflection of an object.

$f_R = (f_R^L, f_R^R)$ the reaction force applied from the other robot's end-effector.

These forces can be represented in object coordinates, making it easier to handle them. Transferring these forces to end-effector coordinates is given by:

$$\begin{aligned} \textit{Left}: \quad f_e^L &= R_e^{o\,L} \begin{bmatrix} f_x^L \\ f_y^L \\ 0 \end{bmatrix} \\ \textit{Right}: \quad f_e^R &= R_e^{o\,R} \begin{bmatrix} f_x^R \\ f_y^R \\ 0 \end{bmatrix} \end{aligned} \tag{4-41}$$

4.2.4 Forces applied on the end-effectors

Forces applied to the end-effectors usually derive from contact motion or payloads. This work focuses on the forces applied on the end-effectors based on a payload. Payloads fall into three object types, as follow:

1) the rigid object
2) the elastic object (low stiffness)
3) the flexible object (high stiffness)

For purposes of experimentation and simulation, a light and elastic object with low stiffness was chosen.

4.2.4.1 Elastic object

In the case of a dual-arm robot handling elastic bodies such as a cable, string, or blanket [49 and 76-78], the controller can be designed to maintain a constant tension between the end-effectors. This prevents sag and also dampens object vibration.

Figure (4-4) and subsequent equations display the three cases to be considered for deriving an elastic object model, where:

L_C length of cable,

L distance between two end-effectors,

d mid-span deflection,

r absolute position vector of the centre of the mass

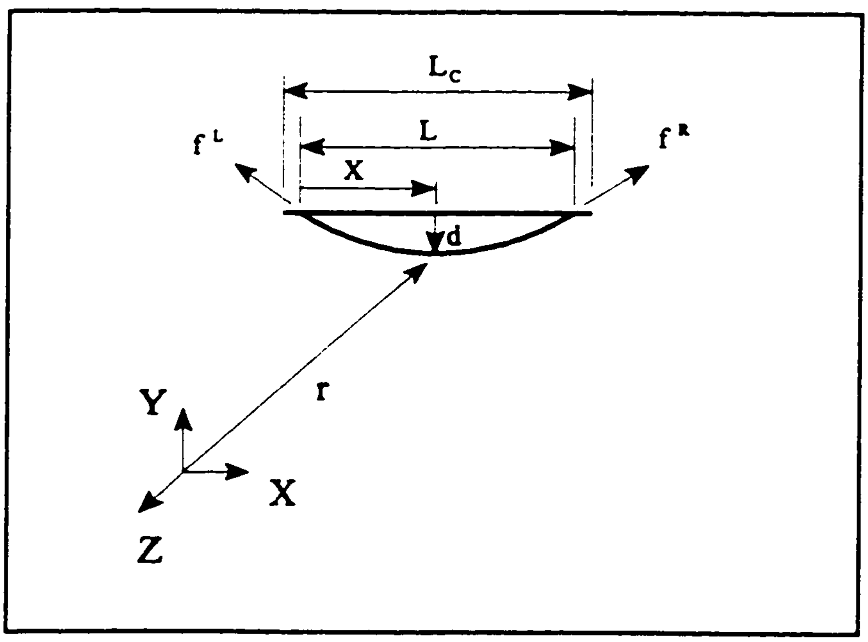

Figure 4-4 Elastic object

1) when: $L_C > L$

$f_o = 0$

$f_R = 0$

where f_O and f_R are defined in section (4.2.3).

therefore:

$$\begin{aligned} f^L &= f_w^L + f_a^L \\ f^R &= f_w^R + f_a^R \end{aligned} \tag{4-42}$$

2) when: $L_C = L$

$f_O = 0$

therefore:

$$\begin{aligned} f^L &= f_w^L + f_a^L + f_R^L \\ f^R &= f_w^R + f_a^R + f_R^R \end{aligned} \tag{4-43}$$

3) when: $L_C < L$

therefore:

$$\begin{aligned} f^L &= f_w^L + f_a^L + f_o^L \\ f^R &= f_w^R + f_a^R + f_o^R \end{aligned} \tag{4-44}$$

In the third case, f_O represents forces that occur from extending the object:

$$f_o = \Delta L \; K \tag{4-45}$$

where:

$\Delta L =$ $L_C - L$

$K =$ stiffness coefficient of the body[76] = $\frac{EA}{L_c}$

E module of elasticity

A cross section of the object

The payload selected for simulation and experimental purposes is very light. The forces due to weight and inertia are therefore small and negligible.

5.0 CONTROL APPROACHES FOR SINGLE AND DUAL-ARM ROBOTS USING IMPEDANCE CONTROL

Impedance control of manipulators was selected as the control methodology for this thesis. The thesis verifies impedance control as a viable candidate in areas of point to point control, on-line trajectory generation, obstacle avoidance, and object handling in single and dual-arm robotic applications. This chapter will discuss the application issues relating to impedance control of single and dual-arm robotic manipulators. Also, the comparison of the end-effector trajectories is selected as comparison method for different methodologists which are introduced and used in this chapter. Therefore, an straight line trajectory is a desired trajectory and any method that creates more straight line trajectory is considered as better method or result. All the simulation and experimental tests are done for high speed robot. The average and maximum end-effector speed are about .7 m/s and .4 m/s respectively. Also, the average and maximum of joint one angular speed is about 2 rad/s and 1 rad/s, and joint two 4 rad/s and 2 rad/sec respectively.

5.1 Impedance Control of a Single Arm Robot

5.1.1 The concept of Impedance control

Robotic manipulator impedance control is based on creating an artificial mechanical impedance between the robots end-effector and the target point[79]. This mechanical impedance is created by an artificial spring and damper, between the end-effector and the target. The artificial mechanical impedance system uses a spring component defined at rest in the target point and produces an equivalent attractive force in Cartesian space, in the direction of a straight line, from the end-effector to the target.

Artificial MBK(mass, damper, spring) impedance, between the position vectors of end-effector (X) and the target(X_d), for free motion, can therefore be defined as:

$$M\ddot{X} + B(\dot{X}-\dot{X}_d) + K(X-X_d) = 0 \quad \textbf{(5-1)}$$

giving the attractive acceleration command:

$$\ddot{X} = M^{-1}(B(\dot{X}_d-\dot{X}) + K(X_d-X)) \quad \textbf{(5-2)}$$

where:

K = the virtual spring coefficient

B = virtual damper coefficient

M = virtual mass

The resulting force vector in Cartesian space for free motion of a mass m is:

$$f = m\ddot{X} \quad \textbf{(5-3)}$$

$$f = mM^{-1}(B(\dot{X}_d-\dot{X}) + K(X_d-X)) \quad \textbf{(5-4)}$$

This Cartesian force, applied on the equivalent Cartesian mass m, can be generated by using the dependence of f and the joint torque vector[32]:

$$\tau = J^T f \tag{5-5}$$

To direct a robot on a straight line trajectory, dynamic terms relating to gravity, inertia, friction, centrifugal force, and the Coriolis effect should be compensated for in Cartesian space. Because of this, a new control scheme made up of three parts has been developed such that compensation of all these terms can be separated between joint space and Cartesian space. This three-part scheme reduces the calculation complexity of all mechanical terms in Cartesian space.

5.1.2 Model-based Impedance control of a robot arm.

Assuming the measurement of Cartesian contact forces 6x1 vector $\mathbf{F}_{ext}$ between the endpoint of a 6DOF jointed robot and the environment is available, the dynamic equation concerning a 6x1 vector θ of joint angles gives the following 6x1 vector τ of joint torques (from eq. 4-4):

$$\tau = M(\theta)\ddot{\theta} + V(\theta,\dot{\theta}) + G(\theta) + F(\dot{\theta}) + J^T(\theta)\ F_{ext} \tag{5-6}$$

Specific to the Artificial Impedance approach is the choice of a desired artificial impedance between the endpoint, positioned at X(a 6x1 vector) and the target positioned at X_d (a 6x1 vector), producing the following dynamics:

$$-F_{ext} = M\ \ddot{X} + B(\dot{X} - \dot{X}_d) + K(X - X_d) \tag{5-7}$$

M, B and K are 6x6 diagonal matrices of the desired artificial impedance in Cartesian space. From robot kinematics, the following relationships between state variables in joint space and in Cartesian endpoint space can be written [32]

$$X = KIN(\theta) \tag{5-8}$$

where:

KIN(θ) is the forward Kinematic function (eq. (4-2)).

$$\dot{X} = J(\theta)\ \dot{\theta} \tag{5-9}$$

$$\ddot{X} = J(\theta)\ddot{\theta} + \dot{J}(\theta)\dot{\theta} \tag{5-10}$$

and:

The Jacobian 6x6 matrix J(θ) relates joint velocities $\dot{\theta}$ to the defined, base frame, endpoint velocities $\dot{X}$.

Using equations (5-8) to (5-10) in equation (5-7), F_{ext} can be written as a function of $\theta, \dot{\theta}, \ddot{\theta}$ as follows:

$$-F_{ext} = MJ(\theta)\ddot{\theta} + M\dot{J}(\theta)\dot{\theta} - B[\dot{X}_d - J(\theta)\dot{\theta}] - K[X_d - KIN(\theta)] \tag{5-11}$$

Based on measured $\theta, \dot{\theta}$ and F_{ext}, eq. (5-11) gives the computed acceleration:

$$\begin{aligned}\ddot{\theta}^{(c)} = & -J^{-1}(\theta)M^{-1}F_{ext} - J^{-1}(\theta)\dot{J}(\theta)\dot{\theta} + J^{-1}(\theta)M^{-1}B[\dot{X}_d - J(\theta)\dot{\theta}] \\ & + J^{-1}(\theta)M^{-1}K[X_d - KIN(\theta)]\end{aligned} \tag{5-12}$$

Substituting $\ddot{\theta}$ in eq. (5-6) by $\ddot{\theta}^{(c)}$ we obtain:

$$\begin{aligned}\tau = & -M(\theta)J^{-1}(\theta)M^{-1}F_{ext} - M(\theta)J^{-1}(\theta)\dot{J}(\theta)\dot{\theta} + M(\theta)J^{-1}(\theta)M^{-1}B[\dot{X}_d \\ & - J(\theta)\dot{\theta}] + M(\theta)J^{-1}(\theta)M^{-1}K[X_d - KIN(\theta)] + \nu(\theta,\dot{\theta}) + G(\theta) + F(\dot{\theta}) \\ & + J^T(\theta)F_{ext}\end{aligned} \tag{5-13}$$

or

$$\tau = M(\theta)J^{-1}(\theta)\{-M^{-1}F_{ext} - \dot{J}(\theta)\dot{\theta} + M^{-1}B[\dot{X}_d - J(\theta)\dot{\theta}] + M^{-1}K[X_d - KIN(\theta)]\} + \nu(\theta,\dot{\theta}) + G(\theta) + F(\dot{\theta}) + J^T(\theta)F_{ext} \tag{5-14}$$

equation(5-14) can be written in to the following format:

$$\tau = J^T(\theta)\{J^{-T}(\theta)M(\theta)J^{-1}(\theta)\}\{-M^{-1}F_{ext} + M^{-1}B[\dot{X}_d - J(\theta)\dot{\theta}] + M^{-1}K[X_d - KIN(\theta)]\} - M(\theta)J^{-1}(\theta)\dot{J}(\theta)\dot{\theta} + \nu(\theta,\dot{\theta}) + G(\theta) + F(\dot{\theta}) + J^T(\theta)F_{ext} \tag{5-15}$$

or

$$\tau = J^T(\theta)\left\{\{J^{-T}(\theta)M(\theta)J^{-1}(\theta)\}\{-M^{-1}F_{ext} + M^{-1}B[\dot{X}_d - J(\theta)\dot{\theta}] + M^{-1}K[X_d - KIN(\theta)]\} + J^{-T}\{\nu(\theta,\dot{\theta}) - M(\theta)J^{-1}(\theta)\dot{J}(\theta)\dot{\theta}\} + J^{-T}G(\theta) + J^{-T}F(\dot{\theta}) + F_{ext}\right\} \tag{5-16}$$

Lets consider the following as Cartesian parameters as defined in reference[32]:

- a Cartesian mass matrix:

$$M_x(\theta) = J^{-T}(\theta)M(\theta)J^{-1}(\theta) \tag{5-17}$$

- centrifugal and Coriolis terms in Cartesian space:

$$\nu_x(\theta,\dot{\theta}) = J^{-T}(\theta)[\nu(\theta,\dot{\theta}) - M(\theta)J^{-1}(\theta)\dot{J}(\theta)\dot{\theta}] \tag{5-18}$$

- gravitational and friction terms in Cartesian space:

$$G_x(\theta) = J^{-T}(\theta)\; G(\theta) \tag{5-19}$$

$$F_x(\dot{\theta}) = J^{-T}(\theta)\; F(\dot{\theta}) \tag{5-20}$$

Using equations (5-15) to (5-20) in (5-16) the results will be as follow:

$$\begin{aligned}\tau = {} & J^T(\theta)\{M_x(\theta)\{-M^{-1}F_{ext} + M^{-1}B[X_d - J(\theta)\theta] + M^{-1}K[X_d - KIN(\theta)]\} \\ & + \nu_x(\theta,\dot{\theta}) + G_x(\theta) + F_x(\theta) + F_{ext}\}\end{aligned} \tag{5-21}$$

For $M=M_x(\theta)$, the coefficient of F_{ext} is $I-M_xM^{-1}=0$ and F_{ext} does not influence the position control commands.

The computed Cartesian acceleration can be calculated by:

$$\ddot{X}^{(c)} = M^{-1}K[X_d - KIN(\theta)] + M^{-1}B[\dot{X}_d - J(\theta)\dot{\theta}] - M^{-1}F_{ext} \tag{5-22}$$

Using the relationship between the Cartesian force 6x1 vector F and τ [32] ($F = J^{-T}\tau$), we obtain the Cartesian space dynamics equation:

$$F = M_x(\theta)\,\ddot{X}^{(c)} + v_x(\theta,\dot{\theta}) + G_x(\theta) + F_x(\dot{\theta}) + F_{ext} \tag{5-23}$$

For F and $\ddot{X}^{(c)}$ defined regarding the robot base frame, equation (5-22) gives the output of a linear Cartesian controller and equation (5-23) contains the Cartesian decoupling scheme terms $M_x(\theta)\ddot{X}^{(c)}, v_x(\theta,\dot{\theta}), G_x(\theta), F_x(\dot{\theta})$ and F_{ext} (figure (5-7)).

The Cartesian decoupling scheme has not been assumed in this derivation, but is a result of imposing the desired artificial impedance in equation (5-7) between the robot endpoint and the target as a reference model.

Collision avoidance features can be incorporated into this scheme by posing artificial repulsive impedances between the robot arm and obstacles [79-80].

5.1.3 Three-part Impedance control scheme

The three-part Impedance controller used in examples shown in the figure (5-2). This control scheme is computation efficient when compared with a two-part Cartesian decoupling scheme(figure (5-1)). This controller replaces the intensive computation of inertial, centrifugal, Coriolis, and friction terms in Cartesian space[79], which are needed in a two-part control scheme. Reduction in computation speeds up linearization approximately by 50 percent.

This Impedance controller consists of three parts:

1. The Linear Cartesian control law (from eq. (5-7)):

$$\ddot{X} = M^{-1} K\,(X_d - X) + M^{-1} B\,(\dot{X}_d - \dot{X}) - M^{-1} F_{ext} \tag{5-24}$$

2. The Cartesian decoupling scheme of a joint decoupled arm:

$$\ddot{\theta} = J^{-1}(\theta)\ [\ddot{X} - \dot{J}(\theta)\dot{\theta}] \qquad \textbf{(5-25)}$$

3. The Joint decoupling scheme:

$$\tau = M(\theta)\ddot{\theta} + V(\theta,\dot{\theta}) + J^{T}(\theta)f_{ext} \qquad \textbf{(5-26)}$$

where:

$$M = M_x = J^{-T}(\theta)\ M(\theta)\ J^{-1}(\theta) \qquad \textbf{(5-27)}$$

M is selected for the worse conditions (i.e., near a singularity point), therefore M is calculated once and is kept constant for all other arm configurations.

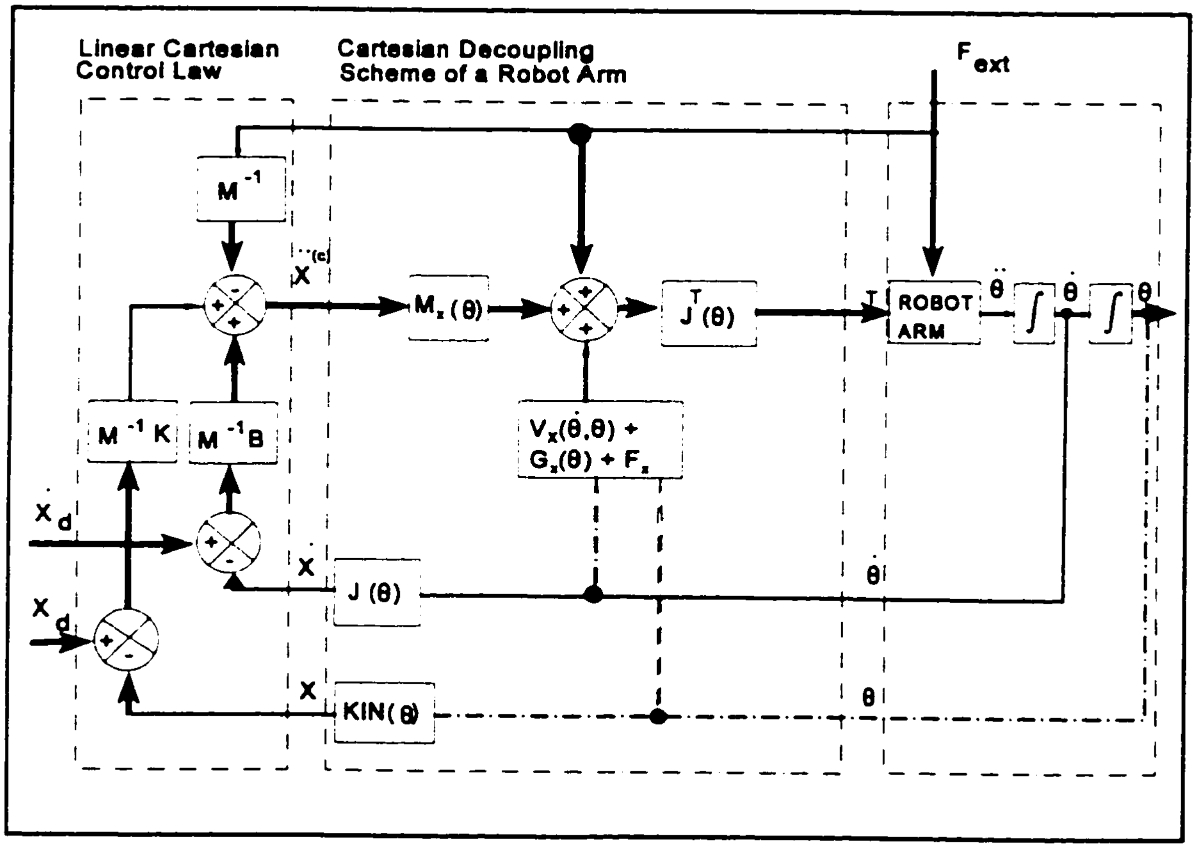

Figure 5-1 Two part(original) Impedance controller

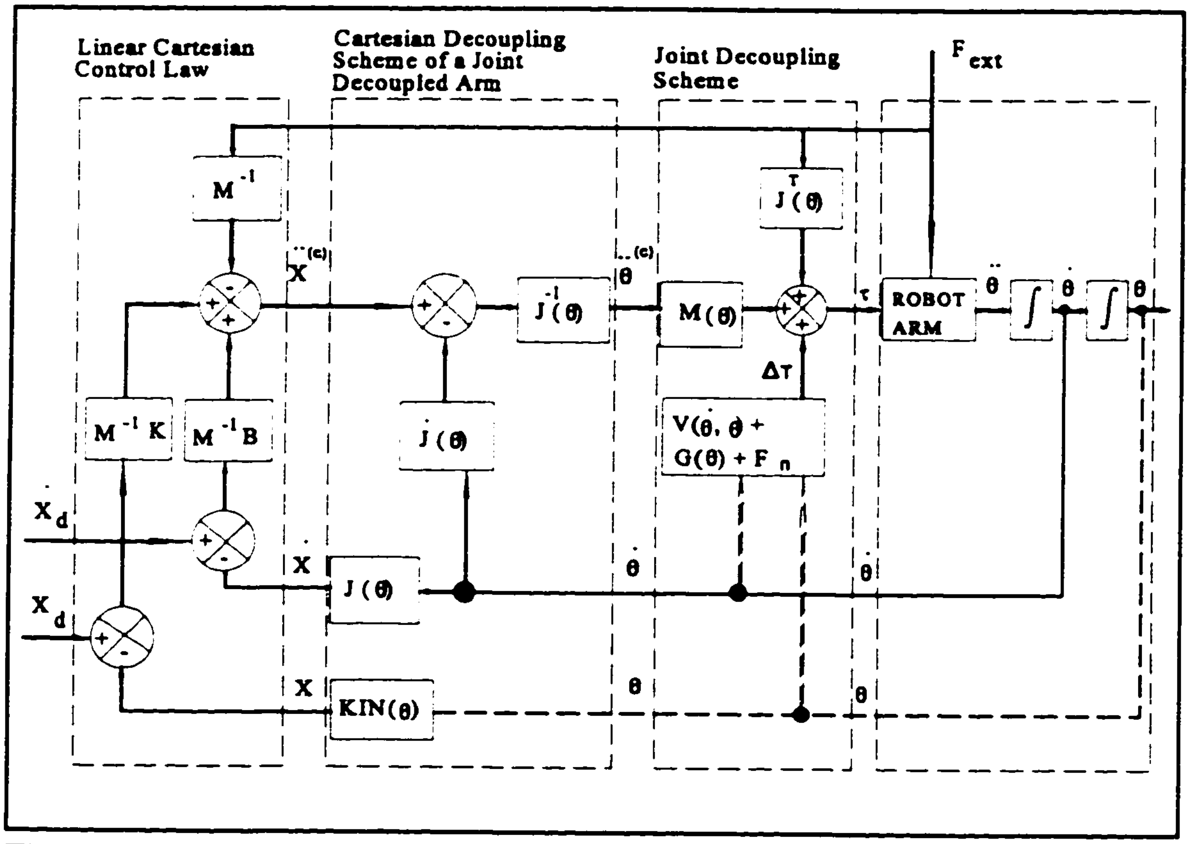

Figure 5-2 Three part Impedance controller

5.1.3.1 Computational complexity issue of a three-part control scheme

The application of a Cartesian two-part control law represented in the figure (5-1) requires, besides the kinematics computations present in all Cartesian control schemes, extensive computations of the highly nonlinear terms $M_x(\theta)$ and $V_x(\theta,\dot{\theta})$. Compared with Cartesian decoupling schemes, joint decoupling schemes are computationally less intensive.

Equations for the proposed three-part control scheme (figure (5-2)), are shown in equations (5-24 to 5-26). In these equations, F_{ext} no longer influences position control commands relating to contact motion. It should be noted that no desired contact force is considered in this scheme. Equation (5-24) would, however, require a desired impedance term M equal to the configuration dependent term $M_x(\theta)$.

In those cases where a configuration dependent mass term is not required to produce the desired impedance, the computation of $M_x(\theta)$, as shown in equation (5-27), is not particularly difficult since it uses matrices needed in decoupling computations. Moreover, in the case that all elements of vector X are defined regarding the inertial frame, $M_x(\theta)$ is a diagonal matrix with configuration dependent diagonal elements.

It will be observed that the two-step control structure, shown in the figure (5-1), has been replaced in the figure (5-2) by a three-step control structure.

This three-part control structure consists of the following:

(a) Linear Cartesian control law,

(b) Cartesian decoupling of a joint decoupled arm

(c) Joint decoupling

Part (b), in this scheme, requires only manipulations of the Jacobian matrix, (i.e., only kinematics computations). When no contact force exists ($F_{ext}=0$), this scheme emulates a Resolved-Acceleration Control where PD gains can be interpreted as $M^{-1}K$ and $M^{-1}B$ [81,82].

The Artificial Impedance approach brings extra compensation for the contact force and a physical interpretation of the PD gains.

Resolved Acceleration control, in fact, did not produce the development of collision avoidance schemes. This is probably because Cartesian control of the actuators is not incorporated in the overall dynamic model. When Cartesian control is incorporated, the actuator effects are replaced by passive artificial impedances in Cartesian space.

5.1.3.2. Computational efficiency of the three-part control scheme for a robot impedance controller

Until now, the impedance controller has been based on a two-part control scheme consisting of, a Cartesian nonlinear compensator part and a linear control law part. In the nonlinear compensator part, a very complex set of computations is required and both kinematic and dynamic parameters are involved.

In the recently proposed three-part control scheme, the nonlinear compensator is split into two further parts, the joint nonlinear compensator and the Cartesian nonlinear compensator for a joint decoupled robot arm. The advantage of this scheme arises from the fact that the Cartesian nonlinear compensator contains only kinematics parameters while the joint nonlinear compensator contains all the dynamic parameters.

Given the robust joint nonlinear compensators proposed in recent years, this separation of parameters increases the robustness of the scheme. Computational complexity is also reduced in the three-part control scheme such that when the configuration changes, updating of various computational blocks can be achieved at different rates. In a two-part control scheme, the highest updating rate, based on the aggregation of computations in a few blocks, dictates the updating rate.

Comparison of computational efficiency between two and three-part impedance controllers, as displayed in the table (5-1), are derived from equations (5-17 to 5-23) for two-part impedance control and equations (5-24 to 5-26) for three-part impedance control.

Computation of Part (a), (Linear Cartesian Control Law), and external force(f_{ext}) was not considered, because they are the same in both controllers. Also, equations(5-17 to 5-23) can be simplified and considered separately in the table(5-1). The table (5-1) shows that the three-part controller produces a better computation efficiency when compared with the two-part controller, because the total number of computations are less than the other two methods. In this comparison the matrices related to any of the described matrix(ie. Inverse of Jacobian) are assumed to be equal in computational time for all three methods.

Description	Two Part Imp. Controller		Three Part Imp.
	Original	Simplified	Controller
Inverse of Jacobian (calculation)	1	1	1
The transpose of Inverse of Jacobian (Calculation)	1	0	0
Derivative of Jacobian (Calculation)	1	1	1
Inverse of Jacobian (used)	2	2	1
The transpose of Inverse of Jacobian (used)	4	0	0
Derivative of Jacobian (used)	1	1	1
Matrix-Matrix Multiplication	2	0	0
Matrix-Vector Multiplication	8	5	3
Vector-Vector Addition	4	4	4
Total	24	14	11

Table 5-1 Comparison of two-part and three-part impedance controllers

5.1.4 Simulation results

The Impedance controller has been previously analytically tested [79]. In this report only the experimental results for an analytical model-based impedance controller are presented.

5.1.5 Experimental results

Experimental testing was carried out on a two degree of freedom planar robot arm as described in the chapter (3). Two tests have been carried out at this time are:

Test #1:

This test has been done to check the impedance controller's capabilities in directing the robot end-effector from one point to another in an almost straight line trajectory. The results of this test are shown in the figure (5-3). As shown in the figure (5-3) the end-effector was able to create an almost straight trajectory line (a) from point P_1 to point P_2 and then creates line (b) from point P_2 to point P_1. It should be noted that these are early results and have since been improved, as will be shown in a later part of this chapter.

Test #2

This test was done to show the performance of the impedance controller in obstacle avoidance. As shown in the figure (5-4), end-effector creates on-line a safe trajectory between two points to avoid the single point obstacle (o) successfully . Line (a) represents the end-effector trajectory from point P_1 to point P_2 and line (b) is trajectory from point P_2 to point P_1. For obstacle avoidance the same algorithm as shown in reference [79] was used.

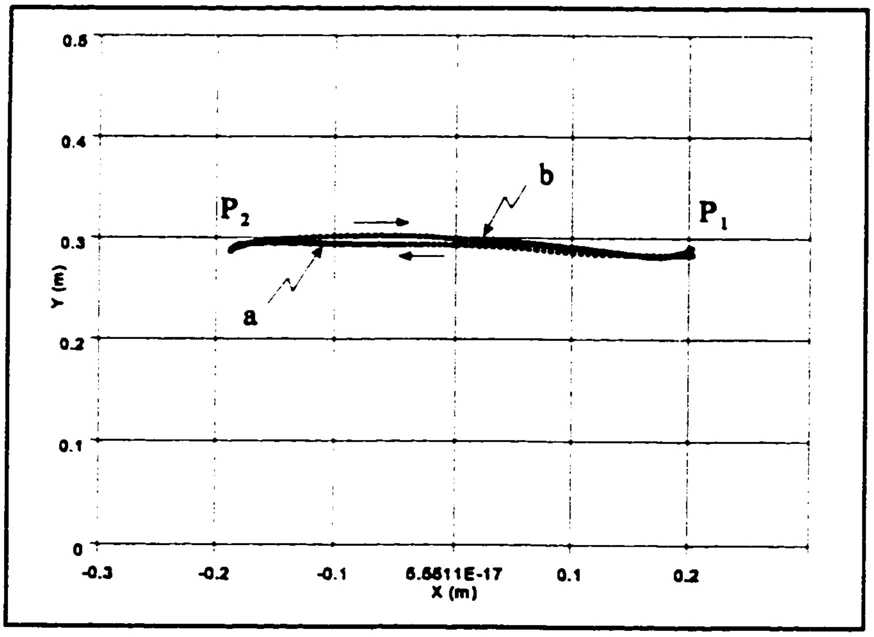

Figure 5-3 Impedance control direct the end-effector from point P_1 to P_2, then from point P_2 to P_1(experimental results).

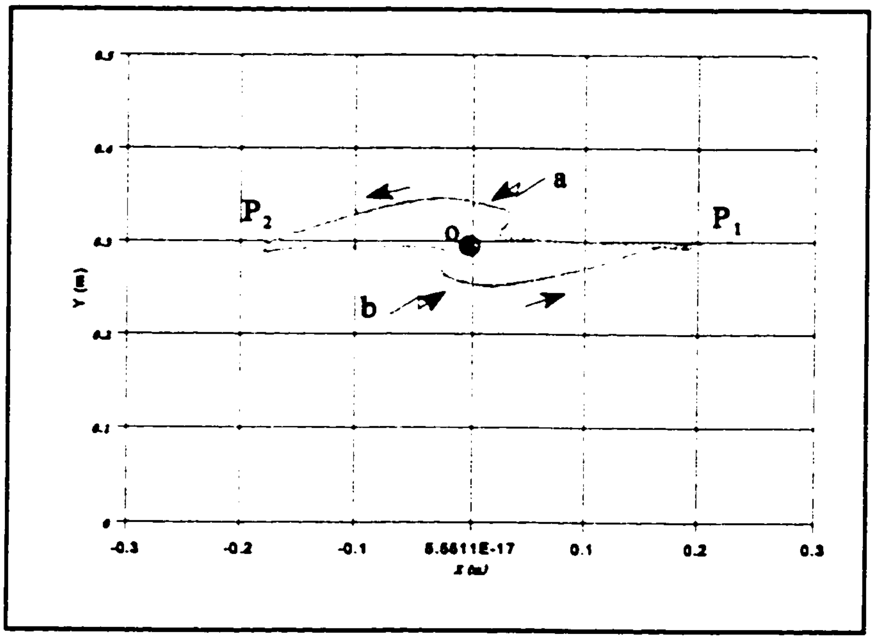

Figure 5-4 Obstacle avoidance by a two DOF planar robot(experimental results).

5.2 Impedance Control of a Dual-Arm Robot

Modelling of all reaction forces on end-effectors and a payload is complex and approximate. To simplify these models and be robust in the cases of shape, size, weight, and environmental changes, a simple model is required. This thesis proposes a model of virtual impedance between two end-effectors. This virtual impedance should be strong enough to replace kinematic and dynamic constraints between the two end-effectors when handling a payload used for simulation and experimentation. Simulations and experimental work have been carried out to test these phenomena for elastic object handling. However, for heavy, more flexible, and enforced objects, forces between the end-effectors and payload must be compensated, before impedance between the end-effectors can be created. Figure (5-5) shows the artificial spring and damper between two end-effectors.

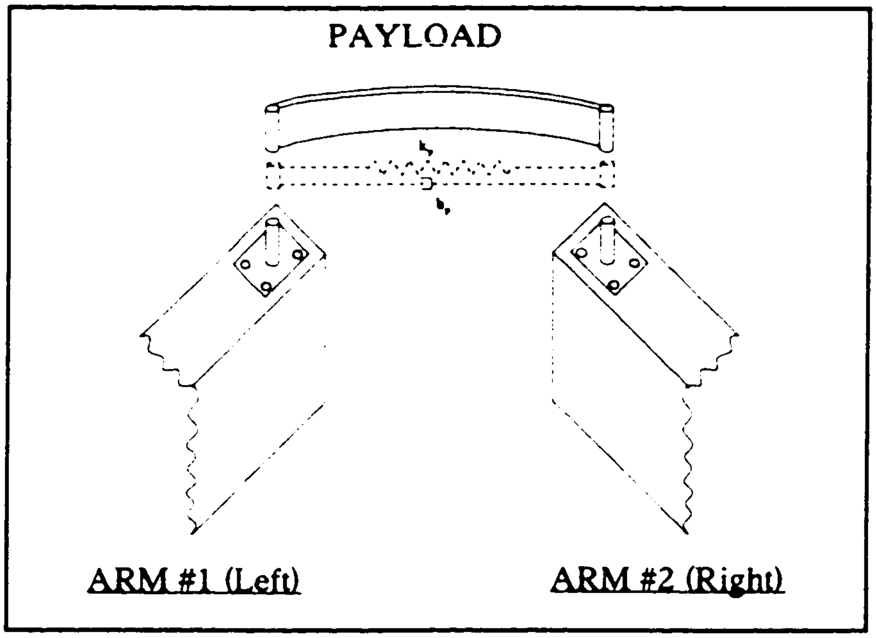

Figure 5-5 Artificial impedance between two end-effectors

Impedance between the end-effectors is created by an artificial spring(k_p=1200 N/m)and damper(b_p=20 N/m/s). The resulting forces on the end-effector, (referring to figure(4-4)), can be postulated as follows:

for $L < L_C$:

$$\begin{cases} f_p^L = k_p \left[\left(|X_b^{LR}| - L_C \right) \dfrac{X_b^{LR}}{|X_b^{LR}|} \right] + b_p \left(\dot{X}_b^{LR} - \dot{X}^R \right) \\ f_p^R = -f_p^L \end{cases} \qquad \textbf{(5-28)}$$

for $L > L_C$

$$\begin{cases} f_p^L = k_p \left[\left(|X_b^{LR}| - L_C \right) \dfrac{X_b^{LR}}{|X_b^{LR}|} \right] + b_p \left(\dot{X}_b^{LR} - \dot{X}^R \right) \\ f_p^R = -f_p^L \end{cases} \qquad \textbf{(5-29)}$$

for $L = L_C$

$$\begin{cases} f_{x_p} = 0 \\ f_{y_p} = 0 \end{cases} \qquad \textbf{(5-30)}$$

and:

$$f_{ext} = f_p \qquad \textbf{(5-31)}$$

These forces are applied on the end-effector and have been transferred to joint space torques by using the Jacobian matrix as shown in equations (4-29 and 4-30).

5.2.1 Stability Analysis

This section will show the stability of the impedance controller for each arm in a multi-arm system. The impedance controller was originally presented in the following format [79]:

$$M\,\ddot{e}_x + B\,\dot{e}_x + K\,e_x = 0 \qquad \textbf{(5-32)}$$

where:

$e_x = x_d - x$ (position error of end-effector, current and desired position)

$\dot{e}_x = \dot{x}_d - \dot{x}$ (velocity error of end-effector, current and desired velocity)

$\ddot{e}_x = \ddot{x}_d - \ddot{x}$ (acceleration error of end-effector, current and desired acceleration ($\ddot{x}_d = 0$))

$F_{ext} = 0$ free motion, no external forces

By simplifying equation (5-32), the computed acceleration command($\ddot{x}$) can be calculated as:

$$\frac{B}{M}\,\dot{e}_x + \frac{K}{M}\,e_x = \ddot{e}$$
$$(B\,M^{-1})\,\dot{e}_x + (K\,M^{-1})\,e_x = \ddot{x} \qquad \textbf{(5-33)}$$

(computed acceleration is also shown in the figure (5-2) as $\ddot{x}^{(c)}$)

The error state space vector is:

$$e = [e_x^T \quad \dot{e}_x^T]^T \qquad \textbf{(5-34)}$$

where:

$e_x = [e_{x_1}^T \; e_{x_2}^T \ldots e_{x_n}^T]^T$ for i=1 ... n

$$\dot{e}_x = [\dot{e}_{x_1}{}^T \ \dot{e}_{x_2}{}^T \ \dots \ \dot{e}_{x_n}{}^T]^T \qquad \text{for i=1 ... n}$$

The control law from the figure (5-2):

$$\tau_i = M(\theta)\, J_i^{-1}\, (\ddot{x}_i - \dot{J}_i\, \dot{\theta}_i) + C_i \qquad \textbf{(5-35)}$$

where:

$$C_i = V_i(\theta, \dot{\theta}) + G_i(\theta) + f_i(\dot{\theta})$$

$$\ddot{x}_i = K_m\, e_{x_i} + B_m\, \dot{e}_{x_i}$$

$$K_m = K\, M^{-1}$$

$$B_m = B\, M^{-1}$$

The robot dynamic equation is:

$$\tau_i = M(\theta)\, \ddot{\theta}_i + C_i \qquad \textbf{(5-36)}$$

For stability analysis Lyapunov theory was used [81]:

<u>Theorem[81]:</u>

If the matrices M and B and K or B_m and K_m in control the law in equation (5-35) are symmetric positive definite and the Jacobian of each robot is nonsingular, then the system is asymptotically stable.

<u>Proof:</u>

The system is asymptotically stable by means of Lyapunov when the following conditions are satisfied:

1) $V(e_x)$ is positive definite

2) $V(e_x)$ --> ∞ when |e| --> ∞

3) $\dot{V}(e_x)$ is negative definite

where the Lyapunov function is:

$$V(e_x) = \frac{1}{2}\, \dot{e}_x^{\,T}\, M\, \dot{e}_x + \frac{1}{2}\, e_x^{\,T}\, K_m\, e_x \qquad \textbf{(5-37)}$$

where: M Is an identity matrix

Since K_m is symmetric positive definite, therefore, V(e) is positive definite and this would satisfy the first condition. Also, as |e| --> ∞ the Lyapunov function goes to infinity ($V(e_x)$ --> ∞) and this would satisfy the second condition. A derivative of the Lyapunov function has to be taken for third condition, therefore:

$$\dot{V}(e_x) = \frac{1}{2}\, \dot{e}_x^{\,T}\, M\, \ddot{e}_x + \frac{1}{2}\, \dot{e}_x^{\,T}\, K_m\, e_x \qquad \textbf{(5-38)}$$

From equation (5-35) and equation (5-33) $K_m\, e_x$ can be calculated as:

$$\tau_i = M(\theta)\, J_i^{-1}\, (K_m\, e_x + B_m\, \dot{e}_x - \dot{J}_i\, \dot{\theta}_i) + C_i \qquad \textbf{(5-39)}$$

therefore:

$$K_m\, e_x = -\, B_m\, \dot{e}_x + \dot{J}_i\, \dot{\theta}_i + J_i\, M^{-1}(\theta)\, (\tau_i - C_i) \qquad \textbf{(5-40)}$$

τ_i can be replaced by equation (5-36):

$$K_m\, e_x = -\, B_m\, \dot{e}_x + \dot{J}_i\, \dot{\theta}_i + J_i\, \ddot{\theta}_i \qquad \textbf{(5-41)}$$

since $(\ddot{x}_i = J_i \ddot{\theta}_i + \dot{J}_i \dot{\theta}_i)$:

$$K_m e_x = - B_m \dot{e}_x + \ddot{x}_i \quad \textbf{(5-42)}$$

and from definition of $(\ddot{e}_x = \ddot{x}_d - \ddot{x})$, where $\ddot{x}_d = 0$:

$$K_m e_x = - B_m \dot{e}_x - \ddot{e}_x \quad \textbf{(5-43)}$$

if the equation (5-43) is put into equation (5-38), then:

$$\dot{V}(e_x) = \frac{1}{2} \dot{e}_x^T M \ddot{e}_x + \frac{1}{2} \dot{e}_x^T (-B_m \dot{e}_x - \ddot{e}_x) \quad \textbf{(5-44)}$$

Since M is an identity matrix, therefore:

$$\dot{V}(e_x) = - \dot{e}_x^T B_m \dot{e}_x \quad \textbf{(5-45)}$$

From the equation (5-45) the third condition is satisfied. Therefore, the controller is asymptotically stable.

5.2.2 Simulation Results

Simulations were carried out to test the effect of applying Impedance control on dual-arm robot, before pursuing them experimentally. Impedance controller was used to control both arms and direct them from their initial to desired positions. The results of these simulations are as follow:

Simulations of a robot handling a string in two phases

The string is modelled as a linear spring with a damper in parallel, having a spring constant of 1.0 N/m and a damping constant of 1.0 N/m/s. The end-effectors are located initially at R_i^L and R_i^R, respectively, and move in the approaching phase to R_{d1}^L and R_{d1}^R where the string is grasped (Figure (5-6)). In the next phase the string is transported to R_{d2}^L and R_{d2}^R where Left arm reaches the final position. Subsequently, only Right arm moves and rotates the string until it reaches its final position in R_{d3}^R. The length of the string varies in time within ± 1%. Further simulation, in which the sampling time is reduced from 0.005 s to 0.001 s, will lead to a reduction of the length variation by a factor of 50 percent. These simulations show that the impedance-based controller can coordinate the motion of a dual-arm robot in a sequential manner. Improvements to the control scheme, discussed in the next section, eliminate the sequential nature of the approach and result in a smooth simultaneous motion of the two arms to the final positions.

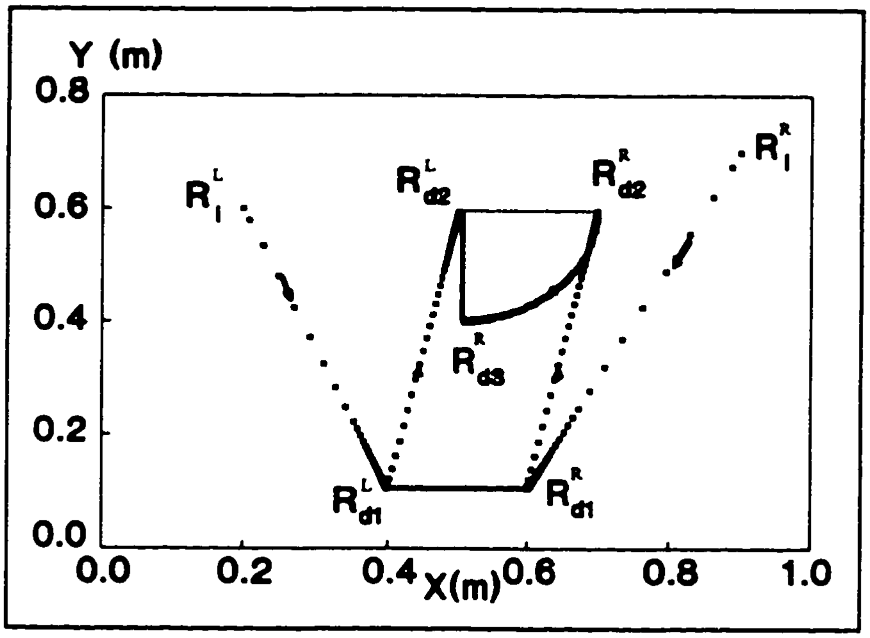

Figure 5-6 Two phase approach by dual-arm robot(simulation results).

Simulations of a robot handling a string in one phase

The string is modelled, when stretched, as a very compliant spring and when compressed produces no reactive forces on the end-effectors. Only "the actual" object handling phase is simulated here, since the same coordination problems exist for all tasks during the approaching phase of the separate arms. The initial positions of the end-effectors are R_i^L and R_i^R, respectively, where the object is assumed grasped (figure (5-7)). The artificial impedances created between the initial positions of the end-effectors and their desired positions R_d^L and R_d^R, respectively, generated straight line trajectories, in the case of a string that when compressed does not produce any reactive forces. The distance between the two end-effectors varies during the motion with a minimum value of 0.14 m. This leads to a large sag of the string which is equal to 0.2 m (figure (5-8)). An artificial impedance approach overcomes the lack of accounting for reactive forces problem when compressing a string in length, by creating an artificial impedance in parallel with the string (figure (5-5)).

Simulations were performed for an artificial parallel impedance with a spring constant of 1.0 N/m and a damper constant of 2.0 N/m/s, with constrain on Left arm to follow straight line trajectory. Simulation results in the figure (5-9) show that the Left's end-effector maintains a straight line trajectory while the follower's end-effector trajectory is generated such that the string has practically no sag during motion, as shown in the figure (5-10). Therefore, using artificial Impedance between the end-effectors was successful. The dual-arm robot was tested for different trajectories and produced the same results with practically no sag during the motion.

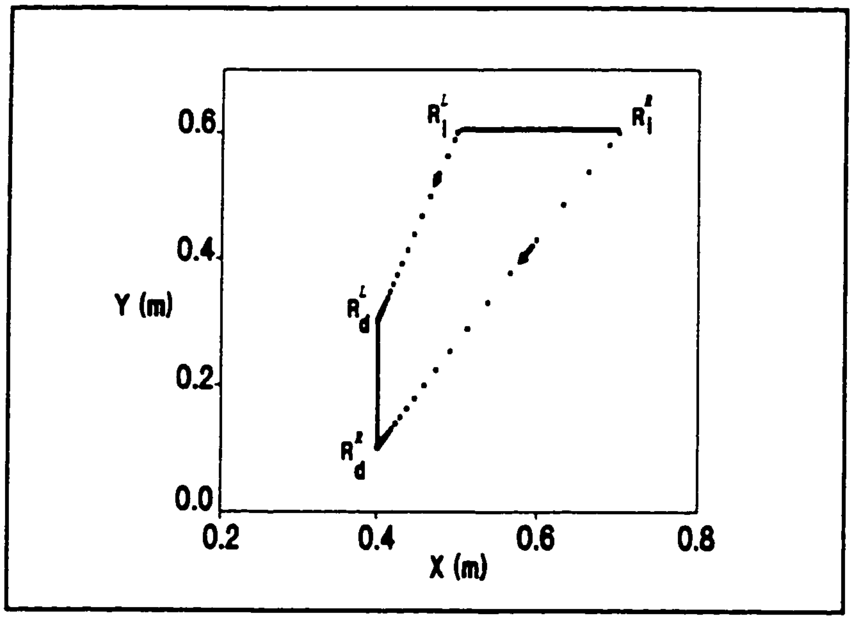

Figure 5-7 Trajectories of both arms from initial to desired position, while handling a flexible object without artificial impedance between two end-effectors (simulation results).

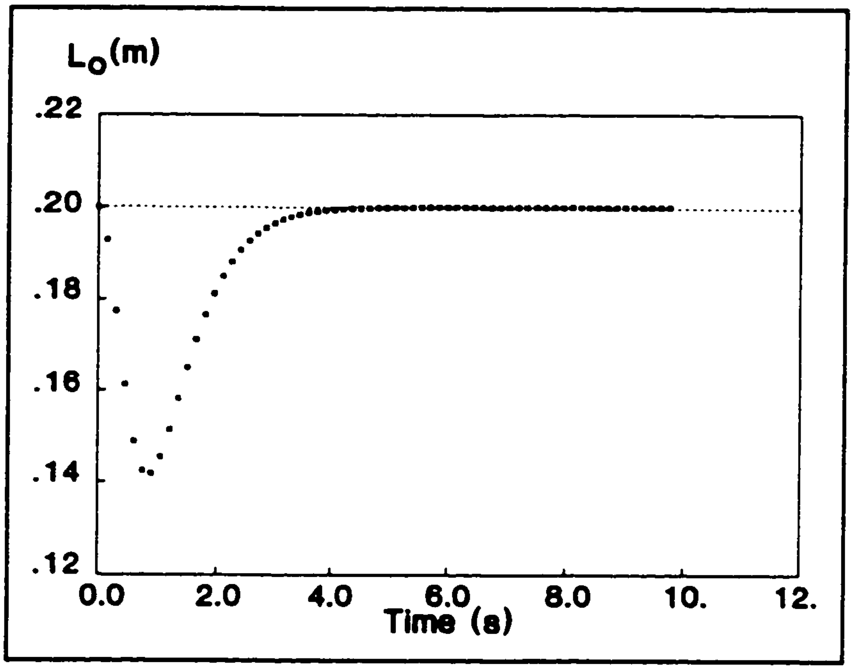

Figure 5-8 Time variation of the distance between the two end-effectors for the case simulated in figure (5-7)(sim- ulation results).

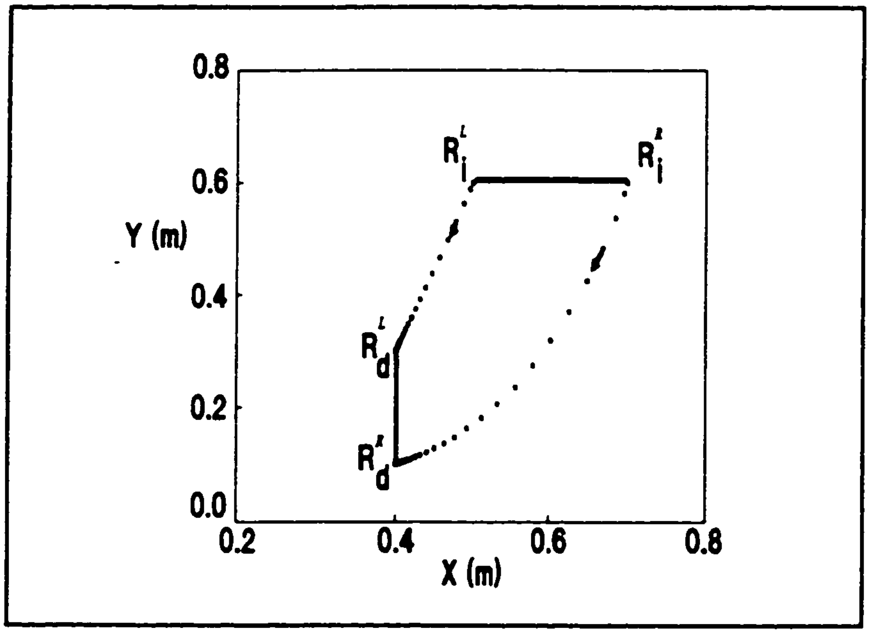

Figure 5-9 Dual-arm robot trajectories from initial to desired position while handling a flexible object with artificial impedance between the two end-effectors (simulation results).

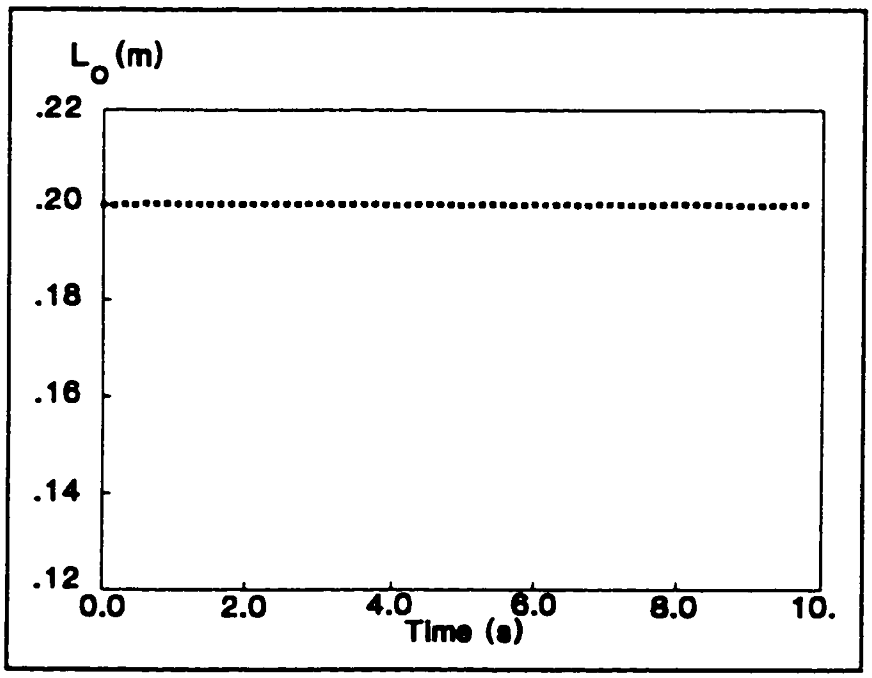

Figure 5-10 Time variation of the dist- ance between the two end-effectors for the case simulated in figure (5-9) (simulation results).

Simulation Results of Collision Avoidance in dual-arm robot motion coordination

Figures (5-11,5-13,5-15,5-17) show different cases of obstacle avoidance. Basically, by creating a repulsive field around the obstacle [78,79], the controller can protect the end-effectors of the dual-arm robot and the string from colliding with the obstacles. Robot links are not protected from obstacle collisions in planer simulations. Avoiding posture collision of the links is more complicated than simply considering a repulsive force field around the robot joints. Using repulsive forces around the links would only work in some cases. In general redundant degrees of freedom in the robot can contribute greatly to obstacle and posture collision avoidance. This is especially true when the obstacle is located at a desired orientation from one of the robot arms. Obstacle and posture collision avoidance, however, were not major topics of this work. Testing has therefore been kept simple. Only the end-effector and payload had been protected from obstacle collision, during simulations.

At this stage, it was assumed that the repulsive force would only be created on the end-effector of the arm that is closest to the obstacle and does not affect the other arm, except the test in figure (5-17). In this test (figure(5-17)) the arm were not constrained to follow a straight line trajectory. Figure (5-11), shows that arm two(Right) was able to avoid the obstacle and reach the target with practically no sag of the string during the motion, as shown in the figure (5-12).

Figures (5-13) and (5-15) show two further examples of obstacle avoidance. In these cases, arm two(Right) could avoid the obstacle but it could not reach the target. This is because of the requirements for no string sag and no change in arm one(Left) trajectory, as shown in figures (5-14) and (5-16). The string was also protected from colliding to obstacle in these simulations.

Figure (5-17) shows another case in which arm one(Left) had to avoid the obstacle, without producing any string sag, as shown in the figure (5-18). In this case both arms changed their trajectory in order to reach the target without colliding into any obstacle, and there is no sag in the string. These results proved further that artificial Impedance between the end-effectors were able to minimize the sag in object successfully.

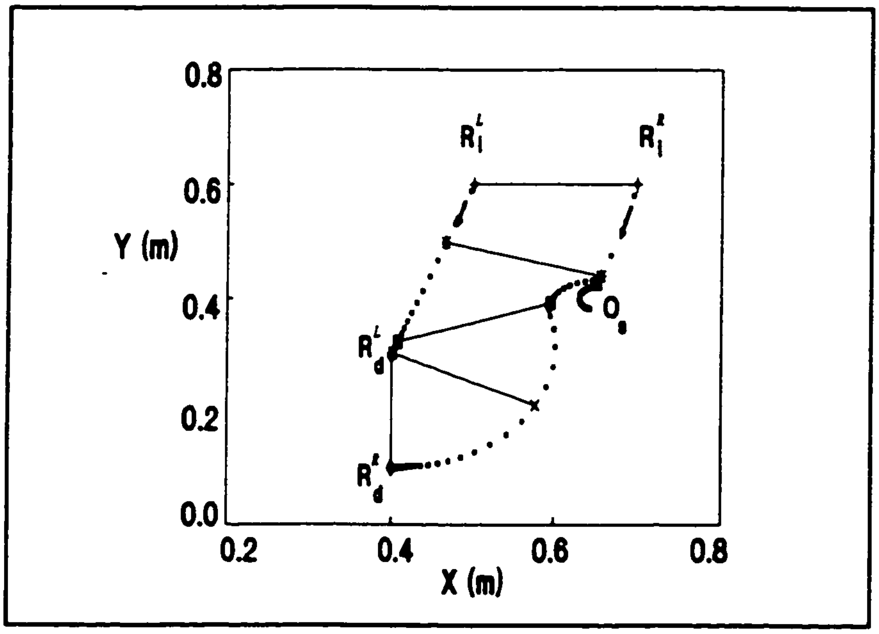

Figure 5-11 Simulation of a robot hand- ling a string during avoiding an obsta- cle (simulation results).

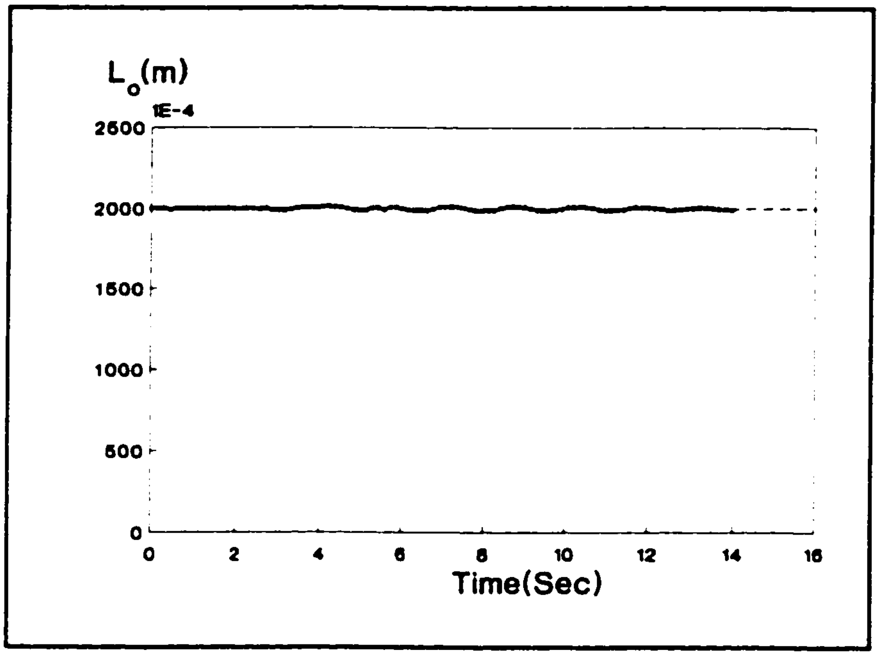

Figure 5-12 Time variation of the dist- ance between the two end-effectors for the case simulated in figure (5-11) (simulation results).

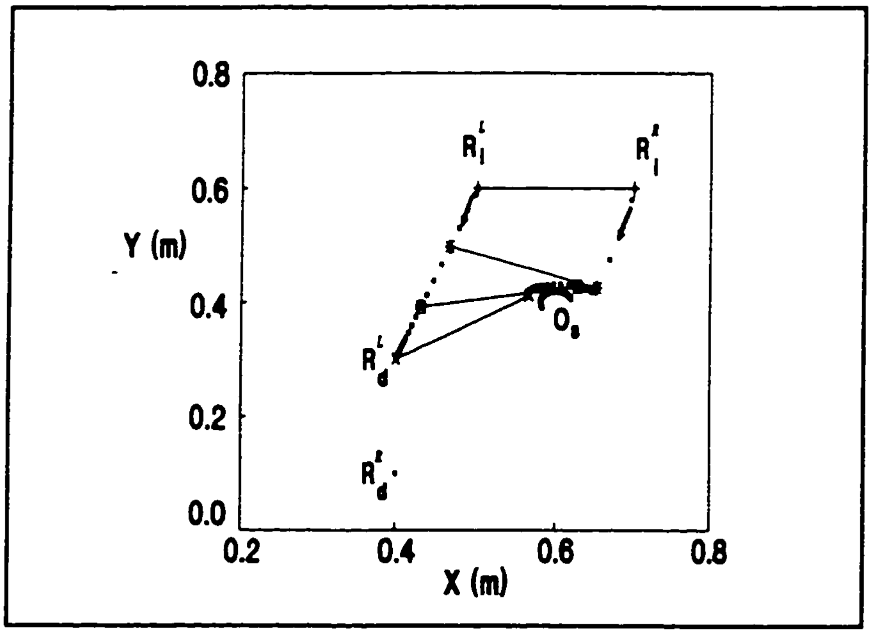

Figure 5-13 Dual-arm handling a string while avoiding an obstacle (simulation results).

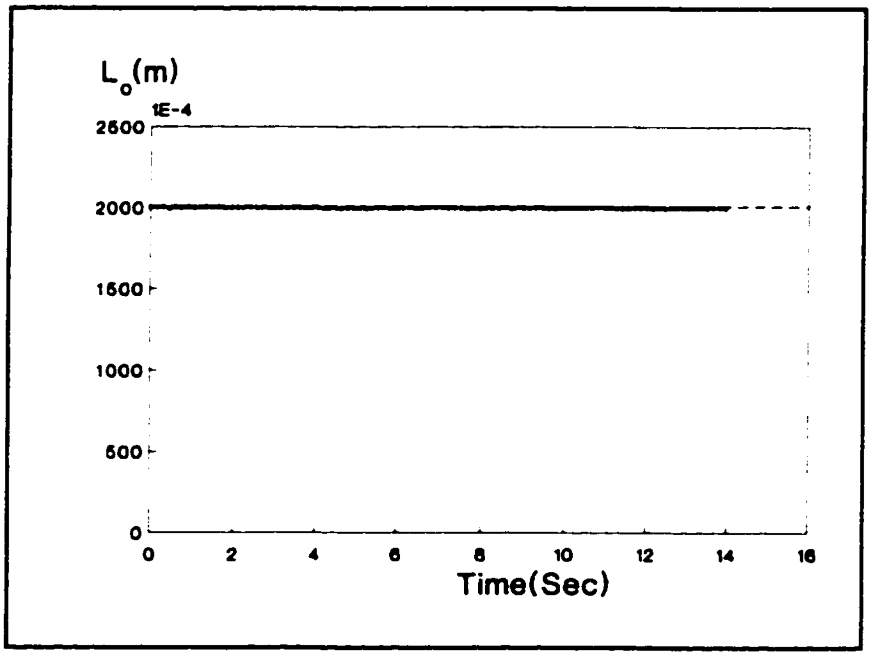

Figure 5-14 Time variation of the dist- ance between the two end-effectors for the case simulated in figure (5-13) (simulation results).

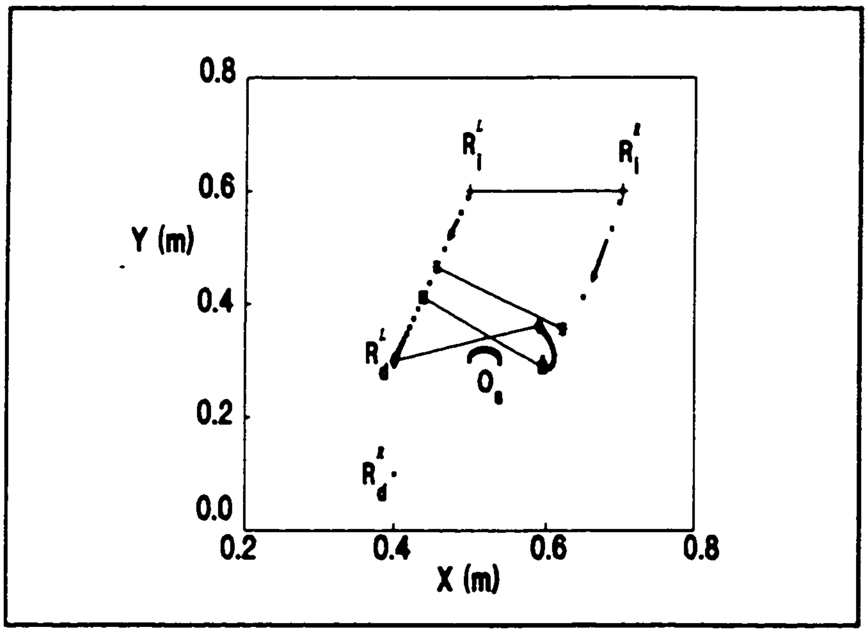

Figure 5-15 Dual-arm robot handling a string while avoiding an obstacle (simulation results).

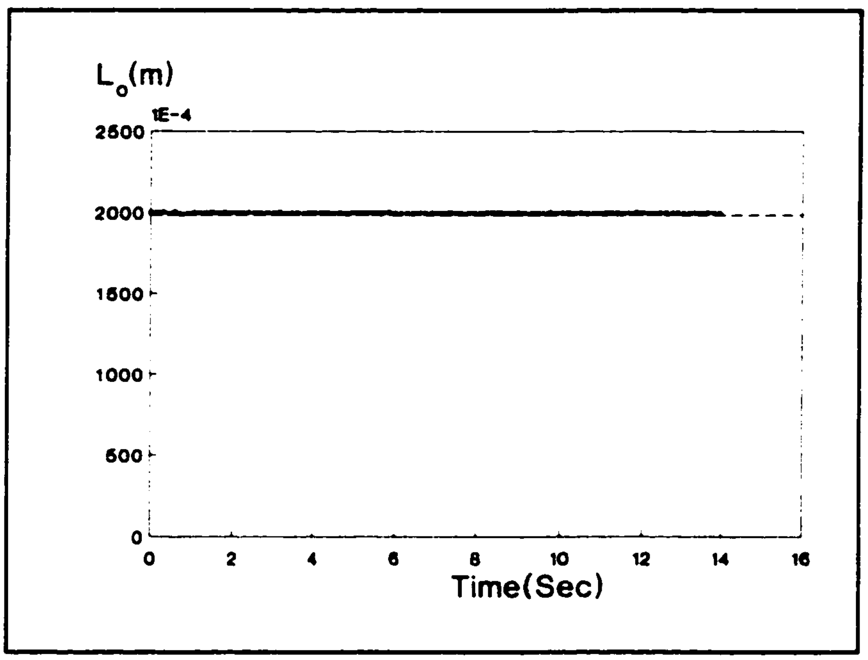

Figure 5-16 Time variation of the dist- ance between the two end-effectors for the case simulated in figure (5-15) (simulation results).

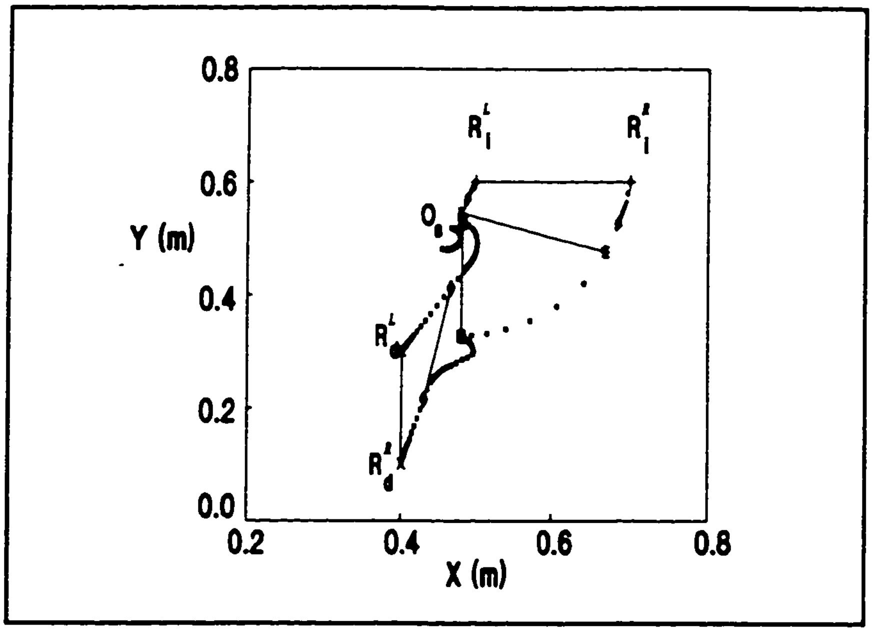

Figure 5-17 Dual-arm robot handling a string while avoiding the obstacle (simulation results).

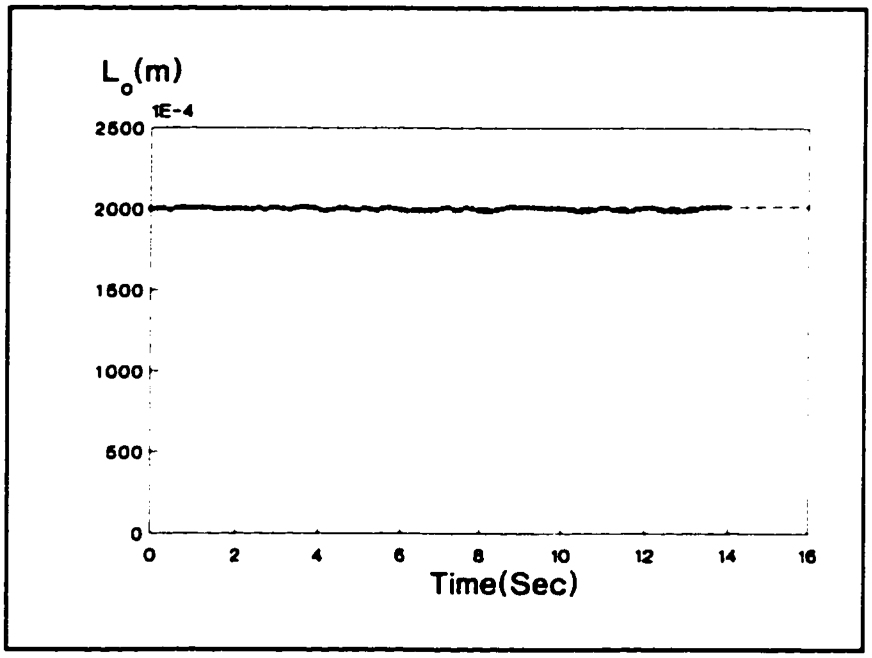

Figure 5-18 Time variation of the dist- ance between the two end-effectors for the case simulated in figure (5-17) (simulation results).

5.2.3 Experimental Results

The artificial Impedance between the end-effector showed promising results in simulations. At this stage the experimental testing is required to prove this phenomena. Therefore, the experimental test-bed which is described in chapter 3 was used.

- Dual-arm control and string handling:

The following notations were used in tests carried out on the experimental dual-arm robot:

I^L	Initial position of arm #one (Left)
D^L	Desired position of arm #one (Left)
I^R	Initial position of arm #two (Right)
D^R	Desired position of arm #two (Right)
o	Obstacle
	The solid line between D^R & D^L or I^R & I^l is the payload.

- The payload(string) specifications:

The payload is a short elastic object approximately 2.35 cm in length, with a low stiffness coefficient. This object is shown in the figure (3-2).

Test #1:

An impedance controller was used to control a dual-arm robot handling a flexible object. Figures (5-19) and (5-20) show how Right arm reaches the target following an almost straight line trajectory and how Left arm successfully corrects its trajectory to keep distance between the end-effectors constant.

Test #2:

Impedance controller proved its capability of controlling a dual-arm robot handling a flexible object. This controller has been tested for obstacle avoidance. Figure (5-21) shows that impedance controller managed to direct the dual-arm to a new trajectory in order to avoid the obstacle. In these tests repulsive forces are created around the obstacle (a known single point) and these forces only affected the Left arm.

Figure (5-21) shows that Right arm followed a straight line trajectory to the desired point while the left arm created a new, safe, trajectory to the desired point while keeping distance between the end-effectors constant.

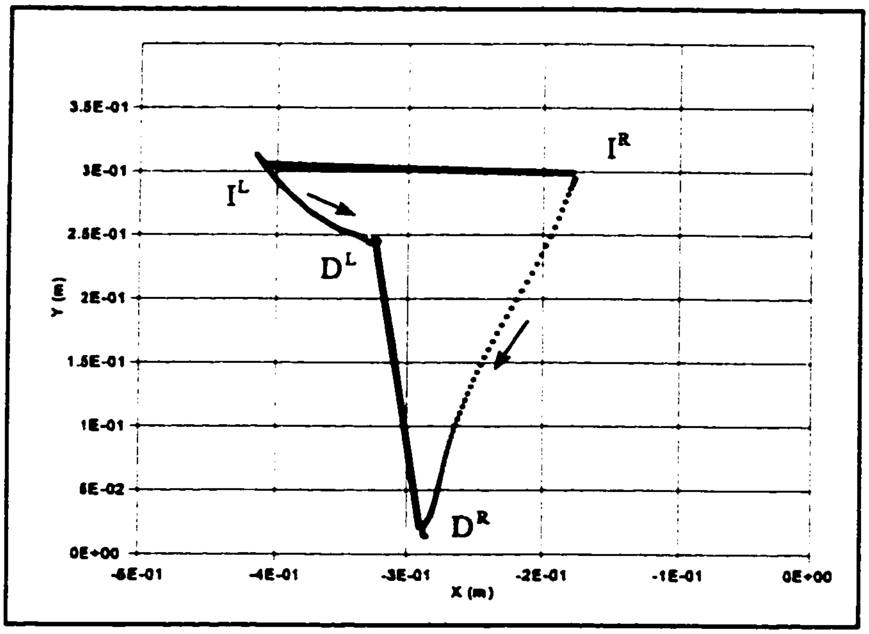

Figure 5-19 Trajectory generation and object handling by impedance controller for dual-arm robot (experimental results).

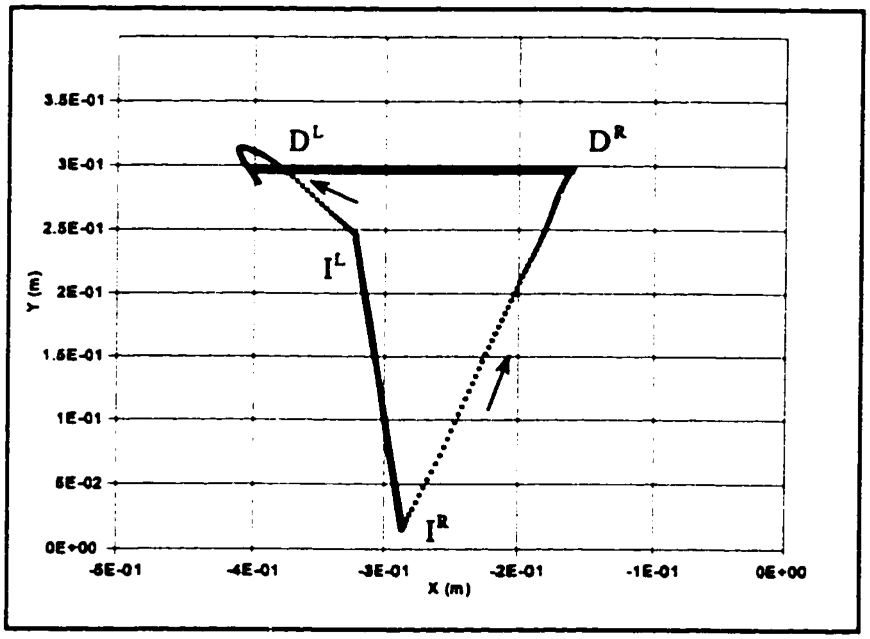

Figure 5-20 This is exactly the same t- est as test on figure (5-19) with different initial and desired positions (experimental results).

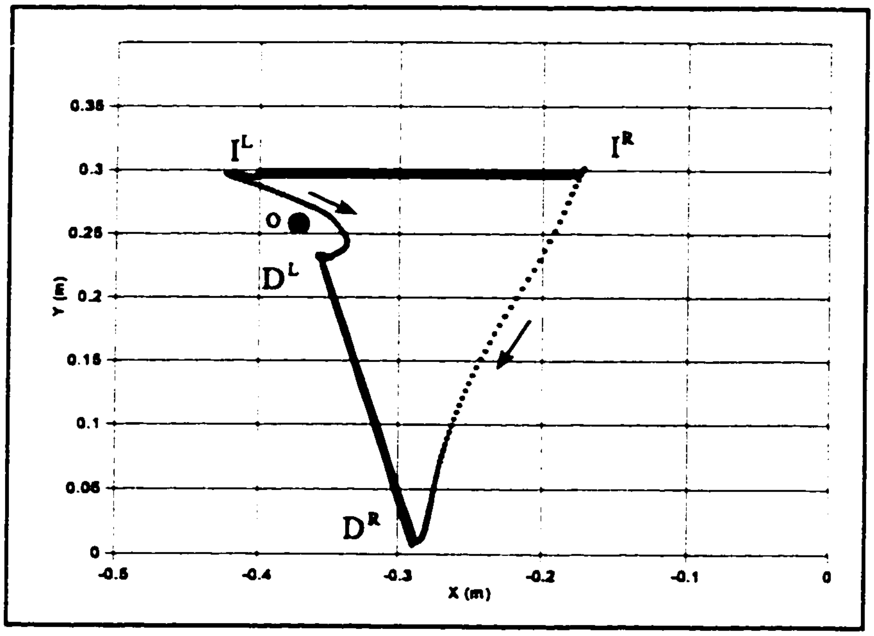

Figure 5-21 Dual-arm robot avoid an obstacle by using impedance controller (experimental results).

5.3 Perturbation Observer

Impedance control scheme is one of the computed torque control schemes, and it has disadvantages such factors as unknown dynamics, friction, and coupled inertial terms,etc. Therefore, to improve robustness and efficiency of the Impedance controller, a perturbation observer was integrated to original three-part impedance control scheme.

Figure (5-22) displays, a three-part impedance controller, using an observer, that compensates noninertial dynamic terms in joint space. The impedance controller originally consists of three parts(as shown in the figure(5-2) and equations (5-24 to 5-27)):

- Linear Cartesian control law for end-effector Cartesian position X:

$$\ddot{X}^{(c)}=M^{-1}K(X_d-X)+M^{-1}B(\dot{X}_d-\dot{X})-M^{-1}F_{ext} \tag{5-46}$$

- Cartesian decoupling scheme of a joint decoupled arm:

$$\ddot{\theta}^{(c)}=J^{-1}(\theta)[\ddot{X}^{(c)}-\dot{J}(\theta)\dot{\theta}] \tag{5-47}$$

- Joint decoupling scheme:

$$\tau=\hat{M}(\theta)\ddot{\theta}^{(c)}+V(\theta,\dot{\theta})+F_n+G(\theta)+J^T(\theta)F_{ext} \tag{5-48}$$

where:

$\ddot{\theta}^{(c)}$ is the acceleration command

$\ddot{\theta}$ is the output acceleration

$\hat{M}(\theta)$ is the inertia matrix of the robot(based on estimated parameters)

$M(\theta)$ is the inertia matrix of the robot(based on actual parameters)

F_n is the unmodelled dynamics torque term

M is an inertia matrix, selected for the worst conditions of a particular task, and kept constant for all configurations of the arm.

F_{ext} is the end-effector reaction force from external interactions

For the robot dynamic equation:

$$\tau = M(\theta)\ddot{\theta} + V(\theta,\dot{\theta}) + F_n + G(\theta) + J^T(\theta)F_{ext} \tag{5-49}$$

Using motor torque output, given by equation (5-48) $\ddot{\theta}^{(c)} = \ddot{\theta}$ is obtained for $M(\theta) = \hat{M}(\theta)$ (i.e., joint decoupling). In the impedance controller scheme shown in the figure (5-2), an observer replaces the last four terms of the equation (5-48). The observer is given by[45,84] as follows:

$$\Delta\ddot{\theta}(t-1) = M^{-1}(\theta)\tau(t-1) - \ddot{\theta}(t-1) \tag{5-50}$$

From equation (5-48):

$$M^{-1}(\theta)\ \tau\ -\ \ddot{\theta}^{(c)} = M^{-1}(\theta)[V(\theta,\dot{\theta}) + F_n + G(\theta) + J^T(\theta)F_{ext}] \tag{5-51}$$

and by approximating $\Delta\ddot{\theta}(t) \approx \Delta\ddot{\theta}(t-1)$ of equation (5-50). The acceleration perturbation observer $\Delta\ddot{\theta}$ estimates the terms:

$$\Delta\ddot{\theta} = M^{-1}(\theta)[V(\theta,\dot{\theta}) + F_n + G(\theta) + J^T(\theta)F_{ext}] \tag{5-52}$$

such that the computed torque is estimated from equation (5-51) by:

$$\tau = M(\theta)(\ddot{\theta}^{(c)} + \Delta\ddot{\theta}) \tag{5-53}$$

Equations (5-49), (5-52) and (5-53) result again in $\ddot{\theta}^{(c)} = \ddot{\theta}$, which represents joint space exact linearization . Computational savings due to inclusion of the observer are obvious, when comparing equations (5-48) and (5-53).

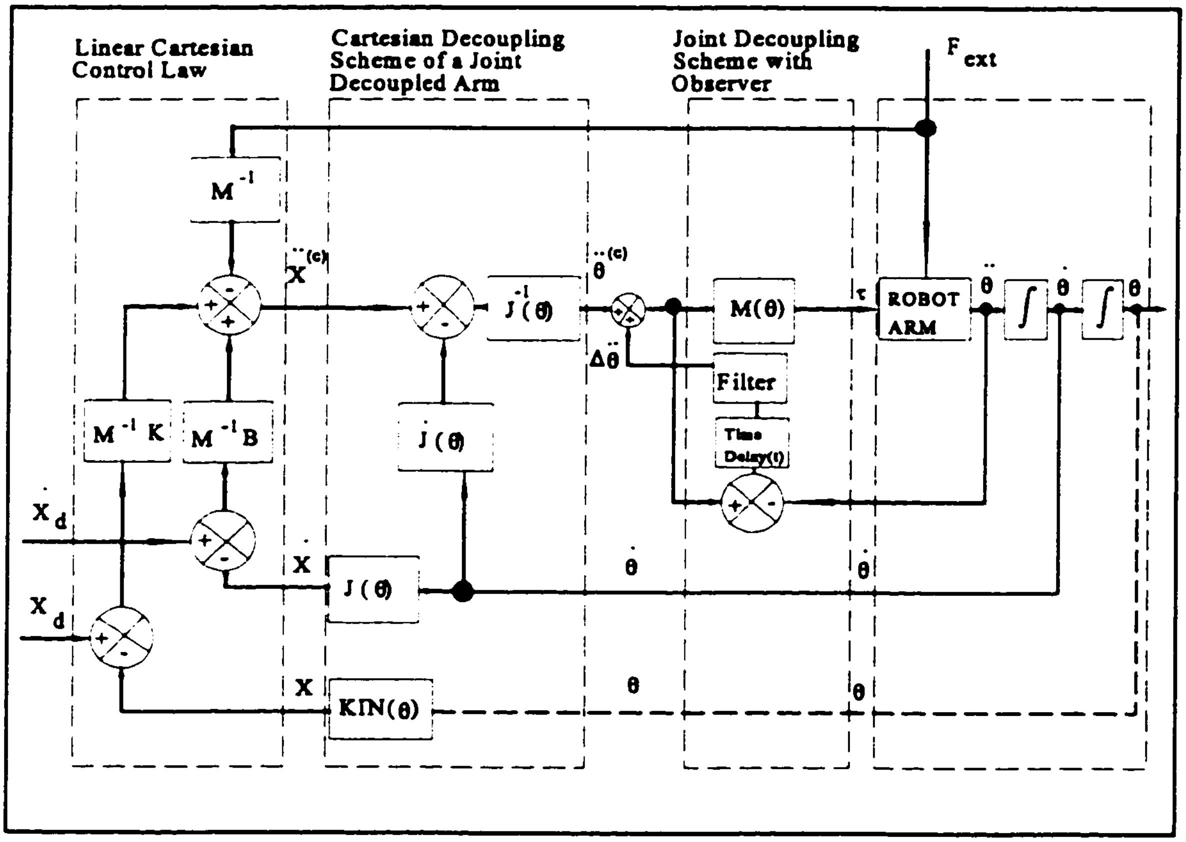

Figure 5-22 Three-part impedance controller with observer

5.3.1 Stability Analysis

This section will show the stability of the impedance controller with a perturbation observer. By referring to the block diagram in the figure(5-22), the linear control law is:

$$\ddot{x}^{(c)} = (B\,M^{-1})\,\dot{e}_x + (K\,M^{-1})\,e_x \tag{5-54}$$

or

$$\ddot{x}^{(c)} = K_m\,e_{x_i} + B_m\,\dot{e}_{x_i} \tag{5-55}$$

where:

$e_x = x_d - x$ (position error of end-effector, current and desired position)

$\dot{e}_x = \dot{x}_d - \dot{x}$ (velocity error of end-effector, current and desired velocity)

$K_m = K\, M^{-1}$ (constant proportional gain)

$B_m = B\, M^{-1}$ (constant derivative gain)

$\ddot{x}^{(c)}$ (computed acceleration)

and the Cartesian decoupling is (figure (5-22)):

$$\ddot{\theta}^{(c)} = J^{-1}\,(\ddot{X}^{(c)} - \dot{J}\dot{\theta}) \tag{5-56}$$

Joint decoupling based on the perturbation observer (figure (5-22)) is:

$$\tau = M(\theta)\,(\ddot{\theta}^{(c)} + \Delta\ddot{\theta}) \tag{5-57}$$

where:

$\Delta\ddot{\theta} = \ddot{\theta}^{(c)}(t-1) - \ddot{\theta}(t-1)$

$\ddot{\theta}^{(c)}$ is computed joint acceleration of the joints

$\ddot{\theta}$ is actual joint acceleration of the joints

$M(\theta)$ is the inertial term

The control law from the figure (5-22) is:

$$\tau = M(\theta)\,J^{-1}\,(\ddot{x}^{(c)} - \dot{J}\,\dot{\theta}) + M(\theta)\Delta\ddot{\theta} \tag{5-58}$$

Add equation (5-55) to equation (5-58) and simplify as follows:

$$\tau = M(\theta)\, J^{-1}\, ((K_m\, e_x + B_m\, \dot{e}_x) - \dot{J}\, \dot{\theta}) + M(\theta)\Delta\ddot{\theta} \qquad \textbf{(5-59)}$$

The robot dynamic equation is:

$$\tau_i = M(\theta)\, \ddot{\theta}_i + C_i \qquad \textbf{(5-60)}$$

where:

$$C_i = V_i(\theta,\dot{\theta}) + G_i(\theta) + f_i(\dot{\theta})$$

For stability analysis Lyapunov theory was used [81];

<u>Theorem[81]:</u>

If the matrices M and B and K or B_m and K_m in control law in equation (5-58) are symmetric positive definite and the Jacobian of the robot is nonsingular, then the system is asymptotically stable.

<u>Proof:</u>

The system is asymptotically stable by means of Lyapunov when the following conditions are satisfied:

1) $V(e_x)$ is positive definite
2) $V(e_x)$ --> ∞ when |e| --> ∞
3) $\dot{V}(e_x)$ is negative definite

where the Lyapunov function is:

$$V(e_x) = \frac{1}{2}\, \dot{e}_x^{\,T}\, M\, \dot{e}_x + \frac{1}{2}\, e_x^{\,T}\, K_m\, e_x \qquad \textbf{(5-61)}$$

where: M is an identity matrix

Since K_m is symmetric positive definite, therefore, $V(e_x)$ is positive definite and this would satisfy the first condition. Also, as $|e|$ --> ∞ the Lyapunov function goes to infinity ($V(e_x)$ --> ∞) and this would satisfy the second condition. A derivative of the Lyapunov function has to be taken for third condition, therefore:

$$\dot{V}(e_x) = \frac{1}{2}\,\dot{e}_x^{\,T} M\,\ddot{e}_x + \frac{1}{2}\,\dot{e}_x^{\,T} K_m\,e_x \tag{5-62}$$

From equation (5-58) $K_m\,e_x$ can be calculated as:

$$K_m\,e_x = J\left[M^{-1}(\theta)\,\tau - \Delta\ddot{\theta}\right] - B_m\dot{e}_x + \dot{J}\,\dot{\theta} \tag{5-63}$$

therefore, from equations (5-59) and (5-62):

$$\begin{aligned} K_m\,e_x &= J\left[M^{-1}(\theta)\left(M(\theta)\,\ddot{\theta} + C\right) - \Delta\ddot{\theta}\right] - B_m\dot{e}_x + \dot{J}\,\dot{\theta} \\ &= J\left[\ddot{\theta} + M^{-1}(\theta)\,C - \Delta\ddot{\theta}\right] - B_m\dot{e}_x + \dot{J}\,\dot{\theta} \\ &= J\,\ddot{\theta} + J\,M^{-1}(\theta)\,C - J\,\Delta\ddot{\theta} - B_m\dot{e}_x + \dot{J}\,\dot{\theta} \\ &= -\,B_m\dot{e}_x + \ddot{x} + J\left(M^{-1}(\theta)\,C - \Delta\ddot{\theta}\right) \end{aligned} \tag{5-64}$$

where: $\ddot{x} = J\,\ddot{\theta} + \dot{J}\,\dot{\theta}$, and from equation (5-52) the $\Delta\ddot{\theta}$ is:

$$\begin{aligned} \Delta\ddot{\theta} &= M^{-1}(\theta)[V(\theta,\dot{\theta}) + F_n + G(\theta) + J^T(\theta)F_{ext}] \\ &= M^{-1}(\theta)\,C \end{aligned} \tag{5-65}$$

From equations (5-63) and (5-64):

$$K_m \, e_x = - B_m \dot{e}_x + \ddot{x} + J \left(M^{-1}(\theta) \; C - M^{-1}(\theta) \; C \right)$$
$$= - B_m \dot{e}_x + \ddot{x} \qquad \textbf{(5-66)}$$

and from definition of $(\ddot{e}_x = \ddot{x}_d - \ddot{x})$, where $\ddot{x}_d = 0$:

$$K_m \, e_x = - B_m \, \dot{e}_x - \ddot{e}_x \qquad \textbf{(5-67)}$$

if equation (5-66) is put into equation (5-61), then:

$$\dot{V}(e_x) = \frac{1}{2} \, \dot{e}_x^{\,T} \, M \, \ddot{e}_x + \frac{1}{2} \, \dot{e}_x^{\,T} \, (-B_m \, \dot{e}_x - \ddot{e}_x) \qquad \textbf{(5-68)}$$

Since M is an identity matrix, therefore:

$$\dot{V}(e_x) = - \dot{e}_x^{\,T} \, B_m \, \dot{e}_x \qquad \textbf{(5-69)}$$

from equation (5-69) the third condition is satisfied. Therefore, the controller is asymptotically stable.

5.3.2 Simulation Results

A two degree of freedom planar robot corresponding to the experimental setup was used in simulations. The new impedance controller, with an observer, was first applied to a single arm robot.

<u>Single arm case:</u>

The impedance controller uses the observer to calculate dynamic terms for compensation purposes. Various conditions have been used to test this impedance controller incorporating an observer.

The first simulation was carried out to test compensation of centrifugal and Coriolis terms. Figure (5-23) shows that, without compensation the trajectory of the end-effector is no longer a straight line, as shown by (b). This error might be very important for some of the robot applications. However, with the addition of an observer, as shown in (c), a very close to a straight line trajectory is achieved. Addition of an observer creates an almost exact match to the full analytical computation case scenario shown in (a).

The second simulation was carried out to test compensation of the friction term ($F(\dot{\theta})$ in equation (4-13)). Figure (5-24) shows, again, that without compensation the trajectory of the end-effector is no longer a straight line, as shown by (b). Again, the addition of an observer, as shown in (c), causes a very close to a straight line trajectory to be achieved. Addition of an observer, again, creates an almost exact match to the case of full analytical computation, as shown in (a).

The third simulation was carried out to test compensation of the combined, friction, centrifugal, and Coriolis terms. Figure (5-25) again shows that, without compensation the trajectory of the end-effector is no longer straight line (b). Again, by adding an observer the trajectory is very close to straight line (c).

The new impedance controller with an observer was tested for modelling errors in the inertial term. For this case the inertial term in the simulation box was changed irregularly.

Example:

for the original inertia term:

$$M = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \qquad \textbf{(5-70)}$$

the following modified an inertia matrix was chosen:

$$M = \begin{bmatrix} 1.9(m_{11}) & 1.4(m_{12}) \\ \\ 2.1(m_{21}) & 1.5(m_{22}) \end{bmatrix} \tag{5-71}$$

The results, as seen in the figure (5-26), show that without an observer(a) the controller cannot create a straight line trajectory. However, with an observer, case (b), the controller could create a straight line trajectory.

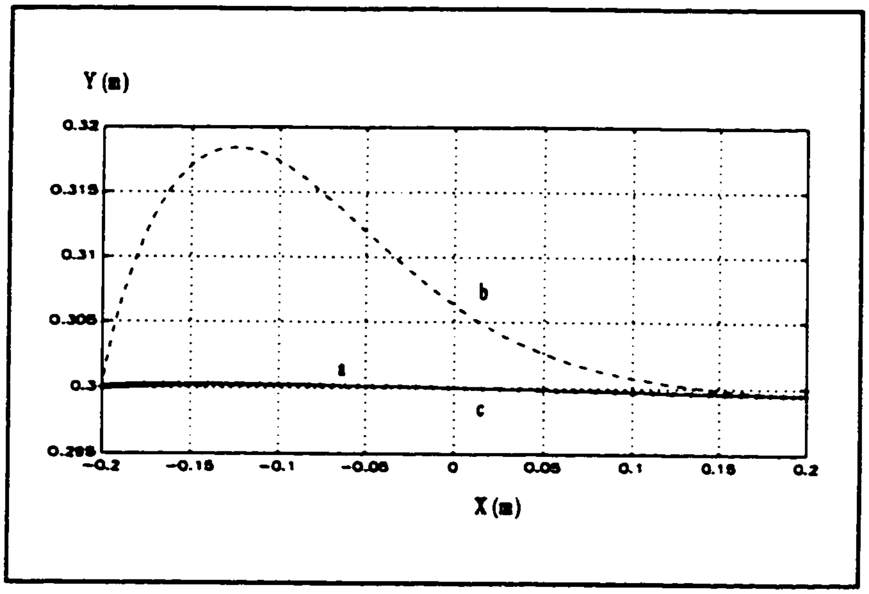

Figure 5-23 Simulation results of comp- ensating centrifugal and Coriolis terms using an observer. a)full compensation, b)without compensation, c)with an obse- rver.

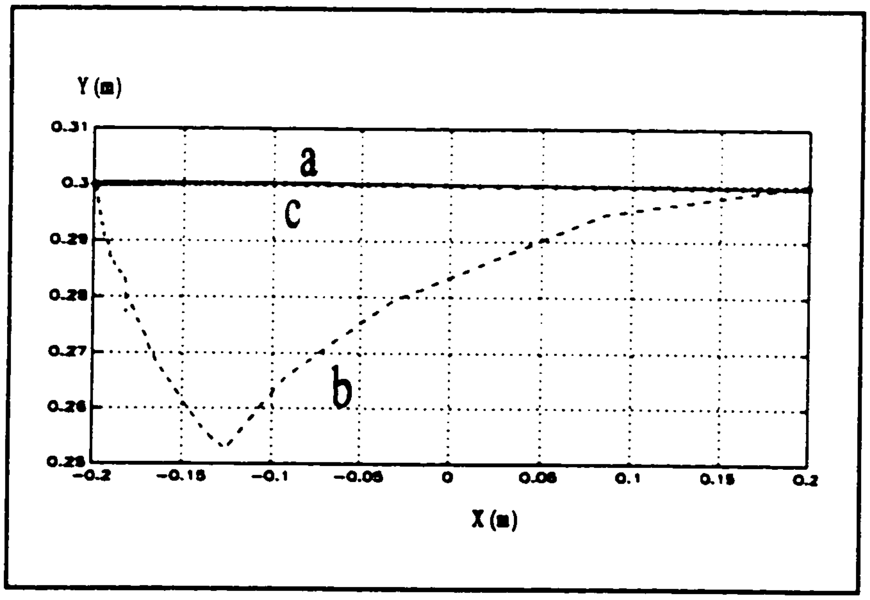

Figure 5-24 Simulation results of comp- ensating the friction term using an ob- server a)full compensation, b)without compensation, c)with an observer.

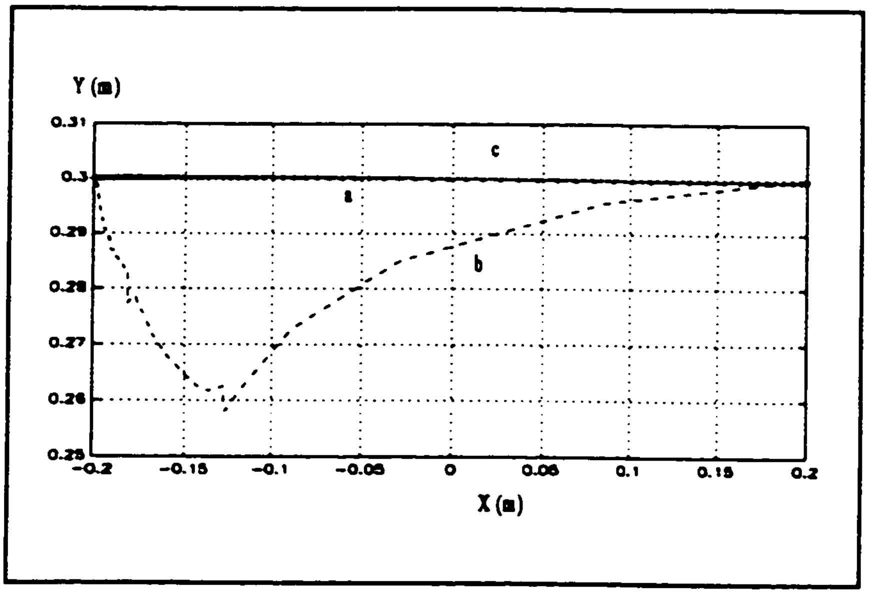

Figure 5-25 Simulation results of comp- ensating the centrifugal, Coriolis and friction terms using an observer.a)full compensation, b)without compensation, c)with an observer.

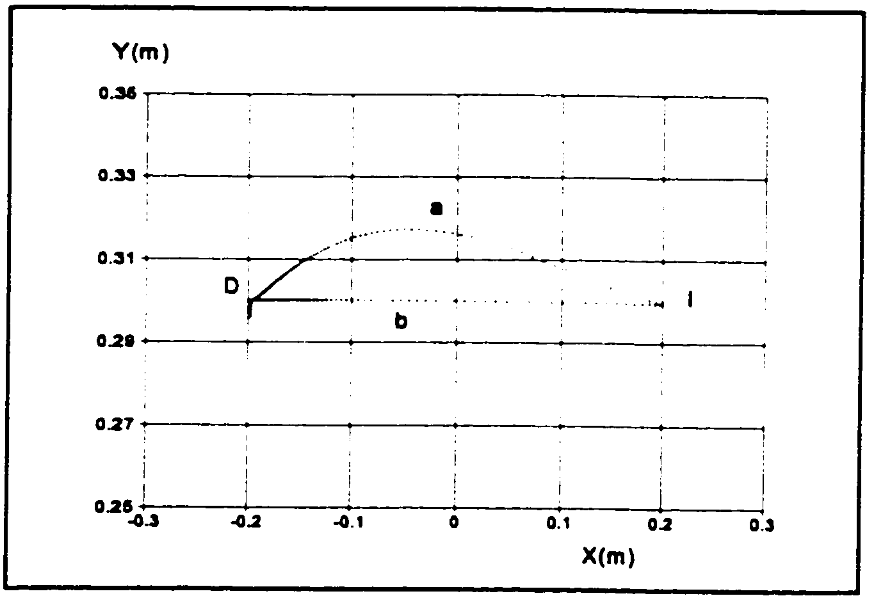

Figure 5-26 Simulation result for miss-modelling the inertial term. a)without an observer, b)with an observer.

5.3.3 Experimental Results

Experimental work was carried out on the setup described in chapter 3. A perturbation observer was tested, first, based on joint acceleration signals from the second derivative of measured angular position (figure(5-27)). It was also tested based on the joint acceleration signal measured by a combination of the joint velocity signal and the Cartesian accelerometer signals. These two methods will be presented in the first two sections for a single-arm robot. The next sections present experiment verification of the proposed methods for a dual-arm robot.

5.3.3.1 Experimental results of impedance controller with an observer (using second derivative of measured angular position of joints)

Acceleration input to the observer was calculated from the second derivative of the position.

$$\ddot{\theta} = \frac{d^2}{dt^2}(\theta_m) \tag{5-72}$$

where: θ_m = angular position of each joint of the robot arm

The results displayed in figures (5-28 and 5-29) show that the signal condition should be improve, to achieve a quasi straight line trajectory.

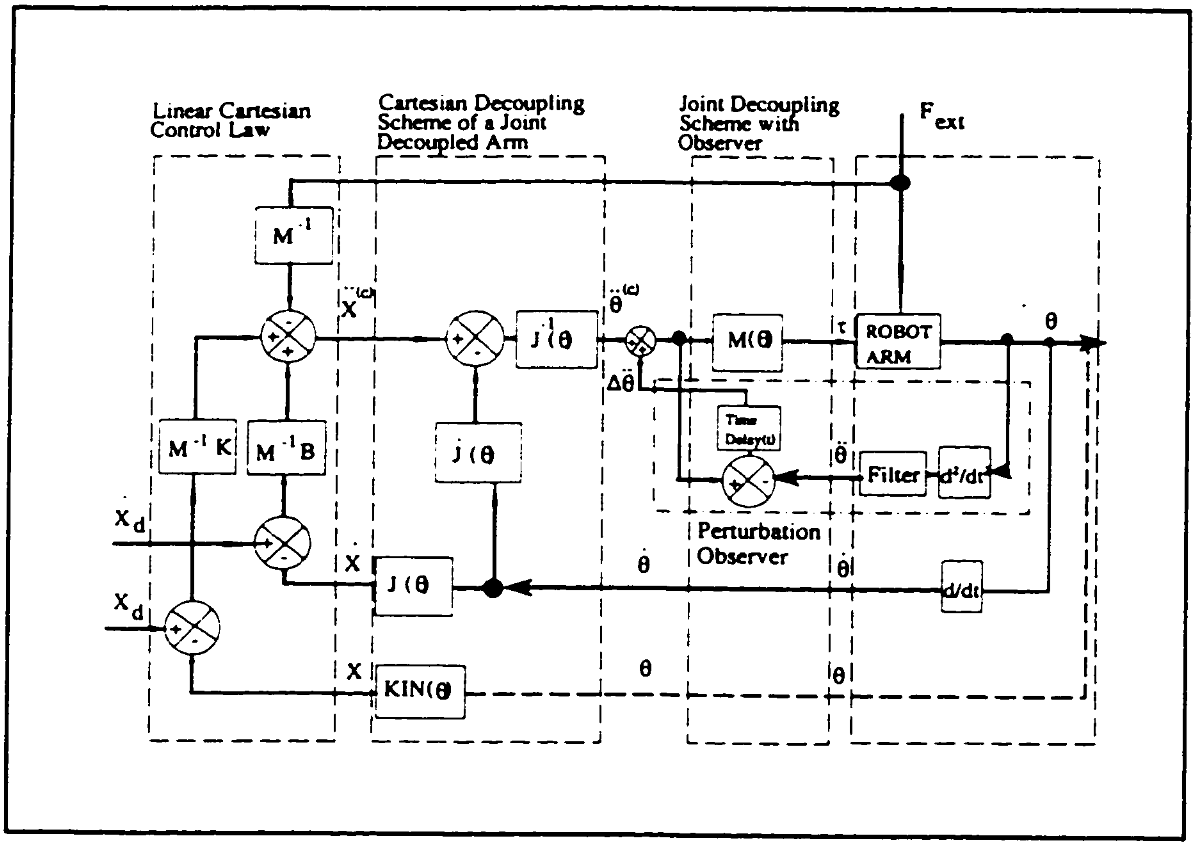

Figure 5-27 Three-part impedance controller with observer based on second derivative of joint position(equation (5-72)).

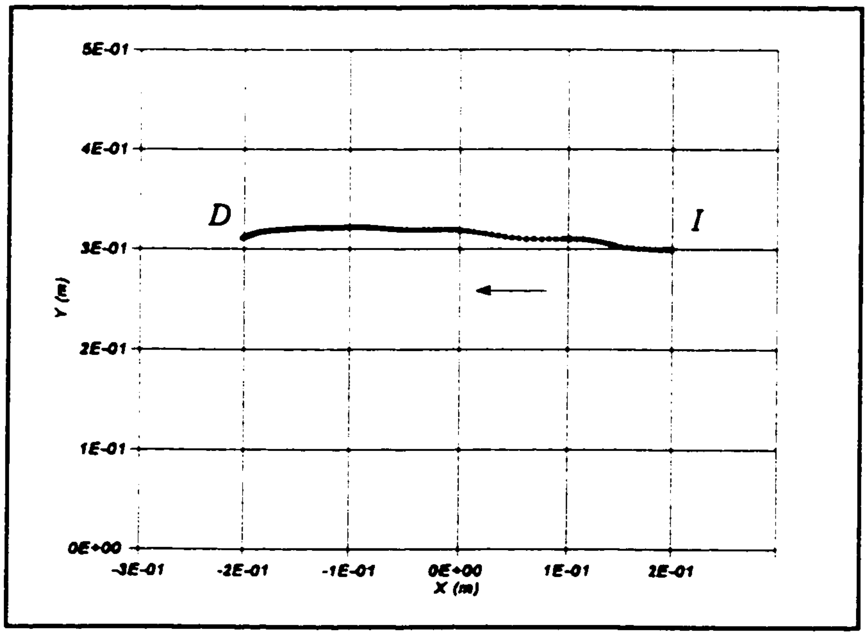

Figure 5-28 Trajectory of end-effector with compensation of unknown dynamic using an observer(experimental results).

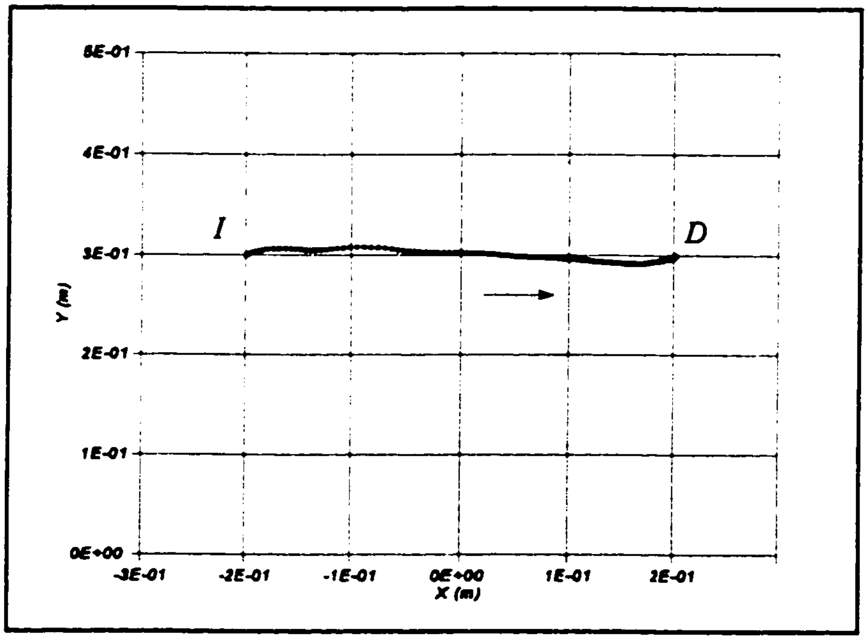

Figure 5-29 Same test as figure (5-28) with different initial and desired positions (experimental results).

5.3.3.2 Experimental results of using an observer and accelerometer signals

The impedance controller with a perturbation observer was tested again, with better acceleration feedback signals(figure (5-30)). Two accelerometers were used to measure task space acceleration of end-effector. Therefore, joint accelerations could be calculated more accurately. Joint acceleration can be calculated by using equation (5-73), which is the result of equation (5-10).

$$\ddot{\theta} = J^{-1} (\ddot{X} - \dot{J}\,\dot{\theta}) \qquad \textbf{(5-73)}$$

Where:

$\ddot{\theta}$	joint acceleration $(\ddot{\theta}_1 \ , \ \ddot{\theta}_2)$
J	Jocobian matrix
$\ddot{X}$	Cartesian acceleration signals from accelerometers $(\ddot{X}_x \ , \ \ddot{X}_y)$
$\dot{J}$	derivative of Jacobian matrix
$\dot{\theta}$	joint velocity $\left(\frac{d\theta}{dt}\right)$
θ	joint position signals from the resolver

Tests carried out, using the new controller, produced the following results:

<u>Test #1:</u>

Figure (5-31), shows the trajectory of the end-effector, without an observer, using original analytical parameters. The result is that the end-effector creates an almost straight line trajectory from the initial point to the desired position.

<u>Test #2:</u>

Figure (5-32), shows the trajectory of the end-effector, without Coriolis, centrifugal and friction terms, and without any observer. Here, the end-effector could not create a straight line trajectory that reached the desired point, due to friction mostly. The computed torque was not high enough to over come the friction forces in the motor. The break off torques of motor #1 and #2 and the chosen low proportional gain, large steady state errors resulted. They can be reduced to much lower error values, but, for the purpose of having comparative conditions in all tests, the original proportional gain was unchanged and the same value was used.

Test #3:

Figure (5-33), shows end-effector trajectory using the original parameters and a perturbation observer to compensate the Coriolis, centrifugal, and friction terms. Better results which is closer to a straight line trajectory are obtained when accelerometer signals are incorporated into the perturbation observer.

Test #4:

Figure (5-34), shows the same type of test as that described in the figure (5-33), except that the inertia parameters were the identified parameters. In both cases, the results were almost identical.

Test #5:

Figure (5-35) shows full compensation of all dynamic parameters using identified parameters and including a perturbation observer for compensation of unknown parameters. The result is that the end-effector can create a trajectory closer to straight line from the initial to desired points.

Figure (5-36) shows a case comparison of figures (5-32 to 5-35),

where:

- curve (a) represent the results from the figure (5-32)
- curve (b) represent the results from the figure (5-33)
- curve (c) represent the results from the figure (5-34)
- curve (d) represent the results from the figure (5-35)

This figure proves that better results are provided when a perturbation observer is used with full compensation of all dynamic terms using identified parameters

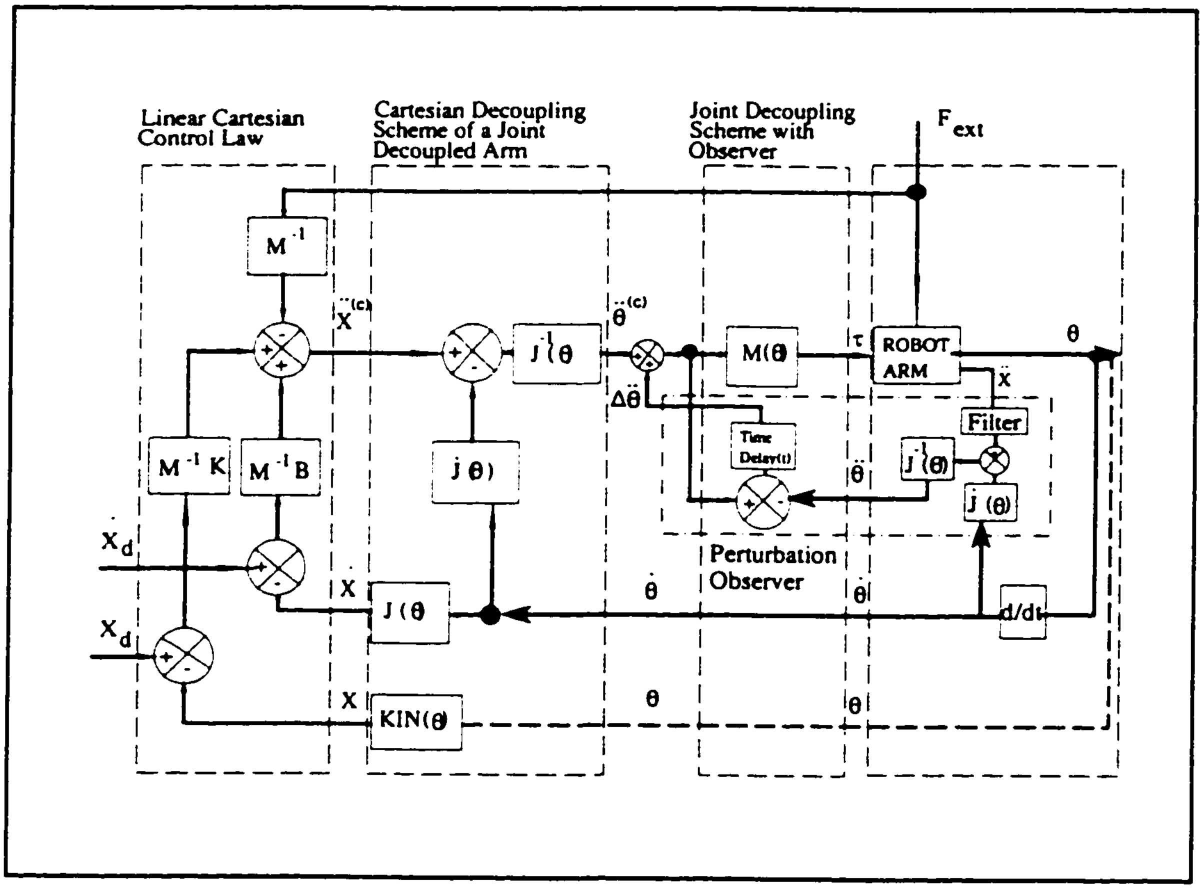

Figure 5-30 Three-part impedance controller with observer based on accelerometer signals and equation (5-73).

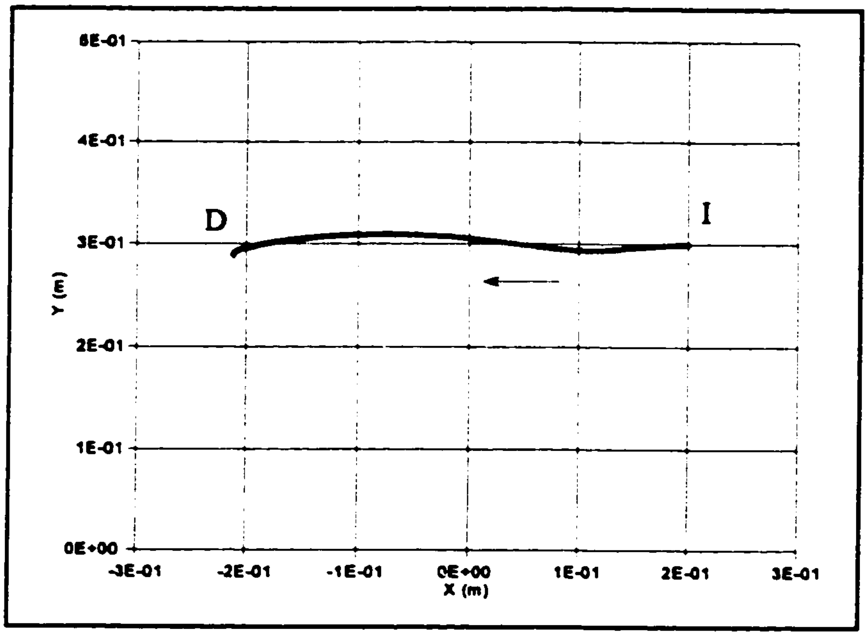

Figure 5-31 Trajectory of end-effector with compensation of analytical param- eter estimation (experimental results).

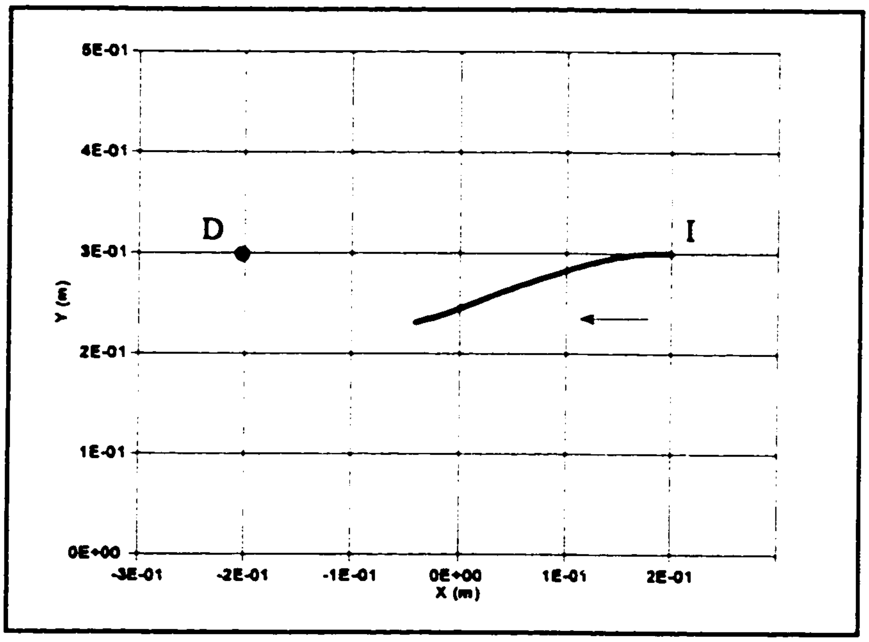

Figure 5-32 Trajectory of end-effector without compensation for Coriolis, ce- ntrifugal and friction term (experime- ntal results).

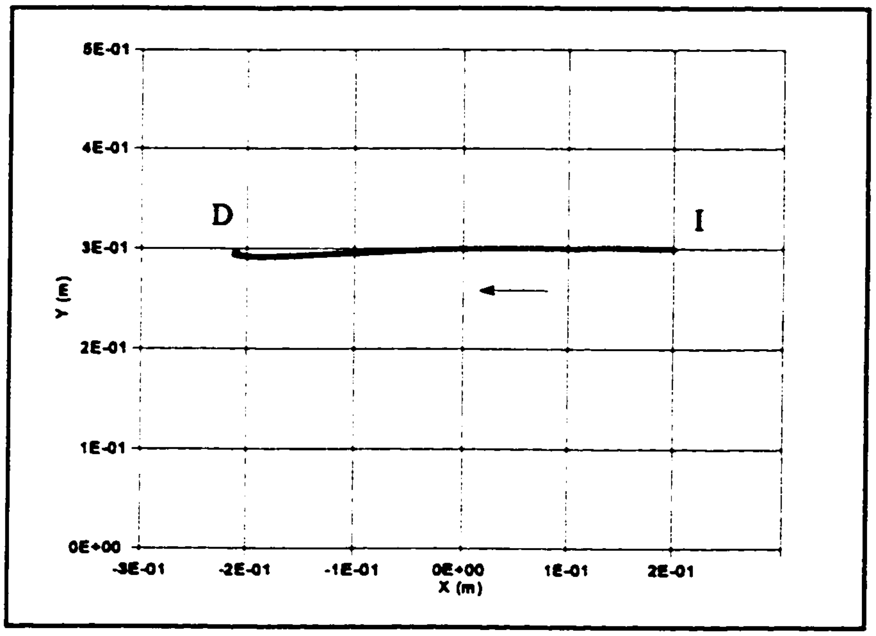

Figure 5-33 Trajectory of end-effector with using perturbation observer for compensation Coriolis, centrifugal and friction term (experimental results).

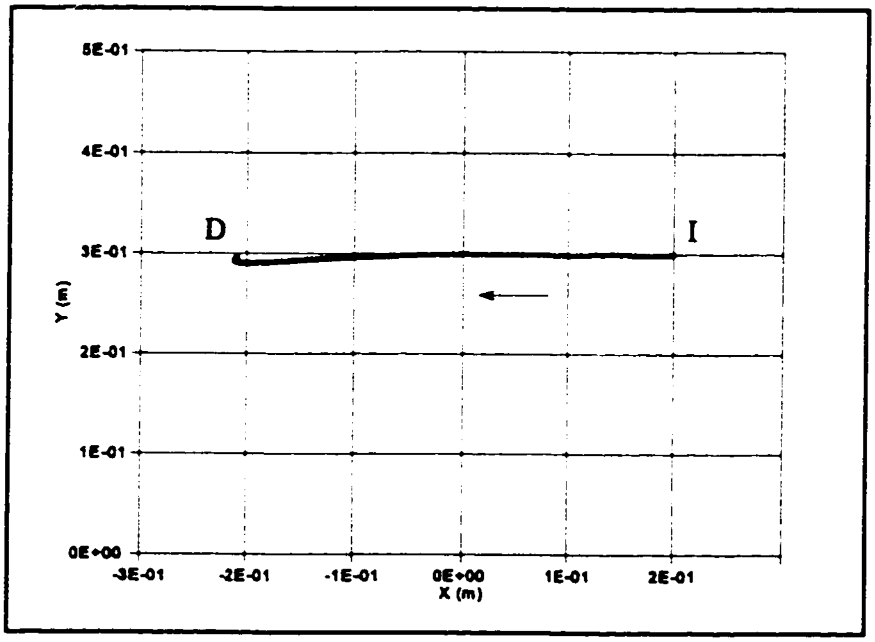

Figure 5-34 Same test as figure (5-33), except with identified parameters (exp-erimental results).

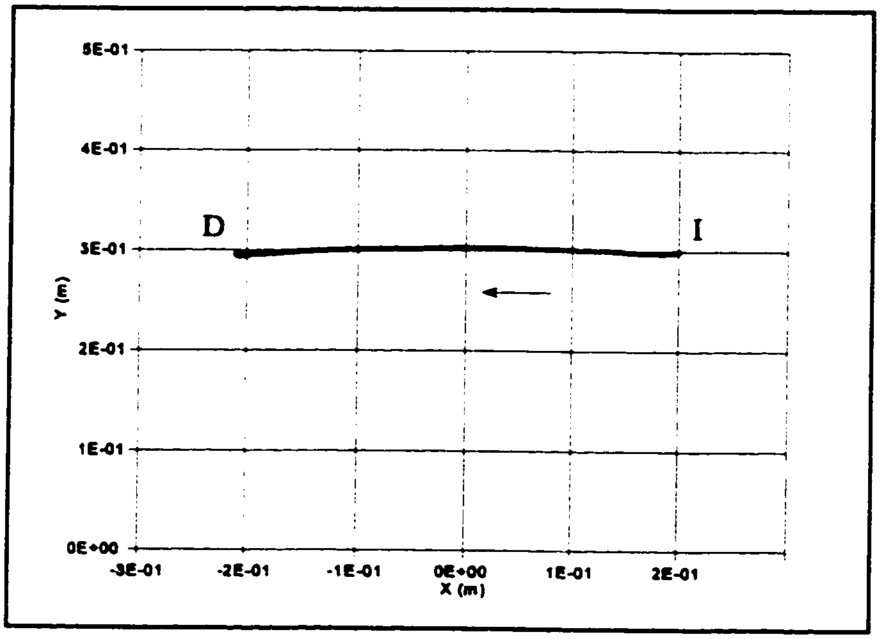

Figure 5-35 Full dynamic terms compens- ation using identified parameters and perturbation observer (experimental results).

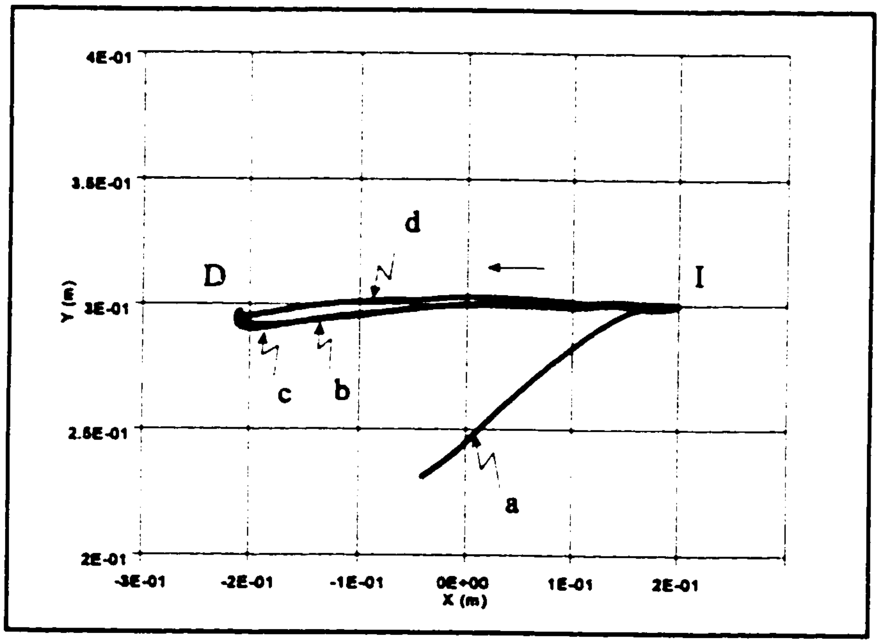

Figure 5-36 End-effector trajectories for different cases of figures (5-32 to 5-35) (experimental results).

5.3.3.3 Experimental results for a dual-arm robot

Impedance control using a perturbation observer has also been tested in the control of a dual-arm robot handling a string. However, due to the limited number of accelerometers available, only one arm was equipped with accelerometers. Therefore, in dual-arm experiments joint accelerations were calculated from the second derivative of position signals for both arms (equation (5-72)). The tests that follow were conducted on the dual-arm robot using the following notations:

I^L	Initial position of arm #1 (Left)
D^L	Desired position of arm #1 (Left)
I^R	Initial position of arm #2 (Right)
D^R	Desired position of arm #2 (Right)

For these tests the controller was designed such that the Right arm trajectory was not affected by the reaction force from artificial impedance between the end-effectors. The Left arm, however, had its trajectory modified to keep the distance between end-effectors constant. The Right arm trajectory, therefore, is usually close to straight line. Left arm however, to maintain a constant distance between the two end-effectors, displays some trajectory curvature.

Test #1:

A perturbation observer was added to the controller, to compensate unknown dynamic terms. The results in figures (5-37) and (5-38) show, that the new controller creates an almost straight line trajectory for both arms, while managed to keep the distance between the two end-effectors almost constant.

Test #2:

The perturbation observer was also tested for compensation of friction, centrifugal and Coriolis terms. Figures (5-39) and (5-40) show the end-effector trajectory of the dual-arm robot without perturbation observer or friction, centrifugal and Coriolis terms compensation. Results show that both arm could not reach the desired points or creates a straight line trajectory and this is due to friction in motors mostly. The computed torque was not high enough to over come the friction forces. Steady state errors are very high again for the same reasons that were mentioned in section 5.3.3.2 (test #2).

Figures (5-41) and (5-42) show the end-effector trajectory of the dual-arm robot using a controller equipped with a perturbation observer. The robustness of the observer to compensate friction, centrifugal, and Coriolis terms while creating almost a straight line trajectory is displayed in the results.

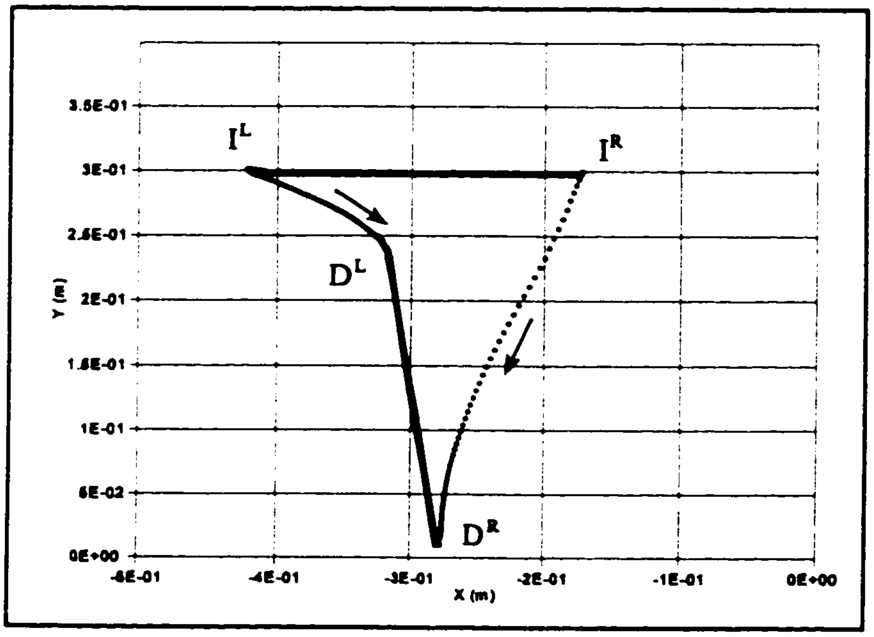

Figure 5-37 Impedance controller with a perturbation observer (experimental results).

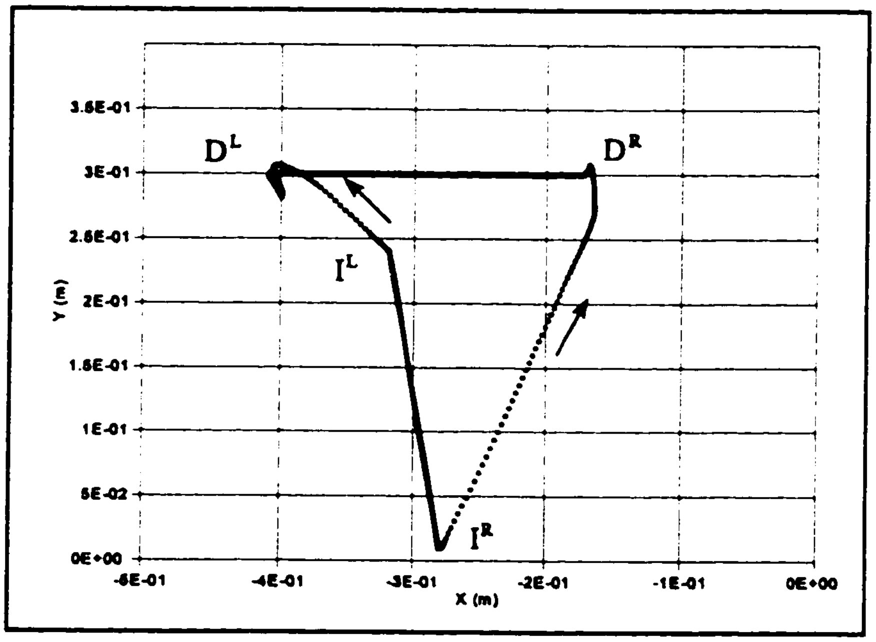

Figure 5-38 Another example, as in figure (5-37), with different initial and desired position (experimental results).

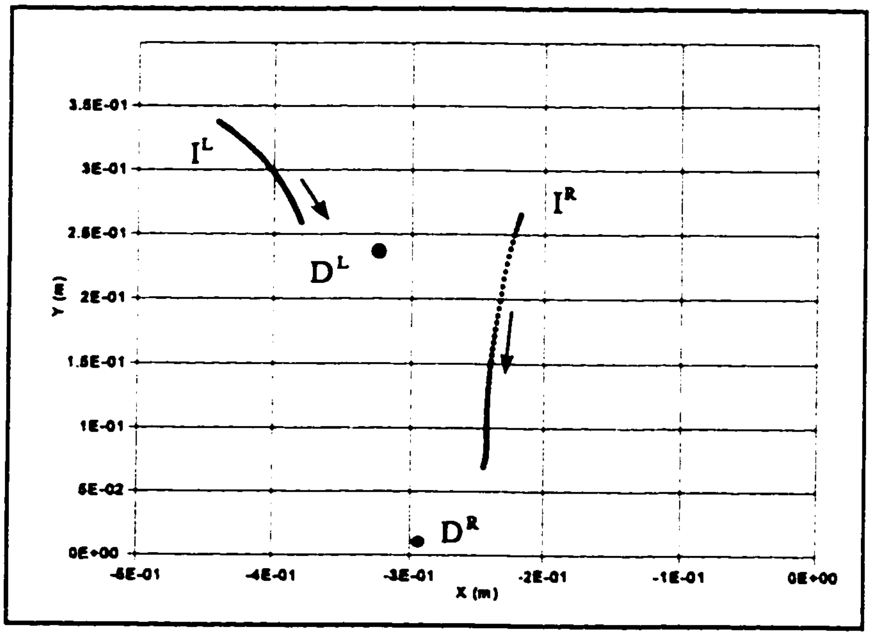

Figure 5-39 Impedance controller without observer for compensation of friction term (experimental results).

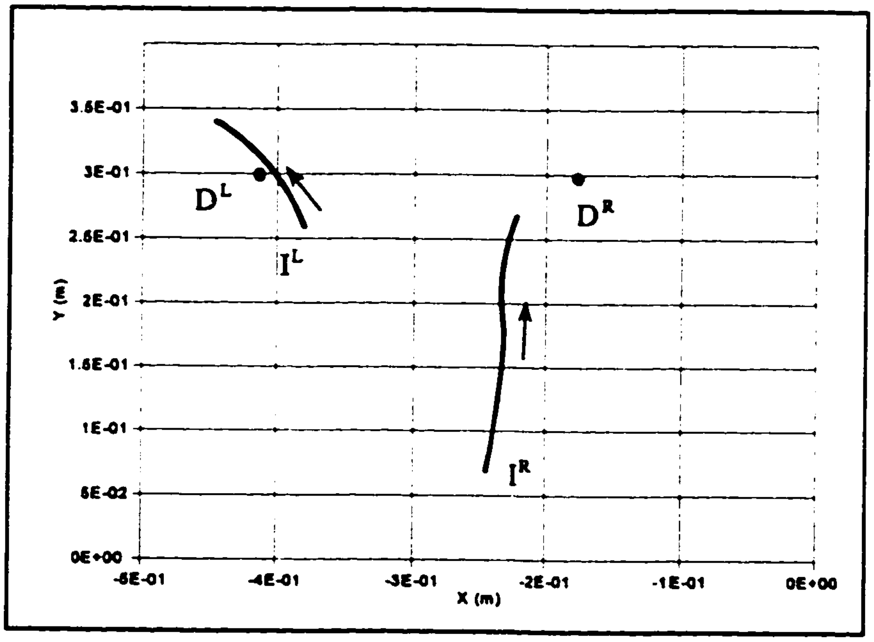

Figure 5-40 The case of compensation of friction term without using observer (experimental results).

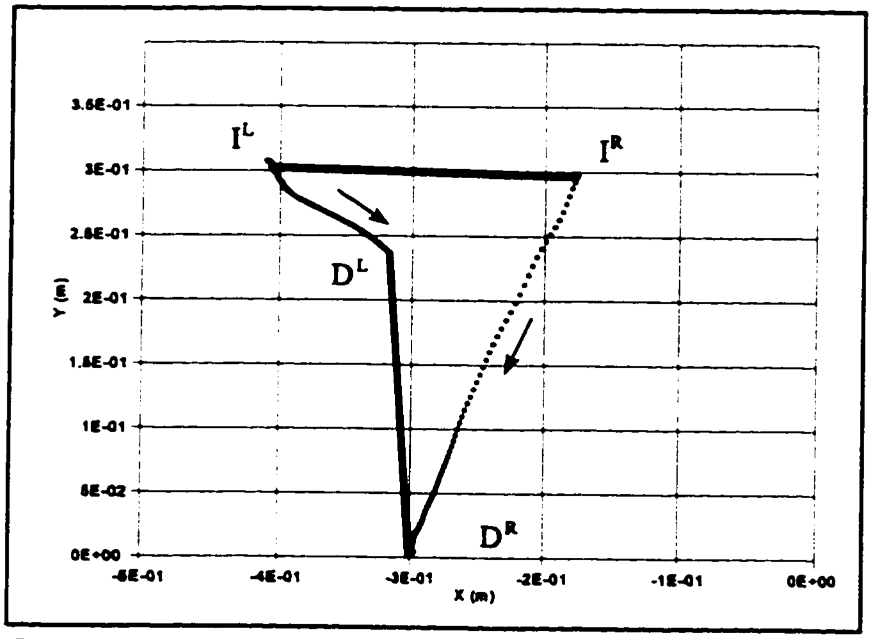

Figure 5-41 Impedance controller with an observer was used to compensate for friction, centrifugal and Coriolis terms (experimental results).

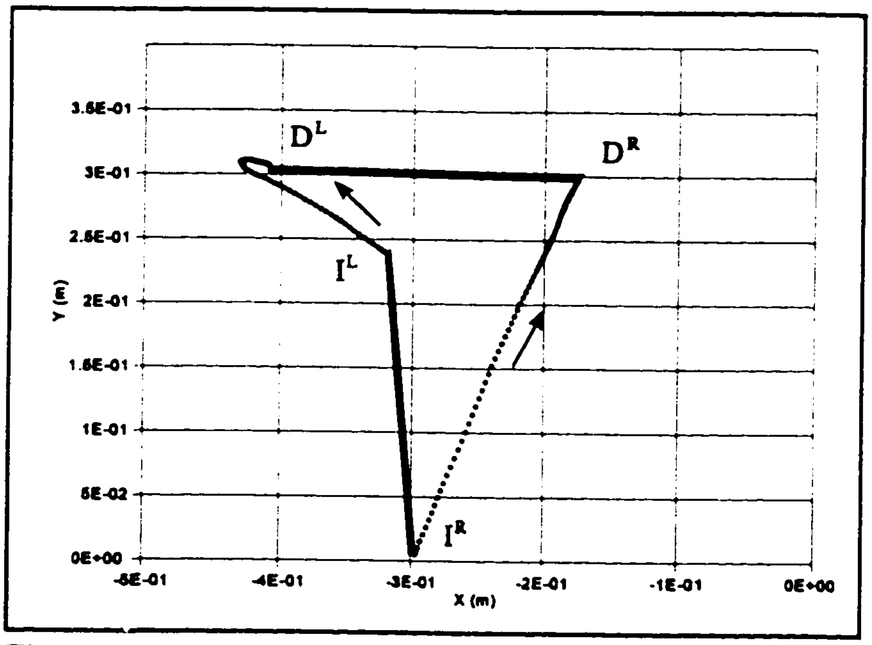

Figure 5-42 Friction, centrifugal and Coriolis terms were compensated by impedance controller with an observer (experimental results).

5.4 Perturbation Observer Using Multi-Step Predictor-Corrector Method

The observer used in section (5.3) is a one step predictor. This predictor is based on a single step time-delay or a one-step backward approach. Equations (5-50) and (5-52) show the resultant as:

$$\Delta\ddot{\theta}(t) \approx \Delta\ddot{\theta}(t-\Delta t) \tag{5-74}$$

where: Δt = sampling time

Different algorithms exist, given simulation delay, that can predict the estimated values of $\Delta\ddot{\theta}$ at time t or even t+Δt. These algorithms are based on numerical methods.

Numerical method:

Many numerical methods exist which can predict the value of $\Delta\ddot{\theta}$ at time t or $t+\Delta t$ based on previous values. The Adams-Moulton predictor-corrector method[85] can be considered as one of the best methods of multi-step prediction. The Adams-Moulton method is based on two parts, predictor and corrector and they are defined as following:

predictor:

$$y_{n+1} = y_n + \frac{h}{24}(55y_n' - 59y_{n-1}' + 37y_{n-2}' - 9y_{n-3}')$$

where: **(5-75)**

$$h = \Delta t$$

$$y_n' = \frac{y_n(t) - y_n(t-\Delta t)}{\Delta t}$$

corrector:

$$y_{n+1} = y_n + \frac{h}{24}(9y'_{n+1} + 19y'_n - 5y'_{n-1} + y'_{n-2}) \tag{5-76}$$

From these equations the predictor and corrector can be calculated, for the observer, as follows:

The predictor:

$$\Delta\ddot{\theta}_{n+1} = \Delta\ddot{\theta}_n + \frac{\Delta t}{24}\left(55\Delta\ddot{\theta}'_n - 59\Delta\ddot{\theta}'_{n-1} + 37\Delta\ddot{\theta}'_{n-2} - 9\Delta\ddot{\theta}'_{n-3}\right) \tag{5-77}$$

where:

$$\begin{aligned}
\Delta\ddot{\theta}'_n &= \frac{\Delta\ddot{\theta}_n - \Delta\ddot{\theta}_{n-1}}{\Delta t} \\
\Delta\ddot{\theta}'_{n-1} &= \frac{\Delta\ddot{\theta}_{n-1} - \Delta\ddot{\theta}_{n-2}}{\Delta t} \\
\Delta\ddot{\theta}'_{n-2} &= \frac{\Delta\ddot{\theta}_{n-2} - \Delta\ddot{\theta}_{n-3}}{\Delta t} \\
\Delta\ddot{\theta}'_{n-3} &= \frac{\Delta\ddot{\theta}_{n-3} - \Delta\ddot{\theta}_{n-4}}{\Delta t}
\end{aligned} \tag{5-78}$$

Therefore the predictor is:

$$\Delta\ddot{\theta}_{n+1} = \Delta\ddot{\theta}_n + \frac{1}{24}\left(55\Delta\ddot{\theta}_n - 114\Delta\ddot{\theta}_{n-1} + 96\Delta\ddot{\theta}_{n-2} - 46\Delta\ddot{\theta}_{n-3} + 9\Delta\ddot{\theta}_{n-4}\right) \tag{5-79}$$

The corrector is:

$$\Delta\ddot{\theta}_{n+1} = \Delta\ddot{\theta}_n + \frac{1}{24}\left(9\Delta\ddot{\theta}^{*}_{n+1} + 10\Delta\ddot{\theta}_n - 24\Delta\ddot{\theta}_{n-1} + 6\Delta\ddot{\theta}_{n-2} - \Delta\ddot{\theta}_{n-3}\right) \qquad \textbf{(5-80)}$$

where:

$\Delta\ddot{\theta}^{*}_{n+1} = \Delta\ddot{\theta}_{n+1}$ is calculated from the predictor

5.4.1 Experimental Results

The following tests were carried out to study the behaviour of a robot arm using an impedance controller equipped with both an observer and a predictor-corrector. For joint acceleration signals, the accelerometers were used as described in section (5.3.3.2).

Test #1:

This test was carried out without the predictor-corrector. The controller was equipped with an observer.

Test #2:

Impedance controller with observer and predictor-corrector were used to control the robot arm.

The results of test #1 and test #2 are shown in figures (5-43) and (5-44). These results show that end-effector trajectory, under these two types of controllers, are very similar. These tests prove that the assumption made in section (5-3) relating to $\Delta\ddot{\theta}(t) \approx \Delta\ddot{\theta}(t-\Delta t)$ was correct. When using a very small sampling time, the error is very small and can be ignored. The perturbation observer was also tested for payload mass change and lengthened time sampling.

Test #3:

The impedance controller, including the observer without prediction, was tested to control the robot with an extra mass added to the end-effector.

Test #4:

Test #3 was repeated using a predictor-corrector in the controller.

Figures (5-45) and (5-46) show the results of tests #3 and #4. Results are almost identical in both cases, because sampling time is very small and the mass involved is not very heavy. The observer, however, was robust handling small variation in mass parameters.

Test #5:

Test #3 was repeated using the full observer signal to compensate unknown dynamic terms. In previous tests, only a partial feedback signal from the observer was used for compensation.

Test #6:

Test #5 was repeated using the predictor-corrector in the controller.

Figures (5-47) and (5-48) show the results of test #5 and #6. These results are almost identical to results seen in tests #3 and #4. These tests show that 'compensation' did not require a full magnitude observer signal.

This perturbation observer using a predictor-corrector was tested using lengthened a sampling time criterion.

In previous tests the sampling time was 1 msec(1/1000Hz). In the next two tests the sampling time was increased to 2 msec(1/500 Hz).

Test #7:

This is a same test as test #1, except the 2 msec(1/500Hz) sampling time.

Test #8:

This test is the same test #7 but with predictor-corrector in the controller.

Figures (5-49) and (5-50) show trajectories of the end-effector relating to test #7 and #8. These results show that end-effector behaviour improved in the cases using the predictor-corrector option. Figures (5-51) and (5-52) show end-effector trajectories using predictor-corrector in the observer (curve a) and not using it in (curve b). Curve (b) in figures (5-51) and (5-52) is closer to a straight line than curve (b). These results demonstrate that as sampling time increases (due to computation duration for control command of systems such as multi degree of freedom robots), the predictor-corrector option for observer becomes more important.

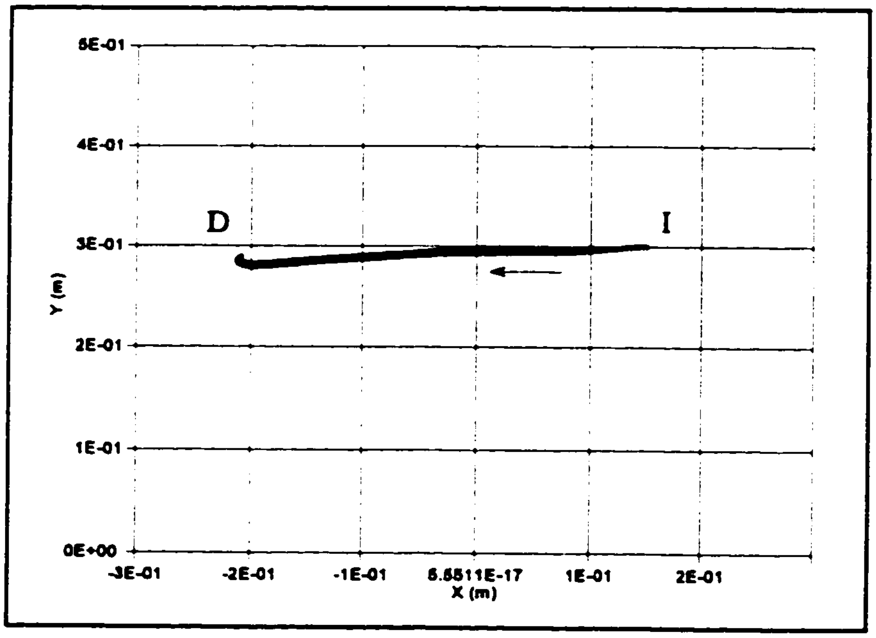

Figure 5-43 End-effector trajectories of single arm robot with and without predictor option (experimental results).

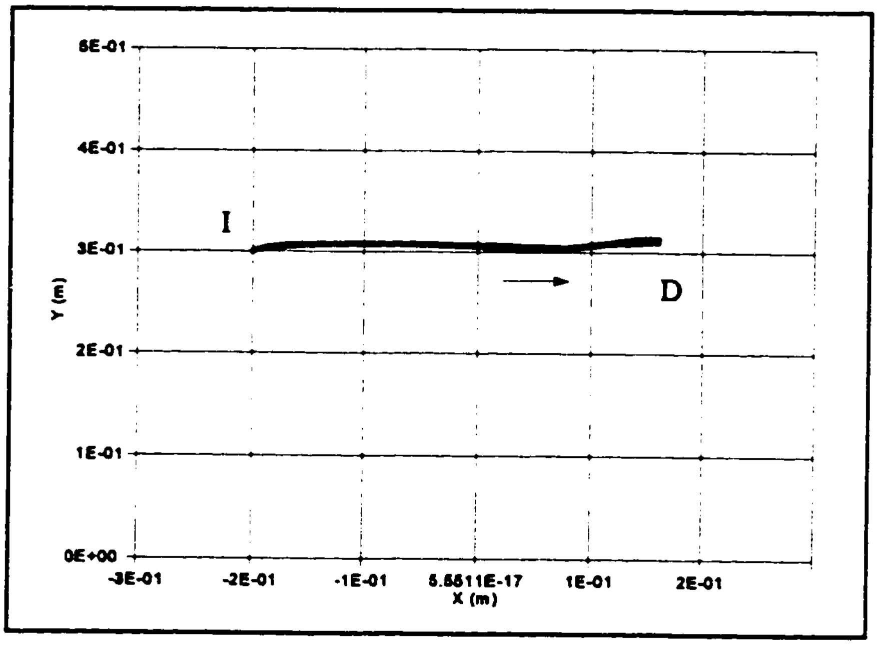

Figure 5-44 Same test as figure (5-43) with different desired trajectory (experimental results).

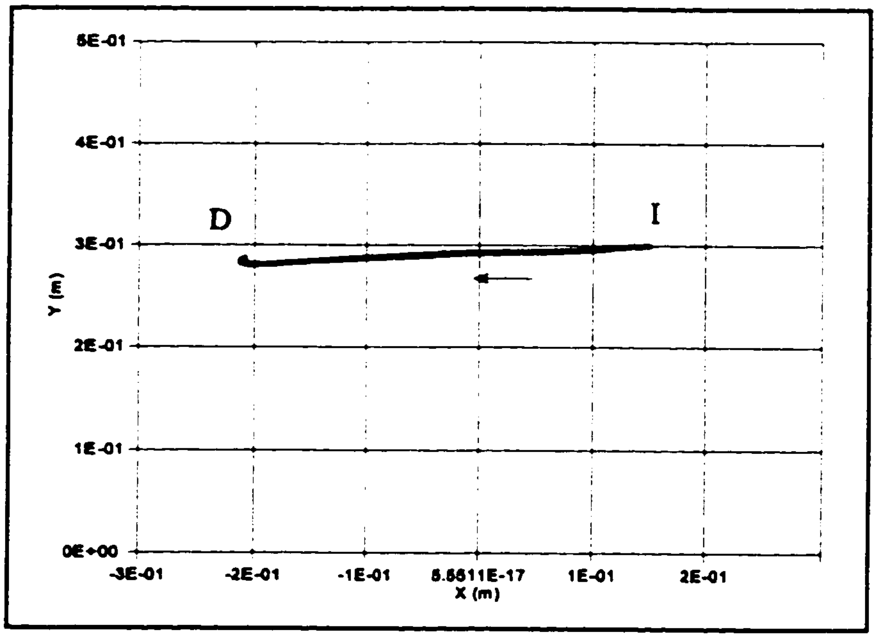

Figure 5-45 End-effector trajectory of single arm robot with and without predictor, and with extra mass at end-effector (experimental results).

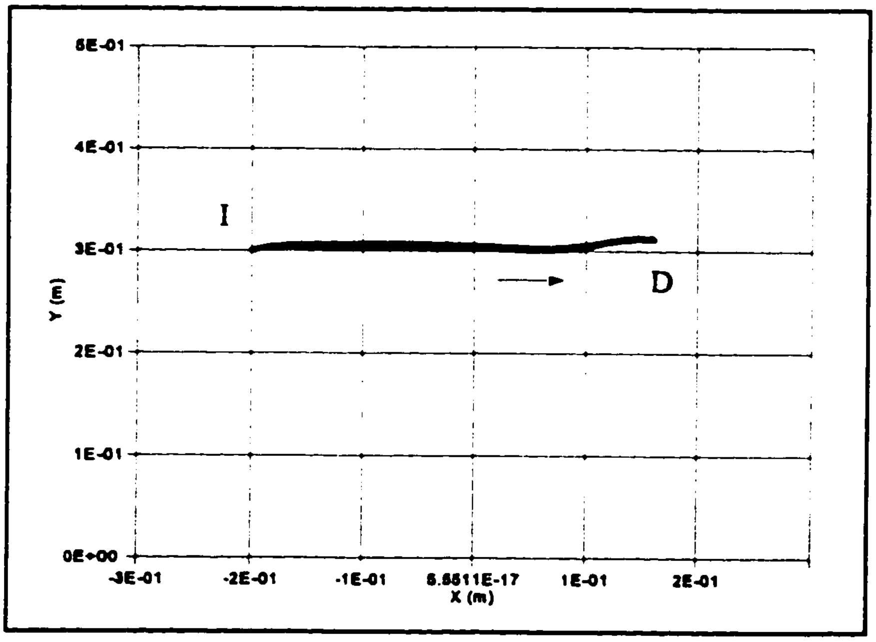

Figure 5-46 Same test as in figure (5-45) with different initial and desired positions (experimental results).

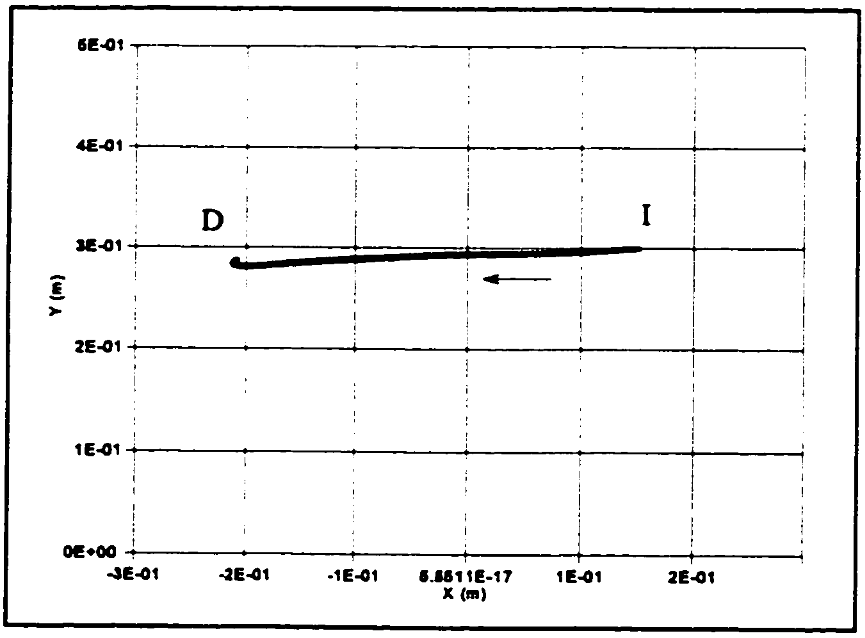

Figure 5-47 End-effector trajectory with full percentage of observer signal and with and without predictor (experimental results).

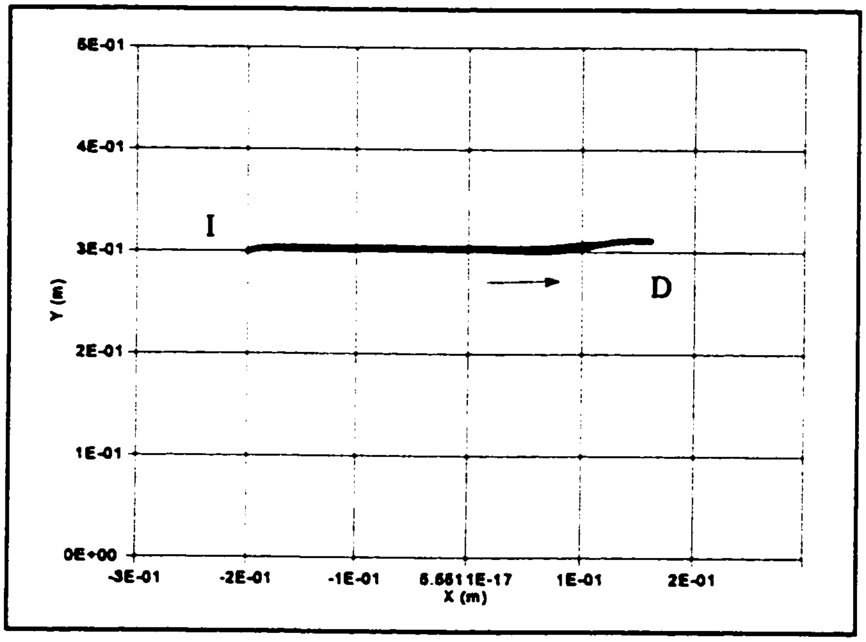

Figure 5-48 Same test as figure (5-47) for different trajectory (experimental results).

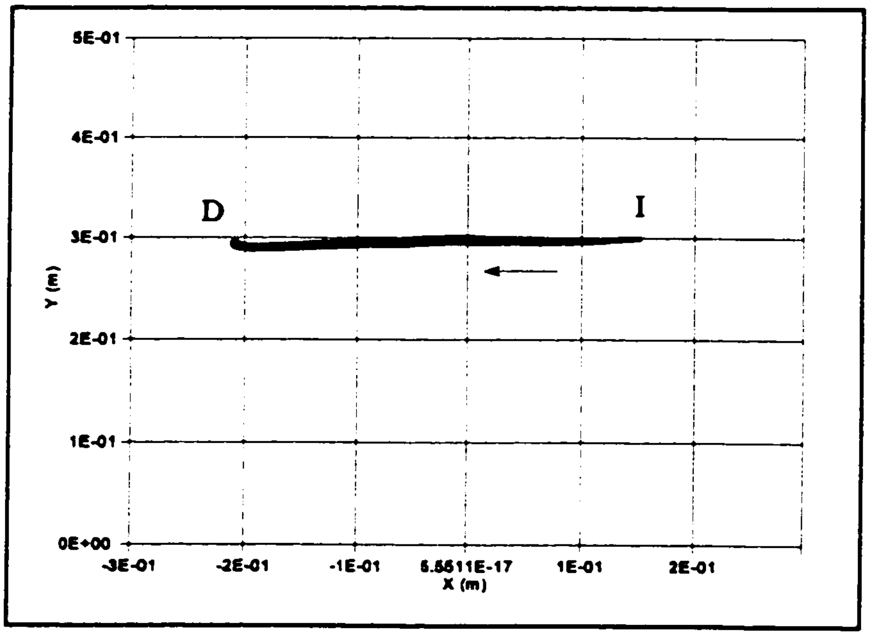

Figure 5-49 End-effector trajectories of single arm robot with sampling time of 2 msec(500 Hz) and with or without predictor (experimental results).

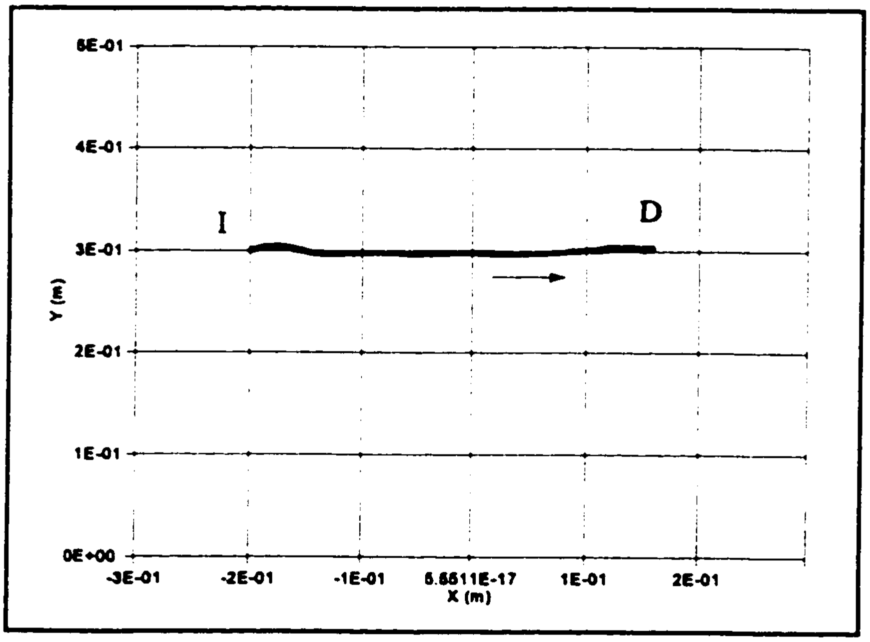

Figure 5-50 Same tests shown as in figure (5-49) for different trajectory (experimental results).

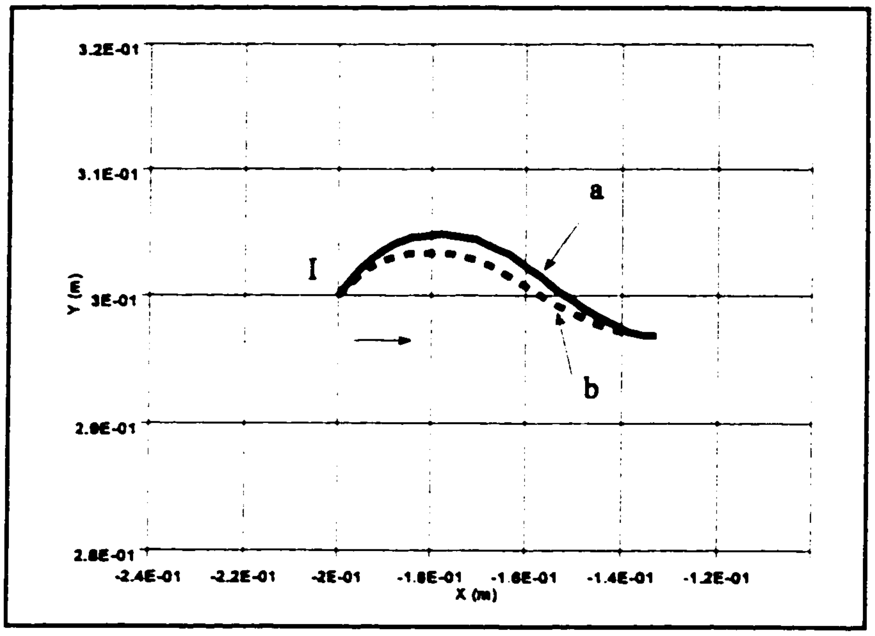

Figure 5-51 Magnified region of figure (5-50) about the initial position I, a) observer without predictor b) observer with predictor (experimental results).

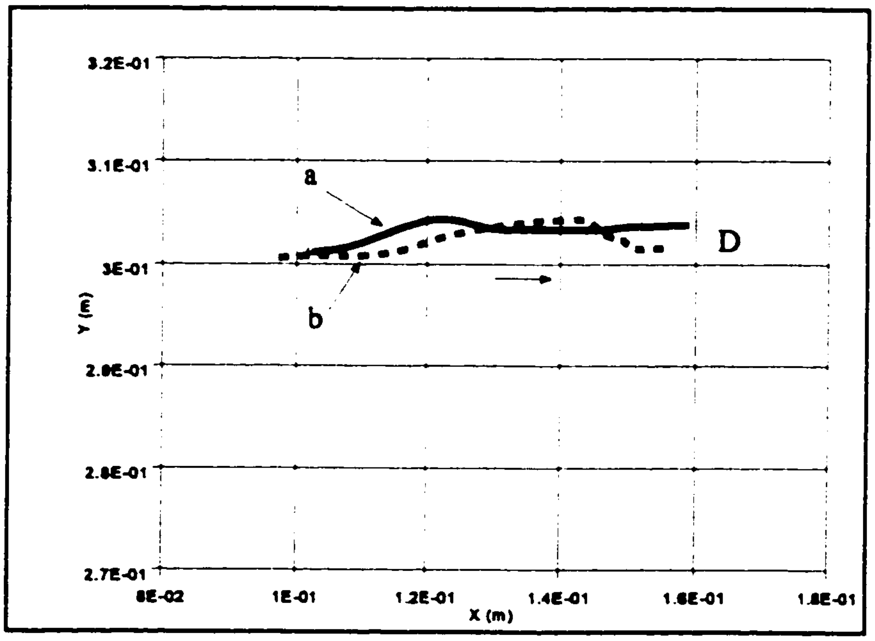

Figure 5-52 Magnified region of figure (5-50) about the desired position D, a) observer without predictor, b) observer with predictor (experimental results).

5.5 Sensor Fusion (Joint Acceleration Estimation)

Since acceleration feedback is needed for a joint observer and there are different concurrent ways to obtain it, sensor fusion can be considered for deriving better estimations based on the measurements of different sources of joint acceleration.

Robot joints resolvers used for these experiments give high resolution position(sec. 3.4.1) measurements and consequently numerical derivation for obtaining the velocity from a position signal give a useful result.

$$\ddot{\theta} = \frac{d^2}{dt^2}(\theta_m) \qquad \textbf{(5-81)}$$

Where: θ_m is angular position of the joint

Numerical derivation for obtaining the acceleration gives however a very noisy signal and a low pass filter with a low bandwidth is needed to filter the signal(figure (5-53)).

Joint acceleration can be calculated from link accelerometer signal and estimated joint speed using the kinematic equation:

$$\ddot{X} = J(\theta)\ddot{\theta} + \dot{J}(\theta)\dot{\theta} - w \qquad \textbf{(5-82)}$$

such that:

$$\ddot{\theta} = J^{-1}(\theta)[\ddot{X} - \dot{J}(\theta)\dot{\theta} + w] \qquad \textbf{(5-83)}$$

where:

$\ddot{X}$ = accelerometers output

$\ddot{\theta}$ = joint acceleration

$\dot{\theta}$ = joint velocity

w = accelerometer noise

J = Jacobian

Accelerometers installed on second link of the arm also give noisy signals and a low pass filter is again needed, but, the bandwidth of the filter can be chosen larger than the filter was used for resolver based schemes. Figure (5-54) shows the accelerometer signal without and with filtering.

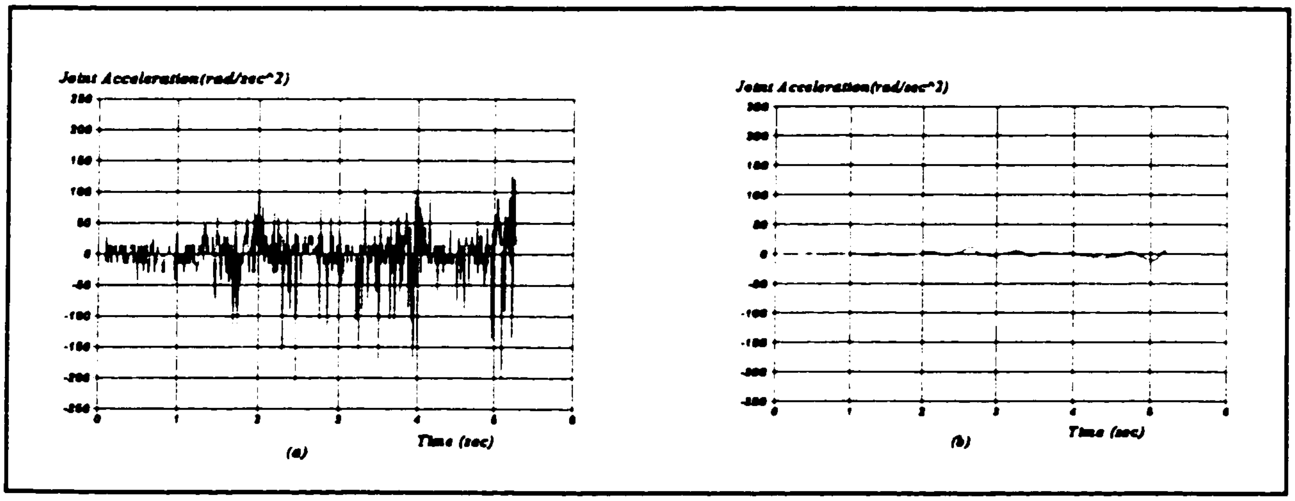

Figure 5-53 Joint acceleration calculation using second numerical derivative of the position measurement. a)non-filtered signal, b)filtered signal (experimental results).

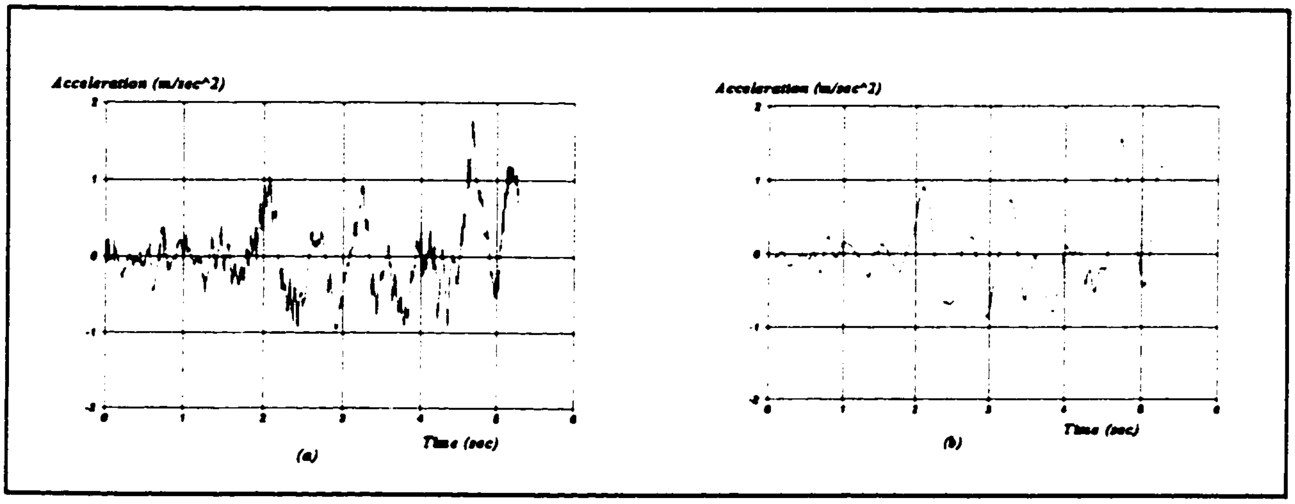

Figure 5-54 Cartesian acceleration using an accelerometer. a)non-filtered, b)filtered signal (experimental results).

The resulting estimation of the joint acceleration using equation (5-83) is shown in the figure (5-55).

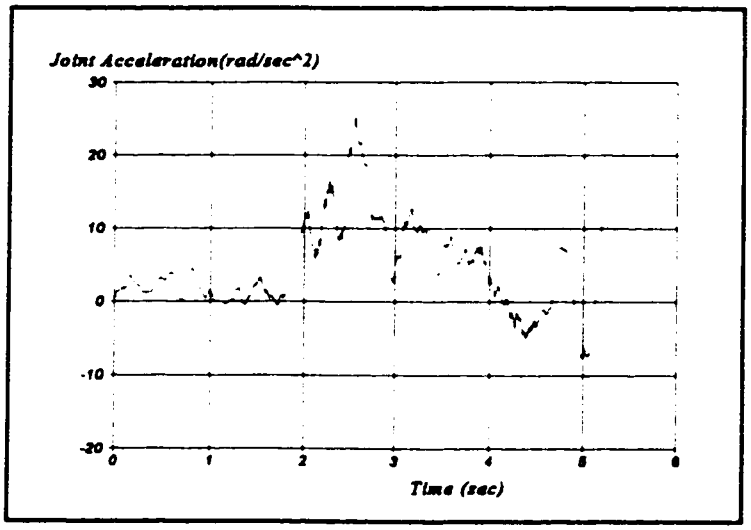

Figure 5-55 Estimated joint acceleration from accelerometer measurements (experimental results).

There are different ways to fusion these signals, as mentioned in the literature survey. Among the sensor fusion methods proposed Extended Kalman Filter(EKF) (figure(5-56)) is more suitable for real time estimation of kinematic states(i.g. Joint acceleration) given the presence of the noise and the nonlinearity of the robot arm dynamics. Kalman filter has proved itself as simple and elegant way of estimating of states of linear dynamic system. For nonlinear systems the Kalman filter was modified to extended Kalman filter that can estimate the states of nonlinear systems by a linearlized approximation about current state estimate[86]. Extended Kalman Filter was chosen as sensor fusion method in this thesis.

5.5.1 Sensor Fusion using Extended Kalman Filter(EKF) (Kinematics Based)

This algorithm is based on kinematic model, because the jerk in third state equation is computed kinematically. The system should be modelled in following format before the extended Kalman filter can be applied.

5.5.1.1 The Model

The states are defined as:

$$\begin{aligned} x_1 &= \theta & &(\textit{Joint Position}) \\ x_2 &= \dot{x}_1 = \dot{\theta} & &(\textit{Joint Velocity}) \\ x_3 &= \dot{x}_2 = \ddot{\theta} & &(\textit{Joint Acceleration}) \end{aligned} \tag{5-84}$$

States equations are:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ \dot{x}_3 &= V \end{aligned} \tag{5-85}$$

where: v is the joint angular jerk

The outputs of the system are $(y_1 = \textit{joint position}, y_2 = \textit{accelerometer}(\textit{end-effector}))$:

$$\begin{aligned} y_1 &= x_1 + w_1 \\ y_2 &= J(x_1)x_3 + \dot{J}(x_1, x_2)x_2 + w_2 \end{aligned} \tag{5-86}$$

where: (w_1, w_2) are noises in output signals

Matrix state equations are:

$$\begin{aligned} \dot{x} &= A\,x + B\,V \\ y &= C\,x + w \end{aligned} \tag{5-87}$$

where:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \dot{J}(x_1,x_2) & J(x_1) \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2(x_1,x_2) \end{bmatrix} \tag{5-88}$$

$$\begin{aligned} x &= [x_1\ x_2\ x_3]^T \\ Y &= [y_1\ y_2]^T \\ V &= [0\ 0\ v]^T \\ w &= [w_1\ w_2]^T \end{aligned} \tag{5-89}$$

The state equations of the system for short sampling time(T) are[88]:

$$\begin{aligned} x(n+1) &= a\ x(n) + b\ v(n) \\ y(n) &= c\ x(n) + w(n) \end{aligned} \tag{5-90}$$

where:

$$\bar{a} \approx I + TA = \begin{bmatrix} 1 & T & 0 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \qquad b \approx TB = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & T \end{bmatrix} \tag{5-91}$$

5.5.1.2 Linearization of nonlinear output equations

Linearization of the system has to be done first, in order to apply EKF(Extended Kalman Filter). The nonlinear equation is:

$$y_2 = J(x_1)x_3 + \dot{J}(x_1,x_2)x_2 + w_2 \tag{5-92}$$

The linearization of the nonlinear equation:

$$y(t) = h\big(x(t),u(t),w(t)\big) \tag{5-93}$$

is obtained [86,87] about: $(x(t)=\hat{x}(t), w(t)=0)$, and the following linear equation results:

$$y(t) \approx h\big(\hat{x}(t),u(t),0\big) + H(t)\big[x(t)-\hat{x}(t)\big] + G(t)w \tag{5-94}$$

where:

$$H(t) = \frac{\partial h\big(x(t),u(t),w(t)\big)}{\partial x(t)}$$
$$G(t) = \frac{\partial h\big(x(t),u(t),w(t)\big)}{\partial w(t)} \tag{5-95}$$

Therefore:

$$y_2 \approx J(\hat{x}_1)\hat{x}_3 + \dot{J}(\hat{x}_1)\hat{x}_2 + H(t)(x-\hat{x}) + G(t)w \tag{5-96}$$

where:

$$H(t) = \left\{ \frac{\partial J(x_1)}{\partial x} x_3 + J(x_1) \frac{\partial x_3}{\partial x} + \frac{\partial \dot{J}(x_1, x_2)}{\partial x} x_2 + J(x_1) \frac{\partial x_2}{\partial x} \right\}_{x = \hat{x}} (x - \hat{x})$$

$$G(t) = 1$$

(5-97)

The system output can be represented as:

$$y = Cx(n) + w + e(n) \qquad \textbf{(5-98)}$$

where:

$$C = \begin{bmatrix} 1 & 0 & 0 \\ \frac{\partial J(\hat{x}_1)}{\partial \hat{x}_1} \hat{x}_3 + \frac{\partial \dot{J}(\hat{x}_1, \hat{x}_2)}{\partial \hat{x}_1} \hat{x}_2 & \frac{\partial \dot{J}(\hat{x}_1, \hat{x}_2)}{\partial \hat{x}_2} \hat{x}_2 + \dot{J}(\hat{x}_1, \hat{x}_2) & J(\hat{x}_1) \end{bmatrix} \qquad \textbf{(5-99)}$$

$$w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \qquad \textbf{(5-100)}$$

$$e = \begin{bmatrix} 0 & 0 & 0 \\ -\frac{\partial J(\hat{x}_1)}{\partial \hat{x}_1} \hat{x}_3 - \frac{\partial \dot{J}(\hat{x}_1, \hat{x}_2)}{\partial \hat{x}_1} \hat{x}_2 & -\frac{\partial \dot{J}(\hat{x}_1, \hat{x}_2)}{\partial \hat{x}_2} \hat{x}_2 & 0 \end{bmatrix} \hat{X} \qquad \textbf{(5-101)}$$

The Extended Kalman Filter algorithm can now be applied to the system and the algorithm is presented in following section.

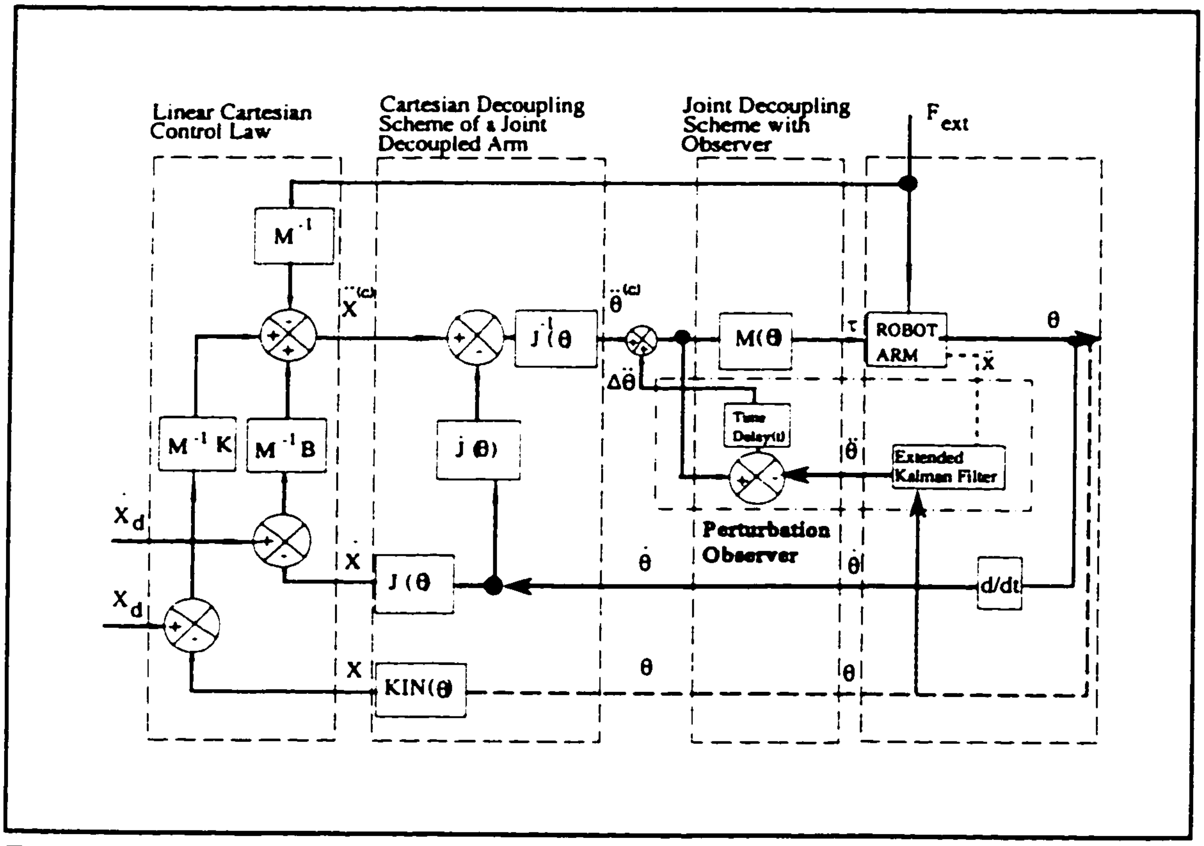

Figure 5-56 Three-part impedance controller with observer and Extended Kalman Filter.

5.5.1.3 Extended Kalman Filter(EKF) Algorithm

The Extended Kalman Filter is given by[88]:

- Kalman Gain matrix:

$$K(n) = P(n)^{-}\ C^{T}(n)\ [C(n)\ P(n)^{-}\ C^{T}(n) + w_d]^{-1} \tag{5-102}$$

- State estimate update:

$$\hat{x}(n)^{+} = \hat{x}(n)^{-} + K(n)\left(y(n) - C(n)\ x(n)^{-} - e(n)\right) \tag{5-103}$$

- State estimate extrapolation:

$$\hat{x}(n+1)^{-} = a(n)\ \hat{x}(n)^{+} + b\ v(n) \tag{5-104}$$

- Error covariance update:

$$P(n)^{+} = \left(I - K(n)\ C(n)\right)\ P(n)^{-} \tag{5-105}$$

- Error covariance extrapolation:

$$P(n+1)^{-} = a(n)\ P(n)^{+}\ a(n)^{T} + V_d \tag{5-106}$$

The noises w_d and V_d are assumed Gaussian white noises, time invariant with zero mean. In these equations, the superscript $^{-}$ refers to a time just before, while the superscript $^{+}$ refers to a time just after a sampling instant. Thus, $P(n)^{-}$ is the calculated error covariance just before the nth sample.

5.5.1.4 Example of two degrees of freedom a planar robot

The example of derivation of EKF (Kinematic based) for a two degree of freedom planar robot that is used in an experimental setup is given in the appendix (I) section (I.1).

5.5.2 Sensor Fusion using Extended Kalman Filter(EKF) (Dynamics Based)

This algorithm is based on dynamic model, because the jerks in fifth and sixth state equations are computed from dynamic model of the robot. With consideration of dynamic terms the state equations will be written as followings (for Two Degrees of Freedom Robot):

The states are defined as:

$$\begin{array}{ll} x_1=\theta_1 & \textit{Position of First Joint} \\ x_2=\theta_2 & \textit{Position of Second Joint} \\ x_3=\dot{x}_1=\dot{\theta}_1 & \textit{Angular Velocity of First Joint} \\ x_4=\dot{x}_2=\dot{\theta}_2 & \textit{Angular Velocity of Second Joint} \\ x_5=\dot{x}_3=\ddot{\theta}_1 & \textit{Angular Acceleration of First Joint} \\ x_6=\dot{x}_4=\ddot{\theta}_2 & \textit{Angular Acceleration of Second Joint} \end{array} \quad \textbf{(5-107)}$$

States equations are:

$$\begin{array}{ll} \dot{x}_1=x_3 & \\ \dot{x}_2=x_4 & \\ \dot{x}_3=x_5 & \\ \dot{x}_4=x_6 & \\ \dot{x}_5=\dddot{\theta}_1 & \textit{Jerk of First Joint} \\ \dot{x}_6=\dddot{\theta}_2 & \textit{Jerk of Second Joint} \end{array} \quad \textbf{(5-108)}$$

$\dddot{\theta}$ can be calculated by from the dynamic equation,

$$\tau = M\ddot{\theta} + N \quad \textbf{(5-109)}$$

where:

τ is torque of each joint

M inertia matrix

$\ddot{\theta}$ angular acceleration of each joint

N Coriolis, centrifugal and friction terms

$$\ddot{\theta} = M^{-1}(\tau - N)$$

$$\dddot{\theta} = \frac{d\ddot{\theta}}{dt} = \dot{M}^{-1}(\tau - N) + M^{-1}(\dot{\tau} - \dot{N}) \tag{5-110}$$

orr

$$\dddot{\theta} = \dot{M}^{-1}\tau + M^{-1}\dot{\tau} - \dot{M}^{-1}N - M^{-1}\dot{N}$$

therefore, the state equation can be written as following:

$$\dot{\boldsymbol{x}} = \boldsymbol{Ax} + \boldsymbol{Bu} + \boldsymbol{EN} + \boldsymbol{FV} \tag{5-111}$$

where:

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{5-112}$$

$$\boldsymbol{B} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & & \\ 0 & 0 & 0 & 0 & \dot{\boldsymbol{M}}^{-1} & \boldsymbol{M}^{-1} \end{bmatrix} \tag{5-113}$$

$$\boldsymbol{E} = -\boldsymbol{B}$$

$$\boldsymbol{F} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5-114}$$

$$\boldsymbol{V} = \textit{Noise of each states}$$

The state equations of the system for short sampling time(T) are [88]:

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{Ax} + \boldsymbol{Bu} + \boldsymbol{EN} + \boldsymbol{FV} \\ \boldsymbol{x}(n+1) &= (\boldsymbol{I}+T\boldsymbol{A})\, \boldsymbol{x}(n) + (T\boldsymbol{b})\, \boldsymbol{u} + (T\boldsymbol{E})\, \boldsymbol{N} + (T\boldsymbol{F})\, \boldsymbol{V} \\ \boldsymbol{x}(n+1) &= \boldsymbol{a}\, \boldsymbol{x}(n) + \boldsymbol{b}\, \boldsymbol{U} + \boldsymbol{e}\, \boldsymbol{N} + \boldsymbol{f}\, \boldsymbol{V} \end{aligned} \tag{5-115}$$

where:

$$
a = \begin{bmatrix} 1 & 0 & T & 0 & 0 & 0 \\ 0 & 1 & 0 & T & 0 & 0 \\ 0 & 0 & 1 & 0 & T & 0 \\ 0 & 0 & 0 & 1 & 0 & T \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad b = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & & & & \\ 0 & 0 & T\dot{M}^{-1} & & TM^{-1} & \end{bmatrix} \tag{5-116}
$$

$$
e = -b
$$

$$
f = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & T & 0 \\ 0 & 0 & 0 & 0 & 0 & T \end{bmatrix} \tag{5-117}
$$

Equation (5-115) should be linearlized about $(x(n) = \hat{x}(n),\ V=0)$ [86], therefore:

$$
x(n+1) = \hat{a}\hat{x} + \hat{b}u + \hat{e}\hat{N} + \hat{f}\hat{V} + \hat{P}(n)\ (x - \hat{x})
$$

where:

$$
p(n) = \frac{\partial(\hat{a}\hat{x} + \hat{b}u + \hat{e}\hat{N})}{\partial\hat{x}} \tag{5-118}
$$

$$
= a + \frac{\partial\hat{b}}{\partial\hat{x}}u + \frac{\partial e}{\partial\hat{x}}\hat{N} + \frac{\partial\hat{N}}{\partial\hat{x}}\hat{e}
$$

equation(5-118) can be written in following format:

$$x(n+1) = \hat{P}(n)\ x + \hat{b}u + \hat{e}\hat{N} + \hat{f}\hat{V} + (\hat{a} - \hat{p}(n))\ \hat{x} \tag{5-119}$$

5.5.2.1 Example for a two degree of freedom planar robot:

The example of derivation of EKF (Dynamic based) for a two degree of freedom planar robot that is used in an experimental setup is given in the appendix (J), section (J.1).

5.5.3 Experimental results

The experimental tests for extended Kalman filter were carried out in three different stages:

1) Study the acceleration signal produced by the extended Kalman filter based on kinematics and dynamics and compare the results with the acceleration signal produced by two other methods (second derivative of joint position and using accelerometers).

2) Experimentally test the behavior of the robot arm using the new impedance controller equipped with a perturbation observer. The perturbation observer used here is equipped with an extended Kalman filter that is based on kinematic models.

3) Using an extended Kalman filter based on dynamic models, repeat the experiment as that described in stage two.

5.5.3.1 Joint acceleration using Extended Kalman Filter(EKF)

Angular acceleration of both joints of the robot arm was calculated by using an Extended Kalman Filter based on kinematic and dynamic models. These models are presented in section(5.5.1 & 5.5.2). The results are compared with joint accelerations calculated from taking second derivative of joint position, as shown in equation (5-72). They were also compared with the joint acceleration signal measured by accelerometers as shown in section(5.3.3.2).

To compare these four methods, a program able to execute all these methods, for the same robot trajectories, simultaneously, was written. Since at this stage the acceleration signals were more important than the robot's trajectory, the robot trajectories were selected randomly.

A major problem of running all the methods at same time was the sampling time(or sampling frequency). The sampling time or frequency was changed from 1 msec(1000Hz) to 6 msec(167Hz) to accommodate the lengthy calculations of all the methods. The new sampling frequency(167 Hz) was much lower than the sampling frequency used in previous experimental tests, and this has to be taken into account when comparing the results.

Because of this lower sampling frequency, the analog filter designed for experimental tests, section 3.4.4, (to avoid, the aliasing problem) was no longer valid. In the earlier filter, the cutoff frequency of the filter was 118 Hz, which was larger than a half of the lower sampling frequency(167 Hz).

In order to avoid the aliasing problem the cutoff frequency of the filter had to be reduced to less than half of lower sampling frequency. The analog filter was redesigned based on 80 Hz cutoff frequency and a new resistance was selected for the filter. With assumption of 80 Hz cutoff frequency and using previous capacitor(.1 μf), the new resistance was calculated using same method as section 3.4.4, by using equation (3-6):

$$R = \frac{1}{1.69\ (50 * 80)\ .1x10^{-6}} = 1479.3\ \Omega$$

where:

$$C = .1\ \mu f$$

$$f_c = 80\ Hz$$

Therefore, as the closest resistance was selected: R = 1559 ohms.

The cutoff frequency was recalculated to make sure that cutoff frequency was still lower than half of sampling frequency(167 Hz).

$$f_c = \frac{1}{1.69\ (50)\ (1559)\ (.1x10^{-6})} = 75.9\ Hz$$

The modified analog filter was able to prevent the aliasing problem.

Besides the analog filters, the digital filters had to be changed too, because all the digital filters were derived for 1000 Hz sampling frequency. Therefore, new coefficients were calculated and replaced the previous coefficients that were shown in the table(3-2). The new coefficients of second order Butterworth filters with different cutoff frequency were calculated for 167 Hz sampling frequency and they are shown in the table (5-2).

Coefficient	5 Hz	10 Hz	15 Hz	20 Hz	25 Hz
a(2)	-1.73472577	-1.47548044	-1.22465158	-0.98240579	-0.74778918
a(3)	0.76600660	0.58691951	0.45044543	0.34766539	0.27221494
b(1)	0.00782021	0.02785977	0.05644846	0.09131490	0.13110644
b(2)	0.01564042	0.05571953	0.11289692	0.18262980	0.26221288
b(3)	0.00782021	0.02785977	0.05644846	0.09131490	0.13110644

Table 5-2 Coefficients of a second order Butterworth filter(167 Hz sampling frequency)

For the new values the experimental setup was used for tests to compare the four methods of calculating or measuring the joint acceleration of the robot arm.

At this stage the EKF was tested with different coefficients for state and output noises(w,v) and other parameters. The approximate values of coefficients were found after a few tests. These coefficients are as following:

The noise on the output signal(white Gaussian noise):

$$w = \begin{bmatrix} 1x10^{-12} & 0 & 0 & 0 \\ 0 & 1x10^{-12} & 0 & 0 \\ 0 & 0 & 0.8 & 0 \\ 0 & 0 & 0 & 0.8 \end{bmatrix}$$

The noise on each states:

$$v = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 5x10^{-5} \\ 5x10^{-5} \end{bmatrix}$$

The results of EKF study for joint one and two of the robot arm shown in figures (5-57) to (5-60) and (J-1) to (J-8). Three tests were taken and the figure showing the joint acceleration signals which were produced by four different methods. The four methods used to calculate the joint acceleration signals are as following:

Method #1: Second derivative of joint positions(eq. 5-72)
Method #2: Using end-effector accelerometers(eq. 5-73)
Method #3: Using EKF base on kinematic models (sec. 5.5.1)
Method #4: Using EKF based on dynamic models(sec. 5.5.2)

For most of the tests, the joint acceleration signals produced by these four methods were very close. Parts of the results had been zoomed and shown on separate graphs, in order to better comparison and observation between these signals.

For these tests a second order Butterworth digital filter was used to filter out the noise in signals for methods 1 and 2. However, the EKF methods(3&4) only uses raw data from joint position and accelerometers. The legends on following graphs are, as example:

- ddq1-res	acceleration of joint one, using method #1
- ddq1-mes	acceleration of joint one, using method #2
- ddq1-kal-kin	acceleration of joint one, using method #3
- ddq1-kal-dyn	acceleration of joint one, using method #4

Test #1:

The digital filter used for method 1 and 2 had 20 Hz cutoff frequency. The results of four methods are shown in figures(5-57) and (5-59). Figures (5-58) and (5-60) are zoomed parts of figures(5-57) and (5-59). As shown in these figures the four methods produce almost the same signal. Therefore, the extended Kalman filter was able to generate the acceleration signal close to other two methods. The extended Kalman filter was further tested for different selected white noises, to verify the behaviour of extended Kalman filter. Therefore, the following tests were taken and the experimental results are provided in appendix J.

Test #2:

Repeat same test as test #1, except with lower cutoff frequency(10 Hz) for digital filter. This change only effects the first two methods(1&2), but does not have any effect on EKF methods. Figures (J-1) to (J-4) shows the results of this test. In this tests the acceleration signals produced by method (1&2) have fewer oscillations.

Test #3:

In this test coefficients of output noise(white noise) is changed to following:

$$w = \begin{bmatrix} 1x10^{-12} & 0 & 0 & 0 \\ 0 & 1x10^{-12} & 0 & 0 \\ 0 & 0 & 8 & 0 \\ 0 & 0 & 0 & 8 \end{bmatrix}$$

Increasing the w coefficients will reduce the bandwidth of the EKF filter . Figure (J-5) to (J-8) shows fewer oscillations produced by methods(3&4).

These tests, do not show obvious advantage of one method over the other. Extended Kalman Filter based on kinematics or dynamics proved however to be a reliable tool for fusing both signals from resolvers and accelerometers. A feature of EKF is that the bandwidth can be controlled by adjusting the white-noise signal parameters. Based on results shown in figures(5-57) to (5-60) and (J-1) to (J-8) EKF was considered for next stage of experiment in which EKF algorithm is integrated with the perturbation observer used in the tests presented in the previous chapter.

Another topic which can be investigated is the prediction of the acceleration signal based on EKF algorithm. Because the sampling time(6 msec) is much larger than the sampling time (1 msec)used section (5-4), prediction of the acceleration signal is considered to play more an important role as sampling time increases.

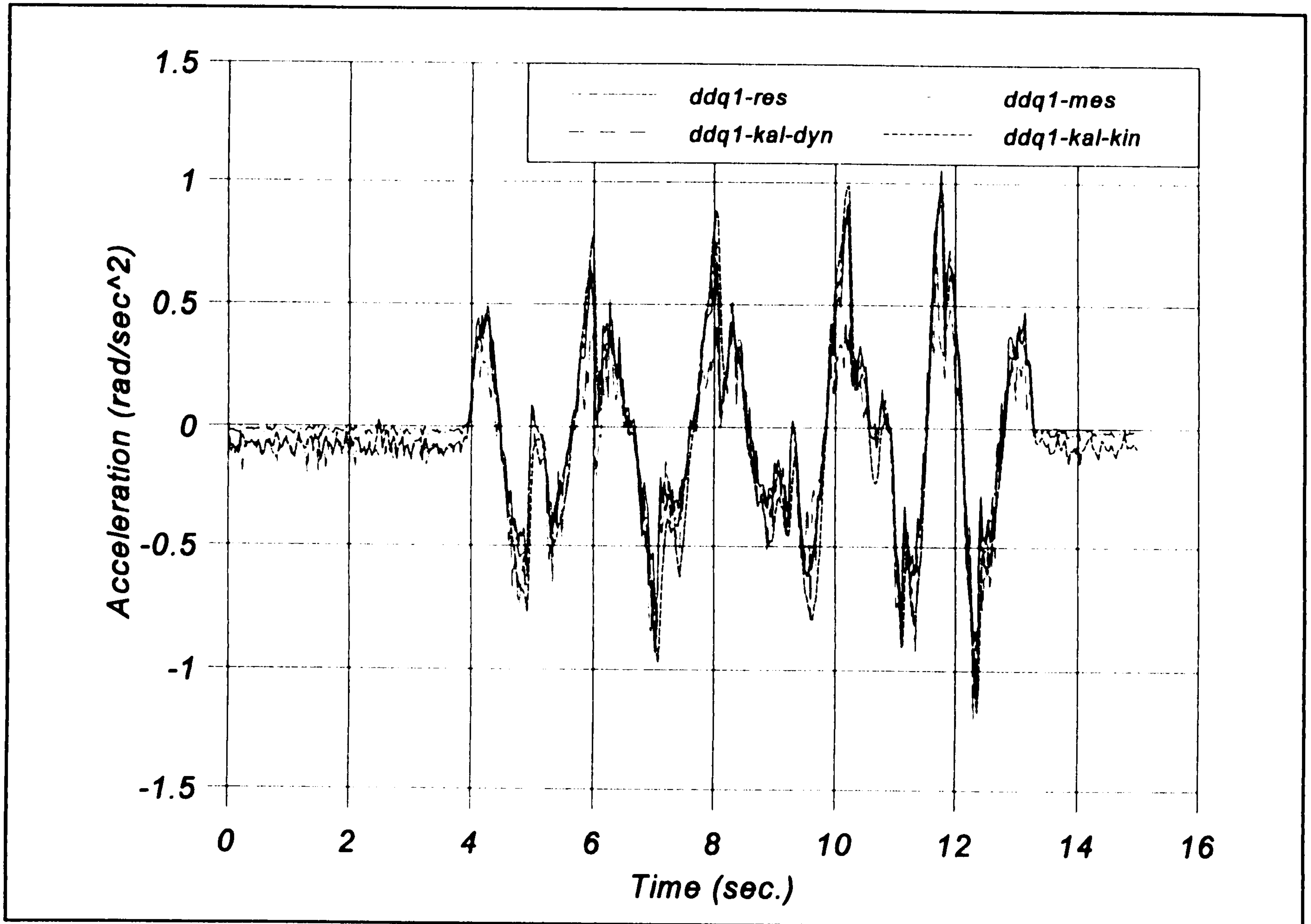

Figure 5-57 Joint one acceleration based on four methods, test #1 (experimental results)

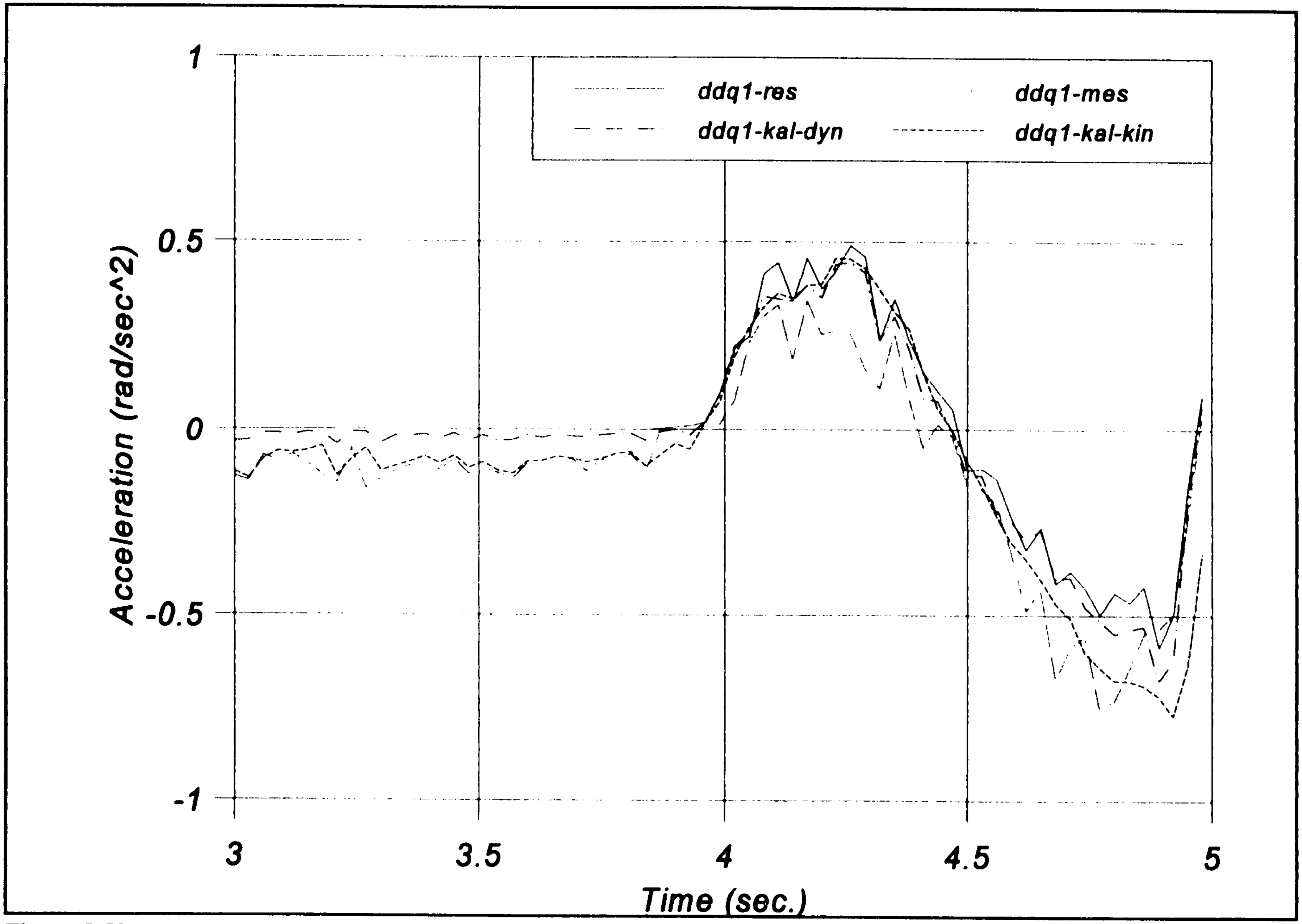

Figure 5-58 Joint one acceleration, zoom area of the figure (5-54), test #1 (experimental results)

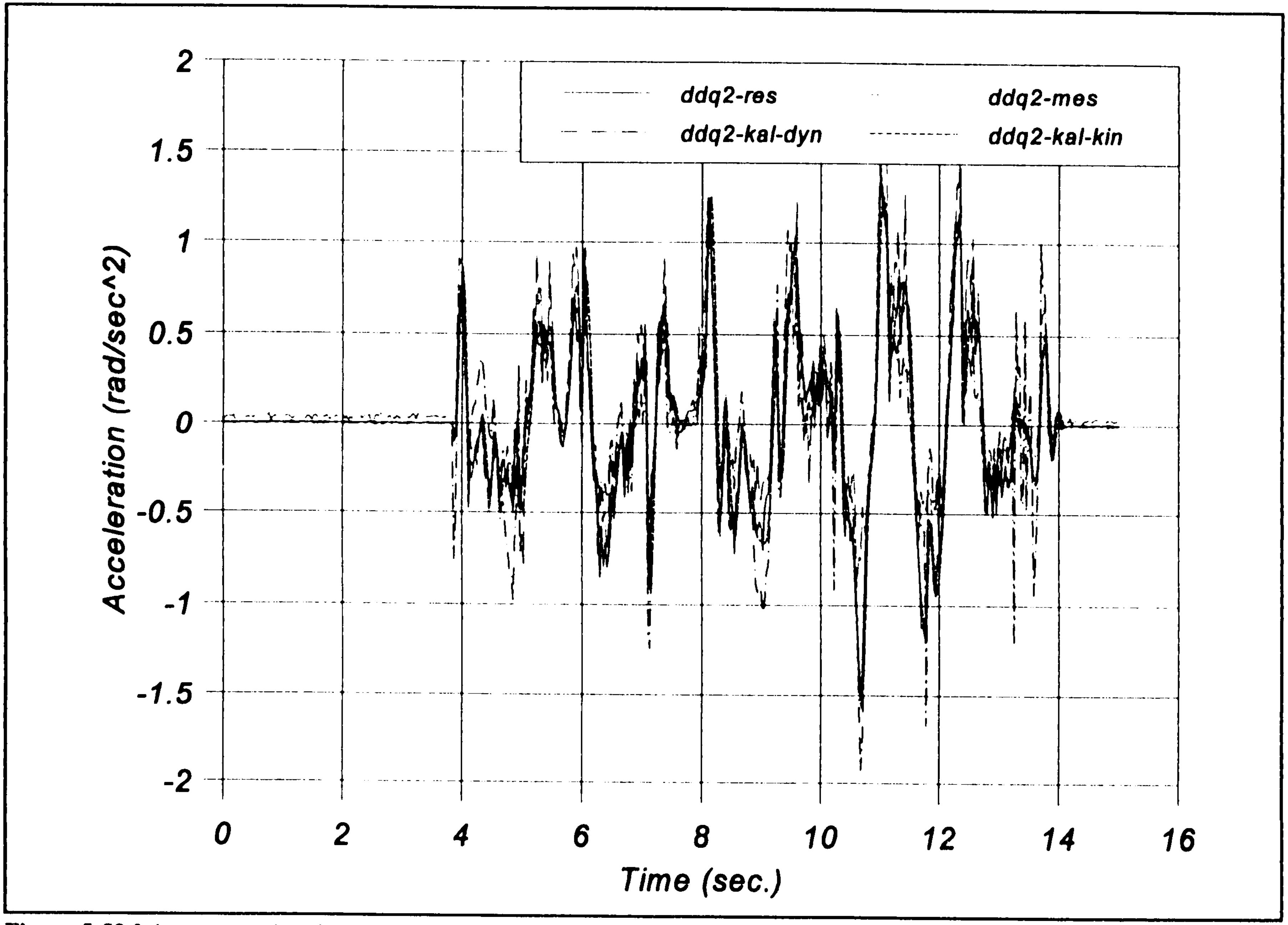

Figure 5-59 Joint two acceleration based on four methods, test #1 (experimental results)

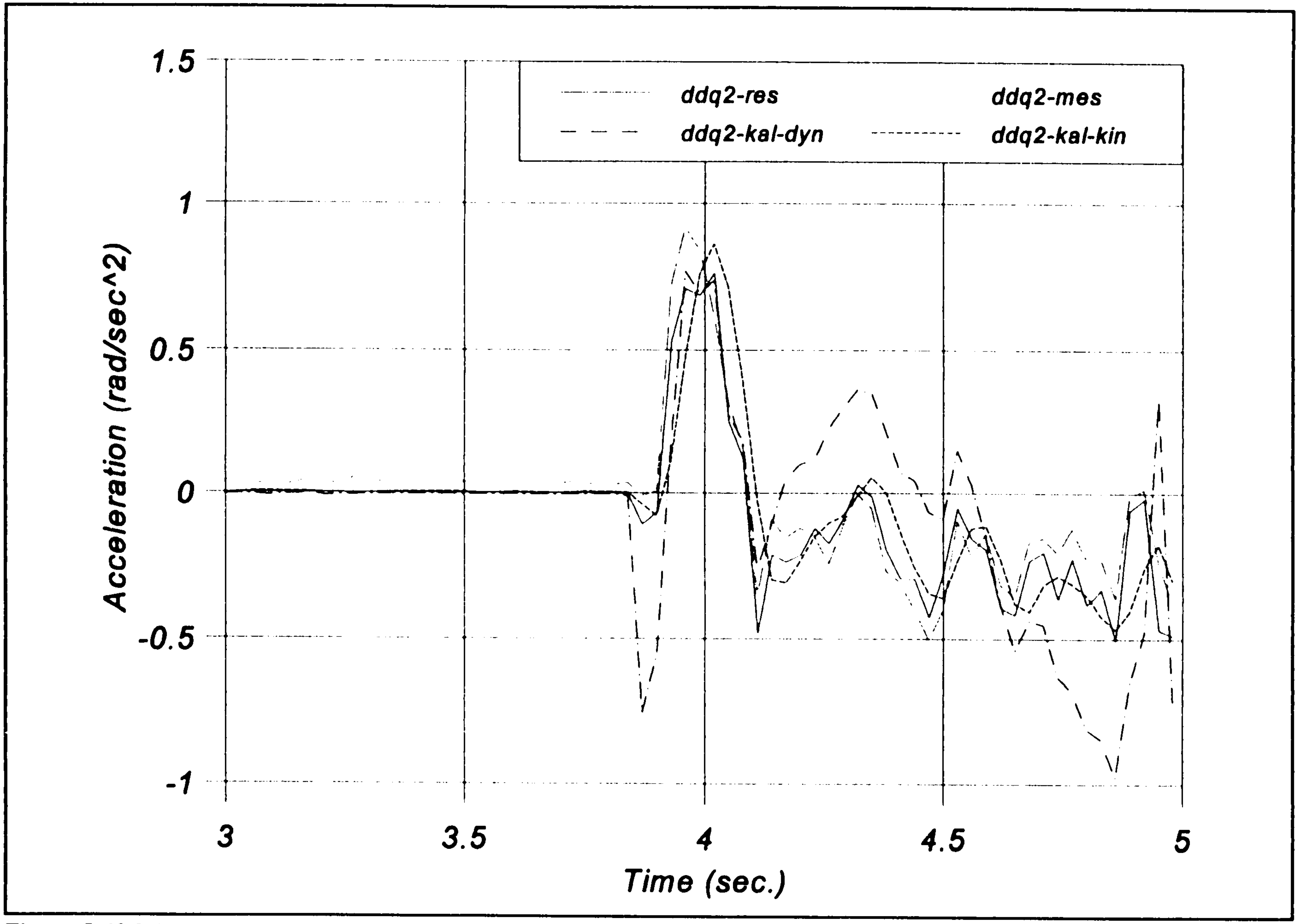

Figure 5-60 Joint two acceleration, zoom area of the figure (5-56), test #1 (experimental results)

5.5.3.2 The impedance controller, using combined approaches: a perturbation observer and an Extended Kalman Filter, based on the kinematic model (EKF-K).

The results, obtained section 5.5.3.1, show that an extended Kalman filter can provide fusion of joint acceleration, derived from resolver and accelerometer signals. This extended Kalman filter was integrated into the perturbation observer using the impedance controller described in section 5.3. The extended Kalman filter was based on the kinematic model presented in section 5.5.1. The modified impedance controller using this Kalman filter was tested for the control of a robot arm.

As described in previous sections, an impedance controller, with an observer, is a robust controller, able to compensate unmodelled dynamic terms. Observer bandwidth and controller performance are increased with the use of an extended Kalman filter. In the following tests, a robot arm is shown moving from an initial position (I) to a desired position (D). Test results, as follow, will show the behavior of a controller using this extended Kalman filter. The ideal trajectory for following tests is a straight line trajectory and for comparison the trajectory which has closer to a straight line with minimum sideway errors consider as better or best result.

Test #1:

The controller compensates for centrifugal, Coriolis, and friction dynamic terms (without using approximate models of dynamic terms). The results show the robustness of this controller, as shown in the figure (5-61). This figure shows the end-effector trajectory, from point I to point D. In this test, the controller was able to direct the end-effector to the target, with a close to straight line trajectory.

Test #2:

In test #2, centrifugal, Coriolis, and friction terms are compensated using approximate models. The observer, in this test, therefore, only compensates differences between approximate and actual dynamic term models. Results of this test can be seen in the figure (5-62). These results show trajectory improvement-the trajectory is closer to a straight line- with respect to test #1. Comparative results of test #1 and test #2 are shown in the figure (5-63).

Tests #3 & #4:

These tests are the same as tests #1 and #2, however, different initial and desired positions are used. The results of test #3 are shown in the figure (5-64) and the results of test #4 are shown in the figure (5-65). The results of both tests are shown in the figure(5-66). The results obtained, again, proved that an impedance controller using dynamic terms compensation and an observer will provide better results, since the trajectory is closer to a straight line.

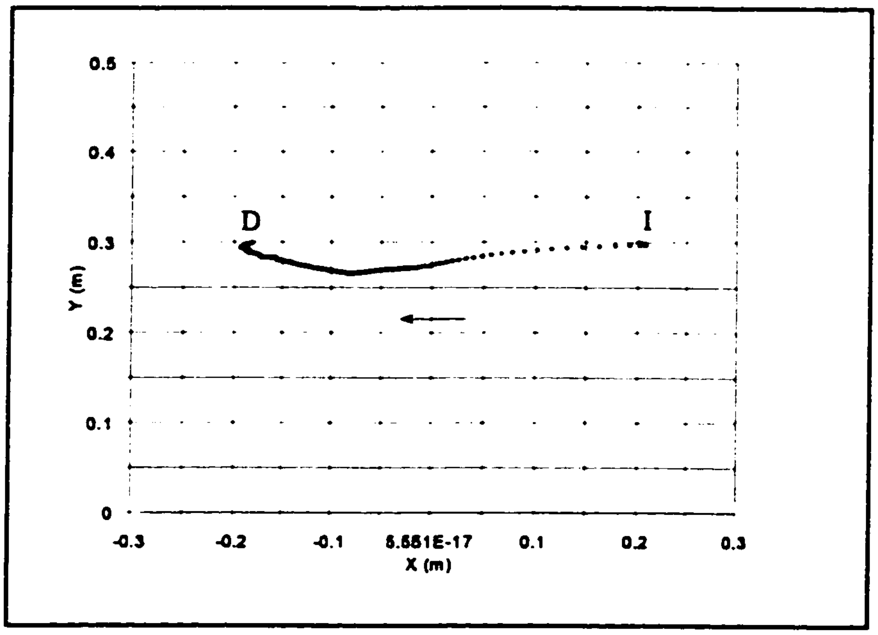

Figure 5-61 End-effector trajectory of robot arm when extended Kalman filter based on kinematic model was used and perturbation observer compensates the centrifugal, Coriolis and friction terms. (experimental results)

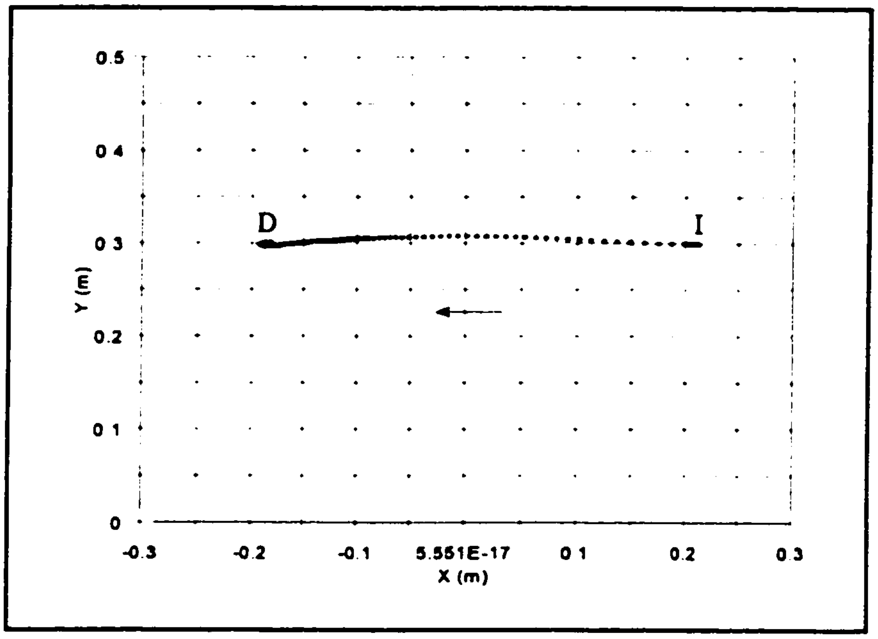

Figure 5-62 End-effector trajectory of robot arm when extended Kalman filter based on kinematic model was used and all dynamic terms were compensated by their approximate models. (experimental results)

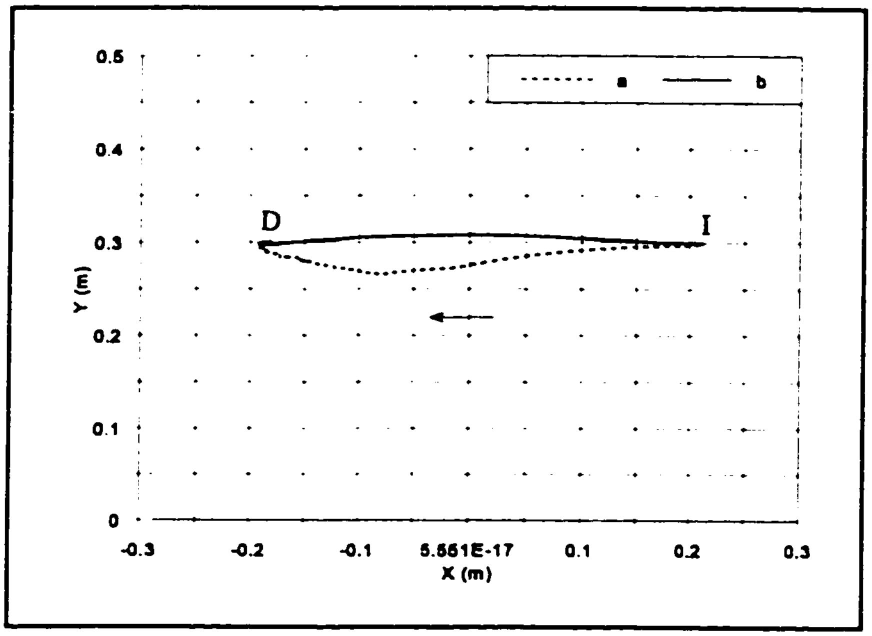

Figure 5-63 The results of both figures (5-61) and (5-62).

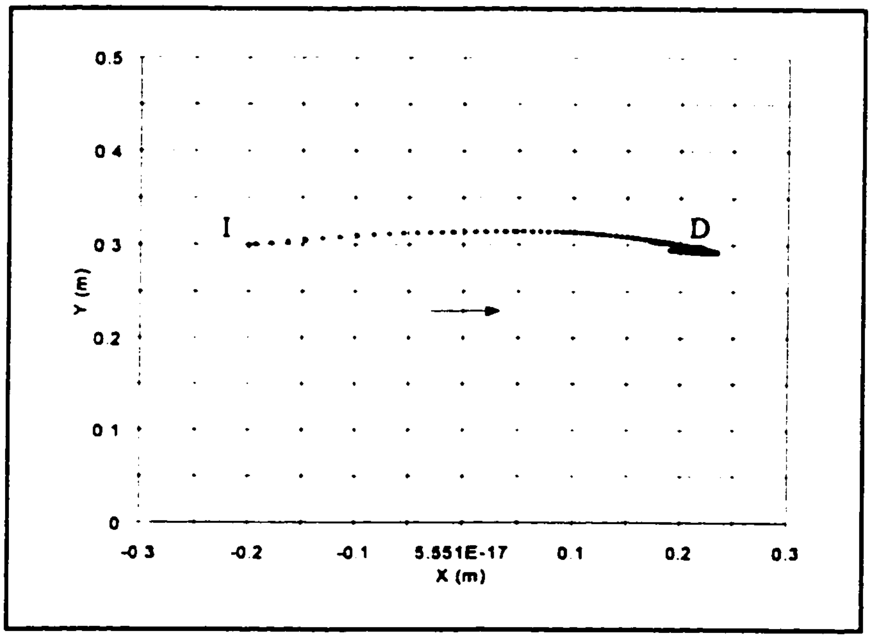

Figure 5-64 End-effector trajectory of robot arm when extended Kalman filter based on kinematic model was used and perturbation observer compensates the centrifugal, Coriolis and friction terms. (experimental results)

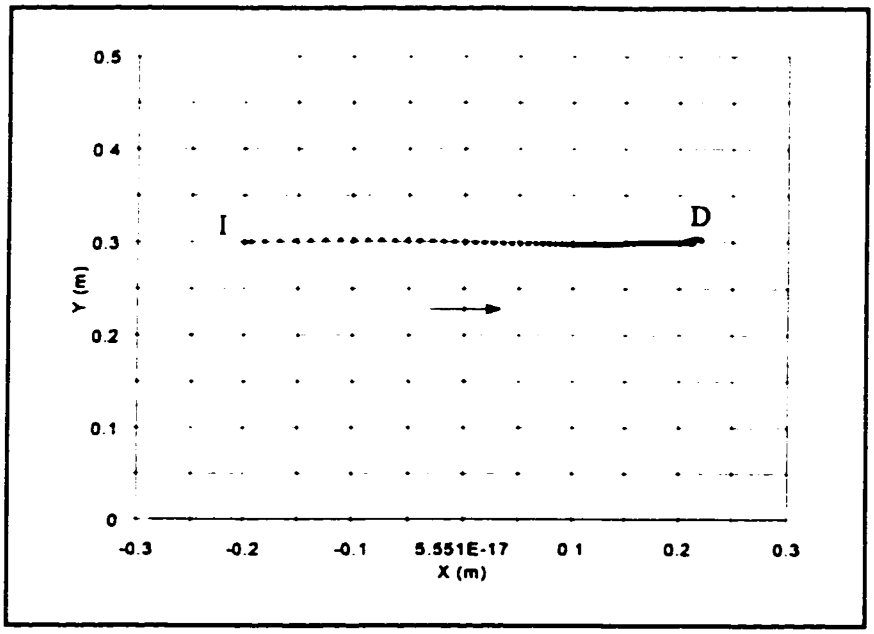

Figure 5-65 End-effector trajectory of robot arm when extended Kalman filter based on kinematic model was used and all dynamic terms were compensated by their approximate models. (experimental results)

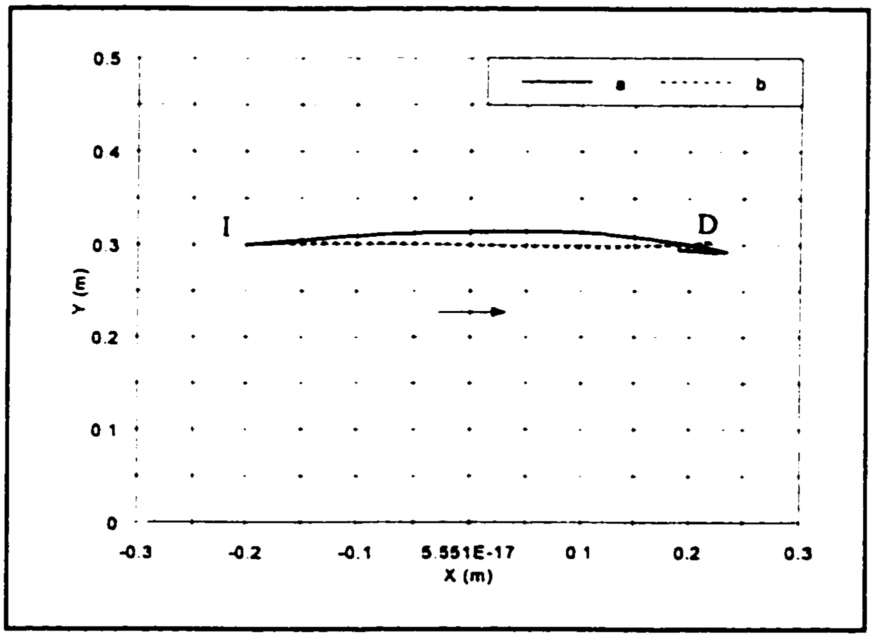

Figure 5-66 The results of both figures (5-64) and (5-65).

5.5.3.3 The impedance controller, using mixed approaches: perturbation observer and an Extended Kalman Filter, based on the dynamic model(EKF-D).

The Extended Kalman Filter based on kinematics, used in section 5.5.3.2, was replaced by an extended Kalman filter based on the dynamic model from section 5.5.2. The behavior of the robot using this new controller is shown in the following tests. Results, from these tests, show an improvement in controller performance. The tests carried out were the same tests as in section 5.5.3.2 .

Test #1:

The controller compensates for centrifugal, Coriolis, and friction dynamic terms(without using approximate models of dynamic terms). The results show the robustness of this controller, as shown in the figure (5-67). This figure shows the end-effector trajectory, from point I to point D. In this test, the controller can reach the target, with a close to straight line trajectory.

Test #2:

This test is almost the same as test #1, except that the centrifugal, Coriolis, and friction dynamic terms were compensated with their approximate models. The observer, therefore, only compensates the difference between actual and approximate models of dynamic terms. The results of this test can be seen in the figure (5-68). These results show that the trajectory improved, slightly, in respect to test #1. The results of both test #1 and test #2 are shown in the figure (5-69).

Tests #3 & #4:

These tests are same tests as #1 and #2, except that the initial and desired positions were changed. The results of test #3 are shown in the figure (5-70) and the results of test #4 are shown in the figure (5-71). The results of both tests are shown in the figure(5-72). The results again prove that an impedance controller using dynamic terms compensation and an observer will provide better results.

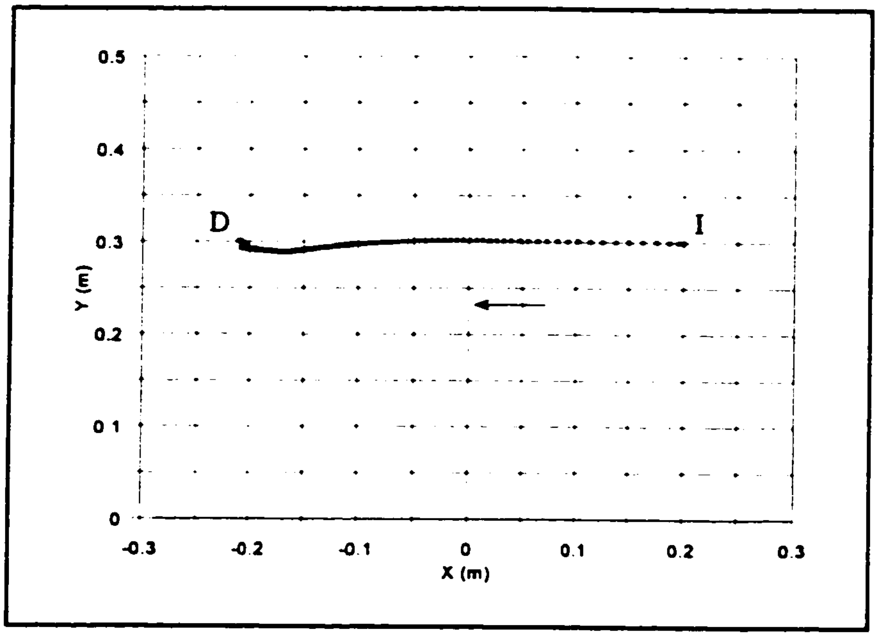

Figure 5-67 End-effector trajectory of robot arm when extended Kalman filter based on dynamic model was used and perturbation observer compensates the centrifugal, Coriolis and friction terms. (experimental results)

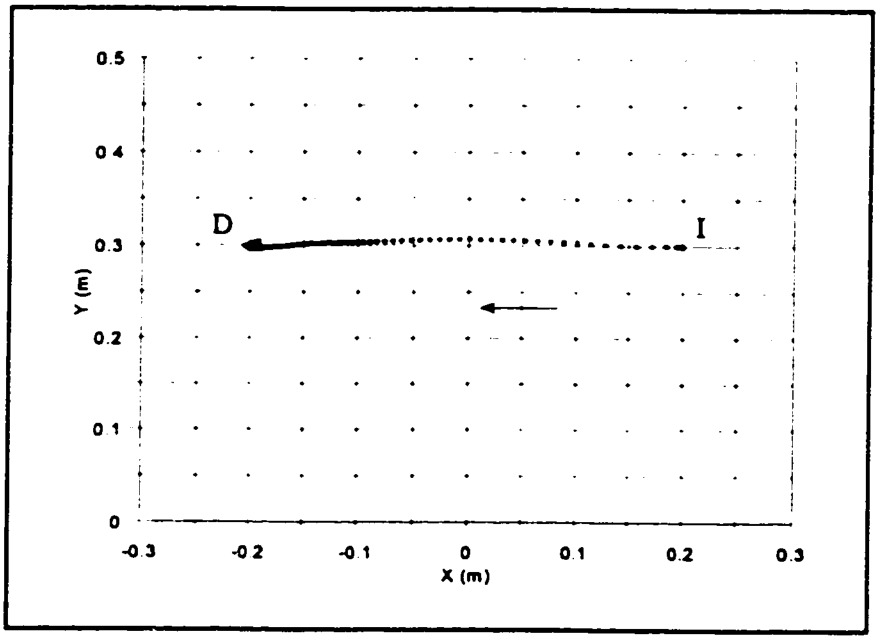

Figure 5-68 End-effector trajectory of robot arm when extended Kalman filter based on dynamic model was used and all dynamic terms were compensated by their approximate models. (experimental results)

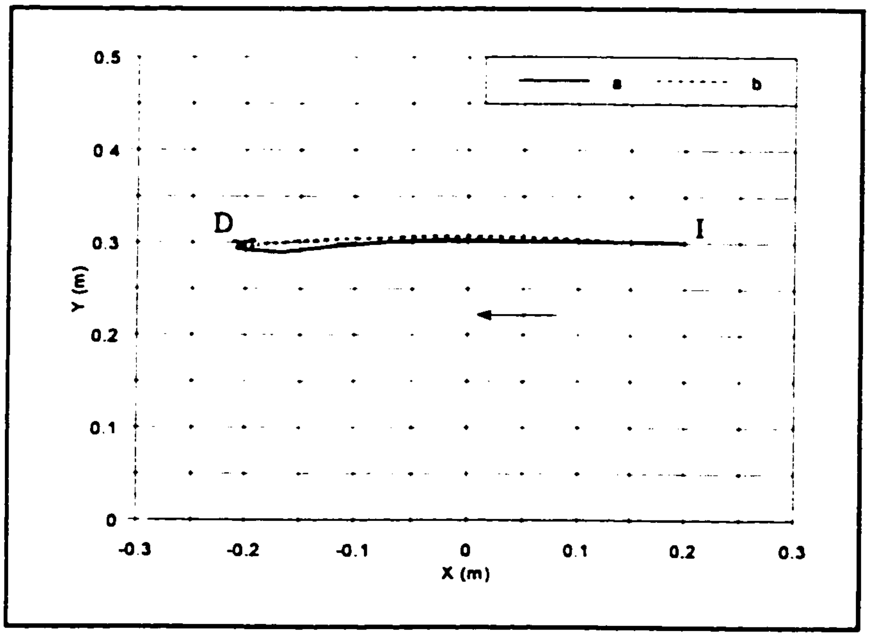

Figure 5-69 The results of both figures (5-67) and (5-68).

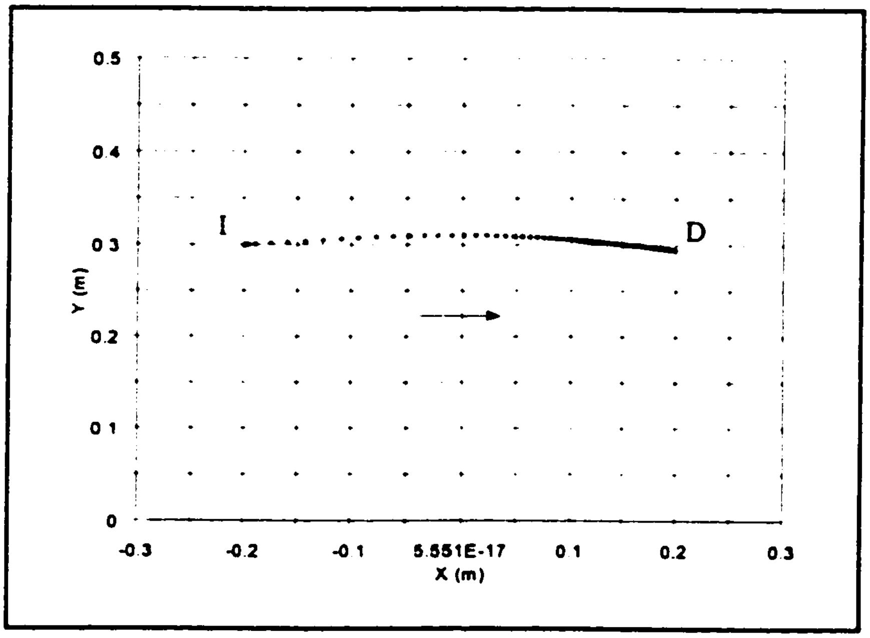

Figure 5-70 End-effector trajectory of robot arm when extended Kalman filter based on dynamic model was used and perturbation observer compensates the centrifugal, Coriolis and friction terms. (experimental results)

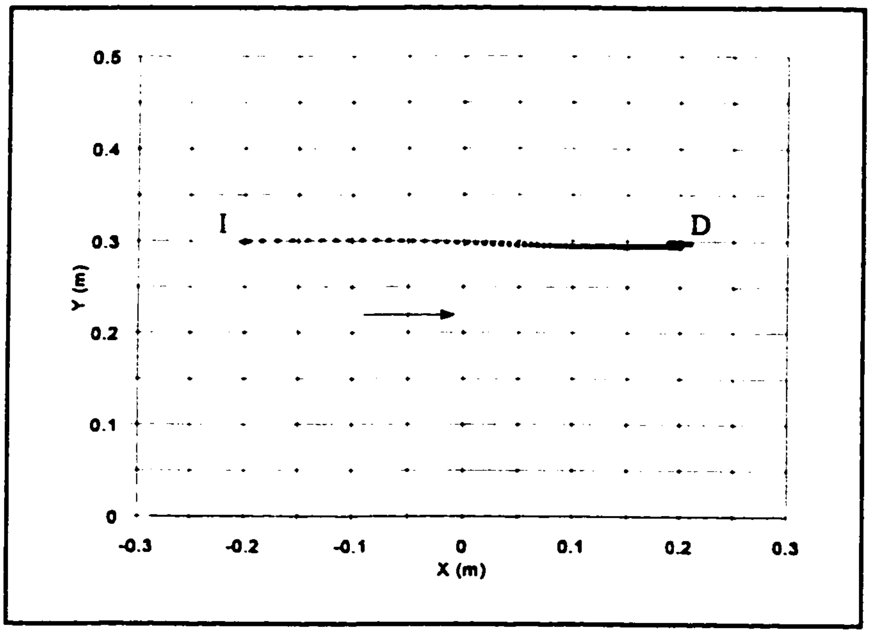

Figure 5-71 End-effector trajectory of robot arm when extended Kalman filter based on dynamic model was used and all dynamic terms were compensated by their approximate models. (experimental results)

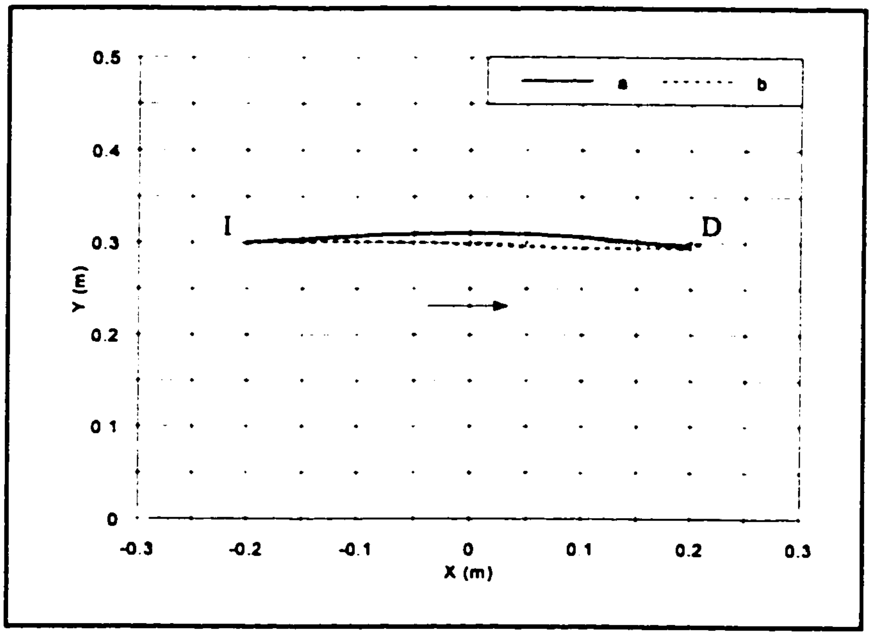

Figure 5-72 The results of both figures (5-70) and (5-71).

The results of sections of 5.5.3.2 and 5.5.3.3 have also been compared and are displayed in the figures (5-73) to (5-76).

Results using the extended Kalman filter, based on the kinematic model and dynamic model, for compensating centrifugal, Coriolis, and friction dynamic terms are shown in the figure (5-73). Results of a further comparison using a different trajectory is shown in the figure (5-74). As these figures show, the Kalman filter based on the Dynamic model provided better results than the Kalman filter based on the Kinematic model, since the trajectory is closer to a straight line..

Figures (5-75) and (5-76) show the results of using the extended Kalman filter based on Kinematic and Dynamic models in parallel with dynamic terms (centrifugal, Coriolis, friction) compensation using approximate models. When both methods are compared, these results show fewer improvements in the robot trajectory.

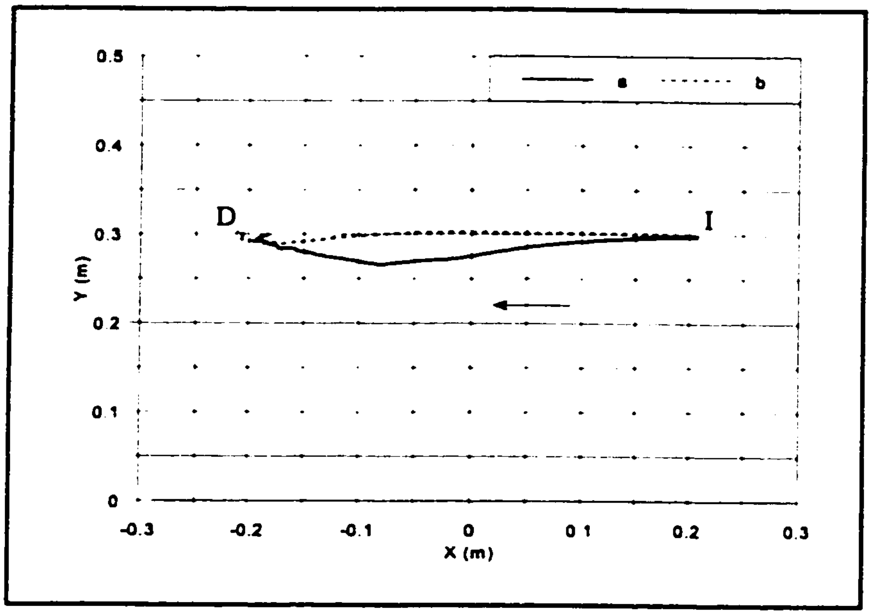

Figure 5-73 End-effector trajectory of robot arm using extended Kalman filter based on kinematic model(curve (a)) and dynamic model (curve(b)) for compensating the centrifugal, Coriolis and friction dynamic terms. (experimental results)

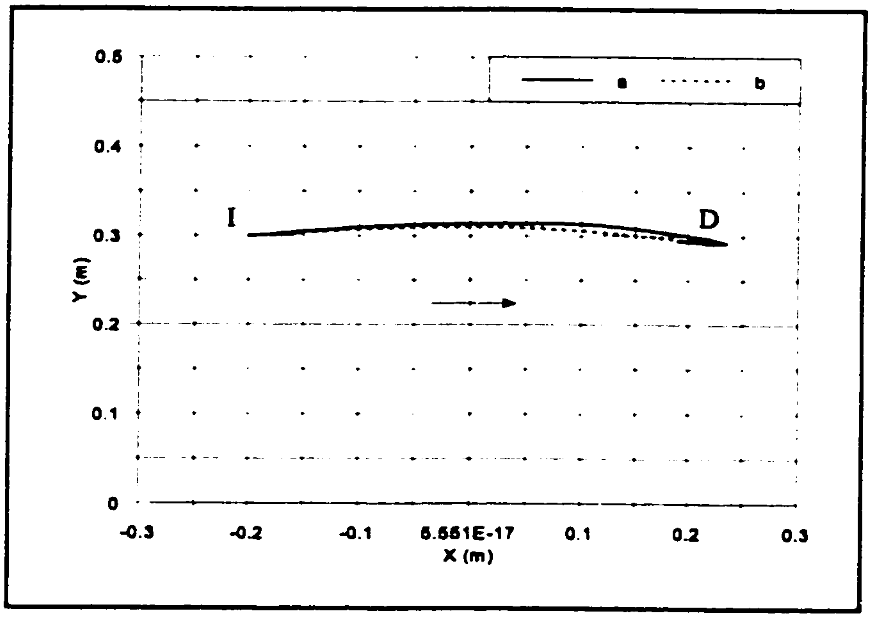

Figure 5-74 End-effector trajectory of robot arm using extended Kalman filter based on kinematic model(curve(a)) and dynamic model (curve(b)) for compensating the centrifugal, Coriolis and friction dynamic terms. (experimental results)

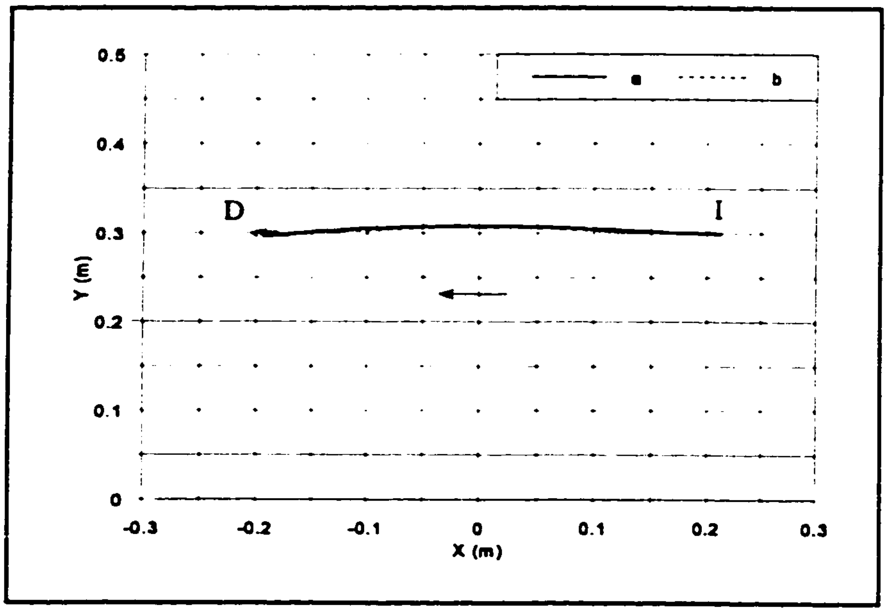

Figure 5-75 End-effector trajectory of robot arm using extended Kalman filter based on kinematic model(curve(a)) and dynamic model (curve(b)) in parallel with the dynamic terms compensation approximate model. (experimental results)

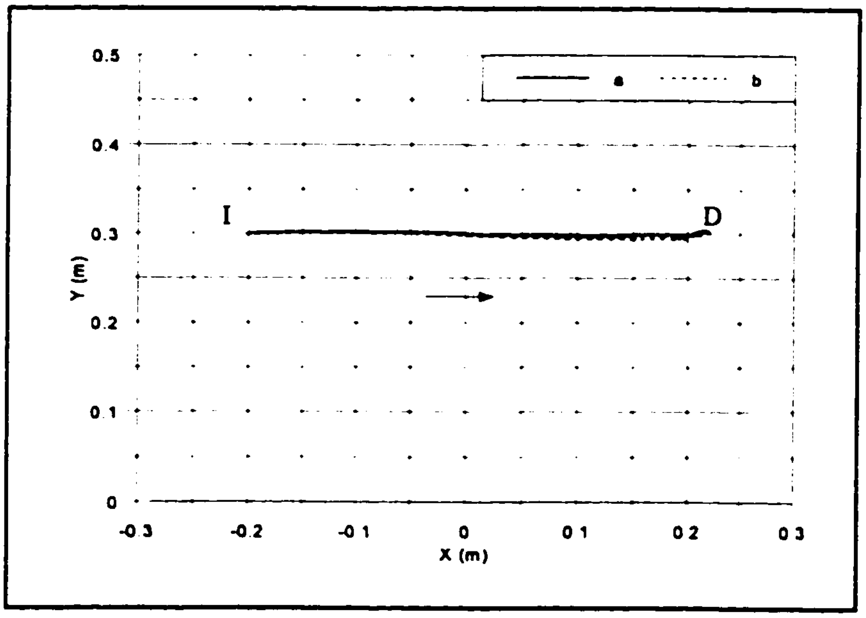

Figure 5-76 End-effector trajectory of robot arm using extended Kalman filter based on kinematic model(curve(a)) and dynamic model (curve(b)) in parallel with the dynamic terms compensation approximate model. (experimental results)

5.5.3.4 Other topics in mixed approaches using the Extended Kalman Filter

There were other topics regarding the Extended Kalman Filters, which required further study. First topic was to compare the results of sections 5.5.3.2 and 5.5.3.3 with the trajectory of the robot arm using a controller based on the acceleration signal from the second derivative of the joint position. Results of the three methods are shown in the figure (5-77). EKF-D gave slightly better results than the other two methods since it creates closer to straight line trajectory respect to other methods. This test was based on centrifugal, Coriolis, and friction dynamic terms compensation. Figure (5-78) shows the results when the dynamic terms were compensated by approximate models and an observer. The results show that the trajectory based on extended Kalman filters was slightly better than the trajectory based on the second derivative of a position signal.

Second topic was to test the effect of prediction for all methods. The effect of prediction in the observer was discussed in section 5.4 by using numerical methods. However, the extended Kalman filter also predicts the states by using equation(5-104) in the extended Kalman filter algorithm. The extended Kalman filter, based on the dynamic model EKF-D, was tested for this purpose. Results of the robot trajectories are shown in the figure (5-79). The trajectory of the end-effector without prediction of the acceleration signal is shown in curve(a). Results based on a predicted acceleration signal are shown in curve(b). Further test results for this comparison are shown in the figure (5-80). The results show that prediction will slightly improve controller behavior.

Third topic was to test the prediction effect on signals again, using the numerical method as it was used in section 5.4 . Figures (5-81) to (5-83) show the results of these tests. They show the slight improvement gained by using the prediction method. Curve (a) represents the results of the perturbation observer without a prediction signal. Curve (b) represents the results of the perturbation observer with prediction. Figure(5-81) shows results based on the second derivative of a joint position signal. Figure (5-82) shows results using the extended Kalman filter based on the kinematic model. Figure (5-83) shows results using the extended Kalman filter based on the dynamic model.

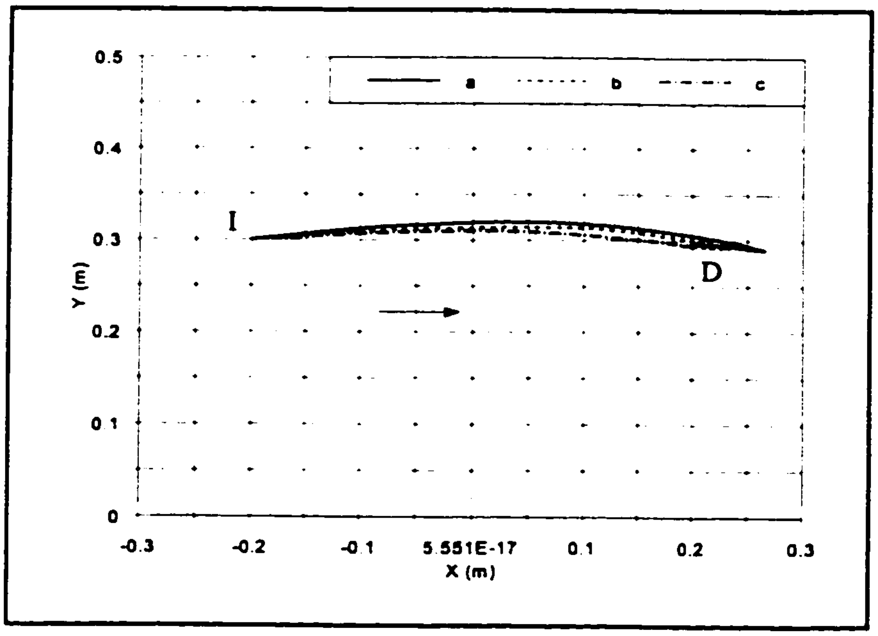

Figure 5-77 End-effector trajectory of robot arm based on three methods: (a) second derivative of joint positions, (b) EKF-K and (c) EKF-D, compensate centrifugal, Coriolis and friction terms. (experimental results)

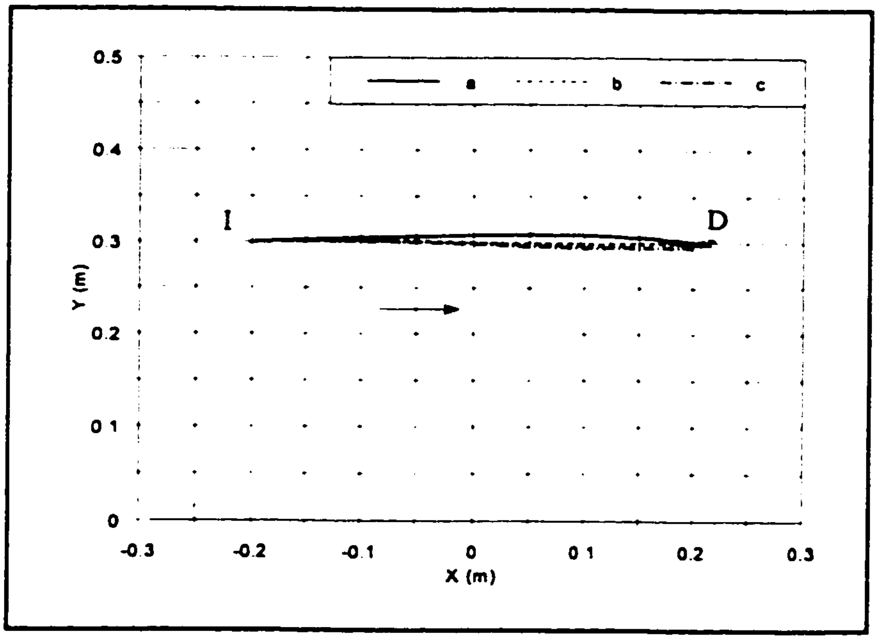

Figure 5-78 End-effector trajectory of robot arm based on three methods: (a) second derivative of joint positions, (b) EKF-K and (c) EKF-D, in parallel with approximate compensation models of dynamic terms. (Experimental results)

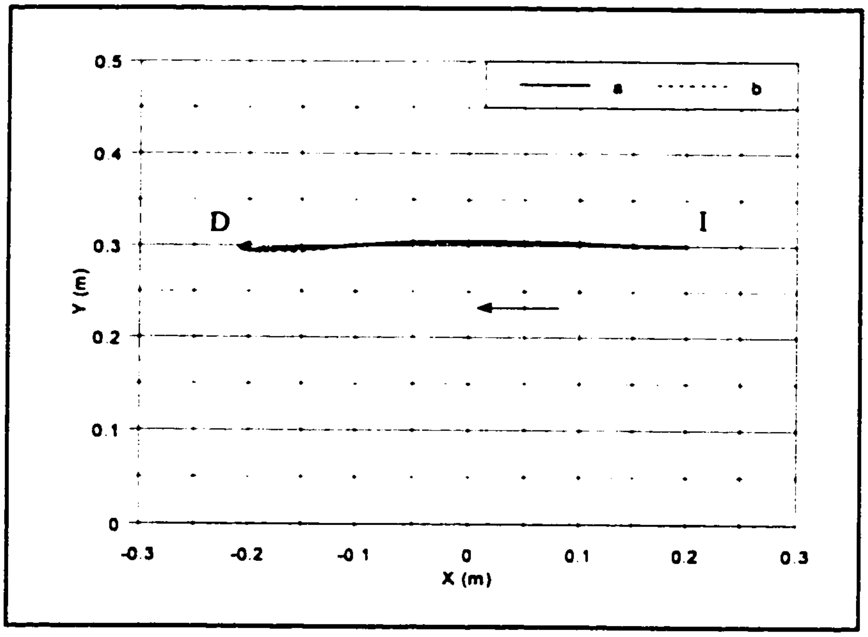

Figure 5-79 End-effector trajectory based on EKF-D and prediction(using Kalman filter method) of acceleration signal, (a) without prediction, (b) with prediction. (experimental results)

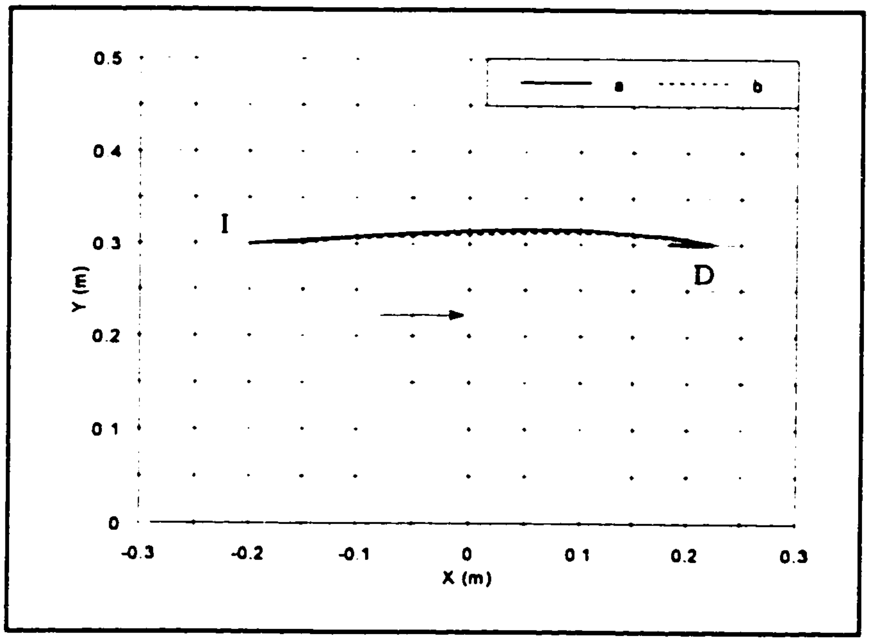

Figure 5-80 End-effector trajectory based on EKF-D and prediction(using Kalman filter method) of acceleration signal, (a) without prediction, (b) with prediction. (Experimental results)

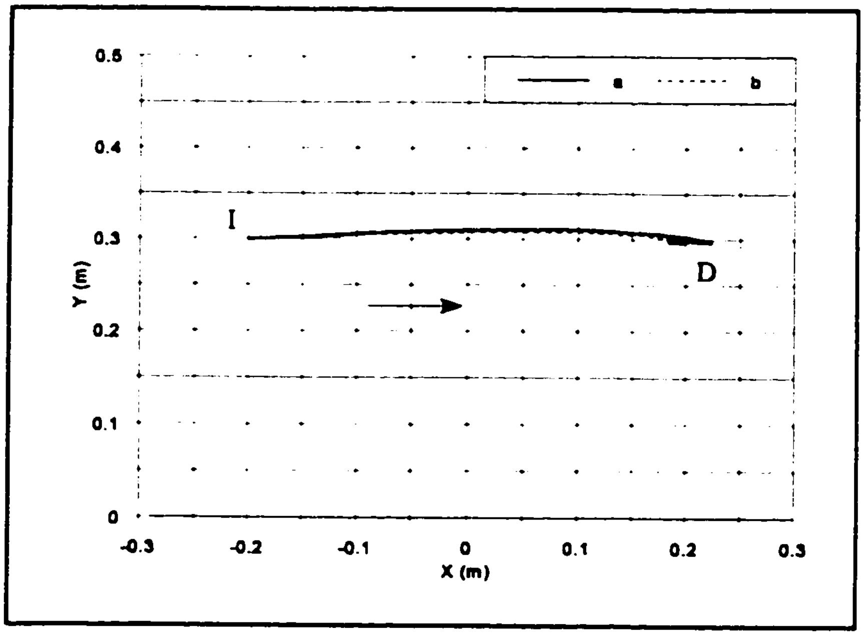

Figure 5-81 End-effector trajectory based on second derivative of joint position and prediction(using numerical method) of acceleration signal, (a) without prediction, (b) with prediction. (experimental results)

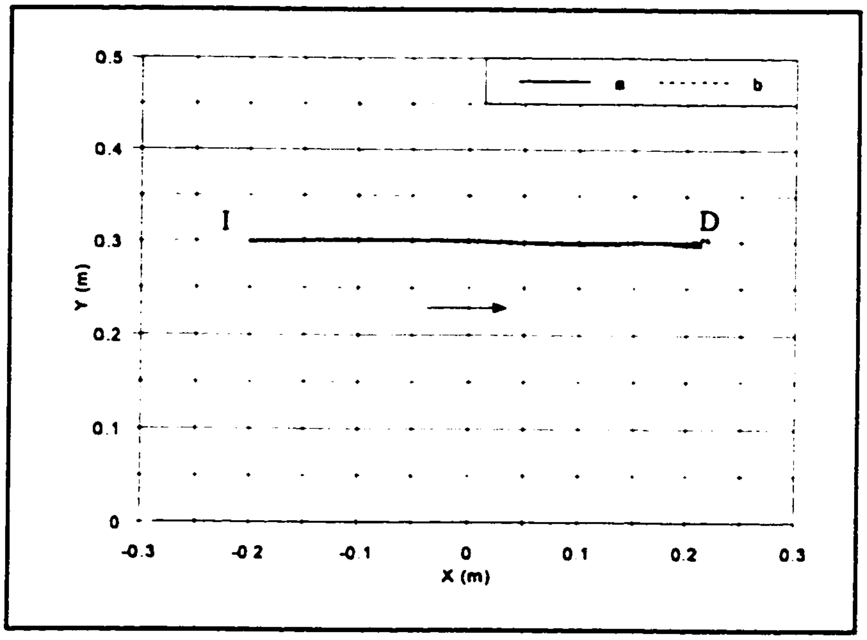

Figure 5-82 End-effector trajectory based on EKF-K and prediction(using numerical method) of acceleration signal, (a) without prediction, (b) with prediction. (experimental results)

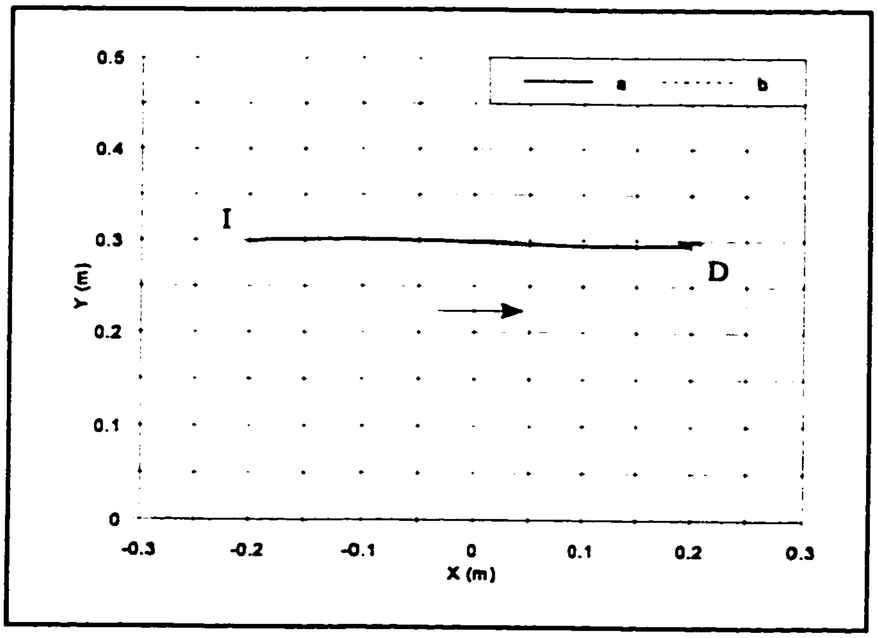

Figure 5-83 End-effector trajectory based on EKF-D and prediction(using numerical method) of acceleration signal. (a) without prediction, (b) with prediction. (experimental results)

6.0 CONCLUSIONS AND RECOMMENDATIONS

As in all research, this work, as it progressed, presented the author with several topics that were very interesting. Only the topics which were expected to produce more relevant results were pursued in detail. The following describes the results achieved in these topics, through theoretical research and experimentation.

Building a dual-arm robot:

The dual-arm planar robot, used in this thesis, was designed and built based on a very simple structure. Besides sensors which give relative displacement measurements(motor resolver), accelerometers were used to provide supplementary information regarding the robot behaviour. This information allows us to achieve a better understanding of robotic dynamics and its parameters. The experimental setup and software, developed in this thesis, will give in the future to interested individuals the capability of an easy transfer from the simulation to experimental study.

Parameter identification was carried out for the robot arms and certain conclusions were reached. The first conclusion was that use of the "Square Root Covariance Update Algorithm" provided better results than those obtained by other methods. The second conclusion reached was that as robot dynamic equations were expanded from four to six parameters, better results were achieved.

Improvements in impedance control:

Before the work reported in this thesis, the impedance controller was reported as only analytically tested and was proven as a strong candidate for real time trajectory generation and obstacle avoidance in single arms. During experimental work however, various problems were experienced revealing weaknesses of this impedance controller.

A known problem of computed torque controllers is the lack of a model for unknown dynamics. Unknown dynamics are very difficult to model and compensate. This thesis proposes a simple method of compensating these unknown dynamic terms and improving the capabilities of an impedance controller. This method proposes the addition of a perturbation observer.

Another impedance controller problem, clearly visible in experimental results, but not in simulations, is robustness. Addition of a perturbation observer creates a very robust impedance controller.

The addition of a perturbation observer increases the effectiveness of the impedance controller. In this thesis, however, a perturbation observer was not only added to the impedance controller, but also the observer was modified to improve its performance. In this thesis, two parts of the perturbation observer are improved. These improvements focus on observer predictability and the quality of signal feedback to it.

Acceleration signals are always very noisy signals. A low pass filter is required to filter out these noises. This filtering creates an observer with a very small bandwidth, especially if the acceleration signals are calculated numerically.

Since perturbation observer performance improves with the use of higher bandwidth signals, accelerometers were added to the observer. This combination of accelerometer and resolver signals produces also a more accurate joint acceleration estimation.

Performance and bandwidth of the controller were further improved by using sensor fusion. Sensor fusion produces improvements in experimental results. Sensor fusion of signals was carried out using an Extended Kalman Filter.

The Extended Kalman Filter method proved to be a reliable sensor fusion algorithm, which can fuse signals on line for every sampling time. The algorithm improves the observer's bandwidth and produces a more accurate acceleration signal. The impedance controller also remained robust when using the perturbation observer with the Extended Kalman Filter. The controller was able to compensate dynamic terms, including centrifugal, Coriolis, and friction terms.

Results would be further improved if the observer would be used only for compensating the difference between actual and estimated dynamic models. The Extended Kalman Filter based on dynamic terms produced slightly better results than the extended Kalman filter based on the kinematic model. This was due to a slight increase in the observer's bandwidth and a better estimation of states, such as joint accelerations.

The perturbation observer has also been improved by use of a predictor. This observer extension improved robotic behaviour. The effectiveness of a predictor extension, however, becomes important only when high sampling times are required. This is because the perturbation observer is a one step predictor and therefore predicts dynamic model error from a one-step-back time delay.

Control dual-arm robot using impedance controller

The coordinated motion of a dual-arm robot is presented based on the artificial impedance approach. Also, introducing an artificial impedance between the end-effectors improved the capability of the dual-arm controller for handling a non-rigid object. The simulation and experimental results show that this approach is a worthwhile candidate for solving some of the difficulties of coordinated dual-arm robot operation both in the approaching phase(free motion) and in the object handling phase, especially for cases when the robot is handling objects which may sag or are in general non-rigid.

The experimental work proves further that the perturbation observer improved the performance of an impedance controller for trajectory generation of the dual-arm robot

Experimental work and results:

Experimental work carried out in this thesis was extremely enlightening in what concerns the understanding of terms that are given little importance in simulations. Most of the new ideas and improvements presented in this thesis were tested experimentally. Simulation has always been the best environment to develop new ideas. True testing, however, occurs only when new ideas are tested experimentally.

Recommendations:

There so many topics which can be follow up this work. A list of few topics are recommended for future students whom will be working in this area. They are as following:

- further study of posture collision avoidance,
- use of a perturbation observer to compensate external forces on the end-effectors from a payload point of view, in dual-arm case scenarios,
- equip the end-effector with a force/torque sensor to measure external forces and to compensate them,
- further study of flexible beam handling and the damping of beam vibrations,
- further study of handling flexible object by six DOF or redundant robot with the same methodology which was used in this thesis.

7.0 REFERENCES

[1] Roach, J., Boaz, M.,Coordinating the Motion of Robot Arms in a Common Workspace, IEEE Conf. on Robotics and Automation, 1985, pp. 494-499.

[2] Luh, J.Y., and Zheng, Y.F., Constrained Relations Between Two Coordinated Industrial Robots for Motion Control, Int. Journal of Robotics Research, 1987, No.3, pp.60-70.

[3] Zheng, Y.F., Luh, J.Y., and Jia, P.F., A Real-Time Distributed Computer System for Coordinate Motion Control of Two Industrial Robots, IEEE Int. Conf. on Robotics and Automation, 1987, pp. 1236-1241.

[4] Ozguner, U., Yurkovich, S. and Al-Abbass, F., Decentralized Variable Structure Control of a Two-arm Robotic System, IEEE Int. Conf. on Robotics and Automation, 1987, pp. 1248-1254.

[5] Swern, F.L., and Tricamo, S.J., An Approach to Controlling Multi-Arm Robotic Manipulation of a Single Body, IEEE Int. Conf. on Robotics and Automation, 1988, pp. 516-521.

[6] Kazerooni, H., and Tsai, T.I., Compliance Control and Unstructured Modelling of Cooperating Robots, IEEE Int. Conf. on Robotics and Automation, 1988, pp. 510-515.

[7] Ahmad, S., and Guo, H., Dynamic Coordination of Dual-Arm Robotics Systems with Joint Flexibility, IEEE Int. Conf. on Robotics and Automation, 1988, pp. 332-337.

[8] Uchiyama, M., and Dauchez, P., A Symmetric Hybrid Position/Force Control Scheme for the Coordination of Two Robots, IEEE Int. Conf. on Robotics and Automation, 1988, pp. 350-356.

[9] Kopf, C.D., and Yabuta, T., Experimental Comparison of Master/Slave and Hybrid two arm Position/Force Control, IEEE Int. Conf. on Robotics and Automation, 1988, pp. 1633-1637.

[10] Suh, I.H., and Shin, K.G., Coordination of Dual Robots Arms Using Kinematic Redundancy, IEEE Trans. on Robotics and Automation, 1989, No.2, pp. 236-242.

[11] Necsulescu, D.S., and Jassemi-Zargani, R., Impedance Control for Coordinated Dual-Arm Robot Motion, CSA. Contract Report, 1990.

[12] Necsulescu, D.S., Jassemi-Zargani, R., and Graham, W.B., Trajectory Generation for Dual-Arm Robots Using Artificial Impedance Approach, Canadian Conf. on El. And Comp. Eng., Ottawa, Sept.3-6, 1990, pp. 50.1.1-4.

[13] Necsulescu, D.S., Jassemi-Zargani, R., and Graham, W.B., Task Space Coordination of Dual-Arm Motion, Sixth Int. Conf. on CAD/CAM, Robotics and Factories of the Future, August 1991, South Bank Polytechnic, London.

[14] Shih, C.L., Sadler, J.P., and Gruver, W.A., Collision Avoidance for Two Scara Robots, IEEE Int. Conf. on Robotics and Automation, April 1991, pp. 674-679.

[15] Yoshiha, K., Kurazume, R., and Umetani, Y., Dual-Arm Coordination in Space Free-Flying Robot, IEEE Int. Conf. on Robotics and Automation, April 1991, pp.2516-1521.

[16] Tao, J.M., and Luh, J.Y.S., Position and Force Controls for two Coordinating Robots, IEEE Int. Conf. on Robotics and Automation, April 1991, pp. 176-181.

[17] Uchiyama, M., and Yamashita, T., Adaptive Load Sharing for Hybrid Controlled Two Cooperative Manipulators, IEEE Int. Conf. on Robotics and Automation, April 1991, pp. 986-991.

[18] Yun, X., Kumer, V., Sarkar, N., and Paljug, E., Control of Multiple Arms with Rolling Constraints, IEEE Int. Conf. on Robotics and Automation, May 1992, pp. 2193-2198.

[19] Yao, B., Gao, W.B., Chan, S.P., and Cheng, M., VSC Coordinated Control of two Manipulator Arms in the Presence of Environmental Constraints, IEEE Trans. of Automatic Control, Nov. 1992, Vol.37, No.11, pp.1806-1812.

[20] Chu, H., and Elmaraghy, H.A., Real-Time Multi-Robot Path Planner Based on a Heuristic Approach, IEEE Int. Conf. on Robotics and Automation, May 1992, pp. 475-480.

[21] Derventzis, C.A., and Davison, E.J., Robust Motion/Force Control of Cooperative Multi-Arm Systems, IEEE Int. Conf. on Robotics and Automation, May 1992, pp. 2230-2237.

[22] Chiacchio, P., Chiaverini, S., and Siciliano, B., Cooperative Control Schemes for Multiple Robot Manipulator Systems, IEEE Int. Conf. on Robotics and Automation, May 1992, pp. 2218-2223.

[23] Hsu, P., and Su, S., Coordinated Control of Multiple Manipulator Systems, Experimental Results, IEEE Int. Conf. on Robotics and Automation, May 1992, pp. 2199-2204.

[24] Liu, G.J., and Goldenberg, A.A., Robust Hybrid Impedance Control of Robot Manipulators, IEEE Int. Conf. on Robotics and Automation, April 1991, pp. 287-292.

[25] Hemmami, A., Kinematic Admissibility in Coordinated Motion of two Robot Arms, Proc. ICCME, Vol.4, May 1993, Shiraz, Iran, pp. 119-126.

[26] Zheng, Y.F., Chen, M.Z., Trajectory Planning for two Manipulators to Deform Flexible Beams, IEEE Int. Conf. on Robotics and Automation, 1993.

[27] Dellinger, W.F., Anderson, J.N., Interactive Force Dynamics of two Robotic Manipulators Grasping a Non-Rigid Object, IEEE Int. Conf. on Robotics and Automation, 1992, pp. 2205-2210.

[28] Paljug, E., Yun, X., Experimental Results of two Robot Arms Manipulating Large Objects, IEEE Int. Conf. on Robotics and Automation, 1993.

[29] Williams, D., Khatib, O., The Virtual Linkage a Model for Internal Forces in Multi-Grasp Manipulation, IEEE Int. Conf. on Robotics and Automation, 1993, pp. 1025-1030.

[30] Xi, N., Tran, T.J., Bejczy, A.K., Event-Based Planning and Control for Multi-Robot Coordination, IEEE Int. Conf. on Robotics and Automation, 1993, pp. 251-258.

[31] Wolovich, W.A., Robotics: Basic Analysis and Design, Holt, Rinehart and Winston, 1987.

[32] Craig, J.J., Introduction to Robotics, Addison-Wesley, 1986.

[33] Freund, E., Fast Nonlinear Control with Arbitrary Pole Placement for Industrial Robots and Manipulators, The Int. Journal of Robotics, No. 1, 1982, pp. 67-78.

[34] Stepanenko, Y., State-Feedback Robot Control, in International Encyclopaedia of Robotics Applications and Automation, 1988, J. Wiley & Sons, pp. 1667-1684.

[35] An, C.H., Atkeson, G.G., Griffiths, J.D., and Hollerbach, J.M., Experimental Evaluation of Feedforward and Computed Torque Control, IEEE Trans. on Robotics and Automation, 1989, No. 3, pp. 368-373.

[36] Spong, M., and Vidyasagar, M., Robust Linear Compensator Design for Nonlinear Robotic Control, IEEE Journal on Robotics and Automation, No. 4, 1987, pp. 345-351.

[37] Khatib, O., Commande dynamique dans l'espace opérationnel des robots manipulateurs en présence d'obstacles, Thèse-docteur ingénieur, ENSAE, Toulouse, 1980.

[38] Khatib, O., A Unified Approach for Motion and Force Control of Robot Manipulators: The Operational Space Formulation, IEEE Journal of Robotics and Automation, No. 1, 1987, pp. 43-53.

[39] Andrews, J.R., Hogan, N., Impedance Control as a Framework for Implementing Obstacle Avoidance in a Manipulator, Manufacturing Processes and Robotic Systems, Eds. Hardt, D., Book, W., ASME, 1983, pp. 243-251.

[40] Hogan, N., Impedance Control: An Approach to Manipulation, Journal of Dynamic Systems, Measurement and Control, Vol. 107, March 1985, pp. 1-24.

[41] Hogan, N., Stable Execution of Contact Tasks Using Impedance Control, The Proceedings of the IEEE International Conference on Robotics and Automation, 1987, pp. 1047-1054.

[42] Wolovich, W.A., and Elliott, H., A Computational Technique for Inverse Kinematics, Proc. 23rd CDC, Dec. 1984, pp. 1359-1362.

[43] Necsulescu, D.S., and Jassemi-Zargani, R., Implementation Issues of an Impedance Controller for a Robot Manipulator, CSA. Contract Report, 1992.

[44] Komada, S., Ohinishi, K., Force feedback Control of Robot Manipulator by the Acceleration Tracing Orientation Method, IEEE Trans. on Industrial Electronics, vol. 37,No. 1., Feb. 1990, pp.6-12.

[45] Hsia, T.C.S., Lasky, T.A., Guo, Z., Robust Independent Joint Controller Design for Industrial Robot Manipulators, IEEE Transactions on Industrial Electronics, vol. 38,no. 1, Feb. 1991.

[46] Yousef-Toumi, K., Fuhlbrigge, T.A., Application of a Decentralized Time Delay Controller to Robot Manipulators, IEEE, Int. Conf. on Robotics and Automation, 1988, pp.1786-1791.

[47] Lu,W.S., Meng,Q.H., Impedance Control with Adaptation for Robotic Manipulations, IEEE Transaction on Robotics and Automation, Vol.7., No.3, June,1991.

[48] Liu, G.J., Goldenberg, A.A., Robust Hybrid Impedance Control of Robot Manipulators, IEEE Int. Conf. on Robotics and Automation, April,1991.

[49] Necsulescu, D.S., and Jassemi-Zargani, R., Experimental Study of Efficient Control Schemes for Dual-Arm robot Manipulation of Flexible Payloads, CSA. Contract Report, 1994.

[50] Jassemi-Zargani,R., and Necsulescu, D.S., Operational Space Virtual Impedance Approach to Single and Dual Arm Robot Control, Proc. ICCME, Vol.4, May 1993, Shiraz, Iran, pp. 143-150.

[51] Zheng, Y.F., Integration of Multiple Sensors into a Robotic System and its Performance Evaluation, IEEE Trans. on Robotics and Automation, Oct. 1989, pp. 658-669.

[52] Hager, G.D., Engelson, S., Atiya, S., On Comparing Statistical and Set-Based Methods in Sensor Data Fusion, Proc. IEEE Int. Conf. on Robotics and Automation, 1993, pp. 352-358.

[53] Durrant-Whyte, H., Integration, Coordination and Control of Multi-Sensor Systems, Kluwer, Boston, 1988.

[54] Hager, G.D., Task-Directed Sensor Fusion and Planning, Kluwer, 1990.

[55] Chevanier, F.,Crowley, J.L., Position Estimation for a Mobile Robot Using Vision and Odometry, proc. IEEE int. conf. on Robotics and Automation, 1992, pp. 2588-2593.

[56] D'Orazio, T., et al., Mobile Robot Navigation by Multi-Sensory Integration, Proc. IEEE Int. Conf. on Robotics and Automation, 1993, pp. 373-379.

[57] Hong, L., Scaggs, T., Real-Time Optimal Multi-resolutional Sensor/Data Fusion, Proc. IEEE Int. Conf. on Robotics and Automation, 1993, pp. 117-122.

[58] Sabater, A., Thomas, F., Set Membership Approach to the Propagation of Uncertain Geometric Information, Proc. IEEE Int. Conf. on Robotics and Automation, 1991, pp. 2718-2723.

[59] Atiya, S., Hager, G., Real-Time Vision-Based Robot Localization, Proc. IEEE Int. Conf. on Robotics and Automation, 1991, pp. 639-644.

[60] Beckermen, M. A Bayes-Maximum Entropy Method for Multi-Sensor Data Fusion, Proc. IEEE Int. Conf. on Robotics and Automation, 1992, pp. 1668-1674.

[61] Abdulghafour, M., Chandar, T., Abidi, M.A., Data Fusion Through Fuzzy Logic Applied to Feature Extraction from MultiSensory Images, Proc. IEEE Int. Conf. on Robotics and Automation, 1993, pp. 359-366.

[62] Bonitz, R.G., Hsia, T.C., Internal Force-Based Impedance Control for Cooperating Manipulators, IEEE Transactions on Robotics and Automation, Vol. 12, No. 1, Feb. 1996, pp.78-89.

[63] Bonitz, R.G., Hsia, T.C., Robust Internal Force-Based Impedance Control for Coordinating Manipulators, Proceedings of the IEEE International Conference on Robotics and Automation Magazine, Vol. 1, April 1996, pp.622-628.

[64] Bonitz, R.G., Hsia, T.C., The Effects of Computational Delays in Coordinating Multiple-Arm Manipulating a Rigid Object Using Robust Internal Force-Based Impedance Control, Proceedings of the 6th International Symposium on Robotics and Manufacturing, 2nd World Automation Congress, May 1996, Invited Paper.

[65] Kraus, W.Jr., McCarragher, Control of Flexible Load Deformations and Environment Contact Forces in Dual-Arm Manipulation, Proceedings of the 13th IFAC World Congress, San Francisco, July 1996,pp.61-67.

[66] Kraus, W.Jr., McCarragher, Hybrid Position/Force Coordination for Dual-Arm Manipulation of Flexible Materials, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'97),8-13 September 1997, Grenoble, France.

[67] Bonitz, R.G., Hsia, T.C., Calibrating a Multi_Manipulator Robotic System, IEEE Robotics and Automation Magazine, March 1997.

[68] TEXAS INSTRUMENTS, TMS320C3x User's Guide, 1990.

[69] DSP-CITpro, Hardware Manual, v.1., dSPACE digital Signal Processing and Control Engineering GmbH, West Germany.

[70] DSP-CITpro, TRACE Manual, v.1., dSPACE digital Signal Processing and Control Engineering GmbH, West Germany.

[71] DSP-CITpro, IMPEX Manual, v.1., dSPACE digital Signal Processing and Control Engineering GmbH, West Germany.

[72] Megatorque Motor System, User's Manual, Motion and Control Technology NSK NIPPON SEIKO K.K., Japan, 1989.

[73] Accelerometer Notes, Nova Sensor, CA 1989, ACCESS Sensors SA, Switzerland, 1992.

[74] National Semiconductor Corporation , Catalogue and Notes on Buffer and Filters, 1989.

[75] Bierman, G.J., Factorization Methods for Discrete Sequential Estimation, Academia Press, New York, 1977.

[76] Blevins, R.D., Formula for Natural Frequency and Mode Shape, Van Nostrand Reinhold Ltd., 1979.

[77] Necsulescu, D.S., and Jassemi-Zargani, R., Analytical and Experimental Study in Dual-Arm Manipulation of Flexible Payloads", CSA. Contract Report, 1993.

[78] Necsulescu D.S., Jassemi-Zargani R., Kalaycioglu S., Dual Arm Operational Space Control Using Efficient Impedance Modulation, 8th CASI Conference, Nov. 1994, Ottawa, pp. 483-491.

[79] Jassemi-Zargani, R., Artificial Impedance Approach to the Collision Avoidance of Robots with Obstacles, M.A.Sc. Thesis, Sept. 1990, U. of Ottawa, Ottawa, Canada.

[80] Necsulescu D.S., Jassemi-Zargani R., Graham W.B., Impedance Control for Robotic Manipulation, Second workshop on Military Robotic Application, August 8-11, 1989, at RMC, Kingston, Canada.

[81] Slotine, J.E., Li, W., Applied Nonlinear Control, Prentice-Hall, Englewood Cliff, New Jersey, 07632, 1991.

[82] Luh, J.Y.S., Walker, M.W., and Paul, R.P.C., Resolved Acceleration of Mechanical Manipulators, IEEE Trans. of AC, No. 3, 1980, pp. 236-241.

[83] An, C.H. and Hollerbach, J.M., The Role of Dynamic Models in Cartesian Force Control of Manipulators, The Int. J. of Robotics Research, No. 4, 1989, pp. 51-72.

[84] Komada, S., Shin, J., Ishida, M., Ohnishi, K., Hori, T., Simplification of Inverse Kinematics Routine of Robot Manipulators Using Disturbance Observer in Task Space, IEEE proc. on Robotics and Automation, France, May 1992, pp. 1381-1386.

[85] Boyce, W.E., DiPrima, R.C., Elementary Differential Equations and Boundary Value Problems, John Wiley & Sons, 1977.

[86] Goodwin, G.C., and Sin, K.S., Adaptive Filtering Prediction and Control, Prentice-Hall Inc., 1984.

[87] Jassemi-Zargani R., Necsulescu D.S., Kalaycioglu S., Multisensor Fusion for Operational Space Control of a Single and Dual Arm Robots, 21st Annual Conference of the IEEE Industrial Electronics Society, Nov.6-10,1995, Orlando, Florida.

[88] Borrie, J.A., Modern Control Systems, Prentice-Hall, International (UK) Ltd., 1986.

[89] Kiguchi, K., Robot Contact motion Control Using Environment Impedance Identification-Application of Neural Network, M.A.Sc. Thesis, Sept. 1994, U. of Ottawa, Ottawa, Canada.

[90] Ogata,K., Modern Control Engineering, 2nd ed.,Prentice-Hall, Englewood Cliff, New

Jersey, 07632, 1990.

[91] Hogan, N., Impedance Control: An Approach to Manipulation: Part.I-Theory, Part.II-Implementation, Part.III-Applications, Journal of Dynamic Systems, Measurement and Control, Trans. Of ASME, Vol. 107, March 1985.

APPENDICES

Appendix A

(Kinematics and Dynamics Models)

A.1 KINEMATICS:

Robot kinematics is the study of manipulator motion in terms of position, velocity, and acceleration. It consists in assigning coordinate frames, homogeneous transformations (for relating the frames), and location of the end-effector position based on joint positions. The Jacobian matrix also provides the relationships between joint velocity and acceleration to Cartesian velocity and acceleration.

In this research, Denavit-Hartenberg conventions(figure(A-1)) were used for homogeneous transformations[32].

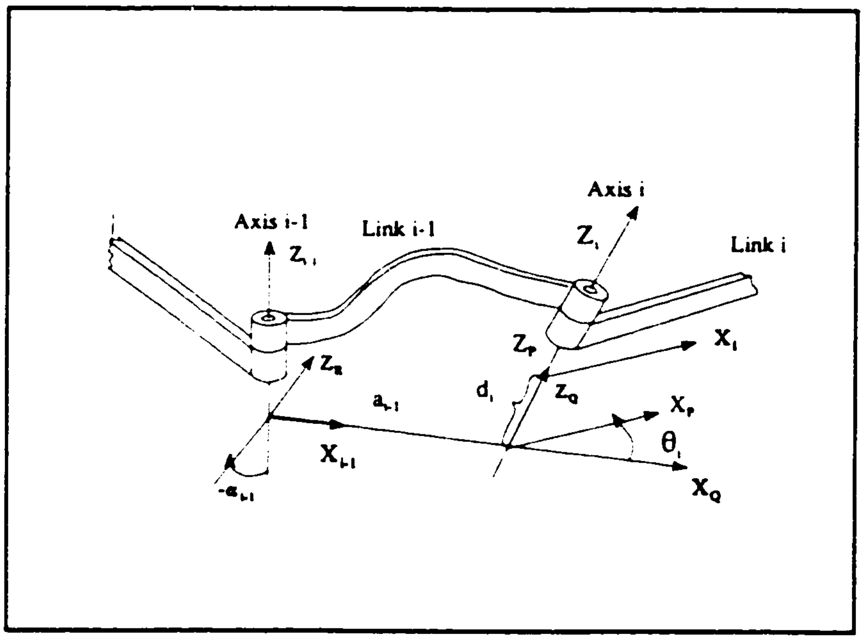

Figure A-1 Denavit-Hartenberg Convention for joint i and i+1

The homogeneous transformation(T) between the i and i-1 frame is:

$$T_i^{i-1} = \begin{bmatrix} C_{\theta_i} & -S_{\theta_i} & 0 & a_i \\ S_{\theta_i} C_{\alpha_{i-1}} & C_{\theta_i} C_{\alpha_{i-1}} & -S_{\alpha_{i-1}} & -S_{\alpha_{i-1}} d_i \\ S_{\theta_i} S_{\alpha_{i-1}} & C_{\theta_i} S_{\alpha_{i-1}} & C_{\alpha_{i-1}} & C_{\alpha_{i-1}} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A-1}$$

where:

a_i = the distance from z_i to z_{i+1} measured along x_i

α_i = the angle between z_i and z_{i+1} measured about x_i

d_i = the distance from x_{i-1} to x_i measured along z_i

θ_i = the angle between x_{i-1} and x_i measured about z_i

$C\theta_i$ = $\cos(\theta_i)$

$S\theta_i$ = $\sin(\theta_i)$

$C\alpha_{i-1}$ = $\cos(\alpha_{i-1})$

$S\alpha_{i-1}$ = $\sin(\alpha_{i-1})$

Total transformation, between the end-effector frame and base coordinate frame, is:

$$T_N^0 = T_1^0 \; T_2^1 \; \ldots\ldots \; T_N^{N-1} \tag{A-2}$$

Since the dual-arm robot consists of two two-degree-of-freedom planar robots, the link parameters of each arm will be as follow:

i	α_{i-1}	a_{i-1}	d_{i-1}	θ_i
1	0	0	0	θ_1
2	0	l_1	0	θ_2
3	0	l_2	0	o

Table A-1 Link Parameters

The transformation matrix is:

$$T_3^0 = \begin{bmatrix} C_{12} & -S_{12} & 0 & l_1C_1+l_2C_{12} \\ S_{12} & C_{12} & 0 & l_1S_1+l_2S_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{(A-3)}$$

The constraint equations, which show the Cartesian position of the end-effector, are:

$$\begin{aligned} x &= l_1C_1+l_2C_{12} \\ y &= l_1S_1+l_2S_{12} \\ z &= 0 \end{aligned} \qquad \text{(A-4)}$$

A.1.1 Inverse Kinematics

To find the joint position from the end-effector position, inverse kinematics is required. This is accomplished by squaring both sides of equation (A-4), and then adding the equations, as follows:

$$x^2=l_1^2C_1^2+l_2^2C_{12}^2+2l_1l_2C_1C_{12}$$
$$y^2=l_1^2S_1^2+l_2^2S_{12}^2+2l_1l_2S_1S_{12}$$

add the equations:

$$x^2+y^2=l_1^2+l_2^2+2l_1l_2C_2 \qquad \text{(A-5)}$$

$$\therefore \quad C_2=\frac{A}{2l_1l_2}$$

where: $A=x^2+y^2-l_1^2-l_2^2$

angles for joint one and two can therefore be calculated using the following equation:

$$\theta_2=atan2(B,A)$$

where:

$$B=\sqrt{4l_1^2l_2^2-A^2} \qquad \text{(A-6)}$$

$$\theta_1 = atan2(y,x) - atan2\left(\frac{l_2S_2}{l_1+l_2C_2}\right)$$

A.1.2 Jacobian Matrix [32]

The nonlinear transformation of the joint variable:

$$\theta(t) \in R^n \tag{A-7}$$

to Cartesian space:

$$\begin{aligned} &y(t) \in R^p \\ &y = h(\theta) \end{aligned} \tag{A-8}$$

is given by the Jacobian matrix, defined in association with $h(\theta)$:

$$J(\theta) = \frac{\partial h(\theta)}{\partial \theta} \tag{A-9}$$

The Jacobian matrix are used for velocity and acceleration transformations, from joint space to Cartesian space, as follows:

$$\begin{aligned} &\dot{y} = J\dot{\theta} \qquad (velocity) \\ &\ddot{y} = \dot{J}\dot{\theta} + J\ddot{\theta} \qquad (acceleration) \end{aligned} \tag{A-10}$$

Using equation (A-9) and the constraints equation (A-4), the Jacobian matrix can be calculated as:

$$\begin{aligned} &y = (h_1, \ h_2, \ldots\ldots h_n) \\ &\theta = (\theta_1, \ \theta_2, \ldots\ldots \theta_m) \\ &J(\theta) = \begin{bmatrix} \frac{\partial h_1}{\partial \theta_1} & \frac{\partial h_1}{\partial \theta_2} & \cdot & \cdot & \frac{\partial h_1}{\partial \theta_m} \\ \frac{\partial h_2}{\partial \theta_1} & \frac{\partial h_2}{\partial \theta_2} & \cdot & \cdot & \frac{\partial h_2}{\partial \theta_m} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \frac{\partial h_n}{\partial \theta_1} & \frac{\partial h_n}{\partial \theta_2} & \cdot & \cdot & \frac{\partial h_n}{\partial \theta_m} \end{bmatrix} \end{aligned} \tag{A-11}$$

For a two-degree-of-freedom planar robot, the constraint equations, from equation(A-4), are:

$$\begin{aligned} x &= l_1 C_1 + l_2 C_{12} \\ y &= l_1 S_1 + l_2 S_{12} \\ \alpha_z &= \theta_1 + \theta_2 \end{aligned} \qquad \text{(A-12)}$$

From equations (A-11 and A-12), the Jacobian can therefore be calculated as:

$$J(\theta) = \begin{bmatrix} \dfrac{\partial x}{\partial \theta_1} & \dfrac{\partial x}{\partial \theta_2} \\ \dfrac{\partial y}{\partial \theta_1} & \dfrac{\partial y}{\partial \theta_2} \\ \dfrac{\partial \alpha_z}{\partial \theta_1} & \dfrac{\partial \alpha_z}{\partial \theta_2} \end{bmatrix} = \begin{bmatrix} -l_1 S_1 - l_2 S_{12} & -l_2 S_{12} \\ l_1 C_1 + l_2 C_{12} & l_2 C_{12} \\ 1 & 1 \end{bmatrix} \qquad \text{(A-13)}$$

Linear velocity (y) and linear generalized forces(f) are calculated as:

$$J_v(\theta) = \begin{bmatrix} -l_1 S_1 - l_2 S_{12} & -l_2 S_{12} \\ l_1 C_1 + l_2 C_{12} & l_2 C_{12} \end{bmatrix} \qquad \text{(A-14)}$$

and angular velocity(ω) or generalized torques(τ) are calculated as:

$$J_\omega(\theta) = \begin{bmatrix} 1 & 1 \end{bmatrix} \qquad \text{(A-15)}$$

A.2 ROBOT DYNAMICS

Most controllers designed by mechanical engineers are based on robot dynamics. Computed-torque controllers are a good example of using dynamic terms. These controllers usually use dynamic terms to linearize the nonlinear robot dynamics, such that a linear control system can be used to control the robot.

Newton and Lagrangian methods are the basic methods used. This is especially true since the recursive method was introduced.

A.2.1 Newton-Euler Recursive Method

Newton Euler mechanics is based on:

$$\begin{aligned} \frac{d(mv)}{dt} &= f = ma \qquad (where:\ a=\dot{v}) \\ \frac{d(I\omega)}{dt} &= \tau = I\dot{\omega} \end{aligned} \tag{A-16}$$

where:

m = mass of the body

v = centre of mass velocity of the body

f = sum of external forces applied on the body

I = moment of inertia of the body

ω = angular velocity of the body

τ = sum of external torques applied on the body

Newton-Euler formulations treat each link as an individual entity. They describe an individual link in terms of linear and angular motion. Since each link, however, is coupled to another link, equations describing a link contain coupling forces and torques that also appear in equations describing neighbouring links. By determining all these coupling terms a description of the manipulator, as a whole, is eventually obtained. The recursive Newton-Euler method consists of two parts (forward and backward recursion)[32].

Forward Recursion:

The forward recursion equations are from frame i=0 to i=n-1, and are described as follows:

- *Angular velocity*:

$$\omega_{i+1}^{i+1} = R_i^{i+1}\omega_i^i + \dot{\theta}_{i+1} z_{i+1}^{i+1}$$

- *Angular acceleration*:

$$\dot{\omega}_{i+1}^{i+1} = R_i^{i+1}\dot{\omega}_i^i + R_i^{i+1}\omega_i^i \times \dot{\theta}_{i+1}\, z_{i+1}^{i+1} + \ddot{\theta}_{i+1} z_{i+1}^{i+1}$$

- *Linear acceleration of each link*:

$$\dot{v}_{i+1}^{i+1} = R_i^{i+1}(\dot{\omega}_i^i \times p_{i+1}^i + \omega_i^i \times (\omega_i^i \times p_{i+1}^i) + \dot{v}_i^i)$$

- *Linear acceleration of centre of mass*: **(A-17)**

$$\dot{v}_{c_{m\theta}}^{i+1} = \dot{\omega}_i^i \times p_{c_{m\theta}}^i + \omega_i^i \times (\omega_i^i \times p_{c_{m\theta}}^i) + \dot{v}_i^i$$

- *Effective inertial forces*:

$$F_{i+1}^{i+1} = m_{i+1}\ \dot{V}_{c_{m\theta}}^{i+1}$$

- *Effective inertial couples*:

$$N_{i+1}^{i+1} = I_{i+1}^{c_{m\theta}}\dot{\omega}_{i+1}^{i+1} + \omega_{i+1}^{i+1} \times I_{i+1}^{c_{m\theta}}\omega_{i+1}^{i+1}$$

Backward Recursion:

The backward recursion equations for frame i=n.....1 are as follows:

- *Force exerted on link i by link i-1:*

$f_i^i = R_{i+1}^i f_{i+1}^{i+1} + F_i^i$

- *Torque exerted on link i by link i-1:*

$$n_i^i = N_i^i + R_{i+1}^i n_{i+1}^{i+1} + p_{c_i}^i \times F_i^i + p_{i+1}^i \times R_{i+1}^i f_{i+1}^{i+1} \qquad \text{(A-18)}$$

- *Torque applied by each actuator:*

$\tau_i = n_i^{i\,T} z_i^i$

Equations (A-17 and A-18) were used to derive the dynamic model of a two-degree-of-freedom planar robot, as follows:

Initial conditions:

- Angular velocity, angular acceleration, and linear acceleration are:

$$\omega_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \dot{\omega}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \dot{v}_0^0 = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} \qquad \text{(A-19)}$$

- centre of mass position and coordinate positions are:

$$p_{c_1}^1 = \begin{bmatrix} a_1 \\ 0 \\ 0 \end{bmatrix} \quad p_{c_2}^2 = \begin{bmatrix} a_2 \\ 0 \\ 0 \end{bmatrix} \quad p_1^0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad p_2^1 = \begin{bmatrix} l_1 \\ 0 \\ 0 \end{bmatrix} \quad p_3^2 = \begin{bmatrix} l_2 \\ 0 \\ 0 \end{bmatrix} \qquad \text{(A-20)}$$

where: a_1 and a_2 are the centre of mass position with respect to the i-1,2 coordinates.

External forces, or $f^3_3=[f_x,f_y,f_z]$, (total force applied on the end-effector) must be considered. Among others, these forces could be the reaction forces related to the payload or contact motion. Let us therefore consider $n^3_3=[\tau_x,\tau_y,\tau_z]$, an external torque applied on end-effector. These torques could result from the payload or contact motion.

The resulting final dynamic equation of the robot is:

$$\begin{bmatrix}\tau_1\\ \tau_2\end{bmatrix}=\begin{bmatrix} I_1+l_1^2m_1+a_1^2m_1+I_2+a_2^2m_2+2l_1a_2c_2m_2 & I_2+a_2^2m_2+a_2l_1m_2C_2\\ I_2+a_2^2m_2+l_1a_2C_2m_2 & I_2+a_2^2m_2\end{bmatrix}\begin{bmatrix}\ddot\theta_1\\ \ddot\theta_2\end{bmatrix}+$$

$$\begin{bmatrix}-l_1a_2S_2m_2(\dot\theta_2^2+2\dot\theta_1\dot\theta_2)\\ l_1a_2S_2m_2\dot\theta_1^2\end{bmatrix}+\begin{bmatrix}l_1S_2 & l_2+l_1C_2 & 1\\ 0 & l_2 & 1\end{bmatrix}\begin{bmatrix}f_x\\ f_y\\ \tau_z\end{bmatrix} \qquad \text{(A-21)}$$

therefore the general equation for the robot dynamic is:

$$\tau = M(\theta)\ \ddot\theta + v(\theta,\dot\theta)+G(\theta)+F(\dot\theta)+J^Tf_{ext} \qquad \text{(A-22)}$$

This equation can also be written in the state space format as follows:

$$\tau = M(\theta)\ \ddot\theta + B(\theta)\,[\dot\theta\dot\theta] + C(\theta)\,[\dot\theta^2]+G(\theta)+F(\dot\theta)+J^Tf_{ext} \qquad \text{(A-23)}$$

where:

Inertia:

$M(\theta) \rightarrow [n \times n]$

$$M(\theta) = \begin{bmatrix} I_1+l_1^2m_1+a_1^2m_1+I_2+a_2^2m_2+2l_1a_2c_2m_2 & I_2+a_2^2m_2+a_2l_1m_2C_2\\ I_2+a_2^2m_2+l_1a_2C_2m_2 & I_2+a_2^2m_2\end{bmatrix} \qquad \text{(A-24)}$$

Torques:

$$\tau \rightarrow [n \times 1]$$
$$\tau = [\tau_1 \quad \tau_2]^T \qquad \text{(A-25)}$$

Centrifugal and Coriolis term:

$$v(\theta,\dot{\theta}) \rightarrow [n \times 1]$$
$$v(\theta,\dot{\theta}) = \begin{bmatrix} -l_1 a_2 S_2 m_2 (\dot{\theta}_2^2 + 2\dot{\theta}_1 \dot{\theta}_2) \\ l_1 a_2 S_2 m_2 \dot{\theta}_1^2 \end{bmatrix} \qquad \text{(A-26)}$$

Coriolis term:

$$\begin{cases} B(\theta) \rightarrow [n \times n(n-1)/2] \\ [\dot{\theta}\dot{\theta}] = [\dot{\theta}_1 \dot{\theta}_2 \quad \dot{\theta}_1 \dot{\theta}_3 \ \ldots\ldots \dot{\theta}_{n-1} \dot{\theta}_n]^T \rightarrow [n(n-1)/2 \times 1] \end{cases}$$
$$\text{(A-27)}$$
$$\begin{cases} B(\theta) = -2 l_1 a_2 S_2 m_2 \\ [\dot{\theta}\dot{\theta}] = \dot{\theta}_1 \dot{\theta}_2 \end{cases}$$

Centrifugal term:

$$\begin{cases} C(\theta) \rightarrow [n \times n] \\ [\dot{\theta}^2] = [\dot{\theta}_1^2 \quad \dot{\theta}_2^2 \ \ldots\ldots \ \dot{\theta}_n^2]^T \rightarrow [n \times 1] \end{cases}$$

$$\begin{cases} C(\theta) = \begin{bmatrix} 0 & -l_1 a_2 S_2 m_2 \\ l_1 a_2 S_2 m_2 & 0 \end{bmatrix} \\ [\dot{\theta}^2] = [\dot{\theta}_1^2 \quad \dot{\theta}_2^2]^T \end{cases} \qquad \text{(A-28)}$$

Gravity term:

$G(\theta) = [n \times 1]$

$G(\theta) = 0$ (planar robot)

Friction term:

$$\begin{cases} F \rightarrow [n \times n] \\ (\dot{\theta}) = sgn(\dot{\theta}) \rightarrow [n \times 1] \end{cases}$$

$$\begin{cases} F = \begin{bmatrix} F_1 & 0 \\ 0 & F_2 \end{bmatrix} \\ (\dot{\theta}) = [sgn(\dot{\theta}_1) \quad sgn(\dot{\theta}_2)]^T \end{cases} \qquad \text{(A-29)}$$

External forces:

$$\begin{cases} J^T(\theta) \rightarrow [n \times n] \\ f_{ext} \rightarrow [6 \times 1] \end{cases}$$

$$\begin{cases} J^T(\theta) = \begin{bmatrix} l_1 S_2 & l_2 + l_1 C_2 & 1 \\ 0 & l_2 & 1 \end{bmatrix} \\ f_{ext} = \begin{bmatrix} f_x \\ f_y \\ \tau_z \end{bmatrix} \end{cases} \qquad \text{(A-30)}$$

The Jacobian is calculated with respect to the end-effector. This is because external forces applied to the end-effector are measured using the local coordinate system (end-effector coordinates). If the forces are transferred to a base coordinate, the Jacobian equation would have to be changed. Equation (A-14) shows the Jacobian with respect to the base coordinate system:

$$J(\theta) = \begin{bmatrix} -l_1 S_1 - l_2 S_{12} & -l_2 S_{12} \\ l_1 C_1 + l_2 C_{12} & l_2 C_{12} \\ 1 & 1 \end{bmatrix} \tag{A-31}$$

APPENDIX B

(Frequency Calculations)

B. NATURAL FREQUENCY CALCULATIONS OF THE LINKS

Since all the robot arms are considered as rigid bodies(Links drawings are presented in reference [89]), a simple frequency calculation is done on all the links to ensure that the frequency of the links is reasonably high. The Robert D. Blevins[76] book on "Formulas for Natural Frequency and Mode Shape" was used for this purpose. The general concept is shown in figure(B-1).

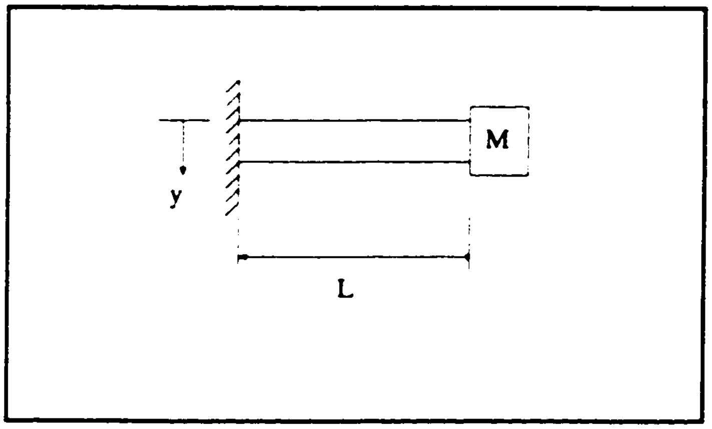

Figure B-1 Link model

Using table 8-15 of reference [76] (case #6), for calculating the frequency of the beams, the natural frequency equation is as follows:

$$f_i = \frac{\lambda_i}{2\pi L}\left(\frac{KG}{\mu}\right)^{1/2} \qquad i=1,2,3 \tag{B-1}$$

where:

λ_i = a dimensionless constant which is a function of the beam boundaries

λ_1 can be found from figure (8-3) of reference [76], or from $\cot \lambda_i = \left(\frac{M}{\mu A L}\right)\lambda$

G = shear modules $\left(G = \frac{E}{2(1+\upsilon)}\right)$

E = modulus of elasticity

υ = Poison's ratio

L = span of beams

M = mass

μ = beam material mass density

K = shear coefficient, where:

$$K = \frac{10(1+\upsilon)(1+3m^2)}{F(m,n)}$$

$$n = \frac{b}{h}$$

$$m = \frac{bt_1}{ht} \tag{B-2}$$

$$F(m,n) = 12+72m+150m^2+90m^3+\upsilon(11+66m+135m^2+90m^3)+10n^2[(3+\upsilon)m+3m^2]$$

The above equation can be used for frequency calculation of the links, as follows:

Link #1

Parameters:

M = 18 Kg

μ = 2700 kg/m^3

L = .2207 m

A = .0058468 m^2

n = m =1.544

υ = .334

$F(m,n) = 1361$

$K = .0799$

$E = 73 \times 10^9$

$G = 2.736 \times 10^{10}$

$\lambda_1 = .415$ (from fig. (8-23) of ref.[76])

therefore:

$f_1 = 270$ Hz

Link #2

Parameters:

$M = 2.5$ Kg

$\mu = 2700$ kg/m^3

$L = .16034$ m

$A = .0041583$ m^2

$n = m = .71875$

$\upsilon = .334$

$F(m,n) = 249$

$K = .13659$

$E = 73 \times 10^9$

$G = 2.736 \times 10^{10}$

$\lambda_1 = .75$ (from fig. 8-23 of ref.[76])

therefore:

$f_1 = 876$ Hz

Appendix C

(Procedure for using the experimental dual-arm robot))

C. PROCEDURE OF USING EXPERIMENTAL SET-UP

The following is a simple procedure for using MatrixX, IMPEX, MON30 and TRACE30 software. This tutorial will show the following steps:

- design of a simple Proportional-Velocity feedback controller
- use of MatrixX software, to create a controller block diagram
- use of IMPEX software, to descritize the controller and generate required "C" code
- modification of the "C" code, for any nonlinearity
- compilation of the "C" code object file
- use of MON30, to download the object file, initialize the DSP, and run the code.
- use of TRACE30 to trace any variable in the code.
- halting a program running in the DSP.

C.1 Design of a simple Proportional-Velocity feedback controller:

A proportional controller with velocity feedback for an inertial load J is shown in figure (C-1). The desired characteristics of this controller would be the damping ratio(ζ) and settling time(t_s).

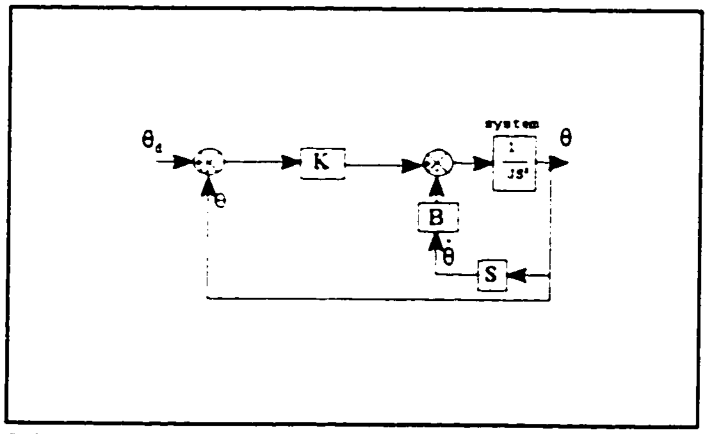

Figure C-1 Proportional and velocity feedback controller

The closed loop transfer function of this controller is:

$$\frac{\theta}{\theta_d} = \frac{K}{Js^2 + Bs + K} \tag{C-1}$$

where:

K = Proportional gain

B = Velocity feedback gain

By comparing to the canonic format of second order equations, it can therefore be concluded from equation (C-1) that:

$$\omega_n = \sqrt{\frac{K}{J}} \tag{C-2}$$

$$2\zeta\omega_n = \frac{B}{J} \tag{C-3}$$

where:

ζ = Damping ratio

ω_n = Natural frequency

t_s = Settling time

and for 5% criterion the settling time is[90]:

$$t_s = \frac{3}{\zeta\omega_n} \tag{C-4}$$

therefore K and B can be calculated as follows:

$$\omega_n = \frac{3}{t_s\zeta} \tag{C-5}$$

$$K = \omega_n^2 J \tag{C-6}$$

$$B = 2\zeta\sqrt{KJ} \tag{C-7}$$

The following data are given:

$\zeta = 1$ (for critical damping)

J = ? (inertial of the load and motor, ie. 1)

t_s = ? (user selected, ie. 5 sec)

For $B \neq 0$ the poles of the characteristic equation are on the left-hand side of s-plane, therefore, the system is stable.

C.2 Controller Design Using MatrixX

MatrixX was used to create the controller block diagram. Matlab (Simulink) can also be used. MatrixX must be run first and then follow these steps:

[Note: // represents an information comment]

- Run MatrixX from the root directory by typing "mat71"

C: mat71

// type build to run the super block module

<> build

- edit super block
    . name of block (E:\rahim\ . . .)
    . sampling time: 0 (continues system)

// go to environment

    . turn table forms on

- define block(#1)
    . location (1)
    . type of block (algebraic, sum vector)
    . block form
    . name of block
- define block(#2)
    . location (2)
    . dynamic system (PID)
    . name (PID)

- connect blocks
    . inter path (1,2)
    . external input

. vector dim(2) to block (1) (connection simple)

. external output

. vector dim(1) from block (2) (connection simple)

- exit to top menu
- analyze super-block (name of file)

// if the number of state (ns) is not zero then type

<> sim(' ')

- select an integration method // i.e. Ruge Kutta 4th order method

<> <s,ns> = lin(.001) // linearization

<> [A,B,C,D]= split(s,ns)

// in the event that ns=0, you must create your own A,B,C, and D matrices

<> define 'c:\matrixX\savebbl'

<> save bbl ('file name',A,B,C,D,T) // T=0 for continues time

<> exit // exit the MatrixX software

C.3 dSPACE Software

- Before using IMPEX software, convert the MatrxX file to a dSPACE format (bbl), as follows:

    - go to the directory where the file was saved (E:\rahim\...)
    - run the MTXBBL file by typing at prompt:
        . mtxbbl
    - or . d:\dsp_cit\exe\mtxbbl (example)
    - convert: filename.dat to *.bbl
    - save IMPEX file

IMPEX:

- Run IMPEX
    - File
        . select working directory
        . load system file
    - Descritize
        . step invariant
        . sampling time
    - Generate code
        . select I/O channels [Note: to change column use]
            . inputs [ctrl-arrow keys]
            . outputs
        . generate C code

IMPEX software creates three types of files:

- filename.c
- filename.trc
- filename.gcd

C.3.1 Compiling and Linking:

The C code which is generated by IMPEX contains a linear system. Code for any nonlinearity that IMPEX is unable to generate must be added by the user. A batch file called "cal" under the DSP_CIT\EXE directory can compile and link the C code as follows::

- cal filename (note: type filename without ".c")

This batch file will create an object and a map files: (*.obj, *.map).

MON30.exe

This software will download the object file to the DSP and run the code. The program can be stopped any time. Before running the MON30, make sure that the drivers for the motors are set to torque mode and AC mode. PROCOMM software can be used to setup the drivers(TM,AC). Also, (SED#) can be used for any changes in setup file(filename.sed).

- run the MON30
    - load object module(1) (type the file name)
    - check the channels to reset (clear all DS3001 channels on card 1,2)
    - restart the DSP
    - exit without stoping the DSP (if everything is safe)

After tracing can go back and stop the program by following procedure:

- run MON30
- -reset (stops the program in DSP)

TRACE30:

This software can trace any variable in code. The software can be run by typing:

- TRACE30
- select working directory
- select map file
- go to SETUP

- select number of points (total time = # points * sampling time)
- select the signals that you want to trace
- trace
- plot

These signals also can be saved in ASCII files. After your observations, you can exit the TRACE30 and go back to MON30 to stop the program.

APPENDIX D

(Motors, Drivers, Sensors, Links Specifications)

Table D-1 Motor & driver parameters

ITEM	UNITS	RS0810 FN001 NSK (MOTOR#1)	RS0408 FN001 NSK (MOTOR#2)
Maximum Torque	Kgf.m (N.m)	9 (88.2)	1 (9.8)
Maximum Velocity	rps	3	4.5
Maximum Friction Torque	Kgf.m (N.m)	0.45 (4.41)	0.10 (.98)
Axial Load Capacity	Kgf	460	180
Moment Load Capacity	Kgf.m (N.m)	8 (78.4)	2 (19.6)
Axial Stiffness	mm/Kgf(mm/N)	.00003(.0000031)	.000025(.0000026)
Moment Stiffness	rad/Kgf.m(rad/N)	.000025(.0000026)	.00003 (.0000031)
Rotor Inertia	$Kgf.m^2$	.084	.009
Weight	Kgf	24	6.5
Position Sensor		Brushless Resolver	Brushless Resolver
Resolver Resolution	Count/rev	614400	409600
Resolver Accuracy	arc-sec	+/- 30"	+/- 60"
Resolver Repeatability	arc-sec	+/-2.1"	+/-3.2"
Mating Driver Unit Type		EE0810A 05-24.2 NSK	EE0408C 05-22 NSK
Control system		Digital Servo	Digital Servo
Control Mode		Position Velocity Torque	Position Velocity Torque
Main AC Line	Volts	220 3@	110 1@
Maximum Current Output	Amp	7.5	6.0
Weight of Driver	Kgf	7.4	7.4

Table D-2 Acceleration sensors specifications

	NOVA	ACCESS
Range	+/- 2 g	±1 g
Output (analog)	0-5 Vdc	2.5 ±2 Vdc
Acceleration limits (any direction)	1500 g	2000
Supply Current	5 mA	20 mA
Supply Voltage	8-16 Vdc	5.4 - 16 Vdc
Operating temp.	-40 to +100 Deg. C	-20 to +70 Deg. C
Sensitivity	1.25 +/- .03 V/g	2 V/g
Frequency Response	0-200 Hz	0-500 Hz
Mounted Resonant Frequency	550 Hz	
Axis	1	1
Weight	10 grams	12
Size	1" X 1" X .5"	25 X 24 X 10 mm
Cost	$ 450.	$ 750.

Table D-3 Links parameters

ITEMS	UNITS	LINK #1	LINK #2
Mass	Kg	5.38	4.05
Position of Centre of Gravity respect to its coordinate	mm	173	98.4
Mass Moment of Inertia respect to centre of gravity	$Kg.m^2$	.138841	.039523
Length	mm	387	251
Distance between the coordinates	mm	220.7	160.3
Material		Aluminum	Aluminum
Natural Frequency	Hz	270	876

APPENDIX E

(Parameters Identification)

E. PARAMETER IDENTIFICATION:

E.1 Square Root Covariance Update Algorithm (SRCUA)

This algorithm is explained in detail in reference [75]. All following equations and definitions are from same reference and the author of this thesis did not modified the algorithm. The concept of this algorithm is one can combine a priori estimate and error covariance $\tilde{x}$ and $\tilde{p}$, respectively, with a scalar observation $z = a^T x + v$ to construct an updated estimate and error covariance $\hat{x}$ and $\hat{p}$. The relevant equations are,

$$K = \frac{\tilde{p}\, a}{\alpha} \qquad \alpha = a^T \tilde{p} a + r$$

$$\hat{x} = \tilde{x} + K(z - a^T \tilde{x})$$

$$\hat{p} = \tilde{p} - K a^T \tilde{p}$$

where:

a	*observation coefficients*
r, v	*measurement noise*
K	*Kalman gain*
α	*residual variance*

It is proved that if a prior estimate and covariance are given by $\tilde{x}$ and $\tilde{p} = \tilde{U}\tilde{D}\tilde{U}^T$, with $\tilde{U}$ unit upper triangular and $\tilde{D}$ diagonal, is to be combined with the scalar observation $z = a^T x + v$, $E(v) = 0, E(v^2) = r$. The Kalman gain K and the updated factors U and D can be obtained from the following algorithm:

Initiarization:

$$\tilde{f}=\tilde{S}^{T}a, \quad \tilde{f}^{T}=(\tilde{f}_1 \ldots \tilde{f}_n)$$

$$\alpha_0 = r, \quad K_2^{T} = (\tilde{S}_{11}\tilde{f}_1 \quad 0 \ldots 0 \text{ for } n-1) \qquad \text{(E-2)}$$

the next set of equations will therefore be calculated in a loop, for j=1, ... n:

$$\alpha_j = \alpha_{j-1} + \tilde{f}_j^{2}$$

$$\beta_j = \left(\frac{\alpha_{j-1}}{\alpha_j}\right)^{1/2}, \quad \gamma_j = \frac{\tilde{f}_j}{(\beta_j \alpha_j)}$$

$$\hat{S}_{jj} = \hat{S}_{jj}\beta_j \qquad \text{(E-3)}$$

$$\hat{S}_{ij} = \tilde{S}_{ij}\beta_j - \gamma_j K_j(i), \quad i=1,\ldots j-1$$

$$K_{j+1}(i) = K_j(i) + \tilde{f}_j\tilde{S}_{ij}, \quad i=1,\ldots j$$

the Kalman gain is therefore:

$$K = \frac{K_{n+1}}{\alpha_n} \qquad \text{(E-4)}$$

Where:

$\tilde{P} = \tilde{S}\tilde{S}^{T}$

$\tilde{P}$ = error coverance
$\tilde{S}$ = priori upper triangular covariance square root
$\hat{S}$ = an updated upper triangular covariance square root
K_j = Kalman coefficent
a = Observation coefficents
α_0 = initial value of residual variance
α = residual variance

This algorithm was coded in the C language as shown below.

The inputs are:

$x1(0)$ = initial estimation from equation (4-19) & page 4-8&4-9

$z = \tau$ (input torques to each joint)

$a = \varphi$ (equation (4-18))

and the output is:

$x1(\infty)$ = final estimation(results on page 4-9)

E.2 Computer Code:

```
for(j=ns1-1;j>-1;j--)
{
    sigma1=0;
    z1=z1-a1[j]*x1[j];
    for(k=0;k<j+1;k++) sigma1=sigma1+s1[k][j]*a1[k];
    a1[j]=sigma1;
}
    alpha1=rr1+a1[0]*a1[0];
    b1[0]=s1[0][0]*a1[0];
    s1[0][0]=s1[0][0]*sqrt(rr1/alpha1);
for(j=1;j<ns1;j++)
{
    b1[j]=s1[j][j]*a1[j];
    gama1=alpha1;
    alpha1=alpha1+a1[j]*a1[j];
    beta1=sqrt(gama1/alpha1);
```

```
     gama1=a1[j]/(alpha1*beta1);
     s1[j][j]=s1[j][j]*beta1;
for(k=0;k<j;k++)
{
     ss1=s1[k][j];
     s1[k][j]=ss1*beta1-gama1*b1[k];
     b1[k]=b1[k]+a1[j]*ss1;
}
}
     z1=z1/alpha1;
     for(j=0;j<ns1;j++) x1[j]=x1[j]+b1[j]*z1;
```

Appendix F

(Architectural Overview Drawings of
the dSPACE Boards)

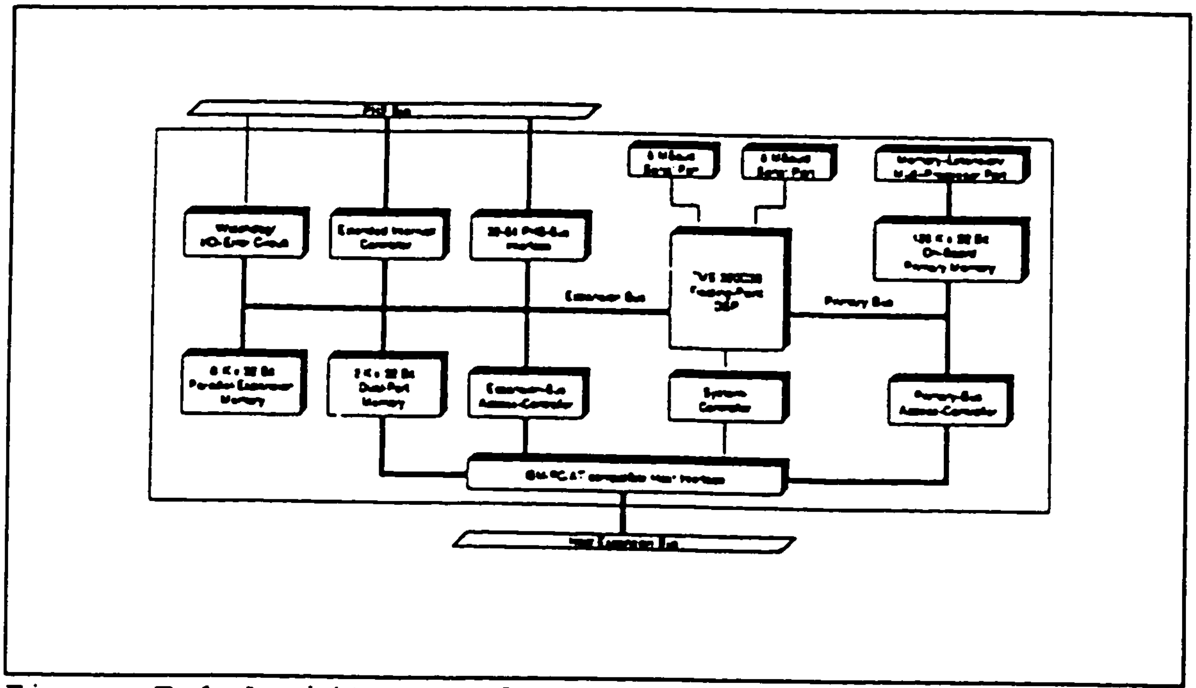

Figure F-1 Architectural overview of DS1002, the DSP board

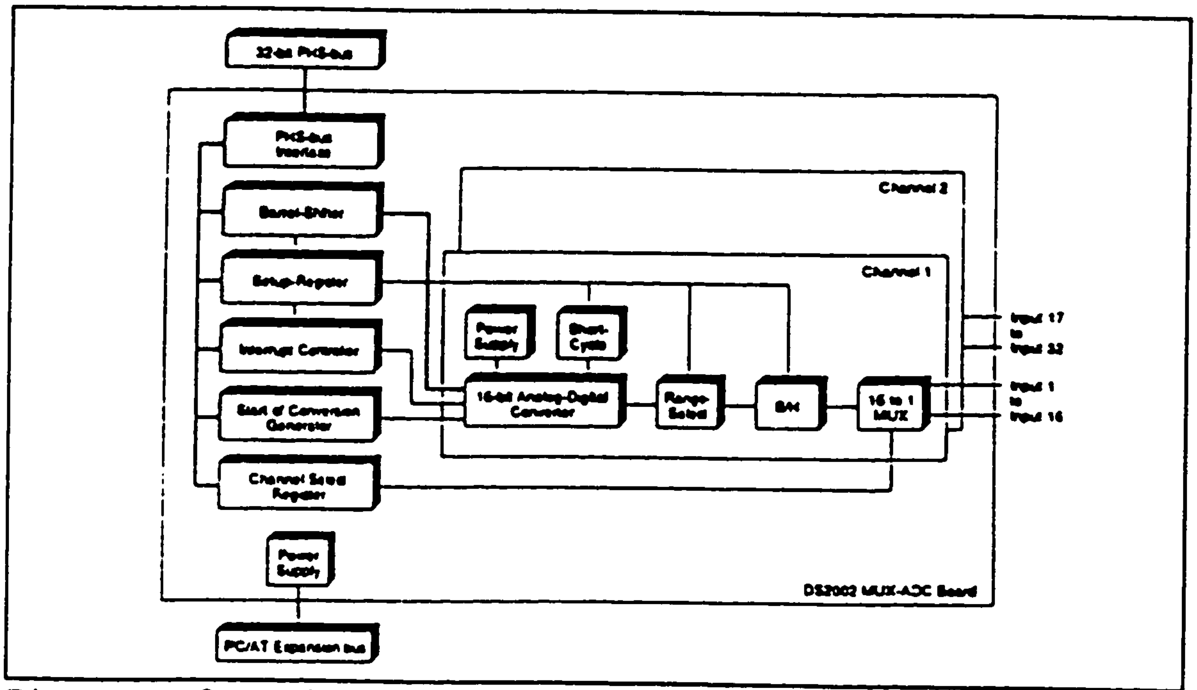

Figure F-2 Architectural overview of DS2002, the A/D converter Board

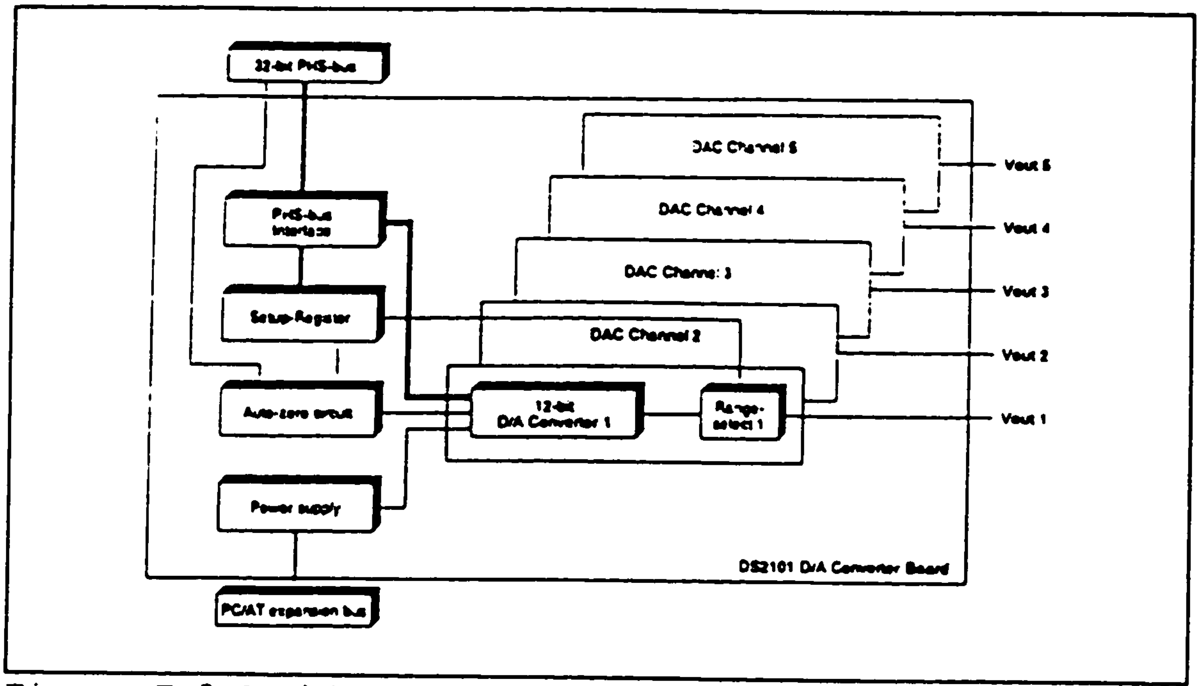

Figure F-3 Architectural overview of DS2101, the D/A converter board

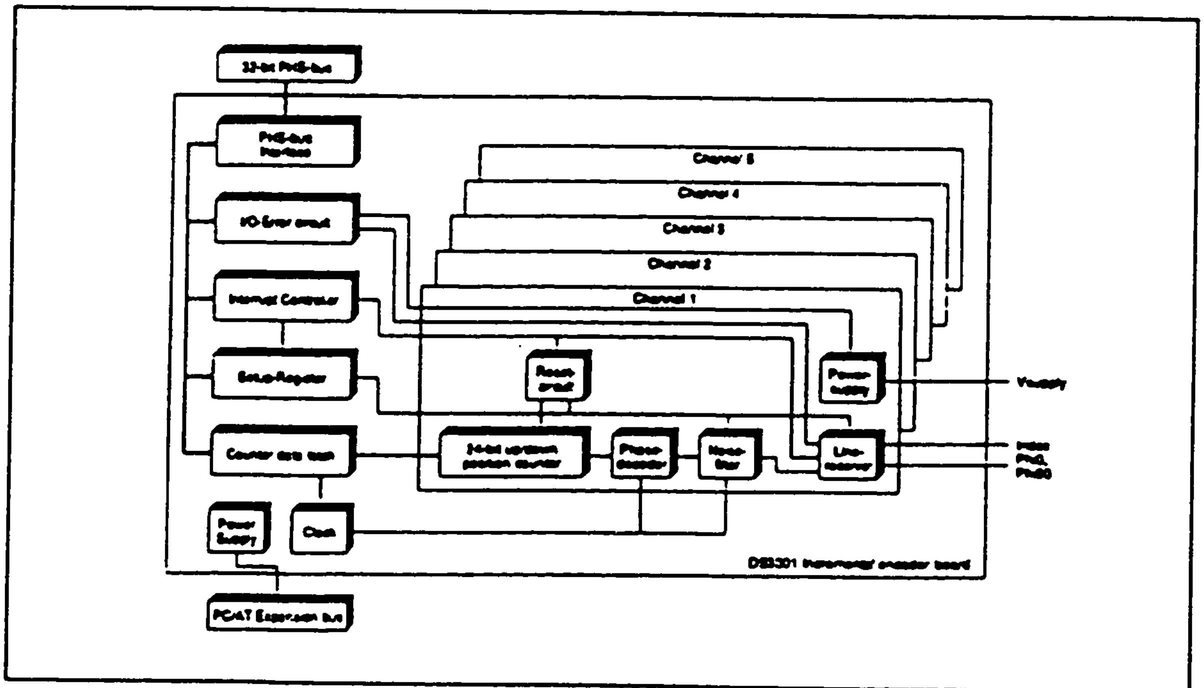

Figure F-4 Architectural overview of DS3001, the 24 bit optical encoder or resolver interface board

Appendix G

(Computer Code)

Note:

Computer source code will be made available, by the author, to those interested in this field of research, upon request.

APPENDIX H

(Time variation of relevant variables in experiments)

The following graphs are just examples of experimental results of few tests, which were described in the thesis. These sample results are provided for section 4.1.3(Parameters Identification) of this thesis.

Experimental Results Regarding the Parameter Identification

The following figures show the parameters identification test results. The two methods (RLSM&SRCUA) were used to identify the robot parameters. These methods identify the parameters of the robot when the robot moves randomly from one configuration to different configuration. These tests were carried out for a given length of time and the parameters were observed and saved for short and long period of time. Only some of the results related to SRCUA method are shown here as example of experimental data. The description of the graphs are explained in the following format.

Figure (H-1) and (H-2) shows the results for SRCUA method using first row of equation (4-15) and equation (4-17), where:

sth11 $=\Phi_1$, *sth21* $=\Phi_2$, *sth31* $=\Phi_3$, *sth41* $=\Phi_4$

Figure (H-3) and (H-4) shows the results for SRCUA method using first row of equation (4-18) and equation (4-19), where:

sth11 $=\Phi_1$, *sth21* $=\Phi_2$, *sth31* $=\Phi_3$, *sth41* $=\Phi_4$, *sth51* $=\Phi_5$, *sth61* $=\Phi_6$

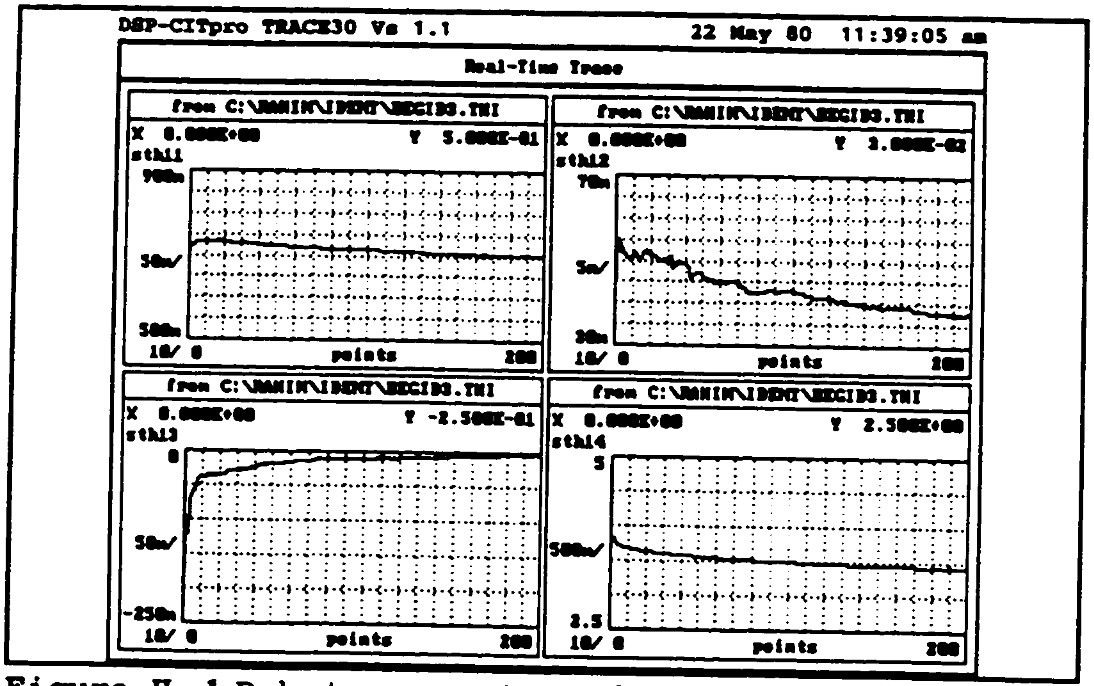

Figure H-1 Robot parameter identification using SR CUA for first few seconds, using equations(4-17)and first row of (4-15).(sth11-> sth14)

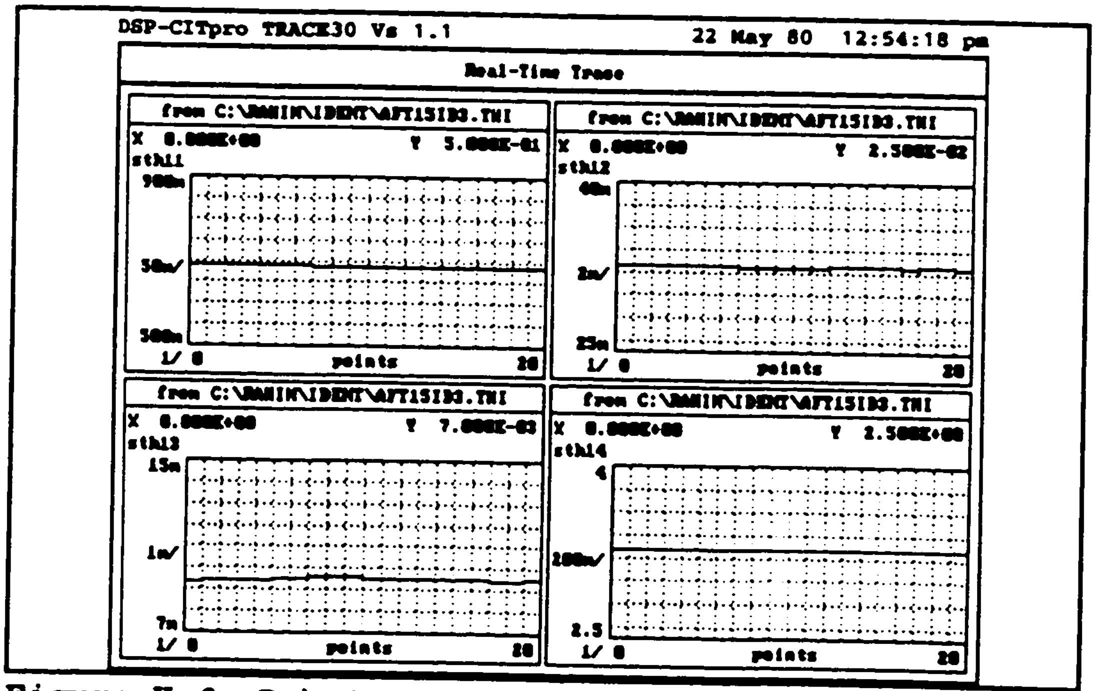

Figure H-2 Robot parameter identification using SRCUA after 15 minutes, using equations(4-17)and first row of (4-15).(sth11-> sth14)

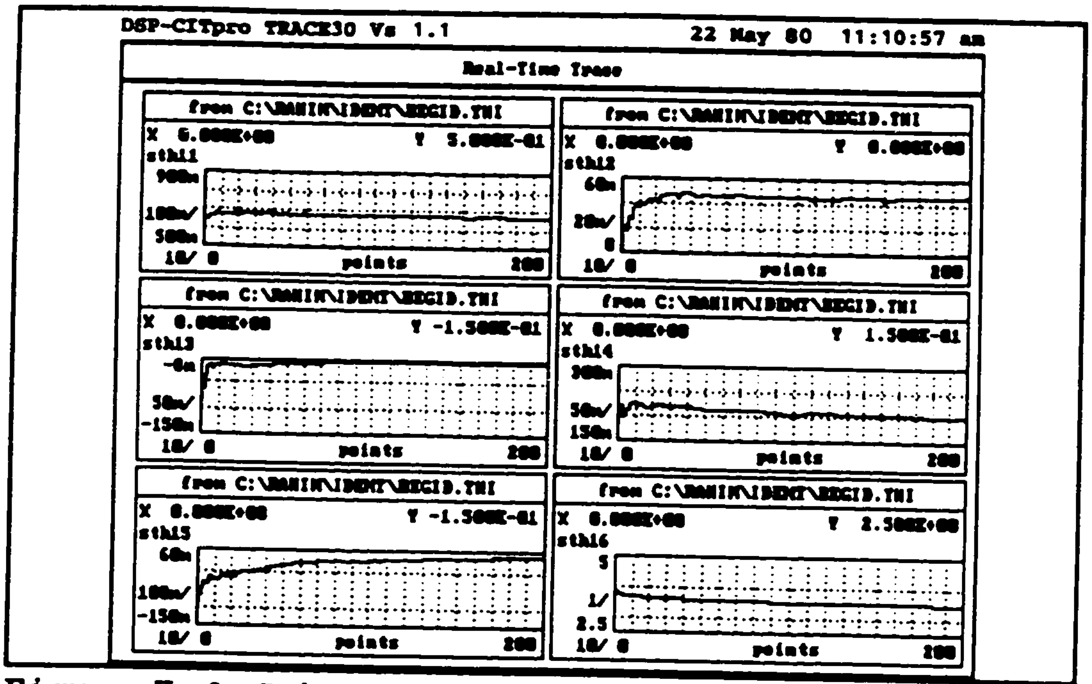

Figure H- 3 Robot parameter identification using SRCUA for first few seconds, using equations(4-19)and first row of (4-18).(sth11->sth16)

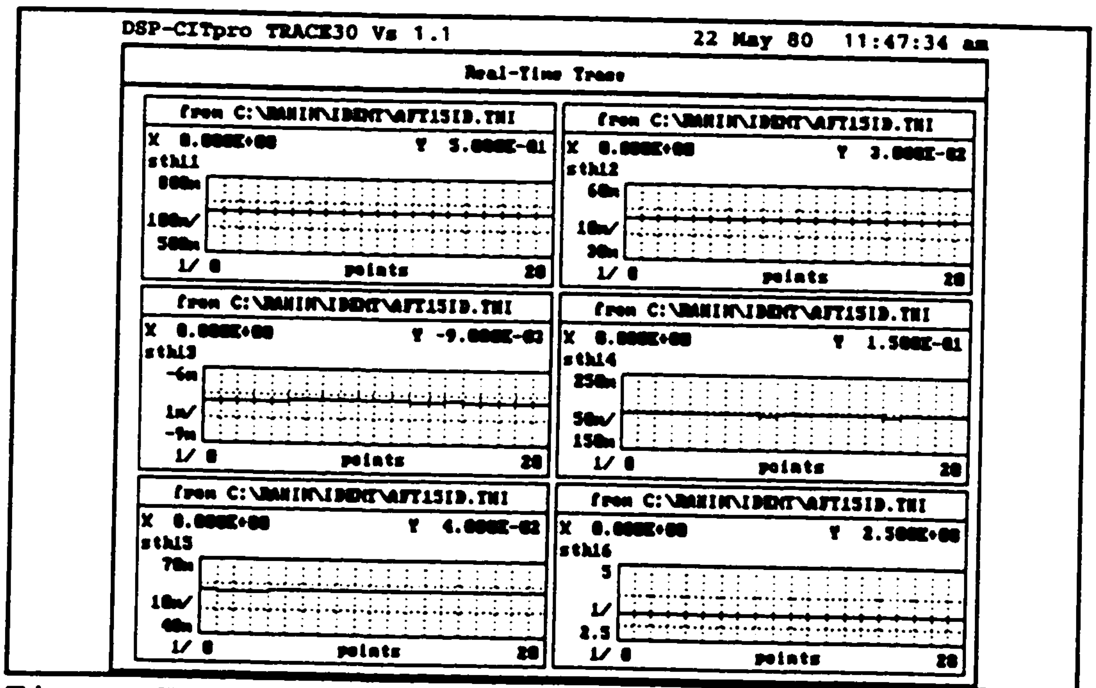

Figure H- 4 Robot parameter identification using SRCUA after 15 minutes, using equations(4-19)and first row of (4-18).(sth11-> sth16)

Appendix I

(Examples of EKF algorithm based on
kinematic and dynamic models)

I.1 Example of two degree of freedom planar robot (Kinematic based)

Sensor Fusion by Extended Kalman Filter for two Degree of Freedom Robot:

The states are defined as:

$$\begin{aligned}
&x_1=\theta_1 && \textit{Position of First Joint}\\
&x_2=\theta_2 && \textit{Position of Second Joint}\\
&x_3=\dot{x}_1=\dot{\theta}_1 && \textit{Angular Velocity of First Joint}\\
&x_4=\dot{x}_2=\dot{\theta}_2 && \textit{Angular Velocity of Second Joint}\\
&x_5=\dot{x}_3=\ddot{\theta}_1 && \textit{Angular Acceleration of First Joint}\\
&x_6=\dot{x}_4=\ddot{\theta}_2 && \textit{Angular Acceleration of Second Joint}
\end{aligned} \tag{I-1}$$

States equations are:

$$\begin{aligned}
&\dot{x}_1=x_3\\
&\dot{x}_2=x_4\\
&\dot{x}_3=x_5\\
&\dot{x}_4=x_6\\
&\dot{x}_5=V_1 && \textit{Jerk of First Joint}\\
&\dot{x}_6=V_2 && \textit{Jerk of Second Joint}
\end{aligned} \tag{I-2}$$

The Output equations are:

Joints positions(resolvers):

$$y_1 = x_1 + w_1$$

$$y_2 = x_2 + w_2$$

End-effectors linear accelerations(accelerometers): **(I-3)**

$$\begin{bmatrix} y_3 \\ \\ y_4 \end{bmatrix} = J(x_1, x_2) \begin{bmatrix} x_5 \\ \\ x_6 \end{bmatrix} + \dot{J}(x_1, x_2, x_3, x_4) \begin{bmatrix} x_3 \\ \\ x_4 \end{bmatrix} + \begin{bmatrix} w_3 \\ \\ w_4 \end{bmatrix}$$

where:

the Jacobian is:

$$J(x_1, x_2) = \begin{bmatrix} j11 & j12 \\ \\ j21 & j22 \end{bmatrix} = \begin{bmatrix} -l_1 S_1 - l_2 S_{12} & -l_2 S_{12} \\ \\ -l_1 C_1 - l_2 C_{12} & -l_2 C_{12} \end{bmatrix} \qquad \text{(I-4)}$$

Derivative of Jacobian is:

$$\dot{J}(x_1, x_2, x_3, x_4) = \begin{bmatrix} jd11 & jd12 \\ \\ jd21 & jd22 \end{bmatrix} = \begin{bmatrix} -l_1 C_1 x_3 - l_2 C_{12}(x_3 + x_4) & -l_2 C_{12}(x_3 + x_4) \\ \\ -l_1 S_1 x_3 - l_2 S_{12}(x_3 + x_4) & -l_2 S_{12}(x_3 + x_4) \end{bmatrix} \qquad \text{(I-5)}$$

Matrix state equations are:

$$\begin{aligned} \boldsymbol{x} &= \boldsymbol{A}\ \boldsymbol{x} + \boldsymbol{B}\ \boldsymbol{v} \\ \boldsymbol{y} &= \boldsymbol{C}\ \boldsymbol{x} + \boldsymbol{w} \end{aligned} \qquad \text{(I-6)}$$

where:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{I-7}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & Jd11 & Jd121 & J11 & J12 \\ 0 & 0 & Jd21 & Jd22 & J21 & J22 \end{bmatrix} \tag{I-8}$$

$$\begin{aligned} x &= [x_1 \ x_2 \ x_3 \ x_3 \ x_4 \ x_5 \ x_6]^T \\ \dot{x} &= [\dot{x}_1 \ \dot{x}_2 \ \dot{x}_3 \ \dot{x}_3 \ \dot{x}_4 \ \dot{x}_5 \ \dot{x}_6]^T \\ y &= [y_1 \ y_2 \ y_3 \ y_4]^T \\ v &= [0 \ 0 \ 0 \ 0 \ v_1 \ v_2]^T \\ w &= [w_1 \ w_2 \ w_3 \ w_4]^T \end{aligned} \tag{I-9}$$

The state equations of the system for a short sampling time(T) are:

$$\begin{aligned} x(n+1) &= \boldsymbol{a} \ x(n) + \boldsymbol{b} \ v(n) \\ y(n) &= \boldsymbol{c} \ x(n) + w(n) \end{aligned} \tag{I-10}$$

where:

$$a = \begin{bmatrix} 1 & 0 & T & 0 & 0 & 0 \\ 0 & 1 & 0 & T & 0 & 0 \\ 0 & 0 & 1 & 0 & T & 0 \\ 0 & 0 & 0 & 1 & 0 & T \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad b = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & T & 0 \\ 0 & 0 & 0 & 0 & 0 & T \end{bmatrix} \tag{I-11}$$

Output equations should be linearized in order to use in Extended Kalman Filter:(from equation (I-3)) for first output(y_1):

$$y_1 = y_1(\hat{x}) + \frac{\partial y_1}{\partial x}\Big|_{x=\hat{x}} (x - \hat{x})$$

$$y_1 = x_1 + w_1 \tag{I-12}$$

for second output(y_2):

$$y_2 = y_2(\hat{x}) + \frac{\partial y_2}{\partial x}\Big|_{x=\hat{x}} (x - \hat{x})$$

$$y_2 = x_2 + w_2 \tag{I-13}$$

for the third output(y_3):

$$y_3 = y_3(\hat{x}) + \frac{\partial y_3}{\partial x}\Big|_{x=\hat{x}} (x - \hat{x})$$

$$y_3 = Jd11(\hat{x})\hat{x}_3 + Jd12(\hat{x})\hat{x}_4 + J11(\hat{x})\hat{x}_5 + J12(\hat{x})\hat{x}_6 + w_3$$

$$+\left[\frac{\partial(Jd11)}{\partial x}\hat{x}_3 + \frac{\partial(Jd12)}{\partial x}\hat{x}_4 + \frac{\partial(J11)}{\partial x}\hat{x}_5 + \frac{\partial(J12)}{\partial x}\hat{x}_6\right]\left[x - \hat{x}\right] \tag{I-14}$$

$$+\left[Jd11\frac{\partial \hat{x}_3}{\partial x} + Jd12\frac{\partial \hat{x}_4}{\partial x} + J11\frac{\partial \hat{x}_5}{\partial x} + J12\frac{\partial \hat{x}_6}{\partial x}\right]\left[x - \hat{x}\right]$$

and for the fourth output(y_4):

$$y_4 = y_4(\hat{x}) + \frac{\partial y_4}{\partial x}\bigg|_{x=\hat{x}} (x-\hat{x})$$

$$
\begin{aligned}
y_4 &= Jd21(\hat{x})\hat{x}_3 + Jd22(\hat{x})\hat{x}_4 + J21(\hat{x})\hat{x}_5 + J22(\hat{x})\hat{x}_6 + w_4 \\
&+\left[\frac{\partial(Jd21)}{\partial x}\hat{x}_3 + \frac{\partial(Jd22)}{\partial x}\hat{x}_4 + \frac{\partial(J21)}{\partial x}\hat{x}_5 + \frac{\partial(J22)}{\partial x}\hat{x}_6\right]\left[x-\hat{x}\right] \\
&+\left[Jd21\frac{\partial\hat{x}_3}{\partial x} + Jd22\frac{\partial\hat{x}_4}{\partial x} + J21\frac{\partial\hat{x}_5}{\partial x} + J22\frac{\partial\hat{x}_6}{\partial x}\right]\left[x-\hat{x}\right]
\end{aligned}
\tag{I-15}
$$

where, partial derivatives of J and $\dot{J}$ are:

$$
\frac{\partial Jd11(\hat{x})}{\partial x} = \begin{cases}
\frac{\partial Jd11(\hat{x})}{\partial x_1} = l_1 s_1 \hat{x}_3 + l_2 s_{12}(\hat{x}_3 + \hat{x}_4) \\
\frac{\partial Jd11(\hat{x})}{\partial x_2} = l_2 s_{12}(\hat{x}_3 + \hat{x}_4) \\
\frac{\partial Jd11(\hat{x})}{\partial x_3} = -l_1 c_1 - l_2 c_{12} \\
\frac{\partial Jd11(\hat{x})}{\partial x_4} = -l_2 c_{12} \\
\frac{\partial Jd11(\hat{x})}{\partial x_5} = 0 \\
\frac{\partial Jd11(\hat{x})}{\partial x_6} = 0
\end{cases}
\qquad
\frac{\partial Jd12(\hat{x})}{\partial x} = \begin{cases}
\frac{\partial Jd12(\hat{x})}{\partial x_1} = l_2 s_{12}(\hat{x}_3 + \hat{x}_4) \\
\frac{\partial Jd12(\hat{x})}{\partial x_2} = l_2 s_{12}(\hat{x}_3 + \hat{x}_4) \\
\frac{\partial Jd12(\hat{x})}{\partial x_3} = -l_2 c_{12} \\
\frac{\partial Jd12(\hat{x})}{\partial x_4} = -l_2 c_{12} \\
\frac{\partial Jd12(\hat{x})}{\partial x_5} = 0 \\
\frac{\partial Jd12(\hat{x})}{\partial x_6} = 0
\end{cases}
\tag{I-16}
$$

$$\frac{\partial Jd21(\dot{x})}{\partial x}=\begin{cases}\dfrac{\partial Jd21(\dot{x})}{\partial x_1} = l_1c_1\dot{x}_3+l_2c_{12}(\dot{x}_3+\dot{x}_4)\\ \dfrac{\partial Jd21(\dot{x})}{\partial x_2} = l_2c_{12}(\dot{x}_3+\dot{x}_4)\\ \dfrac{\partial Jd21(\dot{x})}{\partial x_3} = -l_1s_1-l_2s_{12}\\ \dfrac{\partial Jd21(\dot{x})}{\partial x_4} = -\ l_2s_{12}\\ \dfrac{\partial Jd21(\dot{x})}{\partial x_5} =0\\ \dfrac{\partial Jd21(\dot{x})}{\partial x_6} =0\end{cases}\qquad \frac{\partial Jd22(\dot{x})}{\partial x}=\begin{cases}\dfrac{\partial Jd22(\dot{x})}{\partial x_1} = -l_2c_{12}(\dot{x}_3+\dot{x}_4)\\ \dfrac{\partial Jd22(\dot{x})}{\partial x_2} = -l_2c_{12}(\dot{x}_3+\dot{x}_4)\\ \dfrac{\partial Jd22(\dot{x})}{\partial x_3} = -\ l_2s_{12}\\ \dfrac{\partial Jd22(\dot{x})}{\partial x_4} = -\ l_2s_{12}\\ \dfrac{\partial Jd22(\dot{x})}{\partial x_5} =0\\ \dfrac{\partial Jd22(\dot{x})}{\partial x_6} =0\end{cases}\tag{I-17}$$

$$\frac{\partial J11(\dot{x})}{\partial x}=\begin{cases}\dfrac{\partial J11(\dot{x})}{\partial x_1} = -l_1c_1\ -\ l_2c_{12}\\ \dfrac{\partial J11(\dot{x})}{\partial x_2} = -l_2c_{12}\\ \dfrac{\partial J11(\dot{x})}{\partial x_3} = 0\\ \dfrac{\partial J11(\dot{x})}{\partial x_4} = 0\\ \dfrac{\partial J11(\dot{x})}{\partial x_5} =0\\ \dfrac{\partial J11(\dot{x})}{\partial x_6} =0\end{cases}\qquad \frac{\partial J12(\dot{x})}{\partial x}=\begin{cases}\dfrac{\partial J12(\dot{x})}{\partial x_1} = -\ l_2\ c_{12}\\ \dfrac{\partial J12(\dot{x})}{\partial x_2} = -l_2\ c_{12}\\ \dfrac{\partial J12(\dot{x})}{\partial x_3} = 0\\ \dfrac{\partial J12(\dot{x})}{\partial x_4} = 0\\ \dfrac{\partial J12(\dot{x})}{\partial x_5} =0\\ \dfrac{\partial J12(\dot{x})}{\partial x_6} =0\end{cases}\tag{I-18}$$

$$\frac{\partial J21(\hat{x})}{\partial x}=\begin{cases}\dfrac{\partial J21(\hat{x})}{\partial x_1} = -l_1 s_1 - l_2 s_{12}\\ \dfrac{\partial J21(\hat{x})}{\partial x_2} = -l_2 s_{12}\\ \dfrac{\partial J21(\hat{x})}{\partial x_3} = 0\\ \dfrac{\partial J21(\hat{x})}{\partial x_4} = 0\\ \dfrac{\partial J21(\hat{x})}{\partial x_5} =0\\ \dfrac{\partial J21(\hat{x})}{\partial x_6} =0\end{cases} \qquad \frac{\partial J22(\hat{x})}{\partial x}=\begin{cases}\dfrac{\partial J22(\hat{x})}{\partial x_1} = - l_2 s_{12}\\ \dfrac{\partial J22(\hat{x})}{\partial x_2} = -l_2 s_{12}\\ \dfrac{\partial J22(\hat{x})}{\partial x_3} = 0\\ \dfrac{\partial J22(\hat{x})}{\partial x_4} = 0\\ \dfrac{\partial J22(\hat{x})}{\partial x_5} =0\\ \dfrac{\partial J22(\hat{x})}{\partial x_6} =0\end{cases} \tag{I-19}$$

Therefore, y_3 can be calculated as following by using equation (I-14);

$$y_3 = Jd11\ \hat{x}_3 + Jd12\ \hat{x}_4 + J11\ \hat{x}_5 + J12\ \hat{x}_6 + \omega_3 + \begin{bmatrix} dy31 & dy32 & dy33 & dy34 & dy35 & dy36 \end{bmatrix} \begin{bmatrix} x_1 - \hat{x}_1 \\ x_2 - \hat{x}_2 \\ \cdot \\ \cdot \\ x_6 - \hat{x}_6 \end{bmatrix} \tag{I-20}$$

where:

$$\begin{aligned} dy31 &= \hat{x}_3 \frac{\partial Jd11}{\partial x_1} + \hat{x}_4 \frac{\partial Jd12}{\partial x_1} + \hat{x}_5 \frac{\partial J11}{\partial x_1} + \hat{x}_6 \frac{\partial J12}{\partial x_1} \\ &= \hat{x}_3 (l_1 s_1 \hat{x}_3 + l_2 s_{12} (\hat{x}_3 + \hat{x}_4)) + \hat{x}_4 (l_2 s_{12} (\hat{x}_3 + \hat{x}_4)) \\ &\quad + \hat{x}_5 (-l_1 c_1 - l_2 c_{12}) + \hat{x}_6 (-l_2 c_{12}) \end{aligned} \tag{I-21}$$

$$dy32 = \hat{x}_3 \frac{\partial Jd11}{\partial x_2} + \hat{x}_4 \frac{\partial Jd12}{\partial x_2} + \hat{x}_5 \frac{\partial J11}{\partial x_2} + \hat{x}_6 \frac{\partial J12}{\partial x_2}$$

$$= \hat{x}_3 (l_2 s_{12} (\hat{x}_3 + \hat{x}_4)) + \hat{x}_4 (l_2 s_{12} (\hat{x}_3 + \hat{x}_4)) \quad \text{(I-22)}$$

$$+ \hat{x}_5 (-l_2 c_{12}) + \hat{x}_6 (-l_2 c_{12})$$

$$dy33 = \hat{x}_3 \frac{\partial Jd11}{\partial x_3} + \hat{x}_4 \frac{\partial Jd12}{\partial x_3} + \hat{x}_5 \frac{\partial J11}{\partial x_3} + \hat{x}_6 \frac{\partial J12}{\partial x_3} + Jd11 \quad \text{(I-23)}$$

$$= \hat{x}_3 (-l_1 c_1 - l_2 c_{12}) + \hat{x}_4 (-l_2 c_{12}) + Jd11$$

$$dy34 = \hat{x}_3 \frac{\partial Jd11}{\partial x_4} + \hat{x}_4 \frac{\partial Jd12}{\partial x_4} + \hat{x}_5 \frac{\partial J11}{\partial x_4} + \hat{x}_6 \frac{\partial J12}{\partial x_4} + Jd12 \quad \text{(I-24)}$$

$$= \hat{x}_3 (-l_2 c_{12}) + \hat{x}_4 (-l_2 c_{12}) + Jd12$$

$$dy35 = \hat{x}_3 \frac{\partial Jd11}{\partial x_5} + \hat{x}_4 \frac{\partial Jd12}{\partial x_5} + \hat{x}_5 \frac{\partial J11}{\partial x_5} + \hat{x}_6 \frac{\partial J12}{\partial x_5} + J11 \quad \text{(I-25)}$$

$$= J11$$

$$dy36 = \hat{x}_3 \frac{\partial Jd11}{\partial x_6} + \hat{x}_4 \frac{\partial Jd12}{\partial x_6} + \hat{x}_5 \frac{\partial J11}{\partial x_6} + \hat{x}_6 \frac{\partial J12}{\partial x_6} + J12 \quad \text{(I-26)}$$

$$= J12$$

y_4 can be calculated by using equation(I-14) as follows;

$$y_4 = Jd21\ \hat{x}_3 + Jd22\ \hat{x}_4 + J21\ \hat{x}_5 + J22\ \hat{x}_6 + \omega_4 + \begin{bmatrix} dy41 & dy42 & dy43 & dy44 & dy45 & dy46 \end{bmatrix} \begin{bmatrix} x_1 - \hat{x}_1 \\ x_2 - \hat{x}_2 \\ \cdot \\ \cdot \\ x_6 - \hat{x}_6 \end{bmatrix} \tag{I-27}$$

$$\begin{aligned} dy41 &= \hat{x}_3 \frac{\partial Jd21}{\partial x_1} + \hat{x}_4 \frac{\partial Jd22}{\partial x_1} + \hat{x}_5 \frac{\partial J21}{\partial x_1} + \hat{x}_6 \frac{\partial J22}{\partial x_1} \\ &= \hat{x}_3(-l_1 c_1 \hat{x}_3 - l_2 c_{12}(\hat{x}_3 + \hat{x}_4)) + \hat{x}_4(-l_2 c_{12}(\hat{x}_3 + \hat{x}_4)) \\ &\quad + \hat{x}_5(-l_1 s_1 - l_2 s_{12}) + \hat{x}_6(-l_2 s_{12}) \end{aligned} \tag{I-28}$$

$$\begin{aligned} dy42 &= \hat{x}_3 \frac{\partial Jd21}{\partial x_2} + \hat{x}_4 \frac{\partial Jd22}{\partial x_2} + \hat{x}_5 \frac{\partial J21}{\partial x_2} + \hat{x}_6 \frac{\partial J22}{\partial x_2} \\ &= \hat{x}_3(-l_2 c_{12}(\hat{x}_3 + \hat{x}_4)) + \hat{x}_4(-l_2 c_{12}(\hat{x}_3 + \hat{x}_4)) \\ &\quad + \hat{x}_5(-l_2 s_{12}) + \hat{x}_6(-l_2 s_{12}) \end{aligned} \tag{I-29}$$

where:

$$\begin{aligned} dy43 &= \hat{x}_3 \frac{\partial Jd21}{\partial x_3} + \hat{x}_4 \frac{\partial Jd22}{\partial x_3} + \hat{x}_5 \frac{\partial J21}{\partial x_3} + \hat{x}_6 \frac{\partial J22}{\partial x_3} + Jd21 \\ &= \hat{x}_3(-l_1 s_1 - l_2 s_{12}) + \hat{x}_4(-l_2 s_{12}) + Jd21 \end{aligned} \tag{I-30}$$

$$\begin{aligned} dy44 &= \hat{x}_3 \frac{\partial Jd21}{\partial x_4} + \hat{x}_4 \frac{\partial Jd22}{\partial x_4} + \hat{x}_5 \frac{\partial J21}{\partial x_4} + \hat{x}_6 \frac{\partial J22}{\partial x_4} + Jd22 \\ &= \hat{x}_3(-l_2 s_{12}) + \hat{x}_4(-l_2 s_{12}) + Jd22 \end{aligned} \tag{I-31}$$

$$dy45 = \hat{x}_3 \frac{\partial Jd21}{\partial x_5} + \hat{x}_4 \frac{\partial Jd22}{\partial x_5} + \hat{x}_5 \frac{\partial J21}{\partial x_5} + \hat{x}_6 \frac{\partial J22}{\partial x_5} + J21 \quad = \quad J21 \tag{I-32}$$

$$dy46 = \hat{x}_3 \frac{\partial Jd21}{\partial x_6} + \hat{x}_4 \frac{\partial Jd22}{\partial x_6} + \hat{x}_5 \frac{\partial J21}{\partial x_6} + \hat{x}_6 \frac{\partial J22}{\partial x_6} + J22 \quad = \quad J22 \tag{I-33}$$

$$\boldsymbol{y} = \boldsymbol{c}\ \boldsymbol{x} + \boldsymbol{w} + \boldsymbol{E}\ \hat{\boldsymbol{x}} \tag{I-34}$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ dy31 & dy32 & dy33 & dy34 & dy35 & dy36 \\ dy41 & dy42 & dy43 & dy44 & dy45 & dy46 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix}$$

$$+ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -dy31 & -dy32 & Jd11-dy33 & Jd12-dy34 & J11-dy35 & J12-dy36 \\ -dy41 & -dy42 & Jd21-dy43 & Jd22-dy44 & J21-dy45 & J22-dy46 \end{bmatrix} \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \hat{x}_3 \\ \hat{x}_4 \\ \hat{x}_5 \\ \hat{x}_6 \end{bmatrix} \tag{I-35}$$

$$+ \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix}$$

The equation for the sampled system is:

From equations (I-34) and (I-35), the following matrices are needed for using in Kalman filter algorithm:

$$c = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ dy31 & dy32 & dy33 & dy34 & dy35 & dy36 \\ dy41 & dy42 & dy43 & dy44 & dy45 & dy46 \end{bmatrix} \tag{I-36}$$

$$E = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -dy31 & -dy32 & Jd11-dy33 & Jd12-dy34 & J11-dy35 & J12-dy36 \\ -dy41 & -dy42 & Jd21-dy43 & Jd22-dy44 & J21-dy45 & J22-dy46 \end{bmatrix} \tag{I-37}$$

$$a = \begin{bmatrix} 1 & 0 & T & 0 & 0 & 0 \\ 0 & 1 & 0 & T & 0 & 0 \\ 0 & 0 & 1 & 0 & T & 0 \\ 0 & 0 & 0 & 1 & 0 & T \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad b = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & T & 0 \\ 0 & 0 & 0 & T \end{bmatrix} \tag{I-38}$$

$$\omega = \begin{bmatrix} \omega_1 & 0 & 0 & 0 \\ 0 & \omega_2 & 0 & 0 \\ 0 & 0 & \omega_3 & 0 \\ 0 & 0 & 0 & \omega_4 \end{bmatrix}$$

I.1.1 Extended Kalman Filter(EKF) Algorithm for Kinematic based Example

The Extended Kalman Filter is given by[81]:

- Kalman Gain matrix:

$$K(n) = P(n)^{-} C^{T}(n) \, [C(n) \, P(n)^{-} C^{T}(n) + w_d]^{-1} \tag{I-39}$$

- State estimate update:

$$\hat{x}(n)^{+} = \hat{x}(n)^{-} + K(n) \left(y(n) - C(n) \, x(n)^{-} - e(n) \right) \tag{I-40}$$

- State estimate extrapolation:

$$\hat{x}(n+1)^{-} = a(n) \, \hat{x}(n)^{+} + b(n) \, u \tag{I-41}$$

- Error covariance update:

$$P(n)^{+} = \left(I - K(n) \, C(n) \right) P(n)^{-} \tag{I-42}$$

- Error covariance extrapolation:

$$P(n+1)^{-} = a(n) \, P(n)^{+} \, a(n)^{T} + V_d \tag{I-43}$$

The noises w_d and V_d are assumed Gaussian white noises, time invariant with zero mean. In these equations, the superscript $^{-}$ refers to a time just before, while the superscript $^{+}$ refers to a time just after a sampling instant. Thus, $P(n)^{-}$ is the calculated error covariance just before the nth sample.

I.2 Example for a two degree of freedom robot: (Dynamic based)

Dynamic equation:

$$\tau = M\ddot{\theta} + N + f_c$$

$$\tau = \begin{bmatrix} k_1 + k_2 + 2c_2k_3 & k_2 + k_3c_2 \\ n_1 + n_2c_2 & n_1 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} \tag{I-44}$$

$$+ \begin{bmatrix} -2s_2k_4\dot{\theta}_1\dot{\theta}_2 - k_5s_2\dot{\theta}_2^2 \\ n_3s_2\dot{\theta}_1^2 \end{bmatrix} + \begin{bmatrix} k_6\ sgn(\dot{\theta}_1) \\ n_4\ sgn(\dot{\theta}_2) \end{bmatrix}$$

therefore:

$$M = \begin{bmatrix} M11 & M12 \\ M21 & M22 \end{bmatrix} \quad N = \begin{bmatrix} N_1 \\ N_2 \end{bmatrix} \quad f_c = \begin{bmatrix} f_{c1} \\ f_{c2} \end{bmatrix} \tag{I-45}$$

where(from equation (4-19) and the results from page (4-9):

$$\begin{cases} k_1 = .6613 \\ k_2 = .0466 \\ k_3 = .0073 \\ k_4 = .1973 \\ k_5 = .0578 \\ k_6 = 3.37 \end{cases} \qquad \begin{cases} n_1 = ..054 \\ n_2 = .0934 \\ n_3 = .0614 \\ n_4 = .1585 \end{cases}$$

and inverse of inertia matrix is:

$$M^{-1} = \begin{bmatrix} \frac{m22}{md} & -\frac{m12}{md} \\ -\frac{m12}{md} & \frac{m11}{md} \end{bmatrix} \tag{I-46}$$

where:

md = determinate of the inertia matrix(M)

For further calculations in Kalman filter, there are few terms which can now be calculated, as follows:

derivative of (md) :

$$dmd = \frac{d(md)}{dt} = s_2\dot{\theta}_2\ [n_2k_2 - n_1k_3 + 2n_2k_3c_2] \qquad \textbf{(I-47)}$$

derivative of inertial inverse matrix:

$$\dot{M}^{-1} = \frac{d(M^{-1})}{dt}$$

$$dm11i = \frac{d(m11i)}{dt} = -\frac{dmd\ (m22)}{(md)^2}$$

$$dm12i = \frac{d(m12i)}{dt} = \frac{k_3\dot{\theta}_2s_2(md) + dmd\ (m12)}{(md)^2} \qquad \textbf{(I-48)}$$

$$dm21i = \frac{d(m21i)}{dt} = \frac{n_2\dot{\theta}_2s_2(md) + dmd\ (m21)}{(md)^2}$$

$$dm22i = \frac{d(m22i)}{dt} = \frac{-2\ k_3\dot{\theta}_2s_2(md) - dmd\ (m11)}{(md)^2}$$

Take partial derivative of md respect to each states(x) $\left(\frac{\partial(md)}{\partial\bar{x}}\right)$:

$$
\begin{aligned}
mdx1 &= \frac{\partial(md)}{\partial x_1} = 0 \\
mdx2 &= \frac{\partial(md)}{\partial x_2} = -S_2\ (k_3 n_1 - n_2 k_2 - 2k_3 n_2 c_2) \\
mdx3 &= \frac{\partial(md)}{\partial x_3} = 0 \\
mdx4 &= \frac{\partial(md)}{\partial x_4} = 0 \\
mdx5 &= \frac{\partial(md)}{\partial x_5} = 0 \\
mdx6 &= \frac{\partial(md)}{\partial x_6} = 0
\end{aligned}
\tag{I-49}
$$

Take partial derivative of dmd respect to each states(x) $\left(\frac{\partial(dmd)}{\partial\bar{x}}\right)$:

$$
\begin{aligned}
dmdx1 &= \frac{\partial(dmd)}{\partial x_1} = 0 \\
dmdx2 &= \frac{\partial(dmd)}{\partial x_2} = -c_2\ \dot{\theta}_2\left(k_2 n_2 - n_1 k_3 + 2k_3 n_2 c_2\right) - 2n_2 k_3 s_2^2 \dot{\theta}_2 \\
dmdx3 &= \frac{\partial(dmd)}{\partial x_3} = 0 \\
dmdx4 &= \frac{\partial(dmd)}{\partial x_4} = s_2\left(n_2 k_2 - n_1 k_3 + 2n_2 k_3 c_2\right) \\
dmdx5 &= \frac{\partial(dmd)}{\partial x_5} = 0 \\
dmdx6 &= \frac{\partial(dmd)}{\partial x_6} = 0
\end{aligned}
\tag{I-50}
$$

Partial derivative inverse of inertial matrix respect to each states $\left(\frac{\partial m^{-1}}{\partial x}\right)$:

$$\begin{aligned}
m11ix2 &= -\frac{n_1(dmdx2)}{(md)^2} \\
m11ix4 &= 0 \\
m12ix2 &= \frac{s_2 k_3(md) + dmdx2\ (k_2 + c_2 k_3)}{(md)^2} \\
m12ix4 &= 0 \\
m21ix2 &= \frac{s_2 n_2(md) + dmdx2\ (n_1 + c_2 n_2)}{(md)^2} \\
m21ix4 &= 0 \\
m22ix2 &= \frac{-2\ s_2 k_3(md) - dmdx2\ (2\ k_1 + 2\ c_2 k_3)}{(md)^2} \\
m22ix4 &= 0
\end{aligned} \tag{I-51}$$

Partial derivative of derivative of inverse of inertial matrix with respect to each states $\left(\frac{\partial(\dot{m}^{-1})}{\partial x}\right)$:

$$dm11ix2 = -\frac{dmdx2\ m22\ (md)^2 - 2\ md\ mdx2\ dmd\ m22}{(md)^4}$$

$$dm11ix4 = -\frac{dmdx4\ m22}{(md)^2}$$

$$dm12ix2 = \frac{\left(k_3\dot{\theta}_2 c_2(md) + mdx2\ k_3\dot{\theta}_2 s_2 + dmdx2(m12) - k_3 s_2(dmd)\right)(md)^2}{(md)^4}$$

$$\frac{-\ 2\ md\ mdx2\ \left(k_3\dot{\theta}_2 s_2(md) + dmd(m12)\right)}{}$$

$$dm12ix4 = \frac{k_3 s_2(md) + dmdx4(m12)}{(md)^2}$$

$$dm21ix2 = \frac{\left(n_2\dot{\theta}_2 c_2(md) + mdx2\ n_2\dot{\theta}_2 s_2 + dmdx2(m21) - n_2 s_2(dmd)\right)(md)^2}{(md)^4} \qquad \textbf{(I-52)}$$

$$\frac{-\ 2\ md\ mdx2\ \left(n_2\dot{\theta}_2 s_2(md) + dmd(m21)\right)}{}$$

$$dm12ix4 = \frac{n_2 s_2(md) + dmdx4(m21)}{(md)^2}$$

$$dm22ix2 = \frac{\left(-2k_3\dot{\theta}_2 c_2(md) - 2mdx2\ k_3\dot{\theta}_2 s_2 - dmdx2\ m11 + 2k_3 s_2(dmd)\right)(md)^2}{(md)^4}$$

$$\frac{-\ 2\ md\ mdx2\ \left(-2\ k_3\dot{\theta}_2 s_2(md) - dmd(m11)\right)}{}$$

$$dm12ix4 = \frac{-2\ k_3 s_2(md) - dmdx4(m11)}{(md)^2}$$

From equation(5-118), the $x(n+1)$ can be calculated:

$$x(n+1) = \hat{a}\hat{x} + \hat{b}u + \hat{e}\hat{N} + \hat{f}\hat{V} + \hat{P}(n)\ (x-\hat{x})$$

where:

$$p(n) = \frac{\partial(\hat{a}\hat{x} + \hat{b}u + \hat{e}\hat{N})}{\partial\hat{x}} \qquad \textbf{(I-53)}$$

$$= a + \frac{\partial\hat{b}}{\partial\hat{x}}u + \frac{\partial e}{\partial\hat{x}}\hat{N} + \frac{\partial\hat{N}}{\partial\hat{x}}\hat{e}$$

All the terms of $\mathbf{p}$(n) equation can be calculated as following:

$$a\ \frac{\partial\hat{x}}{\partial\hat{x}} =$$

$$\begin{bmatrix} 1 & 0 & T & 0 & 0 & 0 \\ 0 & 1 & 0 & T & 0 & 0 \\ 0 & 0 & 1 & 0 & T & 0 \\ 0 & 0 & 0 & 1 & 0 & T \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} x_1 + Tx_3 \\ x_2 + Tx_4 \\ x_3 + Tx_5 \\ x_4 + Tx_6 \\ x_5 \\ x_6 \end{bmatrix} \qquad \textbf{(I-54)}$$

$\hat{b}\, u =$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & dm11i & dm12i & m11i & m112i \\ 0 & 0 & dm21i & dm22i & m21i & m22i \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \tag{I-55}$$

$$= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ u_1\ (dm11i) + u_2\ (dm12i) + u_3\ (m11i) + u_4\ (m12i) \\ u_1\ (dm21i) + u_2\ (dm22i) + u_3\ (m21i) + u_4\ (m22i) \end{bmatrix}$$

Taking partial derivative of equation(I-55) with respect to each states:

$$\frac{\partial(\hat{b}u)}{\partial x} = \begin{bmatrix} dbu11 & dbu12 & dbu13 & dby14 & dbu15 & dbu16 \\ dbu21 & dbu22 & dbu23 & dby24 & dbu25 & dbu26 \\ dbu31 & dbu32 & dbu33 & dby34 & dbu35 & dbu36 \\ dbu41 & dbu42 & dbu43 & dby44 & dbu45 & dbu46 \\ dbu51 & dbu52 & dbu53 & dby54 & dbu55 & dbu56 \\ dbu61 & dbu62 & dbu63 & dby64 & dbu65 & dbu66 \end{bmatrix} \tag{I-56}$$

where:

$$dbu11 \rightarrow dbu16 = 0$$
$$dbu21 \rightarrow dbu26 = 0$$
$$dbu31 \rightarrow dbu36 = 0$$
$$dbu41 \rightarrow dbu46 = 0$$

$$\begin{cases} dbu51 = 0 \\ dbu52 = u_1 \ (dm11ix2) + u_2 \ (dm12ix2) + u_3 \ (m11ix2) + u_4 \ (m12ix2) \\ dbu53 = 0 \\ dbu54 = u_1 \ (dm11ix4) + u_2 \ (dm12ix4) + u_3 \ (m11ix4) + u_4 \ (m11ix4) \\ dbu55 = 0 \\ dbu56 = 0 \end{cases} \tag{I-57}$$

$$\begin{cases} dbu61 = 0 \\ dbu62 = u_1 \ (dm21ix2) + u_2 \ (dm22ix2) + u_3 \ (m21ix2) + u_4 \ (m22ix2) \\ dbu63 = 0 \\ dbu64 = u_1 \ (dm21ix4) + u_2 \ (dm22ix4) + u_3 \ (m21ix4) + u_4 \ (m22ix4) \\ dbu55 = 0 \\ dbu56 = 0 \end{cases}$$

Taking partial derivative of third term of equation (I-53) with respect to each states, gives:

$$\hat{e}\,\hat{n} = -\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & dm11i & dm12i & m11i & m12i \\ 0 & 0 & dm21i & dm22i & m21i & m22i \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ n_1 \\ n_2 \\ \dot{n}_1 \\ \dot{n}_2 \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ n_1\ (dm11i) + n_2\ (dm12i) + \dot{n}_1\ (m11i) + \dot{n}_2\ (m12i) \\ n_1\ (dm21i) + n_2\ (dm22i) + \dot{n}_1\ (m21i) + \dot{n}_2\ (m22i) \end{bmatrix} \tag{I-58}$$

Before taking partial derivatives of term($\hat{e}\hat{n}$), for simplicity partial derivative of term ($\hat{n}$) is taken first, as follows:

$$\hat{n} = \begin{bmatrix} 0 \\ 0 \\ n_1 \\ n_2 \\ \dot{n}_1 \\ \dot{n}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -2s_2k_4\dot{\theta}_1\dot{\theta}_2 - k_5s_2\dot{\theta}_2^{\,2} \\ n_3s_2\dot{\theta}_1^{\,2} \\ -2k_4(c_2\dot{\theta}_1\dot{\theta}_2^{\,2} + s_2\ddot{\theta}_1\dot{\theta}_2 + s_2\dot{\theta}_1\ddot{\theta}_2) - k_5(c_2\dot{\theta}_2^{\,3} + 2s_2\dot{\theta}_2\ddot{\theta}_2) \\ n_3(c_2\dot{\theta}_1^{\,2}\dot{\theta}_2 + 2s_2\dot{\theta}_1\ddot{\theta}_1) \end{bmatrix} \tag{I-59}$$

Taking partial derivative of all $\hat{n}$ vector terms with respect to states:

$$\frac{\partial \hat{\boldsymbol{n}}}{\partial \boldsymbol{x}} = \begin{bmatrix} 0 \\ 0 \\ \dfrac{\partial n_1}{\partial \boldsymbol{x}} \\ \dfrac{\partial n_2}{\partial \boldsymbol{x}} \\ \dfrac{\partial \dot{n}_1}{\partial \boldsymbol{x}} \\ \dfrac{\partial \dot{n}_2}{\partial \boldsymbol{x}} \end{bmatrix} \tag{I-60}$$

$$\frac{\partial n_1}{\partial \boldsymbol{x}} = \begin{cases} n1x1 = 0 \\ n1x2 = -2c_2 k_4 \dot{\theta}_1 \dot{\theta}_2 - k_5 c_2 \dot{\theta}_2^{\,2} \\ n1x3 = -2s_2 k_4 \dot{\theta}_2 \\ n1x4 = -2s_2 k_4 \dot{\theta}_1 - 2k_5 s_2 \dot{\theta}_2 \\ n1x5 = 0 \\ n1x6 = 0 \end{cases}$$

$$\frac{\partial n_2}{\partial \boldsymbol{x}} = \begin{cases} n2x1 = 0 \\ n2x2 = n_3 c_2 \dot{\theta}_1^{\,2} \\ n2x3 = 2s_2 n_3 \dot{\theta}_1 \\ n2x4 = 0 \\ n2x5 = 0 \\ n2x6 = 0 \end{cases} \tag{I-61}$$

$$\frac{\partial \dot{n}_1}{\partial x} = \begin{cases} dn1x1 = 0 \\ dn1x2 = -2k_4(-s_2\dot{\theta}_1\dot{\theta}_2^{\,2}+c_2\ddot{\theta}_1\dot{\theta}_2+c_2\dot{\theta}_1\ddot{\theta}_2)-k_5(-s_2\dot{\theta}_2^{\,3}+2c_2\dot{\theta}_2\ddot{\theta}_2) \\ dn1x3 = -2k_4(c_2\dot{\theta}_2^{\,2}+s_2\ddot{\theta}_2) \\ dn1x4 = -2k_4[2c_2\dot{\theta}_1\dot{\theta}_2+s_2\ddot{\theta}_1)-k_5(3c_2\dot{\theta}_2^{\,2}+2s_2\ddot{\theta}_2) \\ dn1x5 = -2k_4s_2\dot{\theta}_2 \\ dn1x6 = -2k_4s_2\dot{\theta}_1-k_52s_2\dot{\theta}_2 \end{cases} \tag{I-62}$$

$$\frac{\partial \dot{n}_2}{\partial x} = \begin{cases} dn2x1 = 0 \\ dn2x2 = n_3(-s_2\dot{\theta}_1^{\,2}\dot{\theta}_2+2s_2\dot{\theta}_1\ddot{\theta}_1) \\ dn2x3 = n_3(2c_2\dot{\theta}_1\dot{\theta}_2+2s_2\ddot{\theta}_1) \\ dn2x4 = n_3c_2\dot{\theta}_1^{\,2} \\ dn2x5 = 2n_3s_2\dot{\theta}_1 \\ dn2x6 = 0 \end{cases} \tag{I-63}$$

The partial derivative of equation (I-53) is:

$$\frac{\partial(\hat{e}\dot{n})}{\partial x} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & den52 & den53 & den54 & den55 & den56 \\ 0 & den62 & den63 & den64 & den65 & den66 \end{bmatrix} \tag{I-64}$$

where:

$$\begin{aligned} dn1 &= \dot{n}_1 \\ dn2 &= \dot{n}_2 \end{aligned} \tag{I-65}$$

and,

$$
\begin{cases}
den52 = -\big(n1x2(dm11i) + n1(dm11ix2) + n2x2(dm12i) + n2(dm12ix2) \\
\quad + dn1x2(m11i) + dn1(m11ix2) + dn2x2(m12i) + dn2(m12ix2)\big) \\
den53 = -\big(n1x3(dm11i) + n1(dm11ix3) + n2x3(dm12i) + n2(dm12ix3) \\
\quad + dn1x3(m11i) + dn1(m11ix3) + dn2x3(m12i) + dn2(m12ix3)\big) \\
den54 = -\big(n1x4(dm11i) + n1(dm11ix4) + n2(dm12ix4) \\
\quad + dn1x4(m11i) + dn1(m11ix4) + dn2x4(m12i) + dn2(m12ix4)\big) \\
den55 = -\big(n1(dm11ix5) + n2(dm12ix5) + dn1x5(m11i) \\
\quad + dn1(m11ix5) + dn2x5(m12i) + dn2(m12ix5)\big) \\
den56 = -\big(n1(dm11ix6) + n2(dm12ix6) + dn1x6(m11i) \\
\quad + dn1(m11ix6) + dn2(m12ix6)\big)
\end{cases}
\qquad \textbf{(I-66)}
$$

$$
\begin{cases}
den62 = -\big(n1x2(dm21i) + n1(dm21ix2) + n2x2(dm22i) + n2(dm22ix2) \\
\quad + dn1x2(m21i) + dn1(m21ix2) + dn2x2(m22i) + dn2(m22ix2)\big) \\
den63 = -\big(n1x3(dm21i) + n1(dm21ix3) + n2x3(dm22i) + n2(dm22ix3) \\
\quad + dn1x3(m21i) + dn1(m21ix3) + dn2x3(m22i) + dn2(m22ix3)\big) \\
den64 = -\big(n1x4(dm21i) + n1(dm21ix4) + n2(dm22ix4) \\
\quad + dn1x4(m21i) + dn1(m21ix4) + dn2x4(m22i) + dn2(m22ix4)\big) \\
den65 = -\big(n1(dm21ix5) + n2(dm22ix5) + dn1x5(m21i) \\
\quad + dn1(m21ix5) + dn2x5(m22i) + dn2(m22ix5)\big) \\
den66 = -\big(n1(dm21ix6) + n2(dm22ix6) + dn1x6(m21i) \\
\quad + dn1(m21ix6) + dn2(m22ix6)\big)
\end{cases}
\qquad \textbf{(I-67)}
$$

therefore, p(n) is:

$$p(n) = \begin{bmatrix} 1 & 0 & T & 0 & 0 & 0 \\ 0 & 1 & 0 & T & 0 & 0 \\ 0 & 0 & 1 & 0 & T & 0 \\ 0 & 0 & 0 & 1 & 0 & T \\ 0 & T(dbu52+den52) & T(den53) & T(dbu54+den54) & 1+T(den55) & T(den56) \\ 0 & T(dbu62+den62) & T(den63) & T(dbu64+den64) & T(den56) & 1+T(den66) \end{bmatrix} \tag{I-68}$$

From equations (I-34), (I-35),(5-116),(5-117), the following matrices are also needed for using in extended Kalman filter algorithm:

$$c = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ dy31 & dy32 & dy33 & dy34 & dy35 & dy36 \\ dy41 & dy42 & dy43 & dy44 & dy45 & dy46 \end{bmatrix} \tag{I-69}$$

$$
\boldsymbol{a} = \begin{bmatrix} 1 & 0 & T & 0 & 0 & 0 \\ 0 & 1 & 0 & T & 0 & 0 \\ 0 & 0 & 1 & 0 & T & 0 \\ 0 & 0 & 0 & 1 & 0 & T \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad \boldsymbol{b} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & & & & \\ 0 & 0 & & T\dot{M}^{-1} & & TM^{-1} \end{bmatrix} \tag{I-70}
$$

$$
\boldsymbol{e} = -\boldsymbol{b}
$$

$$
\boldsymbol{f} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & T & 0 \\ 0 & 0 & 0 & 0 & 0 & T \end{bmatrix} \tag{I-71}
$$

$$
\boldsymbol{\omega} = \begin{bmatrix} \omega_1 & 0 & 0 & 0 \\ 0 & \omega_2 & 0 & 0 \\ 0 & 0 & \omega_3 & 0 \\ 0 & 0 & 0 & \omega_4 \end{bmatrix} \tag{I-72}
$$

I.2.1 Extended Kalman Filter(EKF) Algorithm for dynamic based Example

The Extended Kalman Filter is given by[81]:

- Kalman Gain matrix:

$$K(n) = P(n)^- \; C^T(n) \; [C(n) \; P(n)^- \; C^T(n) + w_d]^{-1} \tag{I-73}$$

- State estimate update:

$$\hat{x}(n)^+ = \hat{x}(n)^- + K(n) \left(y(n) - C(n) \; x(n)^- - e(n)\right) \tag{I-74}$$

- State estimate extrapolation:

$$\hat{x}(n+1)^- = a(n) \; \hat{x}(n)^+ + b(n) \; u + \hat{e}\hat{N} + \hat{f}\hat{V} + \hat{P}(n) \; (x - \hat{x}) \tag{I-75}$$

- Error covariance update:

$$P(n)^+ = \left(I - K(n) \; C(n)\right) \; P(n)^- \tag{I-76}$$

- Error covariance extrapolation:

$$P(n+1)^- = a(n) \; P(n)^+ \; a(n)^T + V_d \tag{I-77}$$

The noises w_d and V_d are assumed Gaussian white noises, time invariant with zero mean. In these equations, the superscript $^-$ refers to a time just before, while the superscript $^+$ refers to a time just after a sampling instant. Thus, $P(n)^-$ is the calculated error covariance just before the nth sample.

Appendix J
(Experimental results of Extended Kalman Filter)

J. Experimental results

The explanation of following experimental results can found in section 5.5.3.1.

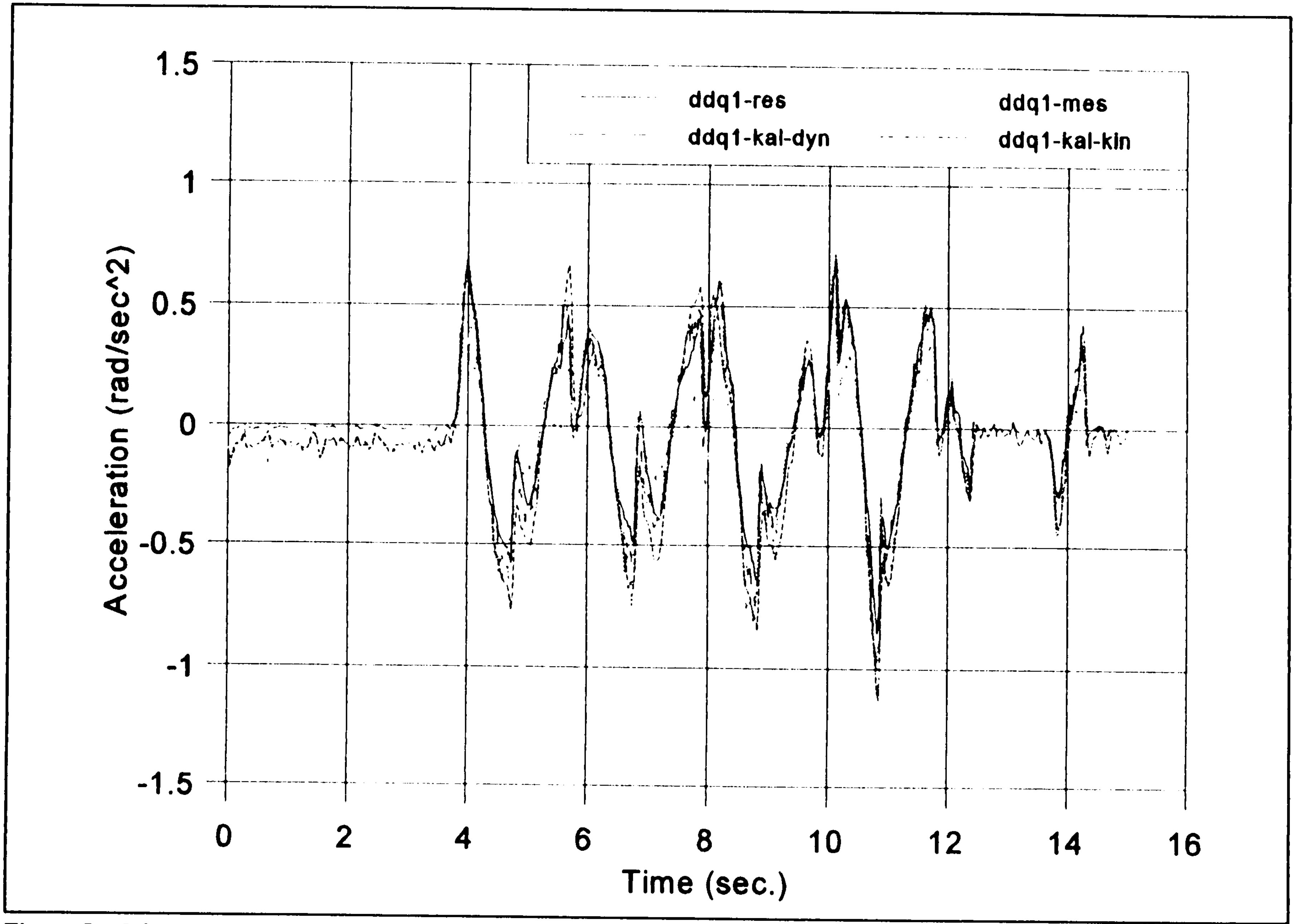

Figure J-1 Joint one acceleration based on four methods, test #2 (experimental results)

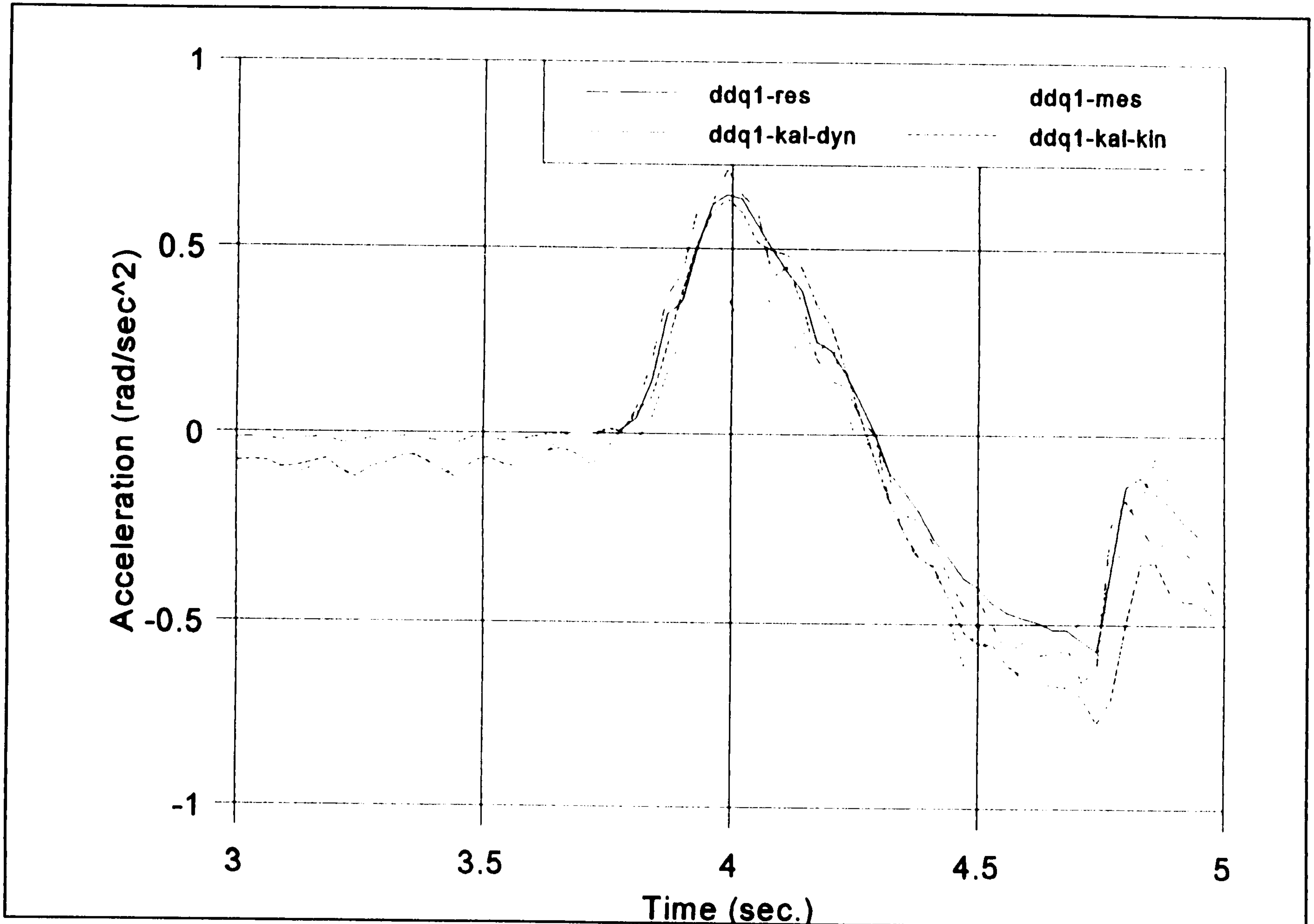

Figure J-2 Joint one acceleration, Zoom area of the figure (5-58), test #2 (experimental results)

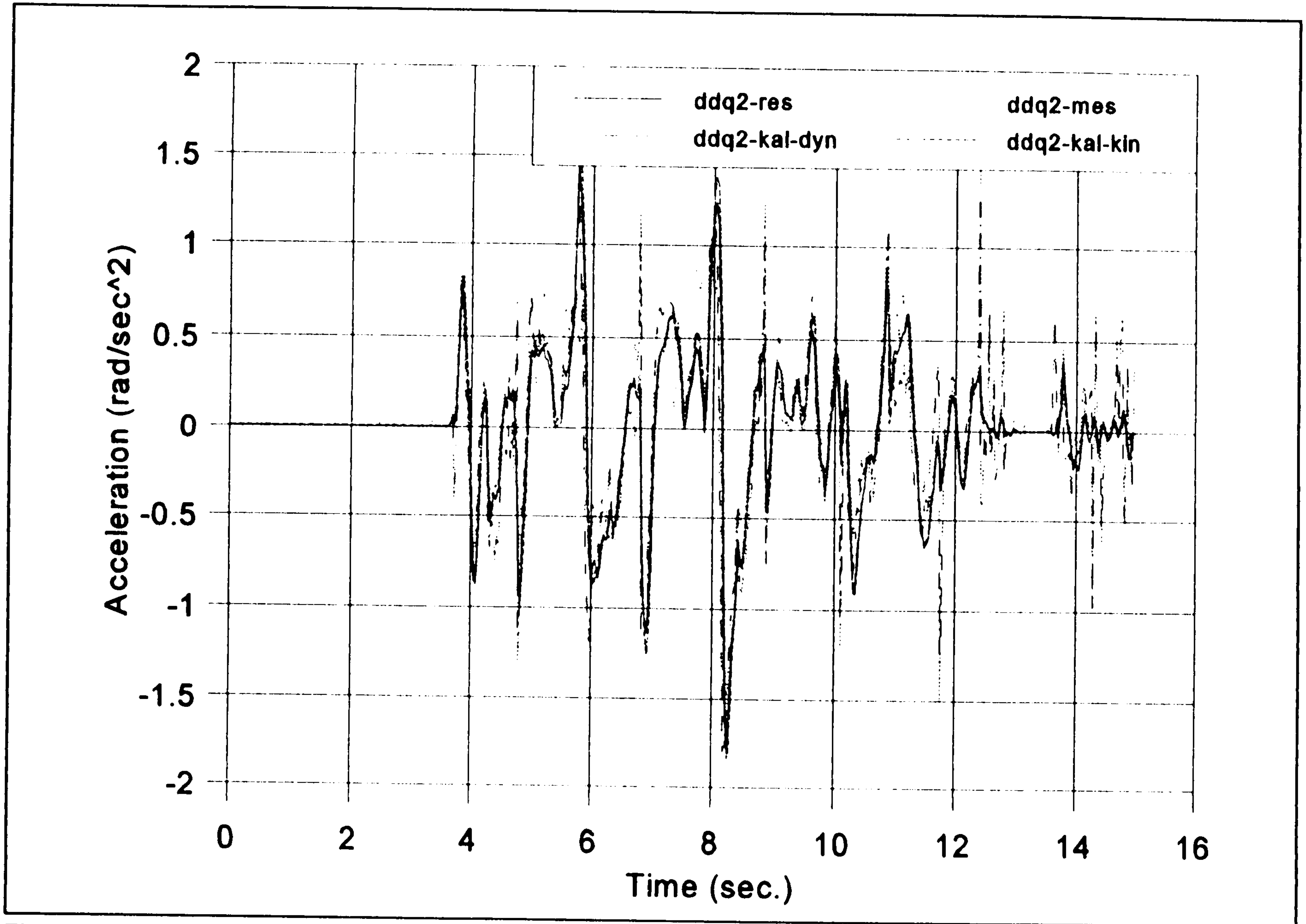

Figure J-3 Joint two acceleration based on four methods, test #2 (experimental results)

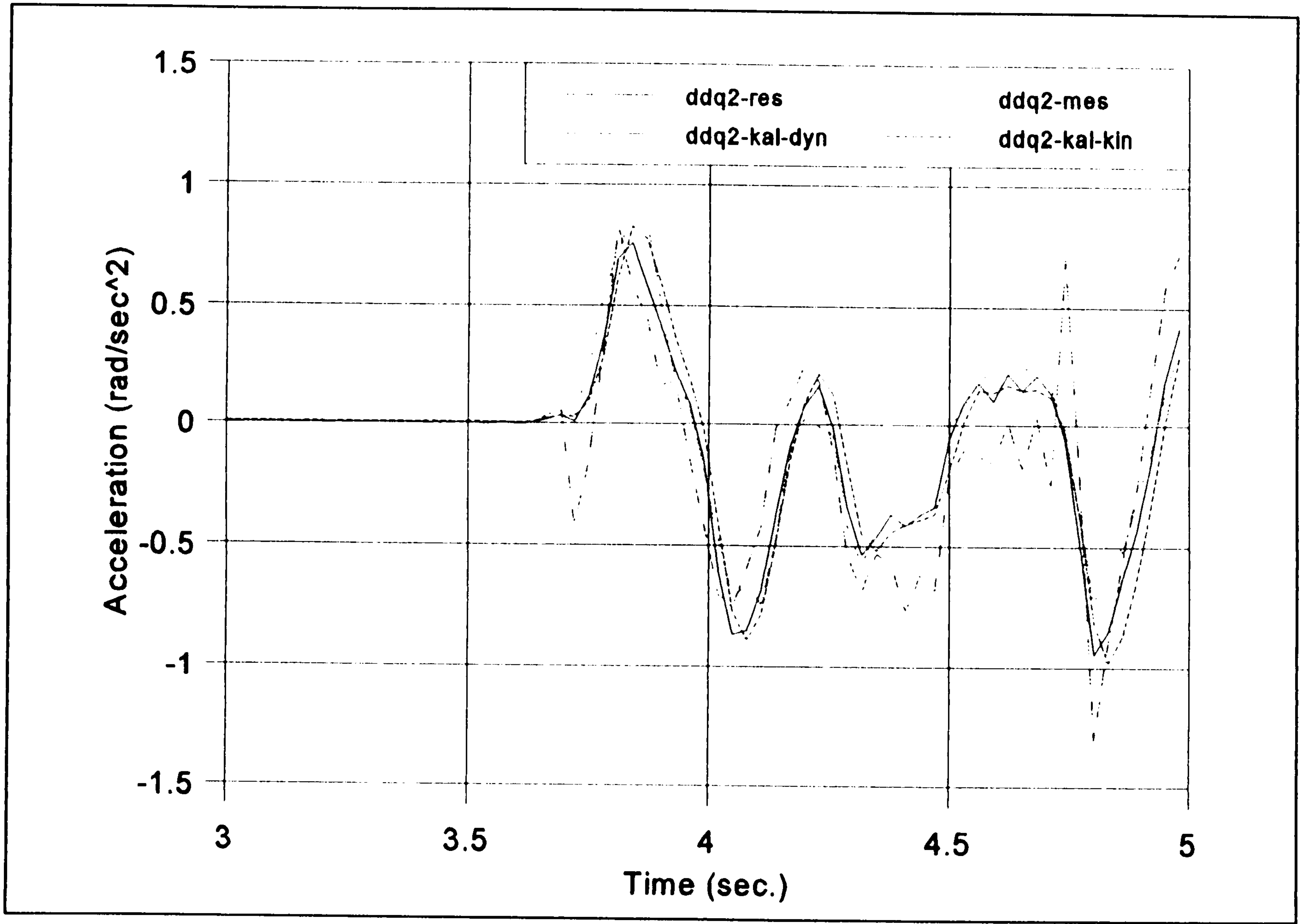

Figure J-4 Joint two acceleration, zoom area of figure (5-60), test #2 (experimental results)

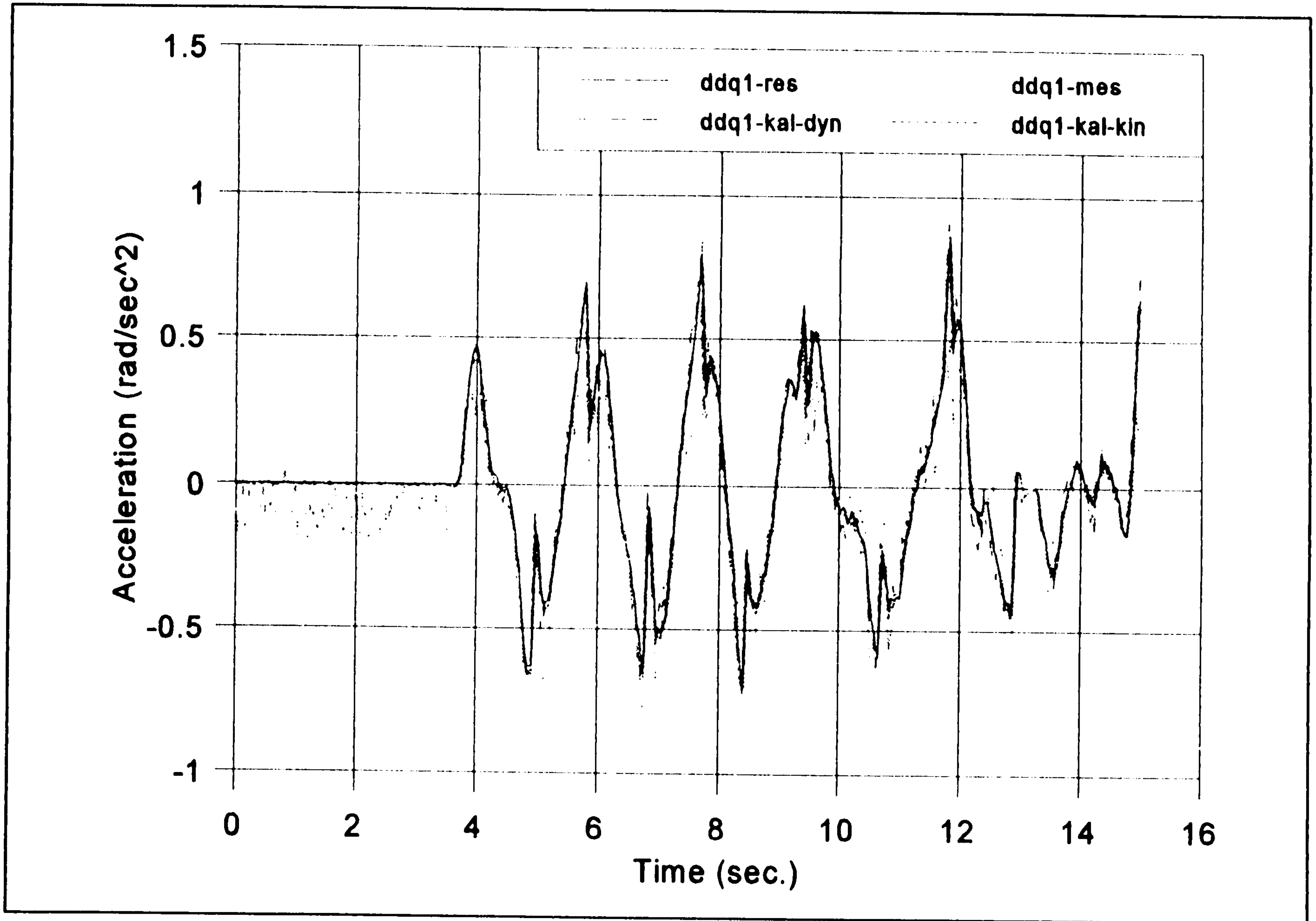

Figure J-5 Joint one acceleration based of four methods, test #3 (experimental results)

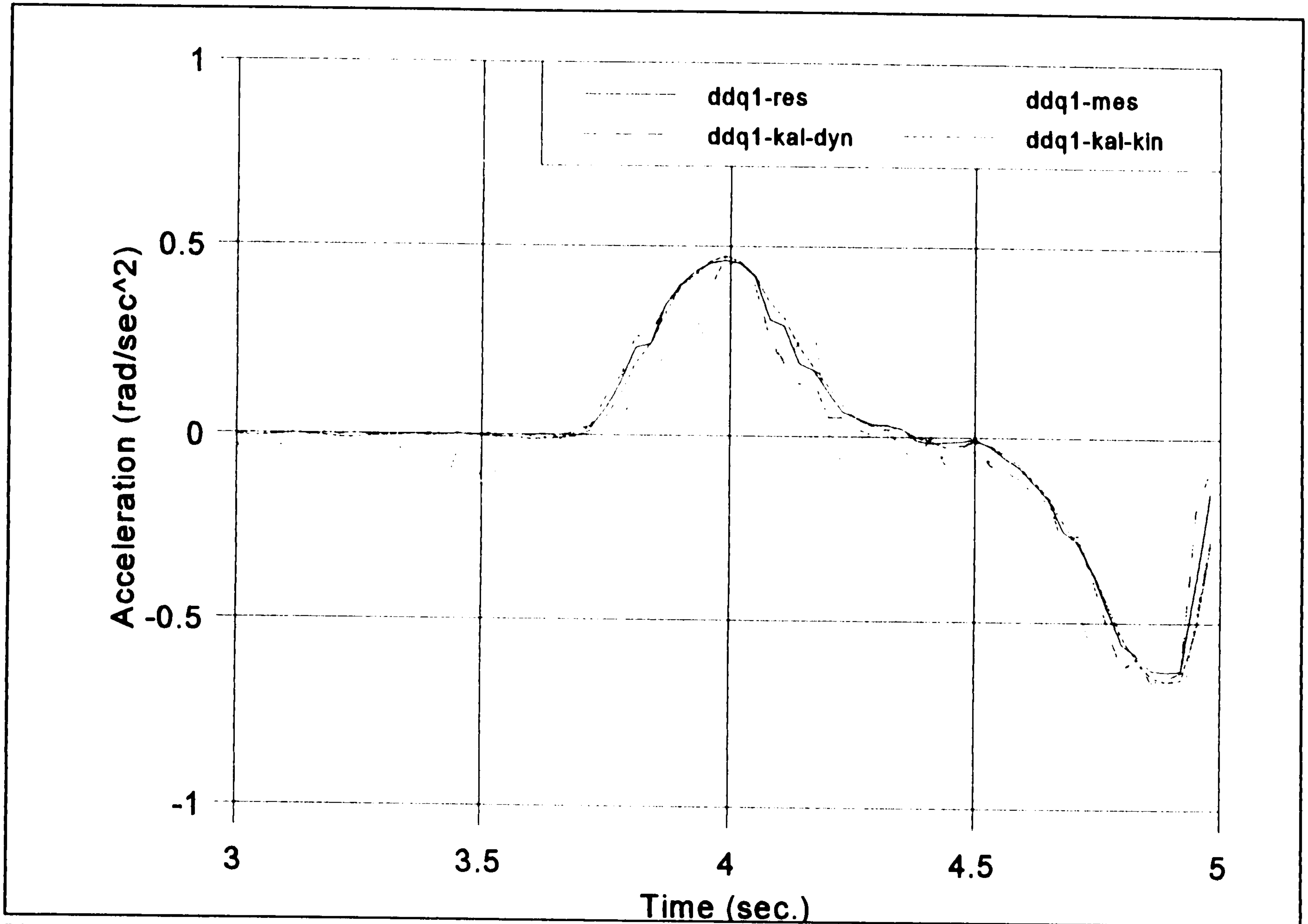

Figure J-6 Joint one acceleration, zoom area of the figure (5-62), test #3 (experimental results).

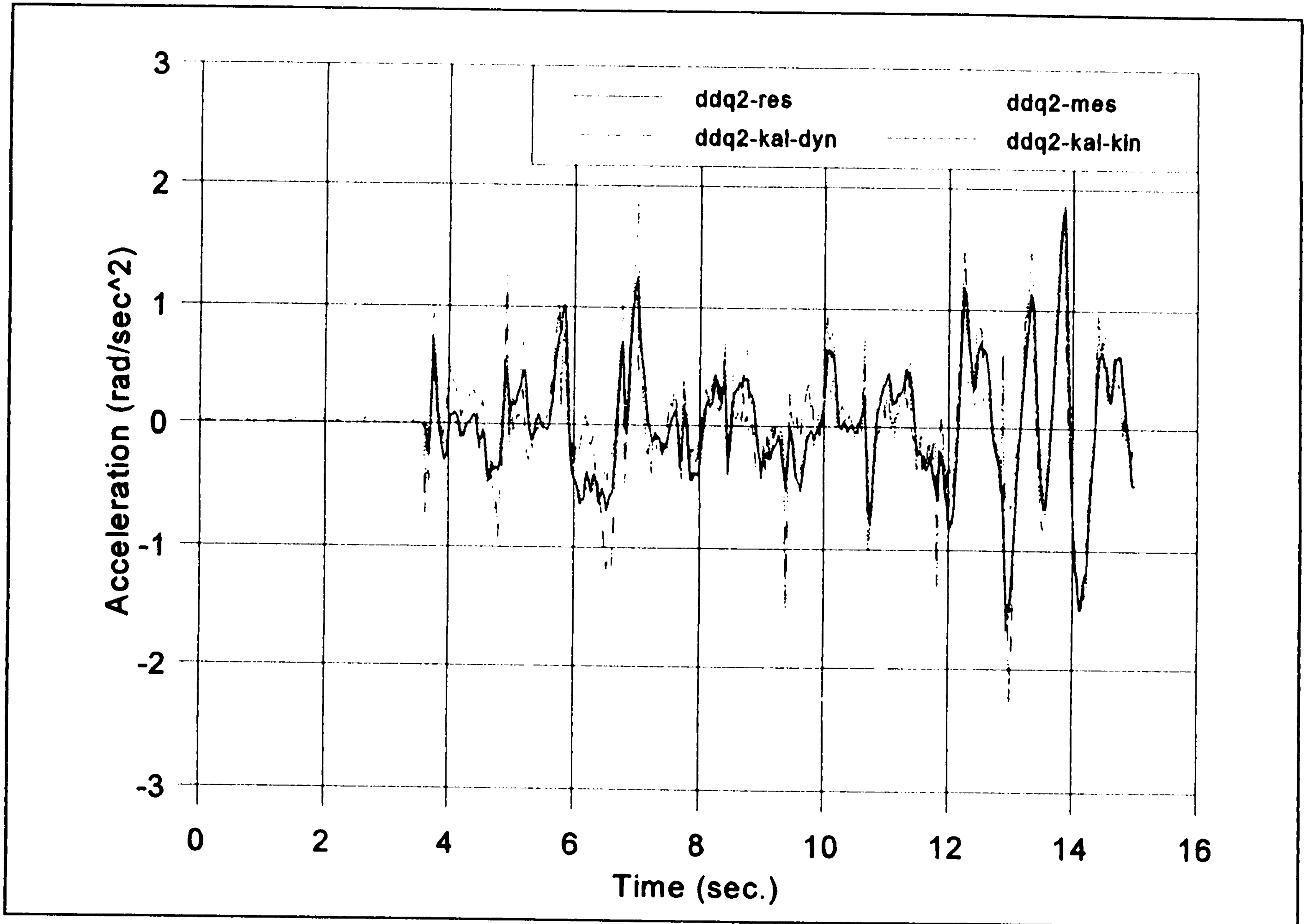

Figure J-7 Joint two acceleration based on four methods, test #3 (experimental results)

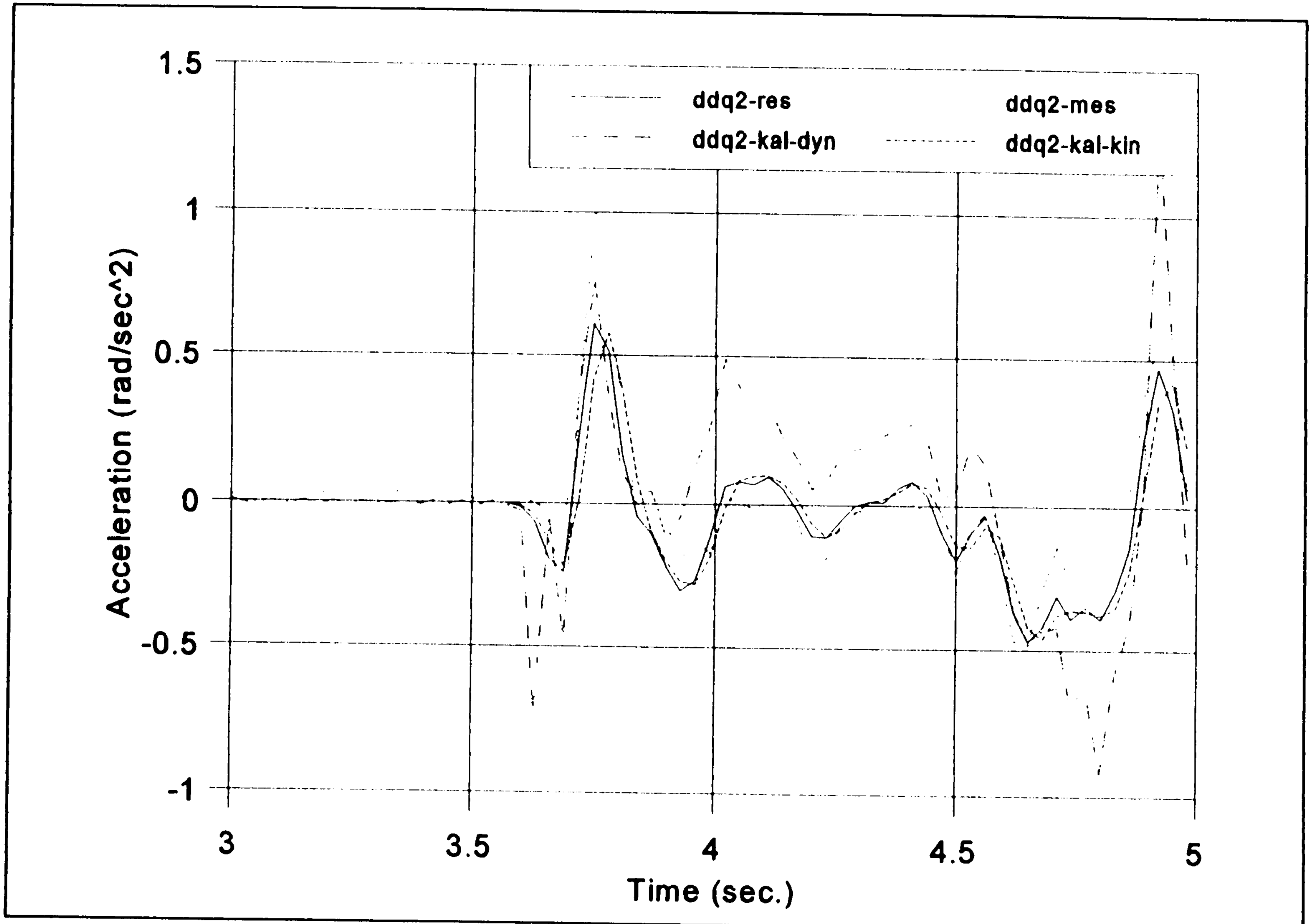

Figure J-8 Joint two acceleration, zoom area of the figure (5-64), test #3 (experimental results)

IMAGE EVALUATION
TEST TARGET (QA-3)

150mm

6"

APPLIED IMAGE . Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989