

**Action Module Planning and Cartesian Based Control
of an Experimental Climbing Robot**

by

David M. Bevly

Bachelor of Science in Mechanical Engineering
Texas A&M University (1995)

Submitted to the
Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

Master of Science in Mechanical Engineering

at the

Massachusetts Institute of Technology

September 1997

© 1997 Massachusetts Institute of Technology

Signature of Author

Department of Mechanical Engineering
August 22, 1997

Certified By

Steven Dubowsky
Thesis Supervisor

Accepted By

Ain A. Sonin
Chairman, Departmental Graduate Committee

JAN 06 1998

LIBRARIES

Acknowledgments

I thank God for giving me both the opportunity and ability to do this work. "I can do all things through him who strengthens me" (Philippians 4:13).

I would like to thank my advisor Professor Steven Dubowsky for his guidance and technical assistance during the past two years. I would also like to thank Professor Dinos Mavroidis for his equally important encouragement and mentorship. I thank Dr. D for pushing and Dinos for pulling. I'd also like to thank the past and present members of the lab, including, Shane Farritor, Hervé Hacot, Karl Iagnemma and Guillaume Morel, for their technical contributions as well as their friendship over the past two years.

I also thank my family, including my two brothers, for all of their support. Special thanks to my dad, whom I strive to be like some day, for teaching me how to work hard on the farm and that life is not always fair. I also thank my mom, who is always right, for her encouragement as well as teaching me the power of prayer and positive thinking.

Special thanks to the Juicy Chicken basketball team members, Shane Farritor, Karl Iagnemma, Bill Kaliardos, Norm Peralta, Jerry Pratt, and Steven Villereal, for the opportunity to play on the 1996-97 B league Intramural Championship team (even if I disagreed with the team name).

I would like to thank my cousins, Lisa and John Gunstrom and their family for giving me a place to get away from the city during my first year in Boston. I also would like to thank Stephanie Shaw for her companionship over the past year, as well as allowing me to win a few arguments.

I would like to recognize the MIT Starr Foundation Fellowship and the Jet Propulsion Laboratory's Award # 960454 for providing financial support for this work.

Gig'em Aggies, _AT_M '94 - whoop!

Table of Contents

1 Introduction	
1.1 Motivation.....	10
1.2 LIBRA (Limbed Intelligent Basic Robot Ascender).....	11
1.3 Background and Literature Review.....	12
1.4 Purpose of this Thesis.....	16
1.5 Outline of Thesis.....	17
2 Action Module Planning of LIBRA	
2.1 Introduction.....	18
2.2 Action Module Planning.....	18
2.2.1 Action Modules.....	19
2.2.2 Level of Action Modules and Sub-Goals.....	22
2.2.3 Evolution of Plans Through a GA Search Technique.....	23
2.3 Physics-Based Simulation.....	26
2.3.1 Kinematic Model of Libra.....	28
2.3.2 Force Analysis.....	30
2.3.3 Additional Analytical Checks.....	35
2.4 The Four Tasks.....	35
2.5 Action Module Planning Results.....	36
2.5.1 Ladder Task.....	36
2.5.2 Modified Ladder Task.....	37
2.5.3 "H" Task.....	40
2.6 Summary and Conclusions.....	42
3 A Simplified Cartesian Computed Torque Controller	
3.1 Introduction.....	43
3.2 Current Cartesian Control Schemes.....	43

3.2.1	Jacobian Transpose Control	44
3.2.2	Operational Space Control.....	46
3.2.3	Resolved Rate Control.....	48
3.2.4	Admittance Control.....	49
3.3	Simplified Cartesian Control Scheme.....	49
3.3.1	Simplification of the Cartesian Computed Torque Control Scheme.....	51
3.3.2	A Different Perspective on the Controller (Implicit Admittance Control).....	52
3.3.3	Additional Issues.....	54
3.4	The SCCT Controller under Position Control.....	55
3.4.1	Implementation.....	55
3.4.2	Model Verification.....	56
3.4.3	Comparison of Jacobian Transpose and SCCT Control.....	57
3.5	Force Control with the SCCT Controller.....	61
3.6	Summary and Conclusions.....	62
4	Experimental Climbing of LIBRA	
4.1	Introduction.....	63
4.2	Basis for Cartesian Control of LIBRA.....	63
4.3	Experimental Setup.....	66
4.3.1	Overall Setup.....	66
4.3.2	LIBRA.....	66
4.3.3	LIBRA's Peg Board.....	67
4.3.4	Power Amplifiers.....	69
4.4	Cartesian Control of LIBRA Applied to the Climbing Task.....	69
4.4.1	The Control Scheme.....	69
4.4.2	Limitation of Endpoint Accelerations.....	71
4.4.3	Calculation of Body and Leg Positions.....	73
4.4.4	Gain Selection.....	76
4.4.5	Gravity Compensation.....	77

4.4.6	Executing Plans.....	78
4.5	Experimental Results.....	80
4.6	Summary and Conclusions.....	84
5	Conclusions and Suggestions for Further Work	
5.1	Contributions of This Work.....	85
5.2	Suggestions for Further Work.....	86
References		
Appendices		
A	LIBRA Kinematics.....	96
B	Inverse Kinematics of LIBRA's Limb.....	100
C	Peg Positions (in inches) for the Four Tasks.....	103
D	Plans for Execution of the Four Tasks.....	104

List of Figures and Tables

List of Figures

Figure 1.1	LIBRA.....	11
Figure 2.1	Action Modules.....	19
Figure 2.2	Example of an Action Plan.....	21
Figure 2.3	Execution of an Action Plan.....	22
Figure 2.4	Low and High Level Action Modules.....	23
Figure 2.5	Chromosome Representation.....	24
Figure 2.6	Genetic Crossover.....	26
Figure 2.7	MSV Graphical Animation of LIBRA.....	27
Figure 2.8	Coordinate Frames for the Leg Ports of LIBRA.....	29
Figure 2.9	Force Analysis of LIBRA.....	30
Figure 2.10	Simplified Force Analysis of LIBRA.....	31
Figure 2.11	Force Analysis of the “Push-pull” Configuration of LIBRA.....	32
Figure 2.12	LIBRA With Only One Arm Holding.....	33
Figure 2.13	The Four Tasks.....	36
Figure 2.14	Nominal Gait for the Ladder Climb Task.....	37
Figure 2.15	Modified Ladder Task.....	38
Figure 2.16	Fitness Score for the Evolution of the Modified Ladder Plan.....	39
Figure 2.17	“H” Task.....	40
Figure 2.18	Maximum Actuator Torques for the Two Different H Task Plans.....	41
Figure 2.19	Fitness Score for the Evolution of the “H” Task Plan..	41
Figure 3.1	Jacobian Transpose Control.....	44
Figure 3.2	J ^T F Applied to a Manipulator.....	45
Figure 3.3	Operational Space Control.....	47
Figure 3.4	Resolved Rate Control.....	48
Figure 3.5	Admittance Control.....	49

Figure 3.6	Cartesian Control of a Manipulator.....	50
Figure 3.7	Simplified Cartesian Computed Torque (SCCT) Control.....	52
Figure 3.8	Implicit Admittance Control.....	53
Figure 3.9	Comparison of Experimental System and Simplified Model.....	57
Figure 3.10	Comparison of $J^T F$ and $K^* J^{-1} \ddot{X}_d$ Transformations.....	58
Figure 3.11	Simulated Step Response of the two Cartesian Controllers.....	59
Figure 3.12	Experimental Step Response of the two Cartesian Controllers.....	59
Figure 3.13	Experimental Path Tracking of the two Cartesian Controllers.....	60
Figure 3.14	Stiffness Selection in Cartesian Space.....	61
Figure 3.15	Force and Motion Sub-spaces of an Environment.....	62
Figure 4.1	Walking Robot on Uneven Terrain.....	64
Figure 4.2	Control of LIBRA.....	65
Figure 4.3	Grabbing a Peg.....	65
Figure 4.4	LIBRA's Hooks.....	66
Figure 4.5	Experimental Setup.....	67
Figure 4.6	LIBRA's Peg Board.....	68
Figure 4.7	Block Diagram for the Control of LIBRA.....	70
Figure 4.8	Body Accelerations and Forces on LIBRA.....	71
Figure 4.9	Hook Coordinate Frame (g-h).....	72
Figure 4.10	Coordinate Frames for LIBRA.....	74
Figure 4.11	Calculation of Body Position.....	75
Figure 4.12	Pull of Legs for Control of LIBRA.....	76
Figure 4.13	Previous Control Approach of LIBRA.....	78
Figure 4.14	Sequence of Steps to Grab a New Peg.....	79
Figure 4.15	Climbing Gait for the LIBRA.....	80
Figure 4.16	Two Steps of LIBRA's Climb.....	81
Figure 4.17	Body Path Tracking.....	82

Figure 4.18	Body Push-up.....	83
Figure 4.19	Body Compliance.....	84
Figure A.1	Kinematic Parameters of LIBRA.....	96
Figure A.2	Kinematic Parameters of LIBRA's Hooks.....	97
Figure A.3	Kinematic Parameters of LIBRA's Body.....	98
Figure B.1	LIBRA's Limb in its Port Frame.....	100

List of Tables

Table 2.1	Action Module Inventory.....	20
Table 2.2	Joint Limits for LIBRA.....	30
Table 2.3	Action Plan for the Modified Ladder Task.....	38
Table 2.4	Action Plan for the "H" Task with <1500 oz-in Torque.....	42
Table 3.1	Model Parameters of the LIBRA Joints.....	56
Table 4.1	Controller Gains for Peg Climbing.....	76
Table A.1	Leg Kinematic Parameters.....	97
Table A.2	LIBRA Denavit-Hartenberg Port Transformation Parameters.....	99
Table B.1	Kinematic Angle Solutions.....	102
Table C.1	Peg Positions for the Four Tasks.....	103
Table D.1	Plan for the Ladder Task.....	104
Table D.2	Plan for the "H" Task.....	104
Table D.3	Plan for the Across Task.....	105
Table D.4	Plan for the Circle Task.....	106

Chapter 1

Introduction

1.1 Motivation

There is an increasing need for the use of mobile multi-limbed robots in such applications as space exploration, nuclear site clean-up, bomb disposal, and infrastructure inspection and maintenance. These applications require highly capable multi-limbed robots in which their actions can be planned and controlled in such unstructured environments. However, the use of multi-limbed robotic systems has been somewhat limited due to their inability to fully operate in these environments. Effective planning and control techniques must be developed in order to fully utilize the capabilities of these systems.

Several wall climbing robots have been developed for the study of maneuvering in difficult environments. One wall climbing robot utilizes pneumatic suction cups on the feet (Nagakubo, 1994) and two other wall climbing robots use the pneumatic suction cups on both the feet and body (Luk, 1991; Gradetsky, 1990) to actively attach to the wall. The ability to attach to any point of the wall with the feet and body somewhat reduces the importance of foot and body placements. Another climbing robot uses friction to allow the robot to travel through ducts (Neubauer, 1993). The author points out the need to plan specific robot actions in order to allow the robot to maneuver around difficult sections of the duct.

In this thesis an experimental climbing robot with non-actuated end effectors which is used to climb on a series of pegs is studied. This problem offers unique challenges of planning, such as which peg to grab and where to

position the body, such that no physical constraints are violated. Secondly, the problem produces a unique multi-limbed control problem, since the robot can not actively attach to its environment. The robot must be able to execute tasks by interacting with the environment of pegs with its non-actuated end effectors. Additionally, uncertainties in the system increases the need for the development of an effective control strategy which can be applied in an unstructured environment.

1.2 LIBRA (Limbed Intelligent Basic Robot Ascender)

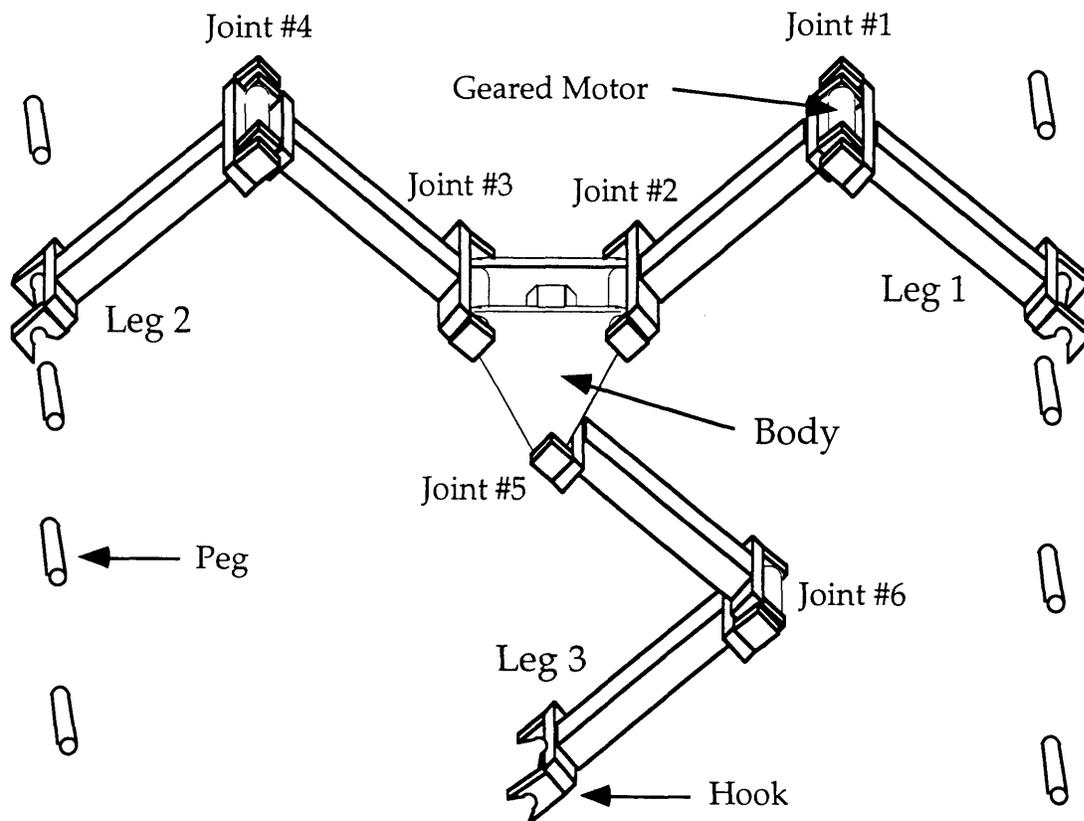


Figure 1.1. LIBRA

As seen in Figure 1.1, LIBRA is a planar, three legged climbing robot, which was built by Dalila Argaez (Argaez, 1993) and first controlled by Craig

Sunada (Sunada, 1994). Additionally, many figures of LIBRA used in this thesis are modified versions of figures which appear in (Sunada, 1992). Each 32 cm limb of the 40 Newton robot, consists of two joints driven by highly geared motors. Each gearhead has 2° of backlash at each joint, which can result in as much as a half inch of error at the endpoint of each limb. Specific details on the design and properties of the climbing robot LIBRA can be found in (Sunada, 1994; Arguez, 1993). In this work, each leg on LIBRA was fitted with hooks to allow the system to climb on pegs mounted to a wall. The hooks have an opening at one end which allows each limb to passively hang onto a peg, as opposed to having the ability to actively grab the peg. A peg board was built such that the pegs, used for LIBRA to climb, could be placed in several patterns. This was done in order to study the planning of LIBRA on different tasks. Peg locations are known to within ± 0.25 inches. More information on the overall setup is presented in Chapter 2 and Chapter 4.

1.3 Background and Literature Review

Much research has been done in the area of mobile robotic systems, especially walking machines (Song, 1988). Many different methods have been considered for the problem of path planning of robotic systems and mobile robots (Latombe, 1991). Much work has been done on the study of legged locomotion gaits (Venkataraman, 1996; Song, 1987) and applied for planning and control of such popular mobile robots as Dante (Wettergreen, 1996) and Genghis and Atilla (Brooks, 1989). Additionally, the control of different gaits was studied for walking in (Pack, 1996) and for wall climbing in (Nagakuba, 1994). Work was done on planning based on physical constraints, such as power consumption, by producing a power map for the robot (Dubowsky, 1995). Planning based on another physical constraint, actuator saturation, through the force workspace was used in (Madhani, 1997). Another approach to the planning and control of mobile robots is known as behavior control (Brooks, 1986), and its application to planetary exploration was investigated in

(Gat, 1994). Work has also been done on the use of potential fields for motion planning and obstacle avoidance of mobile robots (Khatib, 1994; Hwang, 1992).

A recent method which combines the use of physical constraints and plan optimization was developed in (Cole, 1995) and has been further investigated in (Farritor, 1997). The method assumes that the robot has been designed to meet the physical requirements of the task (Farritor, 1996) and that prior knowledge of the environment is available, perhaps from on board sensors (Krotkov, 1994). The action planning methodology attempts to aggressively utilize the full physical capabilities of the robot and incorporate obstacle avoidance (Borenstein, 1991) or obstacle accommodation (Shan, 1995) as required by the robot. The method is based on physically realizable actions by the robot, called action modules, which are placed in the correct sequence for successful execution of a given task. A hierarchical selection process, which includes task and configuration filters, reduces the action module inventory to a reasonable search size. Finally, with the problem reduced to finding a plan with the correct sequence of necessary actions, a Genetic Algorithm (GA) search technique is used to find a successful plan.

The first purpose of this thesis was to study the use of the action module planning methodology by applying it to the experimental climbing robot named LIBRA. The ability of the planning methodology to produce successful plans to execute a task, given different peg locations or patterns for LIBRA to climb, was studied. A similar problem which used an ordinal optimization technique to search for a sequence of foot placements to allow the robot to walk through its environment was studied in (Chen, 1996). The second purpose was to develop and apply an effective control strategy to allow the climbing robot to successfully execute its action plans when small uncertainties were present in the environment. This is a deceptively challenging application due to the large amount of joint backlash, slip between the hook and peg, and uncertainties of the exact peg location.

Much work has been done on the control of multi-limbed robotic systems under similar unstructured environments. A robot must have an

adequate controller to carry out the plans that are developed, regardless of the planning strategy. Most mobile robots use conventional joint PD control because of its ease of implementation as well as suitable performance. However, joint PD methods can suffer in unstructured environments, since forces with the environment can not be easily controlled. Additionally, they have the disadvantage of being a joint level controller such that a cartesian stiffness can not be specified. The importance of setting the stiffness in cartesian space of a manipulated object is presented in (Schneider, 1992). The use of a passive compliance, created by pneumatic actuators, is used for effective control of a wall climbing robot in (Luk, 1991).

Recent controllers that try to control the forces on the body by virtual model control (Pratt, 1997) or the effective stiffness between the body and ground, as well as the effective stiffness between the limbs and the environment, known as Coordinated Jacobian Transpose Control (CJTC) (Sunada, 1992) showed promise because of their ease of implementation. The use of the cartesian based controller in order to control positions and forces of a multi-limbed robot was demonstrated with the CJTC algorithm. These methods try to set a desired cartesian impedance (Hogan, 1985) or cartesian stiffness (Salisbury, 1980) in order to control general forces and positions (Mills, 1996), usually without force feedback, by utilizing a form of Jacobian transpose control (Asada, 1986). However, simple Jacobian transpose control has been shown to have marginal performance in (Plumet, 1995).

Many cartesian based controllers have utilized force feedback to control the forces applied to unknown terrain (Gardner, 1991). One method, called sky-hook suspension, utilizes the same idea of setting cartesian stiffness, but with the addition of force feedback for walking on rough terrain (Yoneda, 1994). More complicated schemes such as a hybrid position/force control scheme introduced in (Raibert, 1990) was applied to a walking robot in (Fujimoto, 1996). Another control scheme for walking on difficult terrain, which according to the authors should be the only type of terrain considered for a walking robot, uses levels of control, including a force compliance level

with force feedback (Celaya, 1996). Several other control schemes have also utilized force feedback for walking in uncertain or partially known environments (Orin, 1981; Gorinevsky, 1990; Klien, 1983; Lehtinen, 1996). However, force feedback may not always be available on a multi-limbed robot such as LIBRA.

Computed torque schemes, such as operational space control (Khatib, 1987), have been used to overcome the limitations of Jacobian transpose control (Plumet, 1995). Computed torque schemes have also been applied to cooperative manipulation in order to specify cartesian impedance for improved performance in (Schneider, 1992) and occasionally applied to walking robots (Shih, 1993; Chevallereau, 1997). Optimal state feedback (Channon, 1996) and μ -synthesis control (Pannu, 1996) utilize linear state space models for the control of a walking robot. Additionally, adaptive control/walking techniques have been used to try to adapt the robot to its terrain or environment (McGhee, 1990; Ilg, 1995; Kun, 1997). However, these methods can be difficult to implement on multi-limbed mobile robots, because full dynamic models of the robot and environment are needed. Additionally, mobile platforms with limited computational capabilities may not be able to implement these computationally expensive controllers.

A simplified cartesian computed torque (SCCT) control scheme which overcomes the need for force sensing and complex models of the robot and environment has been developed and demonstrated on the experimental climbing robot. This method can be applied to mobile robots which are designed to be light weight by having large gear ratios on electric motor driven actuator joints. This limits most of the dynamics to the geared actuator joint, which is the reason joint PD controllers have shown success for controlling mobile robots. Simplifications can be made based on the assumption that the dynamics are dominated at the joint for a highly geared walking machine. This allows simplification of a general control scheme which provides increased performance over Jacobian transpose control, while

allowing the flexibility to choose cartesian stiffness in order to operate in partially known environments by controlling the force interactions with the environment.

1.4 Purpose of this Thesis

Previous work in the planning of LIBRA was limited to force workspace and power map applications (Madhani, 1997; Dubowsky, 1995). The Action Module Planning methodology takes these constraints along with other physical constraints of the system to plan its entire actions. This method has been applied to mobile walking robots and to a rover similar to the Jet Propulsion Laboratory's Lightweight Survivable Rover (LSR), in simulation (Farritor, 1997). The purpose of this thesis was to apply the methodology to an experimental system such that results could be compared with the simulation, as well as reveal any limitation of the planning technique to the experimental system.

Additionally, previous work had been done on the control of LIBRA in simulation (Argaez, 1993), and experimentally (Sunada, 1994) through the use of Jacobian transpose control. In this thesis, the SCCT control algorithm, which can offer improved performance in some circumstances, for the control of LIBRA is explored. The SCCT control algorithm utilizes assumptions for the highly geared climbing robot, which are validated through simulation, in order to simplify a cartesian computed torque control scheme. The two cartesian control methods are then examined in theory, simulation, and experimentation for the control of one limb of LIBRA.

The SCCT control scheme is then applied to the climbing robot LIBRA on a climbing task. Again, the experimental results show improved performance over the conventional Jacobian transpose control scheme. Finally, it is shown that climbing is attainable with an action module plan and the SCCT controller on the peg board environment.

1.5 Outline of Thesis

This thesis is divided into five chapters. This chapter serves as an introduction and overview of the work. Chapter 2 presents the action module planning methodology used in conjunction with a genetic algorithm search method. Details of the planning methodology, the genetic algorithm, and the simulation used to evaluate each plan are presented. Finally, results are presented for several LIBRA tasks.

Chapter 3 provides a brief review of several cartesian controllers and gives an introduction to the SCCT control algorithm. Validation of the simplified assumptions of the climbing robot are presented. Simulation and experimental results are then presented to compare traditional Jacobian transpose control and the SCCT controller applied to one limb of LIBRA.

Chapter 4 introduces the application of the action module planner and the SCCT control scheme to the experimental system. Details of the experimental climbing of LIBRA are presented. Experimental results are provided, comparing the traditional Jacobian transpose controller and the SCCT controller, applied to the multi-limbed robot. Additionally, the experimental results show that the method is capable of climbing on the peg board environment.

Chapter 5 outlines some conclusions regarding the action module planning methodology, the SCCT controller, and there application to LIBRA. Suggestions for further work are also presented.

The appendices A through D to this thesis give details of the practical implementation of the proposed methods. Appendix A gives kinematic parameters and forward kinematic solutions of the actual LIBRA climbing robot, and Appendix B provides the solution of the inverse kinematics of one leg of LIBRA. Appendix C provides the data on the peg positions for the four tasks used in the simulations and experiments, while Appendix D provides hand derived plans required to execute each of the four tasks studied.

Chapter 2

Action Module Planning of LIBRA

2.1 Introduction

This chapter describes the use of the action module planning methodology in conjunction with a genetic algorithm (GA) search method for the planning problem of LIBRA on the peg board. Section 2.2 presents details of the action planning methodology including the GA search technique. Section 2.3 contains details of the physics-based simulation used for the evaluation of each plan. Section 2.4 discusses the four tasks which were developed for the study. Section 2.5 presents some results of the methodology applied to LIBRA.

2.2 Action Module Planning

A task, a robot (LIBRA in the case of this thesis) and an inventory of action modules are inputs to the action module planning methodology used to produce a plan that allows successful completion of the task (Farritor, 1997). Each plan is composed of physically realizable actions by the robot, called action modules, placed in sequence to allow successful completion of a task. A hierarchical planning procedure which includes task and configuration filters reduces the action module inventory to a reasonable search size. A genetic algorithm optimization search technique is then used to find a sequence of action modules which will create a successful plan. A successful plan is one which allows completion of the task without violating any physical constraints. A simulation is built in order to check against the violation of any of these constraints and additionally rank plans for evolution of a successful plan using the GA.

2.2.1 Action Modules

The foundation of the planning technique are action modules which consist of a list of physically realizable actions of a robot, such as those shown in Figure 2.1. A list of all possible action modules for a robot are included into an action module inventory. Table 2.1 contains a list of the action module inventory that is used for LIBRA.

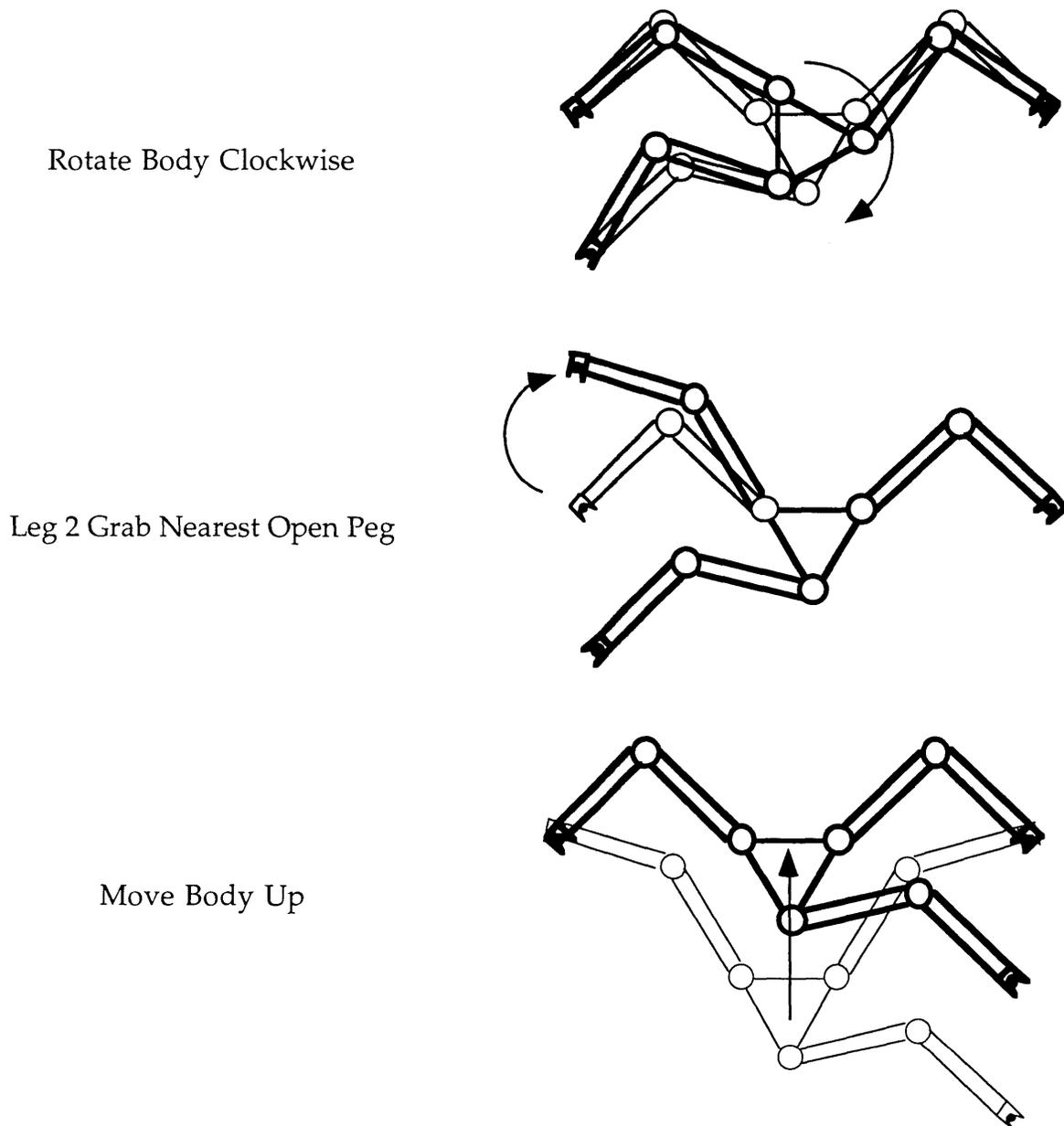


Figure 2.1 Action Modules

Table 2.1 Action Module Inventory

Action #	Description	Action #	Description
001	Move Body +x	N00	Leg N Release Peg
002	Move Body -x	N91	Leg N Grab Nearest Open Peg
003	Move Body +y	N92	Leg N Grab 2nd Nearest Open Peg
004	Move Body -y	N93	Leg N Grab 3rd Nearest Open Peg
005	Move Body +theta	N99	Leg N Grab A Random Open Peg
006	Move Body -theta	NXX	Leg N Grab Peg XX

Because LIBRA has three legs, there is a total of $21+3p$ action modules, where p is the number of pegs in the task. Action modules N99 and NXX (where N represents the leg number) are most often omitted in order to reduce the inventory to 18 action modules. For example, a climbing problem with 30 pegs, will create an action module inventory of 111 actions. A reduced inventory is used in order to reduce the size of the search space which is described by $D=N^m$, where D is the number of possible action plans, N is the number of action modules in the inventory, and m is the number of action modules used in the action plan (Farritor, 1997). Therefore action modules such as “grab nearest peg” are used over “grab peg #” in order to decrease the action module inventory. Since it is only physically necessary to consider pegs with in the vicinity of LIBRA’s workspace, these action modules greatly reduced the action module inventory without sacrificing the ability to derive a successful plan. If a task had more than 3 pegs in the workspace then additional action modules, such as N94 and N95 (leg N grab the fourth and fifth nearest open peg), would need to be added to the action module inventory to insure that all possible pegs within the workspace of LIBRA are accessible to the plan. This demonstrates the need for a well designed inventory in order to utilize the capabilities of the robot to accomplish its task.

Additionally, for more complex systems with more physical capabilities, including data collection and communication abilities, the action module inventory is much greater. For these system, additional task and

configuration filters are first used, in a hierarchical fashion, to reduce the total action module inventory to only those action modules necessary for the task at hand (Farritor, 1997). In the case of the LIBRA climbing problem, the action module inventory is already reduced to its “minimum” such that there was no need to add any additional pre-filters to reduce the inventory.

Action modules are then placed into a sequence of numbered steps to create an action plan, as in Figure 2.2, for successful execution of a task, as shown Figure 2.3.

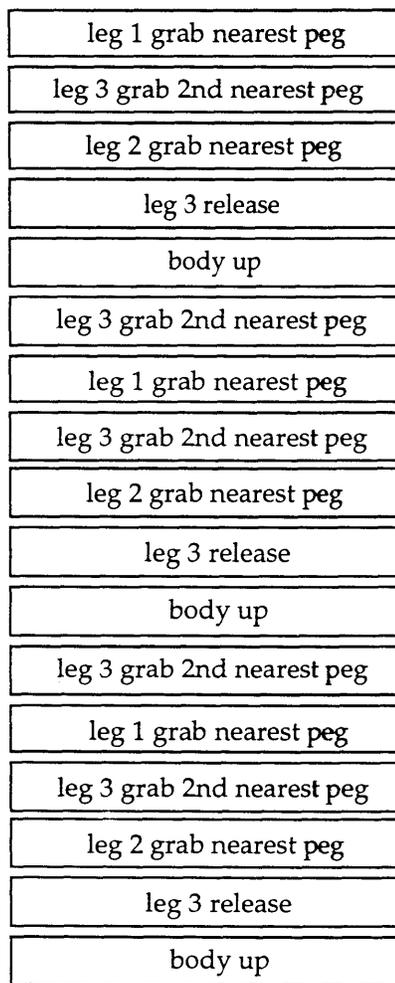


Figure 2.2 Example of an Action Plan

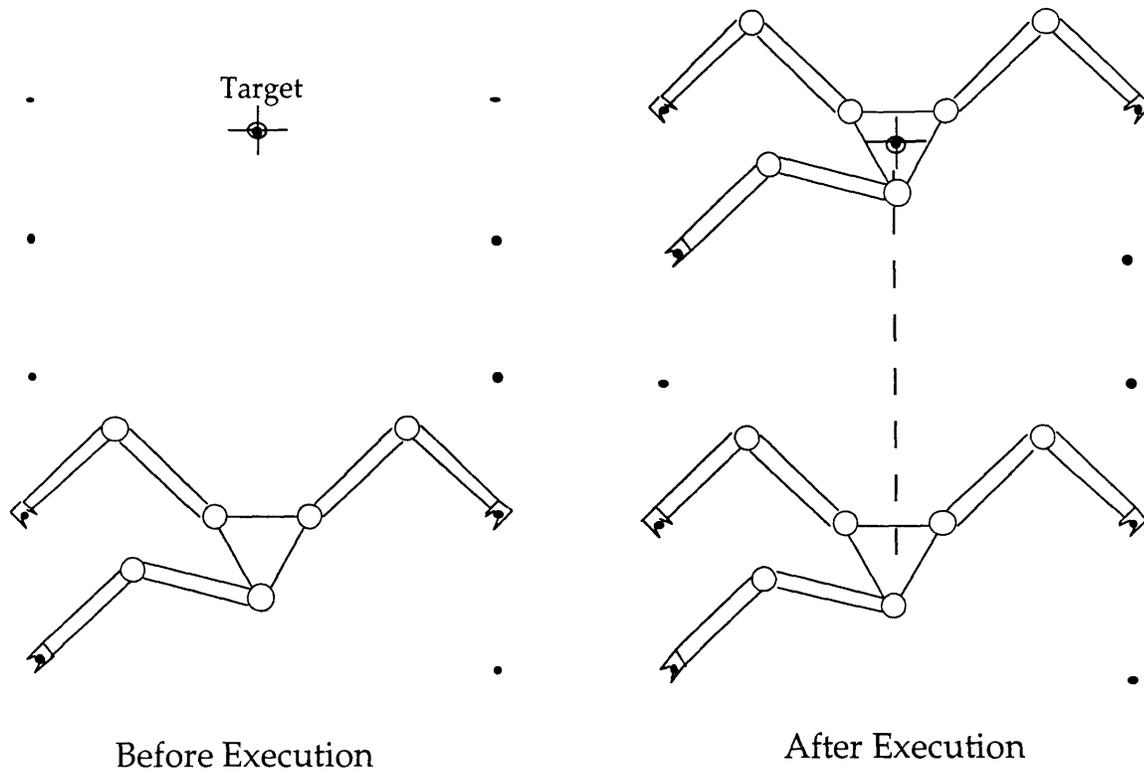


Figure 2.3. Execution of an Action Plan

2.2.2 Level of action Modules and Sub-Goals

Work has been done on the study of lower level vs. higher level action modules (Farritor, 1997). Repeated sequences such as seen previously in Figure 2.2 could be clearly grouped into one action module of climb up one step as in Figure 2.4, in order to decrease the search space.

An action module planning technique can then be applied to find a higher level action module, such as the best gait to climb one step. Once this gait is developed it can be added to the action module inventory and used as its own action module. Work has also been done on the planning method's ability to "learn" to do repeated sequences such as climbing a series of steps (Farritor, 1997).

In some of the tasks, shown later, sub-goals are necessary for successful completion of the task. The need for these sub-goals is due to the limited backtracking and short look-ahead horizon discussed in (Farritor, 1997).

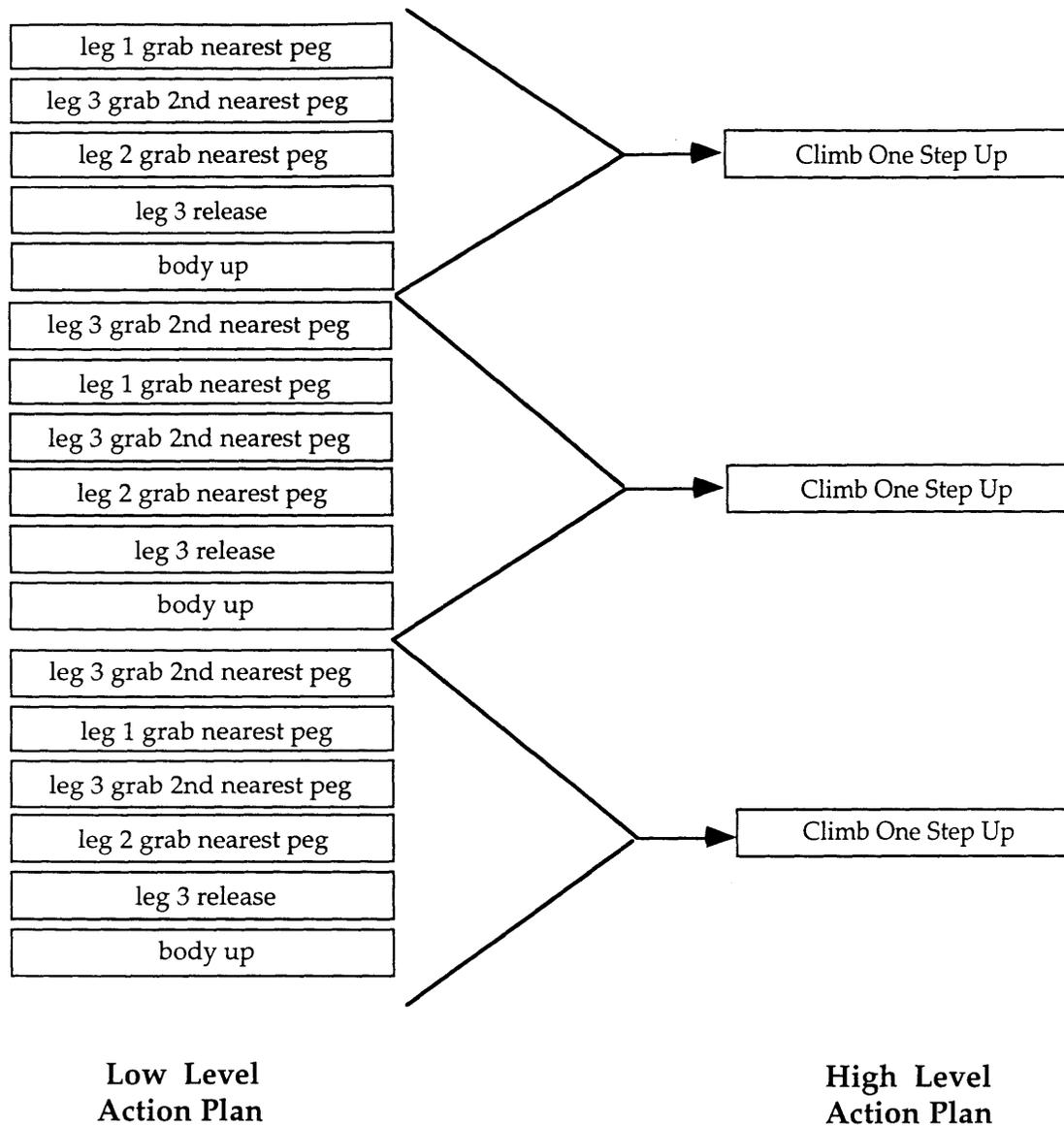


Figure 2.4. Low and High Level Action Modules

2.2.3 Evolution of plans through a GA search technique

A genetic algorithm search technique, which is based on natural biological evolutionary processes, is used to evolve a successful plan. A chromosome represents an action plan composed of a list of action modules in the order which the actions are to be executed by the robot. Through genetic algorithm operators, such as crossover and mutation, a successful plan is evolved through generations of evolution. A complete description of genetic algorithms can be found in (Goldberg, 1989). Each action module is

placed into a chromosome tree as seen in Figure 2.5, which is a direct mapping of each action module, to represent the action plan.

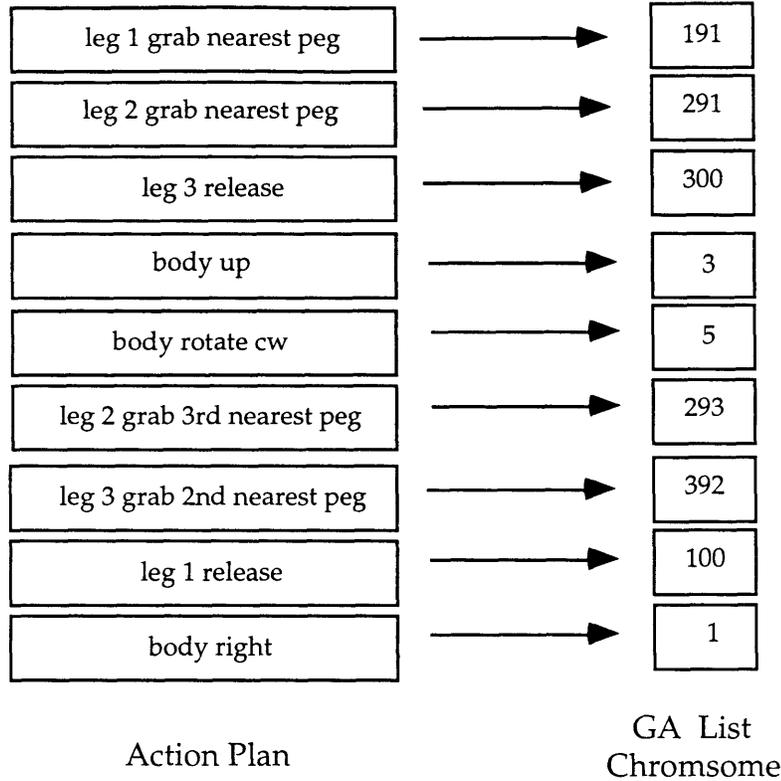


Figure 2.5. Chromosome Representation

Once an original generation of plans has been created, randomly, these plans must be evolved into a successful plan. Evolution of these plans occur through genetic algorithm operators such as mutation and crossover. Each plan is ranked among its generation by a fitness score. A physics based simulation is used to score each plan based on a fitness function which varies depending upon the task and its environment. The general fitness function used for LIBRA is:

$$f = w_1 |D_{\text{body}}| + w_2 |D_{\text{leg}_i}| - w_3(P) + w_4(\delta) - w_5(LP) - w_6(S) \quad (2.1)$$

where:

D_{body} = the distance the body of LIBRA travels toward the target

D_{leg_i} = the distance leg i travels toward the target

P = power consumed

$\delta = 1$ if the target (or sub-goal) is reached and 0 otherwise

LP = length of plan
S = Stability factor
 w_i = weighting factor

A well designed fitness function is a crucial element of the action planning methodology in order to give credit to plans which do things which are useful for final execution of the task. This created the need to award points for plans which moved the legs of LIBRA toward the target, as it was necessary for LIBRA to grab pegs on its way toward the final target.

Two plans are then selected and “mated” to produce a new generation offspring plan. Plans for crossover are randomly selected with a weighting factor based on the fitness score of each plan. This is to ensure that the best plans mate more often and create the stronger generation of plans. The mating occurs through crossover in which part of one plan is crossed with another plan.

Tail crossover, referring to the latter half of the plan, as seen in Figure 2.6, was found to be the most effective method of crossover for the search of a successful plan. A tail crossover maintains the successful part of each plan, by switching the two lower halves of each plan. This enables the evolution of a successful plan by adding to previously successful sections of plans. Building onto successful parts of plans allows the planning methodology to operate with a short look-ahead horizon. However, this also causes the planning technique to have limited backtracking which creates the need for sub goals or high level action modules to prevent the planner from being caught in a local minimum.

Crossover points C1 and C2 (shown in Figure 2.6) are chosen randomly with a distributed weighting factor about the point of failure. Choosing the crossover point in the neighborhood of failure allows the GA to operate near the critical region of each plan in order to facilitate evolution. A replacement factor of 0.8 was specified to ensure that the best 20% of each generation was carried over to the next generation.

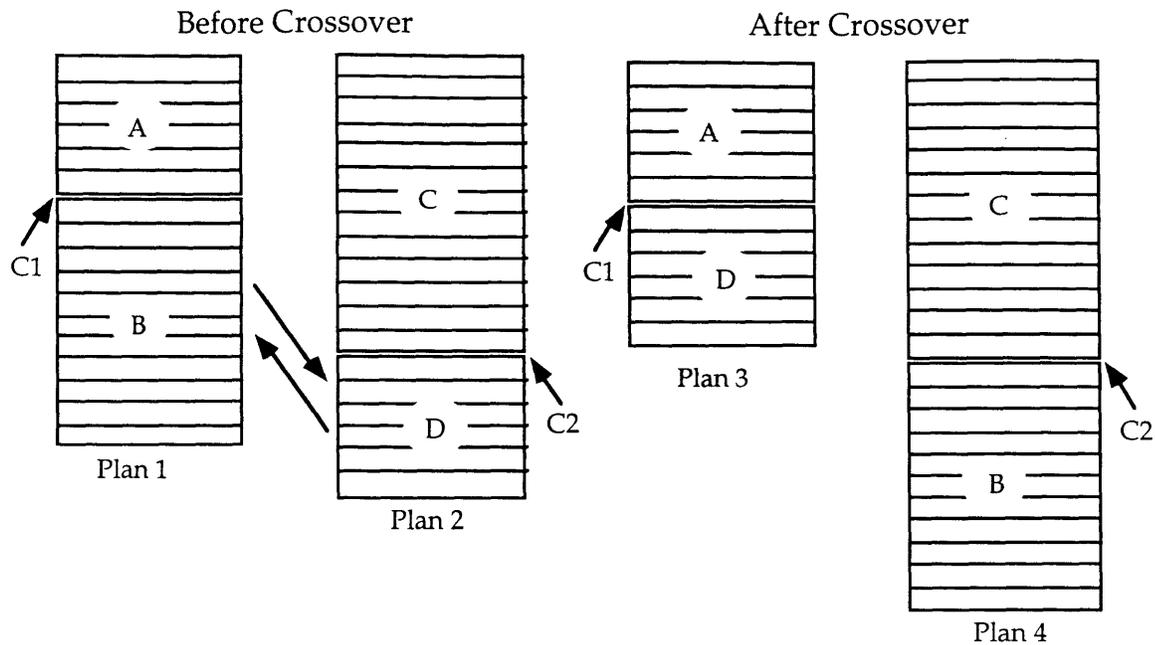


Figure 2.6. Genetic Crossover (Farritor, 1997)

Mutation is a genetic algorithm operator to help with evolution of plans by insuring population diversity. The mutation rate was nominally set at about 5% such that one in every twenty plans would have a random action module inserted into a plan in the neighborhood of failure (as with the crossover techniques).

2.3 Physics Based Simulation

A physics based simulation tool for the evaluation of each action plan is required to check for the violation of any physical constraints of the robot during its execution of a plan. This simulation is also used to assign a fitness score to each plan for the GA search method as mentioned previously. The output of the simulation is made compatible with the MSV (Torres, 1989) graphical interface for graphical animation as seen in Figure 2.7. The simulation checks each plan for physical constraints such as joint and kinematic reach limits, end effector forces, power consumption, limb interference, stability, and simple interference violations. The simulation

insures that each plan evolved will accomplish the task within the physical constraints imposed on the system.

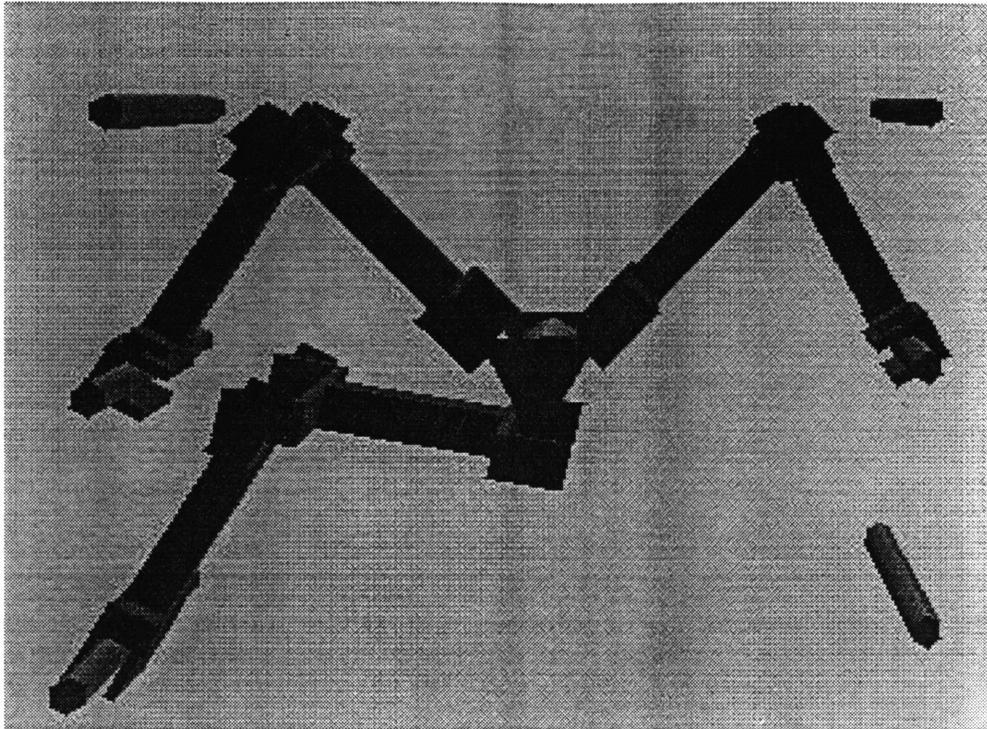


Figure 2.7. MSV Graphical Animation of LIBRA

The simulation selects a peg for leg action modules N91-NXX. Once a peg number has been selected, the simulation checks to see if the peg is within the reach limit of the leg. The reach limit is determined by the length of the leg minus a reach offset, which is included to maintain a factor of safety from singularity. If the peg is out of the reach limit of the limb, the simulation chooses a peg which is within the workspace of the limb. If an action module such as N93 (“grab the 3rd nearest peg”) is used where there are only two pegs in the workspace, the second nearest peg is selected. If no open pegs are in the workspace, the simulation returns a reach limit error.

Once a peg within the workspace of the limb is chosen, the endpoint of the limb is placed at the x-y position of the peg in the fixed frame. It is assumed in simulation that the leg could hold onto the peg in any orientation. Body move action modules (001-006) update the position of the

body in the fixed frame while leaving limb positions of any legs which are holding pegs at the peg position. The endpoint position of any limb which has released a peg (action module N00) is updated to move with the body. Once position of the body and all three legs in the fixed frame are known, a kinematic and force analysis is used to check for the violation of any physical constraints of a plan.

2.3.1 Kinematic Model of Libra

Figure 2.8 shows the coordinate frames used in the analysis of LIBRA. A reference coordinate frame for each leg is placed at each corner of LIBRA's body, called the body port, in order to create a Denavit-Hartenberg kinematic notation for LIBRA (Craig, 1986). Using the kinematic model of LIBRA, the position of each leg in its reference port frame (x_p, y_p) , shown in Figure 2.8, is found through the frame transformations, as discussed in (Asada, 1986):

$$X_p = T_{port}^{-1} X_B \quad (2.2)$$

$$X_B = T_{body}^{-1} X \quad (2.3)$$

or:

$$X_{port} = T_{port}^{-1} T_{body}^{-1} X \quad (2.4)$$

where:

$$\begin{aligned} X_B &= \text{position of the leg in the body frame } (x_B, y_B) \\ X &= \text{position of the leg in the fixed frame } (x, y) \end{aligned}$$

The transformation can be implemented by:

$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = T_{port}^{-1} T_{body}^{-1} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (2.5)$$

where:

$$T_{port}^{-1} = \begin{bmatrix} \cos(\theta_{port}) & \sin(\theta_{port}) & -y_{port} \sin(\theta_{port}) - x_{port} \cos(\theta_{port}) \\ -\sin(\theta_{port}) & \cos(\theta_{port}) & -y_{port} \cos(\theta_{port}) + x_{port} \sin(\theta_{port}) \\ 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

$$T_{body}^{-1} = \begin{bmatrix} \cos(\theta_{body}) & \sin(\theta_{body}) & -y_{body} \sin(\theta_{body}) - x_{body} \cos(\theta_{body}) \\ -\sin(\theta_{body}) & \cos(\theta_{body}) & -y_{body} \cos(\theta_{body}) + x_{body} \sin(\theta_{body}) \\ 0 & 0 & 1 \end{bmatrix} \quad (2.7)$$

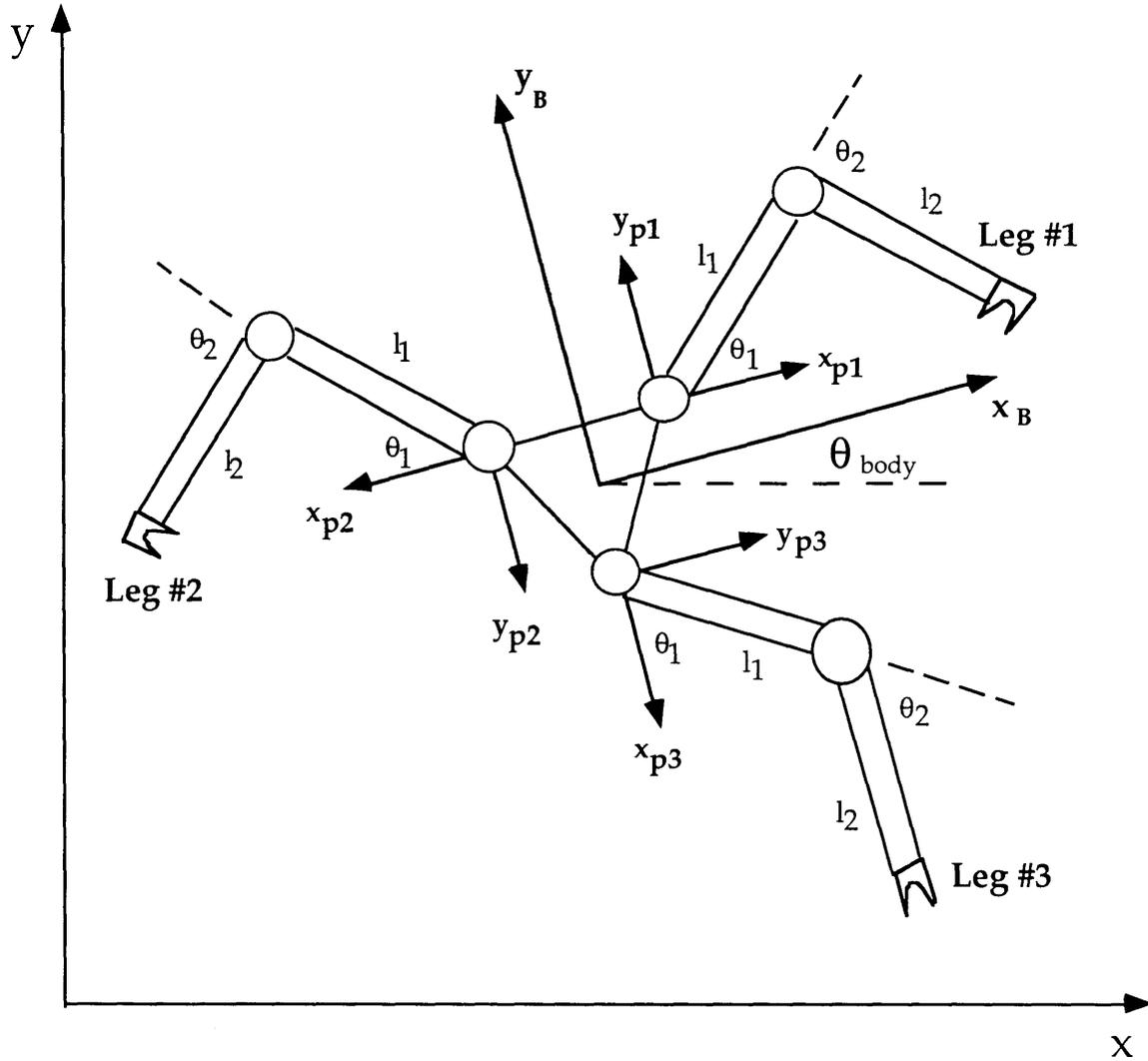


Figure 2.8. Coordinate Frames for the Leg Ports of LIBRA.

The parameters x_{body} , y_{body} , and θ_{body} are the position and orientation of the body of LIBRA. The Denavit-Hartenberg transformation parameters x_{port} , y_{port} , and θ_{port} required to implement the above port transformation for each limb can be found in Appendix A.

Once the position of the leg is determined in its port frame, inverse kinematics shown in Appendix B are used to determine the six joint angles of LIBRA. Joint limits imposed on the system are shown in Table 2.2.

Table 2.2. Joint Limits for LIBRA

	Leg #1	Leg #2	Leg #3
θ_1	$+140^\circ, -80^\circ$	$+80^\circ, -140^\circ$	$\pm 110^\circ$
θ_2	$\pm 110^\circ$	$\pm 110^\circ$	$\pm 110^\circ$

2.3.2 Force Analysis

After solving for the inverse kinematics of each leg, the forces at each leg are determined through a quasi-static analysis. It is assumed that no more than two legs are used to support LIBRA. The third leg, called the “limb in use,” is free to move between pegs and is therefore omitted from the force analysis. For the case of two legs supporting LIBRA, the system is statically indeterminate due to the fact that there are 3 equations and 4 unknowns, shown in Figure 2.9, and seen in Equation 2.8, which creates an infinite number of solutions.

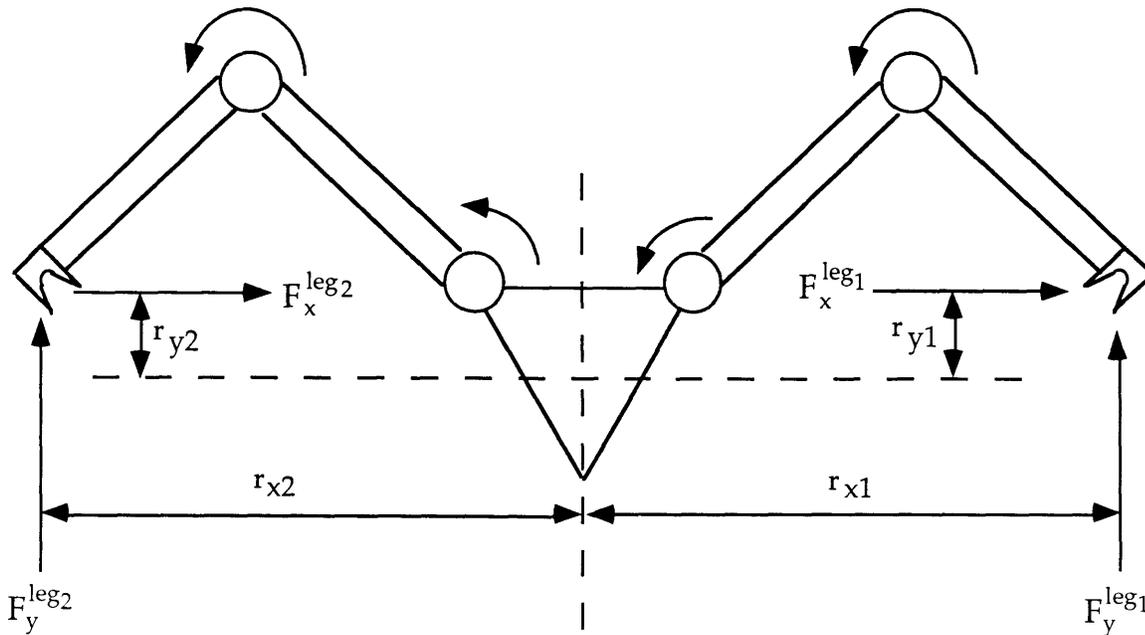


Figure 2.9. Force Analysis of LIBRA

$$\begin{bmatrix} \Sigma F_x \\ \Sigma F_y \\ \Sigma M_\theta \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ -r_{y1} & r_{x1} & -r_{y2} & r_{x2} \end{bmatrix} \begin{bmatrix} F_x^{leg1} \\ F_y^{leg1} \\ F_x^{leg2} \\ F_y^{leg2} \end{bmatrix} \quad (2.8)$$

For the nominal configuration of one leg on each side of the body, the constraint shown in Equation 2.9 is added to obtain one particular solution of the statically indeterminate problem.

$$F_x^{leg1} = F_x^{leg2} = 0 \quad (2.9)$$

Additionally, it is assumed that the limbs are massless such that the total weight of LIBRA (144 oz) is concentrated at the center of the body of LIBRA as seen in Figure 2.10.

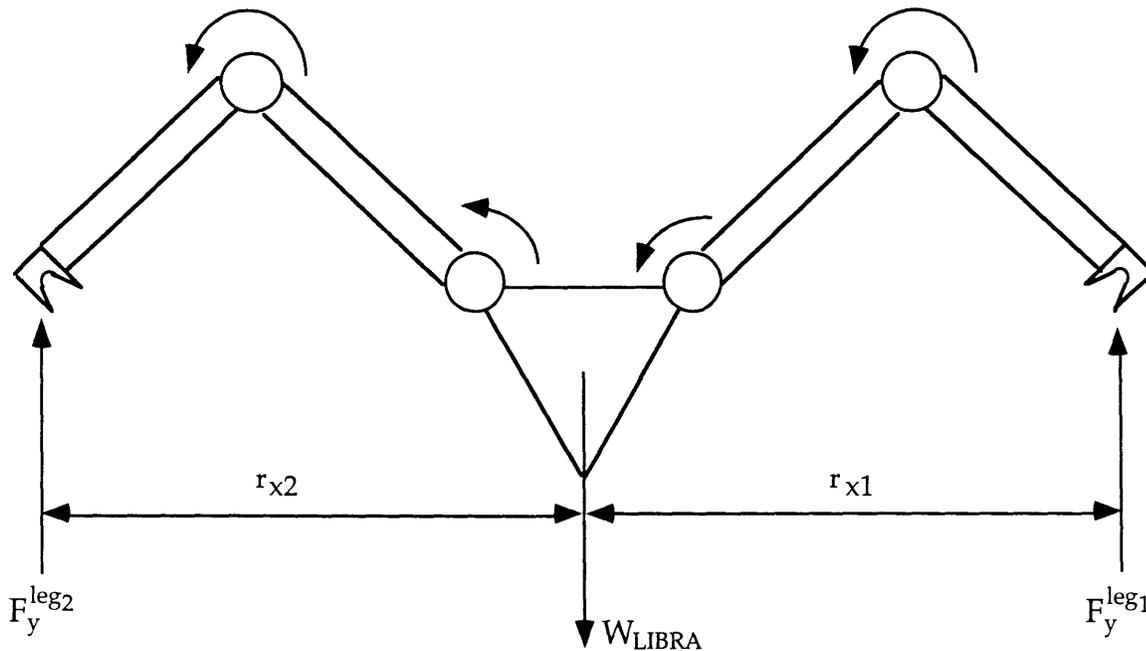


Figure 2.10. Simplified Force Analysis of LIBRA

Using the constraint in Equation 2.9, only the two vertical forces at each leg remain, simplifying Equation 2.8 to the following two equations:

$$\Sigma F_y = F_y^{leg1} + F_y^{leg2} = W_{LIBRA} \quad (2.10)$$

$$\Sigma M_\theta = F_y^{leg1} r_{x1} + F_y^{leg2} r_{x2} = 0 \quad (2.11)$$

Solving the above two equation simultaneously results in:

$$F_y^{leg2} = \frac{W_{LIBRA}}{r_{x1} - r_{x2}} \quad (2.12)$$

$$F_y^{leg1} = W_{LIBRA} - F_y^{leg2} \quad (2.13)$$

For the case with two legs on one side of the body, called the “push pull” case seen in Figure 2.11 , the following alternative constraint equation is added:

$$F_x^{leg1} \tan(\theta_{force}) = F_y^{leg1} \quad (2.14)$$

where:

$$\theta_{force} = \tan^{-1}\left(\frac{r_{y1}}{r_{x1}}\right) \quad (2.15)$$

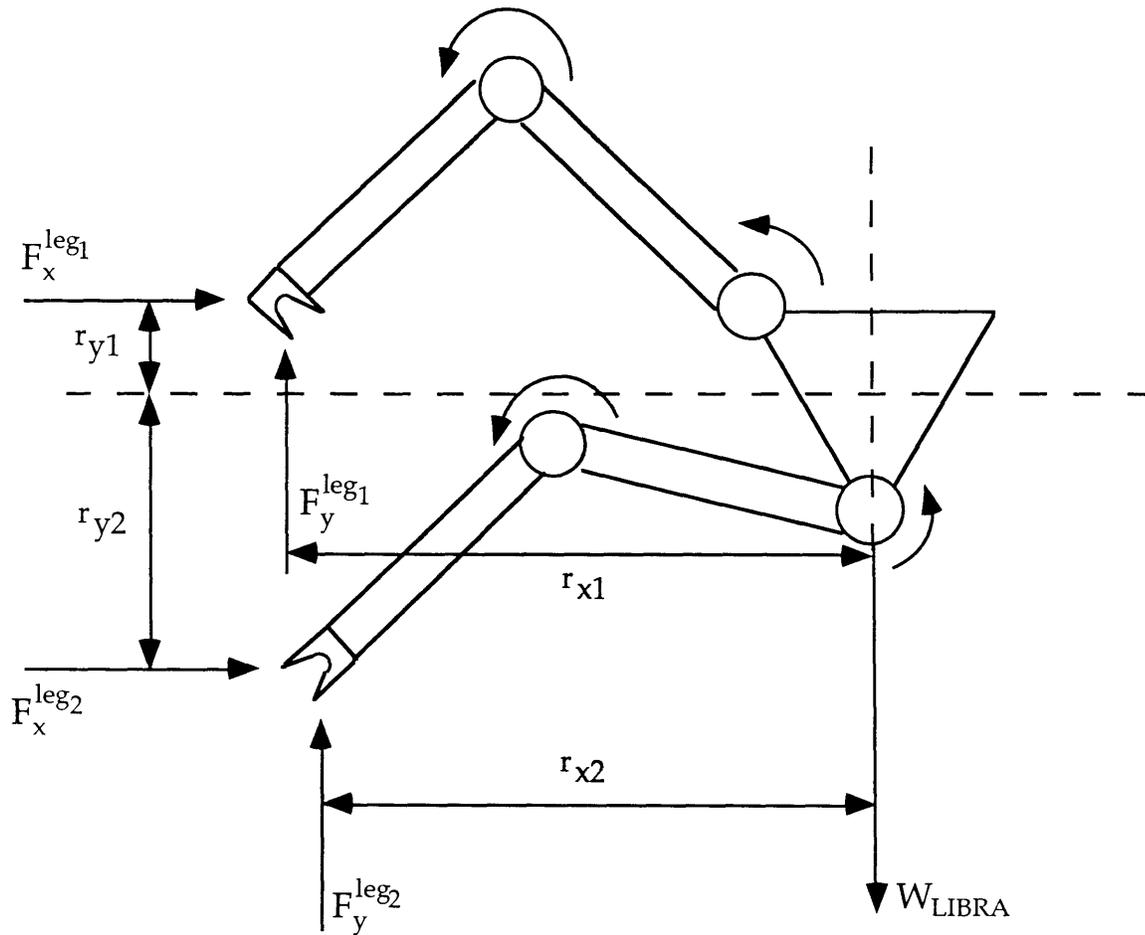


Figure 2.11. Force Analysis for the “Push-pull” Configuration of LIBRA

Adding the above constraint to Equation 2.8 results in:

$$\begin{bmatrix} F_x \\ F_y \\ M_\theta \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ -r_{y1} & r_{x1} & -r_{y2} & r_{x2} \\ \frac{r_{y1}}{r_{x1}} & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} F_x^{leg_1} \\ F_y^{leg_1} \\ F_x^{leg_2} \\ F_y^{leg_2} \end{bmatrix} \quad (2.16)$$

Equation 2.16 is solved for one possible solution to the statically indeterminate problem (for the “push-pull” configuration). This constraint produces a solution in which both leg force vectors pass through the center of LIBRA (which minimized the end effector forces for several configurations). An alternative Lagrange Multiplier constraint optimization technique could have been used to provide a solution with minimum forces or torques.

In some cases LIBRA is allowed to hang by one leg only. In this case:

$$F_y^{leg} = W_{LIBRA} \quad (2.17)$$

Additionally, LIBRA is rotated about the hanging leg until the center of gravity of the body of LIBRA is directly below the hanging leg as seen in Figure 2.12.

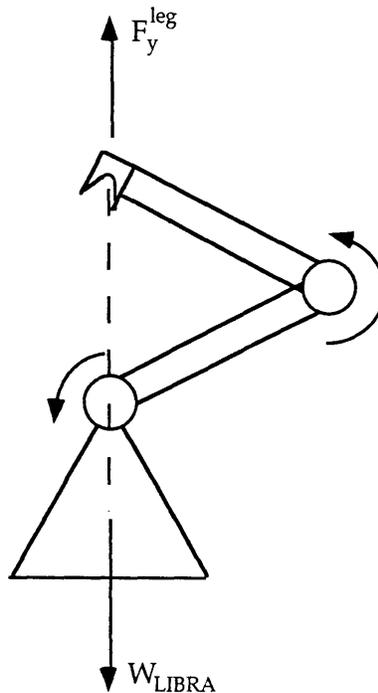


Figure 2.12. LIBRA with only one arm holding

Forces at the endpoint of legs not used (“limb in use”) are assumed to be zero. Once the forces at each leg are determined, they are transformed to the port coordinate frame as discussed in Section 2.3.1 by the following equation:

$$\begin{bmatrix} F_x \\ F_y \end{bmatrix}_{port} = T_p T_b \begin{bmatrix} F_x \\ F_y \end{bmatrix}_{leg} \quad (2.18)$$

where:

$$T_p = \begin{bmatrix} \cos(\theta_{port}) & \sin(\theta_{port}) \\ -\sin(\theta_{port}) & \cos(\theta_{port}) \end{bmatrix} \quad (2.19)$$

$$T_b = \begin{bmatrix} \cos(\theta_{body}) & \sin(\theta_{body}) \\ -\sin(\theta_{body}) & \cos(\theta_{body}) \end{bmatrix} \quad (2.20)$$

The torques at each joint can then be determined from the transpose of the Jacobian matrix of each leg as shown in Equation 2.21.

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}_{leg} = J_{leg}^T \begin{bmatrix} F_x \\ F_y \end{bmatrix}_{port} \quad (2.21)$$

The Jacobian matrix for the leg of LIBRA (J_{leg}) can be found in Appendix A. Actuator torques are then checked against the actuator saturation limit of 3000 oz-in for the LIBRA joints. The actuator saturation constraint is similar to the method used in (Madhani, 1997).

Power consumption is then estimated assuming the actuators are the dominate power consuming element in the system (Dubowsky, 1994). The power consumed in each actuator can be found from the current required to produce a calculated torque by:

$$P = i^2 R = \left(\frac{\tau}{N k_i} \right)^2 R \quad (2.22)$$

where:

τ = actuator torque
 R = armature resistance
 i = motor current
 k_i = motor torque constant
 N = motor gear ratio

It can be shown that for systems such as LIBRA, the joint torques required to statically support the system will dominate over dynamic effects (Dubowsky, 1994).

2.3.3 Additional Analytical Checks

Additionally, simple interference checks are used to insure that limbs do not contact the body, other limbs, or pegs, and to insure that the body remains a “safe” distance from the pegs. A simple stability factor is implemented by reducing the fitness score for hanging with one leg or for using two legs on one side of the body to support the body (as in the “push pull” configuration shown previously in Figure 2.11).

2.4 The Four Tasks

The four tasks shown in Figure 2.13 were developed for the study of the planning problem and were labeled as (from top left clockwise) “ladder”, “H”, “circle”, and “across”. The letters “SG” represent sub-goals that were placed between desired path of “A” to “B”, as discussed previously in Section 2.2.3. Because of physical constraints of the LIBRA system, such as kinematic reach and joint limits, tasks which the real system could accomplish were somewhat limited. Pegs that LIBRA was to travel between needed to be spaced 18-20 inches apart. In addition LIBRA needed pegs approximately every 6 inches to climb. In simulation joint limits could have been removed to determine the planning method on much more complex tasks. However for the scope of this thesis, tasks evaluated were limited to those which the real system could physically accomplish. Peg positions of the four tasks can be found in Appendix C.

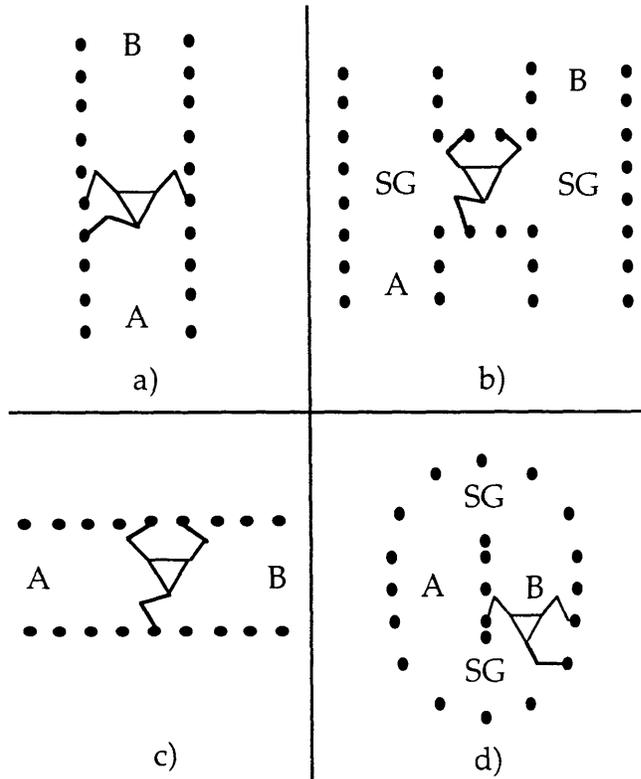


Figure 2.13. The Four Tasks: a) Ladder; b) H; c) Across; d) Circle

2.5 Action Module Planning Results

2.5.1 Ladder Task

Figure 2.14 shows the nominal gait of 191, 392, 291, 300, 3, developed by the action module planning method for successful climbing on the ladder task shown previously in Figure 2.13a. This is not the only gait that exits, however it is the gait with higher stability and lower power consumption due to lower actuator torques. This is the gait which is implemented in the experimental system discussed in Chapter 4.

Additionally, this type of gait is observed in the other tasks, as the most successful plans are those that moved the legs and body such that the body remains between the two legs holding pegs, in order to maximize stability and reduce power consumption.

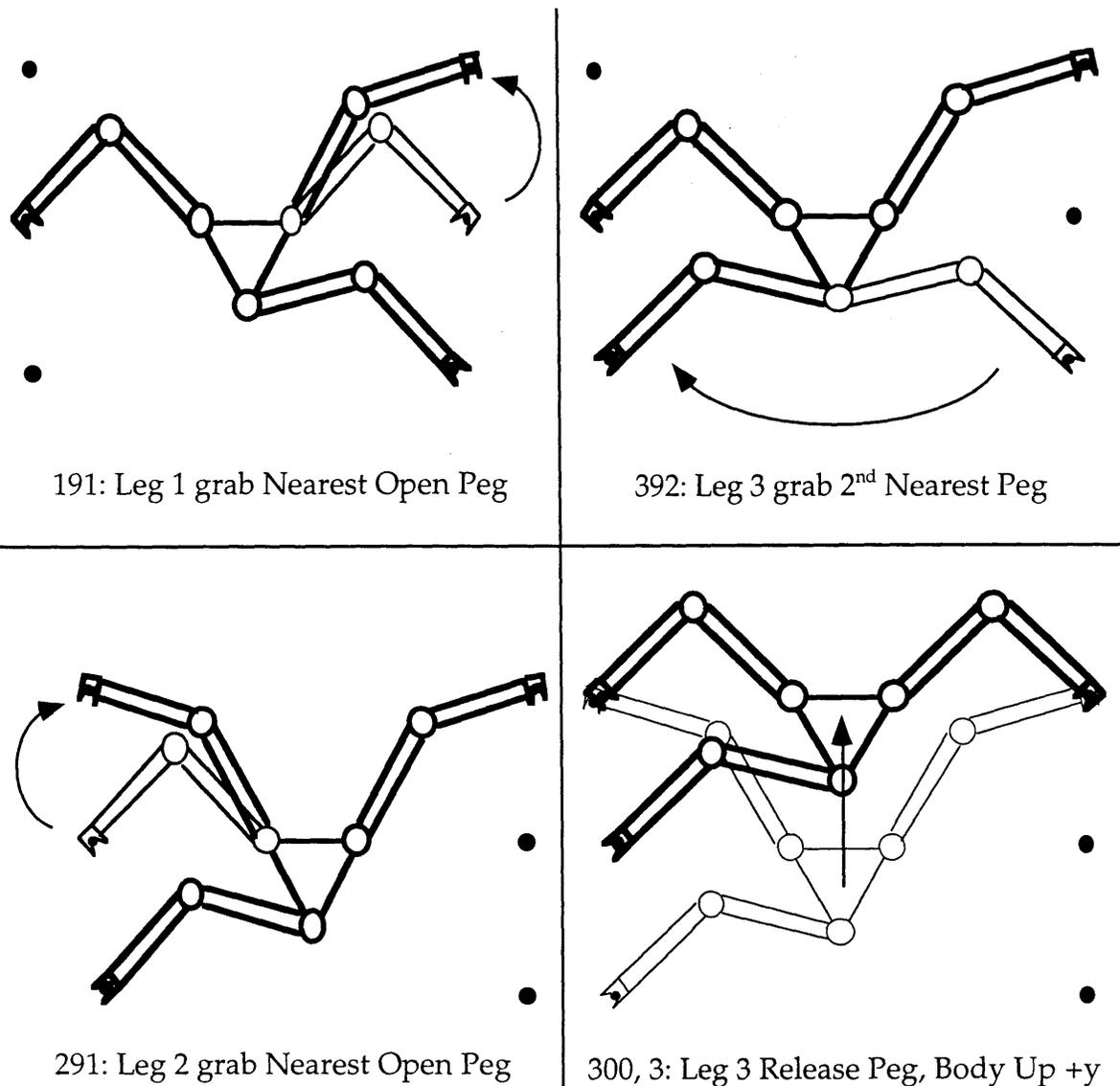
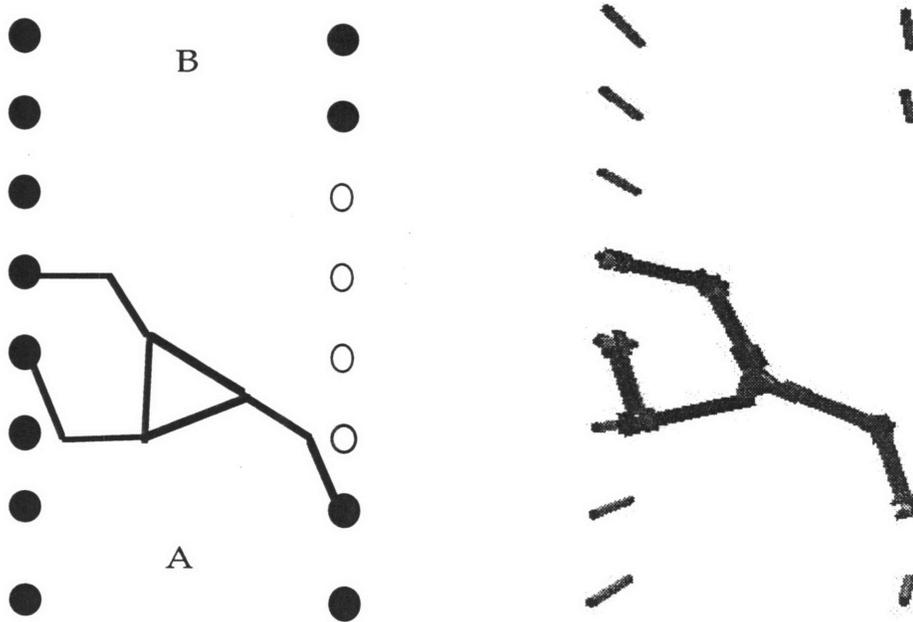


Figure 2.14. Nominal Gait for the Ladder Climb Task

2.5.2 Modified Ladder Task

Figure 2.15 shows the ladder task, modified by removing four pegs from the right half side of the ladder. The planning method was used to find a solution to the task without any rotation of the body, action modules 005 and 006. No solution was obtainable, as it is kinematically impossible to complete the task given the constraint of no body rotation and a one inch reach offset. However, when action modules 005 and 006 are added back to the action module inventory, the planning method is able to derive a

successful plan, shown in Table 2.3, by extending the reach of LIBRA by rotating the body as seen in Figure 2.15. The actual peg numbers selected by the action plans N91-N93 are shown in parentheses.



Schematic of Modified Ladder Task

Animation of GA Derived Plan

Figure 2.15. Modified Ladder Task

Table 2.3. Action Plan for the Modified Ladder Task (step sizes: 1" & 15⁰)

Step #	Action	Step #	Action	Step #	Action	Step #	Action	Step #	Action
1	3	13	6	25	3	37	392(11)	49	3
2	291(5)	14	3	26	3	38	3	50	3
3	6	15	3	27	5	39	1	51	3
4	3	16	293(9)	28	3	40	3	52	3
5	3	17	392(7)	29	3	41	3	53	193(16)
6	391(3)	18	100	30	3	42	3	54	3
7	3	19	3	31	5	43	3	55	3
8	291(7)	20	3	32	3	44	291(15)	56	3
9	3	21	3	33	2	45	5	57	300
10	6	22	3	34	3	46	3	58	3
11	3	23	3	35	193(14)	47	3	59	3
12	3	24	3	36	292(13)	48	391(13)	60	3

This example demonstrates the planning methodology's ability to utilize the full capabilities of the robot, while working within the physical constraints of the system. Additionally, it shows the importance of including all possible actions of the robot as well as the description of each action in the design of the inventory. For example, if a 6 inch step had been used for action module 3, move body +y, the planning methodology would not have been able to find a solution because it was necessary to increment the body in 1 inch steps to position the body between the missing pegs. As the step size is decreased, flexibility is increased at the cost of an increased search space. However, if only high level action modules are used, a solution may be removed, by not allowing the planning methodology to utilize the full capabilities of the robot.

Figure 2.16 shows the evolution of the plan for the modified ladder task. The long flat section of the graph is due to the difficulty in finding a solution to move passed the missing pegs. However, once the planning method increased the reach of the robot by rotating the body, it is able to quickly evolve a plan to move up the ladder which is represented by the large jumps in fitness score after the flat portion of the graph. This show the GA's natural ability to build a successful plan by adding to parts of previously successful plans through crossover.

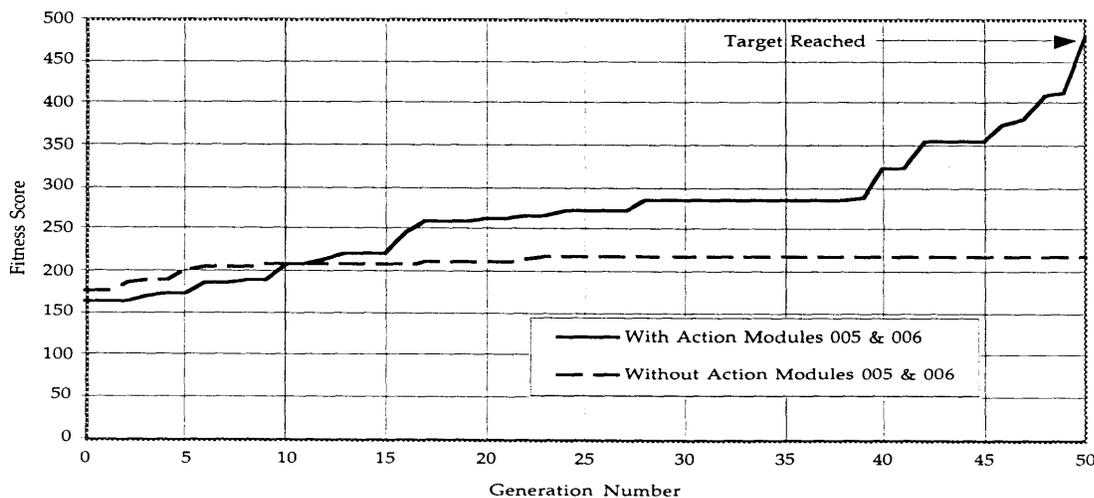


Figure 2.16. Fitness Score for the Evolution of the Modified Ladder Plan

2.5.3 The “H” Task

A hand derived plan was developed for the completion of the H task show previously in Figure 2.13b. The maximum torque required by the plan, was approximately 1900 oz-in, occurring at the location of LIBRA in Figure 2.17. The action module methodology was then used to produce a successful plan that would provide a factor of safety of two from actuator saturation (3000 oz-in). The GA search method is able to find a different solution to the task (shown in Figure 2.17) in order to produce a plan with a maximum torque less than 1500 oz-in, as seen in Figure 2.18, by testing each plan with the physical evaluation tool described previously.

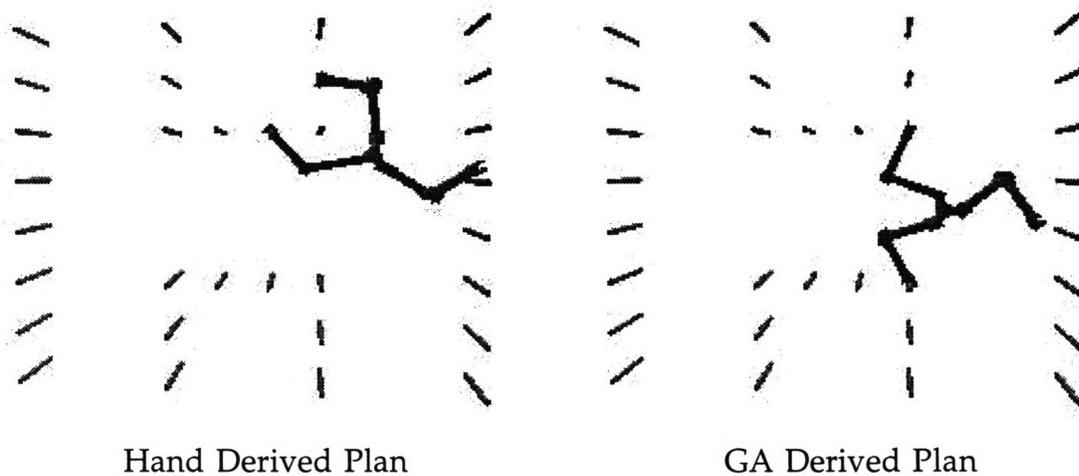


Figure 2.17. “H” Task

Figure 2.19 shows the evolution of the fitness score with generation number and Table 2.4 shows the plan developed which can be compared to the hand derived plan given in Appendix D. Again, the actual peg numbers selected by the action plans N91-N93 are shown in parentheses. Jumps in the fitness score can be attributed to the plan reaching sub-goals (SG) one and two. As seen in Figure 2.19, there is a horizontal slope to the plot around each sub-goal. This is due to the difficulty in finding a solution to move from the transition stages of the “ladder” section of the task to the “across” section, and

from the “across” section back to the “ladder” section, where the sub-goals (SG) in Figure 2.13-b are located. The GA is able to derive a plan relatively quickly for the straight stages of the task, as seen by the larger increases in fitness score in Figure 2.19.

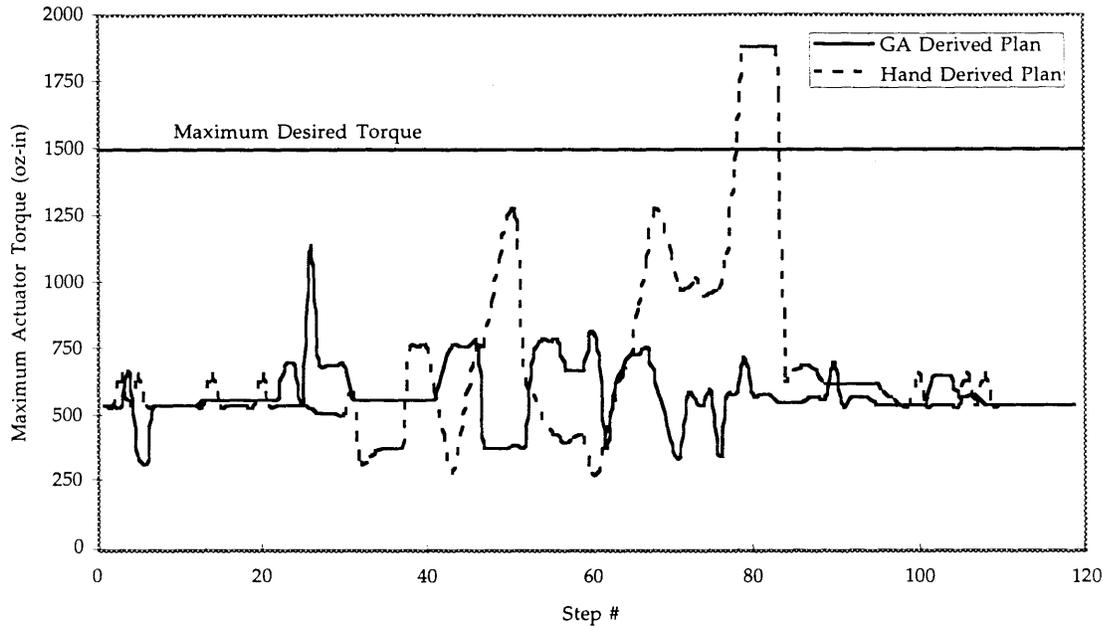


Figure 2.18. Maximum Actuator Torques for the Two Different H Task Plans

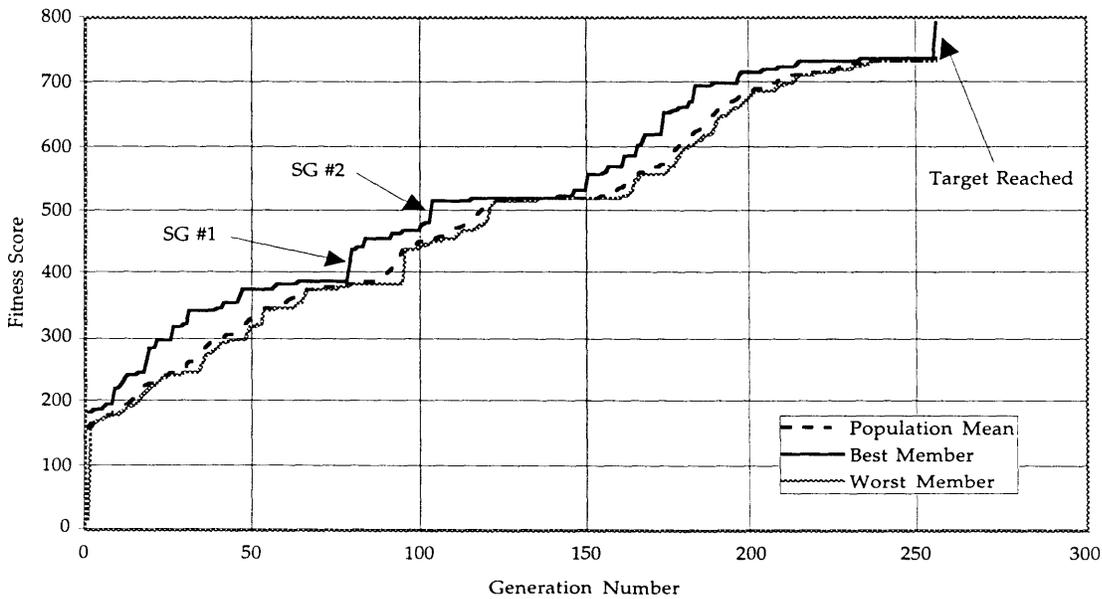


Figure 2.19. Fitness Score for the Evolution of the H Task Plan

Table 2.4. Action plan for H with <1500 oz-in Torque (step sizes: 1" & 15⁰)

Step #	Action								
1	3	25	193(12)	49	4	73	1	97	3
2	3	26	200	50	1	74	1	98	191(22)
3	293(4)	27	3	51	200	75	292(29)	99	3
4	191(11)	28	3	52	300	76	191(28)	100	3
5	3	29	293(6)	53	3	77	3	101	391(24)
6	392(10)	30	392(11)	54	391(18)	78	291(30)	102	3
7	392(3)	31	300	55	200	79	100	103	3
8	3	32	3	56	100	80	3	104	291(32)
9	3	33	3	57	191(17)	81	3	105	3
10	3	34	3	58	291(21)	82	5	106	3
11	3	35	391(5)	59	391(22)	83	3	107	191(23)
12	5	36	3	60	3	84	191(29)	108	3
13	3	37	391(4)	61	100	85	3	109	3
14	392(2)	38	3	62	200	86	1	110	3
15	300	39	391(5)	63	3	87	3	111	3
16	391(3)	40	3	64	3	88	3	112	100
17	3	41	200	65	191(21)	89	291(31)	113	3
18	3	42	300	66	5	90	391(23)	114	3
19	3	43	392(15)	67	1	91	100	115	3
20	3	44	291(11)	68	1	92	3	116	3
21	3	45	191(16)	69	1	93	3	117	3
22	293(5)	46	200	70	1	94	5	118	3
23	3	47	3	71	1	95	3	119	3
24	393(4)	48	3	72	1	96	3	120	3

2.6 Summary and Conclusions

This chapter presented the details of the action module planning module and its application to the climbing robot LIBRA. The methodology, along with information specific to its application to LIBRA, were given. Additionally, details of the genetic algorithm used to evolve a successful plan were presented. Details were also shown for the physics based simulation used to evaluate each plan and ensure that no physical constraints of the system are violated. Finally, the results show that the planning methodology is able to produce plans to execute several different tasks given the physical constraints of the system.

Chapter 3

A Simplified Cartesian Computed Torque Controller

3.1 Introduction

This chapter describes simulation and experimental studies of a simplified cartesian computed torque (SCCT) control scheme for highly geared mobile robots. Section 3.2 presents a discussion of current cartesian control schemes along with their advantages and disadvantages. Section 3.3 introduces the SCCT control scheme based on simplifications made for the control of highly geared robots. Section 3.4 presents experimental and simulation results comparing traditional Jacobian transpose control and the SCCT control scheme. Section 3.5 presents a discussion of implementation of force control using the SCCT controller.

3.2 Current Cartesian Control Schemes

Several controllers have been developed for the control of manipulators in cartesian space. Cartesian controllers have the advantage of being able to set the compliance of the endpoint in Cartesian space, such that the manipulator can be made stiff in one direction and soft in another direction. In this section several popular control schemes are reviewed along with a discussion of the advantage and disadvantage of each.

3.2.1 Jacobian Transpose Control

Jacobian transpose control is a popular cartesian controller, which specifies stiffness and damping in cartesian coordinates, utilizing the static transformation:

$$\tau = J^T F \quad (3.1)$$

The cartesian force, usually developed from a cartesian error, is transformed into joint torques by the above equation to produce the Jacobian transpose controller as shown in Equation 3.2 and Figure 3.1. This transformation is used in impedance control (Hogan, 1985) and stiffness control (Salisbury, 1985) of manipulators, as well as virtual model control (Pratt, 1995) and Coordinated Jacobian Transpose Control (Sunada, 1992) for multi-limbed robots. The controller creates a desired force at the endpoint of the manipulator, then uses the transpose of the Jacobian to relate the endpoint force to desired joint torques supplied to the manipulator by Equation 3.1.

$$\tau = J^T [K_p (X_{des} - X_{act}) + K_d (\dot{X}_{des} - \dot{X}_{act})] \quad (3.2)$$

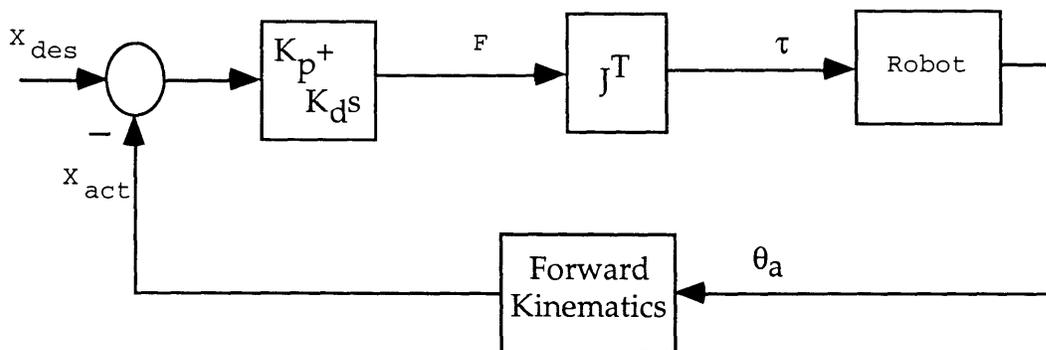


Figure 3.1. Jacobian Transpose Control

However this is a static transformation and therefore leads to tracking error while in motion. As seen by the simple example below the torques produced from the $J^T F$ transformation will not result in a motion (or instantaneous acceleration) of the endpoint in the direction of the force. Figure 3.2 shows a manipulator in which the torques are calculated by the

transformation of Equation 3.1. Starting with the fact that the endpoint velocity can be related to the joint velocity by the manipulator Jacobian by:

$$\dot{X} = J\dot{\theta} \quad (3.3)$$

where:

$$J = \begin{bmatrix} \frac{\partial x}{\partial \theta_1} & \frac{\partial y}{\partial \theta_2} \\ \frac{\partial x}{\partial \theta_1} & \frac{\partial y}{\partial \theta_2} \end{bmatrix} \quad (3.4)$$

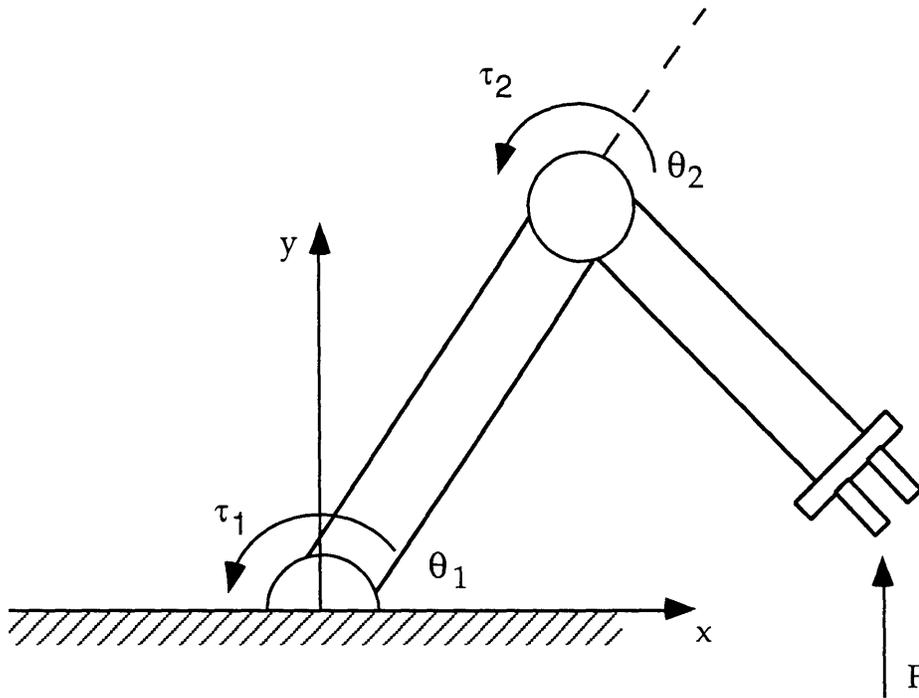


Figure 3.2. $J^T F$ applied to a Manipulator

Differentiating the above equation results in the acceleration of the endpoint shown in Equation 3.4.

$$\ddot{X} = J\ddot{\theta} + \dot{J}\dot{\theta} \quad (3.5)$$

The acceleration of each joint is related to the torque applied at the joint by the manipulator model (Craig, 1986):

$$\tau = H\ddot{\theta} + h + g \quad (3.6)$$

or

$$\ddot{\theta} = H^{-1}(\tau - h - g) \quad (3.7)$$

where:

H = Manipulator Inertia matrix in joint space
h = centripetal and coriolis torques
g = gravity, friction, and other external torques

Substituting the above equation into Equation 3.5 results in:

$$\ddot{X} = JH^{-1}(\tau - h - g) + J\dot{\theta} \quad (3.8)$$

Substituting Equation 3.1, which relates the joint torques to the endpoint forces under Jacobian transpose control, into the above equation results in:

$$\ddot{X} = JH^{-1}(J^T F - h - g) + J\dot{\theta} \quad (3.9)$$

Assuming that $J\dot{\theta}$, coriolis, and centripetal terms are negligible, as well as letting gravity and friction equal zero for simplification, results in the following instantaneous acceleration of the endpoint:

$$\ddot{X} = JH^{-1}J^T F \quad (3.10)$$

As seen from the above two equations, the acceleration of the endpoint is not in the direction of projected force, because $JH^{-1}J^T$ is not equal to the identity matrix in general. This leads to poor performance from traditional Jacobian transpose control in both step responses and tracking as will be seen in Section 3.4.3.

3.2.2. Operational Space Control

Operational Space control (Khatib, 1985), along with other similar computed torque schemes known as resolved acceleration control (Luh, 1980), specify an acceleration at the endpoint (as opposed to a force). Again, the specified acceleration at the endpoint is usually derived from a cartesian error of the manipulator in its workspace. The model of a manipulator in equation 3.6 is equivalent to the following cartesian manipulator model:

$$F = \Lambda \ddot{X} + p + w \quad (3.11)$$

where:

$\Lambda = J^{-T} H J^{-1}$ = Manipulator Inertia matrix in Cartesian space
 $p = -\Lambda J\dot{\theta} + J^{-T} h$ = centripetal and coriolis wrenches

$$w = J^{-T}g = \text{gravity, friction, and other external wrenches}$$

$$\ddot{X} = K_p(X_{des} - X_{act}) + K_d(\dot{X}_{des} - \dot{X}_{act})$$

Using the $J^T F$ transformation to transform the wrench at the endpoint of the manipulator to the joint torques results in the operational space control scheme in Equation 3.12, which is represented in the block diagram in Figure 3.3. Mathematically the above operational space controller simplifies to Equation 3.13 with the joint space model parameters from Equation 3.6.

$$\tau = J^T \left\{ \Lambda \left[K_p(x_{des} - x_{act}) + K_d(\dot{x}_{des} - \dot{x}_{act}) \right] + p + w \right\} \quad (3.12)$$

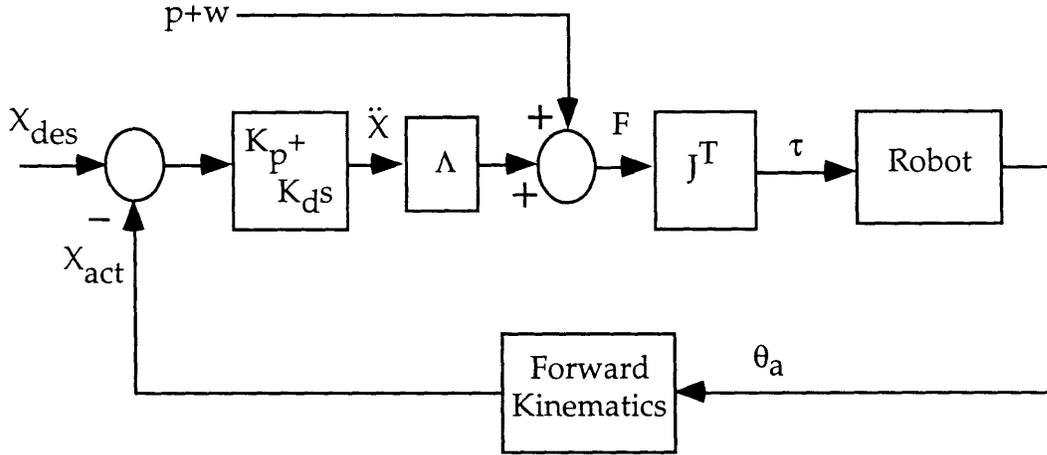


Figure 3.3. Operational Space Control

$$\tau = HJ^{-1} \left[K_p(x_{des} - x_{act}) + K_d(\dot{x}_{des} - \dot{x}_{act}) - \dot{J}\theta \right] + h + g \quad (3.13)$$

Specifying the acceleration at the endpoint ensures that the endpoint of the manipulator moves in the desired direction, assuming the model is correct. Additionally, computed torque controllers, which use the manipulator's dynamic models, can be tuned at higher bandwidths to improve tracking. However, the Jacobian transpose controller, shown previously in Section 3.2.1, was shown to exhibit unstable behaviors at higher bandwidths in (Plumet, 1995).

The increase in position tracking performance of such cartesian computed torque schemes does come at the expense of needing to model the manipulator dynamics. The need to compute the inverse dynamics of a

complex systems, such as a multi-limbed robot, can be computationally expensive, which is a problem for mobile robots carrying their own small control computers.

3.2.3 Resolved Rate Control

Versions of the control scheme introduced in (Whitney, 1969), known as resolved rate control, have also been used as a cartesian controller. However resolved rate control is a joint space controller in which stiffness can not be specified in Cartesian Coordinates.

A pseudo-cartesian stiffness control scheme is sometimes implemented by the use of a gain, K , in some versions of the inverse kinematic resolved rate controller (Siciliano, 1996). However by leaving the damping in joint space, a cartesian direction will either be over-damped or under-damped (depending on the control design), which will lead to a decrease in performance.

As seen in Figure 3.4, these resolve rate controllers attach a “velocity stiffness” in Cartesian space. However the controller is still a joint PD controller. If the wish is to soften the stiffness in one direction, the system will suffer from decreased performance due to the fact that the system will be incorrectly damped in Cartesian space (since the damping is still in joint space). Therefore the system does not allow for optimal cartesian-space gain tuning.

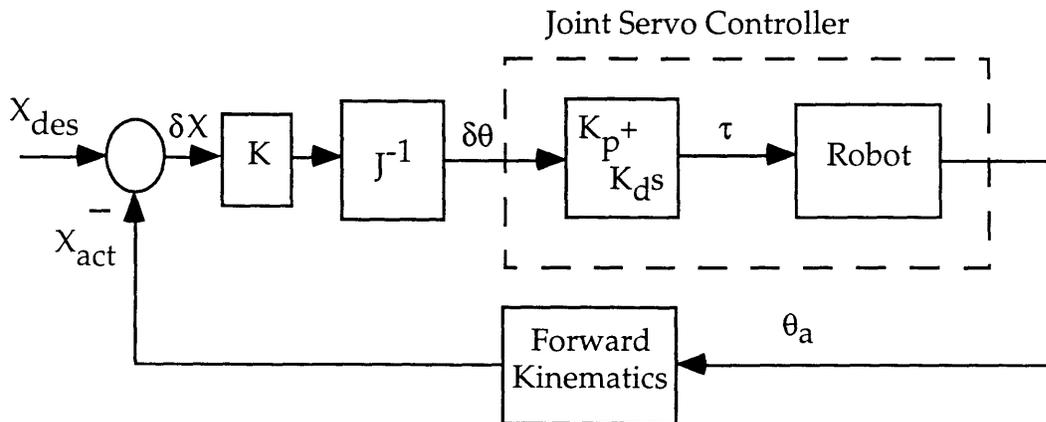


Figure 3.4. Resolved Rate Control

3.2.4 Admittance Control

Admittance or accommodation control (Whitney, 1977), combines force sensing with resolved rate control. The K_{admit} term seen in Figure 3.5 has units of admittance (velocity/force).

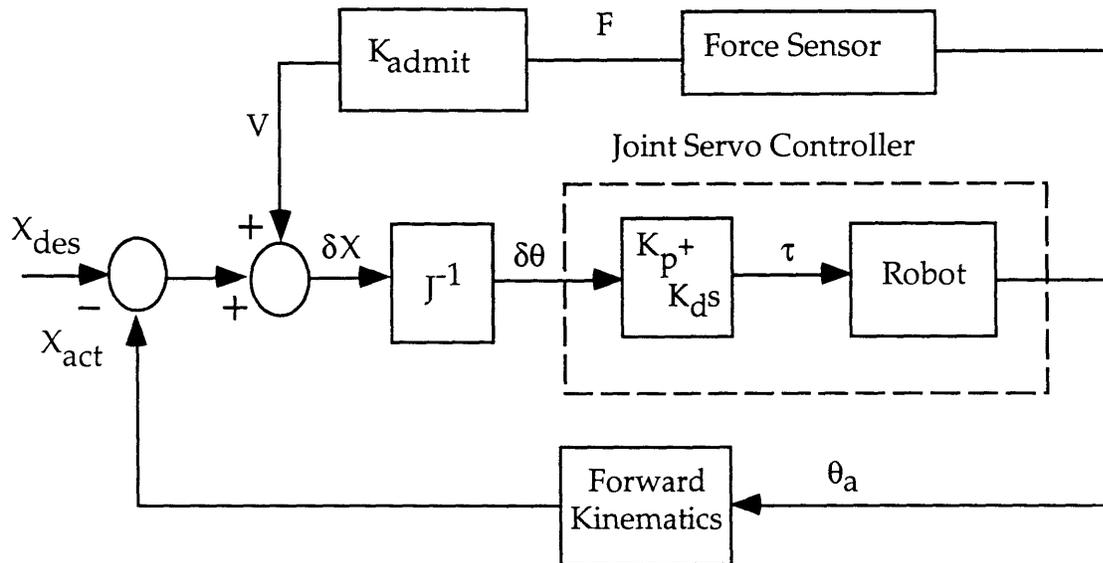


Figure 3.5. Admittance Control

The sensed forces at the endpoint result in a desired velocity which is then fed to the joint servo controller. This control scheme is sometimes used to emulate traditional impedance type controllers in which a manipulator is controlled at the joint servo level, as opposed to joint torque controlled manipulators, as discussed in (Colbaugh, 1995). This method has the drawback of needing force sensors at the endpoint of the manipulator which may not always be available on all limbs of a multi-limbed robot.

3.3 Simplified Cartesian Control Scheme

A Cartesian controller was desired for multi-limbed robots, which could utilize Cartesian stiffness, be computationally simple, and have higher performance than simple Jacobian transpose control. The goal was to design a controller in which the stiffness as well as the commanded positions are set

in cartesian space. The controller would then utilize a virtual spring as seen in Figure 3.6 to move the endpoint of the manipulator.

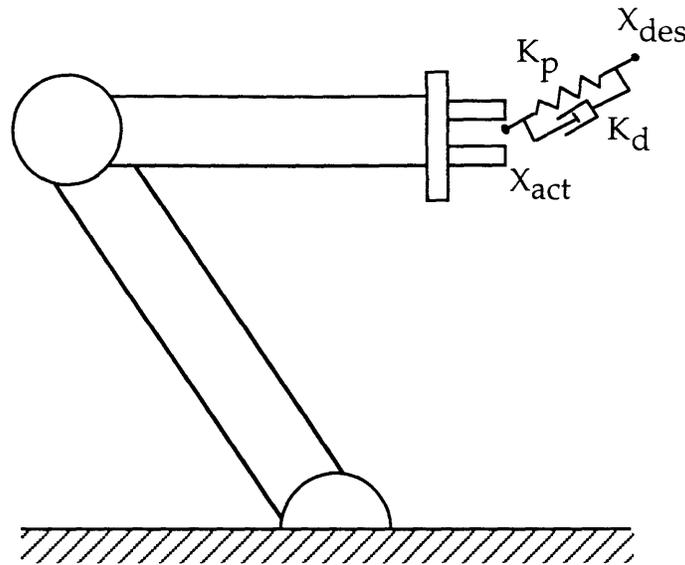


Figure 3.6. Cartesian Control of a Manipulator (Sunada, 1992)

Many multi-limbed robots are designed to be lightweight. However, small actuators which must still support a large load, utilize large gearing to produce the required torque output with little weight. High gear ratios have many disadvantages such as high joint friction and large backlash, but many control simplifications can be made for a light manipulator with high gear ratios, since the actuator inertia will dominate the system. This is a major reason for the use of joint PD controllers in walking machines. However, many reasons given in Section 1.3 demonstrate the utility of controlling these robots in cartesian space. Good results were also shown previously for the use of a cartesian based control of the multi-limbed robot LIBRA on a ladder in (Sunada, 1992). Utilizing these arguments, a simple, yet effective, cartesian controller is developed.

3.3.1 Simplification of the Cartesian Computed Torque Control Scheme

Simplifications can be made to the conventional cartesian computed torque control scheme shown previously in Equation 3.13 for highly geared manipulators by neglecting the centrifugal and coriolis terms. The control scheme can be simplified additionally by assuming that $J\ddot{\theta}$ is negligible to obtain:

$$\tau = HJ^{-1} \left[K_p (X_{des} - X_{act}) + K_d (\dot{X}_{des} - \dot{X}_{act}) \right] \quad (3.14)$$

For highly geared motors, the manipulator inertia matrix, H , is primarily dominated by the joint inertia and therefore not configuration dependent. This results in the following simplified control scheme:

$$\tau = \Phi J^{-1} \left[K_p (X_{des} - X_{act}) + K_d (\dot{X}_{des} - \dot{X}_{act}) \right] \quad (3.15)$$

where:

Φ = non-configuration dependent inertia matrix.

The Φ matrix can be normalized and external forces fed forward to produce the SCCT controller shown in Equation 3.16.

$$\tau = K' J^{-1} \left[K_p (X_{des} - X_{act}) - K_d \dot{X}_{act} \right] + J^T F_{ext} \quad (3.16)$$

where:

K' = normalized non-configuration dependent inertia matrix.

The SCCT control scheme is shown in its block diagram representation in Figure 3.7. Although the SCCT control scheme requires a simple (and constant) inertia model of the manipulator, this model is simpler and easier to obtain and implement than the full dynamic model of the manipulator. Additionally, for a highly geared manipulator such as LIBRA, the K' gain matrix can be approximated by a simple normalized diagonal matrix of the joint inertias as seen later in Section 3.4.

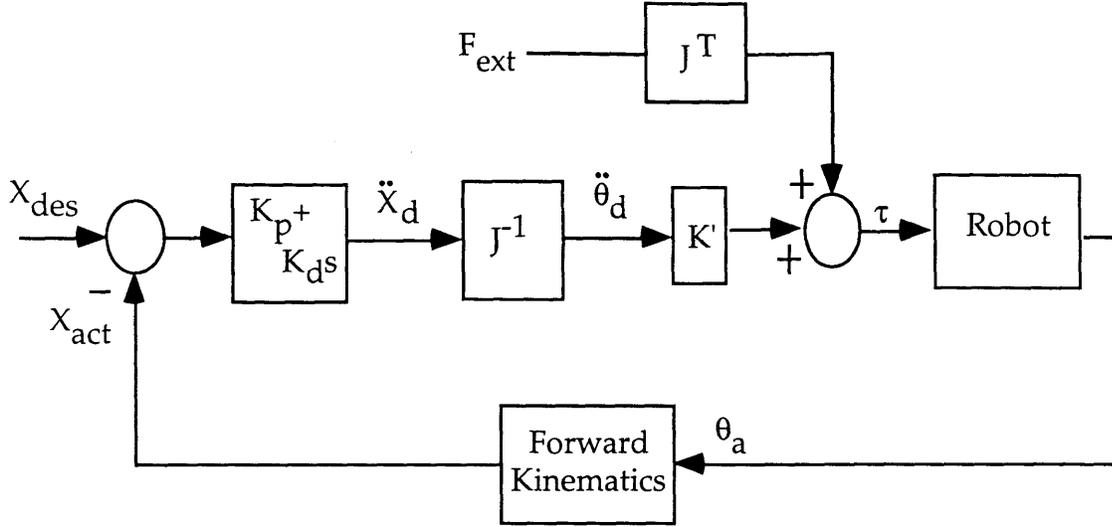


Figure 3.7. Simplified Cartesian Computed Torque (SCCT) Control

3.3.2 A Different Perspective on the Controller (Implicit Admittance Control)

Starting with relationship between the endpoint velocity and joint velocities:

$$\dot{X} = J\dot{\theta} \quad (3.17)$$

or

$$\dot{\theta}_{des} = J^{-1}\dot{X}_{des} \quad (3.18)$$

Utilizing the idea from an admittance type controllers, in which a force at the endpoint results in a desired velocity as:

$$\dot{X}_{des} = K_{admit}F \quad (3.19)$$

where:

$$K_{admit}F = K_p(X_{des} - X_{act}) - K_d\dot{X}_{act} \quad (3.20)$$

As seen in Figure 3.4, some resolved rate controllers use:

$$\dot{X}_{des} = K(X_{des} - X_{act}) \quad (3.21)$$

However when differences in cartesian stiffness are desired, it is better to factor out the damping gain K_d for reason mentioned earlier in Section 3.2.3. Combining Equation 3.19 and 3.20 and substituting into Equation 3.18 results in:

$$\dot{\theta}_{des} = J^{-1} \left[K_p (X_{des} - X_{act}) - K_d \dot{X}_{act} \right] \quad (3.22)$$

Again borrowing the following idea from resolve rate control:

$$\tau = (K_{p\theta} + K_{d\theta} s) \dot{\theta}_{des} \quad (3.23)$$

we obtain:

$$\tau = (K_{p\theta} + K_{d\theta} s) \left\{ J^{-1} \left[K_p (X_{des} - X_{act}) - K_d \dot{X}_{act} \right] \right\} \quad (3.24)$$

Normalizing both $K_{p\theta}$ and $K_{d\theta}$ in order to place the control into K_p and K_d respectively, and feeding forward external forces, we obtain:

$$\tau = (K'_{p\theta} + K'_{d\theta} s) \left\{ J^{-1} \left[K_p (X_{des} - X_{act}) - K_d \dot{X}_{act} \right] \right\} + J^T F_{ext} \quad (3.25)$$

where:

$$0 < K'_{p\theta} < 1$$

$$0 < K'_{d\theta} < 1$$

However, $K_{d\theta}$ can be omitted at the joint level since all real motors will have a small amount of damping at the joint, reducing Equation 3.25 to the SCCT control scheme shown previously in Equation 3.16 (by letting $K' = K'_{p\theta}$). Although the two controller equations can be identical, this type of controller implies that velocities are commanded (as opposed to accelerations in the previous section) as seen in the block diagram in Figure 3.8.

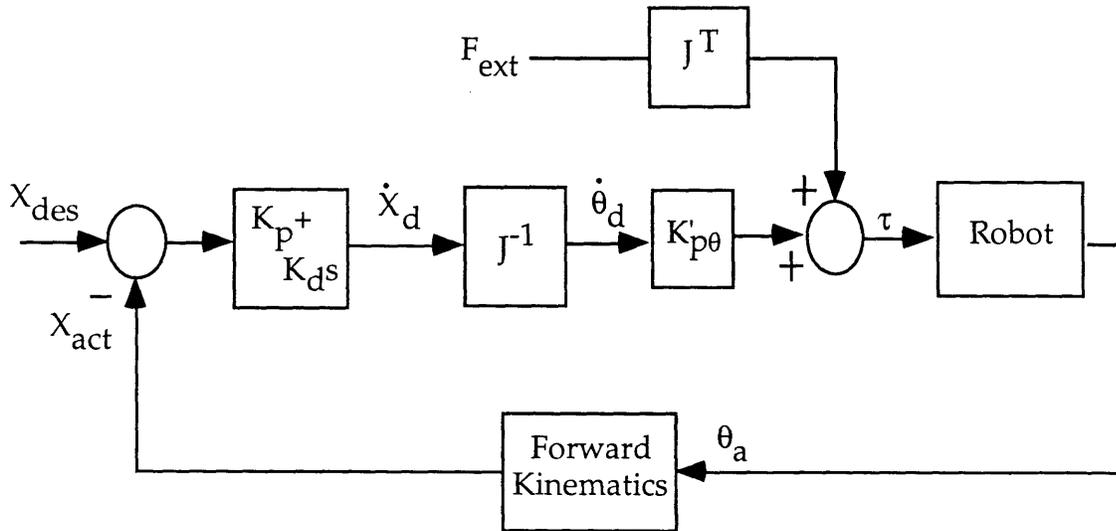


Figure 3.8. Implicit Admittance Control

It is important to note that Equation 3.24 is a general version of Equation 3.14. If joint gains (K_{p0} and K_{d0}) are selected based on the joint inertia, the scheme resembles the cartesian computed torque control scheme or a modification to the even more general control scheme found in (Samson, 1991).

The “gain ratio” K'_{p0} is used to effectively equal out each joint, which may require additional tuning if each joint varies in its dynamic characteristics, including effects such as Coulomb friction. The rest of the stiffness is applied at K_p and K_d . In most systems the K'_{p0} matrix will simply be a ratio of the joint inertias. However for a system where the inertia does not dominate the joints, such as a hydraulic manipulator, or a motor with gearing in a viscous lubricant, the damping ratio of these systems may better serve as the basis for K'_{p0} . Additionally if a system is controlled by a voltage amplifier, K'_{p0} would simply be the back emf constant which relates torque to velocity.

3.3.3 Additional Issues

The term, F_{ext} includes all external forces such as environmentally applied forces and joint friction, as well as forces from the manipulator dynamics such as un-modeled accelerations. External forces applied from the environment can be sensed with a force torque wrist sensor (Khatib, 1987). Joint friction can also be sensed and rejected through torque control using a wrist sensor (Williams, 1995) or a force torque sensor mounted at the base of the manipulator (Morel, 1995).

Computed torque control schemes do have the disadvantage of using the inverse of the Jacobian, which can sometimes be singular. This issue has been addressed in many ways, such as using a pseudo inverse for redundant manipulators in (Siciliano, 1996) or by using a damped least squares Jacobian inverse which yields a damped least squares solution near singularities as in (Wampler, 1988). For this thesis, care was taken to simply avoid singularities.

Additionally, since this type of controller specifies desired joint velocities it is important to work within the limits of the actuators described in (Kircanski, 1997).

3.4 The SCCT Controller under Position Control

3.4.1 Implementation

The SCCT control scheme (Equation 3.16) is implemented on one limb of LIBRA through the block diagram shown previously in Figure 3.7. The experiments were done horizontally, such that it is assumed that no external forces, including gravity, are acting on the limb. The K' matrix used in the following experiments is the ratio of the two joint inertias of the limb shown in Equation 3.26.

$$K' = \begin{bmatrix} \frac{I_1 N_1^2}{I_2 N_2^2} & 0 \\ 0 & \frac{I_1 N_1^2}{I_2 N_2^2} \end{bmatrix} \quad (3.26)$$

where:

I_i = The inertia of the i^{th} joint
 N_i = The gear reduction at the i^{th} joint

Because the joint motor inertia was the same for all joints of LIBRA ($I_1 = I_2$), the K' gain matrix simplifies to:

$$K' = \begin{bmatrix} \frac{N_1^2}{N_2^2} & 0 \\ 0 & \frac{N_1^2}{N_2^2} \end{bmatrix} \quad (3.27)$$

The Jacobian and Inverse Jacobian matrices of the limb used in the following experiments can be found in Appendix A.

3.4.2 Model Verification

A model of one limb of LIBRA was produced, which is simply a two link planar manipulator. The coriolis and centripetal terms were neglected, as well as the inertia matrix simplified to a diagonal matrix of the joint inertias. The simplified model is shown in Equation 3.28.

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} I_1 & 0 \\ 0 & I_2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} - \begin{bmatrix} d_1 & 0 \\ 0 & d_2 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} - \begin{bmatrix} \tau_{f1} \\ \tau_{f2} \end{bmatrix} \quad (3.28)$$

where:

- I = joint inertia
- d = viscous joint damping
- τ_f = coulomb friction torque

Rough estimates of the joint friction, damping terms, and joint inertias were obtained by calculating the acceleration and steady state velocity of the motor due to a constant torque input. Table 3.1 provides a list of the values for the model terms for all six joints of LIBRA.

Table 3.1. Model Parameters of the LIBRA Joints.

	Joint #1	Joint #2	Joint #3	Joint #4	Joint #5	Joint #6
I (kgm ²)	0.4020	0.4020	0.4020	0.7652	0.7652	0.4020
d (Nmsec/rad)	1.1128	0.7143	0.5263	0.5706	0.2697	0.5564
τ_f (Nm)	0.6124	0.4505	0.5118	0.1207	0.2656	0.5861
N	574	574	574	792	792	574

A comparison of the simplified model and the real system was done to verify that the joint inertia's dominate the system, which is the assumption used to simplify the cartesian computed torque controller. A sinusoidal torque input, plus an offset torque to insure a positive joint velocity, was used to drive each joint independently. As seen in Figure 3.9, the simple model matches the experimental system fairly well.

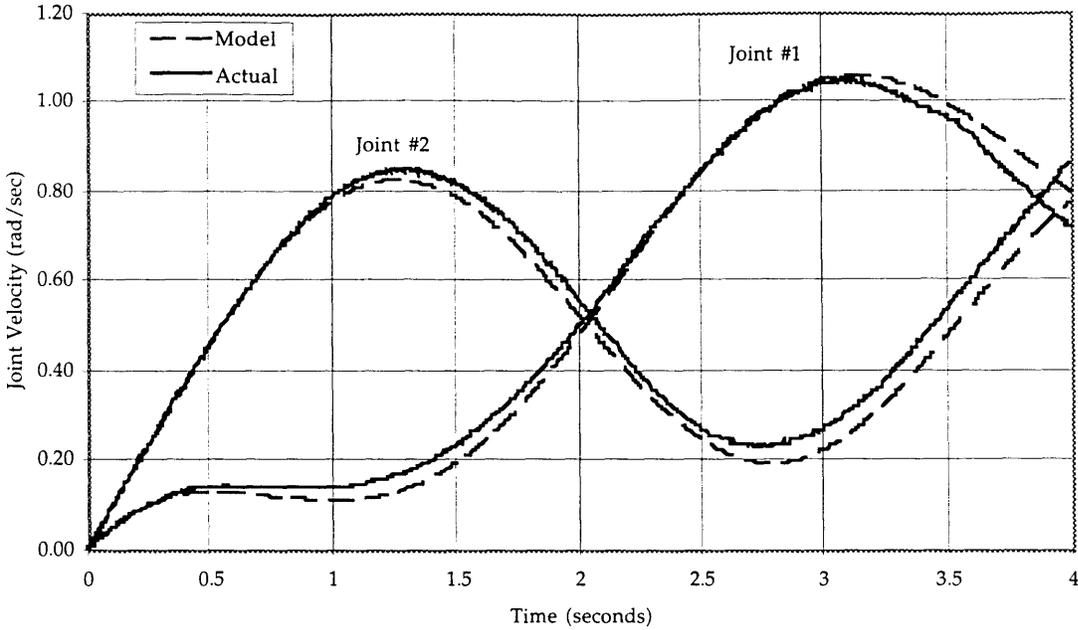


Figure 3.9. Comparison of Experimental System and Simplified Model.

3.4.3 Comparison of the Jacobian Transpose and SCCT Control

This section presents some experimental comparisons of the Jacobian transpose and SCCT control schemes under position control. A nominal configuration of one limb, similar to the configuration shown in Figure 3.2, is used in the following experiments.

A simple experiment is used to compare the $J^T F$ transformation, used with Jacobian transpose control, and the $K^* J^{-1} \ddot{X}_d$ transformation, used in the SCCT controller. The torques produced from a desired -y force, for the $J^T F$ transformation, and desired -y acceleration, for the $K^* J^{-1} \ddot{X}_d$ transformation, were input to one limb of LIBRA in an open-loop fashion. As seen in the Figure 3.10, the torques produced from the $J^T F$ transformation do not cause the endpoint of the manipulator to move in the direction of the desired force as was discussed previously in Section 3.2.1. However the torques generated from the $K^* J^{-1} \ddot{X}_d$ transformation do a better job of moving the endpoint of the manipulator in the desired downward direction. The initial instantaneous

acceleration is in the desired downward direction for the $K J^{-1} \ddot{X}_d$ transformation.

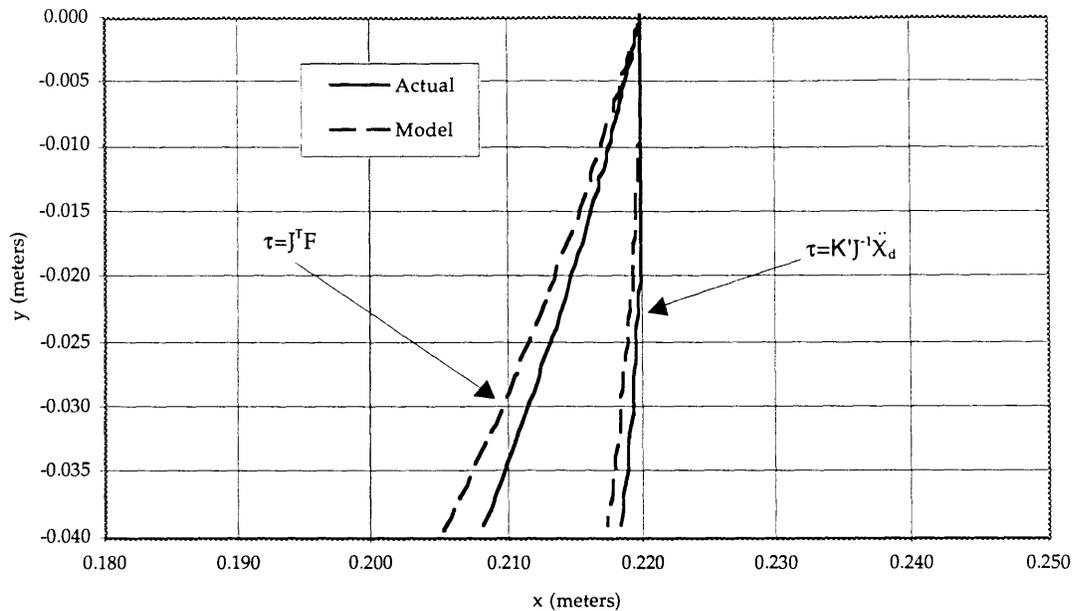


Figure 3.10. Comparison of $J^T F$ and $K J^{-1} \ddot{X}_d$ Transformations.

One of the major assumptions in the development of the SCCT controller is that $\dot{J}\dot{\theta}$ was negligible, compared to $H\ddot{\theta}$. In the beginning of the experiment, the manipulator Jacobian is relatively constant compared to the joint accelerations such that the assumption holds true. This creates the initial instantaneous acceleration of the endpoint in the desired downward direction. However, as the system approaches steady state, $\dot{J}\dot{\theta}$ is not negligible compared to the joint accelerations. This is the reason for the stray of the endpoint, in time, from its directly downward motion.

Figure 3.11 shows the use of the simple model in Equation 3.27 to compare a 2 cm step response, to a commanded position of (0.22,-0.02), for one limb of LIBRA under Jacobian transpose and SCCT control.

Figure 3.12 compares the two control schemes, experimentally, to the same 2 cm step response shown previously in simulation. As seen in the two figures the simplified cartesian controller is able to hold the manipulators

desired x position while moving in the desired y direction. However the Jacobian transpose controller is unable to hold the desired x position of the manipulator due to the fact that the $J^T F$ transformation does not create a motion in the desired direction. The SCCT controller is seen to be able to reject the disturbance of neglecting the $\dot{J}\theta$ term.

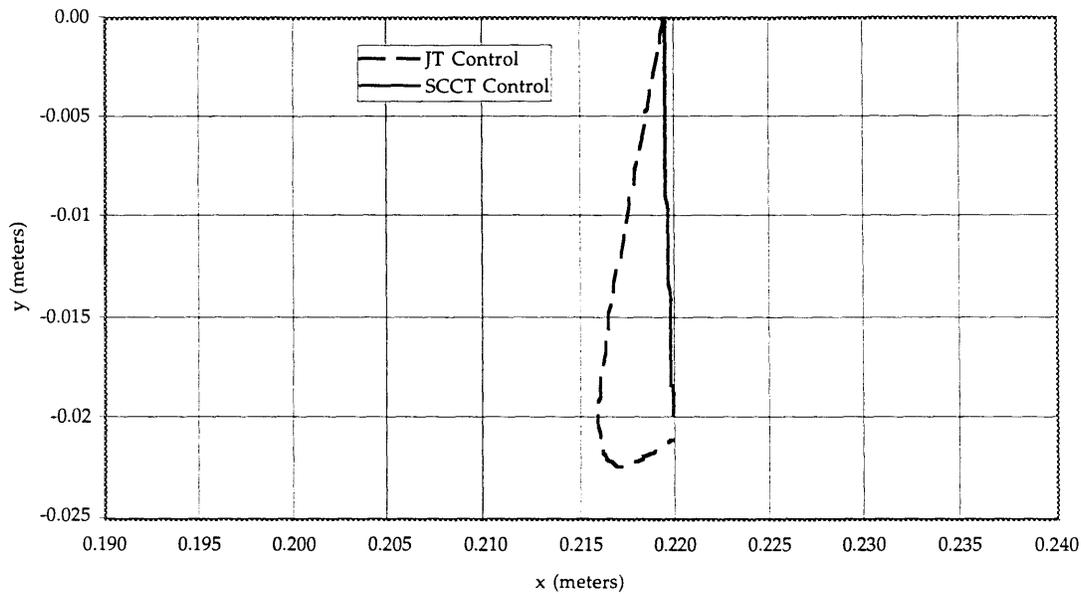


Figure 3.11. Simulated Step Response of the Two Cartesian Controllers

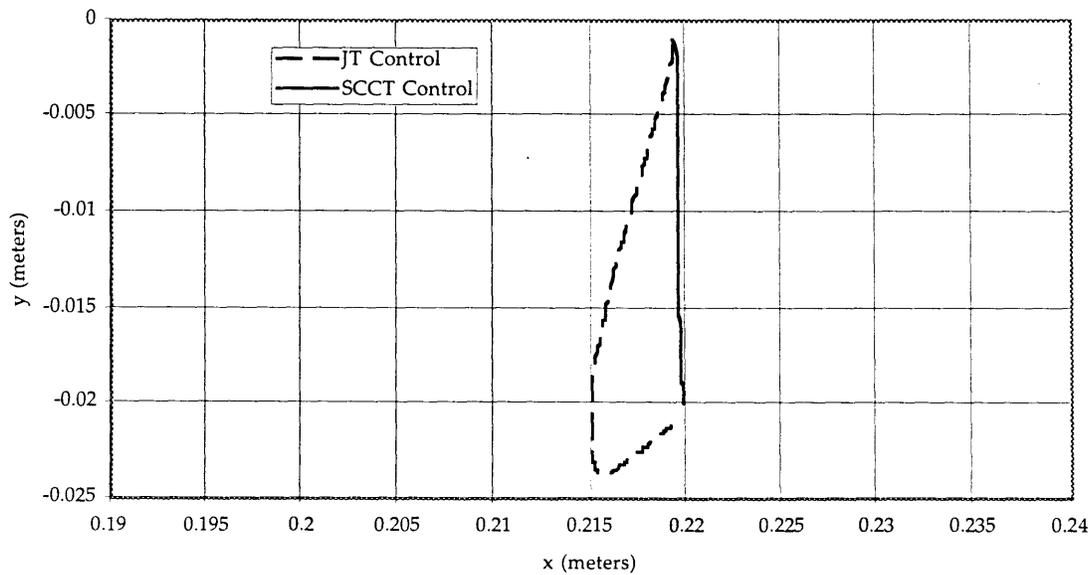


Figure 3.12. Experimental Step response of the Two Cartesian Controllers

Figure 3.13 compares the two controllers in tracking a commanded circle, of radius 5 cm, as opposed to a step response. As seen from the figure, the Jacobian transpose controller suffers slightly while tracking a slow commanded path (about 6 seconds per circle). The Jacobian transpose controller was not able to track a fast commanded path (about 3 seconds per circle). However, the SCCT controller (with the approximately the same effective stiffness) tracks the circle almost perfectly, even at the faster commanded path.

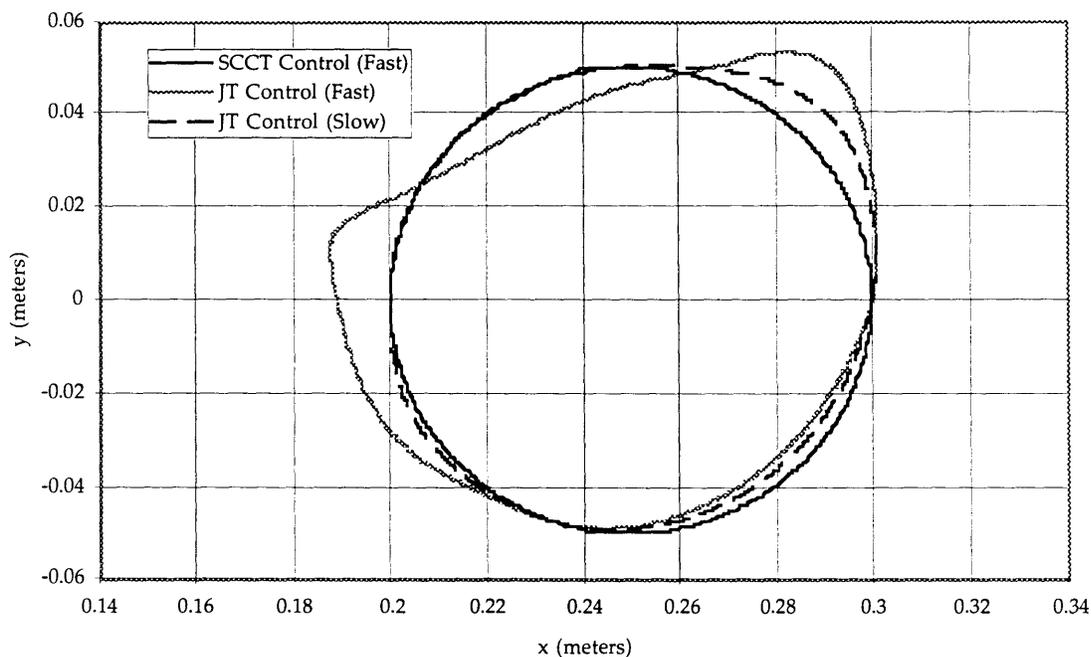


Figure 3.13. Experimental Path Tracking of the two Cartesian Controllers

Figure 3.14 shows the ability to produce a higher bandwidth controller using the SCCT control scheme over the Jacobian transpose controller, in a cartesian direction, while maintaining the flexibility of setting the other cartesian stiffness very soft. In the experiment, the end of the manipulator is pushed away from its commanded center position of (0.22,0.0). The cartesian stiffness is set very high in the y direction, while allowing compliance in the x direction. Once the disturbance is removed, the limb moves back to its center position. As seen in the figure, the SCCT controller is more capable of

maintaining the desired y position of 0.22 meters, while allowing the manipulator to be “pushed away” from its desired x position of 0.0 meters.

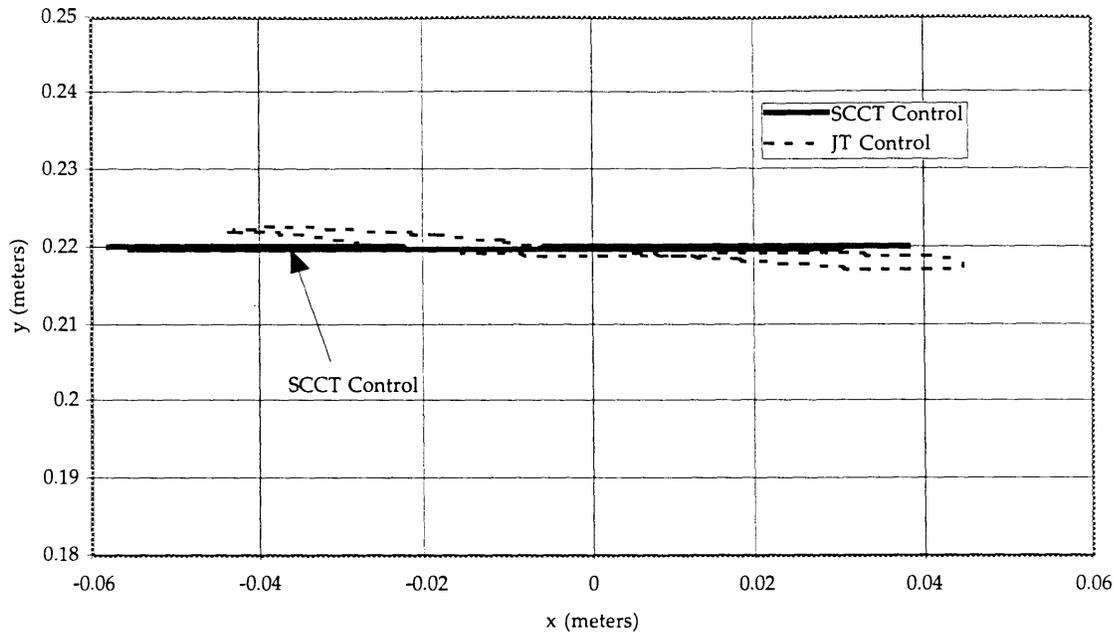


Figure 3.14. Stiffness Selection in Cartesian Space.

3.5 Force Control with the SCCT Controller

This section provides a discussion on the implementation of force control of a manipulator using the SCCT control scheme. Due to time constraints, no results of the force control strategies are presented in this thesis. Force control can be implemented with the SCCT controller by using the hybrid force control method in (Raibert, 1981). If the force control is done against a known environment then the result is the same control scheme as the hybrid controller, consisting of resolved rate control in motion sub-spaces and Jacobian transpose in perpendicular force sub-spaces shown in Figure 3.15.

However, Jacobian transpose control should only be applied in the static force sub-space direction, which is perpendicular to the motion sub-space, since $J^T F$ is a static transformation. This type of control approach requires knowledge of the environment (angle θ) in order to ensure that J^T is

not applied in a motion subspace. The same procedure taken with the motion and force subspaces discussed in (Khatib, 1987) can also be used to implement hybrid position and force control using the SCCT controller.

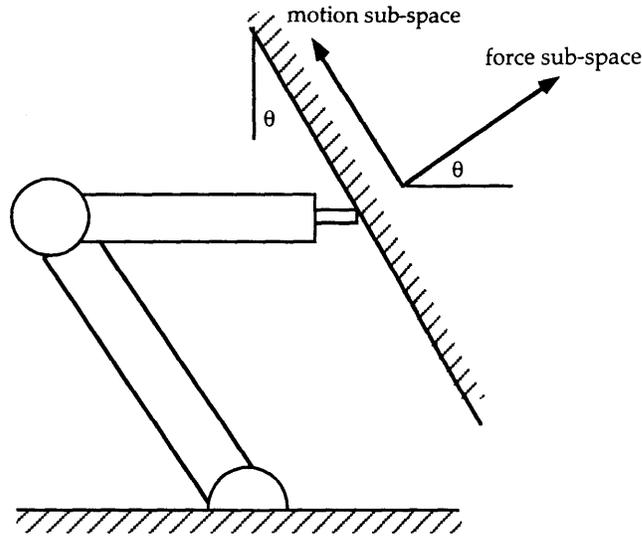


Figure 3.15. Force and Motion Sub-spaces of an Environment

Alternatively, force control can be implemented with the SCCT controller through an implicit force control strategy. This is done by controlling a position against the environment, with knowledge of the environmental stiffness.

3.6 Summary and Conclusions

This chapter presented the development of a simplified cartesian computed torque (SCCT) control scheme for multi-limbed robots. The SCCT control scheme is developed based on the fact that many multi-limbed robotics systems have high gear ratios, which allows for simplification to the full dynamic computed torque control scheme. Simulation and experimental results show the validity of the assumptions and the effectiveness of the SCCT controller for the position control of one limb of a highly geared multi-limbed robot. Additionally, a brief review of cartesian based control schemes was presented, along with the advantages and disadvantages of each.

Chapter 4

Experimental Climbing of LIBRA

4.1 Introduction

This chapter describes the experimental application of the action module planning and the simplified cartesian computed torque (SCCT) control scheme to the experimental LIBRA system. Section 4.2 presents the basis for using cartesian control on LIBRA. Section 4.3 describes the experimental setup, including hardware, for the climbing system. Section 4.4 presents the details of applying the SCCT controller to the peg climbing problem. Finally, Section 4.5 presents experimental results for the LIBRA climbing between pegs on the peg board.

4.2 Basis of Cartesian Control for LIBRA

As mentioned previously, many reasons exist for controlling the body of a multi-limbed robot in cartesian space by controlling the interactions of the limbs with the environment. Figure 4.1 shows a typical walking robot on uneven terrain. As seen in the figure, a robot which is soft in the y direction, would allow it to behave compliantly on the uneven terrain. The experimental control of LIBRA climbing on pegs, without actively grabbing the pegs, presents a similar problem. The use of a cartesian based controller, in which the effective stiffness between both the body and ground and the legs and ground can be specified (as seen in Figure 4.1), was shown to be effective for controlling the multi-limbed climbing robot LIBRA in (Sunada, 1992).

Once again, it is desired to control the body of LIBRA in cartesian space while insuring that the limbs remained on the pegs, even through external disturbances as seen in Figure 4.2. A typical disturbance for the system is the swing leg applying an unknown force against the peg which it was trying to grab. This occurs from controlling the position of the leg towards a partially known peg location. However, setting the compliance of the body along with the compliance of the swing leg allows grabbing of each peg without knocking any of the other legs off of their pegs.

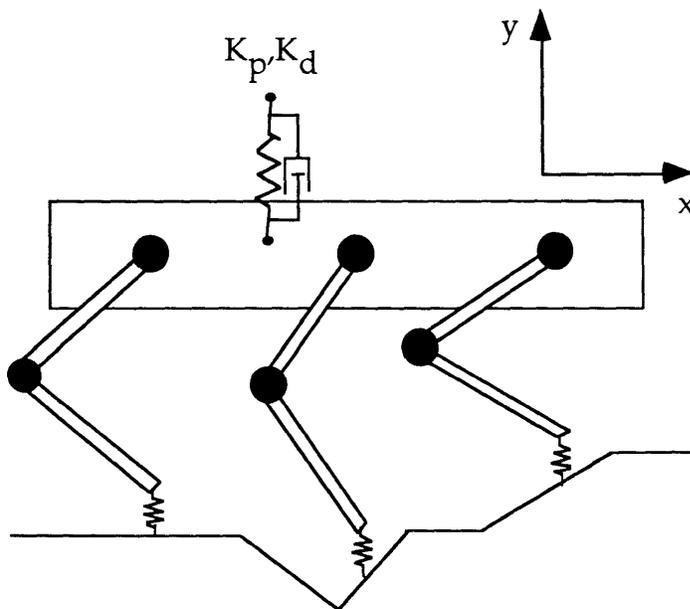


Figure 4.1. Walking Robot on Uneven Terrain.

Another problem which required the use of a cartesian controller was grabbing a peg. As seen in Figure 4.3, the hook was chamfered to allow it to slide on the peg. However, this required that the stiffness of the leg be soft perpendicular to the hook motion to allow the hook to slide onto the peg. A joint PD controller in which the stiffness was set in joint space would not always allow the hook to slide onto the peg.

Additionally, joint backlash and uncertainty in the exact peg location caused conventional joint PD control schemes to lose contact with the pegs,

or to push too hard on other pegs, which caused LIBRA to fall. The cartesian based controller ensured that the hooks always pushed against the pegs evenly, regardless of their position.

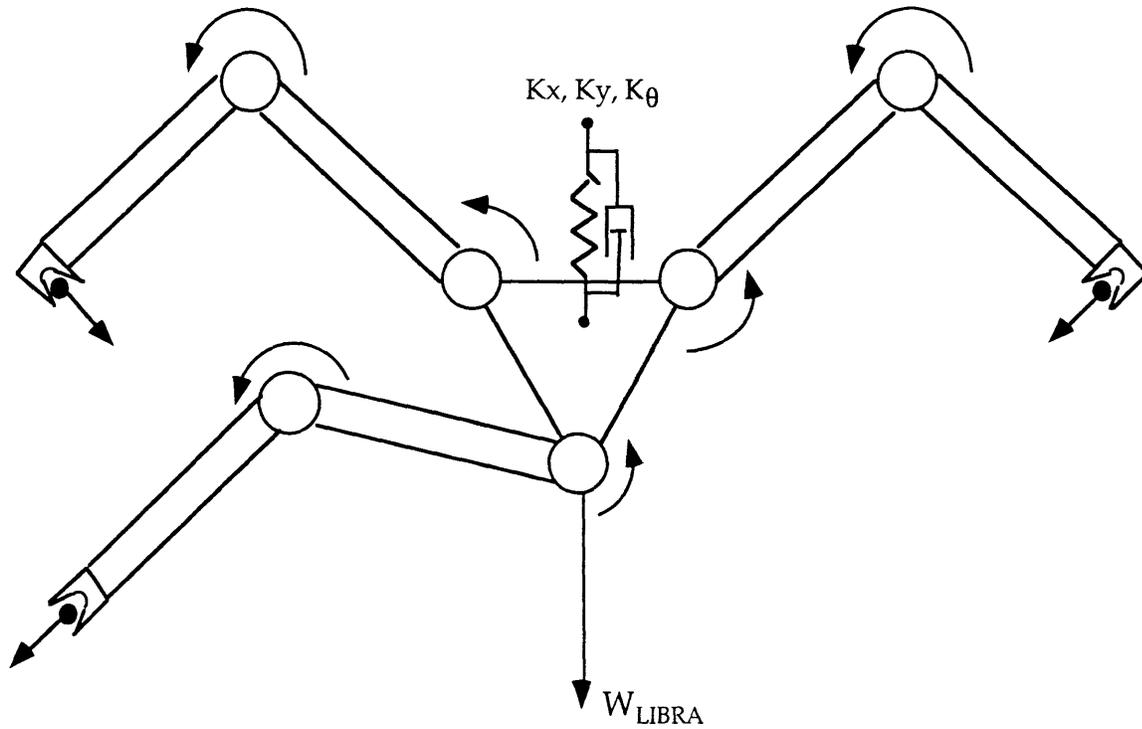


Figure 4.2. Control of LIBRA

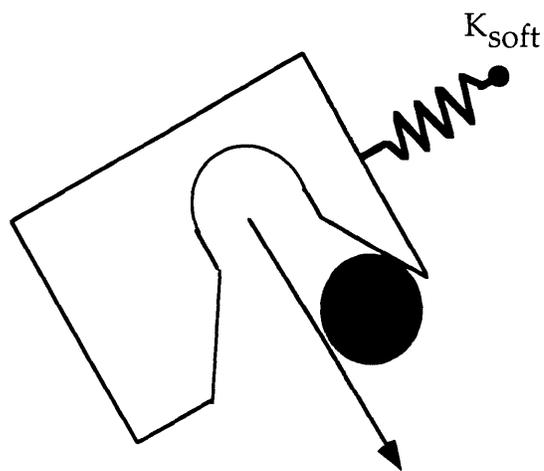


Figure 4.3. Grabbing A Peg

4.3 Experimental Setup

4.3.1 Overall Setup

The overall experimental setup including the peg board, LIBRA, power sources and computing, is shown in Figure 4.5. Seven encoder signals, one for each of the six joints and one for the body angle, are fed to the computer. The 200 Hz control cycle updates commands to the power amplifiers which drive LIBRA.

4.3.2 LIBRA

As described in Chapter 1, LIBRA is a 3 limbed, planar climbing robot. LIBRA weighs approximately 40 Newtons, and each limb has a 31 cm reach. Specifications of the design of the system can be found in (Argaez, 1991). LIBRA was built in order to study the planning and control of multi-limbed robots. In this work, each limb of LIBRA was fitted with hooks shown in Figure 4.4 to allow LIBRA to climb on pegs. The hook used on the third leg was designed to allow it to hold onto the peg on either side of the body, as the required by the climbing gait.

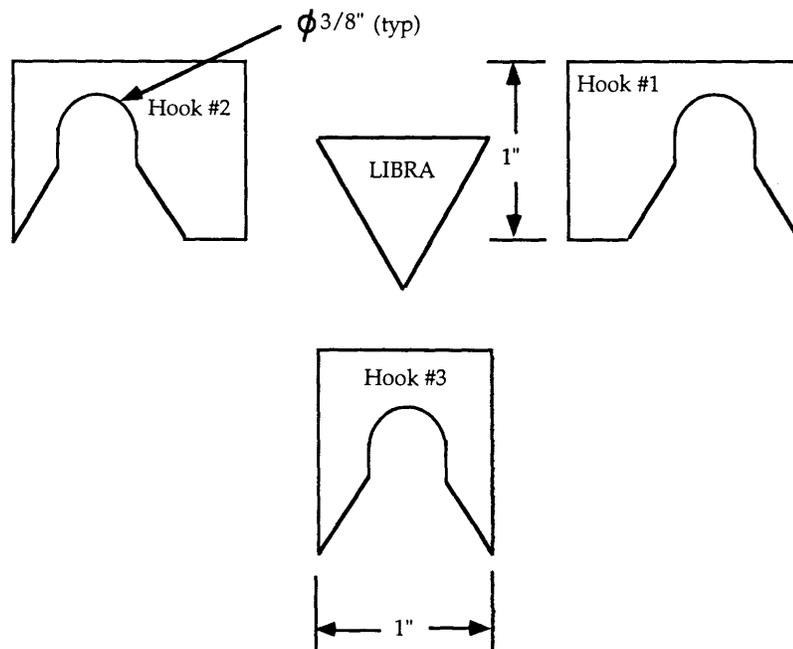


Figure 4.4 LIBRA's Hooks

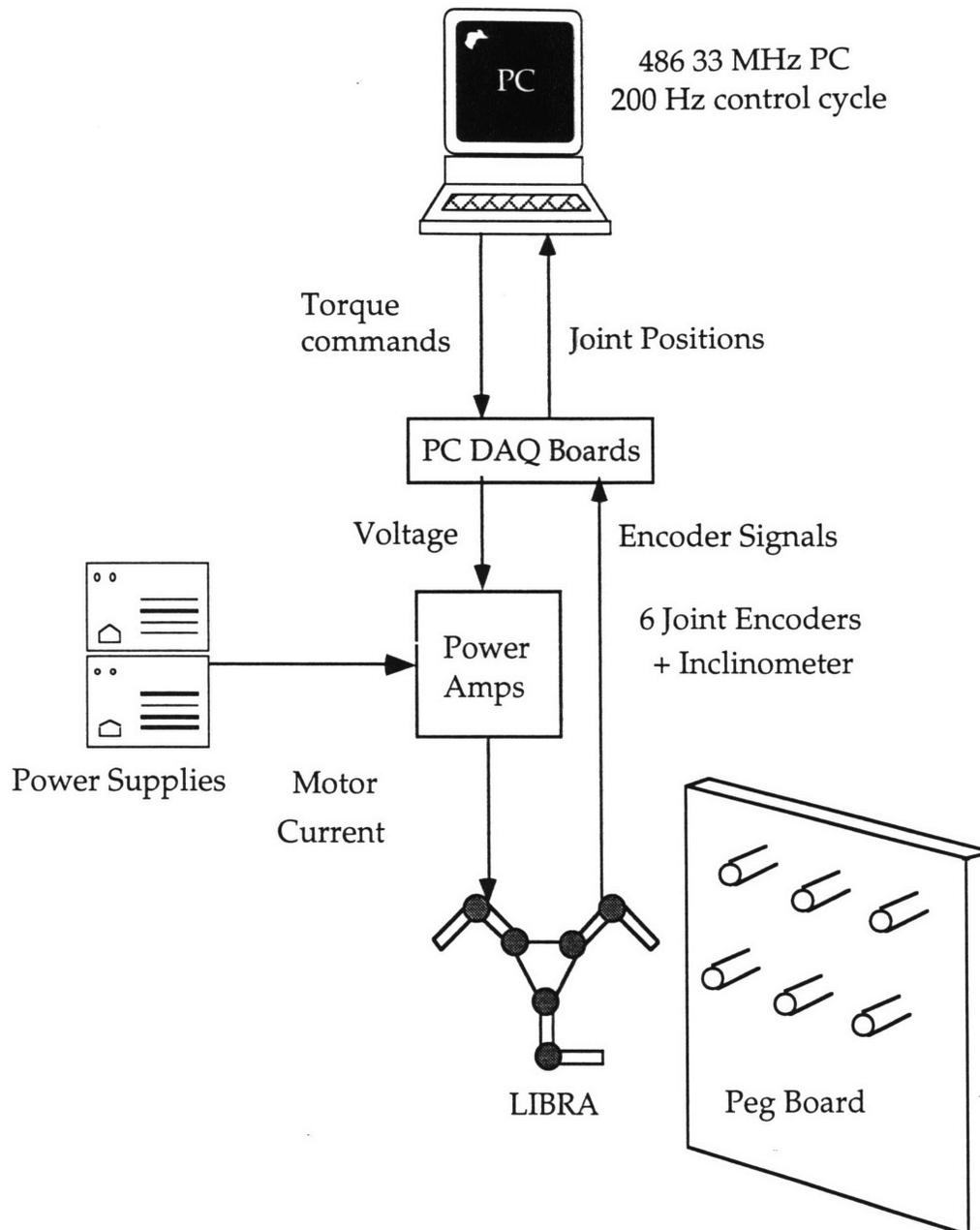


Figure 4.5 Experimental Setup

4.3.3 LIBRA's Peg Board

The peg board was built using 8'x4'x1/4" thick plywood on the front and back face. A sheet of formica was used to cover the front face. The two sheets of plywood were placed two inches apart from each other with 1'x2'x8' pine on each side. Insulation foam filled the inside of the structure for added stiffness between the two sheets of plywood. Two 8'x4' walls were joined to

give an overall wall dimension of 8'x7.5' (half a foot was cut off the top to fit in the room) shown in Figure 4.6. A 3/8"x12" hex bolt was used for pegs. A small groove was cut for an e-clip to prevent the bolt from sliding through the front side of the board. A nut was then used to tighten the back end of the bolt to the wall.

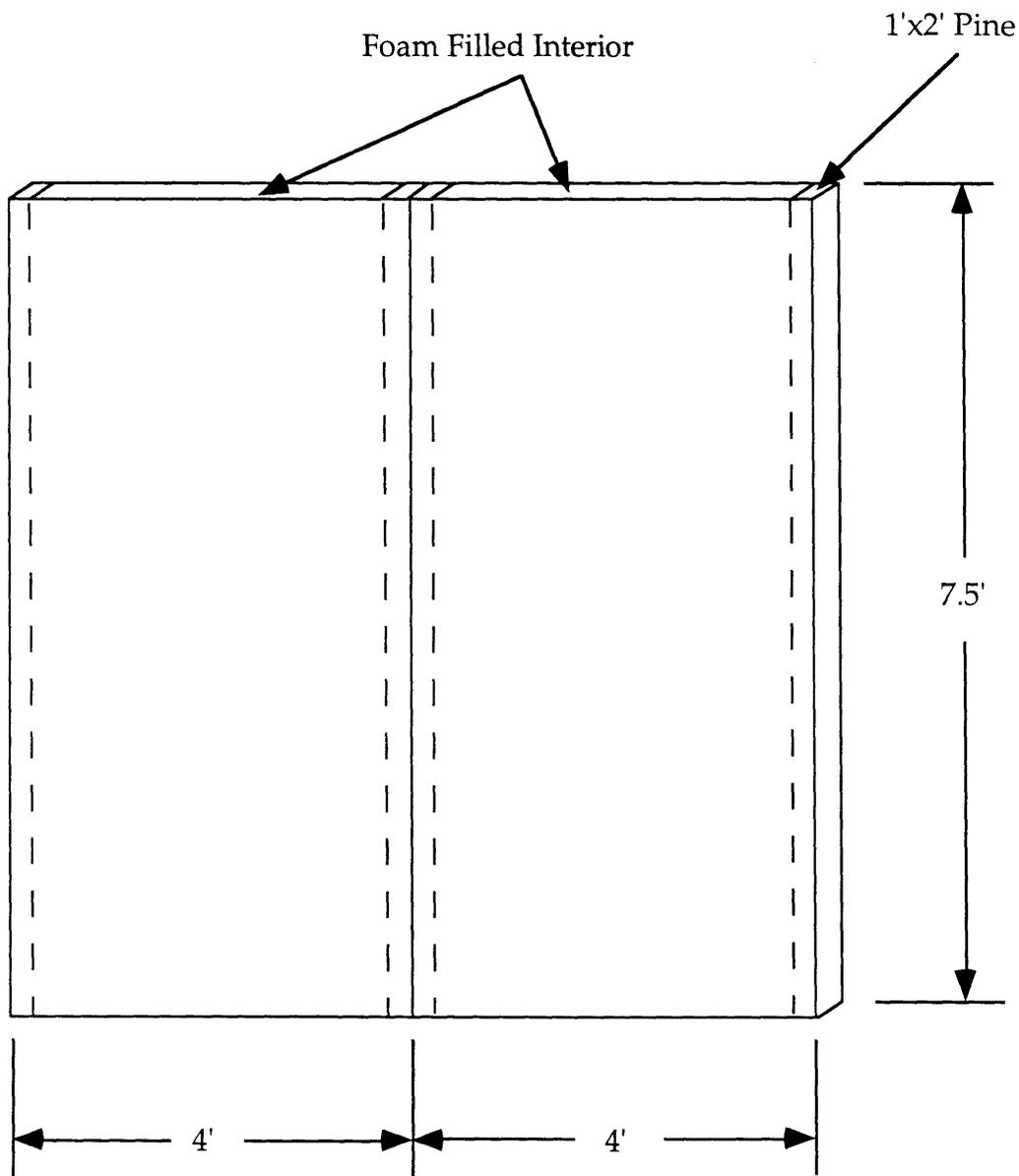


Figure 4.6. LIBRA's Peg Board

4.3.4 Power Amplifiers

Power amplifiers for the LIBRA motors were built by (Sunada, 1992). However the gain of the amplifiers found in (Sunada, 1992) did not seem correct, so a simple test was done to determine them. One motor was placed at its joint limit, in order to ensure that the motor was stalled, and increased voltage signals sent to the amplifiers. The current to the motor was calculated by measuring the voltage across the motor while the motor was stalled. The amplifier gain was found to be 0.2678 amps/volt.

4.4 Cartesian Control of LIBRA Applied to the Climbing Task

4.4.1 The Control Scheme

The control law that was used for the control of one limb of LIBRA, developed in Chapter 3, is shown again in Equation 4.1.

$$\tau = K'_{leg} J_{leg}^{-1} \ddot{X}_d^{leg} + J_{leg}^T F_{ext}^{leg} \quad (4.1)$$

The desired leg accelerations and forces for the free limb are shown in Equation 4.2 and 4.3, resulting in the SCCT controller used in Chapter 3.

$$\ddot{X}_d^{leg} = K_p^{leg} (X_{des}^{leg} - X_{act}^{leg}) - K_d^{leg} \dot{X}_{act}^{leg} \quad (4.2)$$

$$F_{ext}^{leg} = 0 \quad (4.3)$$

The desired endpoint accelerations and forces of the two limbs used to control the desired body accelerations $(\ddot{x}_d^{body}, \ddot{y}_d^{body}, \ddot{\theta}_d^{body})$ are found through the transformation matrix T by:

$$\ddot{X}_d^{leg} = T \ddot{X}_d^{body} \quad (4.4)$$

$$F_{ext}^{leg} = T F_{ext}^{body} \quad (4.5)$$

where:

$$\ddot{X}_d^{body} = K_p^{body} (X_{des}^{body} - X_{act}^{body}) - K_d^{body} \dot{X}_{act}^{body} \quad (4.6)$$

K'_{leg} is the same diagonal matrix of joint gear ratios discussed in Chapter 3 and shown again in Equation 4.7.

$$K'_{leg} = \begin{bmatrix} \frac{N_1^2}{N_2^2} & 0 \\ 0 & \frac{N_1^2}{N_2^2} \end{bmatrix} \quad (4.7)$$

The joint gear ratios N_1 and N_2 of each limb of LIBRA, are listed in Table 3.1. The block diagram for the control of the body of LIBRA is shown in Figure 4.7.

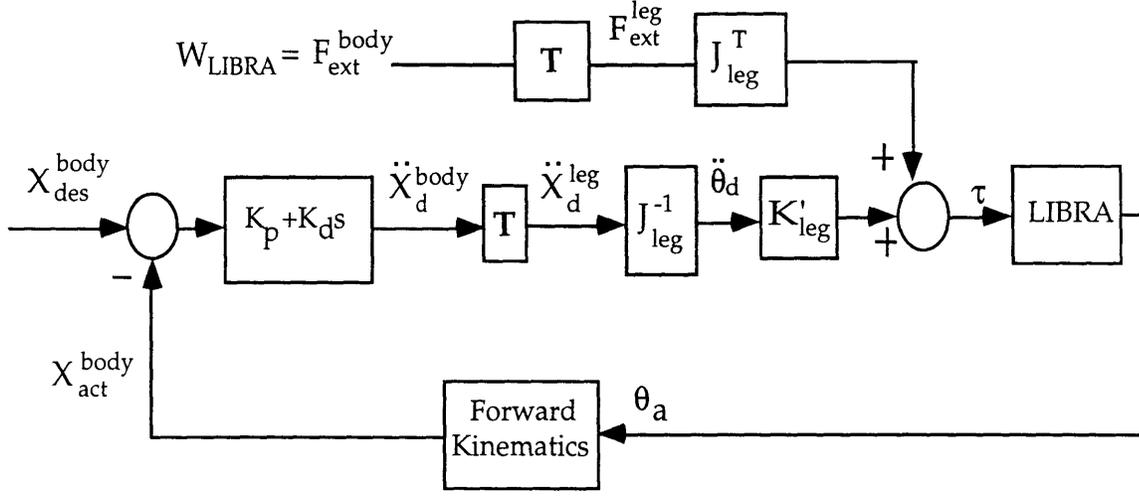


Figure 4.7. Block Diagram for the Control of LIBRA.

The transformation matrix T is used to transform the desired body accelerations and forces to desired leg accelerations and forces as in Figure 4.8, where D is a dummy variable to represent either forces or accelerations. Leg accelerations are limited in order to ensure that the legs did not pull away from the hooks at any time (see discussion in Section 4.4.2).

The following equations are used to transform the desired body accelerations and forces to desired accelerations and forces of the endpoint of the two legs controlling the body of LIBRA:

$$D_x^{leg1} = -\frac{1}{2} D_x^{Body} \quad (4.8)$$

$$D_x^{leg2} = -\frac{1}{2} D_x^{Body} \quad (4.9)$$

$$D_y^{leg1} + D_y^{leg2} = D_y^{Body} \quad (4.10)$$

$$D_y^{leg1} r_{x1} + D_y^{leg2} r_{x2} = D_\theta^{Body} \quad (4.11)$$

Solving the above equations simultaneously results in the transformation matrix from the desired body accelerations and forces to desired leg accelerations and forces shown in Equations 4.4 and 4.5.

$$T = \begin{bmatrix} -\frac{1}{2} & 0 & 0 \\ 0 & \frac{r_{x1}}{r_{x1} - r_{x2}} & \frac{-1}{r_{x1} - r_{x2}} \\ -\frac{1}{2} & 0 & 0 \\ 0 & \frac{-r_{x2}}{r_{x1} - r_{x2}} & \frac{1}{r_{x1} - r_{x2}} \end{bmatrix} \quad (4.12)$$

Additionally, leg accelerations and forces must be transformed to the port coordinate frame by Equation 2.18, as discussed in Section 2.3.2, before multiplying them by the leg Jacobian transpose (J_{leg}^T) and leg Jacobian inverse (J_{leg}^{-1}) matrices (found in Appendix A).

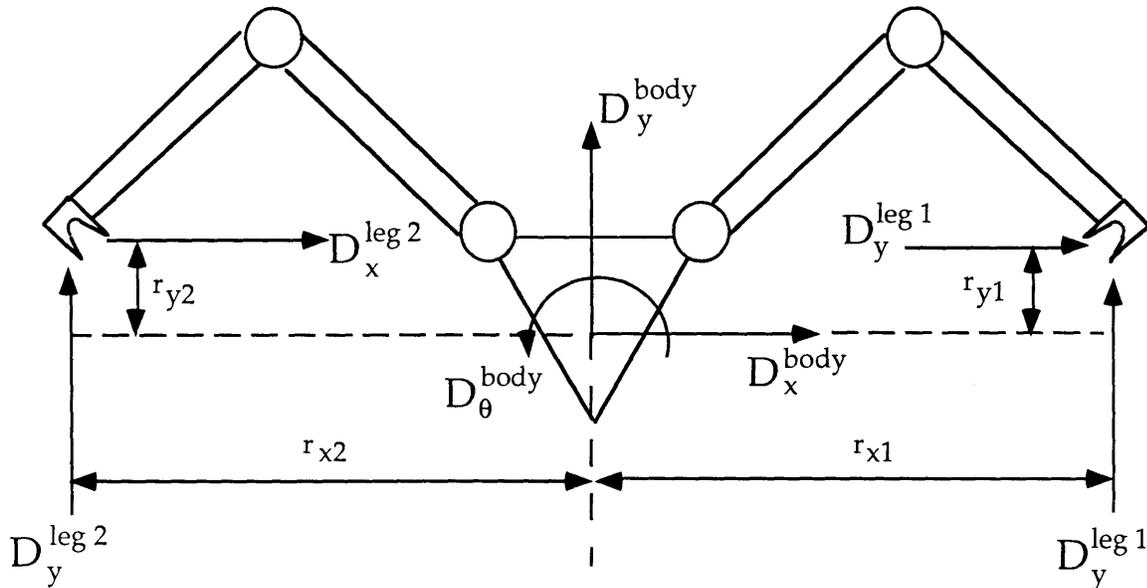


Figure 4.8. Body Accelerations and Forces on LIBRA

4.4.2 Limitation of Endpoint Accelerations

Once the desired accelerations of the endpoint are calculated, they are transformed from cartesian space to the “hook” space of the leg shown in Figure 4.9. Equation 4.13 represents the transformation from cartesian space to the “hook” space.

$$\ddot{X}_{hook}^{leg} = T_{hook} \ddot{X}_d^{leg} \quad (4.13)$$

The transformation is implemented by Equation 4.14.

$$\begin{bmatrix} \ddot{g} \\ \ddot{h} \end{bmatrix} = T_{hook} \begin{bmatrix} \ddot{x}_d^{leg} \\ \ddot{y}_d^{leg} \end{bmatrix} \quad (4.14)$$

where:

$$T_{hook} = \begin{bmatrix} \cos(\theta_{grab}) & \sin(\theta_{grab}) \\ -\sin(\theta_{grab}) & \cos(\theta_{grab}) \end{bmatrix} \quad (4.15)$$

The angle of the hook (θ_{grab}) is the orientation of the hook in the fixed frame, which is a function of the body and joint angles, shown in Equation 4.16.

$$\theta_{grab} = \theta_{body} + \theta_1 + \theta_2 + \theta_{hook} \quad (4.16)$$

where:

θ_{hook} = kinematic hook parameter found in Appendix A.

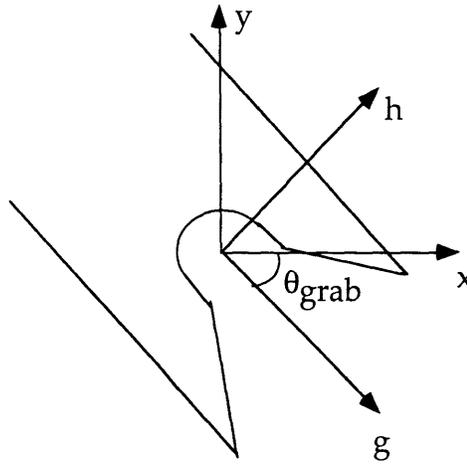


Figure 4.9. Hook Coordinate Frame (g-h)

After the desired endpoint accelerations are transformed to the “hook” frame they are limited to ensure that the leg never pulls away from the peg. A desired cartesian acceleration, which ensures that the acceleration at the hook is in the grab direction (+g direction in Figure 4.10), is used at all times. In some cases a bias acceleration is added to ensure that the hooks pull toward the pegs with some minimum offset acceleration such that:

$$\ddot{g} \geq \alpha$$

where:

α = some offset greater than or equal to zero.

Once the hook accelerations are limited to ensure they do not pull away from the peg, they are transformed back to the cartesian accelerations by:

$$\ddot{X}_d^{leg} = T_{hook}^{-1} \ddot{X}_{hook}^{leg} \quad (4.17)$$

The transformation is implemented by:

$$\begin{bmatrix} \ddot{x}_d^{leg} \\ \ddot{y}_d^{leg} \end{bmatrix} = T_{hook}^{-1} \begin{bmatrix} \ddot{g} \\ \ddot{h} \end{bmatrix} \quad (4.18)$$

where:

$$T_{hook}^{-1} = \begin{bmatrix} \cos(\theta_{grab}) & -\sin(\theta_{grab}) \\ \sin(\theta_{grab}) & \cos(\theta_{grab}) \end{bmatrix} \quad (4.19)$$

4.4.3 Calculation of Body and Leg Positions

The position of the endpoint of each leg in its port frame (x_p, y_p) , shown in Figure 4.10, is found through the forward kinematic equations shown in Appendix A. Once the endpoint positions are determined in each port frame, their positions can be found in the body frame (x_B, y_B) through the forward transformations shown below.

$$X_B = T_{port} X_p \quad (4.20)$$

where:

X_B = position of the leg in the body frame (x_B, y_B)

The transformation is implemented by Equation 4.21 to find the position of the leg in the body frame (x_B, y_B) .

$$\begin{bmatrix} x_B \\ y_B \\ 1 \end{bmatrix} = T_{port} \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} \quad (4.21)$$

where:

$$T_{port} = \begin{bmatrix} \cos(\theta_{port}) & -\sin(\theta_{port}) & x_{port} \\ \sin(\theta_{port}) & \cos(\theta_{port}) & y_{port} \\ 0 & 0 & 1 \end{bmatrix} \quad (4.22)$$

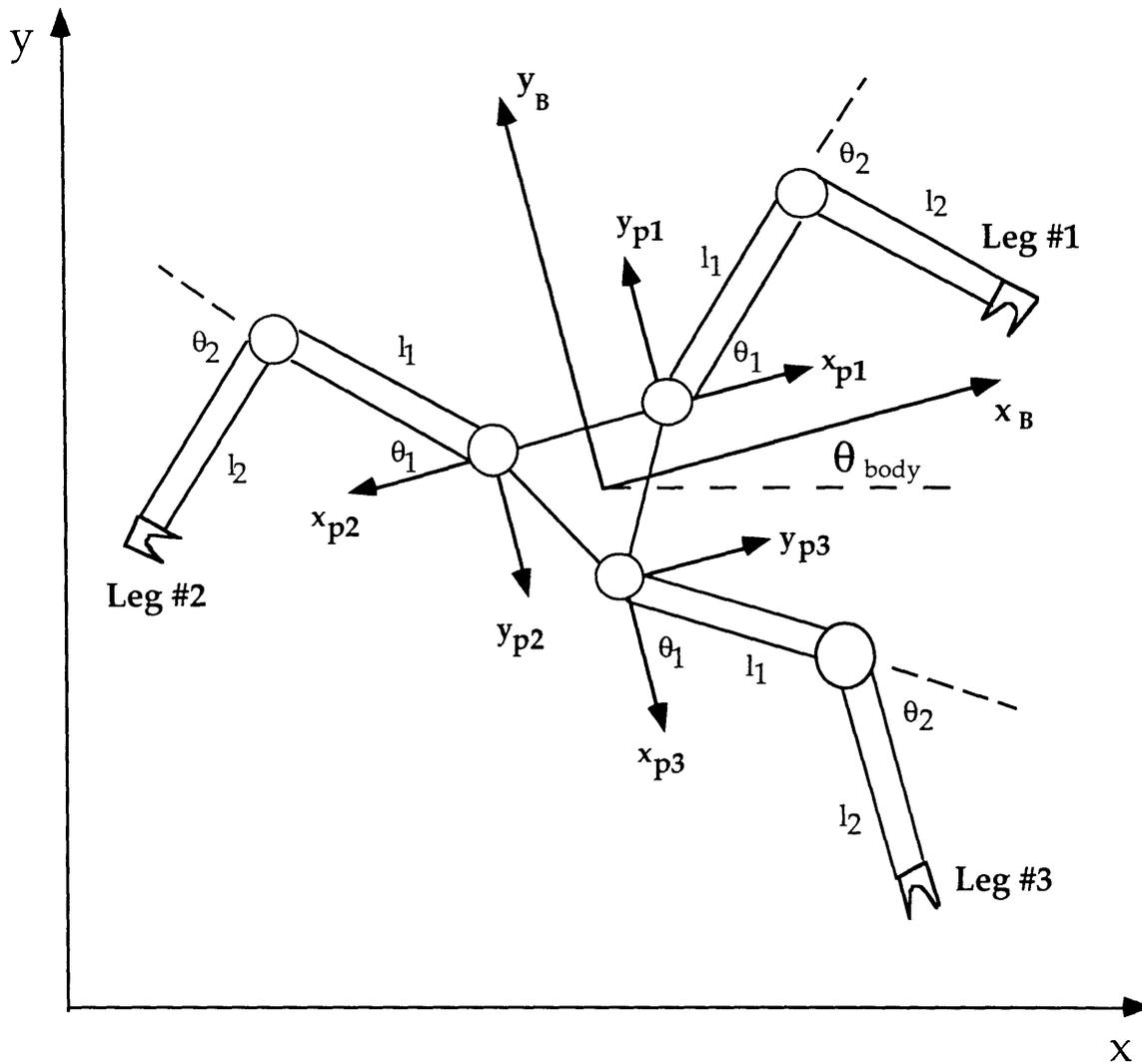


Figure 4.10. Coordinate Frames for LIBRA

The parameters θ_{port} , x_{port} , and y_{port} are Denavit-Hartenberg parameters which can be found in Appendix A for each limb.

Once the position of all three legs in body frame are known, two of the leg positions are used to calculate the body position. This is done by knowing the position (in the fixed frame) of pegs which the two legs are holding. The body frame (x_B, y_B) is moved to the fixed frame (x, y) position such that the position of the first leg is placed at the position of the first peg, with an initial body angle of zero, as shown in Figure 4.11.

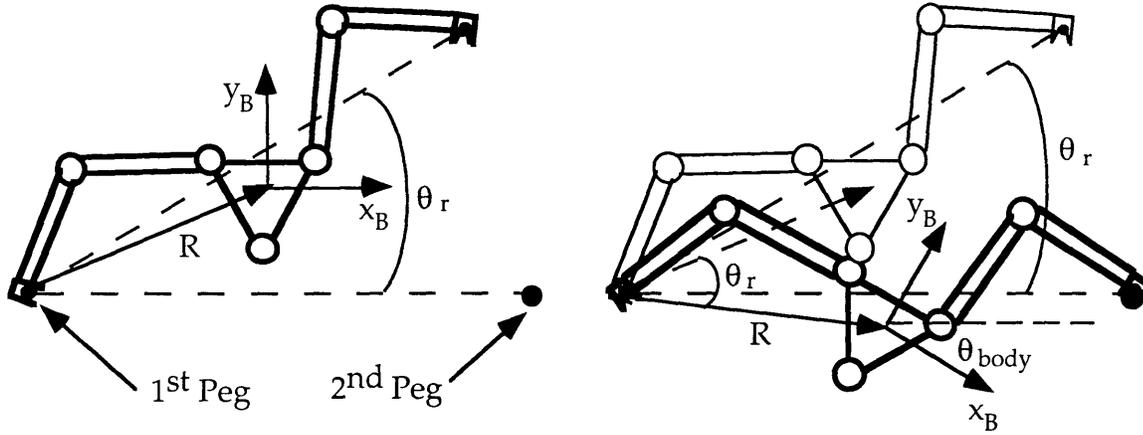


Figure 4.11. Calculation of Body Position

The body frame is then rotated by θ_r about the first peg until the second leg is at the y position of the second peg as shown in Figure 4.11. The x & y positions, as well as the angle of the body (θ_{body}), can then be determined from the amount the initial body frame is rotated about the first peg. Therefore the x-y position of the body is determined with respect to one of the peg positions, and the leg holding that particular peg, while the body angle is determined by the second leg. Once the position of the body is determined, the position of each leg in the fixed frame (x, y) is determined by the forward transformations shown below:

$$X = T_{body} X_B \quad (4.23)$$

The transformation is implemented by Equation 4.24 to find the position of the leg in the fixed frame (x,y).

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = T_{body} \begin{bmatrix} x_B \\ y_B \\ 1 \end{bmatrix} \quad (4.24)$$

where:

$$T_{body} = \begin{bmatrix} \cos(\theta_{body}) & -\sin(\theta_{body}) & x_{body} \\ \sin(\theta_{body}) & \cos(\theta_{body}) & y_{body} \\ 0 & 0 & 1 \end{bmatrix} \quad (4.25)$$

The parameters x_{body} , y_{body} , and θ_{body} are the position and angle of the body in the fixed frame.

4.4.4 Gain selection

Gains for the body are experimentally determined. The gain K_{py}^{body} is made soft to allow LIBRA to “sit down” on the pegs. However the gain K_{px}^{body} is also made fairly soft due to a non-collocation control problem. The measurement of the position of LIBRA is based on one leg as discussed above. However since each leg can only pull, a second leg is used to control the $+x$ position of LIBRA while the first leg controls the $-x$ direction as seen in Figure 4.12. Because the $+x$ position is measured from the first leg, but controlled from the second leg, there is a classical non-collocation between the sensor and controller. The dynamics of the system between the measuring leg and the controlling leg, including joint backlash and play between the hooks and pegs, limit the controllable stiffness of LIBRA in the x direction. Final chosen gains $(K_{px}^{body}, K_{py}^{body}, K_{p\theta}^{body}, K_{dx}^{body}, K_{dy}^{body}, K_{d\theta}^{body}, K_{px}^{leg}, K_{py}^{leg}, K_{dx}^{leg}, K_{dy}^{leg})$ for climbing on the pegs are shown in Table 4.1.

Table 4.1. Controller Gains for Peg Climbing

	Body x	Body y	Body θ	Free Leg x	Free Leg y
K_p	100	50	10	500	500
K_d	5	2.5	0.5	25	25

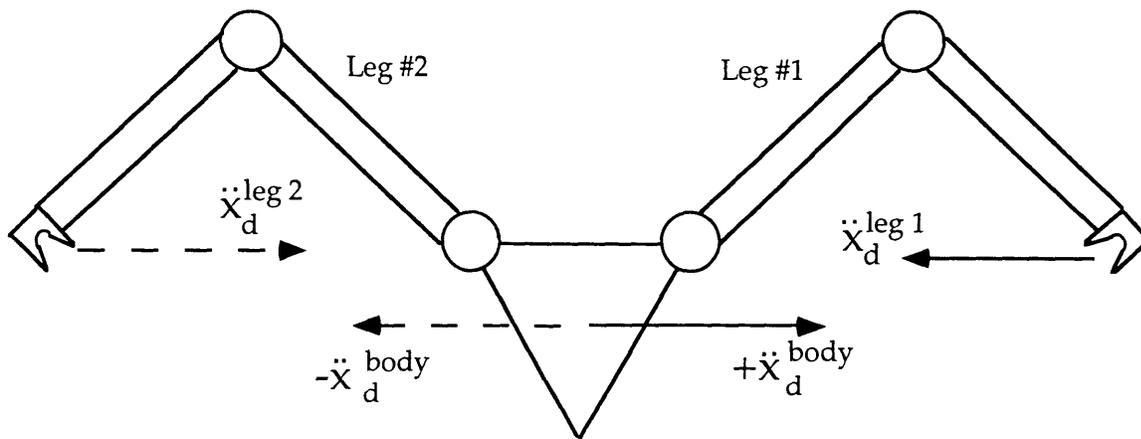


Figure 4.12. Pull of Legs for Control of LIBRA

4.4.5 Gravity Compensation

The second leg is used to control the body position by pulling on the peg, as discussed above, but its position is not specifically controlled. This created the necessity for gravity compensation by feeding forward the weight of LIBRA (which acts as a large disturbance to the uncontrolled leg) to each leg.

Alternatively, position control of the second leg could have been implemented by commanding a position inside the peg, with knowledge of environmental stiffness and peg locations. This type of approach, shown in Figure 4.13 was used previously for LIBRA, as well as a mobility analysis for the selection of the control variables, in (Sunada, 1992). This method would eliminate the need to feed the weight of LIBRA to the legs (as the controller would reject this disturbance). However, this requires good knowledge of the environment, such as the location of the pegs in our experiment, in order to be able to set the commanded position inside the pegs. Additionally, the method assumes that the body can be controlled independently with the inertial coordinate frame leg, which requires this leg to be able to push and pull on the peg. Since it was desired to limit the forces of the leg to always push towards the hook, this option was not used.

The limitation of not being able to push and pull in all directions (as if the legs were able to actively grab each peg) is the major limitation in the accuracy of the controller. High position accuracy is obtainable, however at the increased risk of falling off of the pegs. Therefore the positioning accuracy is sacrificed in order to ensure that LIBRA would not fall off of the pegs.

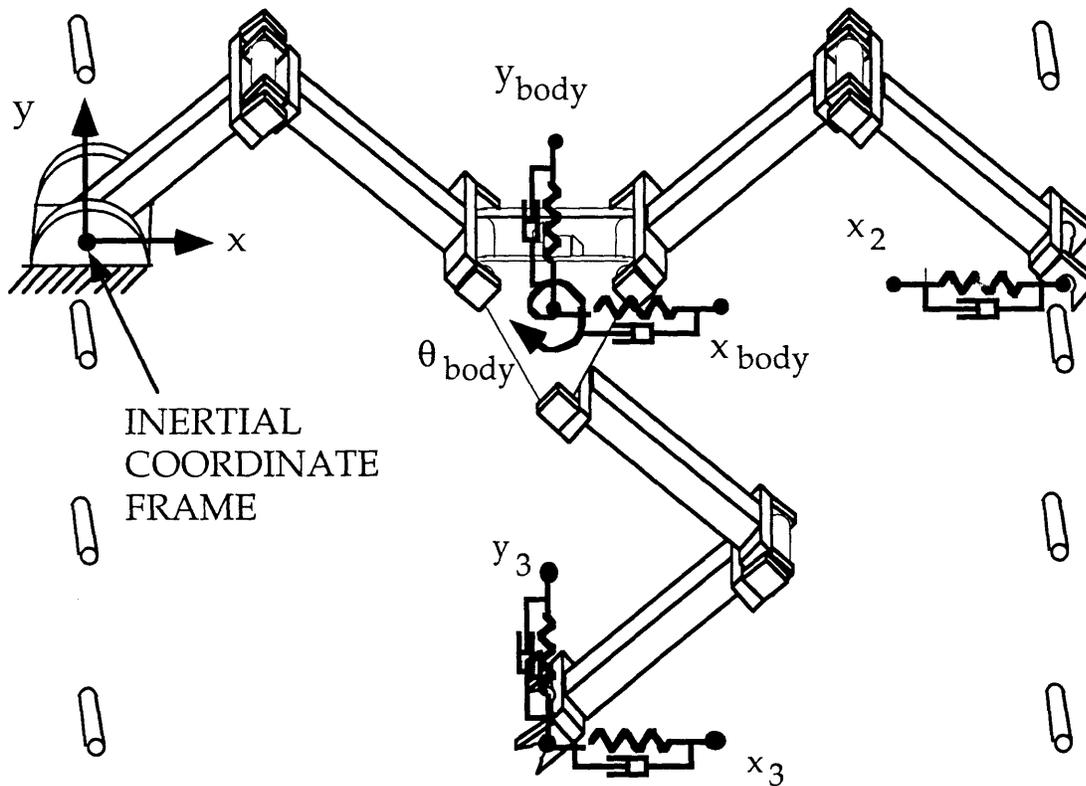


Figure 4.13. Previous Control Approach of LIBRA

4.4.6 Executing Plans

A small amount of local planning is necessary to execute each action module, in the action plan. Path trajectories are created for both the legs and the body between discrete plan positions. A simple virtual potential (Khatib, 1986) is placed around the body to keep the limbs a safe distance from the body. It is also necessary to switch kinematic configuration when leg 3 switches sides of the body, to ensure that the hook is on top of the peg. This configuration switch is done in an open loop fashion between grabbing pegs. Additionally, if a leg is asked to release a peg, the leg was given an updated command path to follow the body which followed the body moved upwards. The control stiffness K_{px}^{leg} of the released leg is made soft to allow the leg to be

pushed away from a peg if contact between the third leg and peg is encountered while the body is in motion.

Once a plan for LIBRA to grab a peg is encountered, a series of sequences shown in Figure 4.14 are executed to grab the new peg. First, the leg is given a commanded acceleration to move away from the current peg in an open loop fashion. A path to the next peg is then generated. Once the leg moves to within 3 mm of the peg a new path around the peg is generated. When the opening of the hook is is aligned with the peg, the hook is given a commanded acceleration toward the peg until the hook is stopped. Alternatively, a type of virtual potential for the execution of grabbing the next peg was investigated, but eventually abandoned, because of the ease of following a desired path around the known position of each peg.

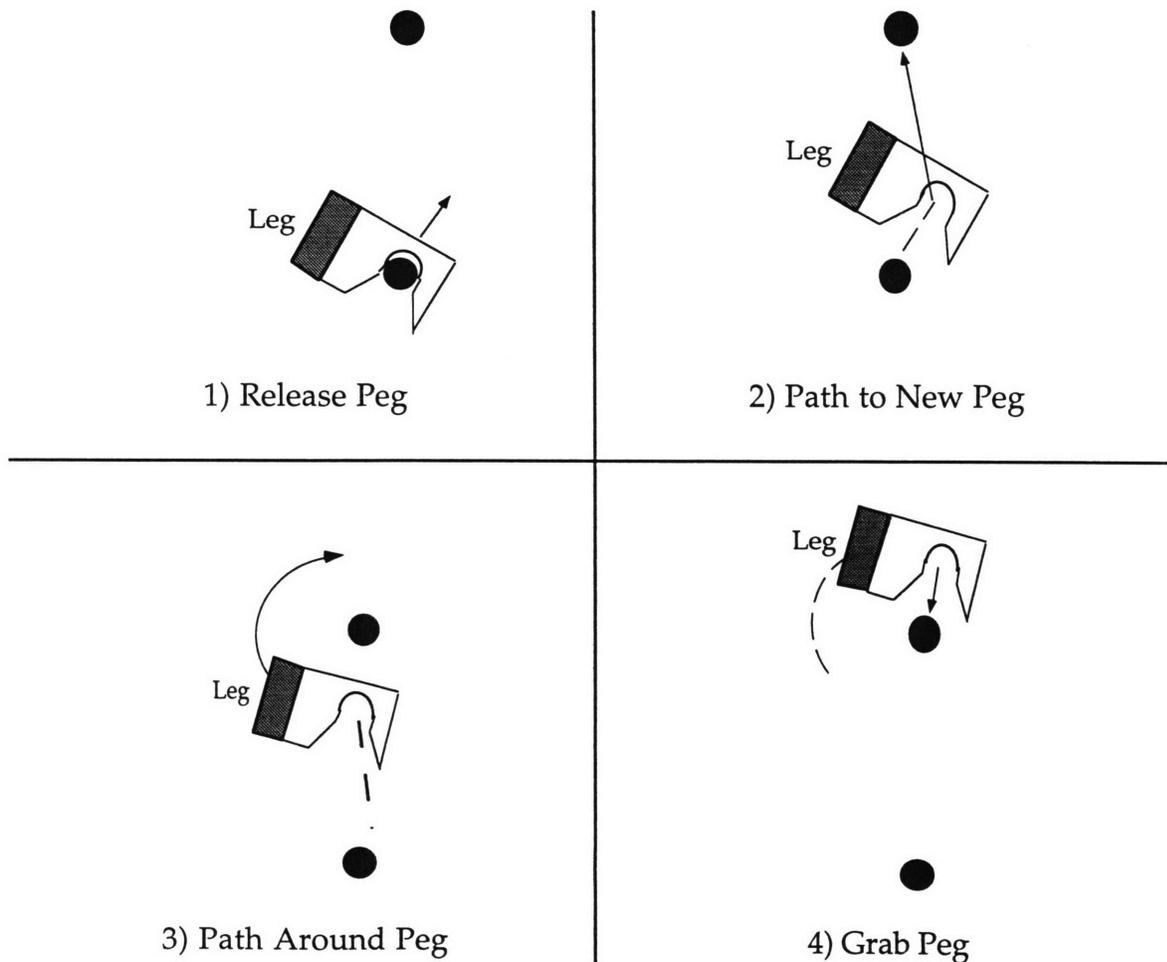


Figure 4.14. Sequence of Steps to Grab a New Peg

4.5 Experimental Results

Figure 4.15 shows the nominal gait of 191, 392, 291, 300, 3, developed by the action module planning method in Section 2.5.1. The switching of the kinematic configuration and the released leg following the body can be seen in stages two and four, respectively, as discussed previously in Section 4.4.6.

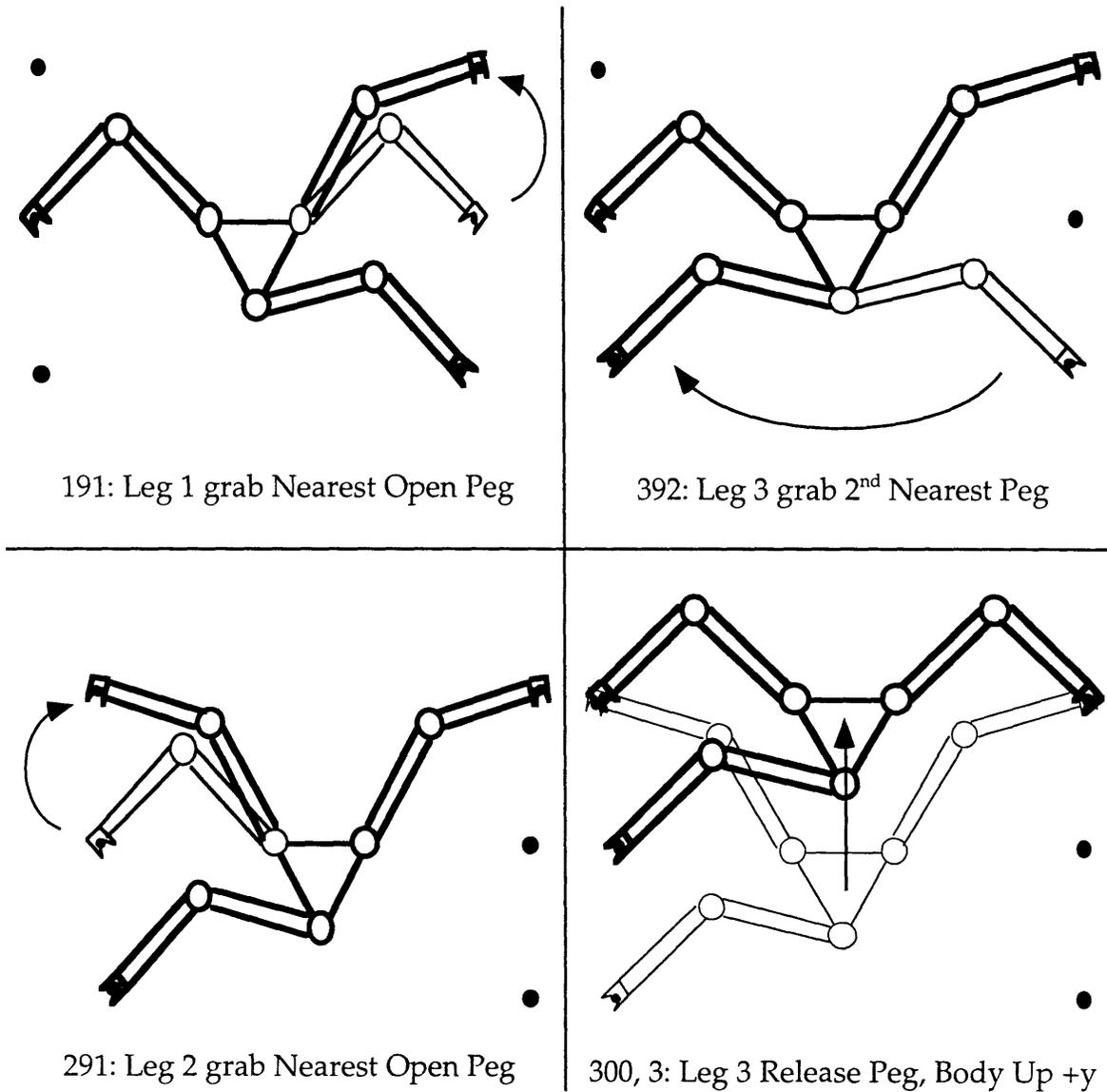


Figure 4.15. Climbing Gait for the LIBRA.

Figure 4.16 shows experimental data for LIBRA climbing two consecutive steps of the ladder task (discussed in Section 2.4 and 2.5.1). Peg positions can be found in Appendix C. Numbers illustrate the order of the actions on the plot. The grab peg sequence discussed in Section 4.4.6 can be seen in the experimental data. Errors can be seen at the “body hold” position as the legs pull way from hooks and grab new hooks. This is due to the controlled compliance of the body to ensure that the hooks never leave the pegs. The body positioning of LIBRA was limited by the inability to actively grab the peg in all directions as discussed in Sections 4.4.3 and 4.4.4.

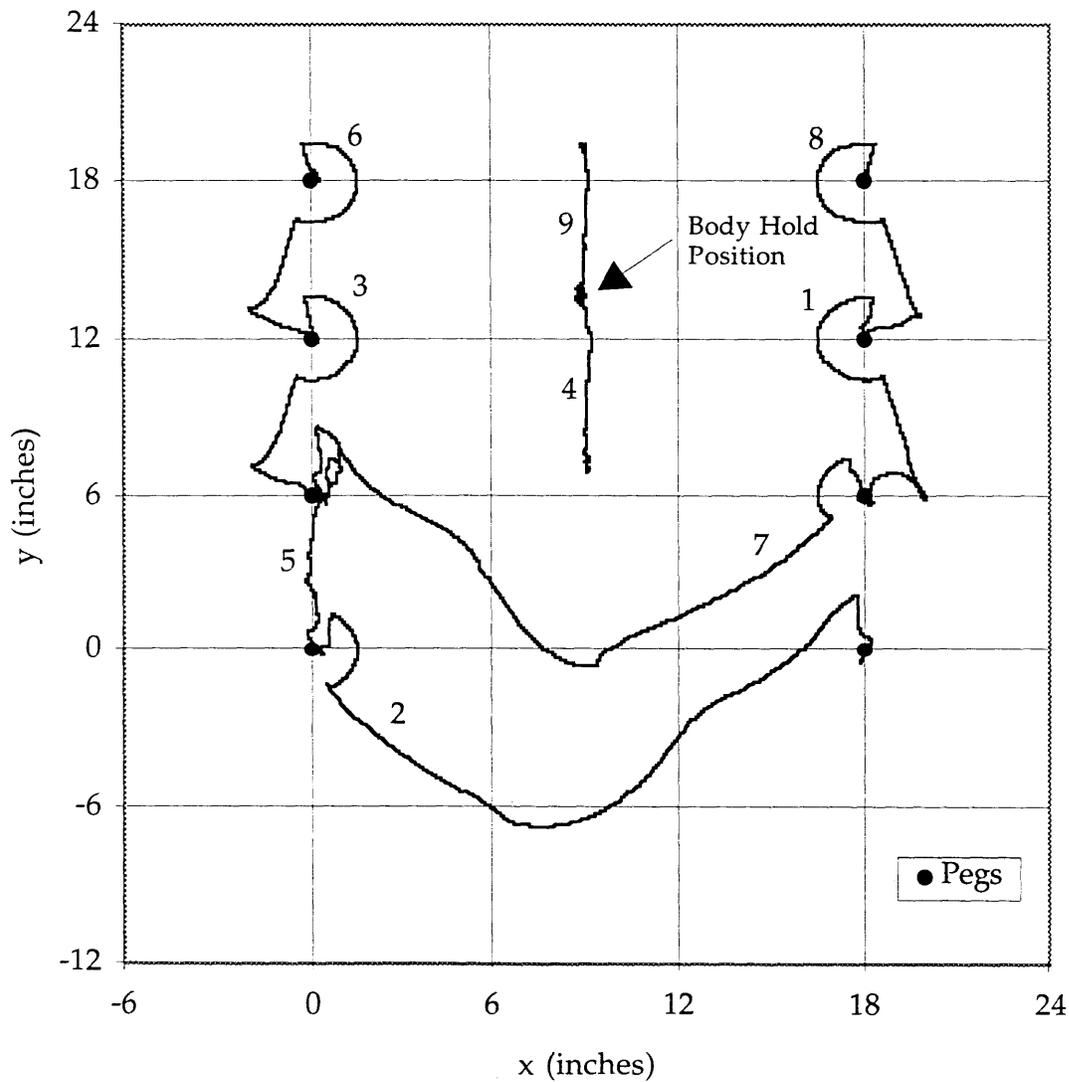


Figure 4.16. Two Steps of LIBRA’s Climb.

Figure 4.17 shows an experiment where the limitation of not being able to push and pull in all directions is removed (by holding two legs of LIBRA fixed) to show that high precision accuracy is attainable with LIBRA. In the experiment, Jacobian transpose control and the SCCT controller are compared by having the body of LIBRA track a circle 5 cm in radius. The SCCT controller tracks the circle almost perfectly. However, the Jacobian transpose controller exhibits tracking error for fast tracking (about 6 seconds per circle) and even at slow tracking (about 12.5 seconds per circle), similar to the results of the control of one limb in Chapter 3.

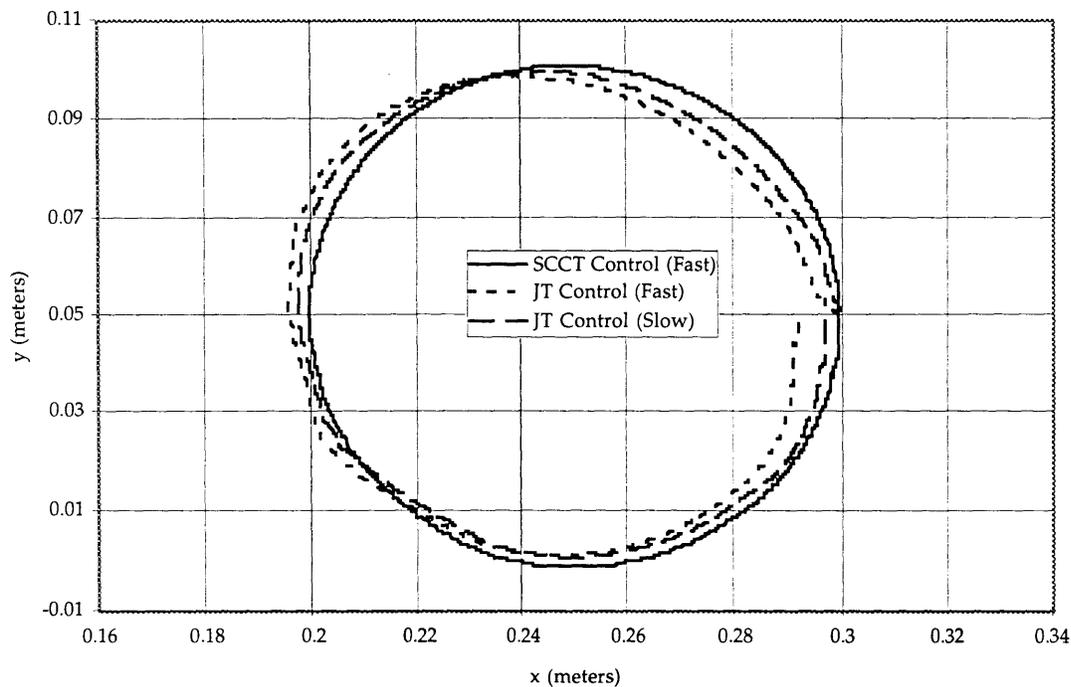


Figure 4.17. Body Path Tracking

Additionally, Figure 4.18, shows a comparison of the two controllers performing the path tracking of a body up (action module 003) motion on the pegs, with the included constraint that the legs can not pull away from the pegs. The path is generated to complete the 7.5 cm (3") step in 0.5 seconds. Again as seen by the figure, the SCCT controller is able to maintain the desired x position of 22.8 cm (9"), while following the commanded y path, slightly better than the Jacobian transpose controller.

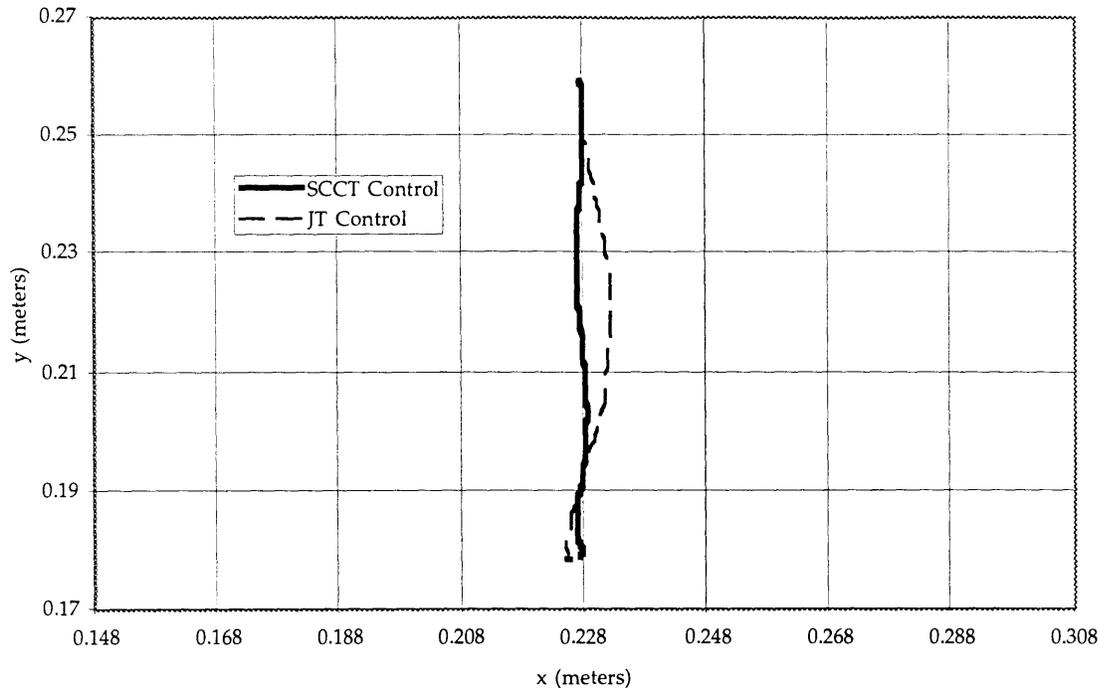


Figure 4.18 Body Push-up

Figure 4.19 shows an experiment of the compliance of the body in the x-direction, while maintaining hook contact in the presence of an external disturbance. In this experiment, LIBRA is placed on the pegs (in a configuration similar to Figure 4.12), and a disturbance is artificially applied to the body to emulate the third leg making contact with a peg. The experiment demonstrates the ability to make the body stiffness soft in one direction and stiff the other cartesian directions. The body was able to be “pushed” 10 times as far in the x-directions because the gain K_{py}^{body} was set ten times as stiff as the gain K_{px}^{body} .

As seen in Figure 4.19, the position of leg #2 moved inward about 3 cm, when the cartesian controller is initiated, from its initialized position. Due to uncertainties in the peg position, the second leg is commanded to simply move toward the peg as discussed in Section 4.4.5. Leg #1 remains at exactly the peg position (0.457 m, 0.152 m) (18”,6”) because the body is measured in reference to that peg as discussed in Section 4.4.3. As the body is pushed, both

legs continue to maintain contact with the peg by constantly pulling the hook toward the peg. This ensures that the legs never fall of the pegs even when and unknown external force is exerted on the body.

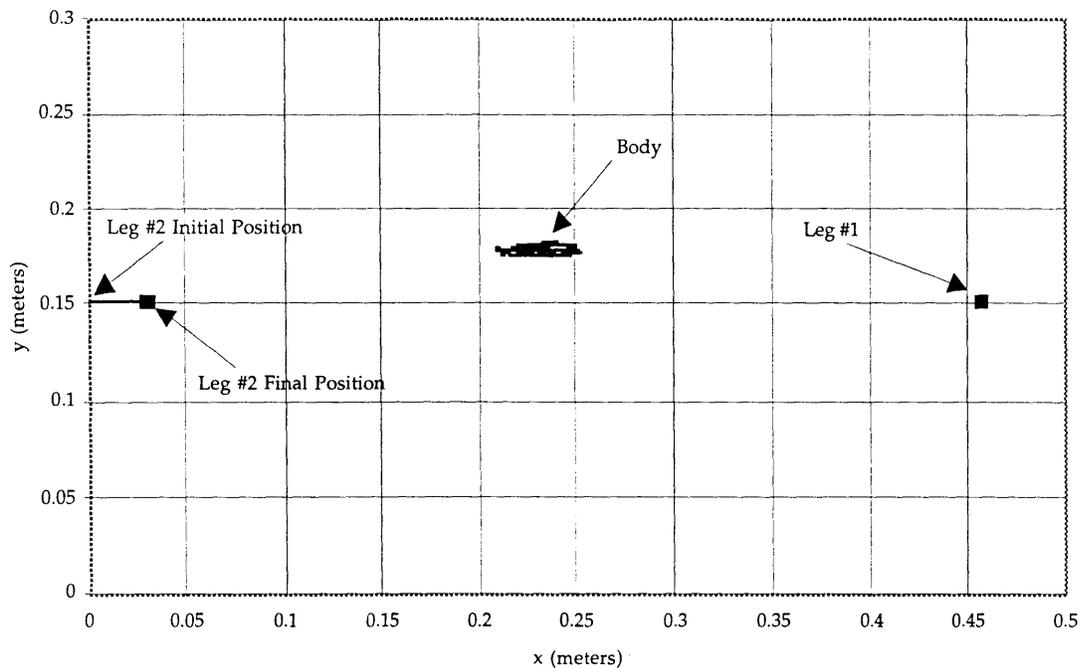


Figure 4.19. Body Compliance

4.6 Summary and Conclusions

This chapter presented the application of the action module planning methodology and SCCT controller to the experimental climbing robot LIBRA. Details of the experimental system were outlined, as well the use of the control scheme (applied to one limb of LIBRA in the previous chapter), now applied to the multi-limbed robot. Results show that the SCCT controller performs better than the Jacobian transpose control scheme. Additionally, the results show that climbing on the peg board environment is attainable using the SCCT controller due to its ability to maintain compliance with a partially known environment.

Chapter 5

Conclusions and Suggestions for Further Work

5.1 Contributions of This Work

This thesis described the application of an action module planning methodology and the simplified cartesian computed torque (SCCT) control scheme to an experimental climbing robot. Simulation and experimental data were presented which show the ability of the methods to execute a climbing task.

Details of the action planning methodology and the genetic algorithm search method used, as well as specific details for their application to the climbing robot, were presented. A simulation was built for the application of the planning methodology in order to ensure that no physical constraints of the system were violated. The methodology's ability to use the physical capabilities of the robot to derive an action plan for successful execution of several tasks was shown. Simulation results demonstrated the feasibility of the method for planning the actions of the different tasks. Additional results were provided that showed the importance of the design of the action module inventory.

Several cartesian control schemes were presented along with the disadvantages and advantages of each. The SCCT control scheme was developed for multi-limbed robots which have large actuator gearing. This allowed for simplification of the dynamics of the limb of LIBRA. Simulation and experimental studies show that the assumptions were valid for the highly geared robot LIBRA. Additional results showed the improved

positioning performance of the SCCT controller over a conventional Jacobian transpose control scheme.

Finally details of the application of the action module plans and SCCT control scheme to the experimental systems were presented. Experimental results show that climbing was attainable in the peg board environment. Results were provided to demonstrate the ability of the SCCT control scheme to control the action of the multi-limbed robot LIBRA for successful execution of an action module derived plan. The difficulty of climbing, even on a fairly well known environment, was also seen.

5.2 Suggestions for Further Work

This thesis has shown the feasibility of the action module planning methodology to plan the actions of the climbing robot on several different tasks. However only one task was implemented experimentally in this thesis due to the difficulty of climbing. Experimental validation of the other tasks would help confirm the ability to create purposeful plans.

Adding grippers at the end of each of LIBRA's limbs would allow the flexibility to study more difficult climbing tasks, and better validate the use of the methodology to produce successful plans for executions of these difficult tasks. Moreover, active grippers at the end effector of LIBRA would remove the limitation which hampered control of the system as discussed in Chapter 4. This would allow a better study of the SCCT control scheme's ability to control LIBRA. Additionally, experiments could be done to study the ability of LIBRA to walk on an unknown and uneven terrain with the SCCT controller, by using the flexibility of setting the cartesian stiffness such that the body was compliant in the direction of the unknown environment, as discussed in Section 4.2.

No work was presented in this thesis on controlling forces applied to the environment while controlling the motions of the body as was demonstrated in (Sunada, 1992). A short introduction to the application of

force control with the SCCT controller was introduced in Chapter 3. However, experiments to validate the use of the simplified control scheme to control both position and forces were limited and not included in this thesis.

Ongoing research is being done on the application of the action module planning methodology to physical systems such as JPL's Lightweight Survivable Rover (LSR). An experimental rover, similar to the LSR, is currently being built in this laboratory in order to validate the planning methodology on another experimental system. Work will also be done to execute the planning methodology with on board computers to allow the rover to plan its actions somewhat more autonomously.

References

- An, C.H., *Model-Based Control of a Robot Manipulator*, MIT Press, Cambridge, USA, 1988.
- Argaez, D.A., *An Analytical and Experimental Study of the Simultaneous Control of Motion and Force of a Climbing Robot*, M.S. Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, USA, June 1993.
- Asada, H. and Slotine, J.-J. E., *Robot Analysis and Control*, Wiley-Interscience, 1986.
- Borenstein, J. and Koren, Y., "The Vector Field Histogram - Fast Obstacle Avoidance for Mobile Robots," *IEEE Transactions of Robotics and Automation*, Vol. 7, No. 3, 1991, pp. 278-288.
- Brooks, R., "A Robust Layered Control System for a mobile robot," *IEEE Journal of Robotics and Automation*, Vol. RA-2, No. 1, 1986, pp. 14-23.
- Brooks, R., "A Robot that Walks: Emergent Behaviors from a Carefully Evolved Network," *Proceedings of the IEEE International Conference on Robotics and Automation*, Scottsdale, USA, Vol. 2, 1989, pp. 692-694.
- Celaya, E. and Porta, J., "Control of a Six-Legged Robot Walking on Abrupt Terrain," *Proceedings of the IEEE International Conference on Robotics and Automation*, Minneapolis, MN, Vol. 4, 1996, pp. 2731-2736.
- Channon, P.H., Hopkins, S.H., Pham, D.T., "Optimal Control of an n-legged Robot," *Journal of Systems & Control Engineering*, Vol. 210, No. 1, 1996, pp. 51-63.

- Chen, C-H. and Kumar, V., "Motion Planning of Walking Robots in Environments with Uncertainty," *Proceedings of the IEEE International Conference on Robotics and Automation*, Minneapolis, MN, Vol. 4, 1996, pp. 3277-3282.
- Chevallereau, C., Formal'sky, A., and Perrin, B., "Control of a Walking Robot with Feet Following a Reference Trajectory Derived From Ballistic Motion," *Proceedings of the IEEE International Conference on Robotics and Automation*, Albuquerque, NM, Vol. 3, 1997, pp. 1094-1099.
- Colbaugh, R., Seraji, H., and Glass, K., "Adaptive Compliant Motion Control for Dexterous Manipulators," *International Journal of Robotics Research*, Vol. 14, No. 3, June 1995, pp. 270-280.
- Cole, J., *Rapid Generation of Motion Plans for Modular Robotic Systems*, M.S. Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, USA, June 1995.
- Craig, J., *Introduction to Robotics*, Addison-Wesley, New York, USA, 1986
- Dubowsky, S., Moore, C., and Sunada, C., "On the Design and Task Planning of Power Efficient Field Robotic Systems," *Proceedings of the ANS 6th Topical Meeting on Robotics and Remote Systems*, Monterey, CA, 1995, pp. 501-508.
- Farritor, S. and Dubowsky, S., "A Genetic Algorithm Based Navigation and Planning Methodology for Planetary Robotic Exploration," *ANS Proceedings of the 7th Topical Meeting on Robotic and Remote Systems*, Augusta, GA, April 1997.
- Farritor, S. and Dubowsky, S., "A Self-Planning Methodology for Planetary Robotic Explorers," *Proceedings of the 8th International Conference on Advanced Robotics*, Monterey, CA, July 1997, pp. 499-504.

- Farritor, S., Dubowsky, S., and Rutman, N., "On the Design of Rapidly Deployable Field Robotic Systems," *ASME Design Engineering Technical Conference*, Irvine, CA, August 1996.
- Farritor, S., Dubowsky, S., Rutman, N., and Cole, J., "A Systems-Level Modular Design Approach to Field Robotics," *Proceedings of the IEEE International Conference on Robotics and Automation*, Minneapolis, MN, Vol. 4, 1996, pp. 2890-2895.
- Fujimoto, Y. and Kawamura, A., "Proposal of Biped Walking Control Based on Robust Hybrid Position/Force Control," *Proceedings of the IEEE International Conference on Robotics and Automation*, Minneapolis, MN, Vol. 4, 1996, pp. 2724-2730.
- Gardner, J., "Force Distribution in Walking Machines over Rough Terrain," *Journal of Dynamic Systems, Measurement, and Control*, Vol. 113, December 1991, pp. 754-758.
- Gat, E., Desai, R., Ivlev, R., Loch, J., and Miller, D., "Behavior Control of Robotic Exploration of Planetary Surfaces", *IEEE Transactions on Robotics and Automation*, Vol. 10, No. 4, August 1994, pp. 480-503.
- Goldberg, D. *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA, 1989.
- Gorinevsky, D. and Schneider, A., "Force Control in Locomotion of Legged Vehicles over Rigid and Soft Surfaces," *The International Journal of Robotics Research*, Vol. 9, No. 2, April 1990, pp. 4-23.
- Gradetsky, V. and Rachkov, M., "Wall Climbing Robot and Its Applications for Building Construction," *Mechanical Systems Engineering*, Vol. 1, 1990, pp. 225-231.

- Hogan, N., "Impedance Control: An Approach to Manipulation: Part I--Theory, Part II--Implementation, Part III--Applications," *Journal of Dynamic Systems, Measurement, and Controls*, Vol. 107, 1985, pp. 1-24.
- Ilg, W. and Berns, K., "Learning Architecture Based on Reinforcement Learning for Adaptive Control of the Walking Machine LAURON," *Robotics and Autonomous Systems*, Vol. 15, No. 4, 1995, pp. 321-334.
- Khatib, O., "A unified approach for motion and force control of Robot manipulators: Operations Space Formulation," *IEEE Journal of Robotics and Automation*, Vol. RA-3, No. 1, February 1987, pp. 43-53.
- Khatib, O. "Real-time obstacle avoidance for manipulators and mobile robots," *International Journal of Robotics Research*, Vol. 5, No. 1, 1986, pp. 90-98.
- Kircanski, M. and Kircanski, N., "Resolved-Rate and Resolved-Acceleration-based Robot Control in the Presence of Actuators' Constraints," *Proceedings of IEEE International Conference. on Robotics and Automation*, Albuquerque, NM, Vol. 1, 1997, pp. 235-240.
- Klein, C. A., Olson, K. M., and Pugh, D. R., "Use of Force and Attitude Sensors for Locomotion of a Legged Vehicle," *International Journal of Robotics Research*, Vol. 2, No. 2, 1983, pp. 3-17.
- Krotkov, E. and Hoffman, R., "Terrain Mapping for a Walking Planetary Rover," *IEEE Transactions on Robotics and Automation*, Vol. 10., No. 6, December 1994, pp. 728-739.
- Kun, A. and Miller, W., "Adaptive Dynamic Balance of a Biped Robot using Neural Networks," *Proceedings of the IEEE International Conference. on Robotics and Automation*, Albuquerque, NM, Vol. 1, 1997, pp. 240-245.
- Latombe, J.C., *Robot Motion Planning*, Kluwer Academic Publisher, 1991.

- Lehtinen, H., "Force Control for Walking on Soft Terrain," *Robotica*, Vol. 14, No. 2, 1996, pp. 165-172.
- Luh, J., Walker, M., and Paul, R., "Resolved-Acceleration Control of Mechanical Manipulators," *IEEE Transactions on Automatic Control*, Vol. AC-25, No. 3, 1980, pp. 468-474.
- Luk, B., Collie, A., and Bingsley, J., "Robug II: An Intelligent Wall Climbing Robot," *Proceedings of the IEEE International Conference on Robotics and Automation*, Sacramento, CA, Vol. 3, April 1991, pp. 2342-2347.
- Madhani, A. and Dubowsky, S., "The Force Workspace: A Tool for the Design and Motion Planning of Multi-Limb Robotic Systems", *ASME Journal of Mechanical Design*, Vol. 119, No. 2, June 1997, pp. 218-225.
- McGhee, R. and Iswandhi, G., "Adaptive Locomotion of a Multilegged Robot over Rough Terrain," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 9, No. 4, 1979, pp. 176-182.
- Mills, J.K., "Simultaneous Control of Robot Manipulator Impedance and Generalized Force and Position," *Journal of Mechanisms and Machine Theory*, Vol. 31, No. 8, 1996, pp. 1069-1080.
- Morel, G. and Dubowsky, S., "The Precise Control of Manipulators with Joint Friction: A Base Force/Torque Sensor Method," *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 1, 1996, pp. 360-365.
- Nagakuba, A. and Hirose, S., "Walking and Running of the Quadruped Wall-Climbing Robot," *Proceedings of the IEEE International Conference on Robotics and Automation*, San Diego, CA, Vol. 2, 1994, pp. 1005-1013.
- Neibauer, W., "Locomotion with Articulated Legs in Pipes or Ducts," *Robotics and Autonomous Systems*, Vol. 11, 1993, pp. 163-169.

- Orin, D. and Oh, S., "Control of Force Distribution in Robotic Mechanisms Containing Closed Kinematic Chains," *Journal of Dynamic Systems, Measurement, and Control*, Vol. 102, 1981, pp. 134-141.
- Pack, D., "Perception-Based Control for a Quadruped Walking Robot," *Proceedings of the IEEE International Conference on Robotics and Automation*, Minneapolis, MN, 1996, Vol. 4, pp. 2994-3001.
- Pannu, S., Kazerooni, H., Becker, G., and Packard, A., " μ -Synthesis Control for a Walking Robot," *IEEE Control Systems Magazine*, Vol. 16, No. 1, 1996, pp. 20-25.
- Plumet, F., Morel, G., and Bidaud, P., "Shall We Use a Dynamic Model to Control the Motions of Industrial Manipulators," *Proceedings of the Ninth World Congress on Theory of Machines and Mechanisms*, Milano, Italy, 1995, pp. 235-240.
- Pratt, J., *Virtual Model Control of a Biped Walking Robot*, M.S. Thesis, Department of Electrical Engineering, Massachusetts Institute of Technology, Cambridge, USA, September 1995.
- Pratt, J., "Virtual Model Control of a Biped Walking Robot," *Proceedings of IEEE International Conference on Robotics and Automation*, Albuquerque, NM, Vol. 1, 1997, pp. 193-198.
- Raibert, M.H. and Craig, J., "Hybrid Position/Force Control of Manipulators," *Journal of Dynamic Systems, Measurement and Control*, Vol. 102, June 1981, pp. 126-133.
- Salisbury, K., "Active Stiffness Control of a manipulator in Cartesian coordinates," *Proceedings of the 19th IEEE Conference on Decision and Control*, Albuquerque, NM, Vol. 1, December 1980, pp. 95-100.
- Samson, C., Borgne, M., and Espiau, B., *Robot Control: The Task Function Approach*. Oxford, Clarendon Press, 1991.

- Schneider, S.A. and Cannon, R. H. Jr., "Object Manipulation Control for Cooperative Manipulation: Theory and Experimental Results," *IEEE Transactions on Robotics and Automation*, Vol. 8, No. 3, June 1992, pp. 383-394.
- Shan, Y. and Koren, Y., "Obstacle Accommodation Motion Planning," *IEEE Transactions on Robotics and Automation*, Vol. 11, No. 1, 1993, pp. 36-49.
- Shih, C., Gruver, A., and Lee, T., "Inverse Kinematics and Inverse Dynamics for Control of a Biped Walking Machine," *Journal of Robotics Systems*, Vol. 10, No. 5, June 1993, pp. 531-555.
- Siciliano, B., Canudas de Wit, C., and Bastin, G., Editors, *Theory of Robot Control*, London, Springer-Verlag, 1996.
- Song, S. and Waldron, K., "An Analytical Approach for Gait Study and Its Application on Wave Gaits," *International Journal of Robotics Research*, Vol. 6. No. 2, 1987, pp. 60-71.
- Song, S. and Waldron, K., *Machines that Walk: The Adaptive Suspension Vehicle*, MIT Press, 1988.
- Sunada, C., Argaez, D., Dubowsky, S., and Mavroidis, C., "A Coordinated Jacobian Transpose Control for Mobile Multi-limbed Robotic Systems," *Proceedings of the IEEE International Conference on Robotics and Automation*, San Diego, CA, Vol. 3, 1994, pp. 1910-1915.
- Sunada, C., *Coordinated Jacobian Transpose Control and its Application to a Climbing Machine*, M.S. Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, September 1994.
- Torres, M., *The Disturbance Map and Minimization of Fuel Consumption During Space Manipulator Maneuvers*, M.S. Thesis, Dept. of Mechanical Engr., Massachusetts Institute of Technology, June 1989.

- Venkataraman, S.T., "A Model of Legged Locomotion Gaits," *Proceedings of the IEEE International Conference on Robotics and Automation*, Minneapolis, MN, Vol. 4, 1996, pp. 3545-3550.
- Wampler, C. and Leifer, L., "Application of Damped-Least Squares Methods to Resolved-Rate and Resolved-Acceleration Control Schemes," *Journal of Dynamic Systems, Measurement, and Control*, Vol. 110, 1988, pp. 31-38.
- Wettergreen, D. and Thorpe, C., "Developing Planning and Reactive Control for a Hexapod Robot," *Proceedings of the IEEE International Conference on Robotics and Automation*, Minneapolis, MN, 1996. Vol. 4, pp. 2718-2723.
- Whitney, D., "Resolved Motion Rate Control of Manipulators and Human Prostheses," *IEEE Transactions on Man-Machine Systems*, Vol. MMS-10, No 2, June 1969, pp. 47-53.
- Whitney, D., "Force Feedback Control of a Manipulator Fine Motions," *Journal of Dynamic Systems, Measurement, and Control*, Vol. 99, No. 2, June 1977, pp. 91-97.
- Whitney, D., "Historical Perspective and State of the Art in Robot Force Control," *International Journal of Robotics Research*, Vol. 6, No. 1, Spring 1987, pp. 3-14.
- Williams, D., and Khatib, O., "Improved Force Control for Conventional Arms Using Wrist-Based Torque Feedback," *International Symposium of Experimental Robotics*, ISER 95, preprints, 1995, pp. 323-328.
- Yoneda, K., Iiyama, H., Hirose., Shigeo S., "Sky Hook Suspension Control of a Quadruped Walking Vehicle," *Proceedings of the IEEE International Conference on Robotics and Automation*, San Diego, CA, Vol. 2, 1994, pp. 999-1004.

Appendix A

LIBRA Kinematics

This section describes the direct kinematics of LIBRA and its limbs, including the Jacobian and Inverse Jacobian Matrix of the limbs, shown below in Figure A.1.

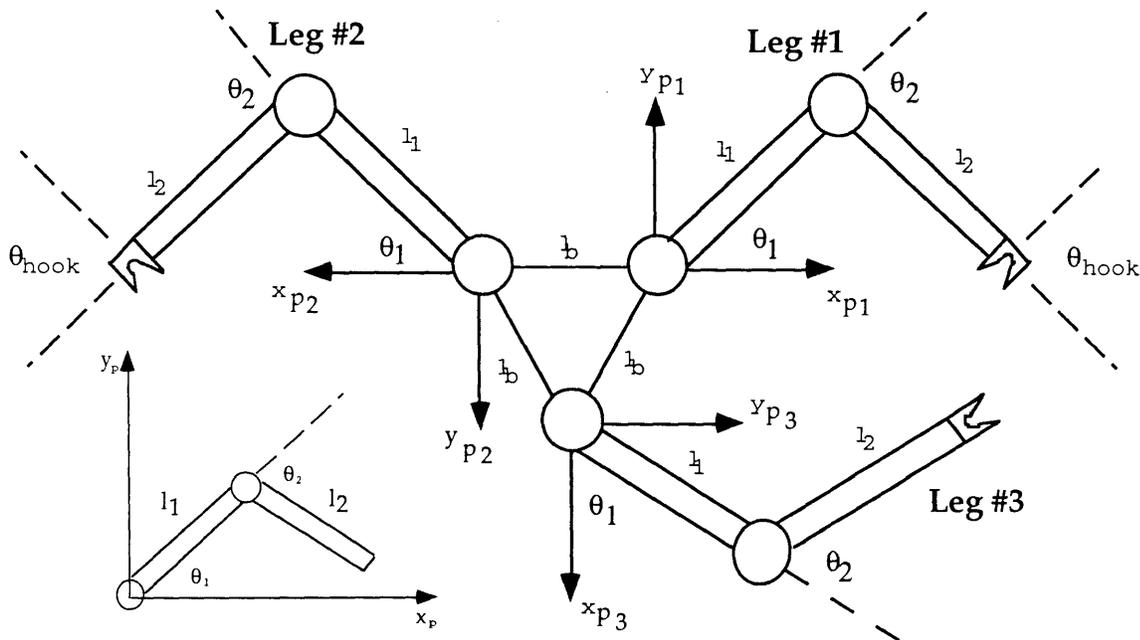


Figure A.1. Kinematic Parameters of LIBRA

Additionally, hooks one and two have an offset of l_3 from the center line of the leg as seen in Figure A.2. The position of each leg's endpoint in its port frame can be found from the following equations:

$$x_p = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_{hook})$$

$$y_p = l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_{hook})$$

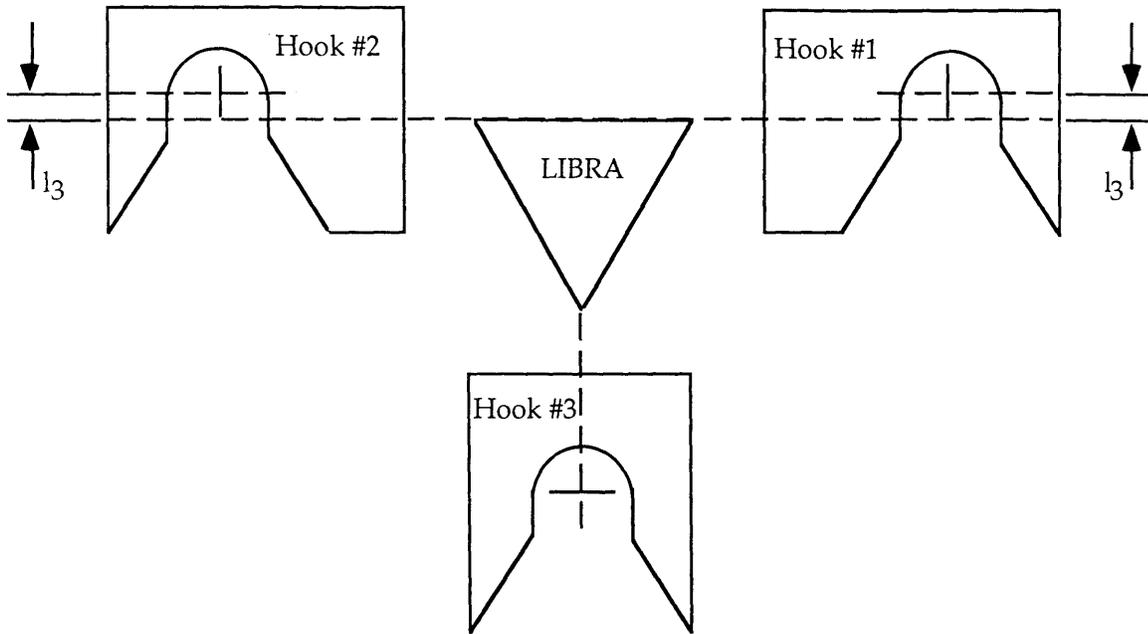


Figure A.2 Kinematics LIBRA's Hooks

The position in the port frame can then be transformed to the fixed frame as discussed in Chapter 4. Table A.1 list the kinematic parameters for each of the three legs.

Table A.1. Leg Kinematic Parameters (dimensions in cm)

	l_1	l_2	l_3	θ_{hook}
Leg #1	15.6	15.5	0.5	90°
Leg #2	15.6	15.5	0.5	90°
Leg #3	15.6	15.0	0.0	0°

For the calculation of the Jacobian and Inverse Jacobian Matrix, the offset of the hook was not considered such that the position equations became:

$$x_p = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2)$$

$$y_p = l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2)$$

Differentiating the above equation results in:

$$\begin{aligned}\dot{x}_p &= -l_1\dot{\theta}_1 \sin(\theta_1) - l_2(\dot{\theta}_1 + \dot{\theta}_2)\sin(\theta_1 + \theta_2) \\ \dot{y}_p &= l_1\dot{\theta}_1 \cos(\theta_1) + l_2(\dot{\theta}_1 + \dot{\theta}_2)\cos(\theta_1 + \theta_2)\end{aligned}$$

or:

$$\begin{bmatrix} \dot{x}_p \\ \dot{y}_p \end{bmatrix} = J_{leg} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

where:

$$J_{leg} = \begin{bmatrix} -l_1 \sin(\theta_1) - l_2 \sin(\theta_1 + \theta_2) & -l_2 \sin(\theta_1 + \theta_2) \\ l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) & l_2 \cos(\theta_1 + \theta_2) \end{bmatrix}$$

Taking the inverse of the above Jacobian matrix results in the following Inverse Jacobian Matrix:

$$J_{leg}^{-1} = \begin{bmatrix} \frac{\cos(\theta_2)}{l_1 \sin(\theta_2)} & \frac{\sin(\theta_2)}{l_1 \sin(\theta_2)} \\ \frac{-l_1 \cos(\theta_1) - l_2 \cos(\theta_2)}{l_1 l_2 \sin(\theta_2)} & \frac{-l_1 \sin(\theta_1) - l_2 \sin(\theta_2)}{l_1 l_2 \sin(\theta_2)} \end{bmatrix}$$

Additionally, the kinematic values for the body of LIBRA and the three port locations are shown in Figure A.3.

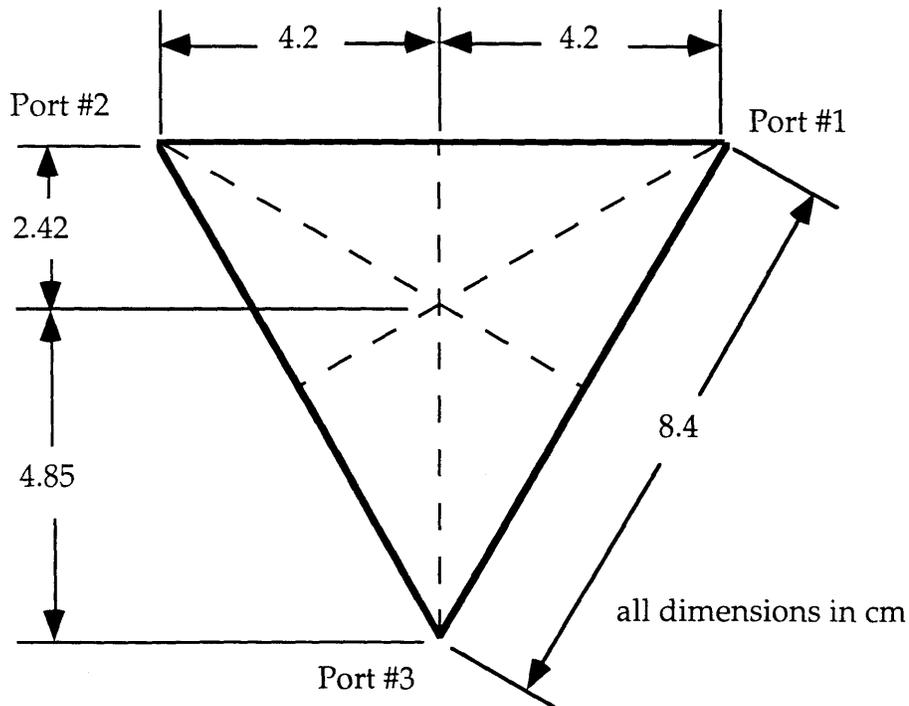


Figure A.3. Kinematics Parameters of LIBRA's Body

The Denavit-Hartenberg port transformation parameters for the three ports (and their corresponding limbs) used in Chapter 2 and Chapter 4 are listed below in Table A.2.

Table A.2. LIBRA Denavit-Hartenberg Port Transformation Parameters

D.H. Parameters	Port #1	Port #2	Port #3
x_{port} (cm)	4.2	-4.2	0
y_{port} (cm)	2.42	2.42	-4.85
θ_{port} (rad)	0	π	$-\pi/2$

Appendix B

Inverse Kinematics of LIBRA's Limb

This section describes the solution for the inverse kinematic problem of one limb of LIBRA in its port frame shown in Figure B.1.

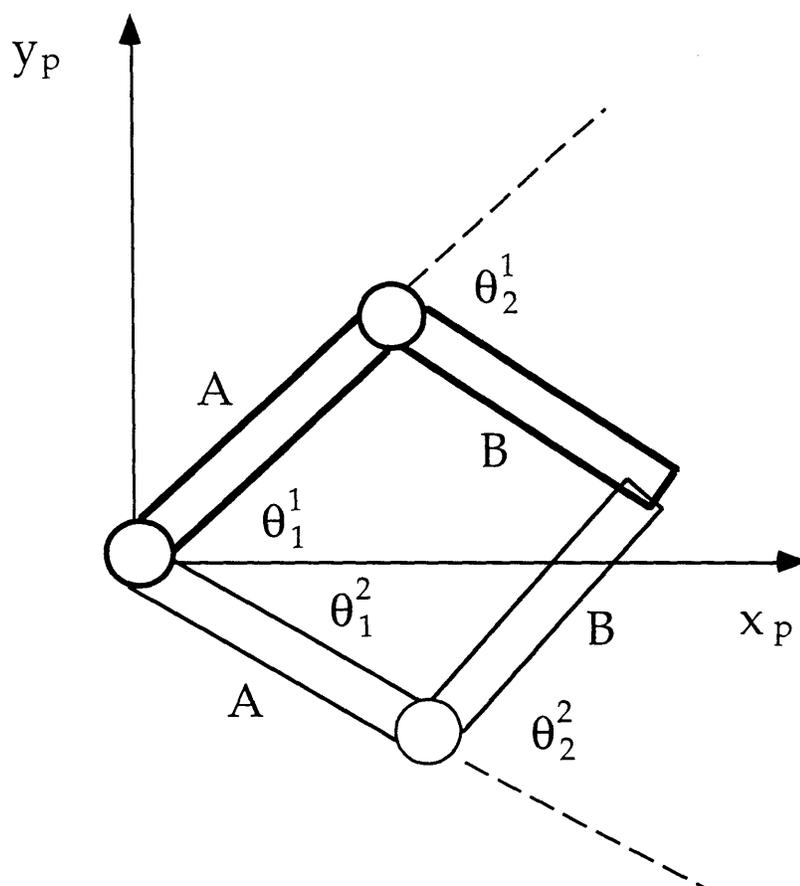


Figure B.1. LIBRA's Limb in its Port Frame

As seen in the above figure, there exist two solutions to the inverse kinematic problem. First, solving for the two solutions of θ_2 :

$$top = x^2 + y^2 - (A^2 + B^2)$$

$$bot = 2AB$$

$$c2 = \frac{top}{bot}$$

$$\theta_2^1 = \cos^{-1}(c2)$$

$$\theta_2^2 = -\cos^{-1}(c2)$$

Then solving for the first solutions of θ_1 :

$$Q_1 = A + B\cos(\theta_2^1)$$

$$R_1 = -B\sin(\theta_2^1)$$

$$T_1 = B\sin(\theta_2^1)$$

$$top_1 = y - \frac{Q_1 x}{R_1}$$

$$bot_1 = T - \frac{Q_1^2}{R_1}$$

$$c_1 = \frac{top_1}{bot_1}$$

$$s_1 = \frac{x}{R_1} - \frac{Q_1 c_1}{R_1}$$

$$\theta_1^1 = \tan^{-1}\left(\frac{s_1}{c_1}\right)$$

Similarly solving for the second solution of θ_1 :

$$Q_2 = A + B \cos(\theta_2^2)$$

$$R_2 = -B \sin(\theta_2^2)$$

$$T_2 = B \sin(\theta_2^2)$$

$$top_2 = y - \frac{Q_2 x}{R_2}$$

$$bot_2 = T - \frac{Q_2^2}{R_{21}}$$

$$c_2 = \frac{top_2}{bot_2}$$

$$s_2 = \frac{x}{R_2} - \frac{Q_2 c_2}{R_2}$$

$$\theta_1^2 = \tan^{-1} \left(\frac{s_2}{c_2} \right)$$

The angles for the inverse kinematic solution can then be selected based on the choice of configuration as seen in Table B.1.

Table B.1. Kinematic Angle Solutions

	θ_1	θ_2
Configuration #1	θ_1^1	θ_2^1
Configuration #2	θ_1^2	θ_2^2

Appendix C

Peg Positions for the Four Tasks

This section lists the peg positions used for the four tasks, described in Section 2.4 of this thesis. The peg positions for Ladder Task, shown in Table C.1, were also used for the experimental climbing in Chapter 4.

Table C.1. Peg Positions (in inches) for the Four Tasks

Peg #	Ladder Task		Circle Task		Across Task		H Task	
	X	Y	X	Y	X	Y	X	Y
1	0	0	20	0	0	0	0	0
2	18	0	28	2	0	22	0	6
3	0	6	34	6	6	0	0	12
4	18	6	38	12	6	22	0	18
5	0	12	40	20	12	0	0	24
6	18	12	40	28	12	22	0	30
7	0	18	40	36	18	0	0	26
8	18	18	38	44	18	22	0	42
9	0	24	34	50	24	0	18	0
10	18	24	28	54	24	22	18	6
11	0	30	20	56	30	0	18	12
12	18	30	12	54	30	22	18	30
13	0	36	6	50	33	0	18	36
14	18	36	2	44	36	24	18	42
15	0	42	0	36	39	0	24	12
16	18	42	0	28	42	24	24	30
17	0	48	0	20	45	0	30	12
18	18	48	2	12	48	24	30	30
19	0	54	6	6	51	0	36	0
20	18	54	12	2	54	24	36	6
21	0	60	20	20	57	0	36	12
22	18	60	20	28	60	24	36	30
23			20	36	63	0	36	36
24			20	19	66	24	36	42
25			20	37	70	24	54	0
26					74	24	54	6
27					78	24	54	12
28					82	24	54	18
29					86	24	54	24
30					90	24	54	30
31							54	36
32							54	42

Appendix D

Plans For Execution of the Four Tasks

This section lists the action plans, which were developed by hand, required to solve each of the four tasks studied in Chapter 2. The step sizes used for commands which move the body in position and angle are indicated. The plan shown in Table B.1 was used as the input for the experimental climbing discussed in Chapter 4.

Table B.1. Plan for the Ladder Task (Step Size = 3" & 15°)

Step #	Action										
1	0	12	300	23	211	34	3	45	313	56	3
2	106	13	3	24	308	35	3	46	217	57	318
3	301	14	3	25	112	36	311	47	300	58	122
4	205	15	306	26	300	37	215	48	3	59	317
5	300	16	110	27	3	38	312	49	3	60	221
6	3	17	305	28	3	39	116	50	315	61	300
7	3	18	209	29	310	40	300	51	219	62	3
8	303	19	300	30	114	41	3	52	316	63	3
9	207	20	3	31	309	42	3	53	120		
10	304	21	3	32	213	43	314	54	300		
11	108	22	307	33	300	44	118	55	3		

Table B.2. Plan for the H Task (Step Size = 3" & 15°)

Step #	Action										
1	0	13	302	25	1	37	1	49	1	61	222
2	309	14	205	26	1	38	1	50	5	62	124
3	111	15	304	27	315	39	3	51	5	63	230
4	301	16	3	28	1	40	123	52	3	64	332
5	203	17	1	29	122	41	300	53	3	65	200
6	300	18	3	30	200	42	5	54	5	66	3
7	3	19	212	31	1	43	5	55	5	67	3
8	3	20	300	32	1	44	5	56	5	68	3
9	302	21	1	33	218	45	1	57	229	69	3
10	204	22	1	34	300	46	3	58	331		
11	300	23	116	35	321	47	330	59	200		
12	3	24	300	36	100	48	200	60	3		

Table B.3. Plan for the Across Task (Step Size = 1" & 15⁰)

Step #	Action										
1	0	37	5	73	4	109	3	145	200	181	1
2	303	38	5	74	4	110	3	146	1	182	1
3	106	39	5	75	317	111	5	147	1	183	1
4	301	40	110	76	100	112	5	148	1	184	328
5	204	41	311	77	2	113	5	149	1	185	200
6	300	42	100	78	114	114	5	150	3	186	1
7	1	43	1	79	215	115	320	151	3	187	1
8	1	44	1	80	100	116	1	152	3	188	5
9	1	45	1	81	1	117	1	153	5	189	5
10	1	46	1	82	1	118	1	154	5	190	5
11	1	47	1	83	1	119	1	155	5	191	5
12	1	48	1	84	1	120	1	156	226	192	5
13	305	49	110	85	1	121	1	157	100	193	5
14	108	50	209	86	1	122	322	158	1	194	5
15	303	51	112	87	1	123	120	159	1	195	5
16	206	52	313	88	116	124	221	160	5	196	5
17	300	53	100	89	319	125	1	161	5	197	1
18	1	54	1	90	100	126	1	162	5	198	1
19	1	55	1	91	2	127	1	163	5	199	1
20	1	56	1	92	116	128	223	164	5	200	1
21	1	57	1	93	217	129	1	165	5	201	229
22	1	58	112	94	100	130	1	166	5	202	100
23	1	59	211	95	1	131	1	167	127	203	1
24	307	60	3	96	1	132	324	168	300	204	1
25	1	61	3	97	1	133	221	169	1	205	5
26	1	62	114	98	1	134	122	170	1	206	5
27	1	63	1	99	1	135	200	171	5	207	5
28	1	64	1	100	1	136	1	172	5	208	5
29	1	65	315	101	1	137	1	173	5	209	5
30	1	66	112	102	118	138	1	174	5	210	5
31	309	67	213	103	321	139	1	175	5	211	5
32	200	68	1	104	100	140	1	176	5	212	5
33	4	69	1	105	2	141	1	177	5	213	5
34	4	70	114	106	118	142	223	178	5	214	130
35	207	71	1	107	219	143	325	179	5		
36	5	72	1	108	300	144	124	180	1		

Table B.4 Plan for the Circle Task (Step Size = 1" & 15⁰)

Step #	Action										
1	0	35	4	69	3	103	206	137	2	171	2
2	300	36	1	70	1	104	308	138	2	172	4
3	4	37	1	71	305	105	222	139	2	173	4
4	4	38	301	72	204	106	125	140	312	174	2
5	4	39	4	73	200	107	200	141	210	175	4
6	4	40	4	74	3	108	3	142	123	176	4
7	4	41	1	75	3	109	3	143	2	177	4
8	4	42	1	76	3	110	3	144	5	178	316
9	4	43	1	77	204	111	3	145	2	179	215
10	4	44	1	78	306	112	5	146	5	180	300
11	318	45	302	79	224	113	5	147	2	181	4
12	217	46	220	80	122	114	208	148	4	182	4
13	324	47	121	81	200	115	309	149	4	183	4
14	121	48	1	82	3	116	5	150	5	184	4
15	300	49	1	83	3	117	3	151	5	185	4
16	4	50	303	84	3	118	2	152	313	186	4
17	4	51	1	85	3	119	5	153	212	187	4
18	4	52	1	86	3	120	3	154	5	188	4
19	4	53	202	87	3	121	310	155	5	189	4
20	319	54	5	88	3	122	5	156	5	190	4
21	218	55	5	89	3	123	3	157	5	191	321
22	5	56	1	90	205	124	3	158	2	192	122
23	4	57	1	91	307	125	2	159	4	193	317
24	1	58	3	92	221	126	2	160	2	194	216
25	5	59	1	93	123	127	209	161	2	195	4
26	4	60	3	94	200	128	5	162	314	196	4
27	320	61	1	95	3	129	3	163	213	197	4
28	124	62	3	96	3	130	3	164	5	198	4
29	5	63	1	97	3	131	2	165	5	199	4
30	1	64	304	98	3	132	2	166	2	200	4
31	1	65	203	99	3	133	311	167	2	201	4
32	219	66	1	100	3	134	3	168	4		
33	5	67	3	101	3	135	3	169	2		
34	4	68	3	102	3	136	2	170	4		