Trajectory Planners for Cooperative Control of Two Industrial Robots and Belt Drives

BY

T.S.S. Jayawardene

M.Sc. in Operational Research, 2003

A dissertation submitted in partial fulfillment of
the requirements for the Doctor of Philosophy
degree in Advanced Systems Control Engineering,
Graduate School of Science and Engineering,
Saga University

March, 2005

Supervisor: Professor Masatoshi Nakamura

Abstract

This thesis focuses on trajectory planning strategies for high-speed, vibration restrained position control of belt drives and cooperative contour control of two robots in view of increasing the speed of cooperative task. The proposed solutions have been devised, implemented and verified for effective functionality. The trajectory planning in this context is carried out considering the relevant kinematic constraints met in actual practice; the maximum joint velocity constraints and the maximum joint acceleration constraints. The proposed planners are based on the principles of kinematics and the trajectory planning scenarios and, the issues are critically reviewed.

For belt driven machine, a fourth order kinematic model integrating belt reaction torque is systematically derived, and thereby explained the spiky phenomenon in velocity profile of motor position, when an acceleration change is experienced. Further, a feed forward dynamic compensator is proposed to restraint vibration and to improve dynamic characteristics of the belt drives. The proposed feed forward compensator is a combination of inverse dynamics of the system and a desirable dynamic filter, which reforms the dynamic characteristics of the existing system. The planned trajectories at low speeds and high speeds are extensively tested for accurate performance with an actual belt driven machine and the results are illustrated.

The proposed trajectory planners for two-robot cooperation are basically of two types. 1) Given objective cooperative trajectory exceeding the dynamic bounds of a single robot is decomposed into two concurrent complementary trajectories of two robots maneuvered simultaneously 2) For a specified objective locus, the minimum time complementary trajectories for cooperation are planned. The objective locus used to exemplify the concept of trajectory planners in both cases is an S-shaped locus and realization of the trajectories are carried out under maximum joint acceleration constraints. In the former cooperative trajectory planner, a fair task distribution is accomplished by minimizing the difference in maximum joint velocities of two robots. The complexities in planning trajectories are coped with a two-stage trajectory-planning paradigm backed with a short-listing criterion. A fourth order spline technique for position, minimizing the joint acceleration is also derived theoretically. The latter, minimum time cooperative trajectory planner, is of bang-bang type in acceleration profile and the fairness of each robot contribution is achieved through an additional contribution constraint for each robot to the cooperative task. The applicability of the trajectory-planning concept has been verified with cooperative trajectories having sharp corners.

Since the proposed trajectory planners concerned under the thesis work are off-line and therefore they can be conveniently applied to existing servo systems irrespective of the computational power of in-use controller. Neither, a dramatic change in the existing hardware setup nor a considerable reconfiguration of the system is demanded in instrumentation point of view. This requirement of minimal changes in adaptation enhances the pragmatic significance of the proposed schemes.

Approval

Graduate School of Science and Engineering
Saga University
1-Honjomachi, Saga 840-8502, Japan

CERTIFICATE OF APPROVAL

Ph.D. Dissertation

This is to certify that the Ph.D. Dissertation of

T.S.S. Jayawardene

B.Sc. Engineering in Electronics and Telecommunication Engineering, 1996
M.Sc. in Operational Research, 2003

has been approved by the Examining Committee
for the dissertation requirement for the Doctor of Philosophy
degree in Robotics and Systems Control
at the March, 2005 graduation.

Dissertation committee: ______________________

Supervisor, Prof. Masatoshi Nakamura
Dept. of Advanced Systems Control Engineering

____

Member, Prof. Katsunori Shida
Dept. of Advanced Systems Control Engineering

____

Member, Prof. Keigo Watanabe
Dept. of Advanced Systems Control Engineering

____

Member, Assoc. Prof. Satoru Goto
Dept. of Advanced Systems Control Engineering

Copyright©

Dedication

To my loving, courageous mother, uncle, aunt
and
To all my loving teachers

Acknowledgements

First and foremost, the author expresses his heartfelt gratitude to his supervisor as well as academic advisor, Professor Masatoshi Nakamura for valuable guidance, his suggestions and encouragement. It has been so indispensable throughout the recent three years of his academic and research carrier in Saga University, Japan. Professor Masatoshi Nakamura has been more than an academic supervisor. He showed a well disciplined thinking pattern which conditioned his students in such a way they be good people in the society. It is both a privilege and a pleasure to be his student. The author owes a debt of gratitude to Associate Professor Satoru Goto for kind assistance, his guidance and support given while being one of his supervisors.

The author would be very grateful to the members of the dissertation committee, Professor Katsumori Shida, Professor Keigo Watanabe, and Associate Professor Satoru Goto for wading through numerous drafts of this document. Their insight contributed through comments and suggestions has been very helpful.

The author extends sincere gratitude to his scholarship donor, Science and Technology personnel development project of The Government of Sri Lankan on ADB loan. The author would like to thank Saga University for the financial support given through research/teaching assistanceship and for providing an invigorating environment to explore new ideas. The author expresses his great gratitude to Dr. Indralingm, the former head of the Department of Mathematics, University of Moratuwa and to Dr. Chandana Perera, a senior lecturer in the Department of Management of Technology, University of Moratuwa for being the supervisors to M.Sc. in Operational Research. Their guidance, support and encouragement in the master's studies immensely help to make the headway towards the doctoral studies. The colleague staff members of the university of Moratuwa, especially Dr. Nirmali de Silva and Mr. N.L. Wanigatunga (the former and the current head of the Department of Textiles and clothing Technology, University of Moratuwa) are also reminded with a great gratitude and a grateful feeling as they basically facilitated the logistics of the higher study opportunity in Saga University.

The author expresses his gratitude to Professor Nobuhiro Kyura, Department of Electrical Engineering, Kinki University (in Kyushu), for his constructive comments and inspiring ideas. A special thank conveys to Dr. N. Egashira, Kurume Institute of Technology and Professor S. Nishida, Fukuoka Institute of Technology for the valuable comments and remarks made in research discussions.

The author has had great discussions with Rohan Munasinghe, Koliya Pulasinghe, Sisil Kumarawadu, Lanka Udawatta, Chandimal Sanjeeva, Liu Peng, Daisuke Kushida and Tao Zang. These discussions have had profound impact on the ideas of this thesis. Duminda Nishantha helped the author to tackle few tough spots of coding through imparting modern programming techniques and coding paradigms.

The author is greatly indebted to Ms. Masuda Chizuko, who taught Japanese language and thereby paved the way to accustom to Japanese society. She deserves the warmest gratitude for the persistent support and kind assistance provided to recognize the cultural norms, which are essential to lead a comfortable and enjoyable life in Japan. She supported the author in all exigency needs and became “a mother in Japan” to the author. Youth Federation for World Peace (YFWP) and Saga Prefecture International Relationship Association (SPIRA) must be thanked for organizing international sports events, in which the author participates and makes his life enjoyed with extra curricular activities. The author shows his gratefulness to SPIRA for their kind assistance and cooperation received. And of course, the author does not forget to remember his mother, uncle, aunt and sisters who have all stood still with him through the best of times and the worst of times. Their unconditional love and support have helped me to get through some very difficult times. So the author does his best to give them a reason to be proud of.

The author extends his sincere thanks to Lecturer Dr. T. Sugi, Mr. K. Nagafuchi, Technician, Ms. M. Egashira and M. Iwanaga, Secretaries of the Advanced Systems Control Engineering Laboratory, for the generosity and assistance given throughout his stay in Japan. Author also extends his heartfelt gratitude to his colleagues of the Advanced Systems Control Engineering Laboratory, Saga University for their great company and indispensable assistance, which helped him to adopt Japanese society and culture. He remembers all his Japanese and Sri Lankan friends with a lovely and grateful feeling.

Contents

Page

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

1.1.1 Brief history and robot definition

History of modern industrial robot runs to early 1940s to the invention of "Machina Speculatrix" by Grey Walter and "Beast" by Johns Hopkins. The first robot company called Universal Automation, later shortened to unimation was established by Engleberger, who was later called the father of robotics [1]. George Devol, who worked with Engleberger, designed the first programmable robot called "unimates" in 1954 and held the patent for the first industrial robot [2]. First ever computer controlled robot was developed by Ernst at MIT in 1961 [3]. Concurrent dramatic development in robotics hardware and theoretical innovations makes robotics into a concrete discipline by itself. In 1980s robot industry entered a phase of rapid growth, when many institutions introduced programs and courses in robotics.

The word "robot" came from the Czech word "robota" meaning forced labour, and Karel Capec coined it in 1923. There are many definitions suggested for industrial robots and all of them encompass the notion of mobility, programmability and the use of sensory feedback in determining subsequent behavior, though the word may conjure up many levels of sophistications. For the sake of completeness, few popular definitions are stated below.

An automatic device that performs functions normally ascribed to humans or machine in the form of a human-Webster Dictionary [4].

A programmable multifunctional manipulator designed to move materials, parts, tools or specialized devices through various programmed motions for the performance of a variety of tasks-Robot Institute of America [5].

1.1.2 Constructional details and robot classifications

Interconnection of links by different kinds of joints constitutes the mechanical structure of the robot and it is an open kinematic chain by its nature. Links could be either rigid (rigid link robots) or flexible (flexible robots) while joints could be prismatic, revolute or twist type. Each joint is equipped with a prime mover; generally an electric motor and sensors are devised to detect position and velocity information of each joint for controlling purposes. Carefully designed separate controllers are devoted to motion control of each joint and PID controllers are most popular in industrial robots due to their intrinsic robust characteristics. In addition to the generic explanation furnished on robot's basic constructional details, a more specific and detailed description is provided in **Appendix C** pertaining to a typical industrial robot called Performer MK3.

A number of robot categorization schemes are available based on constructional features such as power source, type of gripper, anatomy and the intended applications such as under sea, space etc. In control point of view, most relevant categorization is based on robotics anatomy determined by the geometry of the robot links, joint types as well as their arrangement and it could be briefly illustrated in Fig.1.1. It is worth observing that the control schema, the dexterity of robot and the working envelop is highly influenced by this anatomical configuration too.

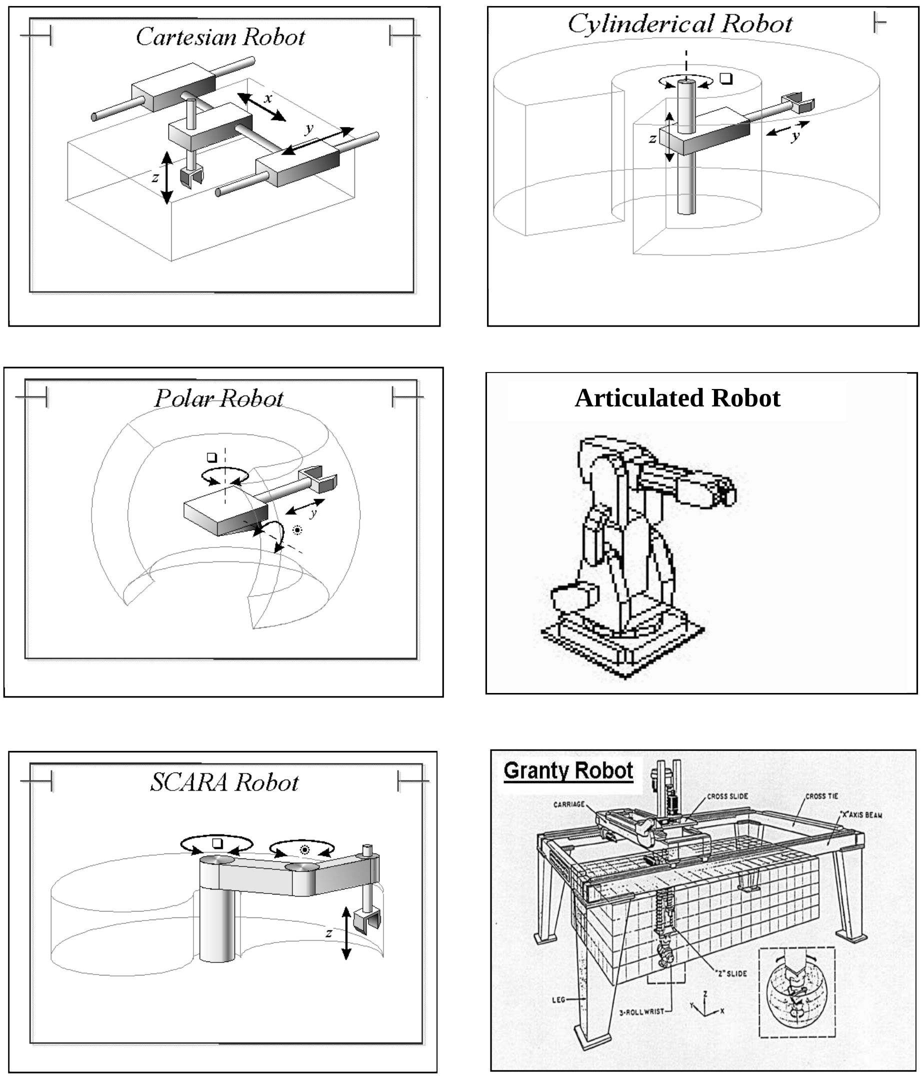

Figure 1.1: Anatomical Categorization of Robots

Figure 1.1 briefly illustrates few basic robot types namely Cartesian robot, cylindrical robot, polar robot, articulated robot, SCARA robot,and Gantry robot. A few more sofisticated types are available and some of them can be stated as insects, walking legs, humanoid robots, mobile robots and automatic guided vehicle (AGV) [6]. Development of cooperative trajectory planners for articulated robot arms and Cartesian robots can be found in Chapters 4 and 5.

In the evolution of robots, Japanese Industrial Association identified six categories referring to classes whereas Robotics Institution of America dealt with only four categories, which were denoted by class 3 to 6 [7]. However Association Francarse de Robotique classified the generation of robots into four types namely telerobotics, sequencing robots, CNC robots and intelligent robots. The six categories of robots defined by Japanese Industrial Association are

1. *Manual Handling Devices:* A device with multiple degrees of freedom that is actuated by an operator
2. *Fixed Sequence Robot:* A device that performs the successive stages of task according to predetermined, unchanging method and it is hard to modify
3. *Sequence Robot:* A device that performs the successive stages of a task according to a predetermined, unchanging method and easily be programmed
4. *Playback Robot:* A human operator performs the task manually by leading the robot, which records the motions for later playback. The robot repeats the same motion according to the recorded information
5. *Numerical Control Robot:* The operator supplies the robot with a movement program rather than teaching it the task manually
6. *Intelligent Robot:* A robot with the mean to understand its environment and the ability to successfully complete a task despite changes in the surrounding conditions under which it is to be performed

1.1.3 Industrial Applications of Robots

The predominant driving force of the usage of robots in industry is to increase the productivity in sustainable manner through reducing the manufacturing cost while producing high quality consistent products with greater accuracy of robots. However as per the current state-of-the-art robotics, robots are proven to be economically viable in middle scale production, where the flexible automation is effective. Robots are successfully implemented for the industrial tasks that poorly suit human capabilities and they can be primarily used in dirty dangerous environments or for dull difficult tasks. In other words, saving money and people are two key concerns for the employment of robots in industry. Another salient application of robots may be found in unusual environments like clean rooms, high radiation areas, and the environments with high pressure (in deep sea), high temperature (furnaces, volcanic operations) or extremely low temperature. Wafer handling needs the involvement of robots because of the high accuracy claimed by the operation. Toxic waste disposal, search and rescue operations, mine clearance are few potential applications of robots due to intrinsic hazard. Few of more general and frequent operations in the industry together with typical characteristics of operation can be briefly described as follows.Few such applicational illustrations can be found in Fig.1.2.

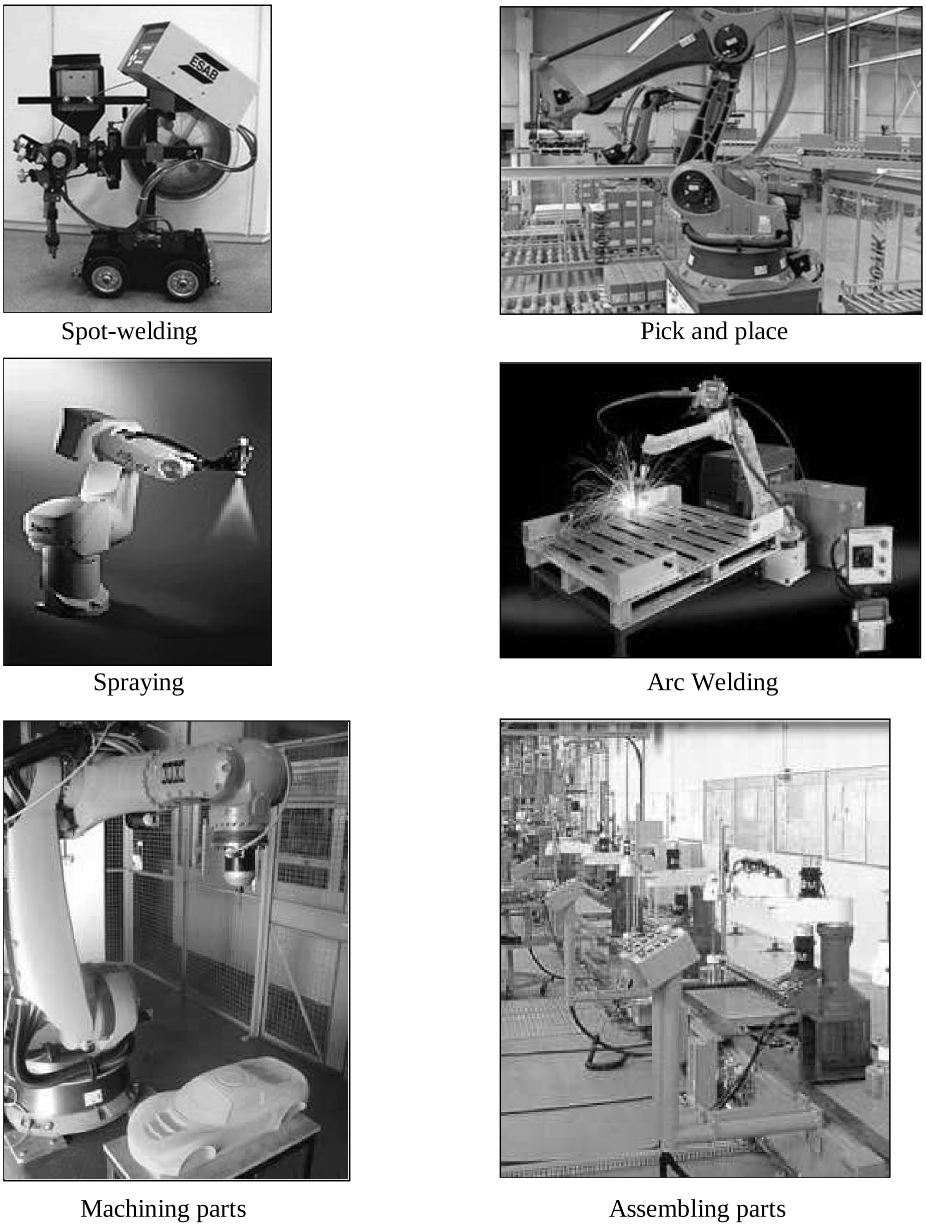

Figure 1.2: Few Industrial Applications of Robots

1. *Spot welding:* This involves applying a welding tool to some object such as a car body at specific discrete locations. End effector of the robot is supposed to achieve point-to-point motion (refer **Appendix F** for definition) across a sequence of positions as fast as possible with sufficient accuracy while avoiding collisions and minimizing jerks so as to ensure longer life span of the robot.
2. *Pick and place:* In this case, object must be held securely enough to prevent it from slipping in the gripper but gently enough to avoid damage. Since the

movement is point to point what happens at the beginning and at the end of the motion is critical but there is some latitude in choosing the intermediate trajectory.

3. *Spraying:* Covering a surface with an even coat of paint is achieved by prespecifying the trajectory along which the arm will move in position and orientation as a function of time. Though spraying is a continuous path application the accuracy of the path is not so crucial.
4. *Seam welding:* This is a continuous path application and usually practiced with real time path correction scheme for path tracking as even a small deviation of welding torch from the seam on the surface is not tolerable.
5. *Electronic Testing:* Detection of flaws in PCBs by probing along the metal races on circuit board, and testing the continuity between the pins through a point-to-point operation are two typical examples.
6. *Metrology:* This is often performed using automated coordinate measuring machines, which are essentially very slow and accurate robots. Through a sequence of point-to-point motions, it measures the dimensions of mechanical parts.
7. *Assembly:* Peg in hole insertion, push and twist insertion, simultaneous multiple peg in hole insertion, screw insertion, force fit insertion, removal of located pins,, flipping parts over, providing and removing temporary support, crimping sheet metal, welding or soldering are few of the basic types of assembly motions. A typical assembly application can be comprised of one or a combination of few basic types of assembly motions listed above.
8. *Machining of mechanical parts:* Grinding, deburring, sanding parts are few of the examples of this category and there should be an ability to follow surface while maintaining the forces required to perform the operation [8].

1.1.4 Introduction to trajectory planning

A meaningful and diligent operation can not be accomplished by robotics hardware alone and the controller should steer the robot along the objective path. In order to realize the objective path, a sequence of adequately close path points are to be input together with the time at which the specified path points to be reached.

A path denotes the locus of points in the joint (configuration) space or operational (working) space, that the manipulator has to follow in execution of the assigned motion. In other words path is a pure geometric description of motion. However, trajectory is a path for which a time law is specified, for instance in terms of velocity and/or acceleration at each point [9]. Therefore, trajectory considers the time history of concurrent positions of every joint when robot has multiple degrees of freedom. In case of Cartesian robots, joint space and working space have straightforward one-to-one mapping relationship, whereas in articulated robots, space transformation is established through kinematics and inverse kinematics (refer **Appendix B** for details), which are inevitably nonlinear because of transcendental functions.

Trajectory planning is the process of generating reference inputs to motion control system ensuring that the robot manipulator executes the planned trajectories from initial posture to final posture. Transition of end effector from one position to another is characterized by motion laws requiring the actuators to exert joint generalized forces, which do not violate the saturation limits and do not excite the typically unmodeled resonance modes of the structure. Therefore consideration of manipulator dynamic limits such as joint velocity and joint torque (for twisting or revolute joints) or joint force (for prismatic joints), alternatively equivalent supremum acceleration, in trajectory planning stage is inescapably essential to avoid potential deteriorations caused in realizing the planned trajectories. However, trajectory planning becomes a tedious task in the light of following issues.

- Time synchronization of concurrent joint positions under imposed dynamic constraints.
- The specifications of the trajectory are given in working space while the constraints are pertinent to configuration space. Hence the problem statement is comprised of mixed constraints in two different coordinate systems.
- Nonlinearity of kinematics and inverse kinematics transformation.
- In general, no closed loop solutions are available for inverse kinematics.
- Space transformation is mildly computationally intensive and quite often leads to longer control intervals (low update rates of trajectory in servoing) in real time planning instances.
- Space transformation is ill defined because it is not one to one mapping.
- Unmodeled characteristics such as neglected resonance modes of the structure.
- Presence of uncertainties like obstacle appearance, payload and inertia variations with robot configuration, estimation errors in servo parameters.
- All but the simplest robots have interference between the joints (coupling effect of the joints).
- Presence of singular points in working space.

As a means of resolving above planning issues, the control architecture of the robots has been divided into three hierarchical layers [10]. The essential features of the trajectory planning can be concisely illustrated with the block diagram given in Fig.1.3.

a) Path Planning
A path planner determines geometric path information without timing information based on collision avoidance and other task requirement.

b) Trajectory planning
This receives a special path descriptions and boundary velocity constraints (zero starting and final velocities) as inputs and it calculates the time history of desired positions and velocities (time synchronization of joint positions)

c) Trajectory tracking
This is also termed as trajectory control in robotics jargon and it is a process of making robots actual position and velocity match some desired values of position and velocity, which are provided to the controller by the trajectory planner.

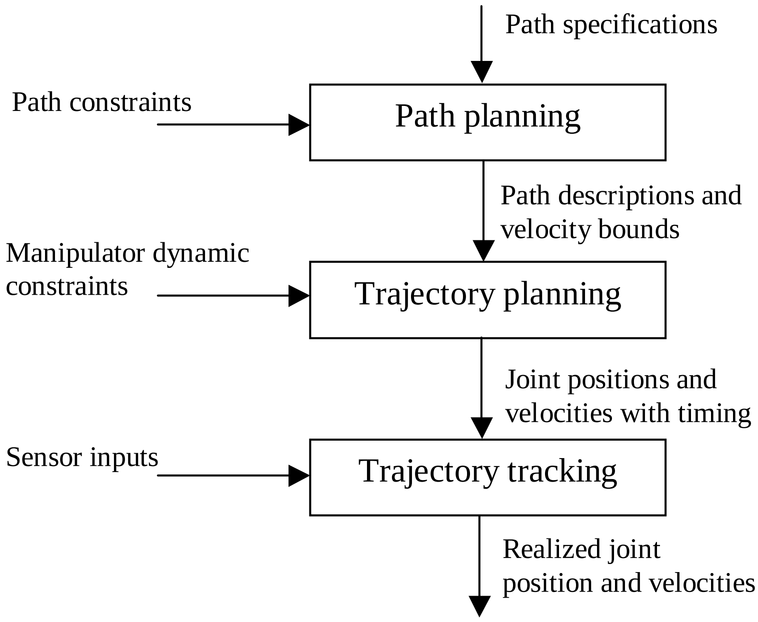

Figure 1.3: Three Layer Hierarchical Model of Trajectory Planning and Controlling

1.1.5 Overview of trajectory planning algorithms and characteristics

A number of stringent requirements are imposed upon robots in order for them to be competitive in the world of manufacturing. Reliability and durability, speed of operation, conformed accuracy, ability to cope with environment uncertainties, sufficient configurability, ease of programming, versatility, and cleanliness are few of them. In accomplishment of these stringent requirements demanded by industrial applications, control schemes devised have to play a key role. Since the control of robots basically hinged on trajectory planning, the trajectory-planning algorithms, essence of trajectory planner perceives utmost importance. Therefore progress on the algorithmic foundations for trajectory planning is crucial for smart and sophisticated control of robots.

The distinct characteristics of trajectory planning algorithms are strictly deterministic by the nature of task and the types of motions involved with it. Most of the real world tasks can be broken down into a sequence of rudimentary control motions, which can be stated as axis limit control motion, linear and rotary motion, and point-to-point control motion or continuous path control motion. Continuous path motion is divided into generic velocity profile control motion, compliant motion and guarded motion according to strategic characteristics associated with. Based on the control objectives, planning algorithms could be categorized as true or near minimal time, accurate or high speed position control, flexible or rigid manipulations, whereas according to the disposition, they can be robust control, on-line or off-line control, point-to-point or continuous path. Approach wise distinction of planning algorithms may find in **Chapter 1.2**.

Volumes of primitive algorithms have been proposed for rudimentary motions, under certain assumptions. Relaxation of assumptions and integration of basic algorithms for complicated tasks are currently under intense securitization of the researchers to reform trajectory planning algorithms into much general framework.

In general terms, trajectory-planning algorithm can ultimately make the robot to realize fast and accurate performance in wide range of repetitive tasks over prolonged shifts under uncertainties. Further to the fulfillment of intended control specifications, the additional characteristics listed below may enhance the effectiveness of planning algorithm [11].

a) The generated trajectories should not be very demanding from computational point of view.

b) Joint positions and velocities are to be continuous functions of time. Continuity of acceleration may also be important for a longer life span of the robot.

c) Undesirable effects should be minimized.

1.2 Literature Review

1.2.1 Belt drives

As a simple low cost lightweight technique of power transmission over moderate distances, belt drives are popular in use. Belt drives provide freedom to locate the motor relative to the load and this phenomenon enables to reduce the inertia of the robot arm in case of robots and therefore belt drives are extensively used in light weight robots. Many unique advantageous characteristics (referred to **Chapter 2.1** for details) of belt drives become inspirational to use in position control systems, but the deterioration of the positioning accuracy due to flexible dynamics of belts is a serious implementation issue encountered by control engineers as most of robot applications claim for a higher level of accuracy.

The compensation of flexibility of belt drives is difficult with a primary actuator due to bandwidth limitations. As a means of making the belt system more "stiff", the usage of second actuator, a dancing bar, was proposed by Gorbet *et al.* in [12]. This approach provides a remedial measure for another control issue, the vibrations of belt drives.

Accuracy of the belt drives is seriously suffered at high speeds and especially under variable load inertia. However, accuracy improvement of belt drives can be realized with a proper and careful controller design taking the flexible characteristics of belts into account. Meantime controller must be sufficiently robust to accommodate uncertainties. Such approach was found in [13] and a controller based on adaptive principle has been proposed.

The jerk (third derivative of position with respect to time) in the planned trajectory plays a significant impact on deterioration of position due to flexible dynamics. Therefore minimum jerk trajectory may enhance the tracking accuracy of belt drives. Nakamura *et al.* [14] suggested a minimum jerk trajectory planner using cubic interpolation techniques. Further, a feed forward dynamic compensator, initially proposed in [15] is devised to improve the tracking accuracy. Through the cancellation of the undesired poles of the system by the zeros of the compensator, delay dynamic compensation is achieved. A theoretical work on the locations of the poles was addressed by Munasinghe *et al.* in [16] and [17].

Intelligent control technique for belt drives has been attempted by Lee *et al.* [18],[19]. In these approaches, he investigated the use of frequency reshaped linear quadratic control in order to implement a low cost intelligent integrated belt driven manipulator, which combines the linear quadratic optimal control with frequency response methods.

1.2.2 Trajectory planning strategies and cooperative planning

In popular contour control approaches, control of the robot was achieved in conveniently separated, two independent sequential stages: off-line trajectory planning and on-line trajectory tracking or servoing [20],[21]. Due to the complexities involved in trajectory due to nonlinear-coupled dynamics and presence of obstacles within working envelop, path-planning stage received a distinctive identity from trajectory planning [10] and own techniques have been developed separately for each type of planning. Path planning has been explored in different avenues; probabilistic path planners [22][23], random path planners [24] and potential field based reactive planners [25]. In the former case a data structure called road maps was constructed in probabilistic way and used to solve individual path planning problems. In random path planner gradient paths were used to get closer to the goal while random walks help to escape from local minima. In potential field based reactive planners, an attractive potential function for the final target and a repulsive potential function for the obstacles were defined. A path is generated to attract the robot to the final point and to repulse away from the obstacles using dynamic programming. Besides, dynamic programming was also employed in path planning [26].

In real industrial systems, constraints and specifications are declared in configuration space (eg:- joint speed and acceleration limits) as well as in working space (path specifications and tolerance limits). Hence Cartesian space trajectory planning and joint space trajectory planning become two viable options. In joint space trajectory planning, only knot points are on the objective path and hence it is lower in accuracy; but it has the following distinct advantages.

- The trajectory is planned in terms of controlled variable during the motion
- Trajectory planning can be done in near real time
- Joint trajectories are easier to plan

Cartesian space planning techniques need frequent space transformation by invoking computationally expensive kinematics and inverse kinematics procedures and hence much appropriate for offline planning. Further transformation from Cartesian coordinate to joint coordinate (inverse kinematics) is ill-defined, because it is not one-to-one mapping.

In classical control approaches of robot manipulators, the end effector motion was resolved into joint motions and joints were actuated with rate and acceleration control [27] [28]. For the sake of simplicity and convenience of trajectory planning, joint dynamics was assumed to be decoupled. Nevertheless, trajectory trackers can generally keep the manipulator fairly close to the desired trajectory even with coupled joint dynamics [29].

In trajectory planners, homogeneous transformations [30] were popularly employed as a means of a generic approach to calculate the position of end effector with respect to the object, though it was not computationally efficient. However, such planning technique were infeasible for real time planning of trajectories and therefore few researchers had probed for fast and efficient calculation paradigms so that the applicability was not restricted to predefined work environment. Computationally efficient inverse kinematic algorithms had been suggested [31], but they were basically confined to non-redundant robot arms in real time planning. As another means of expediting trajectory planning, interpolation based planning techniques were evolved. A limited number of knot points in Cartesian space were converted into equivalent joint coordinates and fixed low degree polynomials were used to interpolate inter-knot-points [32][33]. This technique has been highly exploited in bounded jerk trajectory planning. Dynamic programming based approaches were also admired as a fast means of planning trajectory due to dramatic reduction in space dimension, further to the flexibility granted [34][35].

A number of trajectory planners have been proposed for true minimum time control [21][36] [37], near minimum time [38][39], accurate positioning [40][41], and robust control [20][42][43] despite the type of the path to be point-to-point or continuous. However, above control objectives could be realized with different planning approaches such as intelligent control [44][45], impedance control [46][47][48], resolved acceleration control [49][27], adaptive control [13][50], dynamic [51]-[53] or kinematic [29] control, or hybrid control [54].

Artificial intelligence based trajectory planners are capable of compensating uncertain phenomena like friction, inertia variation with robot's configuration and they can be based on the principles of fuzzy logic, genetic algorithm or neural network [55]. Impedance control is quite effective in improving the interaction between the manipulator and environment, and crucial for successful execution of a certain class of practical tasks, in which the model is a priori known.

Kinematic based planning approaches can be successfully applied in laser cutting, spraying and welding where there is no force interaction between the manipulator and the work piece. Consideration of constant acceleration bounds in kinematic planning became much popular though these bounds varied with position, mass, payload, and even with payload shapes. The worst case bounds, more precisely, the globally greatest lower bounds for acceleration and velocity were selected. However, this could result in under utilization of robots capability.

Computed torque control schemes based on Newton-Euler or Lagrange-Euler formulations [11] were successful in on-line planning due to the advancement in processing power or/and implementation of parallel computer architectures [56][57] despite the time and space complexity associated. This is more suitable for sophisticated tasks demanding force control. There are abundance of tasks like grinding, deburring and so on, which cannot be adequately expressed as a sequence of positions. In such cases, force and motion should be controlled simultaneously in perpendicular directions (compliance motion) and therefore necessarily required a hybrid planning technique.

Adaptive control approach is an efficient way of dealing with robot system uncertainty and complexity, improving the performance in view of unmodeled dynamics, and it does not required a complete knowledge of the system. Adaptive control approaches could be based on the principles of reference adaptive control [58], self tuning type adaptive control [50] or self tuning type adaptive control with feed forward compensator [59]. However in general, adaptive control techniques suffer from the problem of guaranteed global stability.

As these trajectory-planning approaches address fundamental trajectory planning issues, they are equally applicable to plan the trajectories of single robots and plural robots. However, coordination and cooperation are additional issues to be tackled in cooperative trajectory planning and for that many strategies have been suggested. Master slave cooperative strategy has independent controllers, which are easy to implement, and the coordination is achieved through force measurement [60]. In hybrid position cooperative control, a unified robot and object dynamic model have been assumed [61]. Impedance control has systematically extended for cooperative strategies through distributed impedance [62]. Cooperative behavior could be realized at trajectory planning stage only (loose cooperation) or both at trajectory planning and trajectory tracking stages (tight cooperation) [47].

1.3 Motivation

1.3.1 Belt driven machine

Historical pioneering work related to model construction was limited to rigid link manipulators [11] [43] and integrated model considering the inertial belt reaction force for belt drives has not been sufficiently addressed. Further, analytical attempts on belt drives were confined to a single control issue such as vibration [63], or accurate positioning [13]. A detailed analysis or a careful investigation of belt drives considering most appropriate industrial application constraints such as acceleration and velocity limits, may not be found in the literature. Therefore an accurate model for integrated belt driven servo systems as well as cause and effect analysis of belt drives leading to poor accuracy has been existed as outstanding open problem for quite a long time.

1.3.2 Cooperative control

The cost of robotization should be overcome by the benefits gained through alternative means of making the utilization of robots economically justifiable. To support the fact of economical viability, minimum time trajectory planners with required level of precision received a great attention [21][38]. To harness the economical benefits of robots, cooperative control of multiple robots emerged as a discipline and it was inspired by optimal control techniques. Aside from the economical motivations, a number of unanswered scientific motivations have enticed the themes covered in Chapters 3 and 4.

Manipulation of common objects cooperatively held by multiple robots has received much attention of the researchers and few theoretical foundations have been developed [64][65]. Bi arm cooperation is the simplest case and it has been intensively investigated in literature [66]-[69]. This kind of cooperation basically enhanced the payload capacity through parallelism. For a given motion of a common object, paths of individual robots are a priori known for trajectory planning, since path planning of cooperative control could be detached from trajectory planning. However, inter robot force control under secure grasp of a common object, restraint vibrations are key control issues to be addressed.

Cooperative behavior could be achieved by breaking down the complicated entire task into small sub-tasks, which are manageable within the bounds of individual robot capability, and assign such sub-tasks for individual robots. This task decomposition technique was specifically proposed for mobile robots and through the principle of parallelism task completion time could be dramatically reduced. However, this approach was limited to a class of cooperative control problems where the entire task could be optimally divisible into assignable subtasks for individual entities.

In a certain class of strict coordination, neither path planning and trajectory planning be dissociated nor the entire task be resolved into subtasks in a useful way. In such cooperative control instances, cooperative strategy and path planning strategy are embodied in trajectory planner and hence the trajectory planning becomes much intricate. Perhaps due to the complexity of the trajectory planner, this class of strict coordination in view of speeding up the task completion received less attention and hence poorly addressed in literature.

1.4 Contributions of the Thesis

1.4.1 Belt driven systems

The main contribution on belt drives is two fold: First, in the industrial point of view, is to develop a conveniently instrumental vibration restraint high-speed accurate position system for servo controlled belt drives. Second, in the control system research point of view, is to construct an accurate model taking the flexible dynamics and belt reaction torque into account, which is valid even for high-speed operations (refer **Chapter 2.4.2** for details). Further fundamental causes for poor positioning of belt drives are investigated and analyzed. Accuracy of the simulations based on popular numerical techniques has been verified with analytical solutions derived.

1.4.2 Cooperative trajectory planners

This dissertation covers two novel trajectory planners for cooperative control.

- Trajectory planner for bi-arm industrial robot manipulator with a specified cooperative trajectory and bounded cooperative velocity and acceleration under maximum joint acceleration criterion. The fairness of the joint motions of each robot was assured by keeping the maximum joint velocities of two robots as closer as possible.

- Minimum time cooperative trajectory planner for Cartesian robots under given path/locus specifications subjected to joint acceleration constraints of each joints.

1.4.3 Scope of application

Belt drives: The proposed model and the control technique for belt drives were intensively tested with an actual belt driven machine having one degree of freedom and proved the effectiveness with promising results at high speed as well as low speeds. If the coupling effect is negligible or required precision is not too high, the proposed vibration restraint control technique can be conveniently extended to multi axis belt driven robot to drive each individual joint separately. Since the control method has shown substantial robustness, inertia change due to configuration does not degrade the positioning accuracy in multi axis belt driven robots

Cooperative control: Both cooperative control techniques proposed are based on the principles of kinematics, they could be particularly ideal for applications like spraying, laser cutting, or welding where there is no force interaction involved with the motion. The planners are flexible enough to accommodate much complicated contours in view of speeding up the operation through cooperative behavior. Though the cooperative planners are demonstrated with two-dimensional examples, its scope does not restrict to planar cases. However, these planners are confined to prescribed or structured environments since obstacle avoidance issue has not been addressed.

1.5 A Preview: Outline of the Thesis

This section will give the first glimpse of the contents covered by the dissertation and the direction in which the dissertation has been organized. In order to illustrate trajectory planning for servo controllers, two significant areas, belt drives and cooperative control of two robots in view of speeding up, have been selected. Improving position accuracy of belt drives with vibration restraint and decreased task completion time of cooperative control through parallelism are basically investigated.

In **Chapter 2**, mathematical representation of the control problem of belt drives is stated. Stepwise derivation of an accurate model for servo controlled belt drives is presented. Scenario of designing a feed forward compensator to achieve vibration restraint and fast dynamic characteristics are covered. Trapezoidal velocity profile based minimum time trajectory is planned under maximum velocity and acceleration constraints. This trajectory is compensated for delay dynamics and then used for simulation and experiment. Accuracy of the simulation results based on popular numerical techniques has been verified with an analytical solutions derived. The planned trajectories are tested with actual belt driven machine at low speed and high-speed conditions. Further, spiky phenomenon in velocity profile is illustrated and the causes for its generation is discussed.

Chapter 3 describes two-stage cooperative trajectory planner for two industrial robot manipulators under specified objective trajectory. Cooperative maximum velocity and joint acceleration limits of each robot joint are taken into account in planning the cooperative trajectories while a fair task decomposition is ensured through minimizing the difference between the maximum velocities of two robots. Time complexity of the algorithm is outlined and a short-listing criterion is presented as a technique to manage the time complexity. Concept of cooperative control is briefly introduced and the benefits are summarized. The use of RT-Linux as a means of real time servoing is appraised. Simulation and experiment results verify the validity of the proposed planner.

Chapter 4 deals with a time optimal cooperative trajectory planner for two Cartesian robots under bounded acceleration. The path or locus of the objective trajectory is an input and the trajectory planner is of bang-bang type. In accelerative mode planning, condition for no solution is theoretically derived and shown that the necessity of stepping back as a resolution strategy. This chapter includes trajectory planning algorithm and its formulation. Scope of applicability and extensibility of the proposed planner to a more general framework are briefly reviewed.

Chapter 5 is basically devoted to concluding remarks and recommendations. The detailed discussions, possible future developments and generalizations of the present work are provided in this chapter.

Chapter 2

Belt Driven Machine

2.1 Preliminaries

2.1.1 Characteristics of belt drives

Few of the salient characteristics of belt drives could be stated as follows.

1. An efficient low cost light weight power transmission technique especially useful over moderate distances
2. Wheel alignments are not so critical
3. Inherently much quieter
4. Capable of absorbing shock loads and thus isolates vibration of the load from motor part
5. Provide flexibly in positioning the motor relative to load and hence can be reduce the inertia of moving parts
6. Flexible dynamics of belt drives leads to sluggish response, poor positioning and substantial vibration

2.1.2 Experimental setup and schematics of belt driven machine

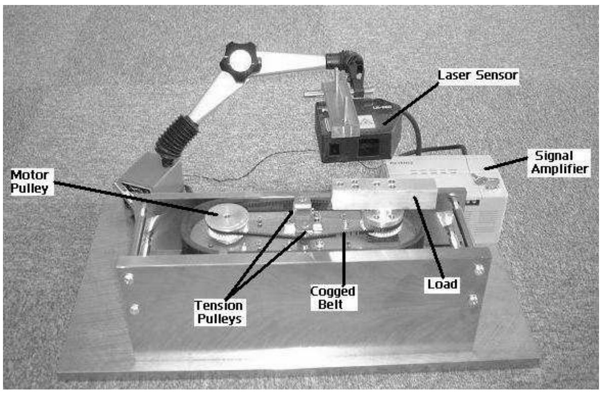

Figure 2.1: Experimental Setup of Belt Driven Machine

The schematic of the experimental setup is illustrated in Fig. 2.2 and its physical arrangement is shown in Fig. 2.1. Load and motor are interconnected with a cogged belt since it can operate accurately at higher velocity and acceleration profiles without

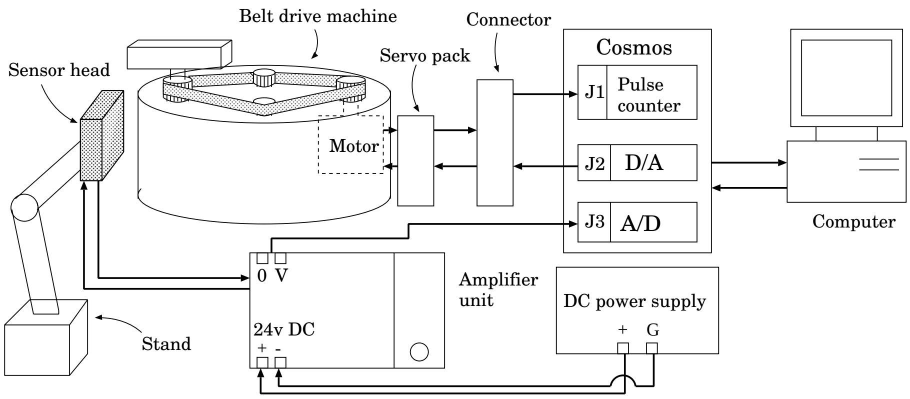

Figure 2.2: Schematic Diagram of Belt driven Machine

any relative slip. The servomotor is excited by an embodied servo controller through resident PI control algorithm.

The reference input, in other words generated trajectory is compensated for delay dynamics and vibrations prior to use it for servoing with the aid of COSMOS, which interfaces digital data with analog servo input. COSMOS is equipped with multi channel A/D and D/A converters, 16MB memory and a digital counter. COSMOS is not only acting as an interface, but also as a data logger to support fast servoing with a sampling time smaller as 125 μs. An optical laser sensor coupled with an amplification unit devised to monitor the actual position and these data are also logged back to the computer used as reference input generator, through COSMOS.

2.2 Problem Statement and Planning Algorithm

2.2.1 Problem statement

Servo controllers undergo current saturation and this phenomenon corresponds to acceleration limits in velocity profiles. The planned trajectories should comply with the acceleration bounds and it can be mathematically expressed as

$$|\ddot{r}(t)| \leq \ddot{r}_{max}; \qquad \forall t, \tag{2.1}$$

where $\ddot{r}(t)$ and $\ddot{r}_{max}$ denote the acceleration of trajectory to be planned and its maximum limit.

Maximum permissible velocity of a joint can either be governed seldom by the hardware limitation of the motors or frequently by the specifications of the application itself. If the operation is limited by a velocity constraint within the entire operation, angular velocity should not exceed the maximum allowable value as constrained by the application itself. If the operation is limited by a velocity constraint, within the entire operation, angular velocity should not exceed the maximum allowable value as constrained by

$$|\dot{r}(t)| \leq \dot{r}_{max}; \qquad \forall t, \tag{2.2}$$

where $\dot{r}(t)$ and $\dot{r}_{max}$ represent the velocity at time t and the maximum velocity of objective trajectory respectively.

Besides, the vibration and oscillations persisted in the actual tracking profile of belt drives should be brought down to an acceptable level, though a limit for it does not consider quantitatively.

2.2.2 Trajectory planning algorithm and overview of compensation

The trajectory goes from initial position to final position with initial and final velocities zero, under speed and acceleration bounds as specified in (2.1) and (2.2). In minimum time trajectories a trapezoidal velocity profile is assigned and it imposes maximum constant acceleration in start phase, cruise velocity in middle phase followed by maximum constant deceleration phase at final stage. Intuitively, this strategy is comparable to flooring the accelerator, then coasting at the speed limit and finally slamming on brakes. Acceleration profile giving rise to such kind of trapezoidal velocity profile can mathematically given by,

$$A = \begin{cases} A_{max} & \dot{r}(t) \leq \dot{r}_{max} \text{ } and \text{ } \ddot{r}(t) > 0 \\ 0 & \dot{r}(t) = \dot{r}_{max} \\ -A_{max} & \dot{r}(t) \leq \dot{r}_{max} \text{ } and \text{ } \ddot{r}(t) < 0 \end{cases} \tag{2.3}$$

Figure 2.3 represents time minimal trapezoidal velocity profile as described above

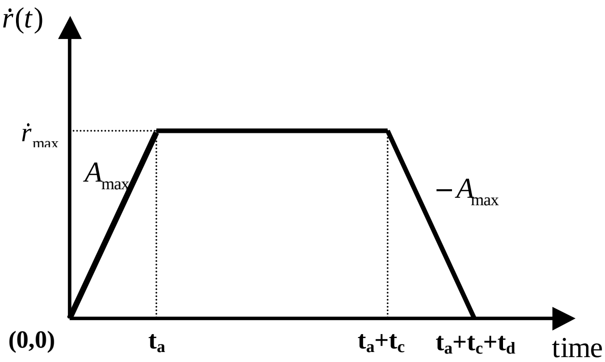

Figure 2.3: Objective Velocity Profile for Belt Driven Machine Control

and time intervals $[0, t_a)$, $[t_a, t_a + t_c)$, $[t_a + t_c, t_a + t_c + t_d)$ represent acceleration phase, cruise velocity phase and deceleration phase.

The proposed algorithm for trajectory planning and trajectory compensation can be illustrated concisely with the flowchart given in Fig. 2.4. Trajectory compensation for delay dynamics and vibration is achieved by means of a modified taught data technique [15] based on a combination of inverse dynamics and desired dynamic filter.

The angular position of the objective trajectory under maximum velocity and

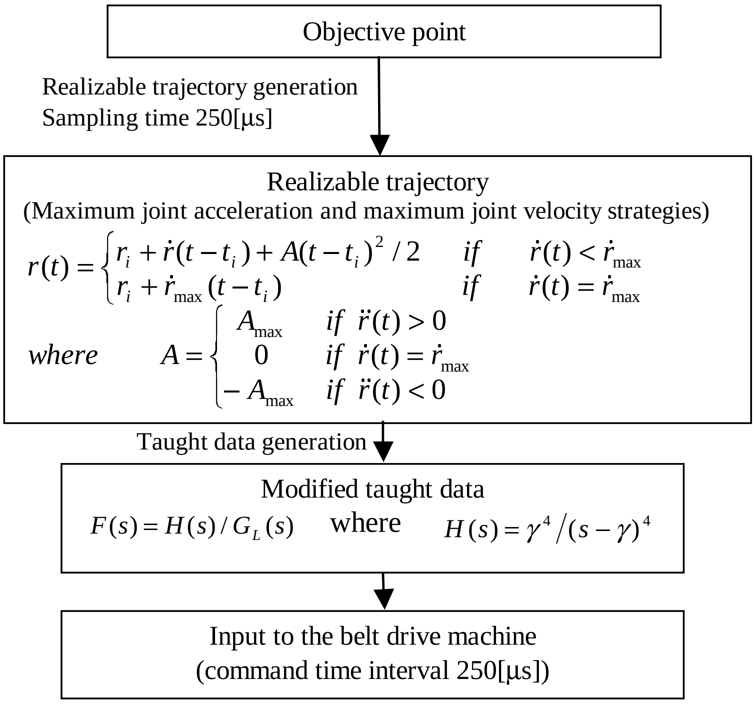

Figure 2.4: Trajectory Generation Criterion for Trapezoidal Velocity Profile

maximum acceleration strategy is governed by

$$r(t) = \begin{cases} r_i + \dot{r}(t-t_i) + A(t-t_i)^2 & \dot{r}(t) \leq \dot{r}_{max} \\ r_i + \dot{r}_{max}(t-t_i) & \dot{r}(t) = \dot{r}_{max} \end{cases} \tag{2.4}$$

On the basis of the above formation, the trajectory-planning algorithm generates a time sequence of joint variables that determines the motor position over time in respect of the imposed constraints. Since the servo controller is of zero order hold, not all continuous timely positions are important but interspaced at sampling time. Therefore, the positions of joints are discretized in time domain with sampling time T for servoing purposes and t takes the discrete values specified by

$$t = iT \quad i = 1, 2, 3, ...N \tag{2.5}$$

where NT is the total time of operation.

2.3 Spiky Phenomenon in Velocity Profile of Belt Drives

A significant spiky phenomenon in motor's velocity profile is evident in the experimental results given in Fig. 2.5. However, well established first order and second order kinematic models of servo system are incapable of characterizing the spiky phenomenon in velocity profile. A non-trivial belt reaction torque gives rise to this spiky phenomenon in velocity profile. The gear ratio of motor to load is 1:1 and it affects the belt reaction torque of becoming significant in two senses,

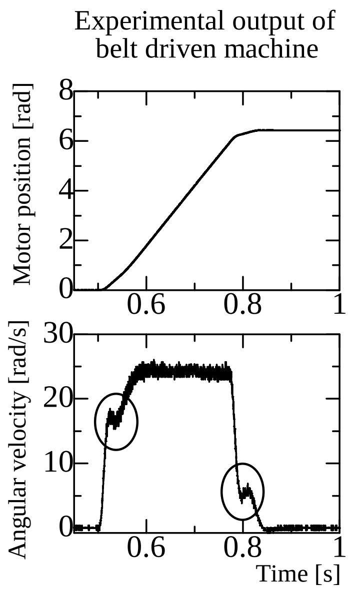

Figure 2.5: Spiky Phenomenon in Velocity Profile of Belt Driven Machine

1. No gear ratio scaling of the inertia torques of the load due to sudden change in acceleration
2. High-speed manipulation of the load associated with higher momentum and in turn, it creates high inertial load torques especially under minimum time operation, as the acceleration is rapid.

Significant reaction torque of the load definitely deviate the following trajectory from objective trajectory and leads to poor tracking as well as inaccurate positioning. Flexible dynamics may further intensify the inaccuracies and therefore proper compensation technique with careful consideration of belt reaction becomes mandatory.

In the experiment, motor angle throughout the operation and the load angle in the vicinity of final position were under investigation on the following grounds.

1. Motor angle encompasses the dynamics of the load and also much sensitive to servo dynamics. Therefore motor position based model validation is much more effective on the contrary to conventional approaches based on load position, as flexible dynamics assimilate sensitive dynamics.
2. Since the final positioning is of utmost importance in case of a position control system, the limited sensor range of the accurate laser sensor was utilized to obtain the exact load position near the final position.

2.4 Proposed Model and Solution Strategy for Belt Driven Machine

2.4.1 Rationale

Though the position accuracy is quite high in PID control, the tracking accuracy is often rather poor, since there is no direct compensation for friction and inertial forces.

In addition, as flexible belt driven mechanism possesses flexible characteristics, system has two degrees of freedom but only one control input, the angular position of the motor. Therefore the input to the system should be compensated for flexible dynamics and delay dynamics as PID controller inherently undergoes tracking error. However, sufficiently acceptable performance could be realized with belt drives even without dynamic compensation when they confined to very low speed operations. In order to respond fast changing sequences of input trajectory with minimum tracking error and restraint vibration, dynamic compensation is essential and crucial.

This is accomplished with a feed forward compensator based on the principle of concatenating the inverse dynamics of belt driven machine and desired dynamic filter. As the key underlying objective of the proposed method is to deploy belt drives in high-speed operations and thereby extend the bounds of application scope, encapsulation of belt reaction torque. The root cause for inaccuracies in inverse dynamics part of the compensator is indispensable as pointed out in **Chapter 2.3**.

2.4.2 Model construction

Load reaction torque does not incorporate within the first and second order kinematics models of servo systems and assumed to be negligible. However, this effect is non-trivial in high-speed belt drives due to 1:1 gear ratio and high-speed motion. Integration of belt reaction torque and consideration of flexible dynamics are the fundamental concerns in the development of an accurate model for servo controlled belt driven joints valid under high-speed operations.

The derivation of the model excogitating the flexible dynamics of belt drives are carried out under following three most applicable and practical assumptions.

1. The inertia and the friction of tension pulleys are negligible,
2. The mass of the belt is negligible and the belt has insignificant bending rigidity, and,
3. Belt drive operates within the linear elastic range of the belt.

The torques experienced by the motor pulley is the effective torque generated by the motor, motor initial torque, and the reaction torque of the belt. Under the torques stated above, the motor pulley attains its equilibrium. The effective torque exerted by the servomotor is equal to the torque generated by the motor due to the servoing action less the reaction torque due to back emf. Therefore, the effective motor torque, τ_M is given by,

$$\tau_M = K_p K_v^g (u - \theta_M) - K_v^g \dot{\theta}_M \tag{2.6}$$

where u, K_p, K_v^g and θ_M represent the input to the servo system, the position loop gain, the velocity gain of the servo amplifier and the position of the motor, respectively.

The inertia torque on the motor due to the mass of the rotational part of the motor and coupled pulley τ_I is expressed by $\tau_I = J_M \ddot{\theta}_M$, where J_M is the moment of inertia of the rotor including the motor pulley. When the motor pulley rotates in the direction indicated in Fig. 2.6, the upper belt segment increases its tension whereas the lower segment reduces its tension by equal amount due to the differential angular

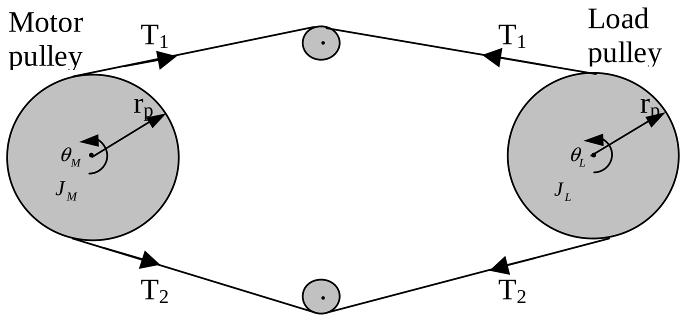

Figure 2.6: Flexible Structure of Belt Drive

motion of the motor pulley and load pulley. Hence the tangential effective force on either pulley, $T_1 - T_2$ is equal to twice the change in belt tension owing to motion

$$T_1 - T_2 = 2k_c r_p(\theta_M - \theta_L) \tag{2.7}$$

where k_c, r_p and θ_L represent the linear coefficient of belt drive elasticity, radius of either pulley and position of the load, respectively. Therefore the reaction torque of the belt on either pulley τ_R is described by

$$\tau_R = K_L(\theta_M - \theta_L) \tag{2.8}$$

where K_L represents the angular coefficient of elasticity of the belt. Considering the equilibrium of torques on the motor pulley, the governing relationship among the input u, motor position θ_M and load position θ_L in Laplace domain can be expressed by

$$K_p K_v^g U(s) = [J_M s^2 + K_v^g s + K_p K_v^g + K_L]\theta_M(s) - K_L\theta_L(s) \tag{2.9}$$

The load pulley is driven by the tension of the belt whereas the load experiences viscous damping torque and inertia torque, under which it achieves equilibrium. The equilibrium of the load pulley can be represented mathematically in s domain by

$$K_L\theta_M(s) = [J_L s^2 + D_L s + K_L]\theta_L(s) \tag{2.10}$$

where J_L and D_L are the load inertia and the viscous damping coefficient of the load, respectively. Combining the relationships stated in (2.9) and (2.10), it is possible to derive the transfer functions $G_M(s) = \theta_M(s)/U(s)$ and $G_L(s) = \theta_L(s)/U(s)$ as follows:

$$G_M(s) = \frac{b_2 s^2 + b_1 s + b_0}{a_4 s^4 + a_3 s^3 + a_2 s^2 + a_1 s + a_0} \tag{2.11}$$

$$
\begin{aligned}
b_0 &= K_P K_v^g K_L \\
b_1 &= K_P K_v^g D_L \\
b_2 &= K_P K_v^g J_L \\
a_0 &= K_P K_v^g K_L \\
a_1 &= K_L K_v^g + K_P K_v^g D_L + K_L D_L \\
a_2 &= K_L J_M + D_L K_v^g + K_P K_v^g J_L + K_L J_L \\
a_3 &= D_L J_M + K_v^g J_L \\
a_4 &= J_L J_M
\end{aligned}
$$

and

$$G_L(s) = \frac{a_0}{a_4 s^4 + a_3 s^3 + a_2 s^2 + a_1 s + a_0} \tag{2.12}$$

$$
\begin{aligned}
a_0 &= K_p K_v^g K_L \\
a_1 &= K_L K_v^g + K_p K_v^g D_L + K_L D_L \\
a_2 &= K_L J_M + D_L K_v^g + K_p K_v^g J_L + K_L J_L \\
a_3 &= D_L J_M + K_v^g J_L \\
a_4 &= J_L J_M
\end{aligned}
$$

Figure 2.7 concisely illustrates the derived in the form of block diagram and dynamics

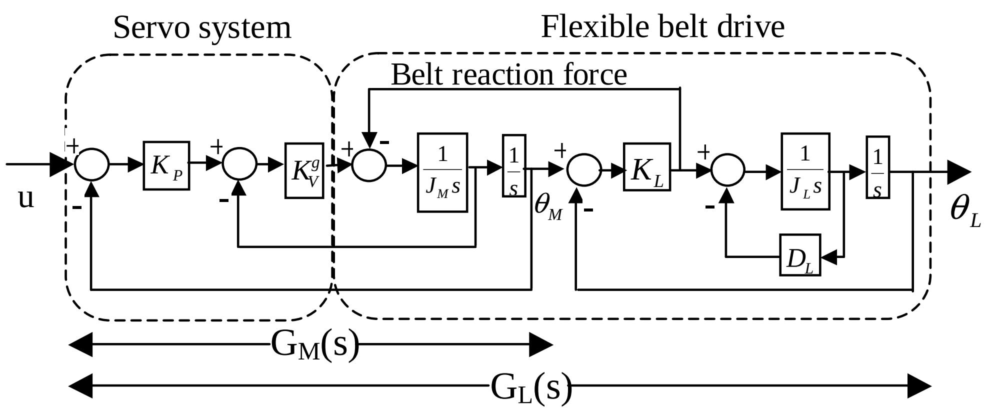

Figure 2.7: Fourth Order Model of Belt Driven Machine

associated with servo motor part and flexible structure part indicates separately.

2.4.3 Modified taught data technique

Goto *et al.* [15] is the proponent of the modified taught data technique to improve the tracking accuracy of the mechatronic servo system. Modified taught data technique is a feed forward compensating strategy to scale or reform the characteristics of planned trajectory well suited to the dynamic characteristics of the system and its concept is illustrated in Fig. 2.8. Every feed forward compensator is worked on the principle of pole assignment or pole zero cancellation in view of improving desired dynamics cum rejecting disturbances and always located at the extreme end of the trajectory planner. Since the taught data modifier, the dynamic compensator can

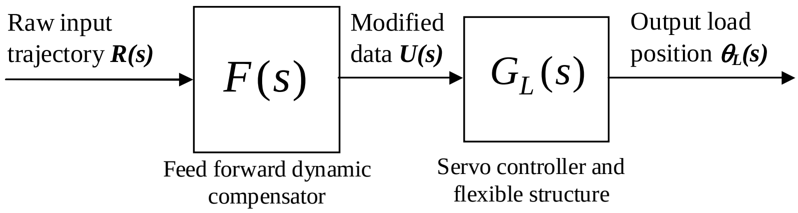

Figure 2.8: Concept of Modified Taught Data Technique

conveniently implemented inside the reference input generator neither any change to hardware setup nor a considerable reconfiguration of the system; this technique is readily welcome by the industry.

2.4.4 Design of Feed Forward Compensator

A detailed analysis of feed forward compensator $F(s)$ in Fig. 2.8 is furnished here. Proper selection of dynamic compensator can not only compensate delay dynamics, but also restrain vibration of the system of flexible structures. In general, selection of feed forward dynamic compensator is an objective selection and many methodologies provide different options. A combination of inverse dynamics and desired dynamic filter constitutes the proposed compensator as depicted in Fig. 2.9.

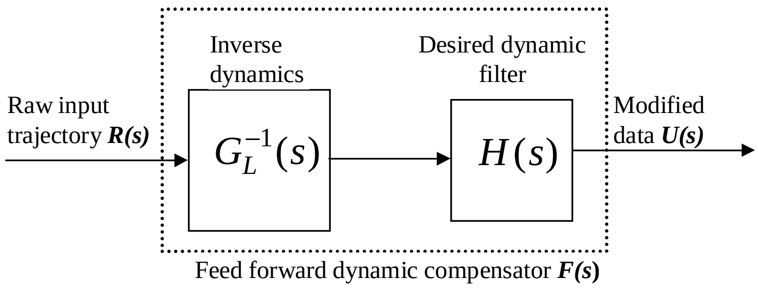

Figure 2.9: Dynamic Compensator for Data Modification

Dynamics of the feed forward compensator can be explained by

$$F(s) = \frac{H(s)}{G_L(s)} \tag{2.13}$$

where $H(s)$ is the desirable dynamic filter, whose dynamics is characterized by

$$H(s) = \frac{\gamma^4}{(s-\gamma)^4} \tag{2.14}$$

where γ is the location of four coincident poles.

The exact cancellation of the system dynamics with the inverse dynamics component of the feed-forward compensator eventually gives rise to attainment of output trajectory $R(s)H(s)$, which is a dynamic filtered version of the objective trajectory.

The characteristics of the desired dynamic filter are directly attributed in the realized trajectory. Therefore, the desired dynamic filter, $H(s)$ should be constituted to impart desired dynamic features such as oscillation free response with optimum settling time.

The numerator of $1/G_L(s)$ is a fourth order polynomial whereas the denominator is of zeroth order. Therefore, four zeros and no poles are introduced to the transfer function $H(s)/G_L(s)$ due to inverse dynamic component $1/G_L(s)$. As the inverse dynamic component of the feed-forward compensator has four zeros, the fourth or higher order dynamic filter, $H(s)$ should be implemented to diminish the effect of the differentiation operation in dynamic compensation. Failure in integrating such partial design of the dynamic filter into the dynamic compensator would result in the appearance of very fast sequences seriously leading to torque saturation of the motor and ultimately obstructing smooth functioning of the system. Reduction in sampling time is a favorable phenomenon to support smooth and vibration free controlling of the belt driven system, however, it itself adversely fetters the smooth performance in instances where there is poor design in the dynamic filter.

Locating the poles of dynamic filter (γ) on the negative real axis causes to realize oscillation free response whereas the coincident of poles optimizes the settling time. By locating the poles, γ further away from the imaginary axis dynamic response can be made fast but increasing the magnitude of the poles unnecessarily will result in generating fast sequences in time domain for servoing subsequently affects for the current saturation of servo amplifier and thus deteriorates the overall performance. However in general, there is no way to know a priory the best pole location [16].

2.4.5 Analytical solutions

As an alternative and much realistic technique for numerical simulation based on recursive iteration (accumulation of numerical error over iteration leads to non-convergent results and impair the accuracy dramatically) system dynamics was analytically solved for the taught data input and obtained the exact solutions for the joint position to conform the numerical robustness of simulation.

The reference input $u(t)$, used for servoing is a time-based input with zero order hold sequence having variable marginal magnitudes. It can be decomposed into a summation of time-shifted step input series with variable step size. Therefore, the modified taught data can be written as:

$$u(t) = \sum_{i=1}^{k} \Delta u(iT) w[(k-i)T] \tag{2.15}$$

where $\Delta u(iT) = u(iT) - u((i-1)T)$, $kT \le t \le (k+1)T$, T is the sampling time and w is the unit step function. If the system is excited with $u(t)$, motor angular position $\theta_M(t)$ and the load position coordinates $\theta_L(s)$ could be expressed with

$$\theta_M(t) = \sum_{i=1}^{k} \Delta u(iT) \{1 + A_1 e^{-\alpha_1 (t-iT)} + A_2 e^{-\alpha_2 (t-iT)} \tag{2.16}$$

$$+A_3 e^{-\xi\omega_L(t-iT)}\sin(\omega_L\sqrt{1-\xi^2}(t-iT)+\phi_1\}$$
$$\text{where}\ \ \mathrm{kT} \le \mathrm{t} \le (\mathrm{k}+1)\mathrm{T}\ \ \text{and}$$
$$\begin{aligned}
&g_1 = a_3/4a_4\\
&g_2 = (2a_2^3 - 9a_1a_2a_3 + 27a_0a_3^2 + 27a_1^2a_4 - 72a_0a_2a_4)^2\\
&l = \sqrt{-4(a_2^2 - 3a_1a_3 + 12a_0a_4)^3 + g_2}\\
&m = 2a_2^3 - 9a_1a_2a_3 + 27a_0a_3^2 + 27a_1^2a_4 - 72a_0a_2a_4\\
&c = \sqrt[3]{(m+l)}\\
&n = \sqrt[3]{2}(a_2^2 - 3a_1a_3 + 12a_0a_4)/3a_4c\\
&p_1 = \sqrt{\tfrac{a_3^2}{4a_4^2} - \tfrac{2a_2}{3a_4} + n + \tfrac{c}{\sqrt[3]{2}3a_4}}\\
&p_2 = -\tfrac{a_3^3}{a_4^3} + \tfrac{4a_2a_3}{a_4^2} - \tfrac{8a_1}{a_4}\\
&q_1 = \sqrt{\tfrac{a_3^2}{2a_4^2} - \tfrac{2a_2}{3a_4} - n - \tfrac{c}{\sqrt[3]{2}3a_4} - \tfrac{p_2}{4p_1}}\\
&q_2 = \sqrt{\tfrac{a_3^2}{2a_4^2} - \tfrac{2a_2}{3a_4} - n - \tfrac{c}{\sqrt[3]{2}3a_4} + \tfrac{p_2}{4p_1}}\\
&\alpha_1 = -g_1 - 0.5p_1 - 0.5q_1\\
&\alpha_2 = -g_1 - 0.5p_1 + 0.5q_1\\
&\omega_L = \sqrt{g_1^2 - g_1p_1 + 0.25(g_1^2 + q_2^2)}\\
&\xi = \tfrac{(g_1 - 0.5p_1)}{\sqrt{g_1^2 - g_1p_1 + 0.25(g_1^2+q_2^2)}}\\
&A_1 = \tfrac{b_2\alpha_1 + b_1\alpha_1 + b_0}{(\alpha_1-\alpha_2)(\alpha_1^2 + 2\xi\omega_L\alpha_1 + \omega_L)}\\
&B_2 = \tfrac{b_2\alpha_2 + b_1\alpha_2 + b_0}{(\alpha_2-\alpha_1)(\alpha_2^2 + 2\xi\omega_L\alpha_2 + \omega_L)}\\
&C = -(A_1 + A_2)\\
&D = \tfrac{b_0}{\alpha_1\alpha_2} + (\tfrac{A_1}{\alpha_1} + \tfrac{A_2}{\alpha_2})\omega_L^2\\
&B_3 = \sqrt{E^2 + \tfrac{(L - E\xi\omega_L)^2}{\omega_L^2}}\\
&\phi_1 = tan^{-1}\{\tfrac{C\sqrt{1-\xi^2}\omega_L}{(D - C\xi\omega_L)}\}
\end{aligned}$$

and

$$\theta_L(t) = \sum_{i=1}^{k} \Delta u(iT)\{1 + B_1 e^{-\alpha_1(t-iT)} + B_2 e^{-\alpha_2(t-iT)} \tag{2.17}$$
$$+B_3 e^{-\xi\omega_L(t-iT)}\sin(\omega_L\sqrt{1-\xi^2}(t-iT)+\phi_2\}$$
$$\text{where}\ \ \mathrm{kT} \le \mathrm{t} \le (\mathrm{k}+1)\mathrm{T}\ \ \text{and}$$
$$\begin{aligned}
&B_1 = \tfrac{a_0}{(\alpha_1-\alpha_2)(\alpha_1^2 + 2\xi\omega_L\alpha_1 + \omega_L)}\\
&B_2 = \tfrac{a_0}{(\alpha_2-\alpha_1)(\alpha_2^2 + 2\xi\omega_L\alpha_2 + \omega_L)}\\
&E = -(B_1 + B_2)\\
&L = \tfrac{\alpha_0}{\alpha_1\alpha_2} + (\tfrac{B_1}{\alpha_1} + \tfrac{B_2}{\alpha_2})\omega_L^2\\
&B_3 = \sqrt{E^2 + \tfrac{(L - E\xi\omega_L)^2}{\omega_L^2}}\\
&\phi_2 = \tan^{-1}\{\tfrac{\mathrm{E}\sqrt{1-\xi^2}\omega_\mathrm{L}}{(\mathrm{L} - \mathrm{E}\xi\omega_\mathrm{L})}\}
\end{aligned}$$

Table 2.1: Parameter Values of Belt Driven Machine

Description of parameter	Symbol	Value
Position loop gain	K_p	15 [1/s]
Servo amplifier velocity gain constant	K_v^g	0.0023[kgm^2/s]
Angular spring constant of belt	K_L	0.0513[kN/rad]
Viscous damping coefficient of load	D_L	5.24X10^{-4}[Ns/rad]
Location of poles	γ	-130
Maximum velocity	$\dot{r}_{\max}$	24[rad/s]
Maximum acceleration	$\ddot{r}_{\max}$	4314[rad/s^2]

2.5 Performance and Evaluation

The trajectory is generated with proposed planner, subsequently compensated for flexible dynamics and then used for real time servoing of actual belt driven machine. Further, a simulation study has been carried using conventional numerical techniques and also proved the validity of it using the analytical solutions derived. The parameters used for trajectory generation and simulation could be given in the Table.2.1

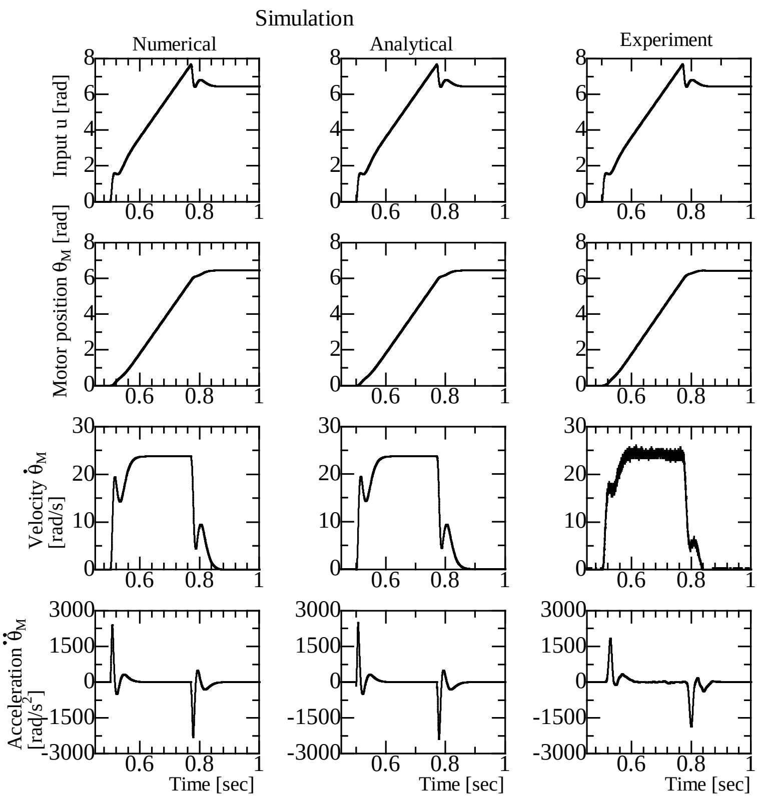

Figure 2.10: Simulation and Experiment Results of Belt Driven Machine

Figure 2.10 shows the results obtained for motor position by experiment as well as simulation based on analytical solutions and recursive techniques. By numerical differentiation corresponding velocities and acceleration were derived. In order to eliminate noises in the acceleration profile of the experiment, acceleration data is obtained by filtering the original data with a fourth order Chebyshev IIR low pass filter of 100Hz.

A significant spike could be noticed in the motor velocity profile of the proposed model at the times of changing the acceleration and it is expectable with belt reaction torque, as sudden changes in load acceleration has the consequences of rapid change in load inertia torque. The magnitude of the spike, the quantification aspect of the motor velocity spike of the proposed model is almost in accordance with the results obtained in experiment and hence proven the validity of the model.

Mathematically, the presence of two zeros in the transfer function (in equation (2.11)) causes the linear combination of first and second derivatives of the input to directly contribute to generate the output giving rise to higher sensitivity to input dynamics. Therefore, zeros in the transfer function assist to represent input dynamics dominantly, especially in the instance of a sudden change in acceleration profile with a spike in velocity profile.

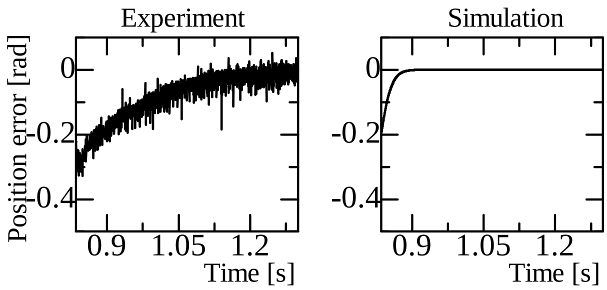

Figure 2.11: Comparison of Load Position Error

A comparison of the load position error profiles are provided in Fig. 2.11 and here load position error is defined as the final desired position subtracted from the current position of the load. The experimental load position error in Fig. 2.11 contaminated with noises due to quantization error of the data acquisition and noise of the laser sensor. No oscillations could be observed in load positioning and thereby the achievement of the vibration restraint with the proposed method has been verified.

The final objective, accurate positioning of the load has been investigated with load position error. However, the sensor range is limited to the vicinity of the final position and hence the experiment results may not be permitted to cover the entire tracking. For the sake of completeness, simulation results of the load position are furnished in Fig. 2.12.

2.6 Concluding Remarks

Numerous applications of position control devices based on servo driven belt drives are ubiquitous in industry. Belt drives are effectively employed as a power transmission

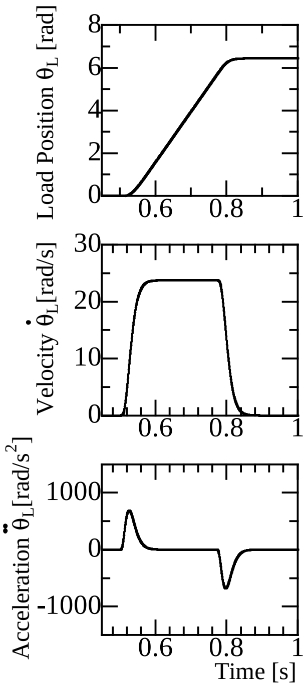

Figure 2.12: Simulation Results of Load Tracking

technique in light weight robots as belt drives enable to reduce the inertia of the robot arm through mounting the servo motors on stationary base. Linear positioning table is another dominant application of belt drives in industry.

Development of an accurate model representing precise dynamics of flexible belt drives, planning of trajectory under maximum acceleration and velocity constraints, designing of a feed forward compensator to compensate flexible dynamics and restraint vibration are basically covered in this chapter. There is a close similarity exist between the derived model for belt drives and the forth order kinematic model for robotic systems given in [70]. The modified taught data method, on which the proposed dynamic compensation is basically developed, has a quite good robustness and hence the proposed control method also inherits robustness characteristics over servo parameter changes. Further, avoidance of oscillations and restraint of vibration would be another key advantageous properties inherited from modified taught data technique, due to proper selection of desired dynamic filter.

Considering the practical perspectives, elimination of vibration and oscillations fundamentally contributed to speed up the operation by diminishing inter-operation settling time and deterioration of product quality due to frictional effects under vibrations especially in wafer transfer applications. In addition proposed technique allows high-speed operation intact with the accuracy of positioning and thereby improve the productivity through contracted cycle time. Using an off-line algorithm to plan the trajectory and associated compensation, the proposed method has strong industrial implications since it can be directly applied to existing belt driven servo systems without any hardware change.

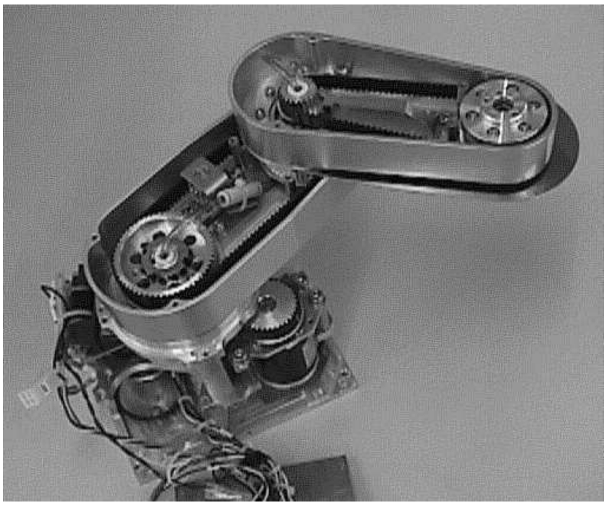

Figure 2.13: Multi-axis Belt Driven Manipulator

Figure 2.13 illustrates a typical multi-axis lightweight robot manipulator equipped with belt drives. The proposed control technique can be easily and conveniently extended to control such multi-axis multi-dimension robot systems with decoupled servoing of each joint. However, change in load inertia with configuration is formidable particularly under high-speed operation of robots. As the load to motor inertia mismatch increases, the system is heavily vulnerable for oscillations with increased settling time as a result of changed in natural frequency of the system, in addition to the power loss due to inertia mismatch [71]. Additional power dissipation dramatically increases as the mismatch gradually increases. These are few of open issues available for further investigation in multi axis belt driven robot system.

Chapter 3

Dual Arm Trajectory Planning for a Specified Cooperative Trajectory

3.1 Cooperative Control

3.1.1 Definition and categorization of cooperative control

Cooperative control is defined as the controlling of multiple dynamic entities that share a common though perhaps not singular objective [72]. Cao *et al.* [73] defined the cooperative control as given some task specified by a designer, a multiple robot system displays cooperative behavior if due to some underlying mechanism (i.e. mechanism of cooperation) there is an increase in total utility of the system. In both definitions, integration of individual robot motions directs towards the accomplishment of a specified cooperative task or tasks under cooperative control.

The core concept of cooperative control encapsulates two issues: pure cooperation where the collaboration of system entities through communication, and pure coordination where collaboration is achieved through a set of complementary rules without the benefit of communication [74]. It has been practically shown that a combination of these two concepts is more effective to optimize the time and the spatial efficiency. The following statement poses the importance of proper coordination for a cooperative task, and pure cooperation does not establish a sufficient support to achieve higher operational efficiency.
"A human being using two arms has a greater advantage over two human beings, each using one hand."

The fundamental motives of cooperative control can be stated as [75]:

1. To increase the scope of mission inherently distributed in space, time and functionality
2. To increase the reliability and robustness through redundancy
3. To decrease the task completion time through parallelism
4. To increase the bounds of payload through parallelism
5. To reduce the cost through much simpler individual robot design

According to the way of realization, cooperative control strategies are basically of two types [47]. They are:

a) Loose cooperation
The manipulation task is executed by controlling the two robots in an independent fashion, so that cooperative operation is realized only at task planning level. Therefore, the approach is simpler, could be off-line, can operate without any

sensory feedback and the operation is pretty good in structured environments.

b) Tight cooperation
The cooperation is realized not only at planning stage, but also at the control level. This can operate even under unstructured environment with a very little prior knowledge, since sensory feedback could guide the task at the operational level.

The following mechanisms can be found in the literature as the solution strategies to cooperative control.

i) Master-slave configuration:
Here the coordination of two robots is achieved through a force measurement. Further, independent controllers are implemented for operation. Therefore, it is convenient in implementation point of view. This mechanism is also applicable to improve the end effector accuracy through cooperative control. A considerable discrepancy of the capabilities of two robots may lead to a remarkable delay and in turn, it may have the danger of degrading the fidelity of master-slave systems.

ii) Hybrid position control:
when all the robots are considered to be an unique entity, the degrees of freedom of the total system proportionally increases with the number of robots. A unified robot approach may be assumed, and an object dynamic model was developed [52]. Motion and force control problem of multiple robots manipulating a cooperatively held common object is basically addressed under this mechanism.

Distributed impedance control strategy technique can be applied to above both solution strategies [76] in which an independent controller controls each robot. Force and position control actions are suitably dispatched to achieve both internal loading control and object position control.

3.1.2 Cooperative control research directions

Researches related to cooperative control were addressed in several research directions and found in number of research domains including cooperative control of mobile robots [73][77], dual robot coordinative control [67][69][78], and multirobot collaboration [79][80][81] for single or multiple goals. Multirobot collaboration for manipulating a commonly held object is addressed by many researchers [82][79] and popularly referred to as cooperative control. Dual arm cooperative control is intensively investigated for accuracy [83], vibration control [83], obstacle avoidance [84] and on many other aspects [85]. Further theoretical foundations have been developed for cooperative handling of objects [64][65].

A number of sophistications has been realized in multirobot systems with the aid of sensory implementation, as sensory feedback can make them much responsive, flexible, efficient and robust. Therefore, sensory driven control schemes become imperative for successful deployment of robots in most variety of tasks, particularly for cooperative tasks. However, inter-robot communication overhead and computational complexity due to additional sensory signal processing becomes key factors as well as implementation issues.

A fair amount of work has been directed towards the multirobot or dual robot manipulation of commonly held object or accomplishment of decomposed cooperative sub-tasks by individual robots. However, the accent on cooperative trajectory planning under strict coordination and fully integrated cooperation is rather weak. Additionally, speed up of the cooperative task through coupled cooperative planning strategies has not been exploited adequately.

3.1.3 Concept of cooperative control

The relative motion between the work piece and the tool constitutes a contour control motion in industrial applications. This is generally realized by keeping the work piece stationary and giving the required motion to the tool, which is usually held by the end effector of a robot. It is also possible to give simultaneous complementary conjugate movements to both work piece and tool such that the relative motion between the two shall be the objective contour. Such motion is called cooperative motion in this context and it could be definitely used to increase the speed of operation and thus to reduce cycle time, eventually the task completion time.

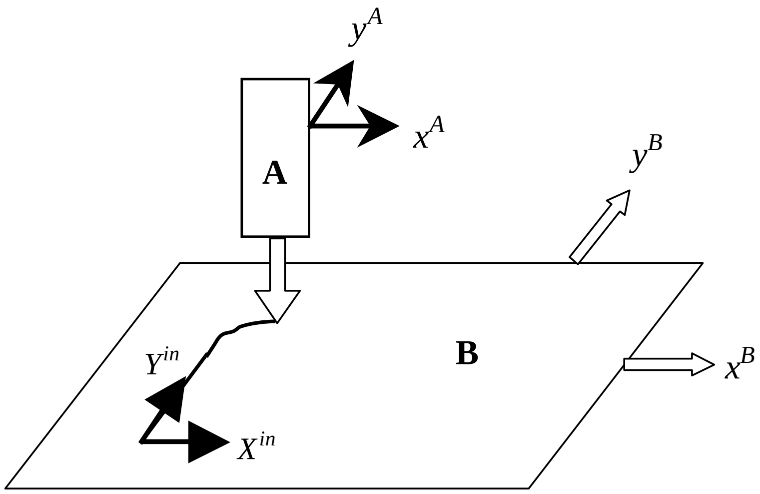

Figure 3.1: Concept of Cooperative Control

Figure 3.1 illustrates the concept of cooperative control with a simple example where tool A and work piece B represent a pen and a paper respectively. The motions given to pen and paper in X and Y directions are denoted by $[x^A, y^A]^T$ and $[x^B, y^B]^T$. The relative motion between the paper and pen $[X^{in}, Y^{in}]^T$ is the plotted shape on the paper refering to a fixed point on the paper.

In cooperative control of robots, the cooperative locus should be decomposed into complementary loci under dynamic and kinematic limitations. When a time optimal criterion is considered, it is not convenient to decompose the objective locus into conjugate loci (loci of A and B) and thereby plan the robot trajectories for each conjugate loci.

3.2 Preliminaries

3.2.1 System overview

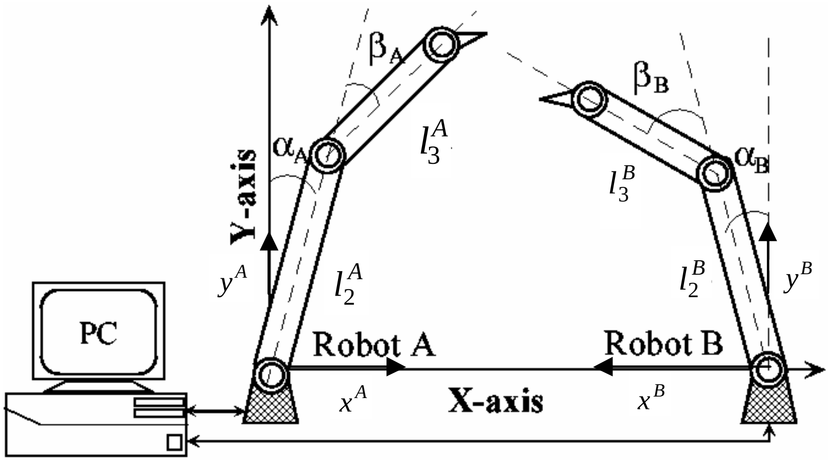

Figure 3.2: Experiment Setup Illustrating Two-Robot Manipulator Configuration

Figure 3.2 gives a comprehensive illustration of the orientation of two industrial manipulators, Performer MK2 and MK3, alone with the selection of coordinate axes. $[\alpha_A, \beta_A]^T$ and $[\alpha_B, \beta_B]^T$ denote the joint coordinate vectors of robot A and B respectively. Similarly, $[l_2^A, l_3^A]^T$ and $[l_2^B, l_3^B]^T$ represent the link length vectors of the two robots. Joint 1 of robot A from the base is selected as the origin of the common coordinate system referred to as "the global coordinate system". Two Cartesian coordinate systems are defined referring to robots A and B for convenient transformation of configuration space to working space and vise versa. As depicted in Fig. 3.2 the end effector positions in space are described with respect to these two coordinate systems as $x^A - y^A$ and $x^B - y^B$ referring to each robot.

3.2.2 Coordinate transformation

Kinematic and inverse kinematic transformations establish the relationship between the configuration space and the working space. Initial locations of end effectors in space determine the relative positions and their relationship established referring to the global coordinate system. The coordinate system transformation among the three Cartesian coordinate systems described above is a one-to-one relationship.

$$\begin{bmatrix} X^A & X^B \\ Y^A & Y^B \end{bmatrix} = \begin{bmatrix} x^A & -x^B \\ y^A & y^B \end{bmatrix} \begin{bmatrix} 0 & X_{BO} \\ 0 & 0 \end{bmatrix} \tag{3.1}$$

where $[X^A, Y^A]^T$ and $[X^B, Y^B]^T$ give the spatial representation of end effectors of the two robots referring to the global coordinate system. $(0, 0)$ and $(X_{BO}, 0)$ are the origins of the two Cartesian coordinate systems, $x^A - y^A$ and $x^B - y^B$, referring to the global coordinate systems.

The cooperative trajectory $[X^{in}, Y^{in}]$ can be expressed by

$$\begin{bmatrix} X^{in} \\ Y^{in} \end{bmatrix} = \begin{bmatrix} x^A \\ y^A \end{bmatrix} - \begin{bmatrix} -x^B \\ y^B \end{bmatrix} \tag{3.2}$$

The relationship between the cooperative velocity v^c and cooperative acceleration a^c could be represented with the equations given by

$$v^c = \sqrt{(\dot{X}^{in})^2 + (\dot{Y}^{in})^2} \tag{3.3}$$

and

$$a^c = \sqrt{(\ddot{X}^{in})^2 + (\ddot{Y}^{in})^2} \tag{3.4}$$

3.2.3 Significance of piecewise linear off-line trajectory planning

With the advent of digital electronics and microprocessor based electronics, digital controllers have identified as concise and precise means of controlling due to inherent flexibility ascribed by programmability. To cope with the digital world, discretization of analog signal in time domain is essential and that permits to apply convenient planning and simulation techniques. The time interval in which a digital controller takes an action is referred to as update rate and it may vary from 0.1 to 100ms depending on the robot controller and the application.

Further, consideration of entire path for trajectory planning is rather complicated and it is much sensible to break the path into segments, each of which could be handled by the trajectory planner easily particularly in cooperative trajectory planning. In the case of piecewise linear path, end effector would have to reduce the speed significantly at end points of each segment because it could not change the direction of velocity vector instantaneously. Therefore, consideration of perfect continuity, successive differentiability of trajectories at the merging points of such segments is of prime importance for high-speed operations since imperfections may lead to considerably deteriorate the trajectory being followed and to generate nontrivial jerks which would be resulted in shortening the life span of the robots. Blending line segments of the path by quadratic arcs and use of spline techniques are accepted techniques to achieve continuity at merging points.

Besides, off-line trajectory planning and on-line trajectory tracking are generally in industrial practice. However on-line trajectory planners are not readily welcome by the industry on the following grounds.

1. On line trajectory planners are relatively complex and demanding for super power reference input generators,
2. Proabability of risk and troublesomeness is quite high as compared to off-line planners and can not be conveniently implemented,
3. Most of the industrial tasks are carried out in structured environments and possible disturbances are a priori known. Hence the trajectory planning can be completely manageable with off-line planners,
4. Specifically, planning of accurate cooperative contours are highly computation intensive and therefore, on-line planning is not feasible under relatively higher

updating rates. Therefore, off-line planning becomes indispensable in cooperative trajectory planning.

3.3 Problem Statement

The relative motion between the end effectors of two robots generates the cooperative trajectory, when the two manipulators described in **Chapter 3.2.1** are maneuvered simultaneously. Therefore the cooperative trajectory should be resolved into two conjugate trajectories such that the relative trajectory between them exactly gives the cooperative trajectory.

Specifications of the path and the accuracy requirement referring to industrial tasks are generally expressed in working space. Besides, cooperative trajectory kinematic specification such as maximum cooperative speed is determined by the application itself. Under bounded cooperative speed, cooperative velocity profile should be of trapezoidal shape for a minimum time realization of cooperative task. In such trajectory, upper bound of the cooperative velocity can be specified by

$$|v^c| = v^c_{lim} \qquad \forall t \tag{3.5}$$

where v^c_{lim} is the maximum limit of cooperative velocity.

However, it is not pragmatic to realize the maximum cooperative velocity in zero time, and hence the maximum cooperative velocity should be reached under practically viable cooperative acceleration. This cooperative acceleration constraint can be mathematically expressed as

$$|a^c| = a^c_{lim} \qquad \forall t \tag{3.6}$$

where a^c_{lim} is the upper limit of cooperative acceleration.

The cooperative velocity profile and the objective locus are completely known as per the specification of the application and hence the cooperative trajectory, the relative motion between two end effectors is a specified priori.

Motion of robot arms are originated by the servomotors connected to the joints and hence the control action should be expressed in joint space. In most practical consequences, the maximum end effector velocity imposed by the application is not limited by the joint velocity limits of the servo motors even in the single robot case [29] and therefore the joint velocity limit governed by the hardware limitations is beyond consideration. However, the torque constraint imposed by hardware referring to the power amplifier current rating, inflicts a joint acceleration limit. In cooperative control of two robot arms, no joint of any robot can exceed the joint acceleration limit. Thus the bounded joint acceleration can be stated as

$$\begin{array}{rcl} |\ddot{\alpha}_A(t)| & \leq & (\ddot{\alpha}_A)_{lim} \\ |\ddot{\beta}_A(t)| & \leq & (\ddot{\beta}_A)_{lim} \\ |\ddot{\alpha}_B(t)| & \leq & (\ddot{\alpha}_B)_{lim} \\ |\ddot{\beta}_B(t)| & \leq & (\ddot{\beta}_B)_{lim} \end{array} \tag{3.7}$$

where $[\ddot{\alpha}_A(t), \ddot{\beta}_A(t), \ddot{\alpha}_B(t), \ddot{\beta}_B(t)]^T$ and $[(\ddot{\alpha}_A)_{lim}, (\ddot{\beta}_A)_{lim}, (\ddot{\alpha}_B)_{lim}, (\ddot{\beta}_B)_{lim}]^T$ denote the acceleration of the joints of two robots at time t and the limits of the acceleration of corresponding joints respectively.

3.4 Two Stage Trajectory-Planning Paradigm

3.4.1 Rationale

The planar example of cooperative control used to illustrate the concept of cooperative trajectory planning has four degrees of freedom (two joints contributed from each robot) to achieve two-dimensional cooperative spatial position and hence two-degrees of freedom are redundant. The redundant two degrees of freedom (DOF) give rise to generate many solutions when the cooperative position at the end of linear segment is solved for joint space vectors of each robot. As the piecewise linear trajectory is realized through sequential planning of linear line segments, the solution space grows exponentially in the form of a tree. Number of branches originated from a particular node is governed by the number of multiple solutions generated at that node and in turn the trajectory generation becomes computationally hard and numerically infeasible at high resolution approaches. Granularity of solutions determines the width of the solution space tree and the size of the segment or time used for segmentation determines the depth. A trade off exists between the accuracy of the generated trajectories and the computational complexity.

The key concept of cooperative trajectory generation is the decomposition of the cooperative task into complementary subtasks. Having completely specified the cooperative trajectory (time history of countering locus) based on application, there is no provision to optimize the operational time but to optimize the performance characteristics of each robot. An even task decomposition between two robots is selected as the criterion to optimize and defined an objective function Z such that

$$\mathrm{Max}\{\mathrm{Z}\} = \{|\max\{|\dot{\alpha}_{\mathrm{A}}|, |\dot{\beta}_{\mathrm{A}}|\} - \max\{|\dot{\alpha}_{\mathrm{B}}|, |\dot{\beta}_{\mathrm{B}}|\}|\}. \tag{3.8}$$

When optimizing the objective function Z, by bringing the maximum joint velocities of each robot as closer as possible, an excessive movement of either robot is minimized.

To facilitate the optimization criterion, a search algorithm should be devised over the solution space. However, yielding no solution for a particular planning segment give rise to terminate the propagation of solution tree structure at that node. The harder the acceleration constraint is, the higher the number of such terminations in propagation (branching to the next level) as well as the tree becomes lean in breadth wise. However the prediction of termination in branching can not be determined in advance through a look-ahead approach due to the nonlinearities of the space transformation. Nevertheless, a proper directive scheme can successfully guide for a breadth wise reduction of the state tree as well as to avert the termination of branching and thereby leading for a promising results within a remarkably lower computational time. Further, reduction in the number of segments compromising the accuracy of the contour followed can also be shrinked the depth of the tree. These provisions in scaling down the size of the solution space (reduce the number of nodes in state tree structure) enable to realize the solutions, even though the algorithm is NP hard.

3.4.2 Realization of cooperative control

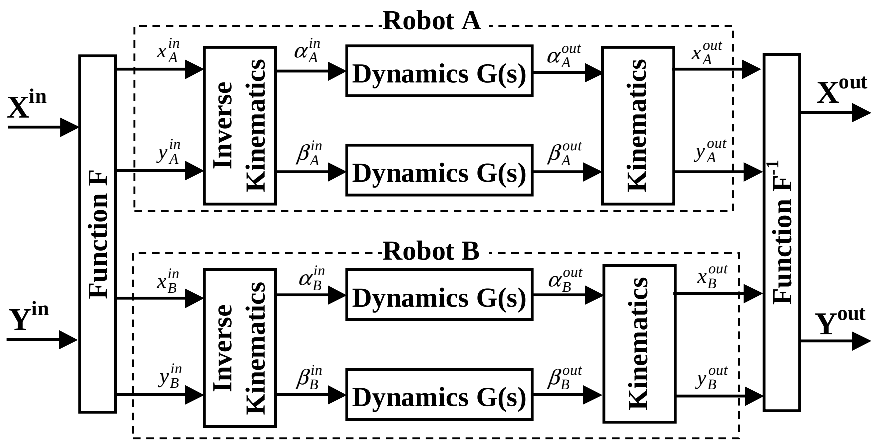

Figure 3.3: Realization of Cooperative Control

The strategy used in the realization of cooperative control is illustrated in Fig. 3.3. Initially, the cooperative locus is transformed into the cooperative trajectory considering the application-oriented specifications such as the maximum cooperative speed and some pragmatic concerns like cooperative acceleration. The input cooperative trajectory $[X^{in}, Y^{in}]^T$ specified in the Cartesian coordinates is decomposed into two Cartesian time sequences for robot A and robot B using the Cartesian trajectory decomposing function F. The generated complementary trajectories of robots A and B in Cartesian space are denoted by x_A^{in}, y_A^{in} and x_B^{in}, y_B^{in} respectively. The corresponding joint space transformations are given by $[\alpha_A^{in}, \beta_A^{in}]^T$ and $[\alpha_B^{in}, \beta_B^{in}]^T$. $[\alpha_A^{out}, \beta_A^{out}]^T$ and $[\alpha_B^{out}, \beta_B^{out}]^T$ represent the output positions obtained under dynamics G(s). Subsequently kinematic transformation is applied to convert them into working coordinates $[x_A^{out}, y_A^{out}]^T$ and $[x_B^{out}, y_B^{out}]^T$. The relative motion between the end effectors specified in Cartesian coordinate constitutes the output cooperative trajectory $[X^{out}, Y^{out}]^T$ in Cartesian space. The well-known second order linear kinematics model, $G(s)$ is used for simulation and it can be represented by

$$G(s) = \frac{K_p K_v}{s^2 + K_v s + K_p K_v} \tag{3.9}$$

where K_p and K_v are position and velocity loop gains of the servo system.

The Cartesian trajectory decomposition function F is more than a mathematical relationship due to imposed optimization criterion and the necessity to foresee the dynamic capabilities of each robot specified in joint space. An algorithm is developed to attain the function F and focuses the attention on it.

3.4.3 Trajectory generation criterion

i) Selection of segmentation time:

At first, the objective trajectory is segmented based on a time interval T such that the end of one segment is the starting point of the next segment. The time used for segmentation of the objective locus is termed as segmentation time and it is essentially an integer multiple of sampling time. As start and end points of each segment become the path points, the tracking profile is improved with lower segmentation time. However, a compromise between the accuracy and the computational overhead in trajectory planning determines the segmentation time.

ii) Generation of the global solution space and state tree:

Planning of complementary trajectories is carried out piecewise manner at segment level and sequentially to cover the entire cooperative trajectory. However, motion planning of two robots at each segment produces the feasible solution space comprising multiple solution candidates, and collection of such feasible solution candidates subsequently form the global solution space. The global solution space can be graphically represented in the form of a tree and it is referred to as state tree.

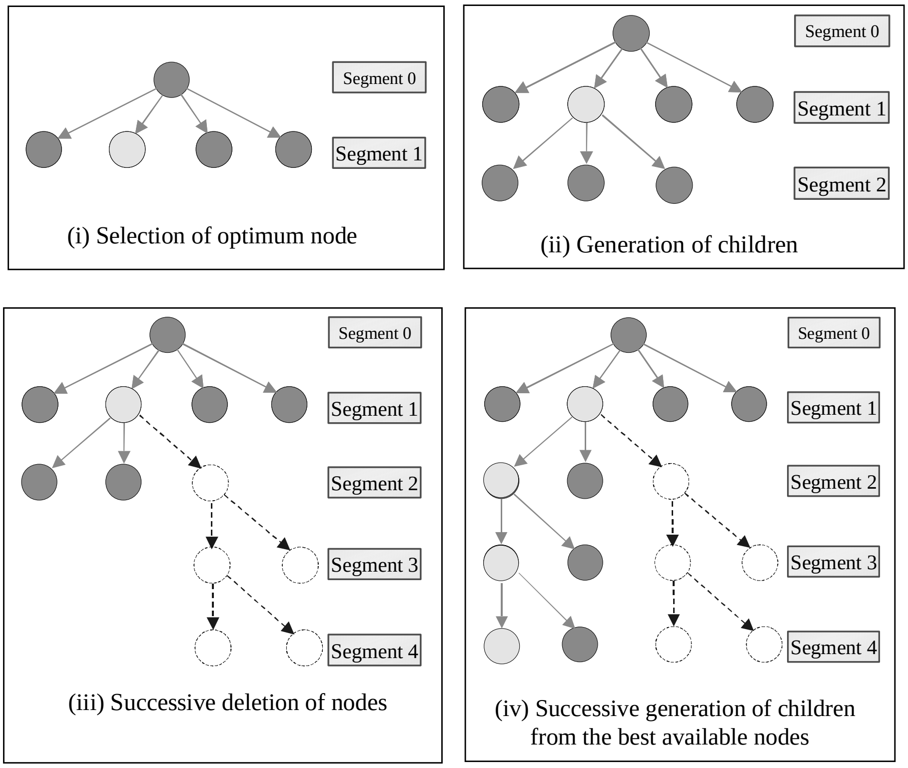

Figure 3.4: State Tree of Global Solution Space

Synopsis of the fundamental steps in developing the state tree is depicted in Fig. 3.4 and they can be stated as

a) Searching for optimum node
Based on the value of the objective function, the optimum node is opted among available nodes at a certain level of the state tree. The selected optimum node is shaded with a light grey in Fig. 3.4-(i).

b) Generation of children at optimum node
Generation of children is allowed only for the optimum node. Transition from one node to another node in next level indicates the advancement of trajectory planning through successive segments. The refined solution space would rather be used in the construction of state tree by appending the refined solutions to span the global solution space as referred in Fig. 3.4-(ii).

c) Successive deletion of nodes in absence of optimal nodes
The occurrence of no solution at the optimal node at a particular segment of the cooperative trajectory causes to delete that node and search the second optimal node at the same level. In case of failure to find such node compels to fold back one level up (move to parent level) in the tree-structured global solution space. The corresponding node at the parent level is also deleted in search of next optimal node for ensuing branching. Successive occurrence of this scenario is depicted in Fig. 3.4-(iii). The deleted nodes are represented with dotted circles whereas the discarded transition to these nodes is denoted with dotted arrows.

d) Successive generation of children from the best available nodes
Fig. 3.4-(iv) elucidates the successive generation of children for the best available optimal node without termination of spanning or without experiencing the occurrence of no solution condition.

iii) Two stage trajectory planning:
For servoing purpose, the time separation of trajectory should be at sampling time but the planning horizon for each segment can not be made to that scale because planning of the cooperative trajectory becomes exceedingly computation intensive. Therefore, planning of two complementary trajectories is realized at two modular stages, coarse level (knot level) or segment level planning (refer **Chapter 3.4.4** for details) and fine level refinement. The refinement of planning trajectories is carried out with an optimized interpolation scheme (details provided in **Chapter 3.4.7**) concerning the errors incorporated. This two-stage trajectory planning paradigm is equipped with short-listing criterion described in **Chapter 3.4.5**, to manage the complexity of calculation.

iv) Algorithm of trajectory planning:
The overall algorithm encompassing two stage trajectory planning can be elucidated with the steps stipulated below.

1. Generation of feasible solutions for a segment (generate feasible solution space)
2. Filter-out the feasible solution space for more probable candidate solutions using short-listing criterion (extraction of the refined solution space)
3. Span the state tree using the refined solution space
4. Selection of optimal node based on the value of the objective function

5. Repeat the above steps 1)∼4) until encountering no solution for a particular segment or covering all the segments of the objective trajectory
6. If no solution exists, delete the node in search of next optimum node among the available nodes at the same level of state tree repeatedly. In case of no such node available, recursively move to the parent level deleting the node, from which the last generation of children have been produced
7. Above steps 1)∼6) have to be reiterated until the completion of the entire cooperative trajectory. The solutions that cover the entire cooperative trajectory in segment wise, constitute the conjugate trajectories for robot A and B
8. Interpolation of joint positions for inter-segmentation time using fourth order, minimum acceleration interpolation scheme

The proposed trajectory-planning algorithm in its entirety can be precisely represented by the flow chart given in Fig. 3.5. This algorithm is capable of yielding complementary trajectories of cooperative robots within acceptable time duration, while optimizing the distribution of the cooperative task between two robots. The mathematical notion of the algorithm will be discussed in **Chapters 3.4.4, 3.4.5** and **3.4.7**.

3.4.4 Segment Level Trajectory Planning

The algorithm responsible for the generation of feasible solution, depicted by the block in between P and Q in Fig. 3.5, is explained in detail under this section. The detailed block diagram for feasible solution generation is illustrated by Fig. 3.6.

The relevant mathematical calculations of generating the feasible solution space for a particular segment of the cooperative trajectory are illustrated in the form of equations. Initially, joint velocities of robot A are selected as independent parameters to generate solutions according to

$$\begin{bmatrix} \dot{\alpha}_A(t_i) \\ \dot{\beta}_A(t_i) \end{bmatrix} = \begin{bmatrix} \dot{\alpha}_A(t_{i-1}) \\ \dot{\beta}_A(t_{i-1}) \end{bmatrix} + \begin{bmatrix} (\ddot{\alpha}_A)_{max} k_1 \\ (\ddot{\beta}_A)_{max} k_2 \end{bmatrix} T \tag{3.10}$$

where k_1 and k_2 are parameters within the interval [-1,1]. The complete continuum of joint velocities, which conforms the acceptable parameter values is beyond consideration, and hence a discrete set of equi-spaced joint velocities with an adequate resolution is selected. The next joint position is evaluated for each joint velocity using the Euler formula given by

$$\begin{bmatrix} \alpha_A(t_{i+1}) \\ \beta_A(t_{i+1}) \end{bmatrix} = \begin{bmatrix} \alpha_A(t_i) \\ \beta_A(t_i) \end{bmatrix} + \begin{bmatrix} \dot{\alpha}_A(t_i) \\ \dot{\beta}_A(t_i) \end{bmatrix} T \tag{3.11}$$

where T is the time interval used to segment the cooperative trajectory or segmentation time. The kinematic equations transform set of solutions in joint space into one in the working space as

$$\begin{bmatrix} x^A(t_{i+1}) \\ y^A(t_{i+1}) \end{bmatrix} = \begin{bmatrix} sin[\alpha_A(t_{i+1})] & sin[\alpha_A(t_{i+1}) + \beta_A(t_{i+1})] \\ cos[\alpha_A(t_{i+1})] & cos[\alpha_A(t_{i+1}) + \beta_A(t_{i+1})] \end{bmatrix} \begin{bmatrix} l_2^A \\ l_3^A \end{bmatrix} \tag{3.12}$$

Within each segment, the relative movement of two robots must ensure corresponding cooperative movement. In other words, planning horizon of relative displacement is confined to the segment size. Therefore the new Cartesian position of the robot B can be evaluated by the principle of complimentary motion and expressed by

$$\begin{bmatrix} X^{in}(t_{i+1}) - X^{in}(t_i) \\ Y^{in}(t_{i+1}) - Y^{in}(t_i) \end{bmatrix} = \begin{bmatrix} x^A(t_{i+1}) - x^A(t_i) \\ y^A(t_{i+1}) - y^A(t_i) \end{bmatrix} \begin{bmatrix} -x^B(t_{i+1}) + x^B(t_i) \\ y^B(t_{i+1}) - y^B(t_i) \end{bmatrix} \tag{3.13}$$

where $X^{in}(t_{i+1}) - X^{in}(t_i)$ and $Y^{in}(t_{i+1}) - Y^{in}(t_i)$ represent the abscissa and ordinate of the i^{th} segment of the cooperative trajectory. New yielding positions of robot B is converted into joint space applying inverse kinematics relationships given by

$$\begin{bmatrix} \alpha_B(t_{i+1}) \\ \beta_B(t_{i+1}) \end{bmatrix} = \begin{bmatrix} sin^{-1}\{\frac{x^B(t_{i+1})}{\sqrt{[x^B(t_{i+1})]^2+[y^B(t_{i+1})]^2}}\} - sin^{-1}\{\frac{l_3^B sin[\beta_B(t_{i+1})]}{\sqrt{[x^B(t_{i+1})]^2+[y^B(t_{i+1})]^2}}\} \\ cos^{-1}\{\frac{[x^B(t_{i+1})]^2+[y^B(t_{i+1})]^2-[l_2^B]^2-[l_3^B]^2}{2l_2^B l_3^B}\} \end{bmatrix} \tag{3.14}$$

The corresponding set of joint accelerations of robot B is calculated by the following equations given by

$$T^2 \begin{bmatrix} \ddot{\alpha}_B(t_i) \\ \ddot{\beta}_B(t_i) \end{bmatrix} = \begin{bmatrix} \alpha_B(t_{i+1}) \\ \beta_B(t_{i+1}) \end{bmatrix} - 2 \begin{bmatrix} \alpha_B(t_i) \\ \beta_B(t_i) \end{bmatrix} + \begin{bmatrix} \alpha_B(t_{i-1}) \\ \beta_B(t_{i-1}) \end{bmatrix} \tag{3.15}$$

If the magnitude of evaluated joint acceleration at either joint of robot B obeys the acceleration constraint, the solution is assumed to be a feasible solution. The set of feasible solutions conforming joint acceleration bounds constitutes the feasible solution space.

3.4.5 Short listing criterion

Generation of the feasible solution space corresponding to a particular segment at coarse level trajectory planning is an intermediate step of a sequential procedure since the final condition of one segment will be the initial condition of the subsequent segment. If an inappropriate solution candidate is selected from the feasible solution space at a particular stage, existence of no solution in a latter stage is inevitable. Therefore the propagation of state tree should be confined to the branches having lesser probability of termination at ensuing levels using a dynamic filtering criterion. Such criterion capable of filtering the candidate solutions in the feasible solution space to generate more probable candidates for trajectory planning is called short-listing criterion and the abstracted solution space derived with short listing criterion is termed as refined solution space. The short-listing criterion devised in trajectory planning can be mathematically expressed by

$${}^{d}a_j^i ={}^{m} a_j^i + a_{allowance} \tag{3.16}$$

where ${}^{d}a_j^i$, ${}^{m}a_j^i$ and $a_{allowance}$ represent the j^{th} joint acceleration of i^{th} segment in dual robot case, the j^{th} joint acceleration of i^{th} segment in single robot case, and the acceleration allowance respectively.

In order to avoid the unproductive time wasted in search of non-termination solution, a non-terminating optimal solution search is started with a non-terminating solution and it is the key principle behind this short-listing criterion. When the joint velocity vector of either robot is set to zero, solution of two-robot case degenerates to single robot case. Therefore, the unique solution corresponding to single-robot case is a special non-terminating solution of the solution space. It can be considered as the initial solution. As the allowance in short listing criterion is a parameter, boundaries of the refined solution space can be varied. Improper diminutive selection of allowance would lead to inexistence of non-terminating optimal solution as the generation of refined space is under too hard constraints. Besides, a more linear short-listing criterion is also of no use, as it does not serve the very intended objective.

3.4.6 Coarse to fine trajectory planning refinement

The reduction in path points, i.e. diminished number of levels in the status tree, intact with the required level of accuracy is an acceptable trade off, to resolve the time complexity issue to a manageable level. However, the calculation of missing intermediate points are mandatory for servoing, since off-line taught data should be inter-spaced at sampling time. Since rough jerky motion tends to increase wear on the mechanism and causes vibrations by exciting resonances in the manipulator, a smooth manipulator movement is significantly important. For achieving this, a smooth function, one which is continuous and continuous at first derivative is to be defined [6]. Therefore, spline based interpolation schemes are proved to be the most effective ways of calculating intermediate points as they achieve required level of continuity by appropriate selection of the order of fitted function, preferably a polynomial. However, spline techniques suffer from the following shortcomings,

1. As the order of polynomial increases, the oscillatory behavior increases and it may lead to trajectories which are not natural for manipulators,
2. Numerical accuracy of polynomial coefficients decreases as the order increases,
3. The resulting system of constraint equations is hard to be solved, and
4. Polynomial coefficients depend on the boundary conditions; if there is a change in a condition, all the coefficients of the polynomial have to be recalculated.

Here, separate spline schemes have been defined for each segment and every joint, to calculate the intermediate joint positions so that the end effector motions are smooth. In this case, joint positions and velocities calculated by segment level planning are used as boundary conditions. In addition, optimization of acceleration at refined stage is a key concern in defining a spline scheme for interpolation.

3.4.7 Optimal interpolation scheme

In making a single smooth motion at every segment considering one segment at a time, at least four constraints are evident for the interpolation of joint position. These four constraints are formulated in the trajectory planning at segment level. The joint status, i.e. joint positions and velocities, at the start and end points have already been determined by coarse level planning (two velocity constraints and two position constraints are imposed). In order to accommodate the above constraints, cubic or higher order polynomial should be selected for interpolation. However, the cubic polynomial is strictly deterministic by the constraints and hence there is no independent parameter to minimize the limits of acceleration. Therefore quartic polynomial is selected with one independent parameter to optimize, and it can be represented by

$$p_{s4}(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + \alpha t^4 \tag{3.17}$$

where $p_{s4}(t)$ is the fourth order polynomial fitted to interpolate the joint position, t is time, α is an independent parameter to minimize the bounds of acceleration and a_0, a_1, a_2, a_3 are coefficients of the polynomial. The first and second derivatives yield the velocity polynomial $v_{s4}(t)$ and acceleration polynomial $a_{s4}(t)$ given by

$$v_{s4}(t) = a_1 + 2a_2 t + 3a_3 t^2 + 4\alpha t^3 \tag{3.18}$$

and

$$a_{s4}(t) = 2a_2 + 6a_3 t + 12\alpha t^2 \tag{3.19}$$

At knot level (segment level) planning of trajectories, the relationship between the joint angle and joint velocity is established by Euler's forward formula. Hence the joint velocity, which is briefly illustrated with solid lines in Fig. 3.7 is a piecewise constant function for each joint in the trajectories planned at segment level. The polynomial used for interpolation $v_{s4}(t)$ is also represented with a dashed line. It could be easily seen in Fig. 3.7 that the true tangential acceleration after interpolation is inevitably greater than the apparent numerical acceleration (corresponding curve in velocity profile is denoted by a dotted line in Fig. 3.7) f, which is given by

$$f = (v_1 - v_0)/T \tag{3.20}$$

where v_0 and v_1 are the velocities at two consecutive path points referring to Fig. 3.7.

Under the velocity and position constraints determined at the start and end points of each segments by segment level trajectory planning, the polynomial coefficients a_0, a_1, a_2, and a_3 can be evaluated.

$a_0 = d_0 \quad a_1 = v_0$

$a_2 = \alpha T^2 - f \quad a_3 = f/T - 2\alpha T$

Therefore the acceleration polynomial can be expressed by

$$a_{s4}(t) = 2(\alpha T^2 - f) + 6(f/T - 2\alpha T)t + 12\alpha t^2 \tag{3.21}$$

where T and f are the segmentation time and the apparent numerical acceleration at segment level trajectory planning. In selecting the optimal value for α, it is required to minimize the maximum acceleration limit a_{max} subjected to

$$|a_{s4}(t)| \leq a_{max} \qquad \forall\ t \in [0, T] \tag{3.22}$$

The conventional approach of finding local maxima and minima by equating the partial derivatives to zero is not acceptable on the following grounds.

1. Global extrema for $t \in [0, T]$ is considered instead of local maxima and local minima
2. The extreme points are not considered to fix the upper and lower boundaries of acceleration

When dealing with global extrema, one possibility is to set the local extrema at $t = 0$ or at $t = T$. Then these local minima or maxima becomes global optima as in equation (3.21), which denotes a set of parabolic curves and therefore the first derivative of a_{s4} with respect to time t is a monotonously increasing or decreasing function depending on the sign of α. This condition satisfies at $\alpha = \pm\frac{f}{2T^2}$
Under either condition for α, $t = 0$ and $t = T$ correspond to global maxima and minima or vise versa. In order to meet the minimum upper and lower boundaries, the algebraic sum of global maximum and minimum should be equal to zero. This condition is fulfilled at

$$\alpha = -\frac{f}{2T^2} \tag{3.23}$$

At $\alpha = -f/2T^2$, both the occurrence of global optima at the end values of time t and the minimization of the acceleration values at the end values of time t are simultaneously satisfied. Therefore, $\alpha = -f/2T^2$ becomes the optimum selection for α. Under this optimum value for α, a_{s4} is limited to the range $\pm 3f$ since $a_{s4}(0) = -3f$ and $a_{s4}(T) = 3f$. The acceleration polynomial resulted from interpolation scheme is reduced to

$$a_{s4}(t) = -3f + 12ft/T - 6ft^2/T^2 \tag{3.24}$$

Hence the position interpolation polynomial yields to

$$p_{s4}(t) = d_0 + v_0 t - 3ft^2/2 + 2ft^3/T - ft^4/2T^2 \tag{3.25}$$

3.5 RT-Linux for Real Time Operation

For accurate and real time trajectory tracking, timely provision of appropriate reference joint position/velocity to servo controller must be ensured. Failure to feed reference input at correct time results in undesirable consequences jeopardizing the safe operation of the robot, and in extreme case it may possibly lead to catastrophic consequences. Correctness of real time system does not depends only on the logical result or correct reference input, but also the physical instances at which the reference inputs are produced. In other words, timeliness is a key characteristic in real time servoing and hence the program in charge of tracking must produce time critical outputs.

Control programs run on general-purpose operating systems may produce intolerable inaccuracies in time, particularly in fine-grained synchronization of program output with hardware operation (when using higher update rates of reference trajectory), since the timing of execution is not guaranteed due to inherent time-sharing execution. However by increasing the priority of the thread, at which the control program executes, can improve the current situation and can make the system a soft real time, but it is not a concrete solution to crack the fundamental issue of making the system hard real time to enable handling of interrupts in real time. A high priority thread cannot preempt a low priority thread during a system call and has to be delayed until either execution returns from system call to kernel decides that it is safe to reschedule (preemption point). Improving kernel preemptability is a viable option to manage this situation, still it may also associated with many pragmatic issues like difficulty in maintenance, non-guaranteed worse case response time, increase in response time due to new drivers. Therefore full-fledged real time operating system, RT-Linux has been selected as a platform for real time servoing, since minute errors are even intolerable.

RT-Linux is a patch to Linux which is basically developed to overcome the conflicts of RT constraints of Linux. RT-Linux possesses efficient time management capability and the ability to direct access of hardware. The basic architecture of the RT-Linux can be briefly given in Fig. 3.8.

RT-Linux creates a virtual machine interface (second kernel) located in between hardware and own operating system, and thereby Linux kernel becomes second kernel's idle thread. The second kernel, RT-Linux module, based on interrupt abstraction can process real time interrupts independent from what is happening under Linux. The FIFO is a communication mechanism based on FIFO of Linux but adapted-real on time. RT-Scheduler is a priority driven scheduler that uses Round Robin or FIFO algorithms. It also supports preemption and priority inheritance.

3.6 Performance and Evaluation

An S-shaped, example objective locus has been selected to evaluate the proposed planner for the cooperative trajectories for bi-arm configuration. The generated trajectory was tested for performance with two industrial articulated robot arms, Performer MK2 and MK3. During the experiment, the resultant joint positions are acquired from the inbuilt shaft encoders and they are logged onto a computer through a data acquisition hardware of reference input generator. Simulation study has been carried out and a comparison is made between experiment and simulation results. In addition, effectiveness of the cooperative control is elaborated with the corresponding results of single robot case. The parameters and the specifications used for simulation and experiment are tabulated in Table 3.1.

Figure 3.9 illustrates the input cooperative trajectory of two robots planned at segmentation level prior to interpolation or the refinement stage of the trajectory. The input trajectory depicted in Fig. 3.10 has the data separation at sampling time, $2ms$ and it is obtained by interpolation of the trajectory given in Fig. 3.9. The joint velocities and the accelerations are derived by successive numerical differentiation. The fine level input trajectories have been used for real time tracking of trajectory using PID controllers and for the simulation with the second order kinematic model given in equation (3.9).

The excitation of servo system with an input trajectory given in Fig. 3.10 yields the experimental results shown in Fig. 3.11. In order to alleviate the noise included

Table 3.1: Parameter Values of Cooperative Control of Two Articulated Robots

Description of parameter	Symbol	Value
Curvature of S-shaped locus	c	1[cm]
Maximum cooperative velocity	$v^c_{\max} / r_s$	3.5[rad/s]
Maximum cooperative acceleration	$a^c_{\max} / r_s$	8.0[rad/s^2]
Apparent numerical acceleration	f	0.3[rad/s^2]
Segmentation time	T	0.1[s]
Sampling time	T_s	2[ms]
Link lengths of robot A Performer Mark 2	l_2^A l_3^A	0.27[m] 0.23[m]
Link lengths of robot B Performer Mark 3	l_2^B l_3^B	0.25[m] 0.215[m]

with acceleration due to quantization errors of shaft encoder and numerical differentiation, acceleration data have been filtered with a fourth order Chebishew LPF having cut-off frequency 20Hz. The simulated output of two robots is obtained with the second order model and it is depicted in Fig. 3.12. The dotted line of the figure indicates the joint acceleration limit.

Kinematic transformation was applied on the joint positions, which were obtained from shaft encoders of robots in the experiment. Thereby, the Cartesian spatial description of the end effector of each robot has been calculated. Simple relative motion between the end effectors of two robots expressed in Cartesian coordinates constitutes the resultant cooperative trajectory and thus obtained trajectories along with the objective trajectory are given in Fig. 3.13. Here, the trajectory is expressed in working space and the first row of Fig. 3.13 shows the variation of cooperative position described in Cartesian coordinates with respect to time. The second row of the same figure illustrates the generated loci, by experiment and the objective locus itself in Cartesian space.

Minimum time contour controlling of the same locus has been planned for single robot under maximum joint acceleration constraints and maximum end effector velocity constraints for comparison purposes. The steps involved in planning of this trajectory are furnished in **Appendix E**. The results of the experiment with single robot case are illustrated in Fig. 3.14.

Simulations have been carried out under limited torque conditions and compared the simulation error profiles for single robot with multi robot case as given in Fig. 3.15. The improvements of cooperative control over single robot are numerically analyzed and tabulated in Table 3.2 in the sense of accuracy as well as in task completion time. The enhancement of criteria/parameters of interest resulted by using cooperative control is also figured out by its absolute value as well as a percentage.

3.7 Concluding Remarks

A novel cooperative trajectory planner based on kinematics, in view of enhancing the speed of operation through cooperative control is presented. The input cooperative

Table 3.2: Comparison of Results in Terms of Accuracy and Task Completion Time

Method	Criterion	Mono-robot	Cooperative control	Improvement
Minimum time trajectories	Task completion time	2.67 [s]	2.38 [s]	0.29 [s] (11.0%)
Equal time trajectories	Maximum error in X	6.1 [mm]	2.7 [mm]	3.4 [mm] (55.7%)
	Maximum error in Y	5.6 [mm]	1.3 [mm]	4.3 [mm] (76.8%)

trajectory with the maximum cooperative velocity and the acceleration, determined by the application itself, is decomposed into two complementary trajectories under maximum joint acceleration constraints. Further, the optimization aspects of trajectory planning algorithm mimics a fair task distribution to avoid over utilization of either robot.

Laser cutting, paint spaying and contour welding are few of the potential typical industrial applications of the proposed planner. These applications are poorly suited for human beings due to heat, danger and the toxic nature of these applications and hence deployment of robots becomes an exigency. Proposed advanced planner expands the dynamic limitations beyond the capacity of each robot and thereby achieves speedy accomplishment of the cooperative task under strict coordination. Because of being an off-line algorithm, the computational time does not impose any limitation and hence this method can be directly adapted to existing servo systems without any change in hardware or without any considerable reconfiguration of the system. Simplicity is another key impressive feature of the proposed planner.

Concerning the theoretical contribution of the proposed algorithm, two complexity management techniques (two stage planning and short listing criterion) without compromising the required accuracy were introduced. The proposed optimum interpolation technique is a more sophisticated alternative version for popular cubic spline method in generic point of view, as it deliberates on optimization aspect; the minimization of acceleration bounds.

Though the trajectory planner is illustrated with an S-shaped locus, it is versatile enough to accommodate any curvy or much complicated form. It can also be easily ported to environments having multiple degrees of freedom, but the time complexity exponentially increases with the number of redundant DOF of the system. The execution time of an off-line algorithm is not so critical, but it should not be exorbitantly detracting from the use or infeasible in practical sense. Therefore at higher redundant DOF, this algorithm may not be attractive and it is a serious limitation of the algorithm. Further, following aspects also impose restrictions for the scope of applications of the proposed planner.

1. Not only the path but also the timing information of the cooperative motion should be provided,
2. This type of strict coordinative planner is applicable to plan the tasks in structured environments, and

3. This does not always guarantee the optimum solution, but a sub-optimum solution.

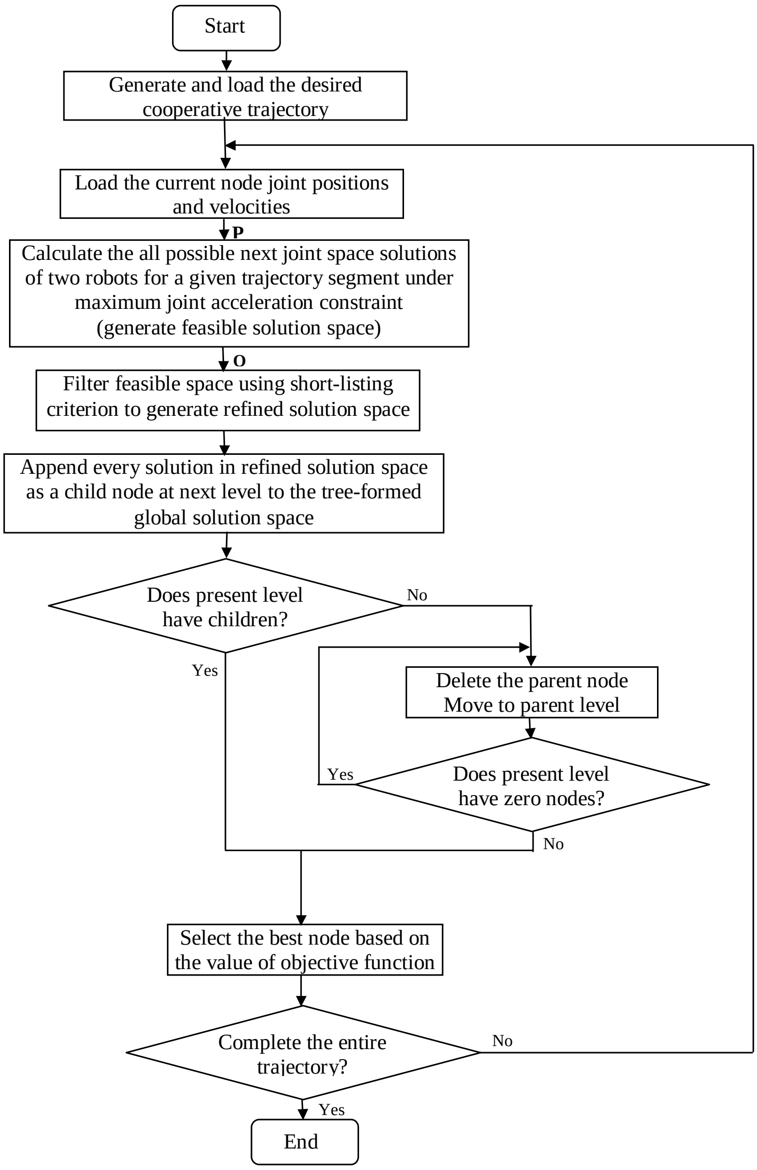

Figure 3.5: Entire Trajectory Generation Algorithm

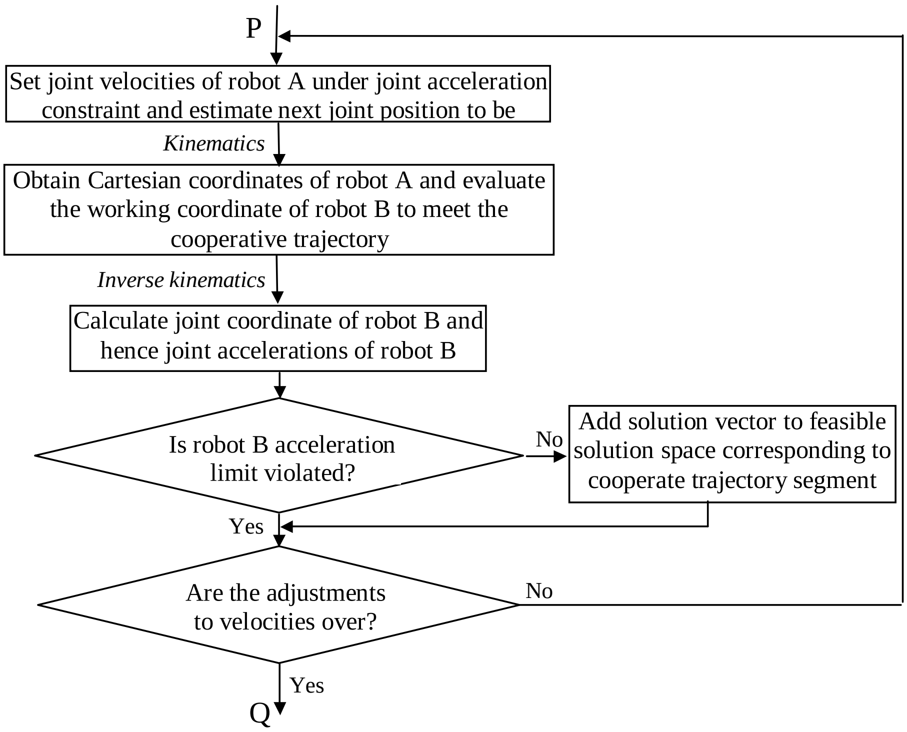

Figure 3.6: Algorithm for the Generation of Feasible Solution

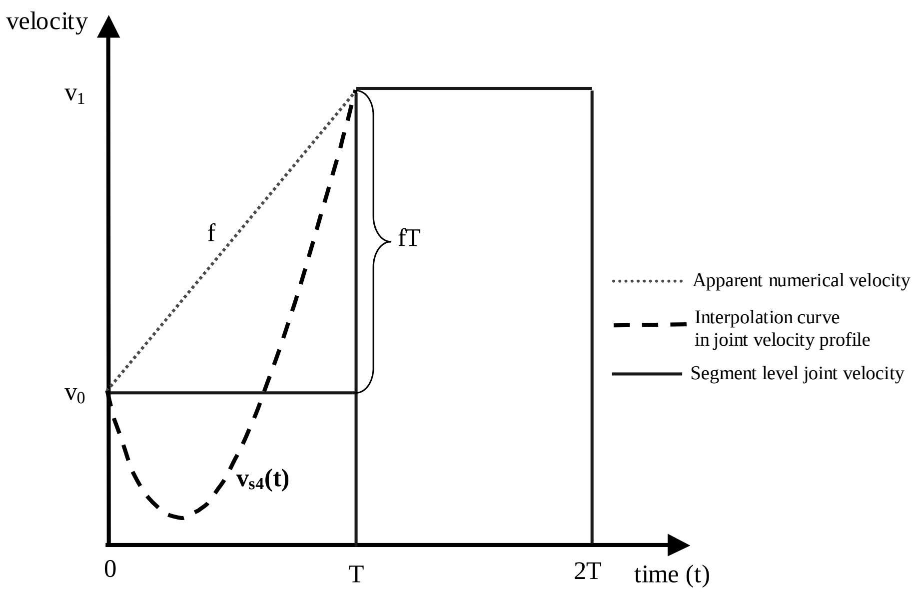

Figure 3.7: Fine Details of Joint Velocity Curves: Inter-Intra Segments

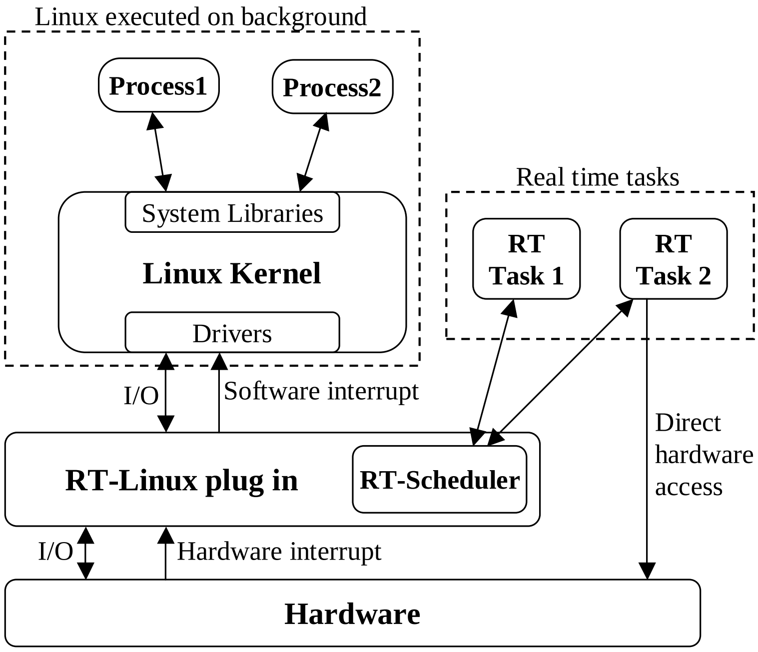

Figure 3.8: Detailed Architecture of RT-Linux Kernel

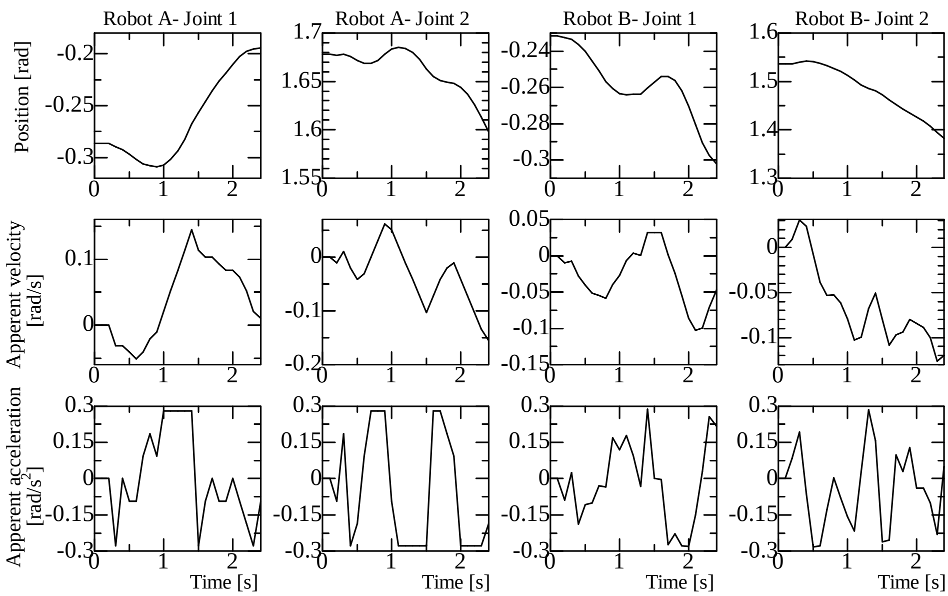

Figure 3.9: Coarse Level Input Trajectory Prior to Interpolation

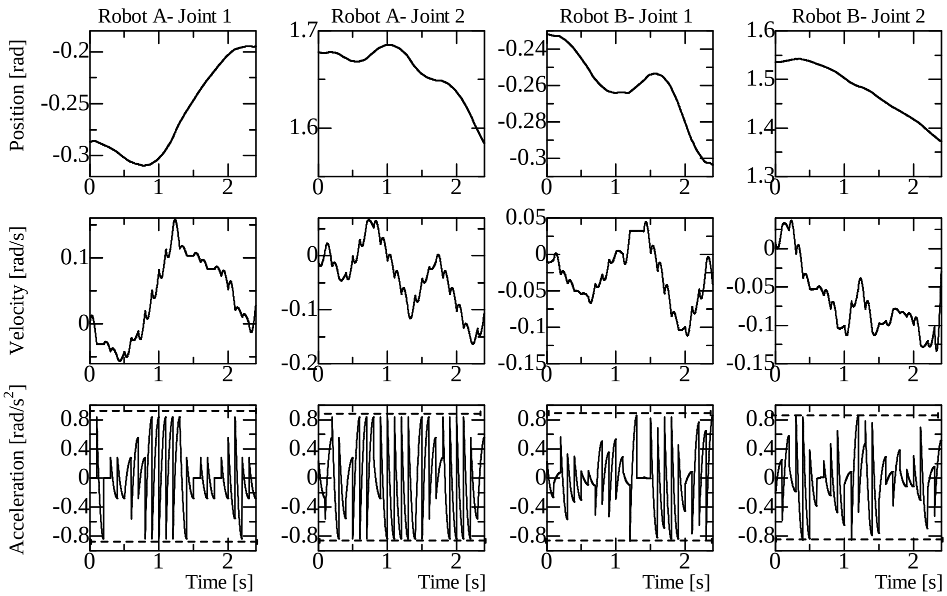

Figure 3.10: Fine Level Input Trajectory after Interpolation

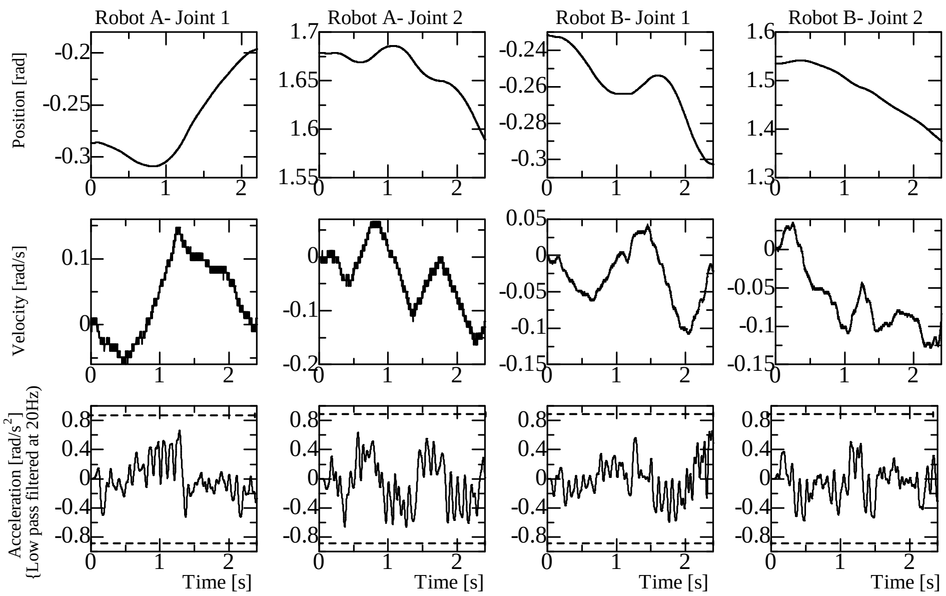

Figure 3.11: Experiment Results of Two Robot Trajectories

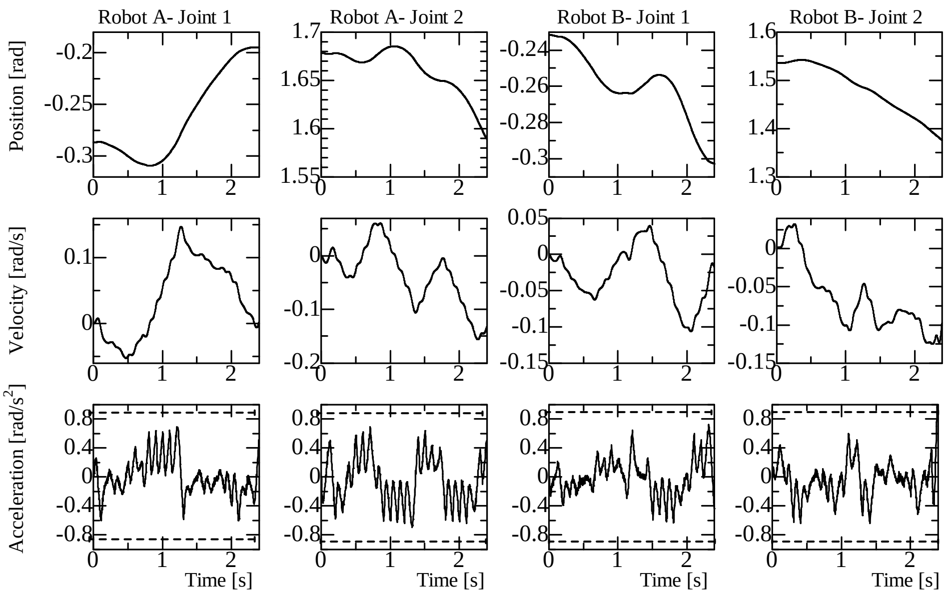

Figure 3.12: Simulation Results with Two Robot Output Trajectories

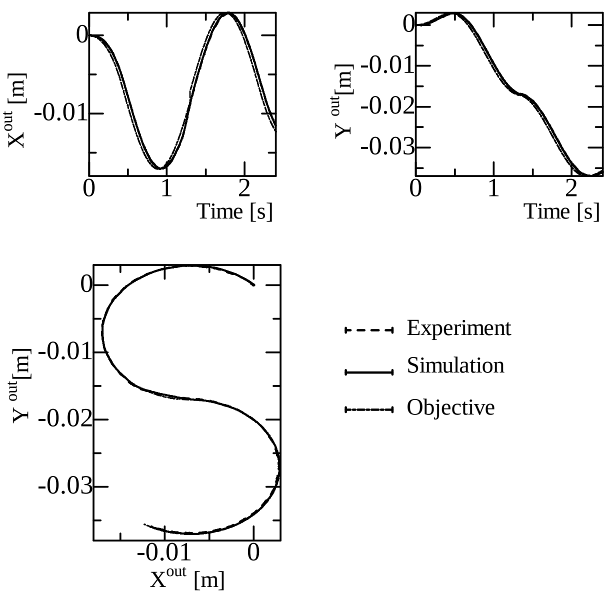

Figure 3.13: Objective and Cooperative Trajectories of Simulation and Experiment

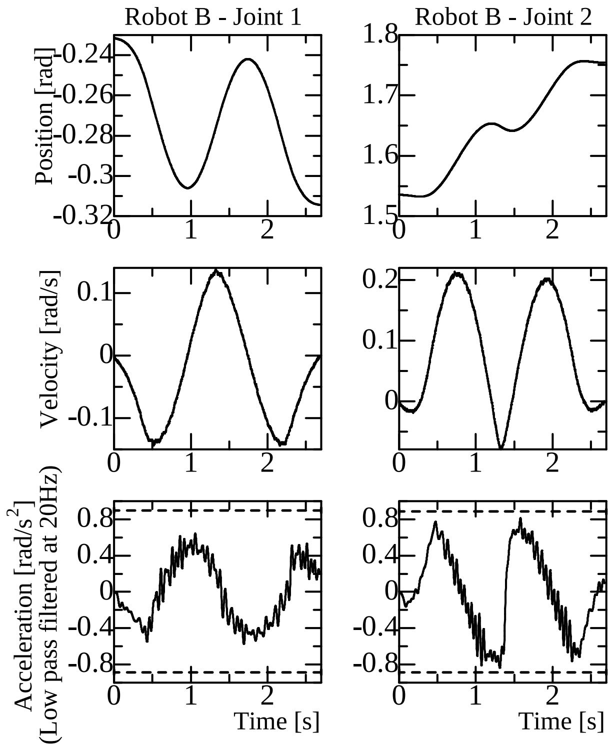

Figure 3.14: Experimental Results of Minimum Time Mono Robot Trajectory Generated under Acceleration Constraint

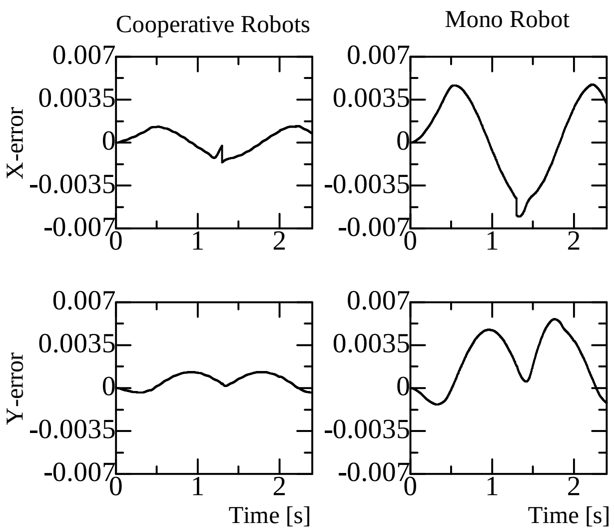

Figure 3.15: Comparison of Simulation Error in Workspace

Chapter 4

Minimum Time Cooperative Control of Two Robots

4.1 Preliminaries

4.1.1 Prelude to minimum time cooperative control

Realization of cooperative control for two articulated robots under specified cooperative trajectory determined by application itself was discussed in **chapter 3**. The timing information of the objective locus is relaxed and intends to achieve the locus in minimum time through cooperative motion of two robots. This type of cooperative control is termed as minimum time cooperative control of two robots in this context. The concept of trajectory planner is elaborated with the same S-shaped locus used in **chapter 3** and the planner is designed for the trajectory planning of two Cartesian robots based on the principles of kinematics.

4.1.2 Cartesian robot configuration

Cartesian robot is a robot capable of realizing lateral (x), longitudinal (y), and vertical motions spatially arranged in three mutually perpendicular directions. The relationship established between configuration space and working space is linear and therefore makes it easy in planning of trajectories particularly in cooperative control, since there is no nonlinear kinematics and inverse kinematics involved as in articulated robot arms.

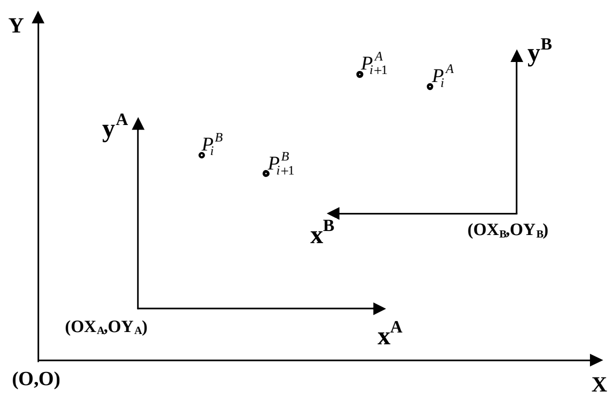

Figure 4.1: Definition of Physical Coordinate Systems

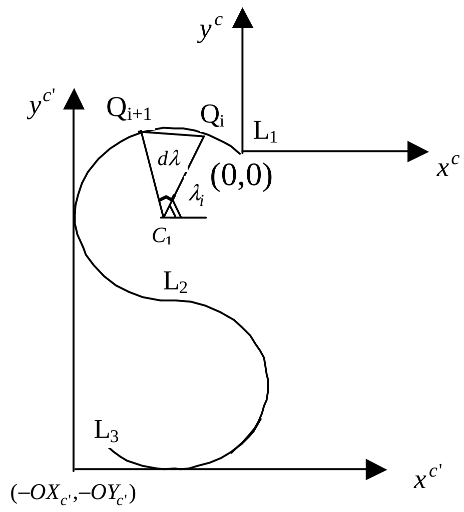

Figure 4.2: Definition of Cooperative Coordinate Systems and Parameterization of Locus

The end effector positions of two robots with respect to a fixed point of each robot are described by two coordinate systems $x^A - y^A$, $x^B - y^B$ attached to two robots. Since the relative initial end effector locations of the robots in space establishes the correlation between two coordinate systems, another Cartesian system $X - Y$ which gives the absolute spatial description is defined referring to a fixed point in the space. These three coordinate systems given in Fig. 4.1 constitute the physical coordinate space whereas the relative motion between the end effectors of two robots gives rise to a virtual coordinate system $x^c - y^c$, the cooperative Cartesian space that defines the cooperative space. The S-shaped objective locus $L_1 L_2 L_3$ is specified in cooperative space as depicted in Fig. 4.2.

The continuous time problem is discretized in time domain using the segmentation time T, which is equal to sampling time, for the sake of simplicity. The timely notation of position, velocity and acceleration is depicted in Fig. 4.3 and this convention is used throughout this chapter.

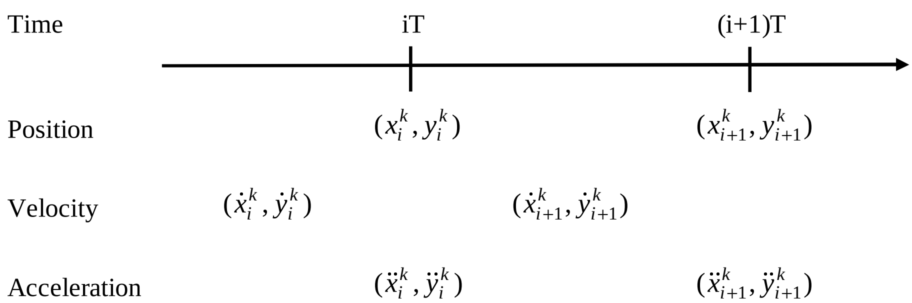

Figure 4.3: Timely Notation of Position Velocity and Acceleration

The subscript i denotes the time stamp $t = iT$ of respective quantity and superscript k $(= A, B)$ indicates the robot to which the quantity is attibuted.

Suppose that the end effector positions of robots A and B are at P_i^A, P_{i+1}^A, P_i^B, P_{i+1}^B locations when $t = iT$ and $(i+1)T$ respectively. The points Q_i and Q_{i+1} denote the respective relative displacements between the two end effectors in cooperative space referring to cooperative coordinate system given in Fig. 4.2. When the points P_i^A, P_{i+1}^A, P_i^B, P_{i+1}^B are expressed referring to $X - Y$ coordinate system

$$\begin{bmatrix} P_i^A \\ P_{i+1}^A \end{bmatrix} = \begin{bmatrix} OX_A + x_i^A & OY_A + y_i^A \\ OX_A + x_{i+1}^A & OY_A + y_{i+1}^A \end{bmatrix} \tag{4.1}$$

and

$$\begin{bmatrix} P_i^B \\ P_{i+1}^B \end{bmatrix} = \begin{bmatrix} OX_B + x_i^B & OY_B + y_i^B \\ OX_B + x_{i+1}^B & OY_B + y_{i+1}^B \end{bmatrix} \tag{4.2}$$

where (OX_A, OY_A) and (OX_B, OY_B) are the origins of the two coordinate systems $x^A - y^A$ and $x^B - y^B$ with respect to common spatial coordinate system $X - Y$.

4.1.3 Parameterization of objective locus

As the end effectors of two robots are coincided with each other at the beginning, the initial relative displacement is zero and hence the origin of cooperative coordinate system corresponding to the starting point. However, for convenient parameterization of the S-shaped locus, a translational version of the cooperative coordinate system. $x^{c'} - y^{c'}$ is selected such that $x^{c'}$ and $y^{c'}$ axes are tangential to the locus at the lowest and the left most points.

The S-shaped locus is assumed to be formed by two circular arcs having equal curvatures $(1/r_s)$. Referring to $x^{c'} - y^{c'}$ coordinate system, the relative displacement in cooperative space at times $t = iT$ and $(i+1)T$ can be denoted by points Q_i and Q_{i+1}. They can be expressed in parametric form as

$$\begin{bmatrix} Q_i \\ Q_{i+1} \end{bmatrix} = \begin{bmatrix} x_i^{c'} & y_i^{c'} \\ x_{i+1}^{c'} & y_{i+1}^{c'} \end{bmatrix} = \begin{bmatrix} p + r_s cos(\lambda_i) & q + r_s sin(\lambda_i) \\ p + r_s cos(\lambda_{i+1}) & q + r_s sin(\lambda_{i+1}) \end{bmatrix} \tag{4.3}$$

and

$$\lambda_{i+1} = \lambda_i + d\lambda_i \tag{4.4}$$

where λ_i and $d\lambda_i$ are the values of the parameter at time $t = iT$ and the trial increment given to the parameter at the same point. (p, q) is the instantaneous center of the locus and it takes $(r, 3r)$ in the first circular arc L_1L_2, whereas $(p, q) = (r, r)$ for the second circular arc L_2L_3. λ value is assumed to be varied $[\pi/4, 3\pi/2]$ and $[\pi/2, -3\pi/4]$ in the circular arcs L_1L_2 and L_2L_3 respectively.

4.1.4 Physical coordinate to cooperative coordinate mapping

The relative displacement between the end effector positions of the two robots referring to common spatial coordinate system $X - Y$ is equivalent to the relative displacement in cooperative space since the initial cooperative displacement is zero.

When this relationship is expressed in mathematical form,

$$\begin{bmatrix} Q_i \\ Q_{i+1} \end{bmatrix} = \begin{bmatrix} P_i^A - P_i^B \\ P_{i+1}^A - P_{i+1}^B \end{bmatrix} \tag{4.5}$$

$$\begin{bmatrix} x_i^{c'} - OX_{c'} & y_i^{c'} - OY_{c'} \\ x_{i+1}^{c'} - OX_{c'} & y_{i+1}^{c'} - OY_{c'} \end{bmatrix} = \begin{bmatrix} x_i^A - OX_A & y_i^A - OY_A \\ x_{i+1}^A - OX_A & y_{i+1}^A - OY_A \end{bmatrix} - \begin{bmatrix} OX_B - x_i^B & y_i^B - OY_B \\ OX_B - x_{i+1}^B & y_{i+1}^B - OY_B \end{bmatrix} \tag{4.6}$$

where $(-OX_{c'}, -OY_{c'})$ is the origin of $x^{c'} - y^{c'}$ coordinate system referring to $x^c - y^c$ coordinates. It can be determined in terms of the radius of circular arcs and yielded as

$$[-OX_{c'}, \ -OY_{c'}] = \left[-\left(1 + \frac{1}{\sqrt{2}}\right) r, \ -\left(3 + \frac{1}{\sqrt{2}}\right) r\right] \tag{4.7}$$

If the incremental cooperative motion (marginal cooperative displacement) achieved during the time interval $[iT, (i+1)T]$ is denoted by the direction vector $[dx_i, dy_i]$ then

$$[dx_i \ \ dy_i] = \left[x_{i+1}^{c'} - x_i^{c'} \ \ y_{i+1}^{c'} - y_i^{c'}\right] \tag{4.8}$$

For a given λ_i and trail $d\lambda_i$, $[dx_i \ \ dy_i]$ is a constant and the gradient in cooperative Cartesian coordinates can be denoted by a constant m_i where

$$dy_i = m_i \ dx_i. \tag{4.9}$$

4.2 Generic Form Problem Statement

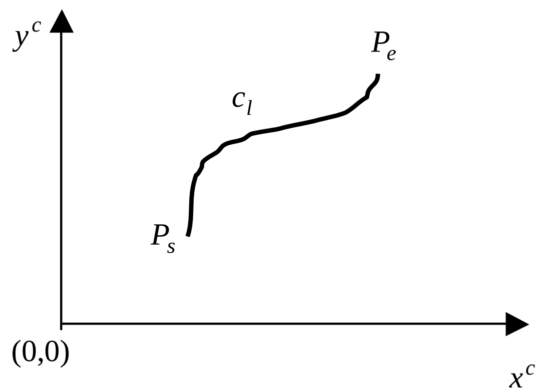

Figure 4.4: Generic Form Objective Cooperative Locus

The objective locus c_l can be mathematically represented by

$$c_l = \{(x^c, y^c) | (x^c, y^c) \in locus \ P_s P_e\} \tag{4.10}$$

where the ordered pair (x^c, y^c) describes the objective cooperative path in cooperative space expressed in Cartesian coordinates, P_s and P_e are the start and the end points of cooperative locus.

Referring to the coordinate frames defined in Fig. 4.1, the relative displacement of end effector of two robots during $[iT,(i+1)T]$ constitutes the cooperative displacement and it can be mathematically stated as,

$$\begin{bmatrix} dx_i^A \\ dy_i^A \end{bmatrix} - \begin{bmatrix} dx_i^B \\ dy_i^B \end{bmatrix} = \begin{bmatrix} dx_i \\ dy_i \end{bmatrix} \tag{4.11}$$

where dx_i^A, dy_i^A, dx_i^B, dy_i^B are the abscissa and ordinate incremental displacements of robots A and B in the time interval $[iT,(i+1)T]$.

The minimum time condition for cooperative control is expressed as

$$Minimize \;\; T_c = \int_{c_l} \frac{d\lambda}{v^c} \tag{4.12}$$

where T_c, $d\lambda$, and v^c are the task completion time, the infinitesimal movement made along the cooperative locus (monotonous infinitesimal increment of parameter) and the cooperative velocity along the locus respectively, subjected to the joint acceleration bounds specified by

$$\begin{array}{rcl} |\ddot{x}^A(t)| & \leq & a_{max} \\ |\ddot{y}^A(t)| & \leq & a_{max} \\ |\ddot{x}^B(t)| & \leq & a_{max} \\ |\ddot{y}^B(t)| & \leq & a_{max} \end{array} \tag{4.13}$$

Here a_{max} is the maximum permissible acceleration of each joint. Since both robots are initially at rest and eventually achieve zero velocity at final positions, boundary conditions of velocity can be stipulated by

$$\begin{bmatrix} \dot{x}^A & \dot{y}^A \\ \dot{x}^B & \dot{y}^B \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad \text{at } \lambda = \lambda_\mathrm{s} \text{ and } \lambda_\mathrm{e} \tag{4.14}$$

where λ_s and λ_e are the parameter values corresponding to the start and the end points P_s and P_e. Alternatively boundary condition for velocity given in equation 4.14 can be stated in another form given by $\dot{\lambda} = 0$ at $\lambda = \lambda_\mathrm{s}$ and λ_e.

4.3 Time Optimal Cooperative Trajectory Generation

4.3.1 Design issues of minimum time cooperative control algorithm

An algorithm for determining true minimum time trajectory of a manipulator along a prescribed path under given dynamics was initially proposed by Bobrows et al. [citaion29], and subsequent developments have been adapted by many researchers to improve computational efficiency [citation21] and many other aspects. Parametric representation of the desired path, substitution of the parameter in manipulator dynamic equations and thereby create a phase-plane plot of the parameter are the core underlying concepts of the method. Bounds on individual joint torque are converted into bounds on parametric acceleration and hence create limit curves as well as maximum tolerable velocity curves on the phase plane. Then using the fact that the minimum time solution will be bang-bang in the acceleration, a set of switching points is found and thereby plan the trajectory.

The prime intension focused in this chapter is to speed up the operation through time optimal cooperative control with a given cooperative path. A systematical extension of the minimal time control of a single manipulator is not convenient even for two-robot case due to the following issues encountered.

1. In true minimum time control, maximum permissible joint velocity bounded by joint acceleration/torque constraints along a curvy locus is a function of the configuration of the robot and hence varies along the locus. Therefore, in minimum time cooperative control of two robots, velocity bounds are inter-dependent in phase plane.
2. There is no strategy avalable to decompose the objective locus into two unique conjugate paths that minimizes the time. Therefore the single robot time optimal trajectory planning techniques such as phase plane method can not be applied in straightforward manner.
3. Even cooperative trajectory can not be uniquely mapped onto two conjugate trajectories due to redundant degrees of freedom of the cooperative two-robot system. In other words, decomposition of objective trajectory into to complementary trajectories is ill conditioned, as described in **chapter3**.

Since synthesizing issues are inter-related to the adoptation of minimum time trajectory-planning algorithm, trajectory planning and path planning can not be dissociated with each other intact with time optimality criterion. A unified trajectory planning option remains potential, but the complexity of planning algorithm is inevitably quite complex. Therefore, this complexity compels to carry out trajectory planning off-line.

4.3.2 Proposed trajectory planning algorithm

By Pontryagin's maximum principle, it is a known result that the time optimum trajectory is a sequence of motion segments consisting of maximum acceleration and maximum deceleration. Therefore the proposed minimum time cooperative trajectory planner is based on bang-bang control in acceleration profile to ensure minimum time. The entire cooperative trajectory planning is carried under two modes, progressive mode corresponding to maximum acceleration and advance through folding back mode corresponding to maximum deceleration of either or both robots under given configuration and path specifications.

In order to reduce the dimension of the space used for planning, path primitives with reference to its geometrical description is mapped onto a sequence of parameter values, which gradually increases or decreases, when the point moves along the path in a given direction. The parameter value of the path is used for trajectory planning. Initially, planning is carried out in progressive mode until no solution condition is encountered due to kinematic limitations. In such situation, planning steps back to the previous segment successively until no solution condition disappears and surpasses the parameter value at which no solution condition has previously met under progressive mode. Then again switches back to progressive mode and proceeds until the next occurrence of no solution in planning. This process is repeatedly iterated until the entire trajectory is planned. The steps involved in trajectory planning can be stated in the following way.

a) Parameterize the objective locus to reduce the dimension of the planning space (refer Fig. 4.1)
b) Get the parameter value corresponding to the current point of the objective locus.
c) Give a trail increment to the parameter, and calculate the ensuing trail point at the end of sampling time, on the locus (equation (4.4))
d) Find the direction vector along the increment in displacement (equation (4.3) and (4.8))
e) As of robot A's velocity and position of end effector, determine the range of positions that would be feasible in parallel to the direction vector. These are corresponding to the achievement of maximum acceleration and deceleration at either joint of robot A, parallel to the direction vector (refer calculation of optimum parameter increment)
f) Select the furthest point corresponding to maximum acceleration
$({}^a x_{i+1}^A, {}^a y_{i+1}^A) = \{(x_{i+1}^A, y_{i+1}^A) | (x_{i+1}^A, y_{i+1}^A) \in max\{\sqrt{dx_i^2 + dy_i^2}\}\}$
g) Evaluate the new joint velocities and positions of robot A.
$\dot{x}_i^A = (x_{i+1}^A - x_i^A)/T \quad \dot{y}_i^A = (y_{i+1}^A - y_i^A)/T$
h) Compute the complement motion that should be made by robot B so that the relative motion between two robots yields the a trail point on the locus
$x_{i+1}^A = x_i^A - ({}^a x_{i+1}^A - {}^a x_i^A) + dx_i$
$y_{i+1}^A = y_i^A + ({}^a y_{i+1}^A - {}^a y_i^A) - dy_i$
i) Estimate the joint velocities and thereby joint accelerations.
$\dot{x}_i^A = (x_{i+1}^A - x_i^A)/T \quad \dot{y}_i^A = (y_{i+1}^A - y_i^A)/T$
$\ddot{x}_i^A = (\dot{x}_{i+1}^A - \dot{x}_i^A)/T \quad \ddot{y}_i^A = (\dot{y}_{i+1}^A - \dot{y}_i^A)/T$
j) Check for the violation of acceleration constraint at the joints of robot B.
$|\ddot{x}_i^A| \leq a_{max}, \quad |\ddot{y}_i^A| \leq a_{max}$
k) Adjust the trial increment to locus parameter $d\lambda_i$ and repeat step b)~j) so as to find the range of trail increments that can be taken without violating the acceleration constraints with respect to robot B.
l) Select the maximum value of the parameter and determine the new point
${}^a d\lambda_i = max\{d\lambda_i\}, \quad Q_{i+1} = [p + r_s cos(\lambda_i + {}^a d\lambda_i), \ q + r_s sin(\lambda_i + {}^a d\lambda_i)]$
m) Increment the segment counter,
n) Reiterate step b)~m) until no solution condition is met at either robot A or robot B.
o) If there is no solution for either/both robot(s), store the value of the parameter corresponding to no solution condition on respective folding back pointer (two folding back pointers are maintained for both robots) if the already stored value is less than the current value. In case of updating the value of folding back pointer, set the degree of folding back to one.
p) Stepping back the states by the required degree of folding back $i = i - foldbacklevel$

q) If the no solution condition corresponds to robot A, then select the shortest point instead of the furthest point in step f). If there is no solution corresponding to robot B, then select the minimum parameter value corresponding to new point in step l).
r) Repeat step b)∼q) without executing step n), replacing steps f) and l) with step p). Inability to find successive feasible solutions until surpassing the folding back point enforces to increment the degree of folding back and repeat step r).
s) When the parameter value exceeds that of folding back pointer, the folding back scenario is released for respective robot, switches back to normal (accelerative) mode and reiterate step n).
t) The above steps are repeated until the parameter value exceeds that corresponding to final position of the locus
u) Check whether the maximum joint velocity greater than the maximum allowable stoppage velocity at near vicinity of final position.
v) If the maximum joint velocity exceeds the allowable stoppage velocity limit, recursively go back to previous states using folding back scenario and select the minimum parameter value as in step l) and/or the shortest point in step f) depending on the robot in which the maximum joint velocity occurs.
w) Step v) is reiterated until stoppage velocity condition satisfied at final position.

The complete trajectory-planning algorithm in a form of flow chart is given in Fig. 4.5. Here joint positions and velocities of robots A and B, parameter value, mode values of two robots constitute the state at a particular knot. The algorithm depicted in Fig. 4.6 is used to determine the parameter increment and it is the detailed flow chart corresponding to block P to Q in Fig. 4.5.

4.4 Theoretical Aspects of the Proposed Algorithm

4.4.1 Philosophical notions

In systematic extension of minimum time single robot planner to cooperative control of two robots, in the light of the design level issues described in **Chapter 4.3.1**, the following assumptions have been made.

1. The minimal distance path at segment level is assumed to be the minimum time path when concerning the complete locus,
2. Optimal solution at each knot constitutes the optimal solution for the entire trajectory,
3. Simultaneous optimization of complementary motions of each robot under cooperative motion, guarantees the optimization of the cooperative motion.

The current position in cooperative space and the next trial point as per trail increment of the parameter in the same space are considered as end points of each segment, and such segments are approximated by straight line segments termed as inter-knot line segments. Each segment has segmentation timing and it is equal to update rate (sampling time) of the zero order hold servo controller. Therefore, every trajectory point becomes a knot point and no deterioration of the trajectory is resulted as a part of the planning process.

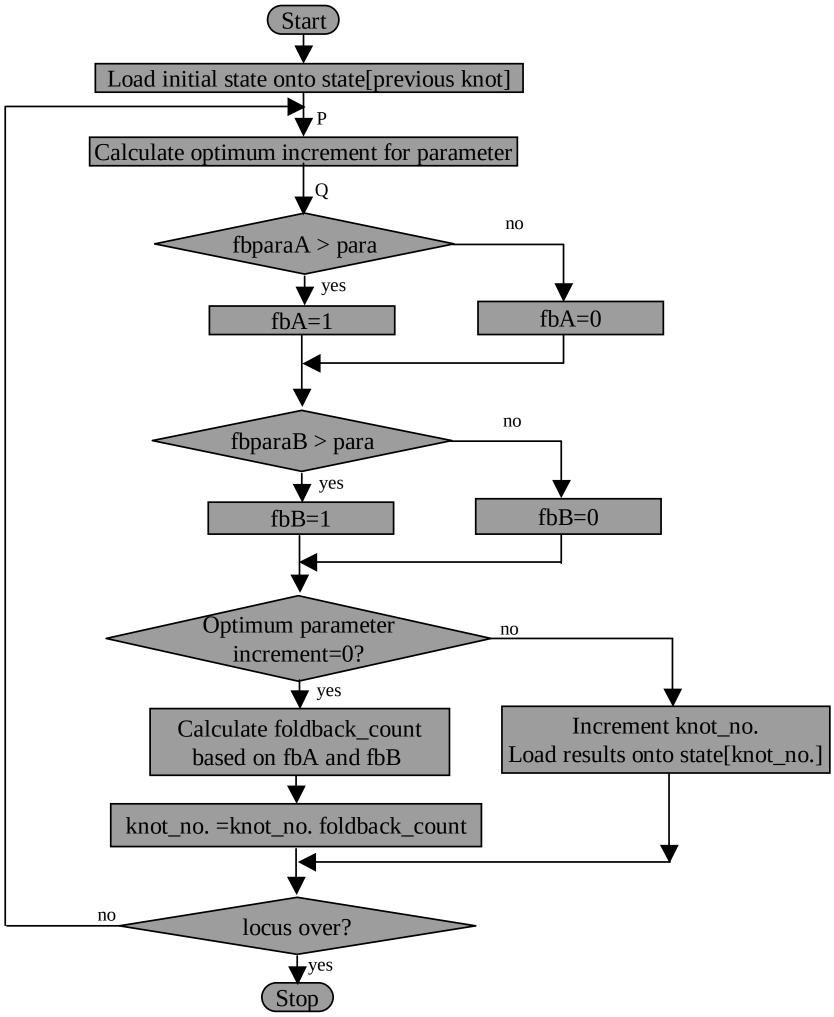

Figure 4.5: Entire Cooperative Trajectory Generation Criterion

Starting from the parameter value of current point, the trial increment to the parameter is optimized as per the rules specified in progressive mode and advancing through fold back mode. In planning of cooperative trajectory through either mode, a concurrent policy based optimum resultant motion is applied for every segmentation time. This is realized through the opposite movements of two end effectors along the inter-knot line segment subjected to maximum joint acceleration constraints. Though this opposite motion of two end effectors sufficiently ensures the minimum time criterion for Cartesian robots, it is not necessarily held for articulated robots due to the nonlinearities involved with kinematics and inverse kinematics. For the sake of completeness theoretical proofs are provided in the mathematical form for

a) When no solution condition is prevailed in progressive mode, the becessity to step back to previous knot level and need to adapt different planning policy as a resolution strategy (in **Chapter 4.4.2**)

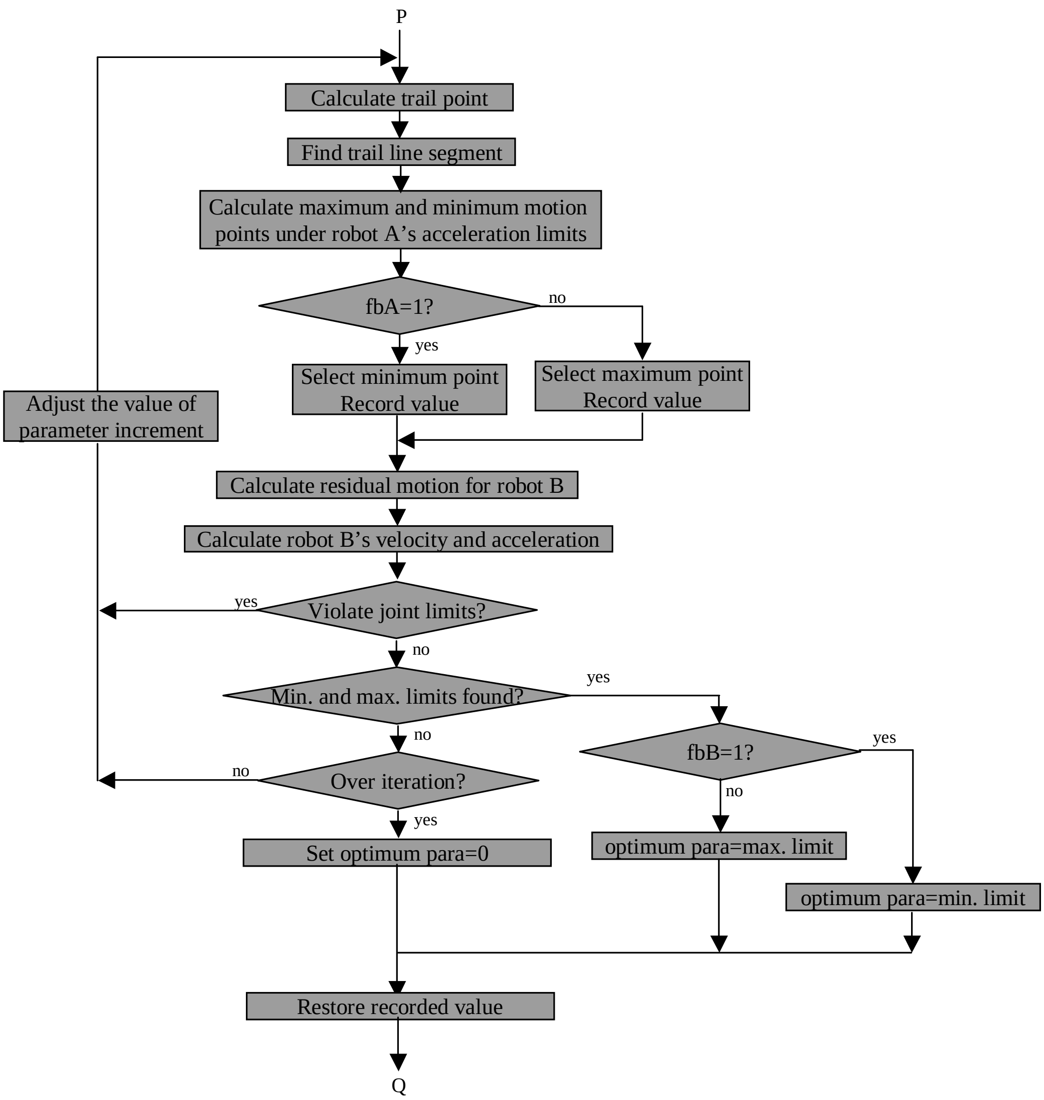

Figure 4.6: Algorithm for Calculation of Optimum Parameter Increment

b) Progressive mode and advancing through fold back mode correspond to maximum accelerative motion and maximum decelerative motion along a specified direction under limiting conditions of kinematic constraints (in **Chapter 4.4.3**)

Further, an additional constraint is introduced for stepping back to make sure a fair task distribution of each robot. The displacement contribution of each robot to the total cooperative displacement is a two tailed bound. The upper and the lower bounds of distance contribution can be mathematically expressed as

$$0.35d_i \leq^k d_i \leq 0.65d_i \tag{4.15}$$

where ${}^kd_i$ and d_i are the displacements of k^{th} robot $(k = A, B)$ and cooperative displacement or the length of the inter-knot line segment during the time interval $[iT, (i+1)T]$.

In the proposed cooperative trajectory planner, advancing through folding back is carried out only at the occurrence of no solution condition until its disappearance by surpassing the point of occurrence. Therefore, inherent tendency of switching back to default progressive mode is ensured with a deliberate priority assigned to the progressive mode so as to ensure the minimal time criterion in bang-bang control.

4.4.2 No solution condition

Prevalence of no solution condition is equally applicable to both robots A and B, when the end effector velocity can not be further increased under progressive mode due to joint kinematic limitations. A mathematical analysis of no solution condition is provided and the means of resolving (solution strategy) no solution condition is briefly discussed.

By definition of acceleration, the relationship between the velocity and acceleration of the k^{th} robot during $[iT, (i+1)T]$ can be mathematically established as

$$\begin{bmatrix} \dot{x}^k_{i+1} \\ \dot{y}^k_{i+1} \end{bmatrix} = \begin{bmatrix} \dot{x}^k_i \\ \dot{y}^k_i \end{bmatrix} + T \begin{bmatrix} \ddot{x}^k_i \\ \ddot{y}^k_i \end{bmatrix} \qquad for\ k = A, B \tag{4.16}$$

and the marginal displacement in the same time interval is expressed by

$$\begin{bmatrix} dx^k_{i,i+1} \\ dy^k_{i,i+1} \end{bmatrix} = \begin{bmatrix} \dot{x}^k_i \\ \dot{y}^k_i \end{bmatrix} T + \begin{bmatrix} \ddot{x}^k_i \\ \ddot{y}^k_i \end{bmatrix} T^2 \qquad for\ k = A, B \tag{4.17}$$

where $[dx^k_{i,i+1}\ \ dy^k_{i,i+1}]^T$ denotes the respective incremental displacement along x and y directions during $[iT, (i+1)T]$.

Suppose $[dx_i\ dy_i]$ is the marginal displacement or directional vector in cooperative space. In order to realize minimum time criterion, the end effector movement should be along the inter knot direction vector but opposite in direction. If the gradient of the i^{th} segment is denoted by m_i, then from equation (4.9)

$$dy^k_{i,i+1} = m_i\ dx^k_{i,i+1} \qquad \text{for k} = \text{A}, \text{B} \tag{4.18}$$

By combining equations (4.17) and (4.18) gives $m_i(\dot{x}^k_i T + \ddot{x}^k_i T^2) = \dot{y}^k_i T + \ddot{y}^k_i T^2$

$$\ddot{y}^k_i = m_i(\dot{x}^k_i/T + \ddot{x}^k_i) - \dot{y}^k_i/T \tag{4.19}$$

Since the acceleration in either direction is bounded by the maximum limit

$$|\ddot{y}^k_i| \le a_{max} \qquad \text{for k} = \text{A}, \text{B} \tag{4.20}$$

and it yields

$$\begin{array}{ll} (\dot{y}^k_i/T - a_{max})/m_i - \dot{x}^k_i/T \le \ddot{x}^k_i \le (\dot{y}^k_i/T + a_{max})/m_i - \dot{x}^k_i/T & \text{if } \mathrm{m_i} > 0 \\ (\dot{y}^k_i/T - a_{max})/m_i - \dot{x}^k_i/T \ge \ddot{x}^k_i \ge (\dot{y}^k_i/T + a_{max})/m_i - \dot{x}^k_i/T & \text{if } \mathrm{m_i} < 0 \\ |-y^k_i/T| \le a_{max} & \text{if } \mathrm{m_i} = 0 \end{array} \tag{4.21}$$

Similarly acceleration in x-direction is also subjected to upper and lower bounds. It can be mathematically expressed by

$$-a_{max} \leq \ddot{x}_i^k \leq a_{max}. \tag{4.22}$$

$\ddot{x}_i^k$ in equation (4.21) is supposed to be complied with the bounds given by equation (4.22) and it deduces to satisfy a simultaneous relationship specified by the inequalities

$$-a_{max} \leq (\dot{y}_i^k/T - a_{max})/m_i - \dot{x}_i^k/T \leq a_{max} \tag{4.23}$$

and

$$-a_{max} \leq (\dot{y}_i^k/T + a_{max})/m_i - \dot{x}_i^k/T \leq a_{max} \tag{4.24}$$

Inability to conform the two concurrent inequalities shown in (4.23) and (4.24) gives rise to the occurrence of no solution condition. According to the inequalities, the joint velocities at time iT is responsible for no solution condition. Therefore, in order to alleviate no solution condition, altering the joint velocities at time iT is mandatory by stepping back and following another planning criterion. Hence stepping back in presence of no solution and switching back to advance through fold back mode is suggested in the proposed cooperative trajectory planner.

4.4.3 Progressive mode and advancing through fold back mode

In progressive mode both robots travel at their maximum possible distances along the direction specified by the direction of the segment to be planned in cooperative space, starting from the current positions and joint velocities. This corresponds to achieve maximum acceleration at one or more joints of each robot. When applying the maximum displacement strategy to each robot, it is convenient to provide a mathematical proof on maximum displacement corresponding to achieve maximum acceleration, at least at one or more joints of each robot.

From equation (4.17) and (4.18) gives

$$m_i(\dot{x}_i^k T + \ddot{x}_i^k T^2) = \dot{y}_i^k T + \ddot{y}_i^k T^2 \tag{4.25}$$

The displacement of the k^{th} robot ($k = A, B$) along the given inter knot line segment ${}^k d_i$ can be stated as

$${}^k d_i = \sqrt{(dx_{i,i+1}^k)^2 + (dy_{i,i+1}^k)^2} \tag{4.26}$$

substituting from equations (4.17) and (4.18) to (4.26) yields

$${}^k d_i = T|\dot{x}_i^k + \ddot{x}_i^k T|\sqrt{1+(m_i)^2} \quad or\, {}^k d_i = T|\dot{y}_i^k + \ddot{y}_i^k T|\sqrt{1+(1/m_i)^2} \tag{4.27}$$

Since m_i is a constant for a given line segment, from simple arithmetic, ${}^k d_i$ maximizes when

a) $\dot{x}_i^k > 0$ and $\ddot{x}_i^k = (\ddot{x}_i^k)_{max}$ (equivalently $\dot{y}_i^k > 0$ and $\ddot{y}_i^k = (\ddot{y}_i^k)_{max}$)
b) $\dot{x}_i^k < 0$ and $\ddot{x}_i^k = (\ddot{x}_i^k)_{min}$ (equivalently $\dot{y}_i^k < 0$ and $\ddot{y}_i^k = (\ddot{y}_i^k)_{min}$)

In both cases, rate of change in velocity supports to increase the joint velocity at maximum capacity within the allowable bounds, no matter in which direction the current velocity prevails. Hence it proves that the progressive mode corresponds to the occurance of maximum acceleration at least one joint of each robot.

In advancing through fold back mode, at least one robot must achieve maximum deceleration. Traversing of minimum possible distance along the direction of inter knot line segment during segmentation time starting from current joint positions and velocities is the principle of putting a robot into decelerative mode. A mathematical consideration is provided to show that this minimum distance criterion is equivalent to maximum deceleration occuring at least one joint of the robot, to which robot this criterion is applied.

Mathematically it is possible to show that the minimum value of ${}^k d_i$ corresponds to hold necessarily either condition;

a) $\dot{x}_i^k > 0$ and $\ddot{x}_i^k = (\ddot{x}_i^k)_{min}$ (equivalently $\dot{y}_i^k > 0$ and $\ddot{y}_i^k = (\ddot{y}_i^k)_{min}$)

b) $\dot{x}_i^k < 0$ and $\ddot{x}_i^k = (\ddot{x}_i^k)_{max}$ (equivalently $\dot{y}_i^k < 0$ and $\ddot{y}_i^k = (\ddot{y}_i^k)_{max}$)

Thereby, it is possible to prove that the folding back mode corresponds to maximum declarative motion.

4.5 Appraisal of Planned Cooperative Trajectory

The same S-shaped example objective locus in **Chapter 3** was used to appraise the proposed trajectory planner for cooperative trajectories of two Cartesian robots. The generated raw trajectory by the proposed planner is required to be dynamically scaled so that it would be better suited to the dynamic properties of the Cartesian robots in which this trajectory implements. The parameters related to path and cooperative trajectory specifications are tabulated in Table 4.1.

Table 4.1: Path and Cooperative Trajectory Specification for S-Shaped Locus

Description of parameter	Symbol	Value
Curvature of S-shaped locus	c	1[cm]
Maximum cooperative acceleration	$a_{\max}^c$	1.0[m/s^2]
Segmentation time	T	2[ms]

In order to demonstrate the viability of planning algorithm even for loci with sharp corners, a V-shaped trajectory was planned. The specifications of the V-shaped path and joint acceleration limits are briefly specified in Table 4.2.

Table 4.2: Path and Cooperative Trajectory Specification for V-Shaped Locus

Description of parameter	Value
Length of straight line segment of V shape	5[cm]
Curvature of the corner	0.5 [cm]
Intended angle between two line segments	$\pi/3$
Sampling time	2[ms]
Maximum acceleration of each joint	0.5 [m/s^2]

Figure 4.7 shows the minimum time cooperative trajectories of two Cartesian robots to constitute S-shaped cooperative path in simultaneous operation whereas Fig. 4.8 depicts the same for V-shaped cooperative locus. The joint accelerations of each robot in two cases are bounded to $1[m/s^2]$ and $0.5[m/s^2]$ respectively. The conformation of acceleration bounds can be easily noticeable from the third row in Fig. 4.7 and Fig. 4.8, but no specific limit is assigned for joint velocities, end effector velocity of each robot or the tangential cooperative velocity.

For comparison purposes, minimum time equivalent trajectories for S-shaped and V-shaped paths are planned for a single robot too by ceasing the movement of one robot. The illustrations can be found in Fig. 4.9 and Fig. 4.10 respectively. Table 4.3 provides a brief comparison of the same trajectories in task completion time's point of view.

Table 4.3: Comparison of Two Robot Cooperative Trajectory with Single Robot Trajectory in Task Completion Time for Both Loci

Case	**Task completion time [s]**	
	S-shape	***V-shape***
Single Robot	0.828	1.226
Two robot	0.584	0.880

The effectiveness of the proposed minimum time cooperative planning scenario is further investigated in the following direction; how cooperative trajectories of individual robots may affect in long run using the fairness criteria and the cumulative contribution of robot A for cooperative task as illustrated in Fig. 4.11 for S-shaped locus. Figure 4.12 demonstrates the cooperative velocity profile to investigate the velocity variation and thereby its impact on the quality of the end product. Here change in tangential velocity may become a performance criterion in cooperative control.

Supplementarily, Fig. 4.13 illustrates the velocity profiles of V-shaped trajectory in the cooperative space and Fig. 4.14 depicts the objective V-shape trajectory in the working space.

4.6 Concluding Remarks

A true minimum time cooperative trajectory planner for two Cartesian robots based on kinematics, to speed up the operation through cooperative control is illustrated in this chapter. For a given cooperative path, minimum time complementary trajectories for cooperative control are planned under joint acceleration limits of each robots. Performance criteria referring to each robot as well as cooperative motion have been evaluated and thereby confirm the effectiveness of input trajectories generated by the proposed planner for cooperative control of two Cartesian robots.

Since the approach is kinematically based, laser cutting, paint spraying and contour welding are few of the potential typical industrial applications of the proposed planner. Because of being off-line algorithm, computational power of the reference input generator has no significant importance. Since off-line programming system will not cause the production equipment to be tied up when it needs to be reprogrammed, automated factories can stay in production mode greater percentage of time. Besides, the proposed planner does not demand for any change in hardware or considerable reconfiguration of the system, it can be readily and directly adopted to existing servo systems. Though the exemplification is carried out with a planar case, the cooperative trajectory planner can be applicable to planning of cooperative trajectories with higher dimensionality. Further, the concept of trajectory planning is extensible to articulated robot arms in conjunction with kinematics and inverse kinematics. This is another imperative feature of the proposed planner.

Bang-bang control in acceleration profile ensures the minimum time, but in turn causes to switching between the limiting acceleration and therefore resultant jerks may detract in case of higher acceleration limits in trajectory planning. This is a fundamental drawback of bang-bang control suffers from and restriction of acceleration limit to acceptable jerk levels is the most simple solution. Accommodation of jerk constraint to the problem formulation and optimize the task completion time with this additional constraint might be another feasible solution. Altering the control objective from true minimum time solution to near minimum time solution can resolve this problem efficiently and easily.

This trajectory-planning algorithm cannot be implemented in real time control due to two following intrinsic drawbacks of the planner,

a) Use of uncertain level of stepping back during planning and alternating the policy of planning, and
b) Higher time complexity involved with the proposed planning algorithm since planning is done through a unified approach.

In addition, the planned trajectory is restricted to structured environments. This is pretty adequate to industrial environments, but in case of extending to much general framework a number of sophistications should be embedded through sensory feed back. Contingency planned trajectories can easily manage the situation if the uncertainties are exclusively and exhaustively known.

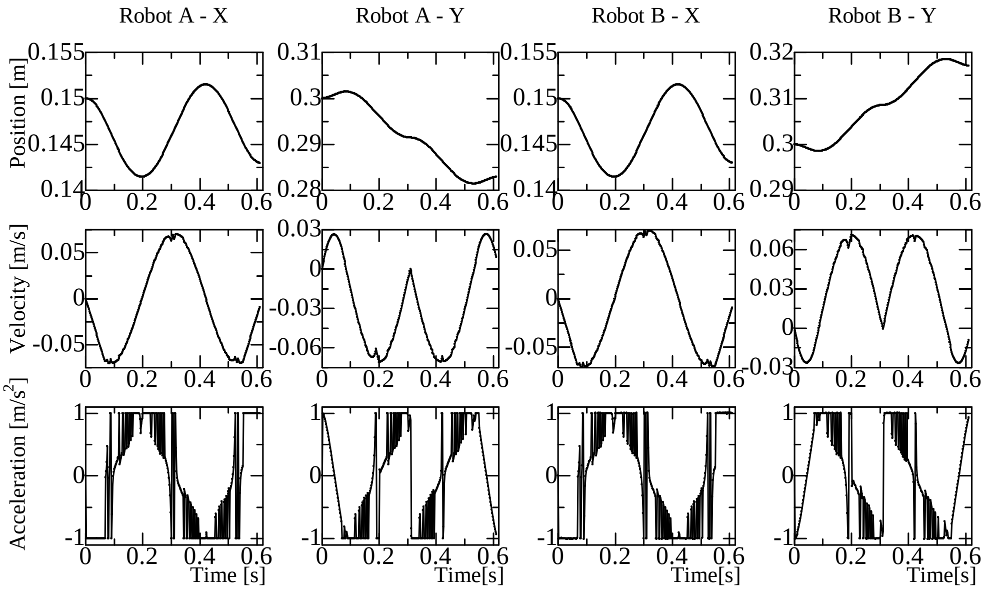

Figure 4.7: Minimum Time Input Cooperative Trajectory of Two Cartesian Robots for S-Shaped Locus

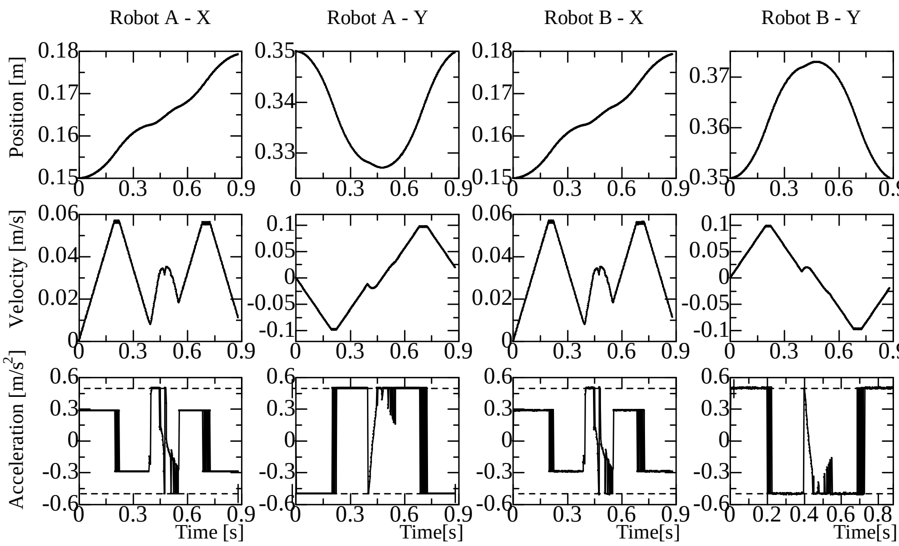

Figure 4.8: Minimum Time Input Joint Space Trajectory of Two Cartesian Robot for V-Shaped Locus

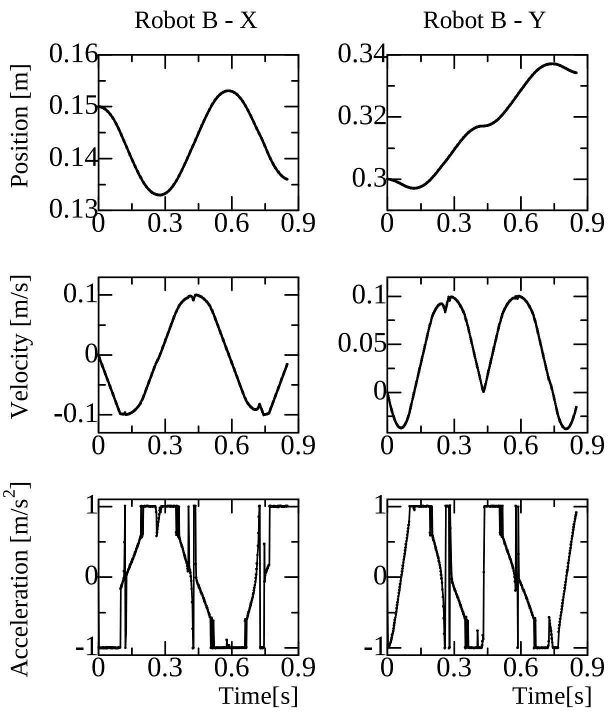

Figure 4.9: Minimum Time Input Trajectory of Single Cartesian Robot for S-Shaped Locus

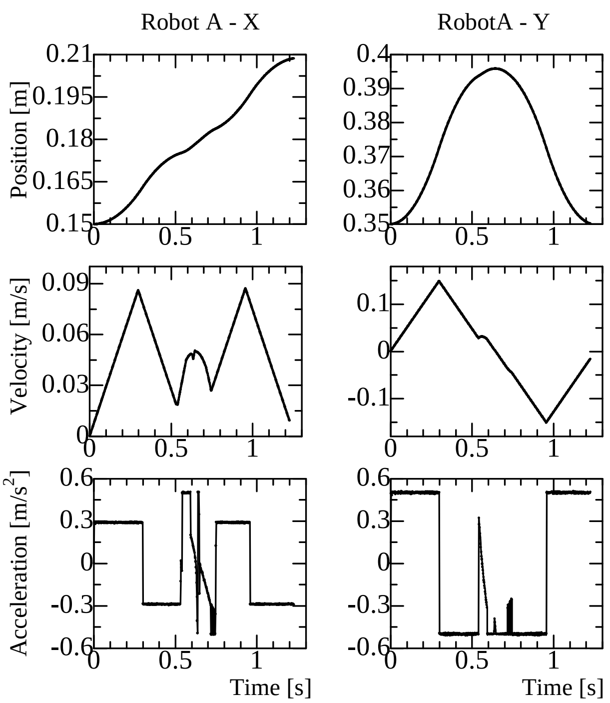

Figure 4.10: Minimum Time Input Trajectory of Single Cartesian Robot for V-Shaped Locus

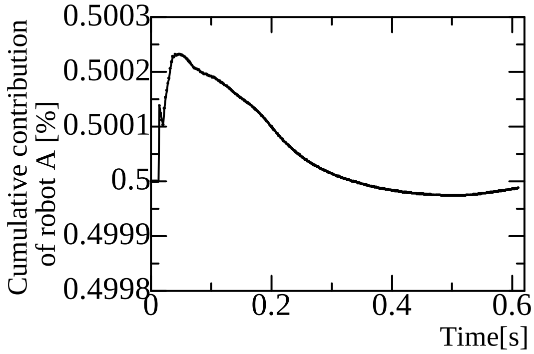

Figure 4.11: Cumulative Contribution of Robot A in Cooperative Control of Two Cartesian Robots for S-Shaped Locus

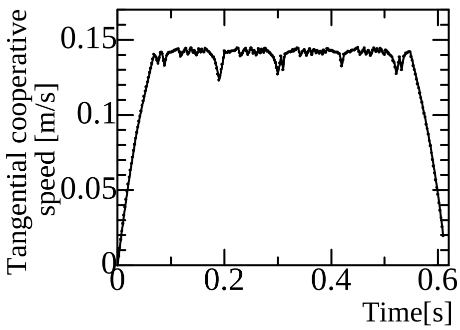

Figure 4.12: Tangential Cooperative Velocity Profile for S-Shaped Locus

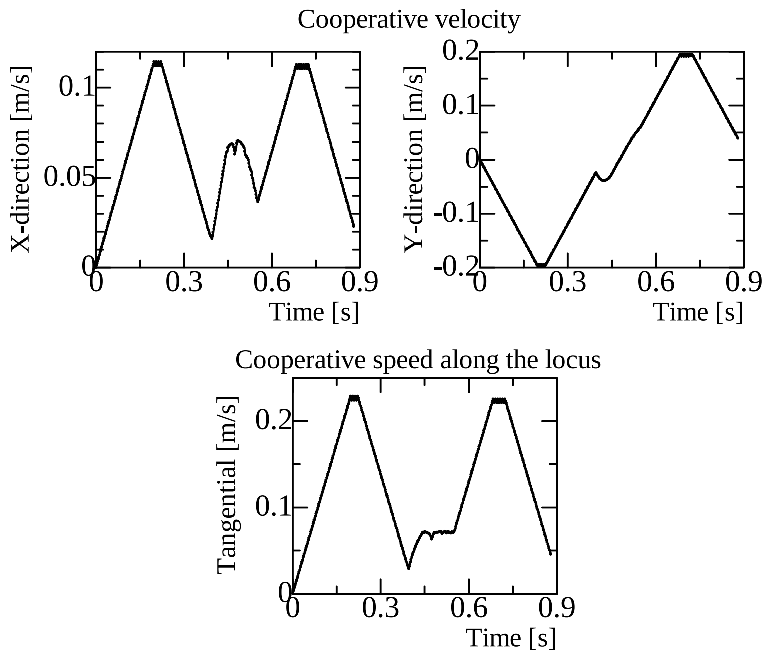

Figure 4.13: Tangential and Cooperative Velocity Profiles for V-Shaped Locus

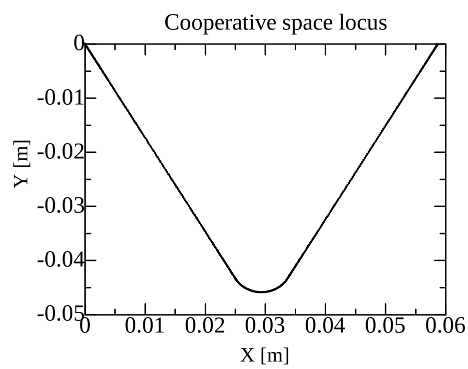

Figure 4.14: V-Shaped Locus in Work Space

Chapter 5

Conclusions and Recommendations

Two key important areas - belt drives and cooperative control to speed up, have been used to demonstrate the trajectory planning scenarios for servo controlled objects. The overall goal of this thesis is to develop a feasible technique for vibration restraint accurate position control of belt driven machines under high-speed operations and to synthesize advanced trajectory planner for cooperative control of two robot arms. The understanding and the applications of the proposed methods are realized substantially by this work. Although a brief summery of each chapter is provided at the end of the respective chapter, this chapter deals with comprehensive discussions of the overall work in chapterwise. Possible future research directions that can be originated from this study are also included in this chapter.

5.1 Conclusions

The thesis covers the theoretical developments on

1. Model construction of a belt driven machine,
2. Management of computational complexity of the cooperative trajectory planning through short-listing criterion,
3. Derivation of minimal acceleration forth order spline scheme and,
4. Philosophical notions and mathematical analysis on minimum time cooperative trajectory planning

to generate

1. Accurate high-speed position control of bet drives with vibration restraint,
2. Complementary trajectories of two articulated robot arms for a specified objective cooperative trajectory under fair task distribution strategy subjected to joint acceleration constraints and,
3. Minimum time complementary trajectories of two Cartesian robots for a given curvy locus under maximum joint acceleration constraints.

The effectiveness of the proposed methods was verified with simulation studies followed by solid confirmation through experiments. The proposed vibration restraint techniques were implemented and experimentally tested with an actual belt driven machine and the cooperative trajectory planner for articulated robots was instrumented with two industrial robot manipulators Performer MK2 and MK3. The performance improvement by the proposed trajectory planners was demonstrated in terms of task completion time, error profiles and tracking profiles. Implementation friendliness feature of all above three approaches makes them conveniently incorporated to the present industrial servo systems and it is one of the vital characteristics of the proposed techniques.

In **chapter 2**, stepwise derivation of an accurate model integrating the belt reaction torque was discussed. Design of a feed forward dynamic compensator to restrain vibrations and to overcome delay dynamics has been reviewed.

Decomposition of objective cooperative trajectory between two articulated robot arms based on the complementary action, in view of increasing the speed of operation by exceeding the dynamic bounds of individual robot, was investigated in **chapter 3**. Specifications for the cooperative trajectory to be achieved such as maximum cooperative speed and maximum cooperative acceleration, determined by the application itself are inputs to the system and fair task distribution becomes an integral part of control objective under limited acceleration of joints. Two stage trajectory planning paradigm has been devised to reduce the time and space complexity, intact with the required level of accuracy. Segment level planning while optimizing the cost function on regular task distribution, generates the trajectory through the knot points and the short-listing criteria has been exploited as a means of resolving insurmountable computational overhead to a manageable level. Optimum forth order spline scheme derived interpolates the knot points such that the acceleration of the entire trajectory is kept at a minimum while guaranteeing the guidance through via points.

Chapter 4 contemplates on a time optimal cooperative control of two Cartesian robots along a given curvy path under the maximum joint acceleration constraint. A bang-bang control scheme in acceleration has been proposed to realize minimum time in cooperative control. Philosophical aspects behind the proposed algorithm were disclosed and necessity of the stepping back to avoid no solution condition was mathematically justified. Fairness in task distribution at each planning stage was ensured through the conformation of even task assignment constraint.

5.2 Significant Remarks

At the end of the thesis, the author would review his three-year Ph.D. work in Nakamura laboratory, Department of Advanced Systems Control Engineering, Graduate School of Science and Engineering, Saga University, Japan.

When he first entered this laboratory, his supervisor Prof. Masatoshi Nakamura gave him several suggestions and recommendations. The deepest impression among them was that, his researches were not only standing on theoretical research, but also were combining with applications, especially paying much attention on collaboration and cooperation with the industry. The author participated quarterly research discussions on mechatronics with Yasakawa Company and particularly with Prof. Kyura, Department of Electrical Engineering, Kinki University (in Kyushu). It could understand the currently existing problems in the industry. The fruitful discussions imparted a valuable insight to practical perspectives and exposed the author to real industrial issues on control and manufacturing. The knowledge gained through these research discussions immensely helped to carefully identify the real research problems and to develop the solution strategies proposed in this thesis.

A person whom the supervisor Masatoshi Nakamura suggests to memorize is the founder of the Cybernetics-**Norbert Wiener**. His books [86]-[88] equipped the author with not only basic control knowledge, but also the Wiener's considerations of the way to develop research, especially the research in control field. Some excerpts in Wiener's book [86] are noteworthy to indicate here, particularly the following paragraph in introduction.

A century ago there may have been no Leibniz, but there was a Gauss, Faraday and a Darwin. Today there are few scholars who can call themselves mathematicians or physicists or biologists without restriction. A man may be a topologist or an acoustician or a coleopterist. He will be filled with a jargon of his field, and will know all its literature and its ramifications, but, more frequently than not, he will regard the next subject as something belonging to his colleague three doors down the corridor, and will consider any interest in it on his own part as unwarrantable breach of privacy.

After Wiener made the above sentences in his book, more than thirty-five years has been past. We are now encountering more and more complex control problems. As Wiener pointed out, researchers should master various disciplines instead of restricting in one field. We must know the current scientific development that is being undertaken by the researcher *three doors down the corridor*, otherwise,

If a physiologist who knows no mathematics works together with a mathematician who knows no physiology, the one will be unable to state his problem that the other can manipulate, and the second will be unable to put the answers in any form that the first can understand.

To achieve a successful cooperation for researchers from different fields, Wiener commented
The mathematician need not have the skill to conduct a physiological experiment, but he must have the skill to understand one, to criticize one, and to suggest one. The physiologist need not to be able to prove a certain mathematical theorem, but he must be able to grasp its physiological significance and tell the mathematician for what he should look.

The author feels very lucky to have a chance to get the know-how of research and to learn how to start cracking research problems by bridging the existing gaps in the field of research as well as many other disciplines. Most importantly, he has learnt the way to extend research work, which will be the fundamental background for his future study.

5.3 Recommendations for Further Developments

This section considers the state of art in robotics in light of cooperative requirements cum control objectives and thereby draws the potential futuristic directions. The field of robotics is still in its infancy, especially in cooperative behavior though the robots have become economically feasible for a number of applications. Robots are still slower than human in many tasks, but are already better than human in certain tasks requiring high positional accuracy. Therefore, the research on robot speed is being addressed on several fronts. Limitations on computational speed are being addressed by steady improvements in microprocessor technology, by multiprocessor architectures and by special purpose hardware for dynamic, kinematic and control computations. Despite these advances in computability control system performance is increasingly limited by speed of computation. Coordinative behavior in cooperative trajectory planning in view of boosting up the speed is another consideration. However, the amount of research being done in strict cooperative control is disproportionately small in comparison with its importance to increase the execution pace.

Narrowing down the general scope and focusing the attention directly onto the work covered by this thesis, I have achieved several objectives that are valuable for industrial articulated robot arm applications in the industry and belt driven machines control. However, the proposed new techniques open the room for many types of sophistications and few of such directions can be stated in the recommendation for further developments in extending this thesis work.

5.3.1 Belt driven machine

1. Especially at low gear ratio of the belt driven machine, change in load inertia may seriously affect the position control of belt drives under high-speed operation. In case of a belt driven robot, variation of inertia with the configuration of robot plays a major role in deteriorating the following trajectory. However, the belt driven machine used for experiment, works in a horizontal plane and hence such variation in inertia does not experience. Further, since the controller COSMOS takes the constant load inertia and motor inertia as control inputs specified in a data file into account, in generating the control torque according to the resident control algorithm, COSMOS cannot directly accommodate variable inertia of belt driven robots. Therefore, designing a new controller capable of accepting variable inertia, or devising some techniques to make the controller much more robust over variable inertia without sacrificing the vibration restraint and accurate positioning are open questions for the future.
2. The concept proposed for belt driven machine is tested with one degree of freedom. A systematic extension can be made for multi-axis robot with decoupled joint dynamics. However, dynamics cannot be assumed as decoupled in nature, principally at low gear ratios. Development of coupled dynamic model for multi-axis robot is a potential research theme yet to be addressed. However, power transmission introduces a number of factors, including static friction, binding wear, backlash, and cogging that make it difficult to model the motion produced. The proposed feed forward dynamic compensator is still valid in such a coupled dynamical control scheme too.
3. Basic motive for the use of belt drives is to make the robot arm light weighted. Some low inertia meal assistant robots work on belt drives, but for such applications, some sophistication should be made with an explicit force control scheme to guarantee that the force at the end effector does not exceed the predetermined

value since the safety of the human beings involved is paramount.

5.3.2 Cooperative trajectory planner for a given objective trajectory

1. The proposed trajectory-planning scheme is derived on the principles of kinematics and hence direct control of force exerted does not under deliberate control. So the application is limited to non-force interactive control applications such as laser cutting, contour welding and spraying. In order to extend the trajectory planner beyond this level, force control in addition to position control is mandatory. Developments of new primitives for describing motion and the design of control systems capable of executing them directly through embodiment of force control scheme to the proposed algorithm has to be tried as a future direction. When adding up such key feature, a closed loop force control with sensory feed back is highly recommended, because it provides much flexibility and thus restructures to a much general platform. Number of sophistications can be further added through sensory feed back and thereby the trajectory-planning scheme can be upgraded enabling to operate in unknown and ever changing environments.
2. The cooperative trajectory planning under strict coordination is achieved based on the complementary principle. However, when extending it to cooperative control of multiple robots, complementary principle is no more applicable and another task distribution criterion along with optimization technique must be instrumented. Therefore a complete revision of the algorithm is demanded when extending it to multiple robots and the systematic adaptation of the proposed method is hard to be realized.
3. Since the proposed algorithm is established on the search principle for non-terminating solution while minimizing the difference in maximum joint velocities of two robots, the generation of results is not deterministic in time. Besides, computation complexity involved with inverse kinematics and kinematics makes the algorithm slow in execution time. Therefore, real time realization is not possible in the present form and therefore it is used as an off-line planning technique since off-line programming can absorb the real time intolerable time lags. Certain level of perfection in computational time can be achieved by contriving a distributed computing architecture with parallel computation. However, construction of such system is a systematic project conceding substantial amount of resources. Therefore a development of a separate real time algorithm serving the same planning objective seems to be more effective and still to be exploited.
4. Since the short-listing criterion restricts the search scope for solution as a means of reducing the execution time of planning the cooperative trajectory, it does not always guarantee a global optimum solution, but a sub-optimal solution. Increasing the tolerance to cover a wider scope of the solution space and thereby looking for the highly probable global optimum is a self-defeating attempt as it exponentially increases the computational time. In the proposed approach, single robot solution is assumed to be the initial non-terminating solution to probe for optimal solutions at its vicinity. Shifting the initial non-terminating

solution in some methodical way, in search of other sub-optimal solutions is a possible direction for global solution with the expense of computational time. Reformation of the existing algorithm or inventing a more effective analytical algorithm for global optimum solution is ahead for future work.

5.3.3 Minimum time cooperative trajectory planner of two Cartesian robots under given objective locus

1. The proposed minimum time cooperative trajectory planner is of bang-bang control in acceleration profile, oscillating the acceleration between maximum limits, at least of one joint of each robot is inevitable at certain points along the trajectory. If the acceleration limit were adequately high, it would develop a substantial jerk particularly at low sampling time. Consideration of jerk constraint for minimum time trajectory planning is one option. Changing the cooperative planning objective to near time optimal might be another feasible solution to resolve higher jerk levels. Incorporation of jerk limits is an improvement to be added to enhance the performance of corporative control.
2. Through parallel computing architectures backed with super power computational hardware resources, computational time can be dramatically reduced. But the level of stepping back is unpredictable and it depends on case by case. Therefore, guaranteed real time performance could not be apprehended even with expedited architectures and better resources due to intrinsic drawback of stepping back mechanism within the algorithm. Algorithmic revision for real time is a timely research direction.
3. The proposed planner is basically developed for structured environment where the motion and the initial locations are a priori known. More accurate error free sensing systems embodied in the trajectory planning with time debugging and error recovery capabilities are potential advancements that can be made to the proposed planner, such that it can suitably appropriate for dynamic and insufficiently known environments. Robot work cell configurability is another significant issue in the design of new controller software and hardware to permit sensors and new motion primitives be easily integrated into the system. However, such crucial amendments require a considerable research effort and I believe that it may be realizable in near future.
4. The following research themes and directives are emerged from the thesis work on minimum time cooperative control of two robots as prospective research direction.
   - Expand to multirobot trajectory planning under minimum time criterion
   - Subjugation of an issue of force interaction in cooperative control of two robots, in view of speeding up the operation
   - Integration of smart and intelligence characteristics so that planning can be done in unknown environments and dynamically changing environments during task execution for autonomous cooperative purposes
   - Exploitation of teleoperated cooperative control possibilities in minimum time for rescue, undersea and space explorations.

Appendix A

Coordinate Systems and Transformations

A.1 Coordinate Systems for Spatial Description

Any point in space can be described by a coordinate system irrespective of its type of definition *i.e.* Cartesian, spherical or any other. Selection of a fixed point in space termed as origin is a prerequisite to define a coordinate system. The way of measurement made from the origin determines the type of coordinate system and a minimum of three parameters is mandatory to uniquely indicate a point in space. However, the discussion is confined to Cartesian coordinate system, which is characterized by a mutually perpendicular right hand set composed of three axes originated from the origin, along which the measurements are made. It is frequently used to represent a point in space.

Though the translational location that is measured along the axis is adequate to describe a point in space completely, it is not sufficient to describe an object since an object has the freedom to rotate and eventually to give a new configuration. Therefore, an object in the space can be necessarily and sufficiently described with two aspects *i.e.* position and orientation. Considering the above two ascriptions of an object together, it is termed as pose. Position is the translational location of a point attached to an object or to an end effector and is usually expressed with respect to a base frame. Orientation is the rotational location of an object or of an end effector with respect to a reference coordinates system.

To describe the orientation of an object relative to the reference coordinate system that is already being defined for the representation of a point in space, another frame or a coordinate system, whose origin is a fixed point on the object, is attached to the object and the description of this coordinate system is given relative to the reference system. Since any orientation could be achieved through a sequence of rotations around three mutually perpendicular axes, a rotation vector with three tuples can uniquely express orientation. Representation of Euler angle illustrates this scenario of rotation sequence, mathematically.

A.2 Mathematical Representation and Operations

Now, the depiction of pose of an object has been reduced to a mathematical problem, which explains the relative arrangement of two coordinate systems. Translation and rotation are two fundamental applicable operations for two coordinate systems.

Translation can completely express the relative positions of the origins of the two coordinate systems whereas rotation describes the misalignments of the respective coordinate axes when two origins coincide with each other.

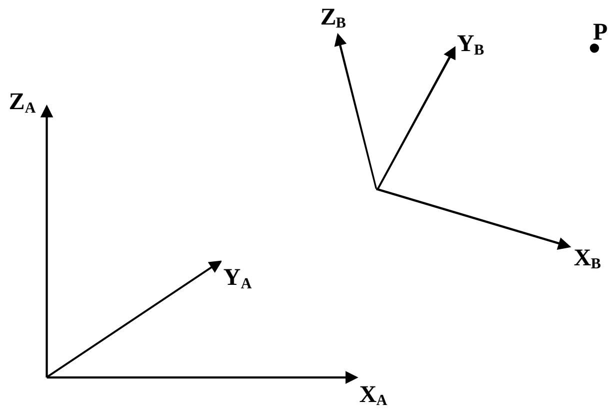

Figure A.1: Translation and Rotation of Coordinate Systems

Referring to Fig.A.1, the position of point P is represented with respect to frame A and frame B. The transformation of coordinate systems in pose can be mathematically expressed in the form of matrix and given by

$$\begin{bmatrix} {}^{A}P \\ 1 \end{bmatrix} = \begin{bmatrix} & {}^{A}_{B}P & & {}^{A}P_{Borg} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^{B}P \\ 1 \end{bmatrix} \tag{A.1}$$

where ${}^{A}P_{Borg}$ and ${}^{A}_{B}P$ denote position of B's origin with respect to frame A and rotation matrix whose values are determined by

$$\begin{bmatrix} {}^{A}_{B}R \end{bmatrix} = \begin{bmatrix} \hat{X}_B\hat{X}_A & \hat{Y}_B\hat{X}_A & \hat{Z}_B\hat{X}_A \\ \hat{X}_B\hat{Y}_A & \hat{Y}_B\hat{Y}_A & \hat{Z}_B\hat{Y}_A \\ \hat{X}_B\hat{Z}_A & \hat{Y}_B\hat{Z}_A & \hat{Z}_B\hat{Z}_A \end{bmatrix} \tag{A.2}$$

of which $[\hat{X}_k, \hat{Y}_k, \hat{Z}_k]$ denotes the direction vectors of frame $k = A, B$. The expression given in A.1 is referred to as homogeneous transform and it is quite computationally intensive as the multiplication is associated with a 4X4 matrix.

A robot can be considered as an open kinematic chain and hence this homogeneous transformation could be directly applied to find the pose of the object under manipulation. A three dimensional orthogonal coordinate system is basically defined such that the origin of the coordinate system lies on the base of the robot and it is designated as the base coordinate system. A coordinate frame for each link can be conveniently defined using DH notation and the pose of the object or the end effector could be comprehensively determined by successive application of homogeneous transform.

Appendix B

Space Transformation

B.1 Forward Kinematics and Inverse Kinematics

Kinematics is the science of motion that treats motion without regard to the forces that cause it. Therefore, in forward and inverse kinematics, forces are not considered. These are used to transform spatial description in position and orientation of the end effector-pose from workspace to configuration (joint) space and vise versa.

Forward kinematics or direct kinematics calculates the end effector pose in Cartesian coordinate description for a given set of joint positions. In more precise terms, for a specified configuration vector, forward kinematics calculates the position alone or the position and orientation of the tool frame with respect to the base frame. This relationship yields a unique solution.

Given the position and orientation of the end effector of the manipulator, inverse kinematics calculates all the possible set of joint angles which could be used to attain this given position and orientation. Inverse kinematic solutions can be grouped into closed form solutions and numerical solutions, which are computationally more expensive and slower. Derivations of closed loop solutions are based on either algebraic methods or geometric methods.

A manipulator is defined to be solvable, if all the sets of joint variables for a given position and orientation can be determined by an algorithm either numerically or analytically. Many manipulators have multiple solutions for a single configuration. Peiper [88] has shown that a general 6 degrees of freedom manipulator does not have a closed form solution. Besides, no solution conditions prevail at singularities in workspace; inverse kinematics is not convenient as forward kinematics, especially at higher degrees of freedom. Even if inverse kinematic solutions are found, they may not be physically realizable due to joint angle limitations.

Basically, these relationships are established considering either geometry (confined to limited cases), or transformation of coordinate systems attached to each link (much generic approach). Adoption of Denavit-Hartenberg convention for the assignment of coordinate frames on robot links is popular, since it immensely helps for convenient transformation of coordinate frames.

The following figure depicts the definition of axis for PERFORMER MK3 robot. The rest of this section is devoted to establish the forward and inverse kinematic relationship in 3D considering the position only.

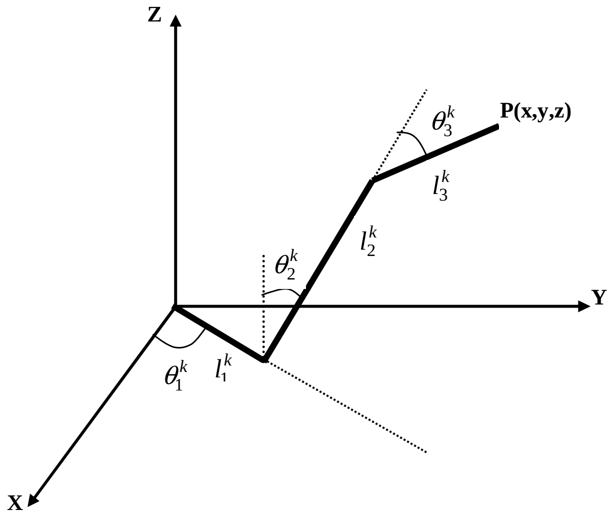

Figure B.1: Diagrammatic Representation of Robot's Link Structure

Referring to the above figure forward kinematic relationship can be expressed by

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} l_1^k cos(\theta_1^k) + l_2^k cos(\theta_1^k) sin(\theta_2^k) + l_3^k cos(\theta_1^k) sin(\theta_2^k + \theta_3^k) \\ l_1^k sin(\theta_1^k) + l_2^k sin(\theta_1^k) sin(\theta_2^k) + l_3^k sin(\theta_1^k) sin(\theta_2^k + \theta_3^k) \\ l_2^k cos(\theta_2^k) + l_3^k cos(\theta_2^k + \theta_3^k) \end{bmatrix} \tag{B.1}$$

where $[x, y, z]^T$, $[\theta_1^k, \theta_2^k, \theta_3^k]^T$ and l_i^k denote the position vector in Cartesian space, the configuration vector in joint space and the length of link i of k^{th} robot. The inverse kinematics for the same is described by

$$\begin{bmatrix} \theta_1^k \\ \theta_2^k \\ \theta_3^k \end{bmatrix} = \begin{bmatrix} tan^{-1}(y/x) \\ tan^{-1}(L/z) - cos^{-1}\left[\frac{L^2+z^2-(l_2^k)^2-(l_3^k)^2}{2l_2^k\sqrt{L^2+z^2}}\right] \\ cos^{-1}\left[\frac{L^2+z^2-(l_2^k)^2-(l_3^k)^2}{2l_2^k l_3^k}\right] \end{bmatrix} \tag{B.2}$$

where $L = \sqrt{x^2+y^2} - l_1^k$. The length of the first link could be regarded as the offset between first two links of the robot manipulator.

By differentiating B.1, mapping between the velocities in work and configuration space can be determined as given by

$$\begin{bmatrix} \dot{x}_1^k \\ \dot{y}_2^k \\ \dot{z}_3^k \end{bmatrix} = [J] \begin{bmatrix} \dot{\theta}_1^k \\ \dot{\theta}_2^k \\ \dot{\theta}_3^k \end{bmatrix} \tag{B.3}$$

where $[J]$ is referred to as Jacobian and its value is expressed by

$$[J] = \begin{bmatrix} -s_1(l_1^k + l_2^k s_2 + l_3^k s_{2,3}) & c_1(l_2^k c_2 + l_3^k c_{2,3}) & l_3^k c_1 c_{2,3} \\ c_1(l_1^k + l_2^k s_2 + l_3^k s_{2,3}) & s_1(l_2^k c_2 + l_3^k c_{2,3}) & l_3^k s_1 c_{2,3} \\ 0 & -l_2^k s_2 - l_3^k s_{2,3} & -l_3^k s_{2,3} \end{bmatrix} \tag{B.4}$$

in that, $s_i = sin(\theta_i^k)$, $c_i = cos(\theta_i^k)$, $s_{i,j} = sin(\theta_i^k + \theta_j^k)$ and $c_{i,j} = cos(\theta_i^k + \theta_j^k)$.

Appendix C

An Overview of Robot Manipulator System

C.1 Schematic Representation of Robot System

Schematics of a typical industrial robot manipulator (Performer MK3) are

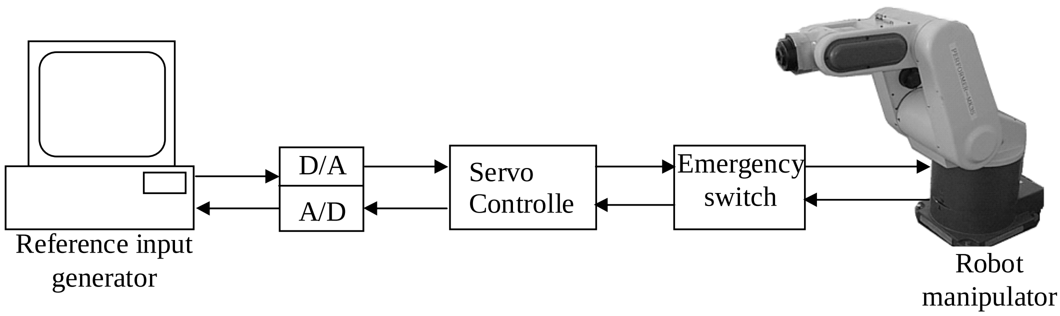

Figure C.1: Schematics of a Typical Robot Manipulator System

depicted in Fig.C.1. Basically, an industrial robotic system is composed of reference input generator, data interface, servo controller, position and velocity sensors, mechanical structure of robot manipulator along with actuators, emergency switch and connecting wires. Reference input generator can be a standalone computer, or preferably a computer connected to a network. Data interface converts digital data generated by reference input generator into its analog format so that it could be used by servo controller and to acquire analog sensor data of the robot manipulator through servo controller. Each joint of the robot manipulator is equipped with an actuator, usually dc or ac servomotor, and sensors to measure position and velocity. Connecting wires establish the electrical connectivity while the emergency button is for safety and it overrides all the operating functionalities.

C.2 Specifications of a Typical Industrial Robot

For a given vector of reference data $[u_1, u_2, u_3]^T$, P controller generates corresponding velocity input, accounting the last configuration vector of the joints as specified by position sensors and consequently fed it to servo controller for servo operation of robot actuators. The coding architecture explains the control of a single decoupled servo joint in detail and Table C.1 shows the set of control parameters for Performer MK3 robot. For simplicity, second order model for servo system has been assumed to illustrate the coding architecture, but it is not necessarily be the second order model (please refer Appendix-D for details).

Table C.1: Control System Parameters

Parameter	Denotation	Value
Shaft encoder resolution	N	8192[pulse/rev]
Gear ratio @joint 1	N_1	120
Gear ratio @joint 2	N_2	160
Gear ratio @joint 3	N_3	160
Conversion factor of velocity pulse to voltage for joint 1	R_1	0.000015161 [volt/(rad/s)]
Conversion factor of velocity pulse to voltage for joint 2	R_2	0.000012484 [volt/(rad/s)]
Conversion factor of velocity pulse to voltage for joint 3	R_3	0.000012169 [volt/(rad/s)]

C.3 Coding Architecture of a Revolute Joint

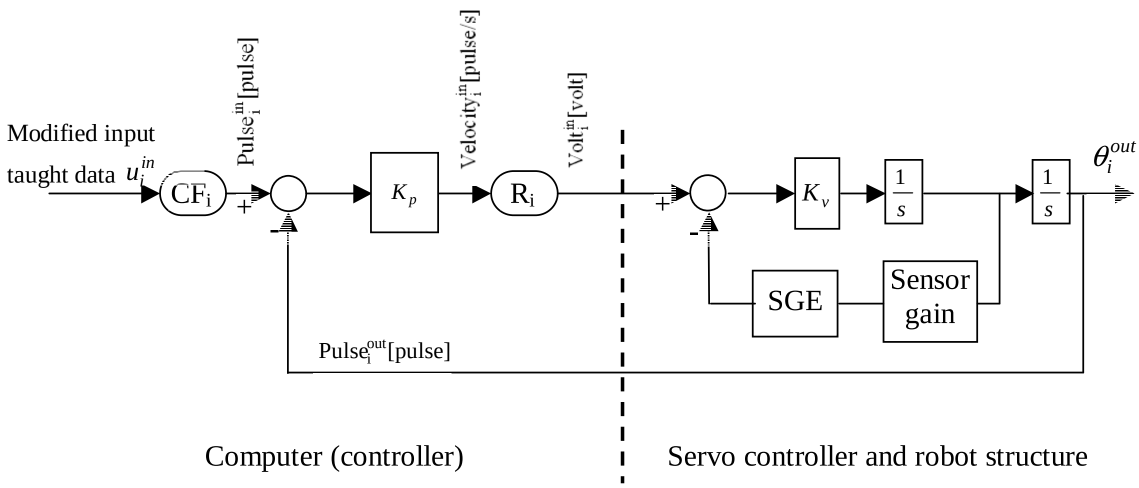

Figure C.2: Coding Architecture of a Single Decoupled Servo Joint

The modified input taught data $u_i^{in}[rad]$ is converted to $Pulse_i^{in}[pulse]$ by conversion factor $CF_i[pulse/rad]$ for the i^{th} joint. The most of the control actions take place in pulse form to comply with the existing pulse counter. The reference pulse input $Pulse_i^{in}[pulse]$, is compared with the reading of the pulse counter $Pulse_i^{out}[pulse]$ so as to generate position error. The position loop gain of the i^{th} joint transforms this position error into velocity input $Velocity_i^{in}$ expressed in pulse per second. The coefficient R_i converts $Velocity_i^{in}$ into a quantity in voltage $Volt_i^{in}[volt]$, which lies within the acceptable bounds of input to the servo pack.

Velocity sensor connected to the i^{th} joint reads the output velocity of the joint and sensor gain enhancer (SGE) conditions this signal to be compatible with voltage input $Volt_i^{in}[volt]$, so that the difference could be used to generate the velocity error at joint i. Velocity loop gain transforms velocity error into acceleration and two successive integrators convert acceleration into output position of joint i, θ_i^{in}.

The value of the conversion factor CF_i corresponding to i^{th} joint governs as

$$CF_i = \frac{N * N_i}{2\pi} \tag{C.1}$$

where N and N_i take the usual meaning as specified in Table C.1.

Appendix D

Famous Kinematic Models of Robot Systems

Path planner gives the description of the path positions without timing and velocity bounds, from which trajectory planner calculates its timing concerning the dynamics of the manipulator so as to give time history of positions and time derivatives. Manipulator characteristics could be represented either with a dynamic model or with a kinematic model. In dynamic models, joint torques and forces are taken into account while in kinematic models; no concern is paid to forces but resulting positions, velocities and accelerations. Though the dynamic constraints are natural in existence, they can be transformed into equivalent kinematic constraints (constant or piecewise constant bounds), concerning the supremum (global minimum of maximum) of accelerations and velocities despite hampering 100% efficiency and capabilities of manipulators.

A detailed explanation of dynamic models is furnished in [43]. However, this discussion is basically focused on popular and well-known kinematic models used for robotic systems. Depending on the operational speed of the servo system and the degree of influence of the simultaneous movements of other joints on a particular joint, the selection of kinematic model may vary according to the accuracy required. On the assumption that the other joints have negligible effect, the joint dynamics can be decoupled and as a result planning of trajectories will be easy.

D.1 First Order Kinematic Model

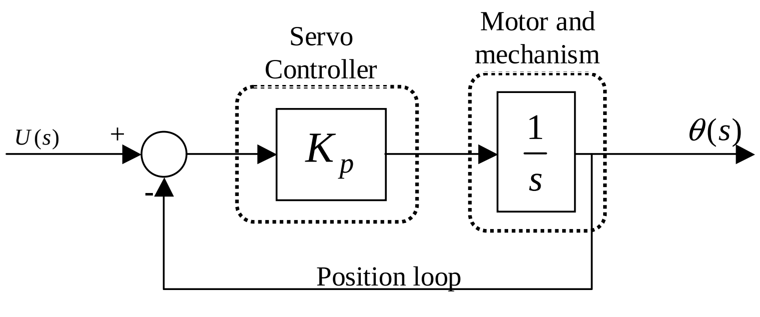

Figure D.1: First Order Representation of Mechatronic Servo System

A decoupled joint of a mechatronic servo system can be mathematically represented with a first order linear kinematic model as follows:

$$G(s) = \frac{K_p}{s + K_p} \tag{D.1}$$

where K_p is the position loop gain of the servo controller and it is briefly illustrated in Fig.D.1. The first order model can be used when the velocity of the servo motors as the actuators of the mechatronic servo system is about 1/100 times of the rated velocity of motors [15].

D.2 Second Order Kinematic Model

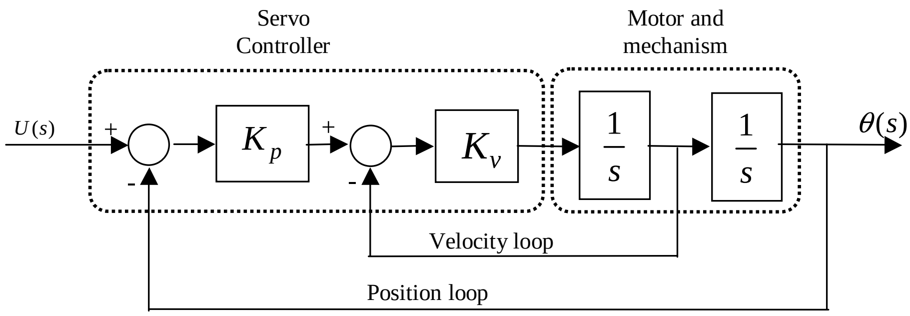

Figure D.2: Second Order Representation of Mechatronic Servo System

When the velocity of the motors are much faster, about 1/20 1/100 times of the rated velocity, the kinematics of the mechatronic servo system must be treated as a second order model. The mechatronic servo system is described independently for each axis with the second order model given below.

$$G(s) = \frac{K_p K_v}{s^2 + K_v s + K_p K_v} \tag{D.2}$$

where K_v is the velocity loop gain as shown in Fig.D.2.

D.3 Fourth Order Kinematic Model

For higher velocities greater than 1/20 times of the rated speed, in other words high-speed operation of servo control joints, second order model can not represent the kinematics of joint adequately, as the reaction force/torque is considerably high and the elastic deformation can not be further neglected. In such cases, fourth order representation of the decoupled servo joint can precisely illustrate the characteristics and it is briefly shown in Fig.D.3.

Fourth order model of independent servo joint can mathematically expressed in the following form:

$$G(s) = \frac{a_0}{a_4 s^4 + a_3 s^3 + a_2 s^2 + a_1 s + a_0} \tag{D.3}$$

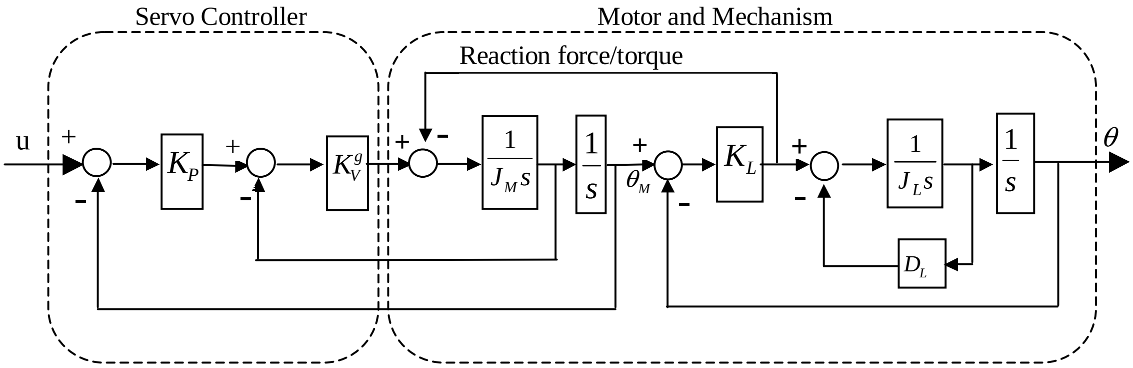

Figure D.3: Fourth Order Representation of Mechatronic Servo System

where $a_0 = K_p K_L K_v^g$, $a_1 = K_L K_v^g + K_p D_L K_v^g$, $a_2 = K_L J_M + D_L K_v^g + K_p J_L K_v^g + K_L J_L$, $a_3 = D_L J_M + K_v^g J_L$ and $a_4 = J_L J_M$. In that case K_p, K_L, K_v^g, J_M and J_L denote position loop gain, elastic constant of the mechanism, servo amplifier gain, damping coefficient of the mechanism alone with load, motor inertia and load inertia respectively. As the load inertia is a function of the manipulator configuration, the payload and the shape of the payload, fourth order model is not a time invariant model.

When the joints move simultaneously, due to many functions including frictional, gravitational and inertial forces of the manipulator, one joint may be influenced by other joints. Therefore each joint can not be dissociated from the others and therefore, model should be developed concerning the complete dynamics of the manipulator; not a single joint. Since, interactive joint model is highly dependant on the number of joints, the complexity proportionately increases as the number of joints is increased. Due to unavailability of generic model, interactive joint model is beyond discussion here.

Appendix E

Generation of Minimum Time Trajectory for a Single Robot

The S-shaped objective locus is basically composed of two circular arcs with equal radii, one appended below the other. Therefore geometrically, the objective locus is divided into two sections. Trajectory planning is carried out in forward and backward directions along the two sections and finally they are merged into a single trajectory. The following steps will concisely explain the algorithm employed to plan the minimum time trajectory of a single robot.

1. Parameterize the objective locus
   The Cartesian coordinates of the objective locus can be related to a parameter λ as:
   $x = p_j + r_s \cos\lambda$
   $y = q_j + r_s \sin\lambda \qquad$ for $j = 1, 2$
   where (p_j, q_j) is the instantaneous center of the circular arc corresponding to the j^{th} segment and r_s is the radius of the circular arc.
2. Calculate the parameter values corresponding to initial and final positions of the each section of the locus.
3. The following sequence of steps are reiterated for one section in the forward direction and in the reverse direction for the other.
   (a) Set a trial increment of parameter $d\lambda$ during the sampling time T_s.
   (b) Calculate the new parameter value $\lambda_{new} = \lambda_{old} + d\lambda$.
   (c) Calculate the new Cartesian position of the end effector (x_{new}, y_{new}).
   (d) Apply inverse kinematics and obtain the new joint positions $[\theta_1^{new}, \theta_2^{new}] =$ Inversekinematics(x_{new}, y_{new}).
   (e) Calculate the joint velocities and end effector velocity
   $\dot{\theta}_j^{new} = (\theta_j^{new} - \theta_j^{old})/T_s \qquad$ for $j = 1, 2$
   $v_{new} = \sqrt{(x_{new} - x_{old})^2 + (y_{new} - y_{old})^2}/T_s$.
   (f) Calculate the joint accelerations
   $\ddot{\theta}_j^{new} = (\dot{\theta}_j^{new} - \dot{\theta}_j^{old})/T_s \qquad$ for $j = 1, 2$.
   (g) Determine the indices corresponding to joint acceleration of two joints and end effector velocity as:
   $index1 = |\ddot{\theta}_1^{new}|/\ddot{\theta}_{max}$
   $index2 = |\ddot{\theta}_2^{new}|/\ddot{\theta}_{max}$
   $index3 = |v_{new}|/v_{max}$.
   Then assign the maximum of these three indices as the index account for $d\lambda$

modification
$index = max\{index1, index2, index3\}$.

(h) If $index \notin [1-\delta, 1+\delta]$ then bisection method is applied to determine new value of $d\lambda$ repeating steps b) through g), else move to step i). The term δ gives the tolerance limit of index under consideration.

(i) Update the old parameter value with the new value ($\lambda_{old} = \lambda_{new}$) and repeat steps a) through h) in order to plan the subsequent trace of the trajectory corresponding to ensuing sampling time, until new parameter value taken in step b) exceeds the limiting value of the section.

4. Concatenate the trajectories planned for each geometrical section of the locus.

Appendix F

Glossary of Terms

Accuracy	The ability of a robot to reach a given point in space repeatedly and how far off it will be in the worst case. Alternatively, the precision with which a computed point can be attained or measured.
Actuator	A transducer that converts electrical, hydraulic, or pneumatic energy to cause motion of the robot.
Analytical methods	Purely mathematical methods that do not require iteration.
Articulated manipulator	A manipulator with an arm that is broken into sections (links) by one or more joints. Each of the joints represents a degree of freedom in the manipulator system and allows translational and rotary motion.
Axis	A direction used to specify the robot motion in a linear or rotary mode.
Base coordinates	The three dimensional orthogonal coordinate system which has the origin on the robot base.
Cartesian coordinates	A means for giving the location of a point in space by measuring its distance from a reference point along straight lines. The lines are at right angles to each other and all meet at the reference point.
Cartesian manipulator	An arm with prismatic joints which allows movement along one or more of the three- axes in the x, y, z coordinate system.
Closed-form	A problem formulation that does not require iteration for its solution.
Cognitive	A machine that is aware of its surroundings. It is also able to generate its own path plan and modify its task plan as necessary.
Compliance	Displacement of a manipulator in response to a force or torque.
Continuous path controlled robot	A robot whose path is controlled by storing a large number or close succession of spatial points in memory during a teaching sequence.
Controlled path	The path or mode of movement which ensures that the end of the robot's arm will follow a predictable (controlled) path and orientation as the robot travels from point to point.

Coriolis force	The deflecting effect opposite to the direction of rotation when the velocity of the robot arm is constant but the length of the arm is changed. An opposing torque (in the direction of the rotation) must be applied to overcome this force. This can be done by having the robot arm retract while making a swing from one target point to another.
Cybernetics	The science or study of communication and control mechanisms in human and machine systems, including computers.
Degree of freedom	Regardless of the anatomy and type of movement, either rotational or a translational, the number of independent ways in which the end effecter can move. It is defined by the number of rotational or translational axes through which motion can be achieved.
Dynamic model	A mathematical model describing the motions of the robot and the forces that cause them.
Dynamics	The study of motion with regard to forces which cause the motion.
End effector	An accessory device or tool specifically designed for attachment to the robot wrist or tool mounting plate of the last link to enable the robot to perform its intended task.
Forward kinematics	For a given set of joint angles, forward kinematic computes the position and orientation of tool frame relative to the base frame. In other words it transforms joint space (configuration space) description into Cartesian space (task space, operational space or work space) description.
Gantry robot	A robot which has three degrees of freedom along the X, Y, and Z coordinate system. Usually consists of a spooling system (used as a crane) which when reeled or unreeled provides the up and down motion along the Z axis. The spool can slide from left to right along a shaft which provides movement along the Z axis. The spool and shaft can move forward and back along tracks, which provide movement along the Y axis. Usually used to position it's end-effector over a desired object and pick it up.
Hard automation	Automated machinery that is fixed, or dedicated, to one particular manufacturing task throughout its life. Fixed automation is a synonym for hard automation.
Hybrid control	A control scheme in which some directions are controlled by position control law and the remaining directions are controlled by force control law.
Incremental movement	Movement is broken up into very small pieces, and then takes one at a time.
Industrial robot	A reprogrammable, multifunctional manipulator designed to move material, parts, tools, or other devices through variable programmed motions for the performance of a variety of tasks. The principle components are: one or more arms that can move in several directions;

Inverse kinematics	Given the position and orientation of the end effector of the manipulator, inverse kinematics calculate all the possible set of joint angles which can be used to attain this given position and orientation.
Iteration	Repeatedly applying a series of predefined operations to progressively advance towards a solution.
Jacobian	Specify a mapping from velocities in joint space to velocities in Cartesian space.
Joint	A part of the manipulator system which allows a rotation or translational degree of freedom to a link or end-effector.
Joint space	A coordinate system used to describe the state of the robot in terms of its joint states. This is also referred to as configuration space.
Kinematics	Science of motion that treats motion without regard to the forces which cause the motion. Kinematics of manipulators refers to all the geometric and time based properties of the motion.
Lagrange multipliers	A mathematical technique for transforming equality constraints into performance criteria, thus expressing a constrained problem as an unconstrained problem.
Lead through programming	A means of teaching a robot by leading it through the operating sequence with a control console or a hand-held control box.
Link	A rigid part of a manipulator which connects adjacent joints.
Locus	The special description of the end effector without consideration of its timing. The term path has the same meaning.
Manipulator	A robotic mechanism consisting of an arm and an end-effector. It contains a series of segments, jointed or sliding relative to one another, for the purpose and moving objects. The manipulator usually has several degrees of freedom. It includes the arm, the wrist, and the end-effector.
Off-line programming	A means of programming a robot by developing a set of instructions on an independent computer and then using the software to control the robot at a later date.
Open-loop control	A robotic control system in which data flows only from the controller to the mechanism and does not flow from the mechanism back to the controller. This does not allow self-correcting action which can be provided with feedback.
Optical encoder	A detection sensor which measures linear or rotary motion by detecting the movement of markings past a fixed beam of light. This can be used to count revolutions, identify parts, etc.
Orientation	The rotational location of an object or an end effector with respect to the reference coordinate system.
Parameter	A set of constant factors applied to a particular situation.

Path points	Include all the via points plus the initial and final points.
Payload	The maximum amount of weight that a robot can handle without causing harm, or malfunction to the robot under normal operational conditions.
Pendant teaching	The mapping and recording of the position and orientation of a robot and/or manipulator system as the robot is manually moved in increments from an initial state along a path to a final goal state. The position and orientation of each critical point (joints, robot base, is recorded and stored in a database for each way point the robot passes through on its trek toward its final goal. The robot may now repeat the trek on its own by following the path stored in the database.
Point-to-point motion	A type of robot motion in which a limited number of points along a path of motion is specified by the controller, and the robot moves from one discrete point to another rather than in a continuous, smooth path.
Pose	Position and orientation taken together.
position	The translational location of a point attached to an object or to an end effector specified with respect to the base frame.
Position control	A technique which suppresses disturbances that perturb the system from the desired trajectory by calculating the velocity; and positioning necessary to counteract such disturbances.
Precision (estimated parameter)	The precision of an estimate of a parameter in a model is a measure of how variable the estimate would be over other similar data sets. A very precise estimate would be one that did not vary much over different data sets. Precision does not measure accuracy. Accuracy is a measure of the closeness of the estimate is to the real value of the parameter. Accuracy is measured by the average distance over different data sets of the estimate from the real value.
Prismatic joint	A joint of two nested links that slide onto or alongside of each other.
real time system	A real-time system is a system in which the correctness of the system behavior depends not only on the logical results of the computations, but also on the physical instants at which these results are produced. A Real-Time System responds in a (timely) predictable way to unpredictable external stimuli arrivals.
Redundancy	Additional independent variables available than the constraints.
Repeatability	The ability of the manipulator arm to position the end effecter at a particular location within a specified distance from its position during the previous cycle. In other words, repeatability specifies how accurately the manipulator can return to a given exact location.
Revolute joint	The joints of a robot which are capable of rotary motion.

Robot	A robot is a re-programmable multifunctional manipulator designed to move material, parts, tools, or specialized devices through variable programmed motions for the performance of a variety of tasks. In practical usage, a robot is a mechanical device which performs automated tasks, either according to direct human supervision, a pre-defined program or, a set of general guidelines, using artificial intelligence techniques. These tasks either replace or enhance human work, such as in manufacturing, construction or manipulation of heavy or hazardous materials. An automatic device that perform functions normally ascribed to human or a machine in the form of a human (Webster's Dictionary).
Robotics	Science of robots.
Servo-system	A system in which the controller issues commands, motor drives arms, and sensors measure the motions and signal the amount of motions back to the controller. This process is continued until the arm is repositioned to the point requested.
Singularity	A position in the robot's workspace where one or more joints no longer represent independent controlling variables. Commonly used to indicate a position where a particular mathematical formulation fails.
Spatial resolution	The minimum or smallest dimension to which the robot system can define the work space. This resolution determines the smallest error that can be sensed by the robot, as limited by the minimum resolution of the controller or the minimum resolving increment of the servo-system.
Spline	A smooth, continuous function used to approximate a set of functions that are uniquely defined on a set of sub-intervals. The approximating function and the set of functions being approximated intersect at a sufficient number of points to insure a high degree of accuracy in the approximation. This allows a robot manipulator to complete a task without jerky motion.
Statics	The study of forces that do not cause motion.
Trajectory	Refer to a time history of position, velocity and acceleration for each degree of freedom.
Trajectory generation	The computation of smoothly controllable trajectory in multidimensional space which describes the desired motion of a manipulator.
Via point	Intermediate locations or points in space in between the initial and final points through which a manipulator must pass en route to a particular destination.
Work space	The maximum reach space refers to all of the points or volume in space that a robot's end effector can possibly reach. The dexterous workspace is all of the possible points or volume in space that a robot's end effector can reach with an arbitrary orientation.

Publications

A. Refereed International Journal papers

1. T.S.S. Jayawardene, M. Nakamura, and S. Goto, "Accurate Control Position of Belt Drives under Acceleration and Velocity Constraints," *International Journal of Control, Automation and Systems*, Vol.1, No.4, pp. 474-483, December 2002.
2. T.S.S. Jayawardene, M. Nakamura, and S. Goto, "Two-Stage Trajectory Planning Paradigm Based on Kinematics for Cooperative Contouring Control of Two Robot Arms," *Control Engineering Practice*, (submitted in 2005).

B. Proceedings of international conferences

1. T.S.S. Jayawardene, M. Nakamura, S. Goto and N. Kyura, "Cooperative Contour Control of Two Robots under Speed and Joint Acceleration Constraints," in *Proc. of the 2003 International Conference on Control, Automation and System (ICCAS'03)*, pp.1387-1391, Gyeongju (Korea), October 2003.

C. Proceedings of domestic conferences

1. T.S.S. Jayawardene, M. Nakamura, S. Goto and N. Kyura, "Model Construction of a Belt Driven Machine under High-Speed Operation," in *Proc. of SICE Kyushu conference*, pp. 219-222, Oita (Japan), Dec. 2002.
2. T.S.S. Jayawardene, S. Goto, N. Kyura and M. Nakamura, "Coarse to Fine Paradigm for Cooperative Trajectory Planning of Two Robot Arms," in *Proc. of SICE Kyushu conference*, pp. 129-132, Nagasaki (Japan), Nov. 2003.
3. T.S.S. Jayawardene, M. Nakamura, S. Goto and N. Kyura, "Minimum Time Cooperative Trajectory Planner for Two Cartesian Robots," in *Proc. of SICE Kyushu conference*, pp. 19-22, Kitakyushu (Japan), Dec. 2004.

References

[1] http://cache.ucr.edu/∼currie/roboadam.htm

[2] Programmed Article Transfer, US Patent No. 2988237, 1961.

[3] H.A. Ernest, "A-Computer Controlled Mechanical Hand," Sc.D thesis, Massachusette Institute of Technology, Cambridge, 1961.

[4] E. Brussel, *Websters Dictionary*, Prentice Hall, 1988.

[5] http://www.frc.ri.cmu.edu/robotics-faq/1.html

[6] J.J. Craig, *Introduction to robotics: Mechanics and Control*, Addison Wesley, 1989.

[7] http://www.cse.dmu.ac.uk/∼aayesh/teaching/robotics/CSCI2012/lec2012-01-1-2003.ppt

[8] D.E. Whitney, "State of the art and research needs in robot contact sensing," in *workshop on intelligent robots: Achievement and issues*, SRI International Menlo Park, pp. 137-144, 1984.

[9] L. Sciavicco and B. Siciliano, *Modeling and Control of Robot Manipulators*, Springer, 1995.

[10] M.W. Spong, F.L. Lewis and C.T. Abdallah, *Robot control:Dynamic Motion Planning and Analysis*, IEEE Press, Newyork, 1992.

[11] K.S. Fu, R.C. Gonzalez and C.S.G. Lee, *Robotics: Control, Sensing, Vision, and Intelligence*, McGraw-Hill Book Company, 1987.

[12] B.G. Robert and S.A. Bortoff, "A Novel Approach to Vibration Reduction in Flexible Belt Drives," in *Topics in Control Theory and its Application*, pp. 67-80, 1999.

[13] W. Li and X. Cheng, "Adaptive High Precision Control of Positioning Tables-Theory and Experiment," in *IEEE Transactions on Control Systems Technology*, vol. 2, pp. 265-270, 1994.

[14] M. Nakamura, T. Iwanaga, S. Goto and N. Kyura, "High Speed High Accuracy Position Control of Belt Drive Machine with Vibration Restraint," in *Proc. of SICE Kyushu conference*, pp. 359-362, Kumamoto (Japan), December 2001.

[15] S. Goto, M. Nakamura, and N. Kyura, "Modified Taught Data Method for Industrial Mechatronic Servo Systems to Achieve Accurate Contour Control Performance," in *Proceedings of Advanced Intelligent mechatronics*, AIM 97, 1997.

[16] S.R. Munasinghe, M. Nakamura, S. Goto, and N. Kyura, "Pole Selection of Feed Forward Compensators Considering Bounded Control Input of Industrial Mechatronic Systems," in *IEEE Transaction on Industrial Electronics*, vol. 50, pp. 1191-1206, 2003.

[17] S.R. Munasinghe, M. Nakamura, S. Goto, and N. Kyura, "Pole Selection for Modified Data Method for Mechatronic Servo Systems by Considering Bounded Control Input," in *Proc. of SICE Kyushu conference*, pp. 183-186, Kumamoto (Japan), December 2001.

[18] K.M. Lee and C. Rutherford, "Frequency Reshaped Quadratic Control of a Belt Driven Robot," in *Proc. of Advanced Intelligent mechatronics*, AIM 97, 1997.

[19] K.M. Lee and C. Rutherford, "Frequency Reshaped Quadratic Control of a Low-Cost Human Level Performance Belt Driven Robot," in *IEEE transaction on Mechatronics*, vol. 9, pp. 95-110, 1999.

[20] K.G. Shin and N.D. Mckay, "Robust Trajectory Planning for Robotics Manipulators under Payload Uncertainties," in *IEEE Transactions on Automatic Control*, vol. 32, pp. 1044-1054, 1987.

[21] K.G. Shin and N.D. Mckay, "Minimum Time Control of Robotic Manipulators with Geometric Path Constraints," in *IEEE Transactions on Automatic Control*, vol. 30, pp. 531-541, 1985.

[22] L.V. Kavraki, P. Svestka, J.C. Latombe, and M.H. Overmars, "Probabilistic road maps for path planning in high dimensional configuration space," in *IEEE transactions on robotics and Automation*, vol. 12, pp. 566-580, 1996.

[23] S. Lee and H. Moradi, "A real Time Dual Arm Collision Avoidance Algorithm for Assembly," in *Journal of Robotics System*, vol. 18, pp. 477-486, 2001.

[24] L.V. Kavraki, and J.C. Latombe, "Randomized processing of configuration space for fast path planning," in *Proceedings of IEEE international conference on robotics and Automation*, pp. 2131-2138, 1996.

[25] F. Fahimi and C. Nataraj, "Obstacle avoidance for spatial hyper redundant manipulators using harmonic potential functions and mode shape technique," in *Journal of Robotics system*, vol. 20, pp. 23-33, 2003.

[26] T.S.S Jayawardene, "Simulation Support Tool for Routing of Automated Guided Vehicles," M.Sc. Thesis, University of Moratuwa, Sri Lanka, 2002.

[27] J.Y.S. Luh, M.W. Walker and R.P.C. Paul, "Resolved Acceleration Control of Mechanical Manipulators," in *IEEE Transactions on Automatic Control*, vol. 25, pp. 468-474, 1980.

[28] R. Paul, "Manipulator Cartesian path control," in *IEEE transaction on system man and cybernetics*, vol. 9, pp. 702-711, 1979.

[29] S.R. Munasinghe, M. Nakamura, S. Aoki, S. Goto, and N. Kyura, "High Speed Precise Control of Robot Arm with Assigned Speed under Torque Constraint by Trajectory Generation in Joint Coordinates," in *Proc. of SMC 99*, pp. 854-859, 1999.

[30] R. Paul, *Robert Manipulators*, MIT Press Cambridge, 1981.

[31] G. Antonelli, S. Chiaverini, and Giuseppe, "Real-Time End-Effector Path Following for Robot Manipulators Subject to Velocity, Acceleration, and Jerk Joint Limits," in *Proc. of IEEE/ASME International Conference on Advanced Intelligent Mechatronic*, Italy, pp. 452-457, 2001.

[32] D.F. Rogers, *An introduction to NURBS with historical perspectives*, Morgan Kaufmann, 2000.

[33] L.J.D. Plessis and J.A. Snyman, "Trajectory Planning through Interpolation by Overlapping Cubic Arcs and Cubic Splines," in *International Journal of Numerical Methods and Engineering*, vol. 57, pp. 1615-1641, 2003.

[34] K.G. Shin and N.D. Mckay, "A Dynamic Programming Approach to Trajectory Planning of Robotics Manipulators," in *IEEE Transactions on Automatic Control*, vol. 31, pp. 491-500, 1986.

[35] A.J. Cahill, M.R. James, J.C. Kiefer, and D. Williamson, "Remarks on the Application of Dynamic Programming to Optimal Path Timing of Robot Manipulators, " in *International Journal of Robust and Nonlinear Control*, vol. 8, pp. 463-482, 1998.

[36] J.E. Bobrow, S. Dubowsky, and J.S. Gibson, "Time-Optimal Control of Robotic manipulators along Specified Paths," in *The international journal of robotics research*, vol. 4, pp. 3-17, 1985.

[37] K.G. Shin and N.D. McKay, "An efficient robot arm control under geometric path constraints," in *Proc. of IEEE conference on Decision and Control*, San Francisco, 1983.

[38] K.G. Shin and N.D. Mckay, "Selection of Near Minimum Time Geometric Paths for Robot Manipulators," in *IEEE Transactions on Automatic Control*, vol. 31, pp. 501-511, 1986.

[39] J.J.E. Slotine and H.S. Yang, "Improving the efficiency of time optimal path following algorithms," in *IEEE transaction on Robotics Automation*, vol. 5, pp. 118-124, 1989.

[40] S.S. Han, S.B. Chol, J.H. Kim, "Position control of a flexible Granty robot arm using smart material actuators," in *Journal of Robotics System*, vol. 16, pp. 581-595, 1999.

[41] P.S. Ghandi and F.H. Ghorbel, "Closed Loop compensation of Kinematic Error in Harmonic Drives for Precision Control Applications," in *IEEE Transactions on Control Systems Technology*, vol. 10, pp. 759-768, 2002.

[42] B.C.O. Maciel, M.H. Tera, and M. Bergerman, "Optimal Robust Control of Under-actuated Manipulators Via Actuation Redundancy," in *Journal of Robotics System*, vol. 20, pp. 635-648, 2003.

[43] H.G. Sage, M.F. de Mathelin and E. Ostertag, "Robust Control of Robot Manipulators: a Survey," in *International Journal of Control*, vol. 72, pp. 1498-1522, 1999.

[44] N. Xi, T.J. Tarn and A.K. Bejczy, "Intelligent planning and control for multi robot coordination: An event based approach," in *IEEE Transactions on Robotics and Automation*, vol. 12, pp. 439-452, 1996.

[45] J.S. Gu and C.W. de Silva, "Development and implementation of a real

time open architecture control system for industrial robot systems," in *Engineering applications in Artificial Intelligence*, article in press available on www.sciencedirect.com.

[46] F. Caccavale, S. Chiaverini, C. Natale, B. Siciliano, L. Villani, "Geometrically consistent impedance control for dual-robot manipulation," in *Proc. of 2000 IEEE International Conference on Robotics and Automation*, San Francisco, pp. 3873-3878, 2000.

[47] F. Caccavale, C. Natale, B. Siciliano, L. Villani, *Interaction control*, pp. 121-154, Lecture Notes in Control and Information Sciences 270, Springer-Verlag, Heidelberg, 2001.

[48] B. Siciliano and L. Villani, "Six-degree-of-freedom impedance robot control,", in *Proc. of 8th International Conference on Advanced Robotics*, pp. 387-392, Monterey, 1997.

[49] F. Caccavale, C. Natale, B. Siciliano, and L. Villani, "Resolved Acceleration Control of Robot Manipulators: A Critical Review with Experiments," in *Robotica*, vol. 16, pp. 565-573, 1998.

[50] N.J. Koivo and T.H. Guo, "Adaptive Linear Controller for Robotics Manipulators," in *IEEE Transactions on Automatic Control*, vol. 21, pp. 162-171, 1983.

[51] M.W. Spong, F.L. Lewis, and C.T. Abdallah, *Robot Control: Dynamics, Motion Planning and Analysis*, IEEE press,1993.

[52] A.K. Swain and A.S. Morris, "A Unified Dynamic Model Formulation for Robotics Manipulator Systems," in *Journal of Robotics System*, vol. 20, pp. 601-620, 2003.

[53] L. Sciavicco, B. Siciliano, and L. Villani, "Lagrange and Newton-Euler Dynamic Modeling of Gear Driven Rigid Robot Manipulator with Inclusion of Motor Inertia Effect," in *Advanced Robotics*, vol. 10, pp. 317-334, 1996.

[54] M. Brady, J.M. Hollerbach, T.L. Johnson, T. Lozano-Perez, M.T. Mason, *Robot Motion: Planning and control*, MIT press, 1998.

[55] T. Zhang, "High precision contour control of industrial articulated robot arm by Gaussian neural network," PhD dissertation submitted to Saga University Japan, September 2002.

[56] J.Y.S. Luh, M.W. Walker, R.P.C. Paul, "On-line computational scheme for mechanical manipulators," in *Journal. of Dynamics. System. Measurement. And Control*, vol. 102, pp. 69-76, 1980.

[57] J.J. Zhang, Y.F. Lu, B.Wang, "A non-recursive Newton-Euler formulation for the parallel computation of manipulator inverse dynamics," in *IEEE transactions on. Systems Man and Cybernetics,* vol. 28, no. 3, pp. 467-471, Aug., 1998.

[58] S. Dobowsky and T.G. DesForges, "The application of model reference adaptive control to robotics manipulators," in *SME journal of dynamic systems, measurement and control*, pp. 193-200, 1979.

[59] C.S.G. Lee and M.J. Chung, "An adaptive control strategy for mechanical manipulators," in *IEEE transaction on automatic control*, vol. 29, pp. 837-840, 1984.

[60] J.Y.S. Luh and Y.F. Zheng, "Constrained relations between two coordinated industrial robots for motion control," in *International Journal of robotics research*, vol. 6, pp. 60-70, 1987.

[61] S. Hayati, "Hybrid position and force control of multi-arm cooperating robots," in *Proc. of IEEE conference on Robotics and automation*, San Francisco, pp. 82-89, 1986.

[62] F. Caccavale and L. Villani, "An Impedance Control Strategy for Cooperative Manipulation," in *Proc. of International Conference on Advanced Intelligent Mechatronics*, pp. 343-348, 2001.

[63] S. Abrate, "Vibrations of Belts and Belt Drives," in *Mechanism and Machine Theory*, vol. 27, pp. 645-659, 1992.

[64] O. Khatib, K. Yokoi, K. Chang, D. Ruspini, R. Holmberg, A. Casal and A. Baader, "Force Strategies for Cooperative Tasks in Multiple Mobile Manipulation Systems," in *International Symposium of Robotics and Research*, Munich, 1995.

[65] K.S. Chang, R. Holmberg, and O. Khatib, "The Augmented Object Model: Cooperative Manipulation and Parallel Mechanism Dynamics," in *Proc. of IEEE international conference on Robotics and Automation*, San Francisco, pp. 475-477, 2000.

[66] C.R. Carignan, D.L. Akin, "Cooperative Control of Two Robot Arms in the Transport of Inertial Load under Zero Gravity," in *IEEE Transactions on Robotics and Automation*, vol. 4, pp. 414-419, 1988.

[67] K. Nagai, S. Iwasa, K. Watanabe, and H. Hanafusa, "Cooperative Control of Dual Arm Robots for Reasonable Motion Distribution," in *Proc. of International Conference on Intelligent Robots and Systems*, Pittsburgh, Pennsylvania, Aug. 1995.

[68] M. Bonert, L.H. Shu, and B. Benhabib, "Motion Planning for Multi-robot Assembly System," in *Proc. of ASME: Design Engineering Technical Conference*, Blacksburg, Jun. 1999.

[69] F. Caccavale, C. Natale, B. Siciliano and L. Villani, "Achieving a Cooperative Behavior in a dual Arm Robot System via a Modular Control Structure," in *Journal of Robotics System*, vol. 18, pp. 691-700, 2001.

[70] M. Nakamura, S. Goto and N. Kyura, *Mechatronic servo system control*, Springer-Verlang, Berlin, Heidelberg, 2004.

[71] Baldor electric company, *Motion Control Application notes*, Baldor motors and drives, pp. 23-25, 2002.

[72] R. Murphy, and P.M. Pardalos, *Cooperative Control and Optimization*, Kluwer Academic Publishers, 2002.

[73] Y.U. Cao, A. Fukuage, and A. Kahng, "Cooperative mobile robotics: Antecedent and directions," in *Autonomous robots*, vol. 4, pp. 7-27, 1997.

[74] R. Murphy, "conference on cooperative control and optimization open discussion," in *conference on cooperative control and optimization*, Florida, 2001.

[75] L.E. Parker, "Adaptive heterogeneous multi-robot teams," in *Neural networks and their applications*, vol. 28, pp. 75-92, 1999.

[76] J. Szewczyk, F. Plumet and P. Bidaud, "Planning and Controlling Robots through Distributed Impedance," in *Journal of Robotics System*, vol. 19, pp. 283-297, 2002.

[77] J.P. Desai and V. Kumar, "Motion Planning for Cooperating Mobile Manipulators, in *Journal of Robotics System*, vol. 16, pp. 557-579, 1999.

[78] J.F. Liu, K.A. Malek, "Robust Control of Planar Dual Arm Cooperative Manipulator," in *Robotics and Commuter Integrated Manufacturing*, vol. 16, pp. 109-120, 2000.

[79] L.T. Wang and M.J. Kuo, "Time Optimal Constant Speed Motion Programme for Multiple Cooperating Manipulators," in *Journal of Robotics System*, vol. 16, pp. 185-194, 1999.

[80] L.E. Parker, "Alliance:An Achitecture for Fault Tolerant Multirobot Cooperation," in *IEEE Transactions on Robotics and Atomation*, vol. 14, pp. 220-240, 1998.

[81] E. Tunstel, M.A.A. de Oliveira, and S. Berman, "Fuzzy Behavior Hierarchies for Multi-Robot Control," in *International Journal of Intelligent Systems*, vol. 17, pp. 449-470, 2002.

[82] J.T. Wen and K.K. Delgado, "Motion and Force Control of Multiple Robotic Manipulators," in *Automatica*, vol. 28, pp. 729-743, 1992.

[83] D. Sun and Y. Liu, "Position and Force Tracking of a Two-Manipulator System Manipulating a Flexible Beam," in *Journal of Robotic Systems*, vol. 18, pp. 197-212, 2001.

[84] M.A. Ridao, E.F. Camacho, J. Riquelme and M. Toro, "An Evolutionary and Local Search Algorithm for Motion Planning of Two Manipulators," in *Journal of Robotics System*, vol. 18, pp. 463-476, 2001.

[85] Y.P. Chien, Q. Xue, and Y. Chen, "Configuration Space Model of Tightly Coordinated Two Robot Manipulators Operating in 3D Workspace," in *IEEE Transactions on Systems, Man and Cybernetics*, vol. 25, pp. 695-704, 1995.

[86] N. Wiener, *Cybernetics or control and communication in the animal and machine*, The MIT press, MIT, 1965.

[87] N. Wiener, *Extrapolation, interpolation, and smoothing of stationary time series*, The MIT press, MIT, 1970.

[88] D. Pieper and B. Roth, "The Kinematics of Manipulators under Computer Control," in *Proc. of the Second International Congress on Theory of Machines and Mechanisms*, vol. 2, pp. 159-169, Zakopane (Poland), 1969.