Dynamic Modeling of an Articulated Forestry Machine for Simulation and Control

Soumen Sarkar

B. Eng. (Jadavpur University, Calcutta, India), 1989

M.Eng. (Jadavpur University, Calcutta, India), 1992

Department of Mechanical Engineering

McGill University

Montreal, Quebec, Canada

A Thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements of the degree of Master of Engineering

June 1996

0-612-29628-8

Abstract

Recently, robotic technology has begun to play an important role in forestry operations. An important class of forestry machines is comprised of systems equipped with a mobile platform fitted with an articulated arm carrying a tree processing head. The dynamics of such systems are needed for simulation and control purposes. In contrast to conventional industrial manipulators, which are mounted on stationary bases, a mobile manipulator is dynamically coupled with its base. Base compliance, non-linearity and coupled dynamics result in positioning inaccuracies which in turn give rise to control problems.

The dynamics of the FERIC forwarder forestry machine including its compliant tires were developed and implemented symbolically in compact form with the help of an iterative Newton-Euler dynamic formulation. Various models with increasing complexity were derived. Based on a simplified dynamics model, a valve-sizing methodology was developed and used to size hydraulic proportional valves of the machine's actuators.

System parameters have been obtained by various methods, including use of blueprints, weighing, solid modeling and various experiments. A set-point feedforward controller was designed and the machine's responses for various inputs were obtained to analyze the dynamic behavior of the system. Although initial simulations were done in Matlab and Simulink, C programs were developed for increased speed of execution. In addition, techniques to minimize computation time have been developed and applied to result in almost real time simulation.

Résumé

Récemment, la technologie robotique a pris de l'importance dans le secteur des opérations forestières. Une importante classe de machines forestières comprend les machines à plate-formes mobiles auxquelles on a ajouté un bras articulé opérant une tête multi-fonction. La dynamique de tel systèmes est requise pour fins de simulation et de commande. Contrairement aux manipulateurs industriels conventionnels, qui sont montés sur une base fixe, un manipulateur mobile est couplé dynamiquement à sa base. Les déformations de la base, de même que la non-linéarité du système et la dynamique couplée, résultent en des positionnement imprécis qui, à leur tour, amène des problèmes de commandes.

La dynamique de la machine forestière FERIC, incluant la déformation des pneus, a été développée et réalisée symboliquement sous forme compacte, en utilisant la formulation dynamique itérative de Newton-Euler. Différents modèles à complexité croissante ont été développés. Basé sur la dynamique d'un modèle simplifié, une méthodologie de dimensionnement des valves a été développée et utilisée pour dimensionner les valves hydrauliques proportionnelles des actuateurs de la machine.

Les paramètres du système ont été obtenus à l'aide de différentes méthodes, incluant les plans originaux, pesage, modelage solide, et diverses expériences. Un système d'asservissement à boucle ouverte avec consigne a été conçu et les réponses temporelles de la machine furent obtenues pour différents signaux d'entré afin d'analyser le comportement dynamique du système. Quoique les simulations initiales ayant été faites avec Matlab et Simulink, des programmes en C ont été développés pour augmenter la vitesse d'exécution. De plus, des techniques pour minimiser le temps de calcul ont été développées et appliquées de sorte que des simulations et animations en temps quasi-réel ont pu être obtenues.

Acknowledgments

First, I would like to acknowledge my supervisor Professor Evangelos Papadopoulos. His untiring guidance, encouragement and support can not be equated with any type of formal thanks giving.

I would like to thank all my colleagues and friends at the McGill Research Centre for Intelligent Machines who extended their helpful hands whenever I needed. I would like to express my appreciation to Bin Mu, Real Frenette and Jean Courteau for their help to carry out various experiments at the east division of Forest Engineering Research Institute of Canada (FERIC).

This research has been funded by Funded by the Ministère de l'Industrie, du Commerce, de la Science et de la Technologie (MICST) for the project entitled Applications des Technologies Robotiques aux Equipements Forestiers, to whom I wish to express my gratitude.

Finally, I would like to thank my parents and brother for their love and support during my entire research career.

To my parents

Contents

List of Figures

List of Tables

Nomenclature

a_i	:	coefficient of the *i*th order polynomial.
$A_{s,b}$	:	average area of the cylinder (subscript *s* for stick and *b* for boom).
b_i	:	coefficient of friction at joint i.
b_x	:	total damping due to four tires in the roll direction.
b_y	:	total damping due to four tires in the pitch direction.
b_z	:	total damping due to four tires in the bounce direction.
B	:	damping matrix of the tire model.
c	:	discharging coefficient.
c_i	:	cosine of angle q_i.
D	:	volumetric fluid displacement of the motor.
e_i	:	position error for the ith joint.
$\dot{e}_i$	:	velocity error for the ith joint.
E_{ss}	:	steady-state error.
${}^i f_i$	:	force vector at joint i expressed in frame i.
${}^i F_i$	:	force vector at the center of mass of link i expressed in frame i.
g	:	acceleration due to gravity.
G	:	Vector of gravity terms.
G_i	:	gravity terms corresponding to joint i.
I_{ixx}	:	moment of inertia of link i with respect to an axis parallel to the $\hat{x}_i$ axis, located at the center of mass of link i.

I_{ixy}	:	product of inertia of link i with respect to a plane parallel to the plane $x_i y_i$ passing through the center of mass of link i.
${}^iI_i^c$	:	inertia tensor of link i with respect to the center of mass of link i expressed in a frame located at the center of mass and with orientation the same as that of the ith D-H frame.
J	:	Jacobian of a three degrees of freedom system.
k_{P_i}	:	ith diagonal element of the control matrix for position gains.
k_{V_i}	:	ith diagonal element of the control matrix for velocity gains.
k_x	:	total stiffness due to four tires in the roll direction.
k_y	:	total stiffness due to four tires in the pitch direction.
k_z	:	total stiffness due to four tires in the bounce direction.
K	:	stiffness matrix of the tire model.
K_P	:	diagonal control matrix for position gains.
K_V	:	diagonal control matrix for velocity gains.
l_i	:	length of link i.
m_i	:	mass of link i.
m_{ij}	:	element (i, j) of a mass matrix.
M	:	mass matrix.
n	:	gear ratio from swing to swing motor.
${}^in_i$	:	moment vector at joint i expressed in frame i.
${}^iN_i$	:	moment vector with respect to the center of mass of link i expressed in frame i.
p_{op}	:	operating pressure.

${}^{a}p_{c}^{b}$	:	position vector of origin of frame c, with respect to point b and expressed in frame a.
$P_{(s,b,sw)}$	:	power required for a trajectory (subscript s for stick, b for boom and sw for swing).
q_i	:	joint variable of link i.
$\dot{q}_i$	:	angular velocity of link i.
$\ddot{q}_i$	:	angular acceleration of link i.
$\hat{\mathbf{q}}_d$	:	desired (3×1) set-point vector to the controller.
$\hat{\mathbf{q}}_{ss}$	:	steady-state condition (3×1) vector.
Q_{sw}	:	flow in the swing motor.
$Q_{s,b}$	:	flow in the cylinder (subscript s for stick and b for boom).
${}^{i-1}R_i$	:	rotation matrix from frame i-1 to frame i.
s_i	:	sine of angle q_i.
t	:	time variable for a trajectory.
t_f	:	final time (end of a trajectory).
T	:	period of oscillation.
${}^{i-1}T_i$	:	transformation matrix from frame i-1 to frame i.
${}^{i}\dot{v}_i$	:	linear acceleration vector of link i expressed in coordinate frame i.
${}^{i}\dot{v}_{c_i}$	:	linear acceleration vector of center of mass of link i expressed in frame i.
V	:	Vector of Coriolis and centrifugal terms.
V_i	:	Coriolis and centrifugal terms corresponding to joint i.
$x_{\min}$, $x_{\max}$	:	minimum and maximum link position.

$\dot{x}_{s,b}$	:	velocity of the piston (subscript *s* for stick and *b* for boom).
$(\hat{x}_i, \hat{y}_i, \hat{z}_i)$	:	unit vectors along the x, y, and, z directions in the coordinate frame i.
X	:	generalized position vector of the base center of mass of the base with respect to world frame.
$\dot{X}$	:	generalized velocity vector of the base center of mass with respect to world frame.
Δp_m	:	pressure drop in the swing motor.
$\Delta p_{s,b}$	:	pressure drop in the cylinder (subscript *s* for stick and *b* for boom).
$\Delta p_{v,(s,b,m)}$	:	pressure drop at the valve (subscript *s* for stick, *b* for boom and *m* for motor).
ζ_i	:	controller damping for ith joint.
ν_i	:	controller frequency for ith joint.
τ	:	torque vector.
τ_{ff_i}	:	gravity compensation feedforward term for the ith joint.
τ_i	:	torque vector at joint i.
ω	:	frequency of oscillation.
${}^i\omega_i$	:	angular velocity vector of link i expressed in frame i.
${}^i\dot{\omega}_i$	:	angular acceleration vector of link i expressed in frame i.

1. Introduction

1.1 Manipulators on Forestry Vehicles

Although industrial robots do not look like humans they may do the work of humans. Present industrial robots are actually mechanical handling devices that can be manipulated under computer control. The mechanical handling device, or the manipulator, emulates the arm of a human. The joints are driven by electric, pneumatic, or hydraulic actuators, which give manipulators more potential power than human beings. The computer, which is an integral part of every modern manipulator system, contains a control program and a task program. The task program is provided by the user and specifies the manipulator motions required to complete a specific job.

At present, more and more researchers are showing interest in employing manipulators in dangerous and hazardous environments and performing undesirable jobs. A typical unstructured and harsh environment includes the forests.

Recently, robotic devices have begun to play an important role in forestry operations. An important class of forestry machines is comprised of systems equipped with a mobile platform fitted with an articulated arm carrying a tree processing head. The dynamics of the system is needed for simulation and control of the machine. In contrast to conventional industrial manipulators which are mounted on stationary bases, a mobile manipulator is dynamically coupled with its base. Base compliance, non-linearity and coupled dynamics result in positioning inaccuracies which in turn give rise to control problems.

Many forestry machines are equipped with manipulators mounted on a mobile platform whose main purpose is to grab a tree close to its roots and cut it, delimb it and cut the main stem to small logs. Due to the tire/ground compliance, the base of the manipulator moves. The total system can be modeled as a manipulator mounted on a compliant base. The degree of compliance depends on the compliance characteristics of the ground and on tire specifications and inflation pressure.

1.2 Motivation

The purpose of the thesis is to develop dynamic models for an electrohydraulic forestry machine, which will be used to develop a training simulator, for sizing components, and for system design and control.

Designing of a training simulator: Training simulators become important now-a-days, as they give the feeling of operating the actual machine without being in it. A simulator can reduce training costs since it eliminates the possibility of machine damage or even personal injury of novice trainees. It helps to realize the critical or dangerous maneuvers, which is risky in an actual machine. In this project one of the goals is to develop a training simulator for the FERIC (Forest Engineering Research Institute of Canada) machine. The simulator is a visual graphic simulator, which consists of a Silicon graphics workstation coupled with a joystick to control the graphical image of the actual machine. The dynamics of the system is necessary to obtain the actual motion of the machine.

Valve Sizing: Field harvesters are heavy duty machines equipped with hydraulically powered actuators and electrohydraulic valves. Accurate sizing of actuation components requires a dynamic model of the system. Valves are sized based on two factors, the pressure drop across the valve and the flow through the valve. The dynamic model is necessary to calculate the pressure drop across the valve for a desired trajectory (i.e. the flow through the valves).

Controller design: "Plant" dynamics is essential in designing, verifying, and evaluating various control algorithms. By playing with different control parameters (especially controller gains) a control engineer can observe various dynamic behaviors of the system and finally choose a proper controller to improve actual machine performance.

1.3 Literature Survey

1.3.1 Field Robotics

During the infancy of robotics, manipulators were used either for research or for industrial purposes. Presently, manipulators are applied in different sectors like mining, nuclear, military, construction, marine, space agriculture and forestry [75]. A large class of these manipulators are mobile and mounted on a compliant base.

An important application of field manipulators is in mining. Remotely operated and autonomous ore-excavation technology could eventually eliminate the need for miners to travel deep underground [4]. A robot named ROSEE, designed by engineers at the Department of Energy's Hanford site, will minimize the risk of radiation exposure to workers cleaning up the residue left by America's manufacture of nuclear weapons [83]. The robot vehicle should have some specific properties in order to operate in nuclear environments, such as being very safe to use [39]. In order to have total control over such a robotic system, the human and computer control are integrated. The "man in the loop" can accomplish non-programmable tasks, while a computer can reduce operator fatigue by performing repetitive tasks [73]. A remote-control shovel [45] allows its operator feel what is happening from a remote site, making the removal of hazardous waste simple and safe. In 1992, the first major international conference for exposition on environmental pollution control and technology to remedy was held [84]. In the case of extraterrestrial surface construction, transportation and mining, low gravity issues become extremely important [37].

Application of the concept of mobile robotics to the operation and maintenance of nuclear facilities has evolved since 1983. The first step in this evolutionary process was the demonstration of legged locomotion technology. The second step was the use of robotics technology in conjunction to locomotion. The final stage so far is the incorporation of enhanced mobility and dexterity, increased intelligence and greater strength in the manipulator arm and transporter. The detail of the evolution and technology development is described by Carlton and Bartholet [7].

The different possibilities of robot applications in underground hard rock mining operations have been discussed by Vagenas [82]. Field robots are moving beyond radioactive cleanups to bomb disposal, fire fighting and more [27]. During late 1985 the Army Materiel Command Headquarters gave a task to the U.S. Army Human Engineering Laboratory (HEL), AMC's lead agency for field oriented robotics, to develop a program in robotics which would achieve "critical mass" for a few key programs. A survey was conducted by Shoemaker in three different domains important in the field of defense, namely Teleoperated Mobile Antiarmor, Material Handling Robotics and Robotic Combat Vehicles [76].

For heavy duty work e.g. applications in Civil Engineering (concrete pouring, building maintenance etc.), a large manipulator with sufficient power is required [74]. A reprogrammable control system allowing for variable motions in performing a variety of pre-planned handling tasks was developed by Smidt et al. [79]. A hierarchical control architecture was designed and a man-machine interface was developed based on a graphic display and a joystick. The basic methods for trajectory planning with collision detection and avoidance can be found in reference [79]. There are many difficulties that must be overcome before robotics can be successfully implemented in construction on an industry wide basis. One of the severe problems is the need for carrying large payloads and for machine mobility. In addition, since the base is not fixed, the compliance due to vehicle suspension and tires affect manipulator accuracy. The various problems include mobility, sensing, gripper design, modeling and control systems, accuracy, hardware weight and stability, and lastly the environmental factor [78].

Following the development of the first industrial robots in the USA in 1961, several companies in UK [12], Federal republic of Germany [87], Finland [41], Canada [61], Sweden [71] came forward to cope up with the new technology. Obayashi has described some social and economical issues due to automation in construction industry [63]. Fukuda has come up with detailed designs of different parts of a manipulator to be used for heavy construction. The self leveling mechanism for bucket control has been found quite interesting and details can be found in [23]. Different concepts of using a robot in Civil Engineering jobs especially in the construction area have been discussed by Okazaki [64], [65]. In general, manipulators with very large reach are used in construction engineering applications. Naturally, low payload devices are not effective while modifications are necessary in designing controller hardware. Some of these issues have been pointed out by Wanner [88].

Presently, automatic control systems for construction machinery are getting the attention of the research community. The control systems consist of a microprocessor based controller, sensors and hydraulic actuators. The non-linear characteristics of hydraulic actuators and the low rigidity of the structure of a construction machinery make it difficult to achieve high control accuracy and high stability performance. Details of a control algorithm consisting of a combination of feedback and feedforward control with non-linear compensation, has been discussed by Kakuzen et al. [35].

Remote handling in hostile environments, including space, nuclear facilities, and mines requires hybrid systems, as close co-operation between state of the art teleoperation and advanced robotics is needed. Teleoperation with kinesthetic feedback is being investigated by researchers since it provides an operator with a feel of the robot workload and hence the robot can be controlled more effectively. Applications such as a prevention of satellite drift or transferring material at sea can be found in detail in [93].

In the agricultural sector too the application of manipulators is quite frequent now-a-days. By 1930, farm machinery began making the transition to larger, more comprehensive machines for large scale farming [33]. Sophisticated agricultural robots can be found in Australia. The University of Western Australia has done extensive work on a robotic sheep shearer [38], [81]. The Agricultural Engineering Department of the Lousiana Agricultural Experiment Station has developed a laboratory model of five degrees-of-freedom robotic seedling transplanter [32]. In 1983 different possibilities of using a manipulator in agriculture were explored by Kulz [43]. Some attempts are made by Edan to control an agricultural robot to pick up melons using 3D real time vision [18].

Robotics has great potential to meet the need for enhancing the productivity and quality of U.S. greenhouse industry. A robotic workcell has been developed at the University of Georgia Experiment Station, which is also where the MSFC (Marshall Space Flight Center) gripper system has been tested and evaluated. A force sensing robotic gripper system has been developed at the Productivity Enhancement Complex at the Marshall Space Flight Center. The details of hardware and software design for the controller and the gripper have been explained by Gill [25].

In Canada, planning for the application of automatic machines in forest industry started late 1970's. The economic importance of forestry in Canada and the potential for robotics in forest operations have been discussed in detail by Courteau [10]. Research on teleoperated excavators for forest applications was initiated by P. Lawrence and his team in British Columbia since 1985. A test-bed machine was loaned to a project aiming at implementing a resolved motion control algorithm. Electric hand controls, on-board computer, electro-hydraulic pilot valves, machine joint angle sensors and machine pressure sensors were added to an excavator machine to control it in cylindrical co-ordinates using inverse kinematics [46].

1.3.2 Dynamics

In order to design, improve performance, simulate the behavior, and finally control a system or "plant", it is necessary to obtain its dynamics.

In order to develop the dynamics of a manipulator, a kinematic model of the manipulator is required first. The kinematics modeling is done first by attaching frames to every link. The usual convention to attach frames in the links of a manipulator is called Denavit-Hartenberg notation [14]. The kinematic modeling of a mobile manipulator can be done by expressing the mobile manipulator's kinematics with homogeneous matrices [62], [59]. For a serial manipulator with more than four degrees-of-freedom, the inverse kinematics problem is quite difficult. Sometimes it is not possible to get a closed-form solution. Thus efficient numerical solution of the inverse kinematic problem has become popular [2]. Different issues and methods of kinematic analysis are discussed by Gupta (zero reference position method), Paul (homogeneous transformation representation method) and McCarthy (dual orthogonal matrix method) [28], [68], [55]. Kreutz-Delgado et al. presented kinematic analysis for a seven degrees-of-freedom serial link spatial manipulator with revolute joints. The redundancy is parameterized by a scalar variable [42]. For a mobile manipulator the base frame moves as a result, the motion propagates to all the links of the manipulator. Minami et al. proposed a method slightly different from the Newton-Euler method as far as frame attachment is concerned, to calculate inverse dynamics [60].

The dynamics of a manipulator can be obtained in various ways namely using a Newton-Euler dynamic formulation, a Lagrangian formulation, Kane's Method, and others. The Newton-Euler method is based on Newton's second law of motion with its rotational analog, called Euler's equation. It describes how forces and moments are related to acceleration. In the iterative Newton-Euler algorithm, the position, velocity and acceleration of the joints are known. With these as input and assuming that the mass properties of the manipulator and any externally acting forces are known, the joint torques required to cause this motion can be calculated. The algorithm is based on a method published by Luh, Walker, and Paul in [52]. Another iterative method has been proposed by Featherstone [20] that uses articulated-body inertia and other spatial quantities. However this method is less efficient for manipulators with many degrees-of-freedom.

The overall Newton-Euler formulation is based on a "force balance" approach to dynamics. On the other hand the Lagrangian formulation is an "energy-based" approach to dynamics. Lagrange's formalism has been applied in two ways. The first employs an independent set of generalized co-ordinates [85]. The second approach uses dependent co-ordinates, which requires the use of Lagrange's multipliers [8]. The second approach has been successfully applied by Megahed and Renaud [57]. Another approach has been developed by Luh and Zheng [51]. They use an equivalent tree structure, which is modeled with a Newton-Euler algorithm, and Lagrange's multipliers to introduce the constraints of the closed loops.

The classical Lagrangian formulation for manipulator dynamics is inefficient. The efficiency of Newton-Euler formulation is due to the two factors: the recursive structure of the computation, and the representation chosen for the rotational dynamics. Recursive Lagrangian dynamics for rigid manipulators has been discussed previously by Hollerbach [31] and for flexible manipulators by Book [5]. A general algorithm is developed to model the dynamic equation of both rigid and flexible arms [50], but the equation is generally larger than that for rigid links. Silver has shown that with a proper choice of variables, the Lagrangian formulation is equivalent to the Newton-Euler formulation [77].

Another method of deriving dynamic equations is by Kane's method which arises directly from d'Alembert's principle in the Lagrangian form. It has the advantages of a Newton's mechanics formulation without the corresponding disadvantages. In this method non-working interactive forces are automatically eliminated from the analysis [36]. Other methods that can be used to derive equations of motion include Roberson-Wittenburg's method and Popov's method [70]. A symbolic analytical procedure to obtain a dynamic model of a manipulator with complex chain structure can be found by using dual vectors and the principle of virtual work [26]. Another method of manipulator modeling is usage of spatial operator algebra [72]. The algebra makes it easy to see the relationship between abstract expressions and recursive algorithms that propagates spatial quantities from link to link. It also reveals the equivalence of Lagrangian and Newton-Euler formulations.

For a simple fixed-base serial chain manipulator the derivation of dynamics is simple and straightforward, but the opposite holds true for a complex robotic system. In deriving manipulator dynamics, the direct differentiation of kinematic functions is inefficient [47]. The efficiency considerations regarding manipulator kinematics necessitate special formulations to compute Jacobians [69], [21]. The comparison of six methods for calculating the Jacobian for a seven degrees-of-freedom manipulator has been reported by Orin and Schrader [66]. There is an inefficiency due to the growth of common subexpressions and is readily observed when using the built-in differentiation functions in symbolic algebra systems such as Mathematica [91], MAPLE [9] and MACSYMA [54]. This problem has been revealed by several researchers [6], [44], [48], [40]. It is well known that using symbolic algebra to simplify the expressions, especially those involving trigonometric functions, can improve efficiency greatly.

A complete dynamic model of a robotic system is a set of non-linear coupled differential equations [30]. Artificial neural networks are well suited for this application due to their ability to represent complex functions and, potentially to operate in real time. The application of an artificial neural network to dynamic modeling of robotic system has been investigated by Eskandarian [19].

1.3.3 Base Compliance

Presently many industrial manipulators are mounted on a fixed rigid base. In order to increase a system's workspace, a manipulator can be mounted on a mobile vehicle, but base compliance hampers system performance. There has been little prior research in the dynamic coupling of the manipulator and the vehicle. Examples can be found in research related to control problems [53], [49], [15].

The dynamic coupling between vehicle and manipulator has been treated as two separate subsystems by Wiens, thereby decoupling the integrated system [90].

Joshi and Desrochers derived dynamic equation for a two link manipulator mounted on a platform subject to random disturbances [34]. They used an equivalent angle-axis pair (K, Θ) to describe the orientation of the base. By changing vector K the effect of roll, pitch and yaw is simulated. In practice, it is not straightforward to know the type of change for the vector K.

Statically, base compliance gives rise to static errors in positioning the manipulator's end effector. The system accuracy can be dramatically improved if the base compliance is incorporated in the model. Further improvement in accuracy has been achieved by West, Hootsmans, Dubowsky, and Stelman with endpoint feedback control of the position of the end effector relative to the task frame [89].

A planar manipulator with three degrees-of-freedom and with bounce and pitch disturbance has been studied by Dubowsky and Tanner [16]. In this study, it was assumed that the vehicle is far more massive than the manipulator system. The main assumption is that the motion of the manipulator does not affect the vehicle. This assumption might not be true for many practical applications. If the masses of the manipulator and the vehicle are of the same order of magnitude the problem becomes more difficult due to coupling. Hootsmans and Dubowsky also show that an extended Jacobian transpose control algorithm can perform well for large motions in the presence of modeling errors and the limitations imposed by sensors available for highly unstructured field environments.

1.3.4 Stability

High speed motions of mobile manipulators can dynamically disturb their vehicles, and it is even possible for the vehicle to tip over. Dubowsky and Vance presented a planning method to ensure the dynamic disturbances do not exceed the capabilities of a vehicle and compromise its stability, while permitting a mobile manipulator to perform its task quickly.

This method is effective for systems in which there is a substantial friction between the vehicle and ground [17].

To avoid tumbling of a manipulator mounted on vehicle and carrying a heavy load, Fukuda et al proposed a center of gravity control method. In this method both the trajectory of the manipulator and the center of gravity of the manipulator are controlled [24].

Currently much work is going on to ensure stability of mobile robotic systems. Sugano, Huang and Kato describes the concepts of degree of stability and of valid stable regions based on the Zero Moment Point criterion [80]. The Zero Moment Point is a point on the ground where the resultant moment of the gravity, the inertial force of the mobile manipulator and the external force is zero. Papadopoulos and Rey suggested a new Force-Angle stability measure which is easy to compute and operates on both even and uneven terrain. The new tipover stability measure is sensitive to top heaviness and is applicable to dynamic systems subject to inertial loads and external forces [67].

In the case of rough terrain, it is preferable to use a legged vehicle rather than one with wheels. Although legged vehicles can negotiate very uneven terrain, the speed of the manipulator becomes very slow. Messuri and Klein developed a computer controlled algorithm to include the incorporation of a body accommodation feature and a body stabilization feature to allow greater vehicle maneuverability, particularly during rough-terrain locomotion [58]. They introduced the concept of energy stability margin. In case of a quadruped walking machine the stability algorithm has been developed by Davidson and Schweitzer [13]. More information on legged locomotion is cited in reference [29].

1.3.5 Real Time Simulation

With the advent of fast digital computers, real time simulation for complex systems has become very important. Real time simulation is needed for model-based control, simulator design for animation and detection of system failures. There are many factors that affect the speed of execution namely, the method of implementing dynamics, the step of integration, the numerical integration algorithm, the CPU, the source code, the compiler etc.

In the case of a multiprocessor system, a parallel processing scheme of manipulator dynamics computation is preferred [86]. McMillan used a supercomputer to simulate manipulator dynamics [56]. Distributed real time computation of manipulator dynamics has been reported by Abdalla et al. [1]. They simultaneously evaluated inertial, coupling and gravity terms. Frenton and Xi reported the use of algebra of rotation is more efficient than the use of homogeneous transformations [22]. However, in their work they used an iterative method for the dynamic simulation which is slower than a closed form solution for the dynamic simulation [3], [6].

1.4 Thesis Organization

The second chapter deals with the kinematic modeling of the forestry machine. This is required for dynamic modeling. The attachment of Denavit-Hartenberg frames and singularity analysis are described in this chapter. The dynamics of the forestry machine is formulated in the third chapter. At first a simplified model of three degrees-of-freedom (dof) is considered. Increasing complexity is added to the simplified model step-by-step, and equations of motion are derived for each case. In order to run simulations, validate the developed code, and obtain results various system parameters are needed. Some parameters (length, mass, thickness etc.) were obtained by direct measurements, weighing or industrial blueprints. But the inertial parameters, and the parameters related with stiffness and damping were found by various experiments. The design of various experiments and the corresponding results are described in Chapter 4. Chapter 5 deals with system analysis and design using inverse dynamics. Actuator valve sizing methodology and power calculations based on system dynamic model are discussed in this chapter. The implementation of forward dynamics and various techniques to minimize simulation time in order to achieve real time systems is discussed in Chapter 6. This chapter also describes the dynamic response of systems of varying complexity. Conclusions and future work are discussed in Chapter 7.

2. Kinematic Modeling

2.1 Introduction

Forestry machines are heavy duty mobile systems capable of working in harsh conditions. Although such machines usually carry articulated manipulators, they can not be considered as "robots" since they are not reprogrammable, multifunctional or autonomous. However kinematic and dynamic modeling methodologies that are routinely applied in robotics can be used to model such machines too. The mobile manipulator used as a test-bed in this thesis is shown in Figure 2.1. This machine was constructed for FERIC as a grapple loader and following structural modifications, it was converted to a harvester. The main links of the machine manipulator are shown in Figure 2.2. A schematic diagram of the machine is depicted in Figure 2.3.

Figure 2.1: Picture of the mobile manipulator.

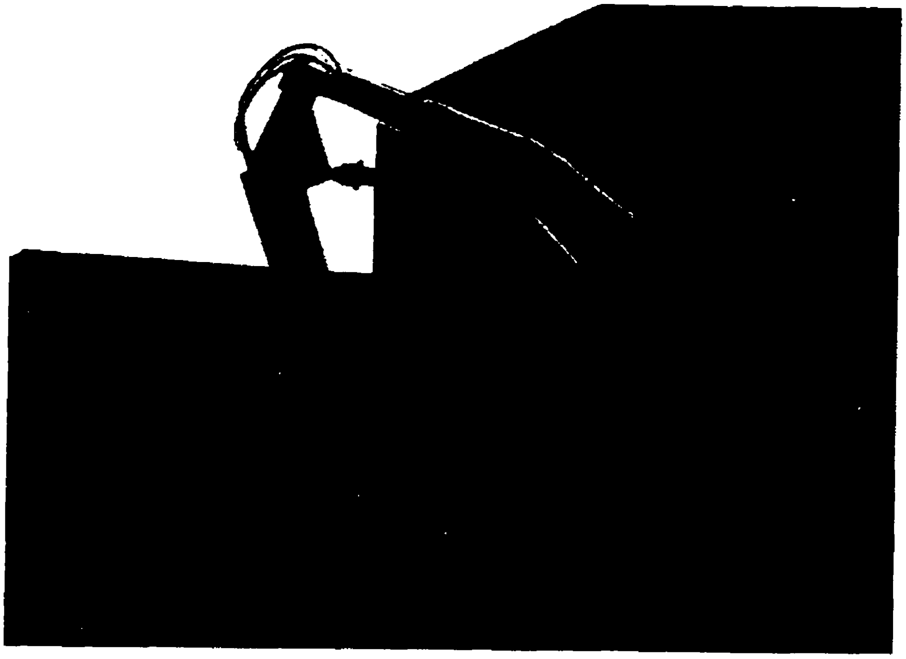


Figure 2.2: The machine's main links: Swing, Boom, Stick.

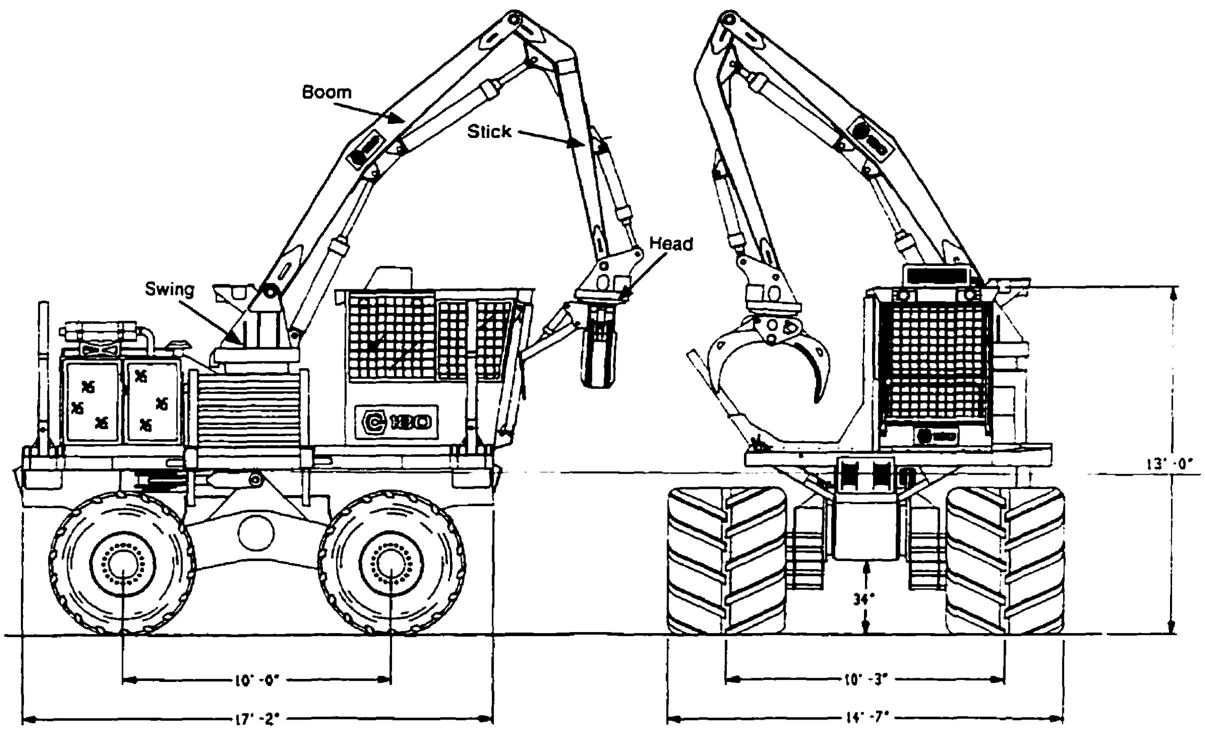


Figure 2.3: Schematic diagram of the machine.

The mobility of this system is due to its wheels, mounted at the end of the bogies. The bogies are interconnected in such a way that when one bogie rotates in a clock-wise direction the other one rotates in a counter clock-wise direction. This design minimizes tilt of the overall machine when one of the wheels is over a bump. The interconnection articulation of the bogies is shown in Figure 2.4.

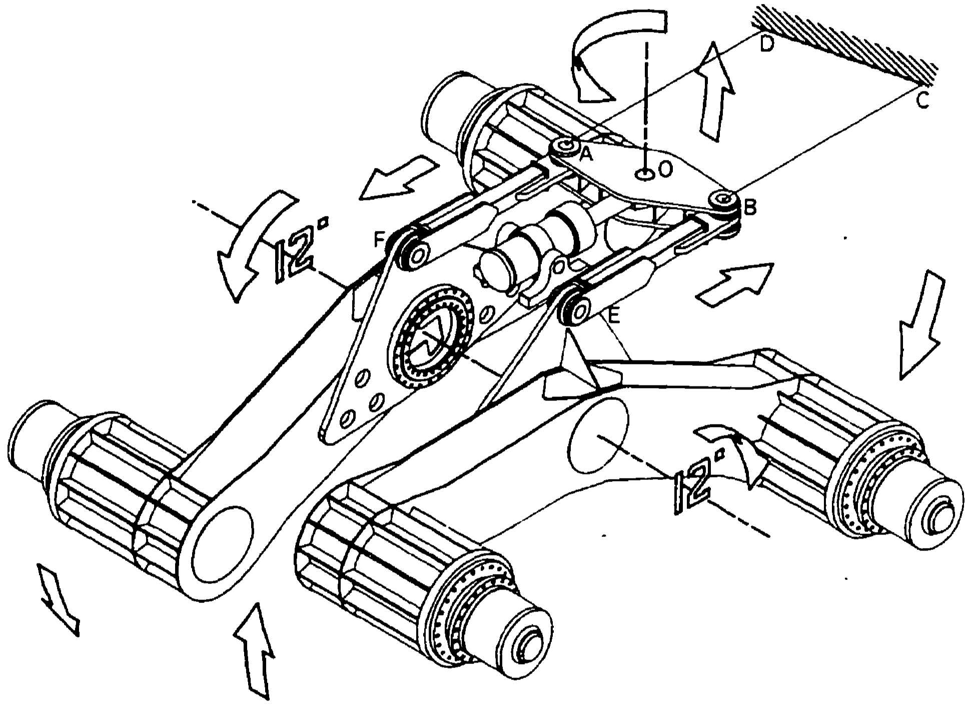


Figure 2.4: Diagram of the base.

In addition to counter-rotations, the bogies can rotate in the same direction with the help of a piston actuator. This feature helps the vehicle to climb a hill without tilting significantly the rectangular platform mounted on the bogies. Besides the articulated manipulator, the major components mounted on the platform include a cabin (encompasses the operator's seat, control panels, joystick etc.), a diesel engine, pumps, and a hydraulic reservoir. The manipulator consists of four major following parts: (1) swing, (2) boom, (3) stick, and (4) head as shown in Figure 2.3. The head and stick are connected through a pin (having two hinge joints perpendicular to each other).

The swing, boom and stick give a PUMA type configuration of the manipulator. The head is attached at the stick endpoint and is used cutting and processing trees. The detailed discussion of the head is beyond the scope of the thesis. The manipulator is hydraulically driven for high power output. The swing is driven by a gear motor while the boom and stick are moved with hydraulic cylinders. The joints between stick and pin, and pin and head are not actuated. The head behaves like a compound conical pendulum whose axes are perpendicular to each other, i.e. as gimbals.

2.2 Kinematics

The kinematics of the manipulator deals with the geometrical and time-based properties of motion. Hence it deals with the position, velocity and acceleration of the manipulator without regard to the forces/torques that cause them. The study of the kinematics focuses on the motion of the manipulator with respect to a fixed co-ordinate system. The complete kinematic and dynamic modeling of the manipulator has been done by step by step. At first, only three links were considered. These include the swing, boom and stick and result in a system with three dof. In the second step, pin and head were attached at the end of the stick, resulting in a five dof system. Next the stiffness and damping of the tires were introduced. Due to the tires the machine can bounce, pitch and roll. The yaw effect is neglected. The complete model has eight-degrees-of-freedom. The details will be discussed later.

2.2.1 Base Kinematics

In this section, the base kinematic equations are developed. The base consists of a platform, on which a piston and a set of connecting links are mounted, as shown in Figure 2.4. For a fixed piston position, when one bogie rotates clockwise the other one rotates counter clockwise direction. As the piston moves, both the bogies move in the same direction by the same angle. The complete base configuration can be obtained with two linear gauges (translation sensors), mounted at a certain distance from the platform. The schematic diagram is shown in Figure 2.5, where the piston is in home position and the bogies are not tilted and at this stage the four wheel axes are in same plane, see Figure 2.4. In this configuration link AB coincides with QP, and side links AF and GE coincide with HN and PM respectively, see Figure 2.5. In such case, the two linear gauge readings are equal. If only the piston moves, the length of both gauges will be changed by the same amount. When the bogies rotate as they go over a bump, the two gauges will indicate different readings. In this section our main objective is to obtain the two bogie angles from the two gauge readings.

The two linear gauge readings are denoted by d_1 ($\overline{AD}$) and d_2 ($\overline{BC}$). The first step is to find the absolute position (d) and angle (θ) from these two parameters (see Figure 2.5).

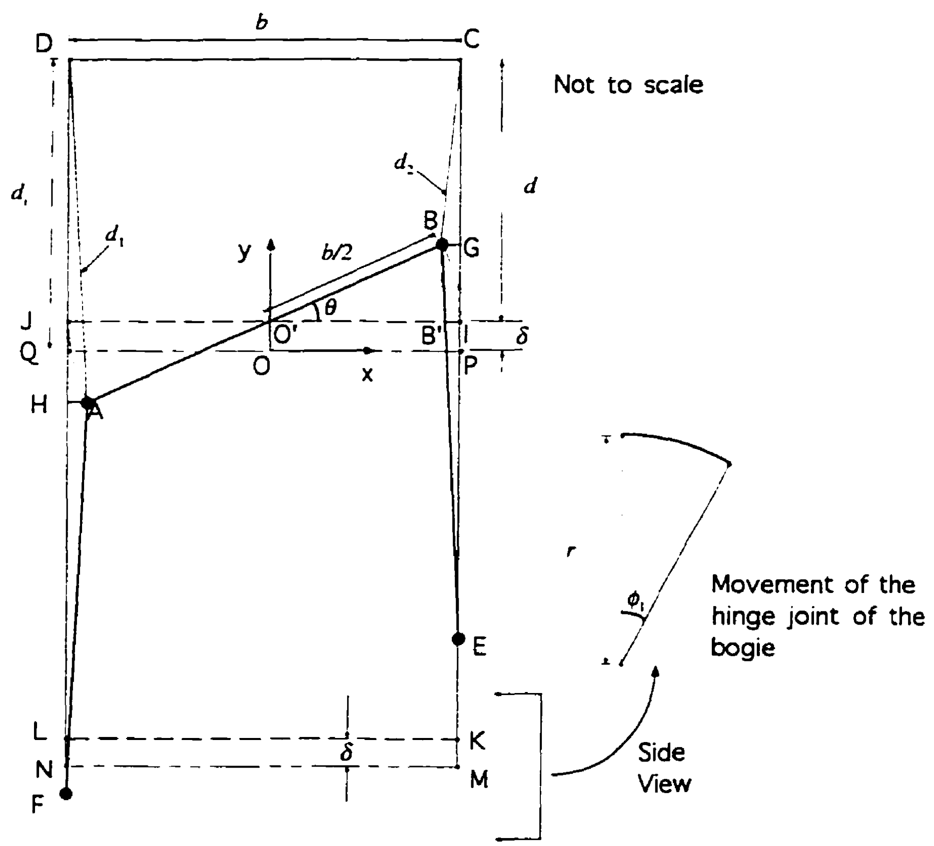


Figure 2.5: Schematic Diagram for the Base.

With reference to Figure 2.5 we get from triangle O'BB'

$$BB' = IG = \frac{b}{2}\sin\theta \tag{2.1}$$

$$O'B' = \frac{b}{2}\cos\theta \tag{2.2}$$

$$\begin{aligned} BG = B'I = O'I - O'B' = OP - O'B' &= \frac{b}{2} - \frac{b}{2}\cos\theta \\ &= \frac{b}{2}(1-\cos\theta) \end{aligned} \tag{2.3}$$

$$GC = IC - IG = IC - BB' = d - \frac{b}{2}\sin\theta \tag{2.4}$$

Again from triangle BCG we get,

$$BC^2 = BG^2 + GC^2 \tag{2.5}$$

$$d_2^{\,2} = \frac{b^2}{4}(1-\cos\theta)^2 + \left(d - \frac{b}{2}\sin\theta\right)^2 \tag{2.6}$$

Similarly from triangle ADH we get

$$AD^2 = AH^2 + HD^2 \tag{2.7}$$

Similarly from the trigonometry we obtain

$$d_1^{\,2} = \frac{b^2}{4}(1-\cos\theta)^2 + \left(d + \frac{b}{2}\sin\theta\right)^2 \tag{2.8}$$

Subtracting Eq. (2.6) from Eq. (2.8) we get,

$$\sin\theta = \frac{d_1^{\,2} - d_2^{\,2}}{2db}, \text{ (take the acute angle)} \tag{2.9}$$

Therefore we have,

$$\cos\theta = \sqrt{1 - \left(\frac{d_1^{\,2} - d_2^{\,2}}{2db}\right)^2} \tag{2.10}$$

Using this value in Eq. (2.8) we get,

$$d_1^2 = \frac{b^2}{4}\left[1-\sqrt{1-\left(\frac{d_1^2-d_2^2}{2db}\right)^2}\right]^2 + \left[d+\frac{b}{2}\left(\frac{d_1^2-d_2^2}{2db}\right)^2\right]^2 \tag{2.11}$$

Rearranging the terms,

$$\sqrt{1-\left(\frac{d_1^2-d_2^2}{2db}\right)^2} = \frac{d_1^2+d_2^2-b^2-2d^2}{b^2} \tag{2.12}$$

Squaring, cross multiplying and rearranging the terms we obtain,

$$16d^6-16\left(d_1^2+d_2^2-b^2\right)d^4+4\left[\left(d_1^2+d_2^2-b^2\right)^2-b^4\right]d^2+b^2\left(d_1^2-d_2^2\right)^2=0 \tag{2.13}$$

Using, $x=d^2$, we get,

$$16x^3-16\left(d_1^2+d_2^2-b^2\right)x^2+4\left[\left(d_1^2+d_2^2-b^2\right)^2-b^4\right]x+b^2\left(d_1^2-d_2^2\right)^2=0 \tag{2.14}$$

In Eq. (2.14), b is a constant (by measuring the actual distance between the bogies we get, $b = 0.946\ m$), and d_1 and d_2 are the independent readings from the linear gauges. By solving the equations numerically, we get three values for the variable x of which two values are complex and one value is real. The positive square root of this real value is the desired value for d.

Special cases result if, $d_1 = d_2 = d^*$. From Eq. (2.14) we have,

$$16x^3-16\left(2d^{*2}-b^2\right)x^2+4\left[\left(2d^{*2}-b^2\right)^2-b^4\right]x=0 \tag{2.15}$$

Case 1:

$$\Rightarrow \quad x=0, \Rightarrow \quad d=0$$

In such case, A & D and B & C coincide, see Figure 2.5.

Case 2:

$$\Rightarrow \quad 4x^2-4\left(2d^{*2}-b^2\right)x+\left(2d^{*2}-b^2\right)^2-b^4=0 \tag{2.16}$$

Simplifying, we get, $d = d^*$. To conclude, using Eq. (2.9) and Eq. (2.14) we can compute d and θ using the measurements d_1 and d_2. The next step is to compute from d and θ the bogie angles ϕ_1 and ϕ_2.

A body-fixed coordinate frame is attached at the center of the platform. When the piston is in its home position and the two bogies are not rotated with respect to the horizontal platform, the Z axis is aligned with the gravity vector. At this position the two linear gauges will show the same initial reading d_i. The initial position of the hinge points of the bogies and the two side hinge joints of the base are MN and PQ respectively, see Figure 2.4 and Figure 2.5. Now due to piston actuation link AB moves by an amount ($\delta = d_i - d$). The value for d can be found from Eq. (2.14). The position of the bogies and link AB following piston motion are described by KL and IJ respectively see Figure 2.5.

When one tire of the vehicle is on a bump both bogies change angles. In such case the final bogie angles are ϕ_1 and ϕ_2. The final position of the bogies and base are EF and AB respectively.

Coordinates of A:

$$x_a = -\frac{b}{2}\cos\theta \tag{2.17}$$

$$y_a = \delta - \frac{b}{2}\sin\theta \tag{2.18}$$

$$z_a = 0 \tag{2.19}$$

Coordinates of B:

$$x_b = \frac{b}{2}\cos\theta \tag{2.20}$$

$$y_b = \delta + \frac{b}{2}\sin\theta \tag{2.21}$$

$$z_b = 0 \tag{2.22}$$

Coordinates of E:

$$x_e = \frac{b}{2} \tag{2.23}$$

$$y_e = -l + \delta + r\sin\phi_1 \tag{2.24}$$

$$z_e = -r(1 - \cos\phi_1) \tag{2.25}$$

Coordinates of F:

$$x_f = -\frac{b}{2} \tag{2.26}$$

$$y_f = -l + \delta + r\sin\phi_2 \tag{2.27}$$

$$z_f = -r(1 - \cos\phi_2) \tag{2.28}$$

$$AF = BE = \sqrt{(x_b - x_e)^2 + (y_b - y_e)^2 + (z_b - z_e)^2} = l \tag{2.29}$$

Squaring both sides, substituting and rearranging terms we get,

$$\left(\frac{b}{2} - \frac{b}{2}\cos\theta\right)^2 + \left(-l - \frac{b}{2}\sin\theta + r\sin\phi_1\right)^2 + r^2(1 - \cos\phi_1)^2 = l^2 \tag{2.30}$$

Introducing a dummy parameter,

$$P_1 = -l - \frac{b}{2}\sin\theta \tag{2.31}$$

Eq. (2.30) becomes,

$$\left(\frac{b}{2} - \frac{b}{2}\cos\theta\right)^2 + (P_1 + r\sin\phi_1)^2 + r^2(1 - \cos\phi_1)^2 = l^2 \tag{2.32}$$

Rearranging terms we get,

$$\frac{\left(\frac{b}{2} - \frac{b}{2}\cos\theta\right)^2 - l^2 + P_1^2 + 2r^2}{2r^2} + \frac{P_1}{r}\sin\phi_1 = \cos\phi_1 \tag{2.33}$$

Introducing additional dummy parameters, we write,

$$Q_1 = \frac{\left(\frac{b}{2} - \frac{b}{2}\cos\theta\right)^2 - l^2 + P_1^2 + 2r^2}{2r^2} \tag{2.34}$$

$$S_1 = \frac{P_1}{r} \tag{2.35}$$

and

$$k_1 = \sin\phi_1 \tag{2.36}$$

Eq. (2.33) reduces to the following,

$$Q_1 + S_1 k_1 = \sqrt{1 - k_1^2} \tag{2.37}$$

Squaring and rearranging,

$$\left(S_1^2 + 1\right)k_1^2 + 2Q_1 S_1 k_1 + \left(Q_1^2 - 1\right) = 0 \tag{2.38}$$

Eq. (2.38) results in two roots, but as ϕ_1 is always an acute angle, one solution is obtained only. For link AF, see Figure 2.5,

$$AF = \sqrt{\left(x_a - x_f\right)^2 + \left(y_a - y_f\right)^2 + \left(z_a - z_f\right)^2} = l \tag{2.39}$$

Squaring both sides, substituting and rearranging terms we get,

$$\left(\frac{b}{2} - \frac{b}{2}\cos\theta\right)^2 + \left(-l + \frac{b}{2}\sin\theta + r\sin\phi_2\right)^2 + r^2\left(1 - \cos\phi_2\right)^2 = l^2 \tag{2.40}$$

If,

$$P_2 = -l + \frac{b}{2}\sin\theta \tag{2.41}$$

Eq. (2.40) becomes,

$$\left(\frac{b}{2} - \frac{b}{2}\cos\theta\right)^2 + \left(P_2 + r\sin\phi_2\right)^2 + r^2\left(1 - \cos\phi_2\right)^2 = l^2 \tag{2.42}$$

Rearranging terms,

$$\frac{\left(\frac{b}{2} - \frac{b}{2}\cos\theta\right)^2 - l^2 + P_2^2 + 2r^2}{2r^2} + \frac{P_2}{r}\sin\phi_2 = \cos\phi_2 \tag{2.43}$$

By defining,

$$Q_2 = \frac{\left(\frac{b}{2} - \frac{b}{2}\cos\theta\right)^2 - l^2 + P_2^2 + 2r^2}{2r^2} \tag{2.44}$$

$$S_2 = \frac{P_2}{r} \tag{2.45}$$

and

$$k_2 = \sin\phi_2 \tag{2.46}$$

Eq. (2.43) reduces to the following,

$$Q_2 + S_2 k_2 = \sqrt{1 - k_2^2} \tag{2.47}$$

Squaring and rearranging,

$$\left(S_2^2 + 1\right)k_2^2 + 2Q_2 S_2 k_2 + \left(Q_2^2 - 1\right) = 0 \tag{2.48}$$

Similarly as ϕ_2 is also always an acute angle, Eqs. (2.48) and (2.46) will yield one solution only. This completes the procedure for finding ϕ_1, and ϕ_2, from gauge measurements d_1 and d_2. Next, manipulator kinematics are studied.

2.2.2 Denavit-Hartenberg Parameters

The link frames used for the swing boom, and stick are shown in Figure 2.6.

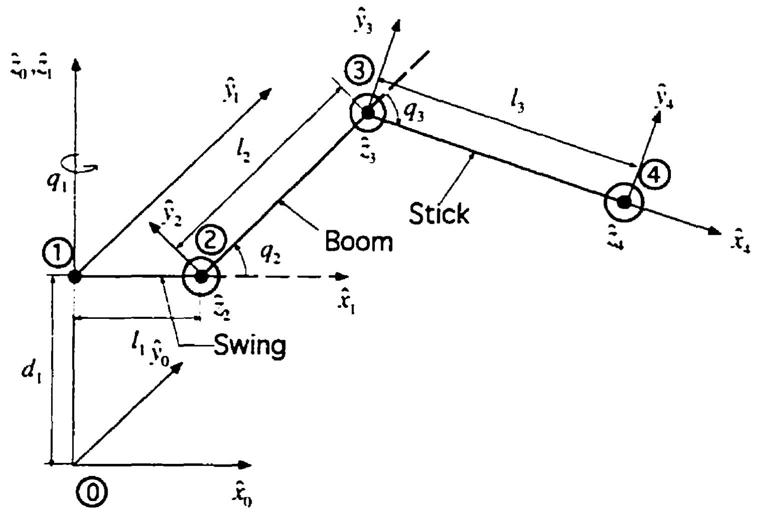


Figure 2.6: Link Frame Attachment to the 3 dof System.

According to the Denavit-Hartenberg notation, the manipulator is described kinematically by four parameters for each link. The link frames are attached as described by Craig [11]. The world or inertial frame is represented by $\hat{x}_0\hat{y}_0\hat{z}_0$ axes. It can be taken anywhere as dynamics of the manipulator will not be dependent on the position of the world frame. The corresponding table of D-H Parameter is shown in Table 2.1.

Table 2.1: D-H Parameters for the 3 dof System.

i	α_{i-1}	a_{i-1}	d_i	q_i
1	0	0	d_1=0.786m	q_1
2	$\pi/2$	l_1=0.153m	0	q_2
3	0	l_2=4.118m	0	q_3
4	0	l_3=4.229m	0	-

Lengths l_2 and l_3 are the boom and stick lengths respectively, while l_1 is defined in Figure 2.6. The distance from world frame to swing frame along $\hat{z}_0$ axis is denoted by d_1, and q_i is the joint variable of *ith* joint. The general form of the transformation matrices can be obtained by the following formula

$$ {}^{i-1}T_i = \begin{bmatrix} cq_i & -sq_i & 0 & a_{i-1} \\ sq_i c\alpha_{i-1} & cq_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1}d_i \\ sq_i s\alpha_{i-1} & cq_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.49} $$

where cq_i and sq_i are the cosine and sine of the angle q_i, respectively. Using Eq. (2.49) and Table 2.1 the transformation matrices from world to swing, swing to boom and boom to stick are found as below

$$ {}^0T_1 = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^1T_2 = \begin{bmatrix} c_2 & -s_2 & 0 & l_1 \\ 0 & 0 & -1 & 0 \\ s_2 & c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^2T_3 = \begin{bmatrix} c_3 & -s_3 & 0 & l_2 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.50} $$

2.2.3 Forward Kinematics

In forward kinematics study, the end-effector position and orientation is found out as a function of the joint variables. The transformation matrix from frame 3 to frame 4 is given by

$$^3T_4 = \begin{bmatrix} 1 & 0 & 0 & l_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.51}$$

The transformation matrix from world frame to end-effector frame is obtained by

$$^0T_4 = {}^0T_1\,{}^1T_2\,{}^2T_3\,{}^3T_4 = \begin{bmatrix} & & & x \\ & {}^0R_4 & & y \\ & & & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.52}$$

where $^0R_4$ is the rotation matrix from world frame to end-effector frame and x, y and z are the co-ordinates of the origin of the end-effector frame (tip) with respect to the world frame. After trigonometric simplifications we obtain

$$x = c_1\left[l_1 + l_2c_2 + l_3c_{23}\right] \tag{2.53}$$

$$y = s_1\left[l_1 + l_2c_2 + l_3c_{23}\right] \tag{2.54}$$

$$z = d_1 + l_2s_2 + l_3s_{23} \tag{2.55}$$

where

$$c_{23} = \cos(q_2 + q_3) \tag{2.56}$$

$$s_{23} = \sin(q_2 + q_3) \tag{2.57}$$

2.2.4 Inverse Kinematics

In an inverse kinematics study, we compute joint space angles from Cartesian space co-ordinates. It is not as straightforward as forward kinematics, since there is a possibility of multiple solutions. In some cases closed form solutions do not exist.

From Eqs. (2.53) and (2.54), we get the following relation:

$$q_1 = \tan^{-1}\left(\frac{y}{x}\right) \tag{2.58}$$

Two solutions exist, if only the ratio $\frac{y}{x}$ is given. However if x and y are given separately, the function atan2, results in one solution only [11].

$$q_1 = \operatorname{atan2}(y, x) \tag{2.59}$$

In our manipulator the joint limits are such that inverse kinematics can be solved in a faster way. For example the stick angle q_3 can never be positive, as the stick is driven by a piston. Our customized faster inverse kinematics is presented below.

From Eqs. (2.53) and (2.54), we write

$$u = \frac{x}{c_1} - l_1 = l_2 c_2 + l_3 c_{23} \tag{2.60}$$

$$v = z - d_1 = l_2 s_2 + l_3 s_{23} \tag{2.61}$$

Squaring and adding we get

$$u^2 + v^2 = l_2^2 + l_3^2 + 2 l_2 l_3 c_3 \tag{2.62}$$

Solving for q_3

$$q_3 = -\cos^{-1}\left(\frac{u^2 + v^2 - l_2^2 - l_3^2}{2 l_2 l_3}\right) \tag{2.63}$$

The stick angle can only be negative. So elbow-down solution is discarded in the program. To solve for the boom angle q_2, two dummy variables k_1 and k_2 are introduced.

$$k_1 = l_2 + l_3 c_3 \tag{2.64}$$

$$k_2 = l_3 s_3 \tag{2.65}$$

After expanding and rearranging the terms of Eqs. (2.60) and (2.61), we get the following expressions for u and v

$$u = k_1 c_2 - k_2 s_2 \tag{2.66}$$

$$v = k_1 s_2 + k_2 c_2 \tag{2.67}$$

Solving for q_2

$$q_2 = \pm \cos^{-1} \frac{u k_1 + v k_2}{k_1^2 + k_2^2} \tag{2.68}$$

By means of a forward kinematics check, we can choose the proper sign of q_2. Another way of computing angle q_2 is to introduce two new variables r and γ as follows

$$r = \sqrt{k_1^2 + k_2^2} \tag{2.69}$$

$$\gamma = a\tan 2(k_2, k_1) \tag{2.70}$$

then

$$k_1 = r\cos(\gamma) \tag{2.71}$$

$$k_2 = r\sin(\gamma) \tag{2.72}$$

From the Eqs. (2.66) and (2.67),we get

$$\frac{v}{r} = \cos(\gamma)\sin(q_2) - \sin(\gamma)\cos(q_2) = \sin(\gamma + q_2) \tag{2.73}$$

$$\frac{u}{r} = \cos(\gamma)\cos(q_2) - \sin(\gamma)\sin(q_2) = \cos(\gamma + q_2) \tag{2.74}$$

$$\gamma + q_2 = \text{atan2}\left(\frac{v}{r}, \frac{u}{r}\right) \tag{2.75}$$

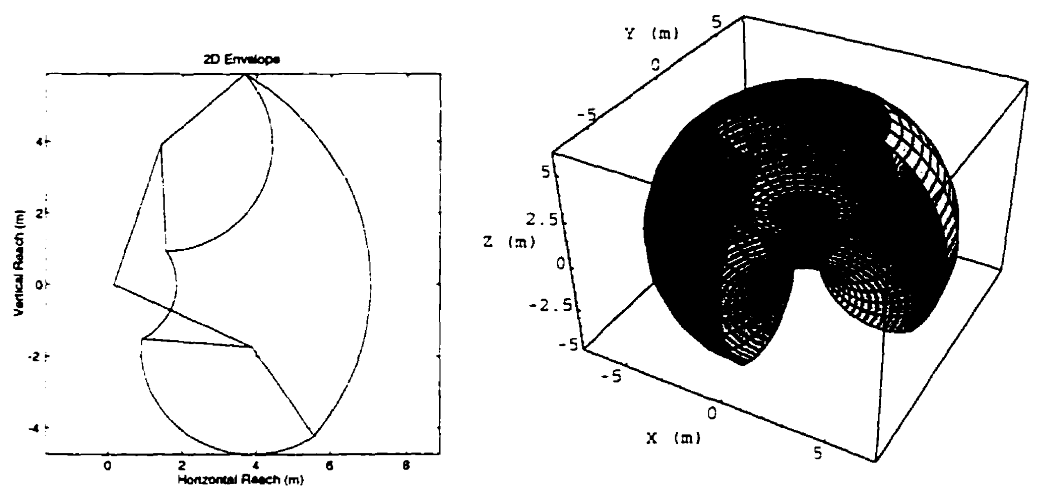
From Eq. (2.70) we get

$$q_2 = \text{atan2}\left(\frac{v}{r}, \frac{u}{r}\right) - \text{atan2}(k_2, k_1) \tag{2.76}$$

Timing program execution has revealed that using of Eq. (2.68) and a forward kinematics check is faster than that using Eq. (2.76).

2.2.5 Work Space Envelope

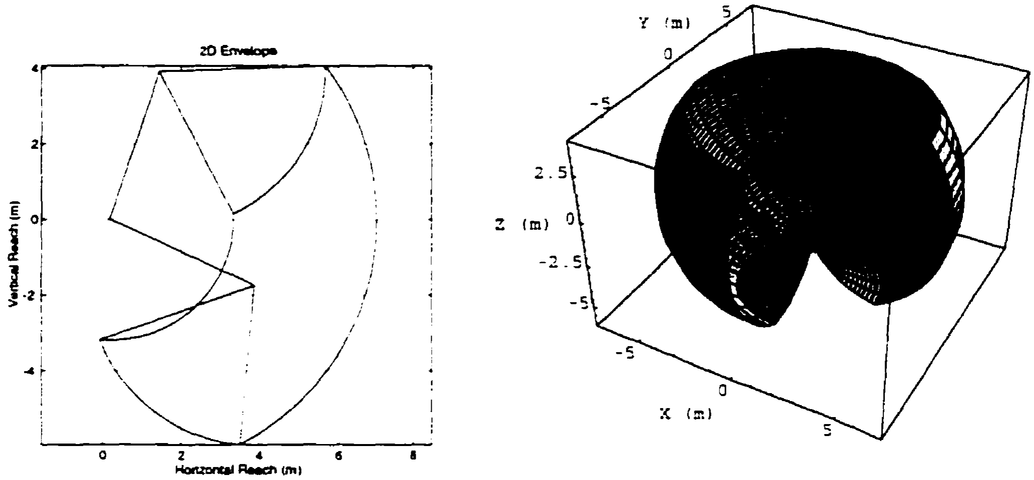
To generate the workspace of a manipulator we need to know how the frames are attached, see Figure 2.6. Here the 3 dof system is considered. The outmost link the stick, and then boom are rotated from their minimum to maximum joint limits with respect the axes of rotation (z_3 and z_2 for stick and boom respectively see Figure 2.6). The minimum and the maximum cylinder lengths for the boom and stick are obtained from the blue print specifications, and those values are used to get the respective minimum and the maximum joint limits. Figure 2.7, (a) shows the workspace envelope in 2D. The section of the 2D envelope is rotated with respect to the swing axis (z_1 see Figure 2.6) to have the 3D envelope. The sections of the 3D envelope are shown in Figure 2.7, (b).



(a) (b)

Figure 2.7: Work Space With Short Stick.

In order to increase the workspace and improve its shape close to the ground, the stick has been made slightly longer and the hinge position has been changed. The workspace for the new stick is given below in Figure 2.8. Note that although the stick is driven by the same cylinder, the joint limits are now different.



(a) (b)

Figure 2.8: Work Space With Long Stick.

2.2.6 Jacobian Matrix

Taking the first time derivative of Eq. (2.53) through Eq. (2.55) we get the components of the Cartesian tip velocities as written below:

$$\dot{x} = -\left[(l_1 + l_2c_2 + l_3c_{23})s_1\dot{q}_1 + (l_2s_2 + l_3s_{23})c_1\dot{q}_2 + l_3c_1s_{23}\dot{q}_3\right] \tag{2.77}$$

$$\dot{y} = \left[(l_1 + l_2c_2 + l_3c_{23})c_1\dot{q}_1 - (l_2s_2 + l_3s_{23})s_1\dot{q}_2 - l_3s_1s_{23}\dot{q}_3\right] \tag{2.78}$$

$$\dot{z} = (l_2c_2 + l_3c_{23})\dot{q}_2 + l_3c_{23}\dot{q}_3 \tag{2.79}$$

The mapping from joint space to Cartesian space velocities can be expressed by a Jacobian matrix J.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = J \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} \tag{2.80}$$

where

$$J = \begin{bmatrix} -(l_1 + l_2c_2 + l_3c_{23})s_1 & -(l_2s_2 + l_3s_{23})c_1 & -l_3c_1s_{23} \\ (l_1 + l_2c_2 + l_3c_{23})c_1 & -(l_2s_2 + l_3s_{23})s_1 & -l_3s_1s_{23} \\ 0 & (l_2c_2 + l_3c_{23}) & l_3c_{23} \end{bmatrix} \tag{2.81}$$

2.2.7 Singularities

A singularity occurs when the determinant of the Jacobian matrix is zero. Usually it arises when the manipulator is fully stretched out or folded back on itself at the boundary of the workspace. This phenomenon is called workspace boundary singularity as described in Eq.(2.83), and the manipulator looses one degree of freedom. Another singularity occurs when the kinematic structures exhibit degeneracy i.e. rotation of one or more joints does not affect the movement of the tip in Cartesian workspace. This phenomenon is called workspace interior singularity as described in Eq.(2.84).

Setting the determinant of the Jacobian matrix equals to zero we get the following equation

$$l_2l_3[l_1 + l_2c_2 + l_3c_{23}]s_3 = 0 \tag{2.82}$$

Case 1:

$$\Rightarrow q_3 = 0, \text{ as } l_2, l_3 \neq 0 \tag{2.83}$$

Case 2:

$$\Rightarrow q_3 = \cos^{-1}\left[\frac{l_1 + l_2c_2}{-l_3}\right] - q_2 \tag{2.84}$$

If q_3 is zero, the manipulator is at a workspace boundary singularity and it has two dof only. Eq. (2.84) describes the relationship between q_2 and q_3 for a workspace interior singularity. Figure 2.9 shows the relationship of boom and stick angle to achieve the above mentioned singular position. In the case of our experimental machine the joint limits are such that the manipulator can not reach any of the above singular configurations.

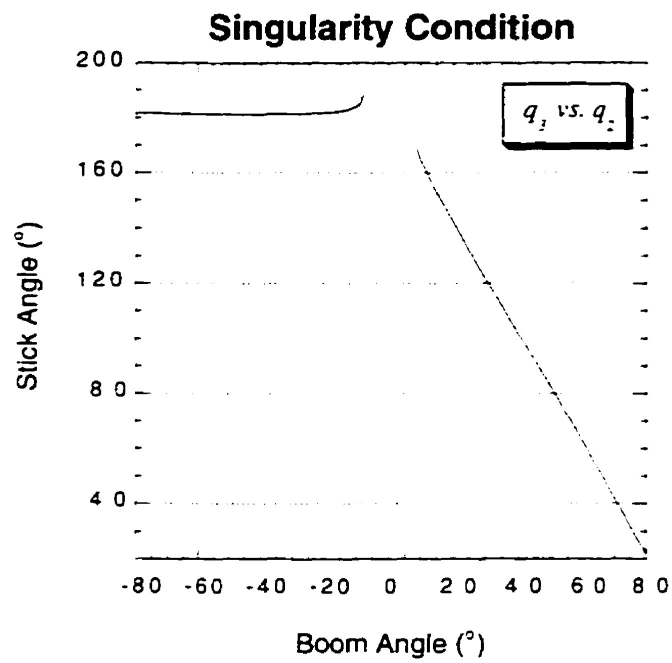


Figure 2.9: Relationship of Boom & Stick Angles for Interior Singularity.

3. Dynamic Modeling

In the study of system dynamics, we consider the forces and/or torques required to cause motion of manipulator. A number of methods are available to formulate manipulator dynamics, including the iterative Newton-Euler dynamic formulation, the Lagrangian formulation, Kane's method, and others. For the needs of this work, the iterative Newton-Euler dynamic formulation was chosen because it is easy to implement in the form of computer code, and it requires a smaller number of computations. In general, in this method kinematic quantities are calculated with outward computations starting from the base and ending at the tip, while actuator forces and torques are computed with inward computations. Gravity forces are included by simply assuming that the base frame is accelerated upwards with an acceleration equal to that of gravity.

3.1 Modeling for a 3 dof system

As reported in the literature survey, a number of methods are available to formulate manipulator dynamics. Here the iterative Newton-Euler dynamic formulation is chosen for the advantages as described in the previous section. In this method, we compute outwards velocities and accelerations, and inwards forces and/or torques). The detailed algorithm and notations are available in [11].

The rotation matrices can be found from first three rows and columns of transformation matrices of Eq. (2.46).

$$ {}^{0}R_1 = \begin{bmatrix} c_1 & -s_1 & 0 \\ s_1 & c_1 & 0 \\ 0 & 0 & 1 \end{bmatrix},\ {}^{1}R_2 = \begin{bmatrix} c_2 & -s_2 & 0 \\ 0 & 0 & -1 \\ s_2 & c_2 & 0 \end{bmatrix},\ {}^{2}R_3 = \begin{bmatrix} c_3 & -s_3 & 0 \\ s_3 & c_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.1} $$

The link parameters are given in vector form as below, see Figure 2.6.

$$
{}^{0}p_{1}^{0}=\begin{bmatrix}0\\0\\d_1\end{bmatrix},\ {}^{1}p_{c_1}^{1}=\begin{bmatrix}x_1\\y_1\\z_1\end{bmatrix},\ {}^{1}p_{2}^{1}=\begin{bmatrix}l_1\\0\\0\end{bmatrix},\ {}^{2}p_{c_2}^{2}=\begin{bmatrix}x_2\\y_2\\z_2\end{bmatrix},
$$

$$
{}^{2}p_{3}^{2}=\begin{bmatrix}l_2\\0\\0\end{bmatrix},\ {}^{3}p_{c_3}^{3}=\begin{bmatrix}x_3\\y_3\\z_3\end{bmatrix},\ {}^{3}p_{4}^{3}=\begin{bmatrix}l_3\\0\\0\end{bmatrix} \tag{3.2}
$$

Where ${}^{a}p_{c}^{b}$ denotes the position vector of origin of frame c, with respect to point b and expressed in frame a. Assuming the fixed base system, i.e., the swing is placed on a rigid platform, the initial conditions are given by:

$$
{}^{0}\dot{v}_0=\begin{bmatrix}0\\0\\g\end{bmatrix},\ {}^{0}\omega_0=\begin{bmatrix}0\\0\\0\end{bmatrix},{}^{0}\dot{\omega}_0=\begin{bmatrix}0\\0\\0\end{bmatrix} \tag{3.3}
$$

Outward iteration for kinematics, starting from swing to stick:

For swing (link 1),

$$
{}^{1}\omega_1=\begin{bmatrix}0\\0\\\dot{q}_1\end{bmatrix},{}^{1}\dot{\omega}_1=\begin{bmatrix}0\\0\\\ddot{q}_1\end{bmatrix} \tag{3.4}
$$

$$
{}^{1}\dot{v}_1={}^{1}R_0\left[{}^{0}\dot{v}_0+{}^{0}\dot{\omega}_0^{\times}{}^{0}p_1^0+{}^{0}\omega_0^{\times}\left({}^{0}\omega_0^{\times}{}^{0}p_1^0\right)\right]={}^{1}R_0\,{}^{0}\dot{v}_0=\begin{bmatrix}0\\0\\g\end{bmatrix} \tag{3.5}
$$

where the superscript $^{\times}$ converts a vector to the corresponding skew-symmetric cross-product matrix.

$$
{}^{1}\dot{v}_{c_1}={}^{1}\dot{v}_1+{}^{1}\dot{\omega}_1^{\times}{}^{1}p_{c_1}^{1}+{}^{1}\omega_1^{\times}\left({}^{1}\omega_1^{\times}{}^{1}p_{c_1}^{1}\right) \tag{3.6}
$$

Substituting

$$
\begin{aligned}
{}^{1}\dot{v}_{c_1} &= \begin{bmatrix}0\\0\\g\end{bmatrix} + \begin{bmatrix}0\\0\\\ddot{q}_1\end{bmatrix}^{\times}\begin{bmatrix}x_1\\y_1\\z_1\end{bmatrix} + \begin{bmatrix}0\\0\\\dot{q}_1\end{bmatrix}^{\times}\left(\begin{bmatrix}0\\0\\\dot{q}_1\end{bmatrix}^{\times}\begin{bmatrix}x_1\\y_1\\z_1\end{bmatrix}\right)\\
&= \begin{bmatrix}0\\0\\g\end{bmatrix} + \begin{bmatrix}-y_1\ddot{q}_1\\x_1\ddot{q}_1\\0\end{bmatrix} + \begin{bmatrix}0\\0\\\dot{q}_1\end{bmatrix}^{\times}\begin{bmatrix}-y_1\dot{q}_1\\x_1\dot{q}_1\\0\end{bmatrix}\\
&= \begin{bmatrix}-y_1\ddot{q}_1 - x_1\dot{q}_1^2\\x_1\ddot{q}_1 - y_1\dot{q}_1^2\\g\end{bmatrix}
\end{aligned}
\tag{3.7}
$$

For boom (link 2),

$$
{}^{2}\omega_2 = {}^{2}R_1\,{}^{1}\omega_1 + \dot{q}_2\,{}^{2}\hat{z}_2 \tag{3.8}
$$

Substituting:

$$
{}^{2}\omega_2 = \begin{bmatrix}c_2 & -s_2 & 0\\0 & 0 & -1\\s_2 & c_2 & 0\end{bmatrix}^{-1\times}\begin{bmatrix}0\\0\\\dot{q}_1\end{bmatrix} + \begin{bmatrix}0\\0\\\dot{q}_2\end{bmatrix} = \begin{bmatrix}s_2\dot{q}_1\\c_2\dot{q}_1\\\dot{q}_2\end{bmatrix} \tag{3.9}
$$

$$
{}^{2}\dot{\omega}_2 = \frac{d}{dt}\left({}^{2}\omega_2\right) = \begin{bmatrix}s_2\ddot{q}_1 + c_2\dot{q}_1\dot{q}_2\\c_2\ddot{q}_1 - s_2\dot{q}_1\dot{q}_2\\\ddot{q}_2\end{bmatrix} \tag{3.10}
$$

$$
{}^{2}\dot{v}_2 = {}^{2}R_1\left[{}^{1}\dot{v}_1 + {}^{1}\dot{\omega}_1^{\times}\,{}^{1}p_2^1 + {}^{1}\omega_1^{\times}\left({}^{1}\omega_1^{\times}\,{}^{1}p_2^1\right)\right] \tag{3.11}
$$

Substituting:

$$
\begin{aligned}
{}^{2}\dot{v}_2 &= \begin{bmatrix}c_2 & -s_2 & 0\\0 & 0 & -1\\s_2 & c_2 & 0\end{bmatrix}^{-1}\left[\begin{bmatrix}0\\0\\g\end{bmatrix} + \begin{bmatrix}0\\0\\\ddot{q}_1\end{bmatrix}^{\times}\begin{bmatrix}l_1\\0\\0\end{bmatrix} + \begin{bmatrix}0\\0\\\dot{q}_1\end{bmatrix}^{\times}\left(\begin{bmatrix}0\\0\\\dot{q}_1\end{bmatrix}^{\times}\begin{bmatrix}l_1\\0\\0\end{bmatrix}\right)\right]\\
&= \begin{bmatrix}-c_2 l_1\dot{q}_1^2 + s_2 g\\s_2 l_1\dot{q}_1^2 + c_2 g\\-l_1\ddot{q}_1\end{bmatrix}
\end{aligned}
\tag{3.12}
$$

$$
{}^{2}\dot{v}_{c_2} = {}^{2}\dot{v}_2 + {}^{2}\dot{\omega}_2^{\times}\,{}^{2}p_{c_2}^2 + {}^{2}\omega_2^{\times}\left({}^{2}\omega_2^{\times}\,{}^{2}p_{c_2}^2\right) \tag{3.13}
$$

Further substitutions are not shown, they are done in Mathematica systematically. A sample systematic programming is shown in Appendix B.

For stick (link 3),

$$^{3}\omega_3 = {}^{3}R_2\,{}^{2}\omega_2 + \dot{q}_3\,{}^{3}\hat{z}_3 \tag{3.14}$$

$$^{3}\dot{\omega}_3 = \frac{d}{dt}\left({}^{3}\omega_3\right) \tag{3.15}$$

$$^{3}\dot{v}_3 = {}^{3}R_2\left[{}^{2}\dot{v}_2 + {}^{2}\dot{\omega}_2{}^{\times}\,{}^{2}p_3^2 + {}^{2}\omega_2{}^{\times}\left({}^{2}\omega_2{}^{\times}\,{}^{2}p_3^2\right)\right] \tag{3.16}$$

$$^{3}\dot{v}_{c_3} = {}^{3}\dot{v}_3 + {}^{3}\dot{\omega}_3{}^{\times}\,{}^{3}p_{c_3}^3 + {}^{3}\omega_3{}^{\times}\left({}^{3}\omega_3{}^{\times}\,{}^{3}p_{c_3}^3\right) \tag{3.17}$$

Newton's Equation:

$$^{3}F_3 = m_3\,{}^{3}\dot{v}_{c_3} = m_3\left[{}^{3}\dot{v}_3 + {}^{3}\dot{\omega}_3{}^{\times}\,{}^{3}p_{c_3}^3 + {}^{3}\omega_3{}^{\times}\left({}^{3}\omega_3{}^{\times}\,{}^{3}p_{c_3}^3\right)\right] \tag{3.18}$$

Euler's Equation:

$$^{3}N_3 = {}^{3}I_3^c\,{}^{3}\dot{\omega}_3 + {}^{3}\omega_3{}^{\times}\,{}^{3}I_3^c\,{}^{3}\omega_3 \tag{3.19}$$

where, $^{3}I_3^c$ is the inertia tensor with respect to a frame located at the center of mass of the stick and with orientation the same as that of D-H frame 3.

$$^{i}I_i^c = \begin{bmatrix} I_{ixx} & -I_{ixy} & -I_{ixz} \\ -I_{ixy} & I_{iyy} & -I_{iyz} \\ -I_{ixz} & -I_{iyz} & I_{izz} \end{bmatrix},\ i = 1,\ 2 \text{ and } 3 \text{ (swing, boom and stick)} \tag{3.20}$$

Similarly for boom and swing the equations are summarized below

$$^{2}F_2 = m_2\,{}^{2}\dot{v}_{c_2} = m_2 \times \left[{}^{2}\dot{v}_2 + {}^{2}\dot{\omega}_2 \times {}^{2}p_{c_2}^2 + {}^{2}\omega_2 \times \left({}^{2}\omega_2 \times {}^{2}p_{c_2}^2\right)\right] \tag{3.21}$$

$$^{2}N_2 = {}^{2}I_2^c \times {}^{2}\dot{\omega}_2 + {}^{2}\omega_2 \times {}^{2}I_2^c \times {}^{2}\omega_2 \tag{3.22}$$

$$^{1}F_1 = m_1\,{}^{1}\dot{v}_{c_1} = m_1\left[{}^{1}\dot{v}_1 + {}^{1}\dot{\omega}_1{}^{\times}\,{}^{1}p_{c_1}^1 + {}^{1}\omega_1{}^{\times}\left({}^{1}\omega_1{}^{\times}\,{}^{1}p_{c_1}^1\right)\right] \tag{3.23}$$

$$^{1}N_1 = {}^{1}I_1^c\,{}^{1}\dot{\omega}_1 + {}^{1}\omega_1{}^{\times}\,{}^{1}I_1^c\,{}^{1}\omega_1 \tag{3.24}$$

Inward iteration for dynamics, starting from stick to swing: The manipulator is moving in free space, so no external forces/torques are applied at the end of the stick, i.e.

$$^4f_4 = {^4n_4} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.25}$$

Stick:

$$^3f_3 = {^3R_4}\,{^4f_4} + {^3F_3} = {^3F_3} \tag{3.26}$$

$$\begin{aligned} ^3n_3 &= {^3N_3} + {^3R_4}\,{^4n_4} + {^3p_{c_3}^3} \times {^3F_3} + {^3p_4^3} \times {^3R_4}\,{^4f_4} \\ &= {^3N_3} + {^3p_{c_3}^3} \times {^3F_3} \end{aligned} \tag{3.27}$$

Boom:

$$^2f_2 = {^2R_3}\,{^3f_3} + {^2F_2} \tag{3.28}$$

$$^2n_2 = {^2N_2} + {^2R_3}\,{^3n_3} + {^2p_{c_2}^2} \times {^2F_2} + {^2p_3^2} \times {^2R_3}\,{^3f_3} \tag{3.29}$$

Swing:

$$^1f_1 = {^1R_2}\,{^2f_2} + {^1F_1} \tag{3.30}$$

$$^1n_1 = {^1N_1} + {^1R_2}\,{^2n_2} + {^1p_{c_1}^1} \times {^1F_1} + {^1p_2^1} \times {^1R_2}\,{^2f_2} \tag{3.31}$$

Extracting the $\hat{z}$ components of the ^in_i, we find the expression of the joint torques as listed in Appendix A.

3.2 Head Attachment

In the previous sections the three dof system modeling was described. As the manipulator will be used to cut trees, there must be a head/end-effector on which a saw is mounted. The end-effector is mounted at the end of the stick with a pin. The total system now has five dof and link attachments as shown in Figure 3.1. As there is no torque/force input at the tip of the stick, the two gimbals will swing like a compound conical pendulum.

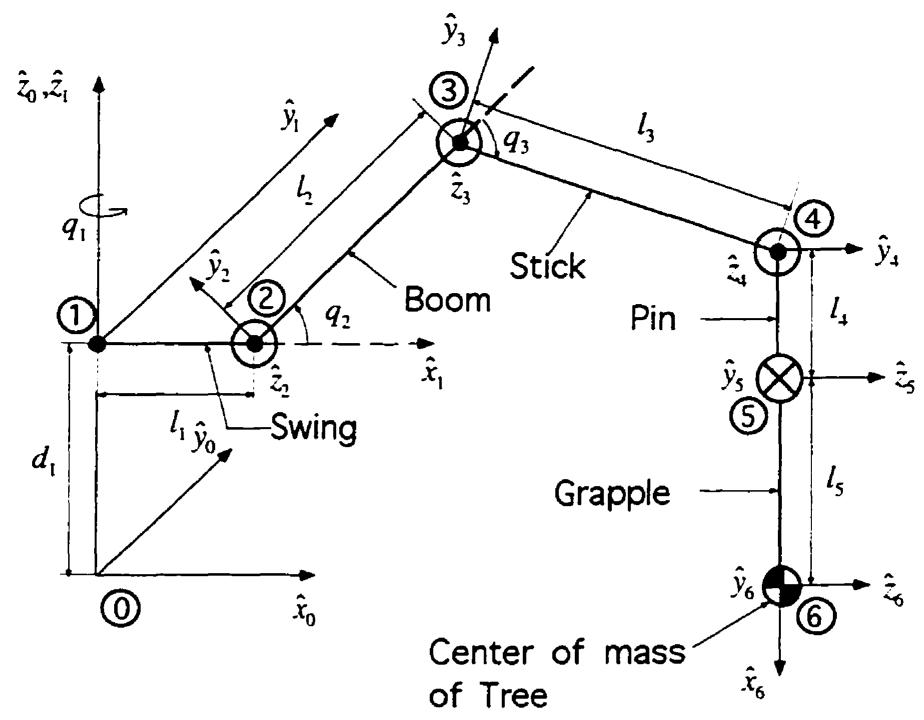


Figure 3.1: Link Attachment for the 5 dof System.

The total system now has five dof with the D-H parameters are shown in the following Table 3.1.

Table 3.1: D-H Parameters for the 5 dof System.

i	α_{i-1}	a_{i-1}	d_i	q_i
1	0	0	d_1	q_1
2	$\pi/2$	l_1	0	q_2
3	0	l_2	0	q_3
4	0	l_3	0	q_4
5	$-\pi/2$	l_4	0	q_5
6	0	l_5	d_6	0

Where the new parameters l_4 (= 0.24 m) and l_5 (= 2.0 m) are the pin length and distance from $\hat{z}_5$ to $\hat{z}_6$ along $\hat{x}_5$ respectively, and d_6 (= 0 m) (is the distance from $\hat{x}_5$ to $\hat{x}_6$ along $\hat{z}_6$, see Figure 3.1. For the dynamic formulation of the five dof manipulator we need additional rotation matrices and link parameters. The additional rotational matrices are given by

$$ {}^4R_5 = \begin{bmatrix} c_5 & -s_5 & 0 \\ 0 & 0 & 1 \\ -s_5 & -c_5 & 0 \end{bmatrix}, {}^5R_6 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.32} $$

and the link parameters are

$$ {}^4p_{c_4}^4 = \begin{bmatrix} x_4 \\ y_4 \\ z_4 \end{bmatrix}, {}^4p_5^4 = \begin{bmatrix} l_4 \\ 0 \\ 0 \end{bmatrix}, {}^5p_{c_5}^5 = \begin{bmatrix} x_5 \\ y_5 \\ z_5 \end{bmatrix} \tag{3.33} $$

The velocity and acceleration propagation are give below

$$ {}^4\omega_4 = {}^4R_3\,{}^3\omega_3 + \dot{q}_4\,{}^4\hat{z}_4 \tag{3.34} $$

$$ {}^4\dot{\omega}_4 = \frac{d}{dt}\left({}^4\omega_4\right) \tag{3.35} $$

$$ {}^4\dot{v}_4 = {}^4R_3\left[{}^3\dot{v}_3 + {}^3\dot{\omega}_3{}^{\times\,3}p_4^3 + {}^3\omega_3{}^{\times}\left({}^3\omega_3{}^{\times\,3}p_4^3\right)\right] \tag{3.36} $$

$$ {}^4\dot{v}_{c_4} = {}^4\dot{v}_4 + {}^4\dot{\omega}_4{}^{\times\,4}p_{c_4}^4 + {}^4\omega_4{}^{\times}\left({}^4\omega_4{}^{\times\,4}p_{c_4}^4\right) \tag{3.37} $$

$$ {}^5\omega_5 = {}^5R_4\,{}^4\omega_4 + \dot{q}_5\,{}^5\hat{z}_5 \tag{3.38} $$

$$ {}^5\dot{\omega}_5 = \frac{d}{dt}\left({}^5\omega_5\right) \tag{3.39} $$

$$ {}^5\dot{v}_5 = {}^5R_4\left[{}^4\dot{v}_4 + {}^4\dot{\omega}_4{}^{\times\,4}p_5^4 + {}^4\omega_4{}^{\times}\left({}^4\omega_4{}^{\times\,4}p_5^4\right)\right] \tag{3.40} $$

$$ {}^5\dot{v}_{c_5} = {}^5\dot{v}_5 + {}^5\dot{\omega}_5{}^{\times\,5}p_{c_5}^5 + {}^5\omega_5{}^{\times}\left({}^5\omega_5{}^{\times\,5}p_{c_5}^5\right) \tag{3.41} $$

Newton's Equation:

$$ {}^5F_5 = m_5\,{}^5\dot{v}_{c_5} = m_5\left[{}^5\dot{v}_5 + {}^5\dot{\omega}_5{}^{\times\,5}p_{c_5}^5 + {}^5\omega_5{}^{\times}\left({}^5\omega_5{}^{\times\,5}p_{c_5}^5\right)\right] \tag{3.42} $$

Euler's Equation:

$$ {}^5N_5 = {}^5I_5^c\,{}^5\dot{\omega}_5 + {}^5\omega_5{}^{\times\,5}I_5^c\,{}^5\omega_5 \tag{3.43} $$

For pin the Newton's and Euler's equations are given below

$$^4F_4 = m_4{}^4\dot{v}_{c_4} = m_4\left[{}^4\dot{v}_4 + {}^4\dot{\omega}_4{}^{\times 4}p^4_{c_4} + {}^4\omega_4{}^{\times}\left({}^4\omega_4{}^{\times 4}p^4_{c_4}\right)\right] \quad (3.44)$$

$$^4N_4 = {}^4I^c_4{}^4\dot{\omega}_4 + {}^4\omega_4{}^{\times 4}I^c_4{}^4\omega_4 \quad (3.45)$$

The end-effector of the machine will move freely in space after cutting a tree. There is no external force/torque applied at the end. Hence we can write

$$^6f_6 = {}^6n_6 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.46)$$

Inward iteration for the outer link to inner links is given below

End-effector:

$$^5f_5 = {}^5R_6{}^6f_6 + {}^5F_5 = {}^5F_5 \quad (3.47)$$

$$\begin{aligned} {}^5n_5 &= {}^5N_5 + {}^5R_6{}^6n_6 + {}^5p^5_{c_5}{}^{\times 5}F_5 + {}^5p^{5\times 5}_6R_6{}^6f_6 \\ &= {}^5N_5 + {}^5p^5_{c_5}{}^{\times 5}F_5 \end{aligned} \quad (3.48)$$

Pin:

$$^4f_4 = {}^4R_5{}^5f_5 + {}^4F_4 \quad (3.49)$$

$$^4n_4 = {}^4N_4 + {}^4R_5{}^5n_5 + {}^4p^4_{c_4}{}^{\times 4}F_4 + {}^4p^{4\times 4}_5R_5{}^5f_5 \quad (3.50)$$

The rest of the equations are same as described before.

The equations of motion are now quite complex. In matrix form they are expressed as

$$M\ddot{q} + V(q,\dot{q}) + G(q) = \tau \quad (3.51)$$

where M is a 5 $\times$ 5 mass matrix, V includes the Coriolis and centrifugal terms, G includes the gravity terms, and τ is the torque vector. Their structures are as follows

$$M = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} & m_{15} \\ m_{12} & m_{22} & m_{23} & m_{24} & m_{25} \\ m_{13} & m_{23} & m_{33} & m_{34} & m_{35} \\ m_{14} & m_{24} & m_{34} & m_{44} & m_{45} \\ m_{15} & m_{25} & m_{35} & m_{45} & m_{55} \end{bmatrix},\ V(q,\dot{q}) = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \end{bmatrix},\ G(q) = \begin{bmatrix} G_1 \\ G_2 \\ G_3 \\ G_4 \\ G_5 \end{bmatrix},\ \tau = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ 0 \\ 0 \end{bmatrix} \quad (3.52)$$

As there are no actuators in the head, the corresponding entries in the torque vector are zero.

3.3 Base Compliance

The vehicle and all other sub-systems excluding the manipulator, are modeled as a lumped mass, called thereafter as the 'base', see Figure 3.2. The base may oscillate around its home position, but it will not translate, i.e. the wheels are assumed locked. A body-fixed frame 0 is attached to the base, that coincides with a world-fixed frame when the vehicle is at its home position. The $\hat{x}_0$ axis of the body-fixed frame is along the direction of forward motion of the vehicle, while at home position, its $\hat{z}_0$ axis is in opposite direction to the gravity vector.

A force/torque set, (f, n), is applied to the base through the tires and the ground. Here it is assumed that the soil has been compacted, and that most of the base compliance is due to the vehicle's pneumatic tires. Therefore, these forces depend on the state of the tires. The four tires of the forestry vehicle are modeled as four parallel springs and dampers. The simultaneous vertical motion of the springs gives rise to a bouncing effect of the system. Due to the parallel spring structure, the base is also subject to pitch (rotation of the base around the $\hat{y}_0$ axis) and roll (rotation of the base around the $\hat{x}_0$ axis) motions. For small deviations from the home position, the yaw effects are negligible, and are therefore neglected.

The iterative Newton-Euler algorithm was developed for fixed-base systems in which all dofs are actuated. In such case, known desired trajectories for all joints, or dofs, are used to calculate numerically the forces and torques necessary to cause the desired motion. This is not possible in the case of a manipulator mounted on a compliant base, since the base is not actuated, and its position, velocity and acceleration will depend on how fast the arm moves, the load being manipulated, etc. However, if this formulation is applied symbolically, then it results in a closed set of symbolic equations of motions. This is the approach taken here, and is explained in detail below.

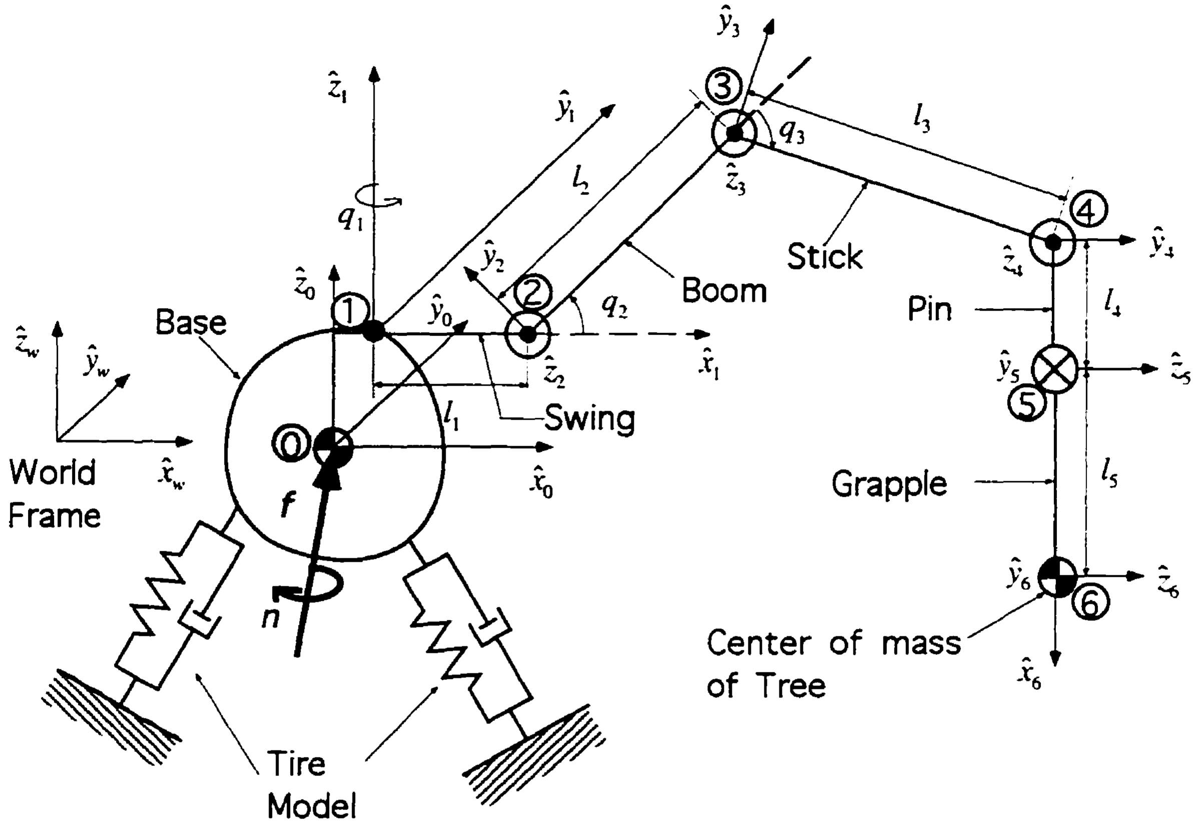


Figure 3.2: The 8 dof System as a Lumped Model.

On all five links, frames are attached following the modified Denavit-Hartenberg methodology as described earlier. Frame 0, i.e. the base frame $(\hat{x}_0\hat{y}_0\hat{z}_0)$, is attached at the center of mass of the base and has the same orientation with the swing frame, when the angle of swing rotation is zero. The rotation matrix that transforms the vector in the base frame to those in the world frame is computed based on a zyx Euler angle succession and is given by

$$ {}^{w}R_0 = \begin{bmatrix} c_z & -s_z & 0 \\ s_z & c_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_y & 0 & s_y \\ 0 & 1 & 0 \\ -s_y & 0 & c_y \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_x & -s_x \\ 0 & s_x & c_x \end{bmatrix} \tag{3.53} $$

where c_z is the cosine of q_z and s_z is the sine of q_z, etc. The angles q_x, q_y, and q_z are the roll, pitch and yaw respectively. Since the yaw is neglected, q_z is set equal to zero and this rotation matrix becomes

$$ {}^{w}R_0 = \begin{bmatrix} c_y & 0 & s_y \\ 0 & 1 & 0 \\ -s_y & 0 & c_y \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_x & -s_x \\ 0 & s_x & c_x \end{bmatrix} \tag{3.54} $$

The position vectors are shown below

$$ {}^{w}p_0^{w} = \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix},\ {}^{0}p_1^{0} = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix},\ {}^{0}p_{c_0}^{0} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.55} $$

The manipulator dynamics will not depend on the values of x_w, y_w, and, z_w but it will depend on the parameters x_0, y_0, and, z_0 (will be available by solid modeling of Auto-CAD with Advanced Modeling Extension (AME) package). The initial conditions are given below

$$ {}^{w}v_w = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},\ {}^{w}\dot{v}_w = \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix},\ {}^{w}\omega_w = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},\ {}^{w}\dot{\omega}_w = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.56} $$

Velocity and acceleration propagation equations are derived below, assuming that roll, q_x, and pitch, q_y, angles are small

$$ {}^{0}\omega_0 = {}^{0}R_w\,{}^{w}\omega_w + \begin{bmatrix} \dot{q}_x \\ \dot{q}_y \\ 0 \end{bmatrix} \tag{3.57} $$

Only vertical motion (bounce) effect is considered here, so

$$ {}^{0}v_0 = {}^{0}R_w\left({}^{w}v_w + {}^{w}\omega_w \times {}^{w}p_0\right) + \begin{bmatrix} 0 \\ 0 \\ \dot{z} \end{bmatrix} \tag{3.58} $$

$$^{0}\dot{v}_0={}^{0}R_w\left[{}^{w}\dot{v}_w+{}^{w}\dot{\omega}_w{}^{\times w}p_0^w+{}^{w}\omega_w{}^{\times}\left({}^{w}\omega_w{}^{\times w}p_0^w\right)\right]+2\left({}^{0}R_w{}^{w}\omega_w\right)^{\times}\begin{bmatrix}0\\0\\\dot{z}\end{bmatrix}+\begin{bmatrix}0\\0\\\ddot{z}\end{bmatrix} \tag{3.59}$$

$$^{0}\dot{v}_{c_0}={}^{0}\dot{v}_0+{}^{0}\dot{\omega}_0{}^{\times 0}p_{c_0}^0+{}^{0}\omega_0{}^{\times}\left({}^{0}\omega_0{}^{\times 0}p_{c_0}^0\right)={}^{0}\dot{v}_0 \tag{3.60}$$

$$^{0}F_0=m_0{}^{0}\dot{v}_{c_0} \tag{3.61}$$

$$^{0}N_0={}^{0}I_0^{c\,0}\dot{\omega}_0+{}^{0}\omega_0{}^{\times 0}I_0^{c\,0}\omega_0 \tag{3.62}$$

$$^{0}f_0={}^{0}R_1{}^{1}f_1+{}^{0}F_0 \tag{3.63}$$

$$^{0}n_0={}^{0}N_0+{}^{0}R_1{}^{1}n_1+{}^{0}p_{c_0}^{0}{}^{\times 0}F_0+{}^{0}p_1^{0\times 0}R_1{}^{1}f_1 \tag{3.64}$$

The force and torque vectors at the center of mass of the base (${}^{0}f_0$ and ${}^{0}n_0$) can be found as the last component of the inward iterations as shown by Eq. (3.63) and Eq. (3.64). This force and torque can be expressed in world frame as follows

$$^{w}f_0={}^{w}R_0{}^{0}f_0 \tag{3.65}$$

$$^{w}n_0={}^{w}R_0{}^{0}n_0 \tag{3.66}$$

where ${}^{w}R_0$ can be found from Eq. (3.54). Next, a generalized force vector *(F)* is introduced as

$$F=\begin{bmatrix}{}^{w}f_0\\{}^{w}n_0\end{bmatrix} \tag{3.67}$$

Vector F can be equated with a force and torque generated by the tires, as follows

$$F=-KX-B\dot{X} \tag{3.68}$$

where X and $\dot{X}$ are generalized displacement and velocity vectors with respect to world frame as given in (3.69).

$$X=\begin{bmatrix}x & y & z & \theta_x & \theta_y & \theta_z\end{bmatrix}^T \tag{3.69}$$

where x, y and z describe the position of the center of mass of the base with respect to world frame, and θ_x, θ_y and θ_z are the rotation of the base with respect to $\hat{x}_w$, $\hat{y}_w$ and $\hat{z}_w$

respectively see Figure 3.2. K and B are the stiffness and damping matrices and capture the effect of the tire model. For simplicity, and for small motions, these matrices are assumed to be diagonal.

$$K = diag(k_x, k_y, k_z, k_1, k_2, k_3) \tag{3.70}$$

$$B = diag(b_x, b_y, b_z, b_1, b_2, b_3) \tag{3.71}$$

The symbols k_x, k_y and k_z represent the total linear stiffness along the corresponding directions, as denoted by subscript with respect to the world frame. The term 'total' stiffness is used to represent the combined stiffness of the four tires. The other parameters k_1, k_2 and k_3 represent the total angular stiffness namely roll, pitch and yaw (rotation with respect to $\hat{x}_w$, $\hat{y}_w$ and $\hat{z}_w$ as experienced at the center of mass of the base). The same notation is applied in case of damping. Since only three base motions are considered important, i.e., bounce, roll and pitch, the remaining base equations are dropped. When this is done, the other two displacements are constant and the yaw angle is zero.

Finally, the equations of motion are written as

$$M\ddot{q} + V(q, \dot{q}) + G(q) = \tau \tag{3.72}$$

where M is an 8×8 symmetric and positive definite mass matrix, V contains the Coriolis and centrifugal terms, G the gravity terms and τ is the force/torque vector. Appendix B shows the source code for dynamic formulation for an 8 dof system in Mathematica.

4. System Parameter Estimation

Model parameters are needed to run the simulations, validate the developed code, and design controllers. Geometrical parameters such as lengths can be found from blueprints, and verified by direct measurements. Some masses are also found from drawings, or by directly weighing the body of interest. But parameters like center of mass locations, and moments and products of inertia, can not be obtained from drawings. In the case of the boom and stick, pendulum experiments were carried out to measure the moments of inertia of those links. In the case of products of inertia no such experiments can be made easily. For these, solid modeling techniques and the Advanced Modeling Extension package of AutoCAD were used. Another set of parameters was required to characterize the base compliance due to the tires. The stiffness and damping ratio of the tires are found by static load-deflection experiments and drop experiments, respectively. From the estimated values of stiffness and damping ratio the parameters for the roll, pitch and bounce were calculated.

4.1 Pendulum Experiment

Pendulum experiments are not always possible, because they require disassembling a system to its components. In this case, it was possible to do them while the machine was disassembled for maintenance reasons. During a pendulum experiment, a rigid body is first suspended from a point, see Figure 4.1, usually one of its joints. After the body comes to a rest, it is angularly displaced with respect to some axis, and then it is set free to swing. The period of the resulting oscillation is recorded, and is subsequently used to calculate the moment of inertia around the axis of rotation according to the following equation

$$\omega = \frac{2\pi}{T} = \sqrt{\frac{mgl}{I}} \tag{4.1}$$

where, ω is the frequency of oscillation, T is the period and l is the length from the point of suspension to the center of mass of the body.

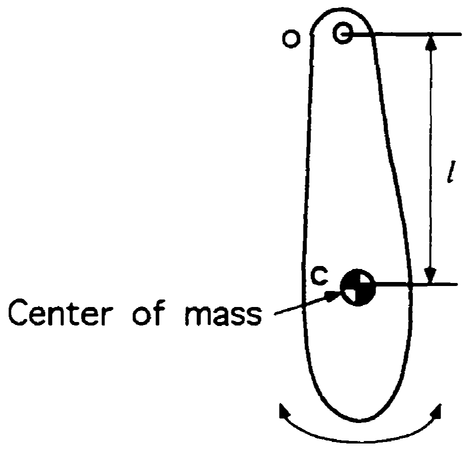


Figure 4.1: Schematic Diagram for Pendulum Experiment

Rearranging the terms of Eq. (4.1) we get

$$I = \frac{mglT^2}{4\pi^2} \tag{4.2}$$

Note that the moment of inertia is calculated with respect to the point of suspension. But the inertia tensor used in the Newton-Euler formulation should be expressed with respect to the center of mass. Using the parallel axis theorem we get

$$I_{zz}^{c} = I_{zz}^{o} - ml^2 \tag{4.3}$$

The calculated mass properties for the boom are reported here. Swinging around the z-axis (joint axis) results in a period equal to *3.25 sec*. For a mass of *635 kg* and for a length *(l)* of *1.952 m* (based on the agreement of experiment and solid modeling) we find that

$$I_{2zz}^{c} = 833.65 \quad kgm^2$$

For the y axis (conforms to D-H axis notation) we have obtained a time period equal to *3.33 sec*. For a mass of *635 kg*. and length *(l) 2.219 m*, we find that

$$I_{2yy}^{c} = 756.05 \quad kgm^2$$

Note that the point of suspension is different here. For the x axis it proved very difficult to suspend either the boom or the stick around their x-axis, and therefore these experiments were not done.

Eq. (4.2) shows that the inertia is proportional to the square of the time period. Therefore inaccurate timing results in a substantial amount of error in the moment of inertia. Moreover swinging a body with respect to a single axis is quite difficult. So the pendulum test is not very satisfactory in obtaining moments of inertia. However with the help of solid modeling, used with weight matching, we can obtain more accurate mass properties.

4.2 Solid Modeling

Solid modeling techniques can be used in obtaining all mass properties and center-of-mass positions, assuming that the material and the geometry of a body or link are precisely known. However, this is not always the case. To match solid modeling estimates to measurements, links of interest were weighted, and some moments of inertia were calculated using pendulum experiments. Then, solid models were refined to the point that both the estimated and measured total mass and moment of inertia were in agreement.

The basic procedure in solid modeling requires that first a closed boundary (*polyline* in Auto-CAD) should be drawn around a two dimensional surface, and then the boundary is extracted to a certain height with an appropriate taper angle to result in a solid shape (*solext*). To add holes or cavities in a solid, another cylinder or solid body is drawn, and then the child solid is subtracted from the parent solid (*solsub*). A complex rigid body, i.e. one comprised of many simple bodies, is obtained by uniting these to a single one (*solunion*). Different portions of a solid can be verified by observing sections at various positions (*solsect*). Once the solid is drawn it can be moved to any position and orientation (*solmove*). Also the user co-ordinate system can be moved to any position with respect to solid (*ucs*). The measuring system can be chosen according to the specific requirements (*sollength, solmass*). The accuracy of the measurement can be selected according to the specific requirement (*solsubdiv, soldecomp*). Once the solid is positioned and oriented with respect to a frame, the mass properties for the solid with respect to the user coordinate system can be computed (*solmassp*) by giving the density of the material (the links are made of steel, whose density is 7800 kg/m^3). Some solid models are shown in Figure 4.2.

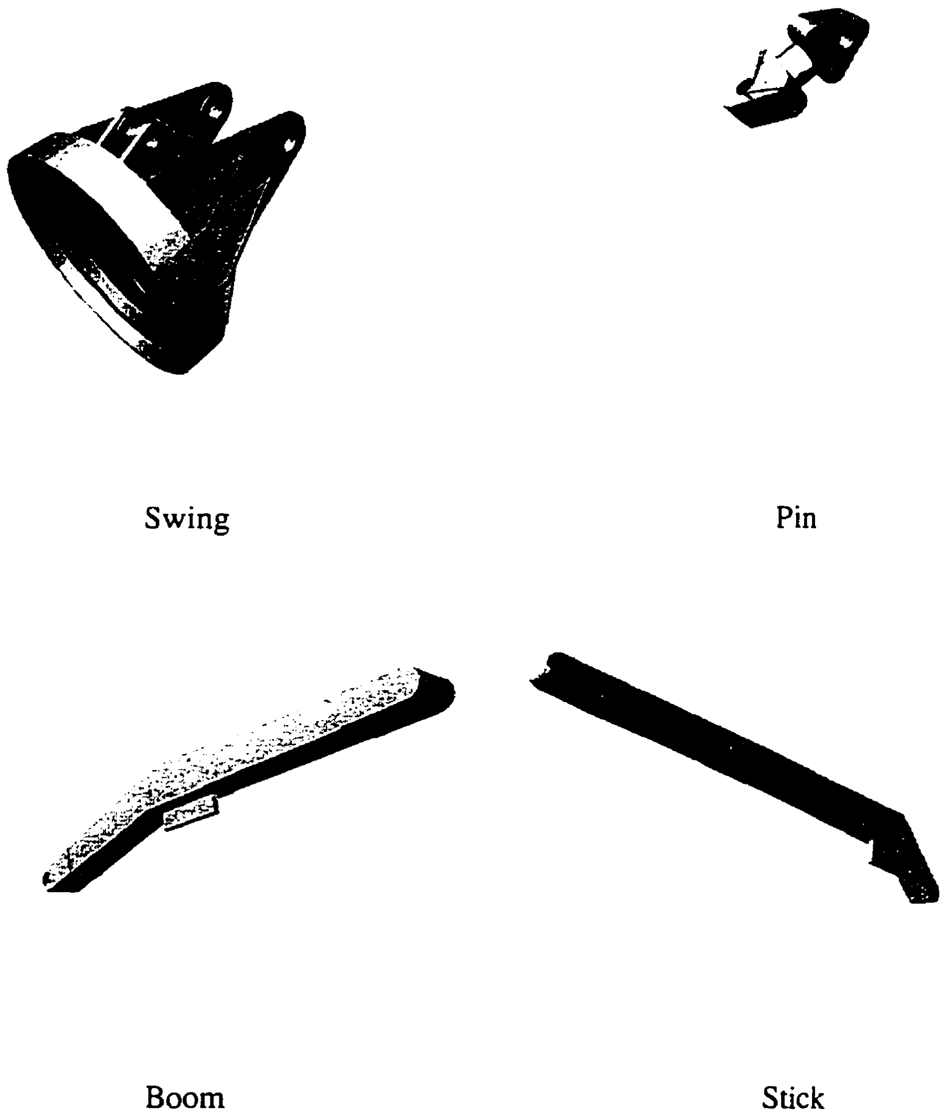


Figure 4.2: Solid Models

The solid models were refined to the point that both the estimated and measured total mass were in agreement. The model can be stored in the popular DXF format to be readable from other software.

The inertia parameters as obtained from the solid modeling are given in Table 4.1.

Table 4.1: Inertia Properties of Different Links.

in kgm^2	I_{xx}	I_{yy}	I_{zz}	I_{xy}	I_{yz}	I_{zx}
Swing	52	53	56	.01	.02	5
Boom	17	926	929	.36	-.09	-.70
Stick	16	816	826	32	.14	.51
Pin	0	3.28	3.28	0	0	0
Head & Tree	0	1265	1265	0	0	0

Note that pendulum experiments gave the following results: $I_{yy,boom} = 756\ kgm^2$, $I_{zz,boom} = 833\ kgm^2$, $I_{yy,stick} = 536\ kgm^2$ and $I_{zz,stick} = 869\ kgm^2$. However, these were not used due to the problem discussed in Section 4.1.

4.3 Load-Deflection Experiment

To obtain the stiffness of a tire, *k*, load-deflection experiments were conducted, where some load was applied on the tire and its vertical deflection was measured. The load on the tire was measured by weigh scales, see Figure 4.3.

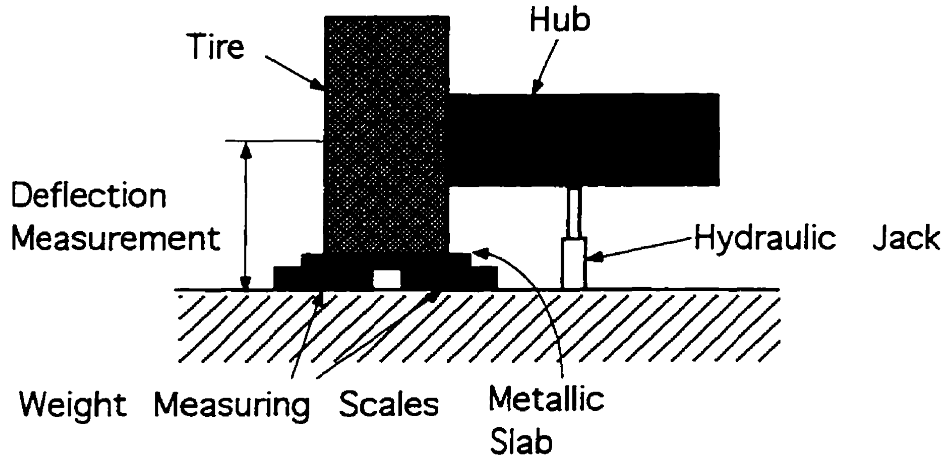


Figure 4.3: Schematic Diagram for Load-Deflection Experiment.

In executing this experiment, a main problem was to measure the load on the tire because the weight measuring scale was too small in size compared to the width of the tire. As two weight measuring scales were available it was decided to place a thick metallic slab on the two scales and then one tire was placed on the slab. Note that the candidate tire is not removed from the machine, the vehicle is driven onto the metallic slab. The variation of the load on tire is performed by operating a hydraulic jack. The sum of the two scale readings indicated the total load taken by the tire. Table 4.2 shows the actual data obtained from the experiment, while Figure 4.4 shows the plot corresponding to this data. The graph shows that the tire behaves like a linear spring. From the average slope we compute the stiffness is equal to *49.23 kgf/mm*.

Table 4.2: Load-Deflection Data.

Load in *kg*		Deflection (*mm*)
Scale 1	Scale 2	
100	1000	25
325	1750	45
650	2400	60
825	2750	73
1125	3000	80
1200	3100	90

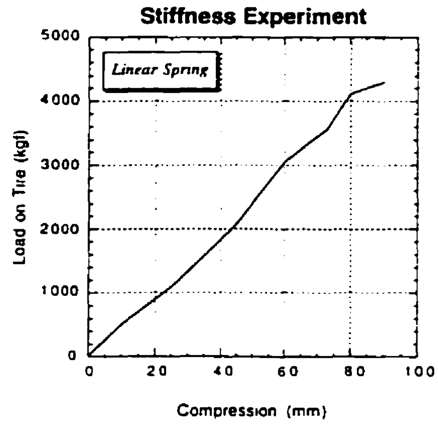


Figure 4.4: Measurement of Stiffness of a Tire.

There are four tires in the machine, modeled as four parallel springs. From Figure 4.5 we compute the relationship between the linear stiffness and roll or pitch stiffness. Basically, the derivation for the roll and pitch is exactly same; one only needs to use a different length (l), as the four wheels are located at the corners of a rectangle and not of square base.

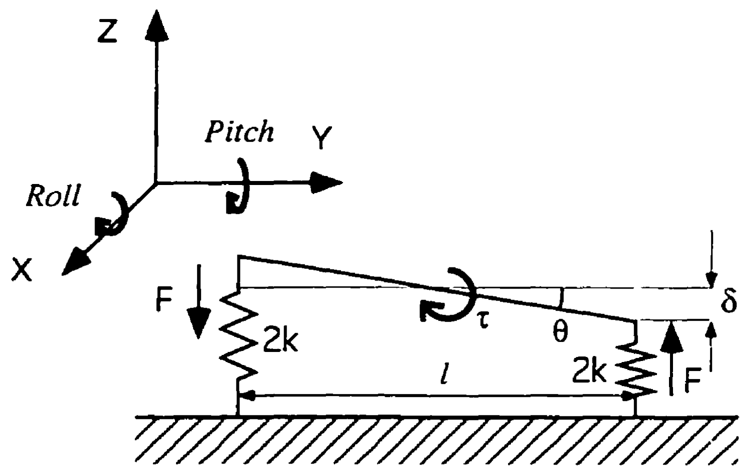


Figure 4.5: Schematic Diagram for Estimating Roll Stiffness.

From Figure 4.5 we write

$$\theta \ (in\ rad) = \frac{\delta}{½} = \frac{2\delta}{l} \qquad (4.4)$$

$$2k = \frac{F}{\delta} \tag{4.5}$$

$$\tau = Fl = k_\theta \theta \tag{4.6}$$

From Eqs. (4.4), (4.5) and (4.6) we get,

$$k_\theta = kl^2 \tag{4.7}$$

For stiffness in the roll and pitch directions, we write

$$k_x = kl_x^2 \tag{4.8}$$

$$k_y = kl_y^2 \tag{4.9}$$

Eq. (4.8) and (4.9) give the general relationship between the linear stiffness to the roll and pitch stiffness respectively. From the average slope of the value of k is estimated as *49230 kgf/m*, the total bounce stiffness for the four tires is

$$k_z = 4k = 1931785 \;\; \frac{N}{m}$$

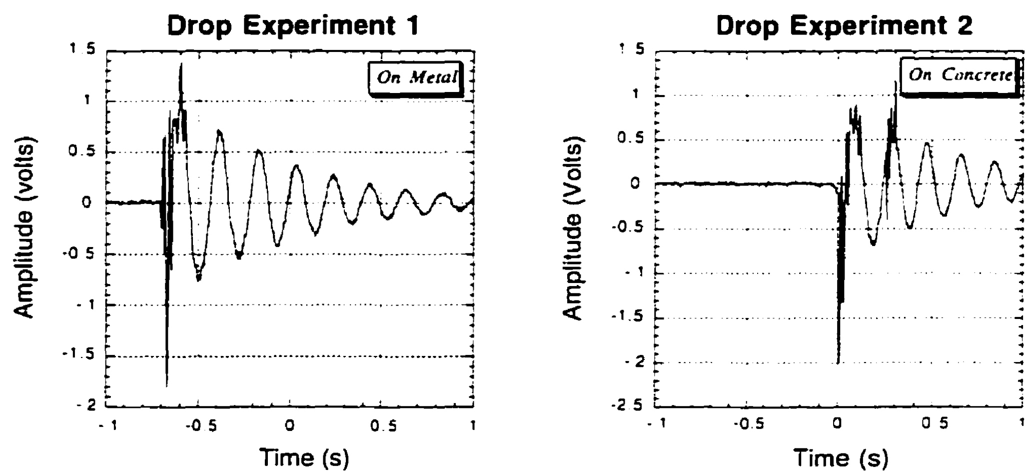
From the actual data (length for roll l_x = *3.124 m* and l_y = *3.048 m*) we get the following values for stiffness as listed in Table 4.3.

Table 4.3: Values for Stiffness.

Stiffness		
Roll $\left(\frac{Nm}{rad}\right)$	Pitch $\left(\frac{Nm}{rad}\right)$	Bounce $\left(\frac{N}{m}\right)$
4713255	4486718	1931785

4.4 Drop Experiment

One of the simplest methods to estimate the damping ratio of a non-rolling tire is the so-called drop test. The experimental procedure for standard automotive tires is described in [92]. In the case of a light tire, a load is added to the hub of the tire, which is just in contact with a steel slab, without deforming it (the load is supported externally). The load is then set free, and the loaded tire is allowed to deform freely from its initial position. Throughout the test, the tire must be in contact with the slab, otherwise obtained results will not be valid due to the physics of the collisions. An accelerometer mounted on its hub records the tire transient response, which corresponds to an under-damped oscillation. Figure 4.6 (a), (b) display typical accelerometer reading during drop experiments. All the data was digitized with the help of a scope. Different cases have been experimented by dropping the tire on a metallic slab, concrete or iron grill etc. As only the first few oscillations are needed to calculate the damping ratio, there is no need to record the entire response.



(a) (b)

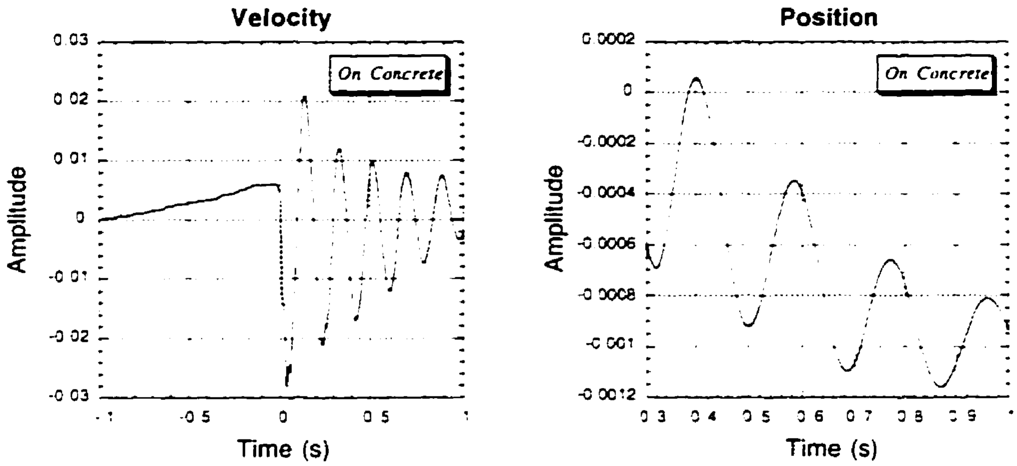
Figure 4.6: Measurement of Damping Ratio.

The velocity response is obtained by integrating the acceleration curve, shown in Figure 4.7 (a). The data on concrete was used to find the damping ratio. Before starting the experiment there was a constant drift in sensor output. This drift was identified and eliminated from the data. The resulting data was integrated again to obtain the position response as shown in Figure 4.7 (b). The first two oscillations are not taken as they include undesirable spikes. It is to be noted that as the ratio between the two consecutive amplitudes of oscillation are important, the units and the values in the vertical axes for both plots are not important. Hence initial conditions for the velocity or position plots are not important.

The damping ratio is obtained from the position response, by taking the ratio between two consecutive amplitudes measured from an average curve manually.

The calculation of damping ratio is shown below

$$\ln\left(\frac{x_1}{x_2}\right) = \frac{2\pi\zeta}{\sqrt{1-\zeta^2}} \tag{4.10}$$



(a) (b)

Figure 4.7: Velocity Plot.

The ratio of two consecutive amplitudes are found to be *1.25*, resulting in ζ is equal to *0.035*.

To compute the roll, pitch and bounce damping, we use the following procedure first. For bounce, the motion of a linear spring mass and damper is expressed as follows

$$m\ddot{x} + b\dot{x} + kx = F \tag{4.11}$$

$$\Rightarrow \ddot{x} + \frac{b}{m}\dot{x} + \frac{k}{m}x = \frac{F}{m} \tag{4.12}$$

where m, b and k are mass, damping coefficient and stiffness of a tire. The natural frequency of the second order system can be found by

$$\omega_n = \sqrt{\frac{k}{m}} \tag{4.13}$$

The mass of a tire is measured as 759 kg, the value of the stiffness is estimated as 49230 kgf/m (482946 N/m). The natural frequency is found to be 25.22 rad/s. From Figure 4.6 time period is found to be 0.25 s resulting the natural frequency ($2\pi/T$) is equal to 25.13 rad/s.

The relationship between damping coefficient and natural frequency can be found as below

$$2\zeta\omega_n = \frac{b}{m} \tag{4.14}$$

$$\Rightarrow b = 2\zeta\sqrt{km} \tag{4.15}$$

Substituting the values we get

$$b = 2\times .035\times\sqrt{49230\times 9.81\times 759} = 1340 \quad \frac{Ns}{m}$$

Since there are four tires, substituting the values we get the total damping in bounce direction for the four tires

$$b_z = 4b = 5360 \quad \frac{Ns}{m}$$

The total damping in roll and pitch direction is found using the model depicted in Figure 4.8. From Figure 4.8 we write

$$F = 2bv = 2b\frac{l}{2}\dot{\theta} = bl\dot{\theta} \tag{4.16}$$

$$\tau = Fl = bl^2\dot{\theta} = b_x\dot{\theta} \tag{4.17}$$

Therefore for roll

$$b_x = bl_x^2 \tag{4.18}$$

and for pitch

$$b_y = bl_y^2 \tag{4.19}$$

Substituting the corresponding lengths for pitch and roll we get the following values for damping coefficients as listed in Table 4.4.

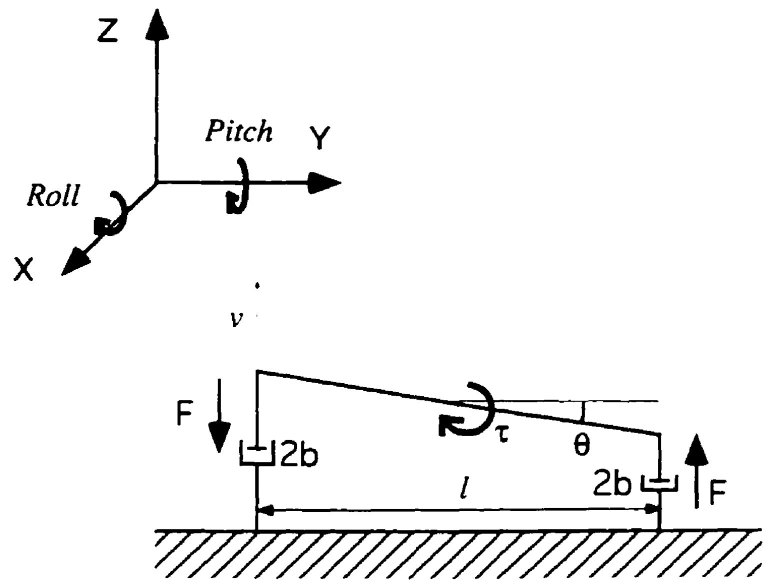


Figure 4.8: Schematic Diagram for Estimating Roll Damping

Table 4.4: Values for Damping Coefficients.

Damping Coefficients		
Roll $\left(\frac{Nms}{rad}\right)$	Pitch $\left(\frac{Nms}{rad}\right)$	Bounce $\left(\frac{Ns}{m}\right)$
13080	12450	5360

5. System Analysis and Design Using Inverse Dynamics

In this section various typical trajectories are generated to size valves and to check the total power required. Trapezoidal, cubic and quintic polynomial trajectories are used. As the actual power input is provided by the hydraulic cylinders and motor, the trajectory input is given at the actuator level rather than the joint level. The minimum and maximum piston position is measured and taken into account. For the swing, the joint limits are taken into account.

Valve sizing and power calculations are done based on the dynamic models developed earlier. Trajectories are fed to an inverse dynamics block of the manipulator to result in torque-time history of the actuated joints. Since the boom and stick are driven by hydraulic pistons, the force-time history is obtained from the history of torque. The pressure in the cylinders and pressure drops for the corresponding valves are computed. Also, from the velocity profile in the joint space, the velocity of the pistons and thereby the flows through the corresponding valves are computed. Finally, plotting the flow through the valve versus the pressure drop across it allows us to select a proper valve.

It is very important to know the total power required for the forestry operation. To this end, power requirements for individual joints are computed and added up. In the simulation results the total power requirements are shown for various conditions.

5.1 Trajectory Planning

In a typical trajectory, all joints move simultaneously. For the typical trajectory selected here, the boom and stick pistons move from their minimum to maximum positions, while the manipulator swing rotates from 0° to 90°. All joints start and finish moving at the same time, although different time limits can be programmed.

5.1.1 Trapezoidal Trajectory

The trapezoidal trajectory is a common trajectory used in common practice. Figure 5.1 shows a trapezoidal *velocity* trajectory.

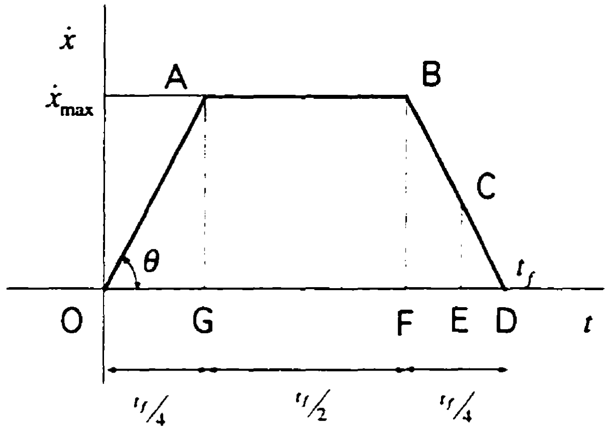


Figure 5.1: Trapezoidal Velocity Trajectory.

In the first part of the trajectory, OA, the piston of the corresponding link accelerates taking one fourth of the total time (t_f). In the second part, AB, the piston coasts with constant velocity $\dot{x}_{max}$ for half of the total time. Finally in part BD, the piston decelerates for the same amount of time as that for acceleration. The trapezoidal trajectory is parameterized using minimum and maximum joint limits (x_{min} and x_{max}), which are factory-set constants, and the total time (t_f). This particular type of trajectory planning is done in order to have only one variable, t_f to characterize the trajectory. For coordinated and simultaneous motion of all actuators, this type of trajectory planning is helpful as all joint movements start and finish at the same time. The slope of the line OA and the coasting velocity, $\dot{x}_{max}$ are also a function of the total time, t_f. The equations for the different segments can be written as follows

For the first segment, we can write for piston velocity

$$\dot{x} = \tan(\theta)t, \qquad 0 \le t \le \tfrac{t_f}{4} \tag{5.1}$$

Integrating we get the piston position,

$$x = \int_0^{\frac{t_f}{4}} \tan(\theta) t dt = \frac{t_f^2}{32} \tan(\theta) \tag{5.2}$$

The second segment is for coasting with constant velocity, therefore we get,

$$\dot{x} = \tan(\theta) \frac{t_f}{4}, \qquad \tfrac{t_f}{4} < t \leq \tfrac{3t_f}{4} \tag{5.3}$$

Integrating we get,

$$x = \frac{t_f}{4} \tan(\theta) \int_{\frac{t_f}{4}}^{\frac{3t_f}{4}} dt = \frac{t_f^2}{8} \tan(\theta) \tag{5.4}$$

For the last segment, the equation for piston velocity becomes,

$$\dot{x} = -\tan(\theta) t + \tan(\theta) t_f, \qquad \tfrac{3t_f}{4} < t \leq t_f \tag{5.5}$$

Integrating we get,

$$x = -\tan(\theta) \int_{\frac{3t_f}{4}}^{t_f} t dt + t_f \tan(\theta) \int_{\frac{3t_f}{4}}^{t_f} dt = \frac{t_f^2}{32} \tan(\theta) \tag{5.6}$$

From Eqs.(5.2), (5.4) and (5.6) we get the total displacement of the piston during the travel from minimum joint limit to maximum joint limit. Hence by adding we have,

$$x_{max} - x_{min} = \tan(\theta) \left(\frac{t_f^2}{32} + \frac{t_f^2}{8} + \frac{t_f^2}{32} \right) = \frac{3}{16} t_f^2 \tan(\theta) \tag{5.7}$$

$$\Rightarrow \quad \theta = \tan^{-1} \left(\frac{x_{max} - x_{min}}{0.1875 t_f^2} \right) \tag{5.8}$$

Eq. (5.8) is important and shows us what should be the value of the velocity gradient of the pistons so that one joint coasts for the half of the total time span and it accelerates and decelerates with the same velocity gradient.

For the position trajectory it is easy to calculate the area under the velocity trajectory. For the first segment we can write,

$$x = x_{min} + \tfrac{1}{2} t^2 \tan(\theta), \qquad 0 \leq t \leq \tfrac{t_f}{4} \tag{5.9}$$

In the next segment the velocity is constant ($\dot{x}_{max}$), the position trajectory will be a straight line. The equation of the line is given by,

$$x = x_{min} + \tfrac{1}{2}\dot{x}_{max}\frac{t_f}{4} + \dot{x}_{max}\left(t - \frac{t_f}{4}\right), \qquad \tfrac{t_f}{4} < t \le \tfrac{3t_f}{4} \tag{5.10}$$

and, the maximum velocity is given by,

$$\dot{x}_{max} = \frac{t_f}{4}\tan\theta \tag{5.11}$$

Substituting the expression for $\dot{x}_{max}$ in Eq. (5.10) and simplifying we get,

$$x = x_{min} - \frac{t_f^2}{32}\tan\theta + t\frac{t_f}{4}\tan\theta \tag{5.12}$$

For the last segment, the area BCEF in Figure 5.1 is found as follows

$$CE = \frac{\dot{x}_{max}\left(t_f - t\right)}{{}^{t_f}\!/_4} \tag{5.13}$$

$$area\ of\ BCEF = \frac{t_f^2}{32}\tan\theta - \frac{\left(t_f - t\right)^2}{2}\tan\theta \tag{5.14}$$

Therefore, the equation of the last segment can be written as,

$$x = x_{min} + \tfrac{1}{2}\dot{x}_{max}\frac{t_f}{4} + \dot{x}_{max}\left(\frac{3t_f}{4} - \frac{t_f}{4}\right) + area\,of\ BCEF, \qquad \tfrac{3t_f}{4} < t \le t_f \tag{5.15}$$

After simplification we get,

$$x = x_{min} + \tfrac{3}{16}t_f^2\tan\theta - \tfrac{1}{2}\left(t_f - t\right)^2\tan\theta \tag{5.16}$$

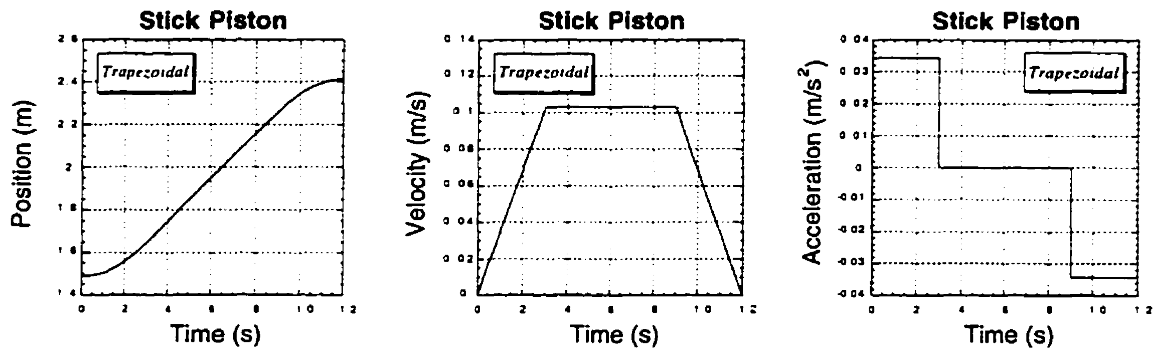
It is very straightforward to compute the acceleration trajectory , for the first segment acceleration is equal to the slope of the line OA, in the coasting region the acceleration is zero, and finally for the deceleration region it is equal to the slope of the line BD. Hence we can write,

$$\ddot{x} = \tan\theta \qquad 0 \le t \le \tfrac{t_f}{4} \tag{5.17}$$

$$\ddot{x} = 0 \qquad \tfrac{t_f}{4} < t \le \tfrac{3t_f}{4} \tag{5.18}$$

$$\ddot{x} = -\tan\theta \qquad \frac{3t_f}{4} < t \le t_f \tag{5.19}$$

Figure 5.2, (a), (b) and (c) show the actual trajectories for the stick piston, for $x_{max} = 2.41\, m$, $x_{min} = 1.49\, m$, and $t_f = 12$ sec.



(a) (b) (c)

Figure 5.2: Trapezoidal Trajectory for the Stick Piston

5.1.2 Cubic Polynomial Trajectory

Another trajectory which was used in the simulation is the cubic polynomial trajectory. Unlike in trapezoidal trajectories, in this method the velocity is smooth.

The initial and final positions of the piston are given by

$$x_{t=0} = x_{min} \tag{5.20}$$

$$x_{t=t_f} = x_{max} \tag{5.21}$$

The piston starts from zero velocity at x_{min} and stops at x_{max}. Therefore we can write the following

$$\dot{x}_{t=0} = 0 \tag{5.22}$$

$$\dot{x}_{t=t_f} = 0 \tag{5.23}$$

The cubic polynomial can satisfy the above four constraints. It can be written as

$$x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \tag{5.24}$$

$$\dot{x}(t) = a_1 + 2a_2 t + 3a_3 t^2 \tag{5.25}$$

$$\ddot{x}(t) = 2a_2 + 6a_3 t^2 \tag{5.26}$$

Using the boundary conditions as shown in Eqs. (5.20) through (5.23), we get the four equations with four unknowns as follows

$$a_0 = x_{\min} \tag{5.27}$$

$$a_1 = 0 \tag{5.28}$$

$$a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3 = x_{\max} \tag{5.29}$$

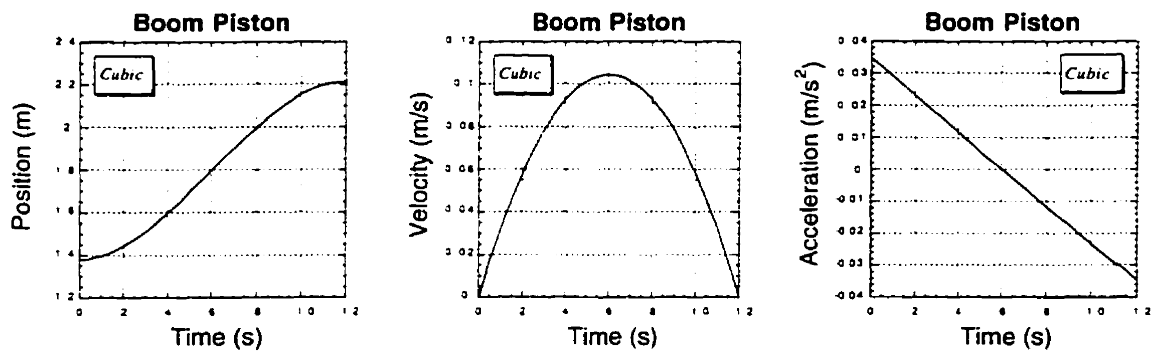
$$a_1 + 2a_2 t_f + 3a_3 t_f^2 = 0 \tag{5.30}$$

Solving the above four equations to obtain the coefficients we get,

$$a_2 = \frac{3}{t_f^2}\left(x_{\max} - x_{\min}\right) \tag{5.31}$$

$$a_3 = -\frac{2}{t_f^3}\left(x_{\max} - x_{\min}\right) \tag{5.32}$$

A candidate trajectory for the boom joint is considered. Substituting the actual joint limits and the trajectory time ($x_{\max} = 2.21\,m$, $x_{\min} = 1.38\,m$ and $t_f = 12\,\sec$) for the boom piston we get the following graphs as shown in Figure 5.3 (a), (b) and (c).



(a) (b) (c)

Figure 5.3: Cubic Polynomial Trajectory for the Boom Piston.

5.1.3 Quintic Polynomial Trajectory

The acceleration profile of the cubic polynomial starts from a non-zero value. However it is more realistic to have acceleration and deceleration equal to zero at starting and finishing time of the trajectory.

This can be obtained using a quintic polynomial is given by

$$x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \tag{5.33}$$

The desired boundary conditions are written as

$$x_{t=0} = x_{\min} \tag{5.34}$$

$$x_{t=t_f} = x_{\max} \tag{5.35}$$

$$\dot{x}_{t=0} = 0 \tag{5.36}$$

$$\dot{x}_{t=t_f} = 0 \tag{5.37}$$

$$\ddot{x}_{t=0} = 0 \tag{5.38}$$

$$\ddot{x}_{t=t_f} = 0 \tag{5.39}$$

By taking derivative of Eq. (5.33) and satisfying the boundary conditions we get the following equations

$$a_0 = x_{\min} \tag{5.40}$$

$$a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3 + a_4 t_f^4 + a_5 t_f^5 = x_{\max} \tag{5.41}$$

$$a_1 = 0 \tag{5.42}$$

$$a_1 + 2a_2 t_f + 3a_3 t_f^2 + 4a_4 t_f^3 + 5a_5 t_f^4 = 0 \tag{5.43}$$

$$2a_2 = 0 \tag{5.44}$$

$$2a_2 + 6a_3 t_f + 12a_4 t_f^2 + 20a_5 t_f^3 = 0 \tag{5.45}$$

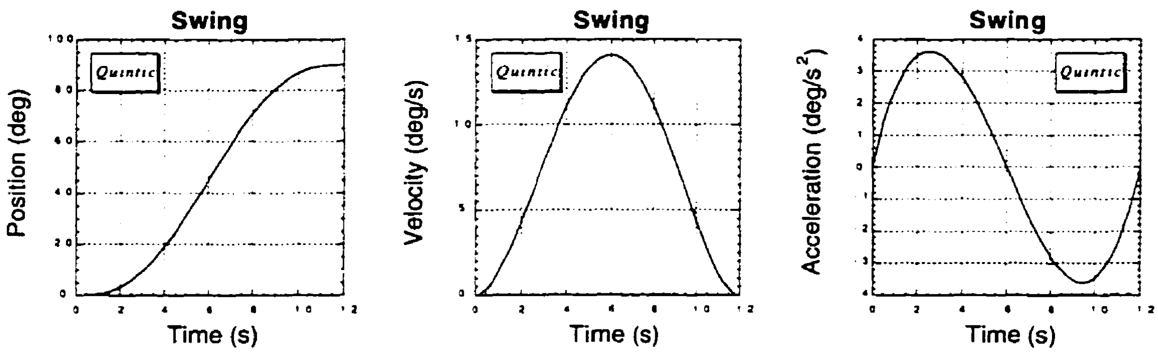
Solving Eqs. (5.41), (5.43) and (5.45) we obtain the remaining coefficients of the polynomial as

$$a_3 = \frac{10(x_{max} - x_{min})}{t_f^3} \tag{5.46}$$

$$a_4 = \frac{15(x_{min} - x_{max})}{t_f^4} \tag{5.47}$$

$$a_5 = \frac{6(x_{max} - x_{min})}{t_f^5} \tag{5.48}$$

Next, a candidate trajectory for the swing joint is considered. Providing the limits $x_{min} = 0^o$ and $x_{max} = 90^o$ we get the following graphs as shown in Figure 5.4 (a), (b), and (c). Note that the true joint limits are $\theta_{min} = -131.21^o$ and $\theta_{max} = 94.54^o$.



(a) (b) (c)

Figure 5.4: Quintic Polynomial Trajectory for the Swing.

5.2 Simulation Results

Many different desired trajectories were tested for valve selection. However only a few are reported here. Manipulator loads (head and tree) is modeled as a lumped mass at the end of the stick. All results are based on quintic polynomial trajectories unless it is mentioned otherwise.

5.2.1 Flow Profiles

The flow in the stick cylinder is given by the following

$$Q_s = A_s \dot{x}_s \tag{5.49}$$

where Q_s is the flow in the stick cylinder, A_s is the average area of the stick cylinder, and $\dot{x}_s$ is the velocity of the stick piston. This analysis assumes there is no compressibility to the hydraulic oil - probably okay except for large actuators.

Similarly, for the boom

$$Q_b = A_b \dot{x}_b \tag{5.50}$$

where Q_b is the flow in the boom cylinder, A_b is the average area of the boom cylinder, and $\dot{x}_b$ is the velocity of the boom piston.

The swing is driven by a gear pump, and the flow equation can be written as

$$Q_{sw} = D(n\dot{q}_1) \tag{5.51}$$

where Q_{sw} is the flow in the swing motor, D is the volumetric displacement of the motor (0.00018 $^{m^3}/_{rev}$), $\dot{q}_1$ is the angular velocity of the swing, and n is the gear ratio from swing to swing motor, from blue print we calculated the value of n is 221.778. Figure 5.5 shows the flow in the actuators using a quintic polynomial trajectory for all joints.

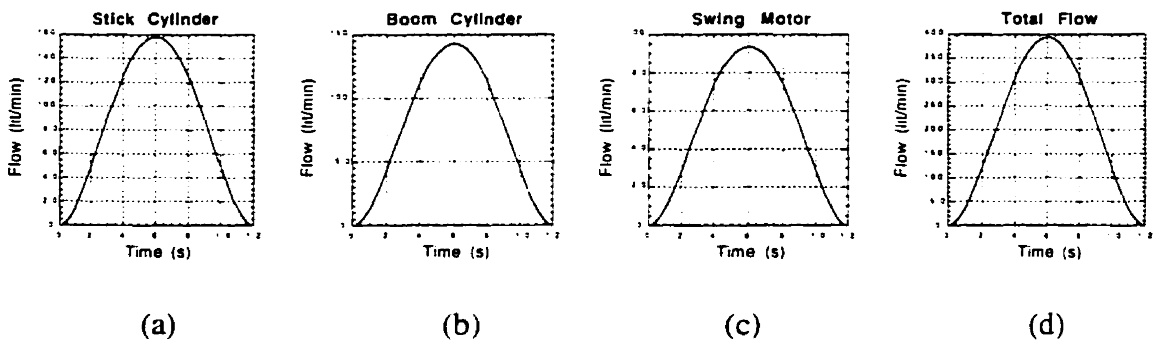


Figure 5.5: Flow Profiles.

Flow is directly proportional to the linear velocity of the piston in case of the boom and stick, and in case of the swing motor the flow is directly proportional to the angular velocity of the swing. So the nature of the curve is the same with the velocity profile of the corresponding link.

5.2.2 Parameter Changing due to Load Variation

Manipulator loads are modeled here for simplicity as a lumped mass at the end of the stick. In this three dof system, the stick with the load is modeled as new third link. The inertia parameters and center of mass of the new link vary as the amount of load to be carried varies. Figure 5.6 depicts the stick with a load.

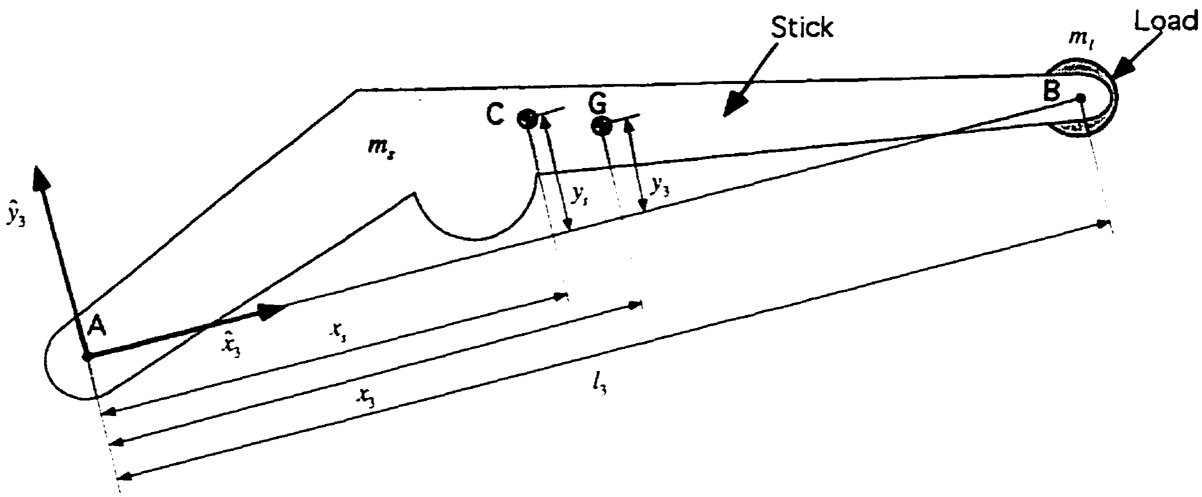


Figure 5.6: Stick With Load at the End.

In Figure 5.6, point C denotes the center of mass of the stick only. Point G denotes the position of the center of mass of the whole link when a load (m_l) is attached to the end of the stick. As the amount of load changes, the location of the point G changes too. Again with the change in the mass, the inertia parameters of this link change too. In the next formulation the relationship between the load and the location of the center of mass is derived.

The stick is assumed symmetric with respect to the $\hat{x}_3\hat{y}_3$ plane passing through its center of mass. It is also assumed that the machine grips the tree at its center of mass, otherwise we have to consider the z co-ordinate of the link. Actually at the end of the stick there is a processing head. The head can be modeled as a set of gimbals (two pendulums with axis of revolution perpendicular to each other). So if the head does not grip the tree at the center of mass, due to gravity effect the force for the load will act at the end of the stick. The total mass of the link with the load is

$$m_3 = m_s + m_l \tag{5.52}$$

where m_s is the mass of the stick and m_l is the mass of the load (head and tree). A moment balance with respect to point A yields

$$m_s x_s + m_l l_3 = m_3 x_3 \tag{5.53}$$

and,

$$\Rightarrow \quad x_3 = \frac{m_s x_s + m_l l_3}{m_3} \tag{5.54}$$

Similarly, for y co-ordinate, we get,

$$m_s y_s + m_l .0 = m_3 y_3 \tag{5.55}$$

$$\Rightarrow \quad y_3 = \frac{m_s y_s}{m_3} \tag{5.56}$$

where (x_3, y_3), is the new location of the center of mass of the stick with respect to the D-H frame at the boom-stick joint (point A). Note that if m_l is equal to zero, then x_3 and y_3 are equal to x_s and y_s respectively.

From the parallel axis theorem, we can write the inertia of the whole link as follows

$$I^c_{3zz} = I^c_{szz} + m_s l^2 + m_l l^{*2} \tag{5.57}$$

where, the subscript 3 denotes the third link, s denotes the stick and xx, denotes the inertia measured with respect to the x axis, similarly for the other axes. The superscript c denotes the inertia with respect to center of mass. The symbols l and l^* denote the distances between the points C and G, G and B respectively. The expression for l and l^* can be written as,

$$l = CG = \sqrt{(x_3 - x_s)^2 + (y_3 - y_s)^2} \tag{5.58}$$

$$l^* = BG = \sqrt{(L_3 - x_3)^2 + y_3^2} \tag{5.59}$$

Similarly, by inspection for the inertia with respect to x axis can be written as,

$$I_{3xx}^c = I_{sxx}^c + m_s(y_3 - y_s)^2 + m_l(y_3 - 0)^2 \tag{5.60}$$

Similarly, by inspection for the inertia with respect to y axis can be written as,

$$I_{3yy}^c = I_{syy}^c + m_s(x_3 - x_s)^2 + m_l(L_3 - x_3)^2 \tag{5.61}$$

For the product of inertia, the formula can be written as,

$$I_{3xy}^c = I_{sxy}^c + m_s(x_3 - x_s)(y_3 - y_s) + m_l(L_3 - x_3)y_3 \tag{5.62}$$

Other product of inertia are zero due to symmetry of the third link with respect to $\hat{x}_3\hat{y}_3$ plane passing through the center of mass. Hence we can write,

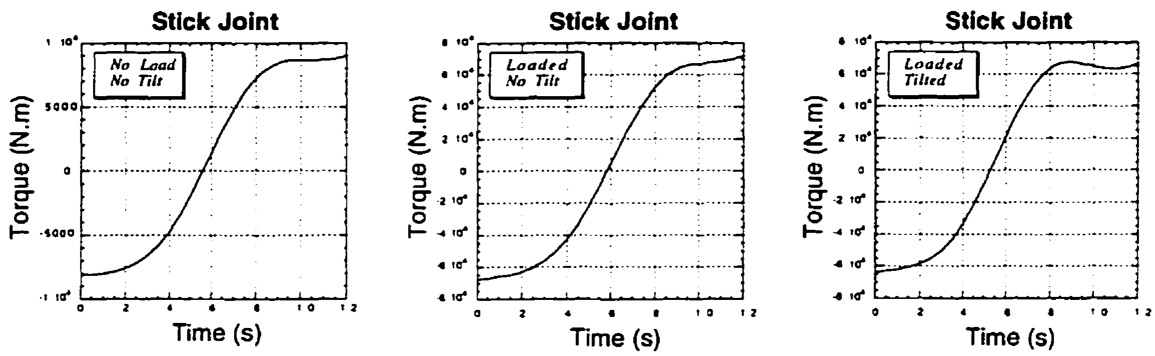
$$I_{3xz}^c = 0 \tag{5.63}$$

$$I_{3yz}^c = 0 \tag{5.64}$$

5.2.3 Torque Profiles

From the inverse dynamics we compute the torque-time history applied at the three joints of the system. Three distinct cases are studied. The first case, labeled as *'no load & no tilt,'* corresponds to no end point load and to a horizontal base. The second case, labeled as *'loaded & no tilt,'* corresponds to an extreme load of *1500 kg* and to a horizontal base. This load is due to the weight of the head *(955 kg)* and the remaining weight of a large tree *(545 kg)*. The third case labeled as *'loaded & tilted'* case. Here the machine is carrying a load of *1500 kg* as well as the base is tilted with respect to roll axis $\hat{x}_0$ by an amount of *20°*.

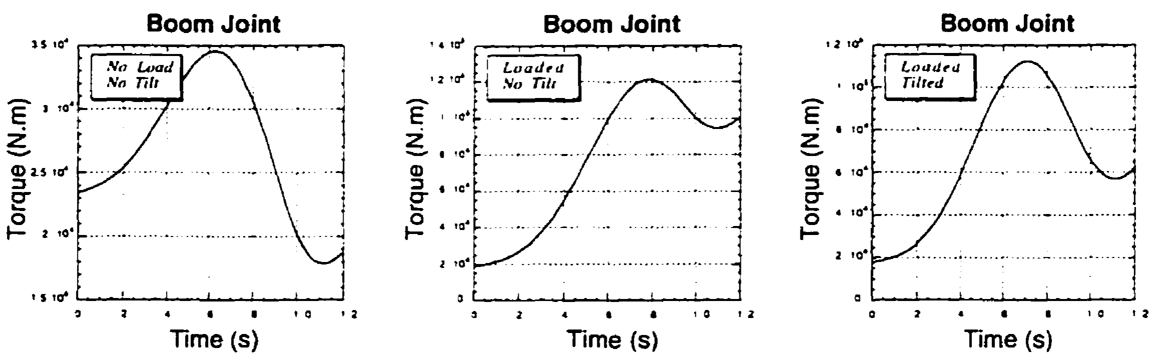
The torque histories as obtained from Eq. (3.71) for the stick are depicted in Figure 5.7. For the stick the torque starts from negative to positive as during the motion the stick goes through the vertical.



(a) (b) (c)

Figure 5.7: Torque Histories at Stick Joint for 1500 kg load and 20° tilt.

The torque histories as obtained from Eq. (3.71) for the boom are depicted in Figure 5.8. Since the boom is heavier than the stick and since it also supports the stick, the values for the boom torque are much larger than that of the stick. By observing the nature of the torque history for the boom joint, we observe that, unlike the stick, the torque does not go from negative to positive due to the configuration of the manipulator and the joint limits.



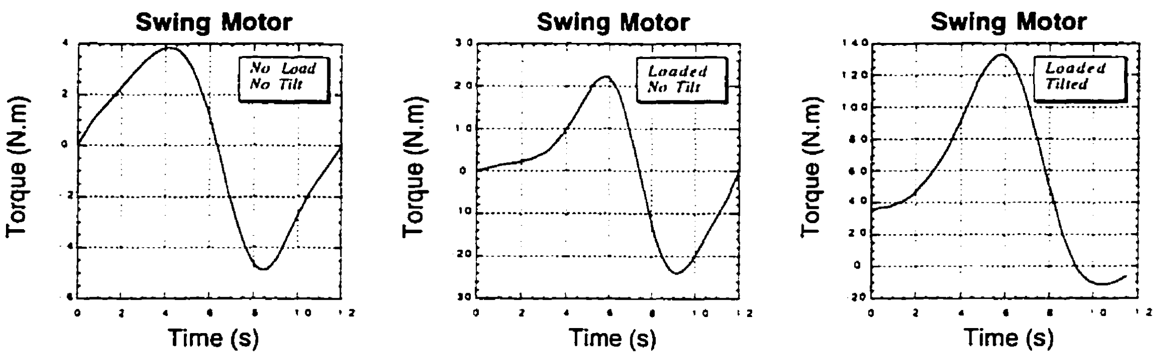
(a) (b) (c)

Figure 5.8: Torque Histories at Boom Joint for 1500 kg load and 20° tilt.

The torque generated at the swing motor is related to the torque accelerating the manipulator, τ_{sw}, by

$$\tau_m = \frac{\tau_{sw}}{n} \tag{5.65}$$

Note that τ_{sw} is the torque used in the equations of motion, see Eq. (3.71). Figure 5.9 shows the torque profiles at the shaft of the swing motor, after gear reduction, from swing to swing motor. The gravity vector does not affect the torque for the swing, if the base is not tilted, so in the case of a horizontal base, the torque requirement for the motion of the swing is considerably less than that of the boom or stick. Note that the initial and final values for the torque are zero for a horizontal base, see Figure 5.9 (a) and (b).



(a) (b) (c)

Figure 5.9: Torque Histories for Swing Motor in Different Cases.

5.2.4 Force Profile

Figure 5.10 depicts a schematic diagram of the actuation systems of the machine. From the equations of motion we obtain the actuator torque for the swing, boom, and, stick. In the ease of the boom and stick we have piston and cylinder actuators so we need a mapping from torque to force. This is obtained from a kinematic analysis as described in the next section.

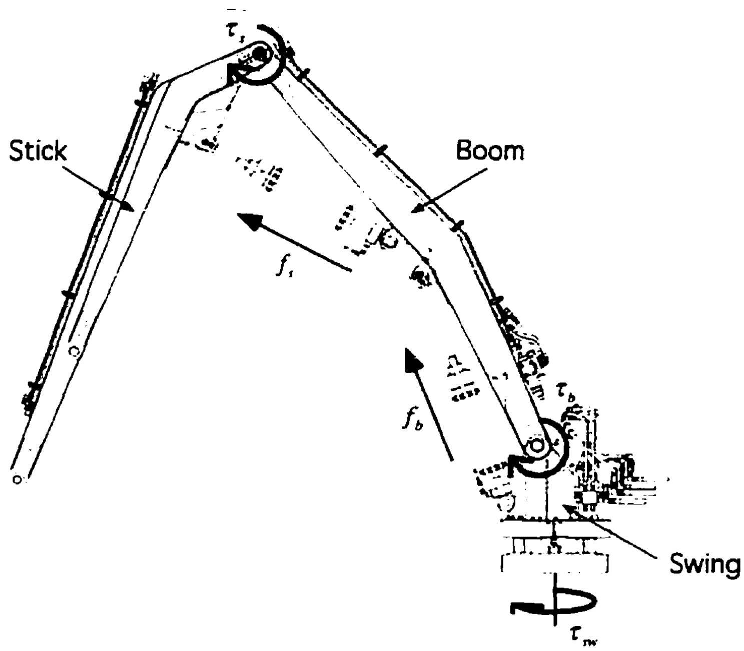


Figure 5.10: Schematic of Actuation Systems

5.2.4.1 Force at Stick Piston

Figure 5.11 shows the connection of the stick with its cylinder.

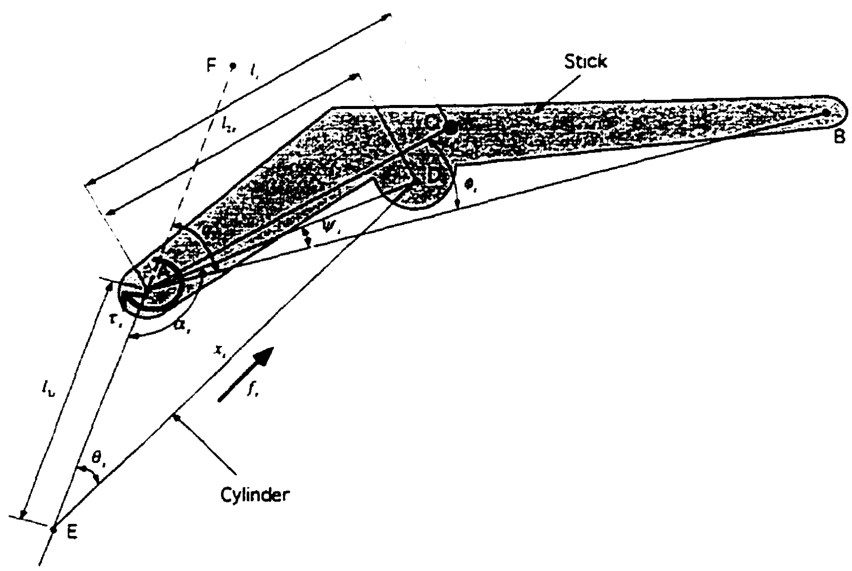


Figure 5.11: Schematic Diagram of Stick and its Connection (not to scale).

From Figure 5.11, the stick is connected to the boom at point A. The end of the piston rod is connected at D, while the other end of the cylinder is connected at E.

The force generated by the stick piston can be found by taking moments with respect to the boom-stick joint (point A). From triangle ADE by inspection we write the following

$$f_s = \frac{\tau_s}{l_{1s}\sin\theta_s} \tag{5.66}$$

where τ_s is the torque generated at the boom-stick joint, and f_s is the force generated at the piston of the stick see Figure 5.10 and Figure 5.11.

To continue, the relationship between θ_s and x_s is needed. To this end Figure 5.11 yields

$$\cos\theta_s = \frac{l_{1s}^2 + x_s^2 - l_{2s}^2}{2l_{1s}x_s} \tag{5.67}$$

Note that the angle θ_s varies with the time as the link moves. Another important relationship is the piston position to joint angle. The parameter x_s corresponds to the stick piston position to some angle q_3. Physically the stick angle can not be positive. From Figure 5.11 we get,

$$\psi_s = \angle BAD, \quad \alpha_s = \angle EAD, \quad \phi_s = \angle BAC, \quad \theta_s = \angle AED$$

The relationship $x_s(q_3)$ is found according to geometry as

$$x_s = \sqrt{l_{1s}^2 + l_{2s}^2 - 2l_{1s}l_{2s}\cos\alpha_s} \tag{5.68}$$

$$\Rightarrow \quad x_s = \sqrt{l_{1s}^2 + l_{2s}^2 - 2l_{1s}l_{2s}\cos(\pi + q_3 + \psi_s)} \tag{5.69}$$

$$\Rightarrow \quad x_s = \sqrt{l_{1s}^2 + l_{2s}^2 + 2l_{1s}l_{2s}\cos(q_3 + \psi_s)} \tag{5.70}$$

From triangle ADE,

$$\cos\alpha_s = \frac{l_{1s}^2 + l_{2s}^2 - x_s^2}{2l_{1s}l_{2s}} \tag{5.71}$$

$$q_3 = \alpha_s - \pi - \psi_s = \cos^{-1}\left(\frac{l_{1s}^2 + l_{2s}^2 - x_s^2}{2l_{1s}l_{2s}}\right) - \pi - \psi_s \tag{5.72}$$

By taking derivative of Eq. (5.70) we get,

$$\dot{x}_s = -\frac{l_{1s}l_{2s}\sin(q_3 + \psi_s)}{\sqrt{l_{1s}^2 + l_{2s}^2 + 2l_{1s}l_{2s}\cos(q_3 + \psi_s)}}\dot{q}_3 \tag{5.73}$$

5.2.4.2 Force at Boom Piston

Boom force expressions are derived following the procedure detailed in the previous section. In Figure 5.12, point A denotes the revolute joint of the boom with swing. The boom cylinder is connected at point E to the base, and point D to the boom.

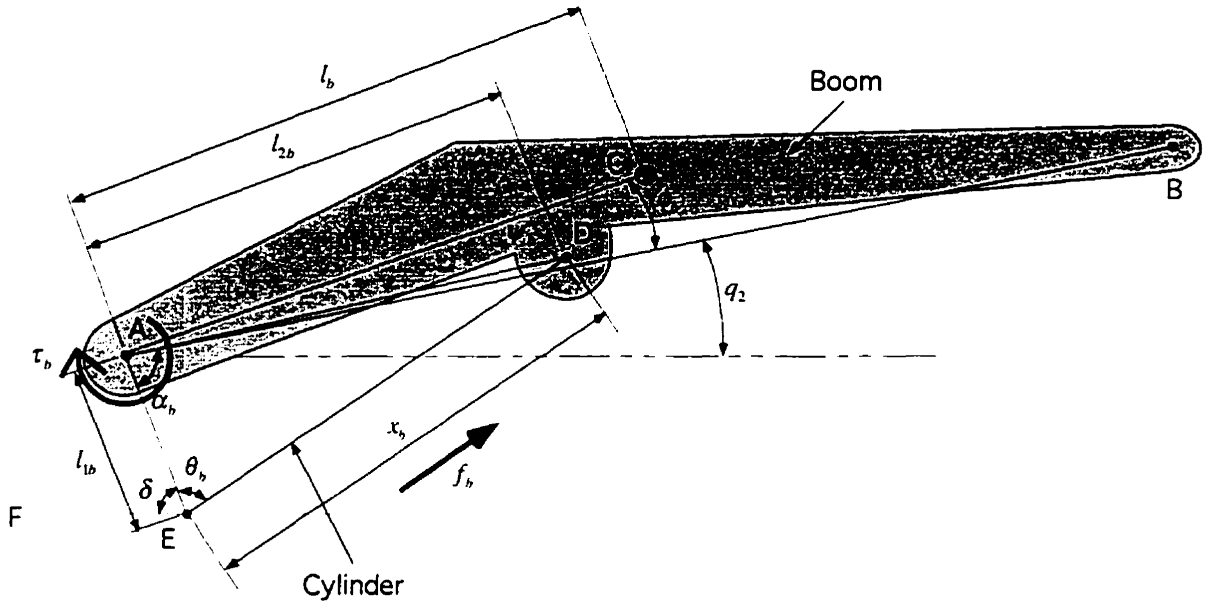


Figure 5.12: Schematic Diagram of Boom and its Connection (not to scale).

Similarly, as in the case of the stick, the force applied by the boom piston is found by taking moments with respect to the swing-boom joint (point A). Hence we get by inspection

$$f_b = \frac{\tau_b}{l_{1b}\sin\theta_b} \tag{5.74}$$

where, τ_b is the torque generated at the boom-base joint, and f_b is the force generated by the boom piston. From triangle ADE see Figure 5.12,

$$\cos\theta_b = \frac{l_{1b}^2 + x_b^2 - l_{2b}^2}{2 l_{1b} x_b} \tag{5.75}$$

Again from Figure 5.12, we get

$$\psi_b = \angle BAD, \quad \alpha_b = \angle EAD, \quad \phi_b = \angle BAC, \quad \delta = \angle AEF, \quad \theta_b = \angle AED$$

From triangle ADE,

$$\cos\alpha_b = \frac{l_{1b}^2 + l_{2b}^2 - x_b^2}{2 l_{1b} l_{2b}} \tag{5.76}$$

$$q_2 = \alpha_b - \delta - \psi_2 \tag{5.77}$$

$$x_b = \sqrt{l_{1b}^2 + l_{2b}^2 - 2 l_{1b} l_{2b} \cos(q_2 + \delta + \psi_b)} \tag{5.78}$$

By taking derivative of Eq. (5.78) we get,

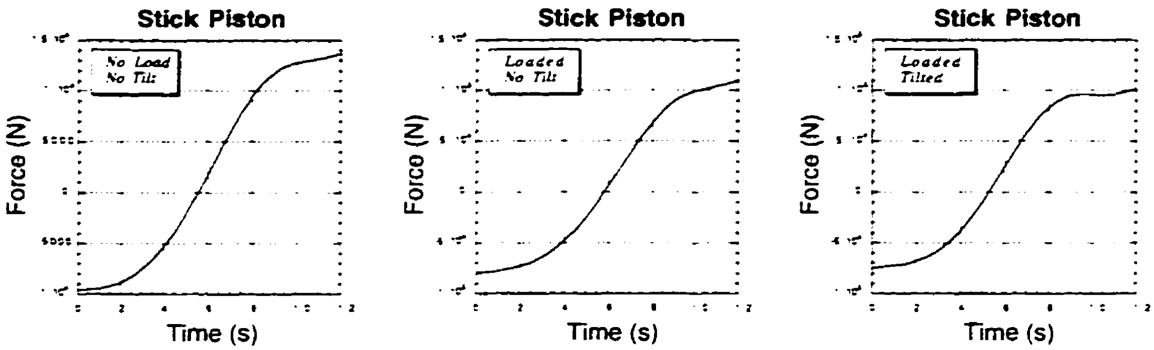
$$\dot{x}_b = \frac{l_{1b} l_{2b} \sin(q_2 + \delta + \psi_b)}{\sqrt{l_{1b}^2 + l_{2b}^2 - 2 l_{1b} l_{2b} \cos(q_2 + \delta + \psi_b)}} \dot{q}_2 \tag{5.79}$$

Table 5.1 displays parameters obtained from machine blueprints.

Table 5.1: Values of Parameters.

Stick Piston		**Boom Piston**	
l_{1s}	$1.934\ m$	l_{1b}	$0.563\ m$
l_{2s}	$0.884\ m$	l_{2b}	$1.786\ m$
l_s	$1.249\ m$	l_b	$1.994\ m$
ψ_s	2.5^o	ψ_b	0.31^o
ϕ_s	6.3^o	ϕ_b	4.18^o

The force profiles for the stick piston are computed using Eq. (5.66) and shown in Figure 5.13. The mapping from torque to force is also a function of manipulator configuration as the "moment arm" changes with the manipulator configuration. If we compare Figure 5.13 with Figure 5.7, we see changes in the corresponding profiles.



(a) (b) (c)

Figure 5.13: Force Histories at Stick Piston in Different Cases.

5.2.5 Pressure Profile

The expressions for the pressure drops in the stick cylinder, Δp_s, boom cylinder, Δp_b, and, swing motor, Δp_m are given below

$$\Delta p_s = \frac{f_s}{A_s} \tag{5.80}$$

$$\Delta p_b = \frac{f_b}{A_b} \tag{5.81}$$

$$\Delta p_m = \frac{\tau_m}{D} = \frac{\tau_m}{nD} \tag{5.82}$$

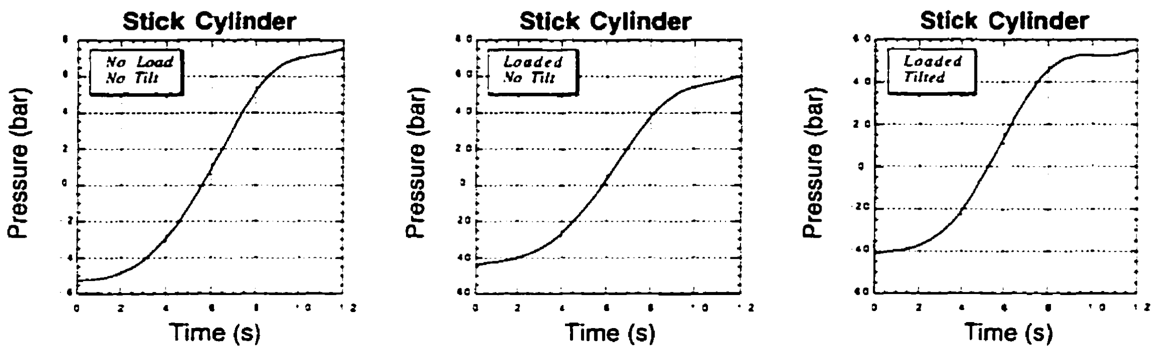
The pressure drops at the corresponding valves are given as,

$$\Delta p_{v,s} = p_{op} - |\Delta p_s| \tag{5.83}$$

$$\Delta p_{v,b} = p_{op} - |\Delta p_b| \tag{5.84}$$

$$\Delta p_{v,m} = p_{op} - |\Delta p_m| \tag{5.85}$$

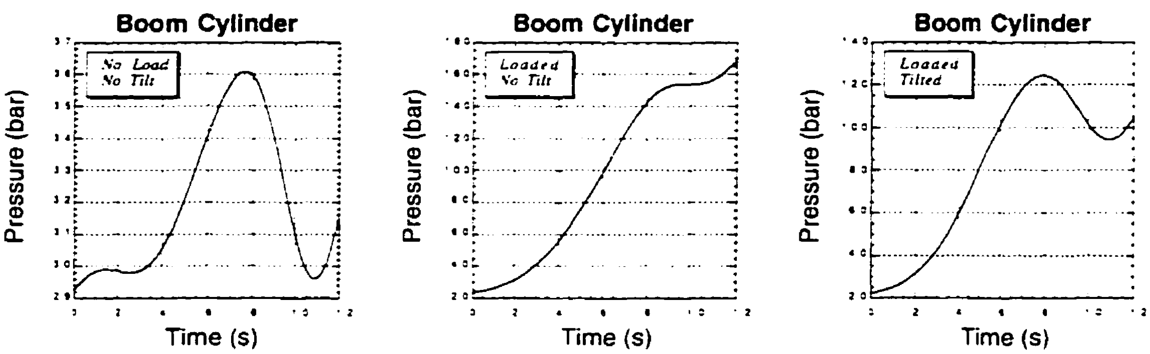
where p_{op} is the operating pressure. For the stick, boom and swing motor, the pressure profiles are given for the three cases in the following plots, see Figure 5.14 through Figure 5.16.



(a) (b) (c)

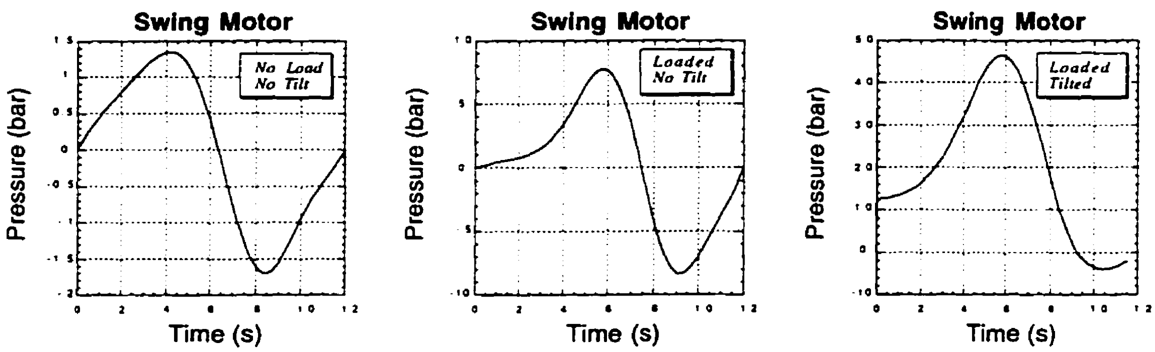
Figure 5.14: Pressure Drop Histories in Stick Cylinder in for the 3 Cases.

Due to Eqs. (5.80) - (5.82), the pressure drop profiles in the cylinders and motor are a scaled version of the force at the piston in the case of boom and stick, and the motor torque in the case of the swing.



(a) (b) (c)

Figure 5.15: Pressure Drop Histories in Boom Cylinder in Different Cases.



(a) (b) (c)

Figure 5.16: Pressure Drop Histories in Swing Motor in Different Cases.

5.2.6 Power Profile

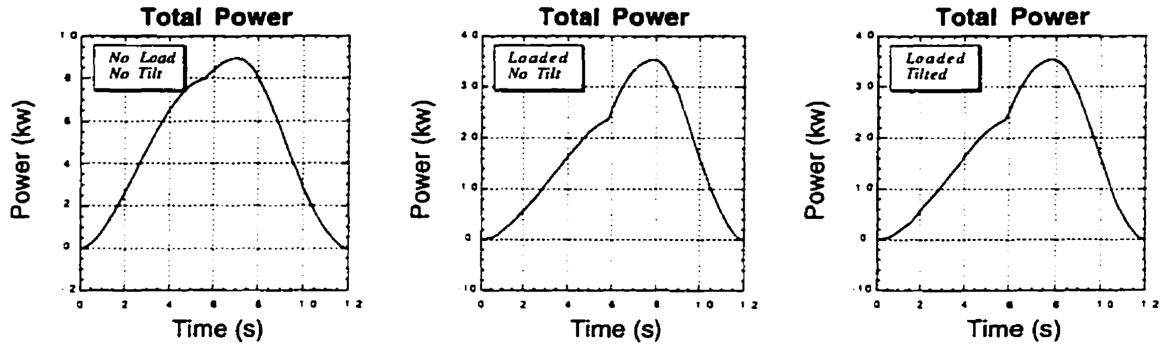
Note that the dynamic models obtained permit also either sizing of the system power supply (pumps) or checking whether the desired trajectory can be followed without exceeding the power capacity of the supply. The total power requirement is the sum of all individual power requirement for the powered joints. For example the power required for the stick is given by

$$P_s = |\tau_s \dot{q}_3| = |f_s \dot{x}_s| \tag{5.86}$$

where τ_s is the torque required to move the stick at an angular velocity of $\dot{q}_3$. Obviously, the total power required for a given trajectory can be obtained by

$$P_{total} = P_{sw} + P_b + P_s + P_{losses} \tag{5.87}$$

Based on the above equations, the total power, flow and all other variables can be plotted against time to permit easy evaluation of the system performance and requirements. The total power is obtained by adding the individual power requirement for swing boom and stick assuming there are no losses. The power requirements for the three considered cases are shown in Figure 5.17. It is clear that when there is no load the total power consumption is much lower than that when load of *1500 kg* is present at the tip of the stick.



(a) (b) (c)

Figure 5.17: Power Profiles in Different Cases.

5.2.7 Velocity Profile

The magnitude of the absolute velocity of the tip (end of the stick) is also important and for given joint trajectories. This magnitude does not depend on the load at the end of the stick or the tilt, see Figure 5.18.

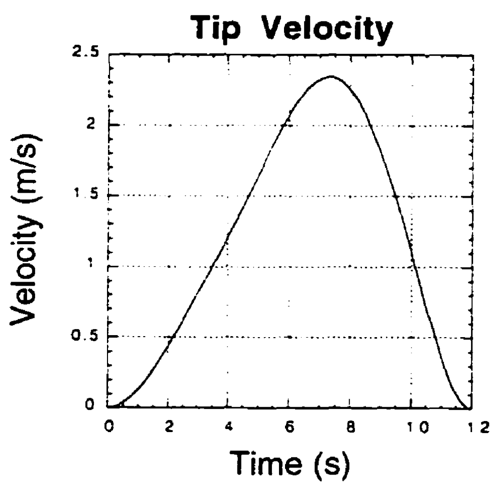


Figure 5.18: Velocity Profile at the End of the Stick.

5.3 Valve Sizing Methodology

An important application of the dynamic modeling is sizing of actuators. In the case of the experimental electrohydraulic machine, it has been decided not to replace the existing boom and stick hydraulic cylinders, and the swing motor. However, the need to select new proportional valves for the constant-pressure supply to replace the old load-sensing ones provided the first application for the derived dynamic models. According to typical industrial practice, proportional valves are selected based on a nominal load and duty cycle. However, no such nominal quantities exist for a manipulator arm whose configuration changes continuously, and may carry no load, or be loaded with a heavy tree. Therefore, a systematic methodology for valve sizing is needed.

A valve is properly sized when it can supply the demanded flow at the required pressure drop across it. Therefore to size a valve, flow and pressure requirements must be obtained as a function of time for a given task. Obviously, the task becomes more demanding when the manipulator is moving a heavy payload, or when it operates on a slope.

To this end, typical average as well as worst-case trajectories of the manipulator end-point were specified by observation of actual forestry machines. Using inverse kinematics relationships, these end-point trajectories were resolved at the actuator level, to result in trajectories for the swing angle, and the boom and stick displacements. Then, these can be used to obtain the flow requirements for all three actuated dofs. The flow through the valves for the three actuator is obtained from Eqs. (5.49) through (5.51). The pressure drop across the valves is obtained from Eqs. (5.83) through (5.85). If necessary, these estimates can be decreased by a 10% factor to allow for pressure drops in the transmission lines. Eqs (5.49)-(5.51) and (5.83)-(5.85) can be used to plot valve flow versus valve drop for the desired end-point trajectories. The resulting Q-Δp curve should lie below the valve pressure-flow characteristic at full valve opening, Q_V-Δp_V, typically a curve described by a relationship of the form,

$$Q_v = c\sqrt{\Delta p_v} \tag{5.88}$$

where Q_v is the flow through a valve, Δp_v is the pressure drop across the valve and c is the discharge coefficient. Normally, c is a function of the valve area which is a function of the applied voltage, therefore c is a function of the applied voltage. The discharge coefficient depends on the valve spool position which is a function of command voltage. If Q-Δp curve does not lie below the valve pressure-flow characteristic at full valve opening, this means that the pressure drop across the valve is less because pressure drop across the actuator is large, and the value is not able to provide the motion to the manipulator at the specified speed at a particular operating pressure. In this case a valve of larger capacity must be specified. For the selected valve, it has been found that c is equal to $16.4\,{}^{lit}/_{\min\sqrt{bar}}$ when Q_v is expressed in *liter/min.* and Δp_v is in *bar*. Figure 5.19 through Figure 5.22 show typical plots of such curves for the boom, stick, and swing, when the base is working in various conditions. We have shown the simulation for the loaded case with load at the end of the stick of *1500 kg*, which is the simulated mass of the head and a heavy tree. Again the trajectory duration for all of joints is 12 sec. The boom and stick move from minimum joint limit to maximum joint limit, while the swing moves from 0° to 90°. Since all plots lie under the valve characteristic, this valve can be used for driving all manipulator actuators along the desired trajectory.

Different candidate trajectories were tested with different loads at the tip of the stick to examine if the selected valve is adequate. It is clear that in the case of no load the pressure drops across all of the valves are very small. When a load is suspended at the end of the stick, the pressure drops across all the valves are quite significant. This is depicted in Figure 5.20.

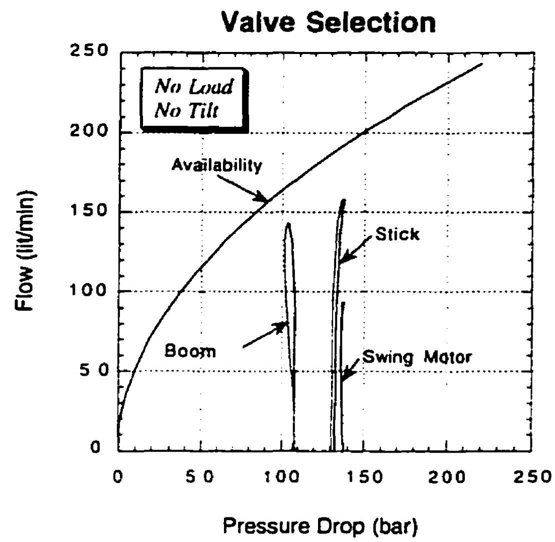


Figure 5.19: Pressure Drop Vs. Flow for First Case.

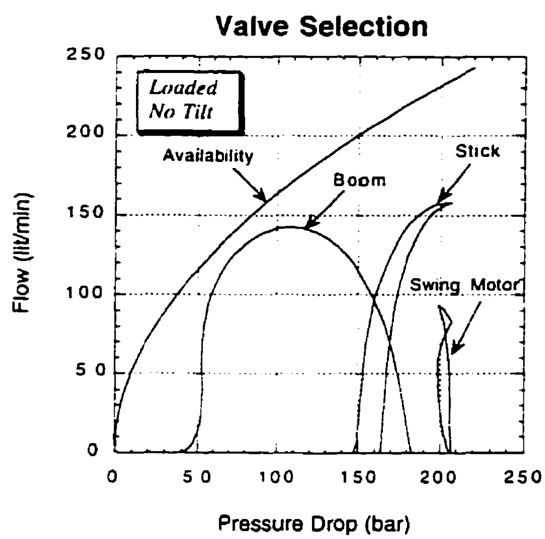


Figure 5.20: Pressure Drop Vs. Flow for Second Case.

When the base is tilted the gravity effect in the swing becomes dominant as shown in Figure 5.21. The pressure drop across the valve of the swing motor is larger in the tilted case than in the no tilt case, shown in Figure 5.20.

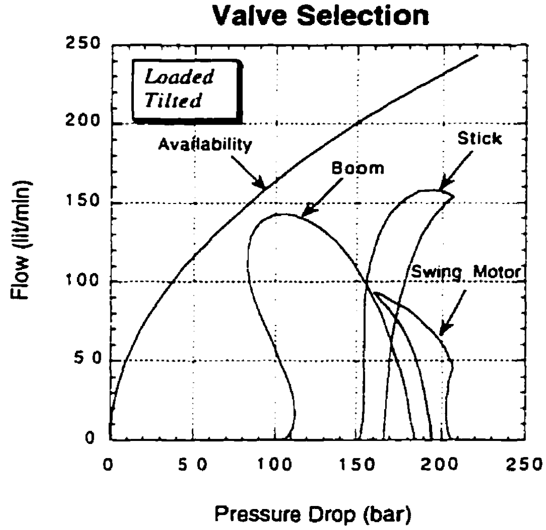


Figure 5.21: Pressure Drop Vs. Flow for Third Case.

Since in all the simulation runs, the Q-Δp curves are well below the valve characteristic curve, we conclude that the machine will be able to operate with the selected valve.

5.3.1 Addition of Check Valves

Many proportional valves have spring-loaded spools that will center the spool in case of signal loss. However, the selected valves do not have this feature, and this may cause safety problems. In addition, valve contamination can cause uncontrolled boom or stick motion. For this reason, the hydraulic circuit is modified by adding check valves on each of the ports of the actuator valves. This results in additional pressure drops across the check valves. The value of the discharge coefficient (k) for the check valve was calculated as $61 \, {}^{lit}\!/_{\min\sqrt{bar}}$ with pressure drop in bar and flow in lit/min.

The additional pressure drop is also incorporated in the program and simulation results are shown in Figure 5.22 for the second case (loaded and no tilt). Comparing Figure 5.22 with Figure 5.20, it is observed that the addition of the two check valves does not affect the $Q - \Delta p$ profiles much.

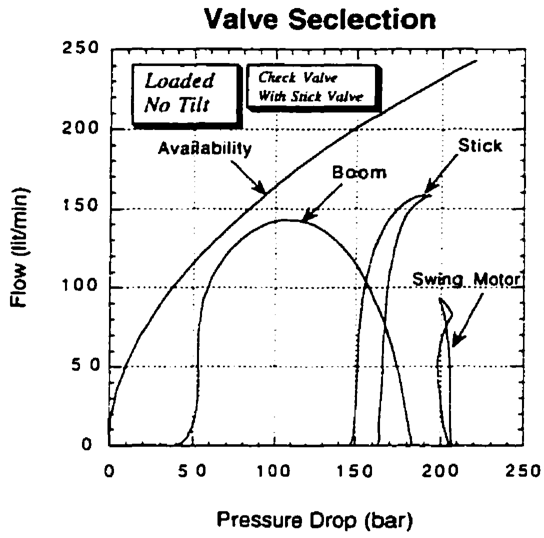


Figure 5.22: Pressure Drop Vs. Flow with Check Valve for Second Case.

6. Forward Dynamics, Simulations and Implementation Issues

In Chapter 5 we discussed inverse dynamics of the forestry vehicle where the input is the trajectory at each joint and the output is the set of joint torques. Based on these torques, the force in the case of the boom and stick, and the pressure drops in the corresponding cylinders, and the swing motor were calculated. In this chapter we will discuss the motions that result when the input is the force/torque at the three joints. Firstly the three dof system will be considered. Secondly the gimbals will be attached at the end of the stick, and finally the compliance at the base will be added resulting in an eight dof system. At first, various Simulink models are developed to simulate systems of various degrees-of-freedom. The Simulink models are also helpful to integrate the hydraulic controls and to validate the C code, which was developed later for faster execution speeds.

6.1 Forward Dynamics for Various DOF Systems

The three dof system consists of the swing, boom and stick. The simulation for the dynamics will be shown in detail for the three dof system in a Simulink model. As the dof increase, some Simulink blocks will be changed and a few more branches will be added. In the five dof system the links, pin and head/end-effector are attached at the end of the stick. A six dof model corresponds to a system where a three dof base compliance is introduced at the base. The eight dof system is the complete system of the forestry vehicle, including compliance, pin and end-effector links.

6.1.1 Simulink Model of 3 the dof System

In Figure 6.1 the complete Simulink model of the three dof system is shown.

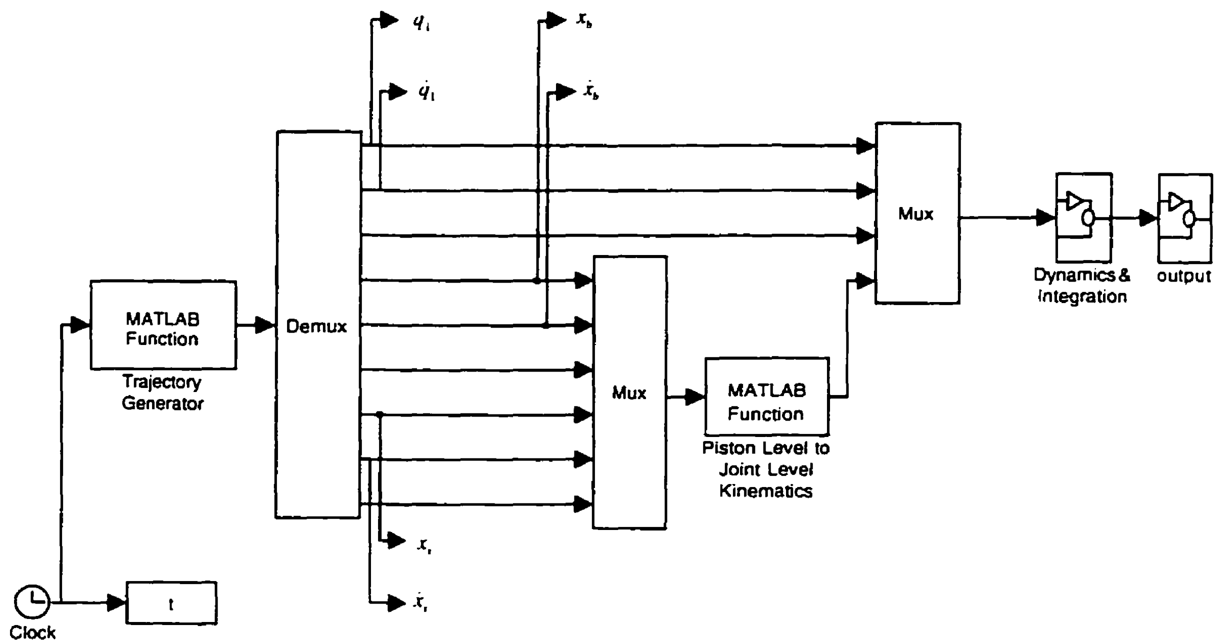


Figure 6.1: The Simulink Block Diagram of the 3 dof System.

This model is the test-bed model for all the simulations in Simulink and for all dof system. The whole system (see Figure 6.1 through Figure 6.3) is run in the same time frame, as a result the output must match the generated trajectories. The *trajectory generator block* produces desired boom, stick and swing trajectories. The closed form equation of the manipulator dynamics requires joint level information, so piston level information are converted to joint level information in the *piston level to joint level kinematics block* and fed to the *inverse dynamics block*. Figure 6.2 shows the dynamics and integration block.

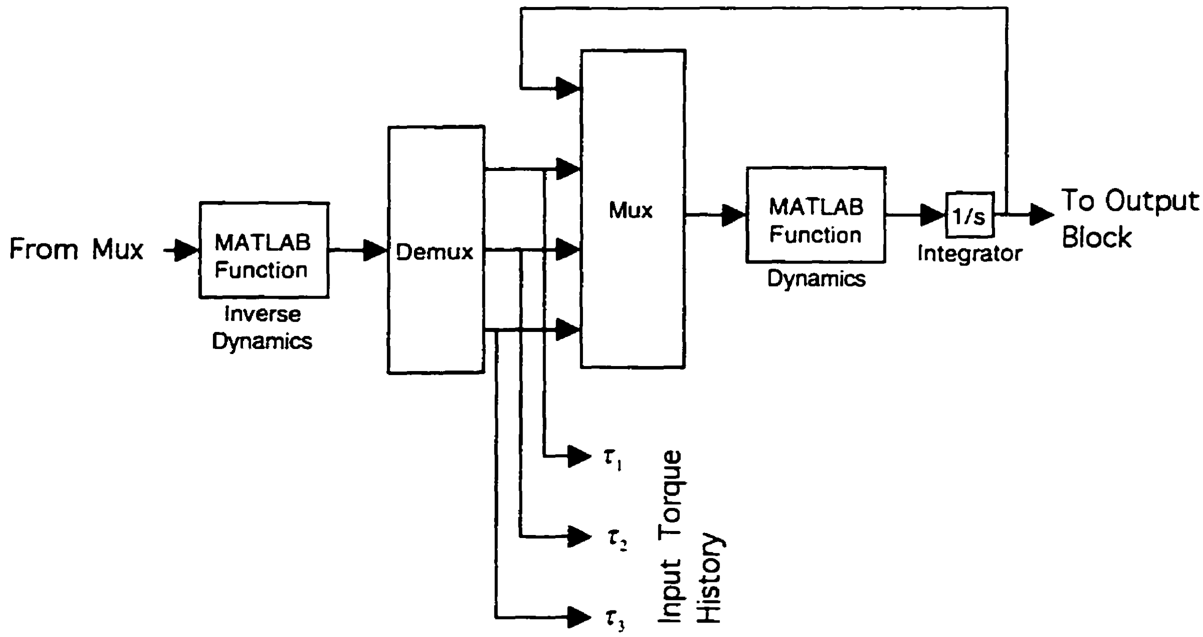


Figure 6.2: Simulink Block for *Dynamics and Integration*.

The *inverse dynamics block* contains the equations of motion for all the joints in symbolic form. The output of the inverse dynamics block is the torque history. This is the first step in order to generate the appropriate torque history for the system. In the second step the torque history is fed to the *dynamics block* of the system. The dynamics block contains all the equations of motion also found in the *inverse dynamics block*. However, the structure of the block is different than that of the inverse dynamics block. The *integrator* is placed after the dynamics block and it integrates the accelerations to compute the velocity and position information for each joint. The initial conditions of the integrator should be given in this block.

Figure 6.3 shows the output of the integrator, which includes joint level responses. This response is fed to the *Jacobian block* to get the output tip velocity components v_x , v_y and v_z with respect to the world frame and the velocity magnitude v (the magnitude of the tip velocity is more important than the components). The Jacobian block contains the transformation equations from joint space to Cartesian space, see Eqs. (2.53) through

(2.55). In order to observe the response at the piston level, all joint level responses are fed to the *joint level to piston level kinematics block.*

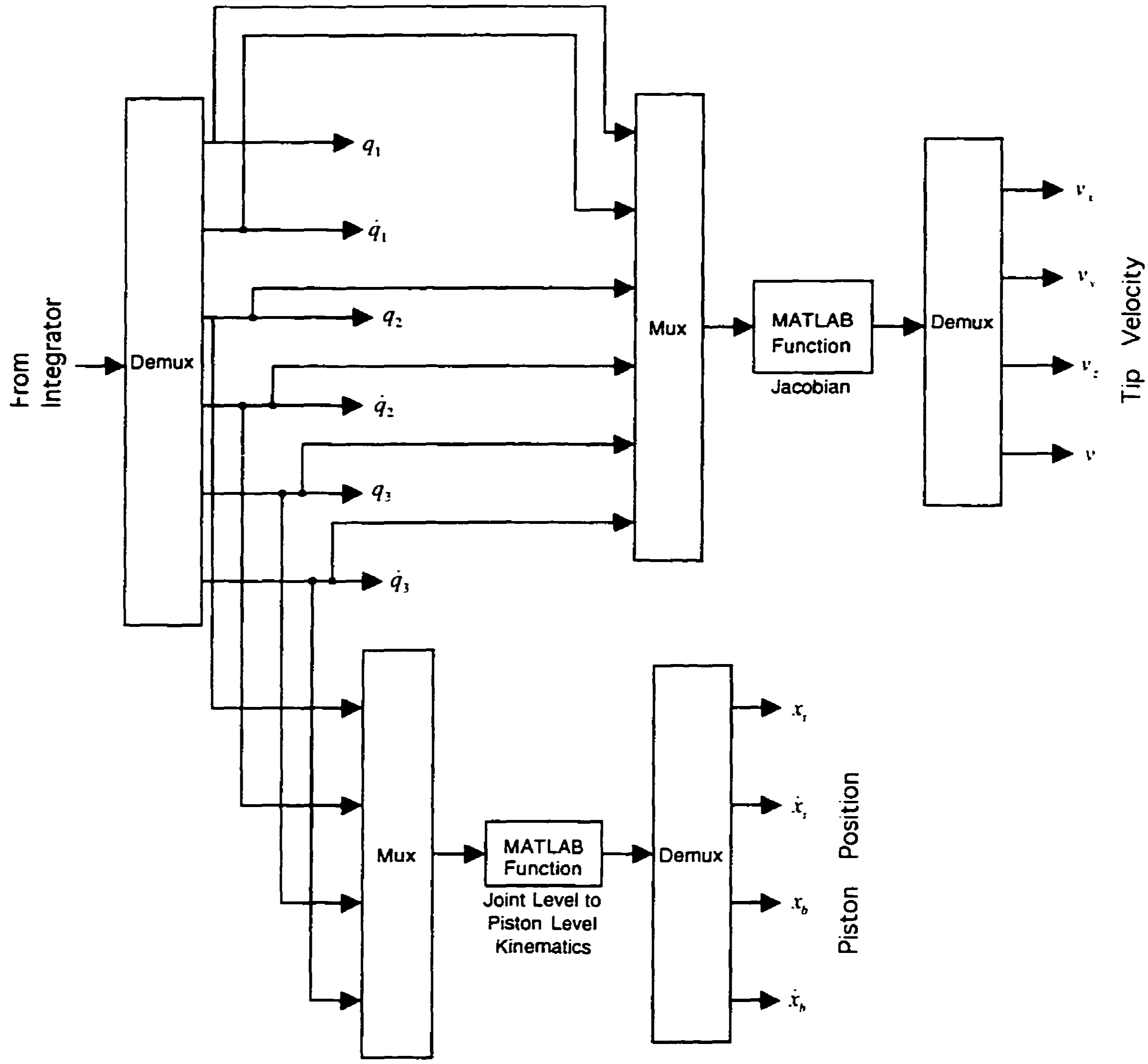


Figure 6.3: Simulink Block for *Output Block.*

In the case of an actual system with a hydraulic subsystem, the controller generates the torque-time history so that the end-effector follows operator commands. Here, we give torque/force input at different joints (as the case should be) and we observe the actual dynamic behavior of the system.

6.1.2 The 5 dof System

To simulate the forestry machine in a real time environment it has been decided to employ the five dof system, as the equations of motion can be run fast enough on the available Silicon Graphics workstation (CPU R4400). The Simulink Block diagram can be modified easily to simulate the whole system. All function blocks are modified to accommodate the five dof system. If we want to plot the movement of pendulums, then after the integrator block we need to add four additional output branches for position and velocity (q_4, $\dot{q}_4$, q_5 and $\dot{q}_5$). At the fourth and fifth joint there are no actuator torque inputs, therefore the Simulink block diagram is modified as shown in Figure 6.4.

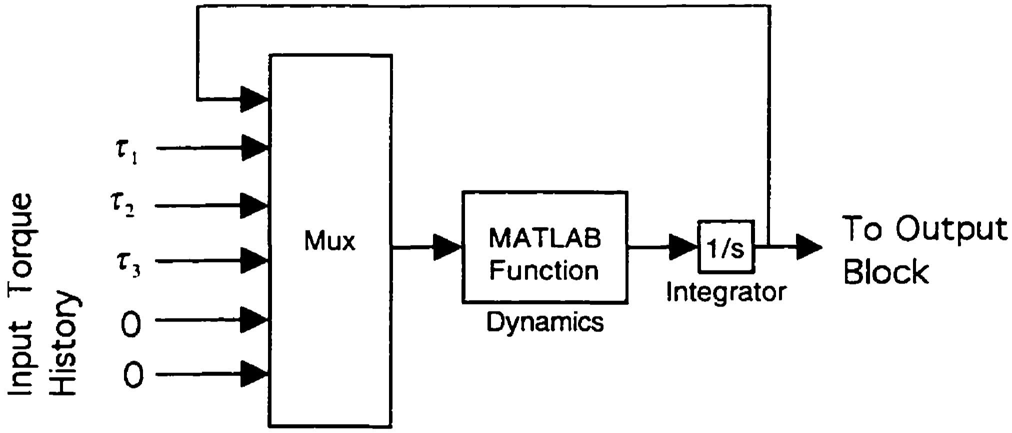


Figure 6.4: Simulink Block for the Dynamics of the 5 dof System.

6.2 Measures to Minimize Simulation Time

In order to simulate the forestry machine in a real time environment we need to minimize the simulation time. The simulation time can be minimized at the hardware or software level. At the hardware level, obviously a machine with a faster CPU results in a faster the simulation. Our interest is in software level and is described step by step next. At first, we need to have the "plant" dynamics as efficient as possible. This is done by using special functions in Mathematica [91]. Once the dynamics are obtained symbolically, the numerical values of all the parameters are introduced (such as mass, link length, link twist, inertia properties etc.).

At first we made the simulation in the Matlab environment faster by compiling all the Simulink blocks using a utility called CMEX. CMEX results in approximately 18 times faster simulation for the 3 dof case. Another problem is that to get the simulation results we need to run Matlab and Simulink. To avoid this, and to make the simulation even faster, the whole program is written in C code with minimum number of function calls. Unlike in Matlab, where various integration routines are built in, these need to be developed for the code in C. The Runge-Kutta-Nyström numerical integration algorithm is developed here as this is an accurate fourth order algorithm and is extensively used in many practical applications. Table 6.1 gives the comparison in execution time for the systems of various complexity in SGI (R4400) machine.

Table 6.1: Simulation Time Comparison in Different Systems

System dof	Simulation time for one time step, *dt (ms)*	
	Matlab	C
3	80	0.23
5	712	1.50
8	Not Available	11.87

The scheme of Runge-Kutta-Nyström algorithm is given below:

Implemented Runge-Kutta-Nyström Method:

$$k_1 = \frac{h}{2}\ddot{q}[q, \dot{q}] \tag{6.1}$$

$$k_2 = \frac{h}{2}\ddot{q}\left[q + \frac{h}{2}\left(\dot{q} + \frac{k_1}{2}\right), \dot{q} + k_1\right] \tag{6.2}$$

$$k_3 = \frac{h}{2}\ddot{q}\left[q + \frac{h}{2}\left(\dot{q} + \frac{k_1}{2}\right), \dot{q} + k_2\right] \tag{6.3}$$

$$k_4 = \frac{h}{2}\ddot{q}\left[q + h(\dot{q} + k_3), \dot{q} + 2k_3\right] \tag{6.4}$$

$$q_{i+1} = q_i + h\left\{\dot{q}_i + \frac{1}{3}(k_1 + k_2 + k_3)\right\} \tag{6.5}$$

$$\dot{q}_{i+1} = \dot{q}_i + \frac{1}{3}(k_1 + k_2 + k_3 + k_4) \tag{6.6}$$

$$t_{i+1} = t_i + h \tag{6.7}$$

where k_1, k_2, k_3, and k_4 are intermediate variables, q denotes the joint variable, t denotes the time variable, i represents the time counter of the integrator and h is the step size. From Eq.(3.71) in Chapter 3 we get the general equation of motion of a robot as follows:

$$\ddot{q} = M^{-1}\left(\tau - V(q,\dot{q}) - G(q)\right)$$

As the dof of the system increases, the size of the mass matrix M increases too. In general, to compute the accelerations, we need to invert the mass matrix. The matrix can be inverted symbolically for a 5 dof case. But as the size of the mass matrix increases, the symbolic inversion becomes gigantic, and for an 8 dof model the mass matrix can not be inverted symbolically on an SGI (R4400) machine as it takes a lot of resident memory.

But as the mass matrix is always symmetric and positive definite we do not need to invert the mass matrix to get the accelerations if we use Cholesky's method. The complete program of the Cholesky's method is implemented in C code to have the fastest possible simulation run. This method improves the computation efficiency two times compared to a standard matrix inversion method. The scheme of Cholesky's method is given below:

Implemented Cholesky's Method:

The general equation of motion can be written as,

$$M\ddot{q} = \tau - V - G \tag{6.8}$$

Eq.(6.8) can be written in the following form:

$$A\ddot{q} = b \tag{6.9}$$

As A is symmetric and positive definite, it can be written as a product of two triangular decomposed matrices transposed to each other as below:

$$A = LL^T \tag{6.10}$$

The elements of the L matrix are given below:

$$m_{11} = \sqrt{a_{11}} \tag{6.11}$$

$$m_{jj} = \sqrt{a_{jj} - \sum_{s=1}^{j-1} m_{js}^{\,2}} \tag{6.12}$$

$$m_{j1} = \frac{a_{j1}}{m_{11}} \tag{6.13}$$

$$m_{jk} = \frac{1}{m_{11}}\left(a_{jk} - \sum_{s=1}^{k-1} m_{js} m_{ks}\right) \tag{6.14}$$

Solving for a dummy variable y:

$$Ly = b \Rightarrow y = L^{-1}b \tag{6.15}$$

Solving for acceleration ($\ddot{q}$):

$$L^T\ddot{q} = y \Rightarrow \ddot{q} = L^{-T}y \tag{6.16}$$

6.3 Dynamic Response Using Forward Dynamics

The dynamic behavior of various systems is studied based on torque/force inputs generated by a set-point feedforward controller. The focus here is to analyze system transient and steady state response for various commands. Also of interest is tracking performance degradation due to tire compliance.

The controller is designed mainly to provide the gravity terms required to hold the three manipulator joints in static equilibrium at some desired configuration. This set of gravity torque is computed off-line and added to the feedback controller, shown in Figure 6.5.

This controller is basically a PD type controller with gravity compensation, for improved tracking and for reducing the static errors.

As shown in Figure 6.5, the 3×1 gravity compensation vector $\hat{G}$ is evaluated at the 3×1 set-point vector $\hat{\mathbf{q}}_d$, which includes desired swing, boom, and stick angles. However, $\hat{G}$ is a function of all the dofs $\mathbf{q}$, and therefore, nominal roll, pitch, bounce and pendulum angle values are used for the 8 dof system. For a 5 dof system, $\hat{G}$ is a function of joint angles (q_1 to q_5). Due to this approximate computation of the gravity term $\hat{G}$, it is expected that the steady-state error given by

$$\mathrm{E}_{ss} = \hat{\mathbf{q}}_d - \hat{\mathbf{q}}_{ss} \tag{6.17}$$

will be small but not exactly zero, even if all system parameters are exactly known. For 5 dof system the vector, τ in *Forestry Machine Block* see Figure 6.5 is written as

$$\tau = \begin{bmatrix} \tau_a & 0_{2\times 1} \end{bmatrix}^T \tag{6.18}$$

where τ_a denotes the actuator torque viz. swing, τ_{sw}, boom, τ_b and stick, τ_s as shown below

$$\tau_a = \begin{bmatrix} \tau_{sw} & \tau_b & \tau_s \end{bmatrix}^T \tag{6.19}$$

Similarly for 8 dof system the vector, τ is written as

$$\tau = \begin{bmatrix} 0_{3\times 1} & \tau_a & 0_{2\times 1} \end{bmatrix}^T \tag{6.20}$$

The elements of the diagonal control matrices K_P, and K_V, shown in Figure 6.5, are computed by

$$k_{P_i} = \omega_i^2 m_{ii} = (2\pi v_i)^2 m_{ii} \tag{6.21}$$

$$k_{V_i} = 2\zeta_i \omega_i m_{ii} = 4\pi \zeta_i v_i m_{ii} \tag{6.22}$$

where m_{ii} corresponds to diagonal elements of the mass matrix, ζ is the damping ratio, and v the frequency of the controller.

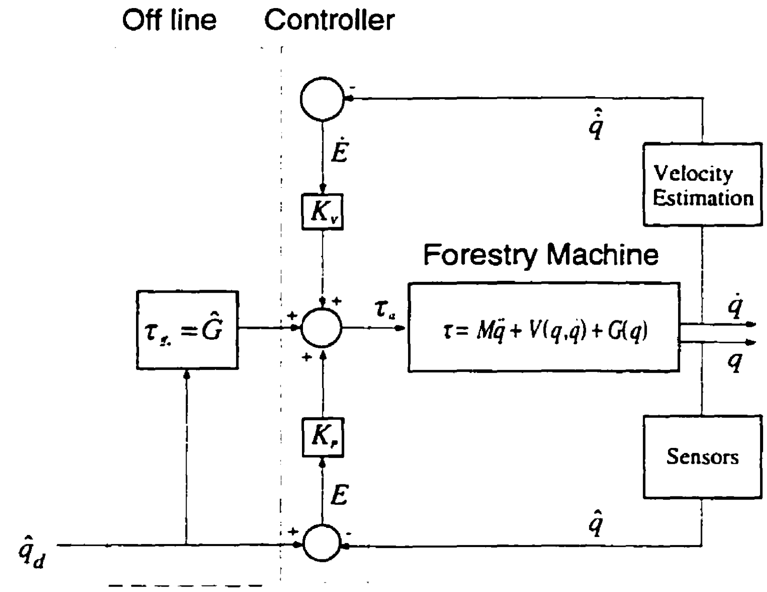


Figure 6.5: Set-point Feedforward Controller.

Finally, the equation for the applied torque is given by,

$$\tau_i = k_{P_i} e_i + k_{V_i} \dot{e}_i + \tau_{ff_i} \tag{6.23}$$

where τ_{ff_i} is the gravity compensation feedforward term. Note that this controller is not applicable as such, since in general, in a hydraulic system it is not possible to specify actuator torque/forces. However, it can be used to evaluate the developed models, and result in better understanding of system behavior.

6.3.1 Simulation for the 5 dof System

In this section the simulation for the 5 dof system is done and results are shown. An initial error of 10° is given as a command to each joint, and system response is obtained by simulation. The input to the program is the damping ratio and frequency of the controller at each joint (for gain calculation), the initial and desired conditions (position, velocity and acceleration) for each joint, the step size and total simulation time. As the viscous friction model is incorporated at each joint, the coefficient of friction (b) at each joint should be given as input. Depending on the desired conditions, the feedforward torque for the three joints are calculated off-line. The position and velocity gains are also dependent on the desired conditions as the corresponding mass matrix elements are validated with the desired conditions. The Table 6.2 shows the actual data used for the simulation.

Table 6.2: Input Data for the Simulation.

	b	Initial Conditions			Desired Conditions			ζ	ν
	$N.m.s/rad$	q_i (°)	$\dot{q}_i$(°/s)	$\ddot{q}_i$ (°/s^2)	q_d (°)	$\dot{q}_d$ (°/s)	$\ddot{q}_d$ (°/s^2)		Hz
Swing	0	0	0	0	10	0	0	1	.15
Boom	0	0	0	0	-10	0	0	1	.34
Stick	0	0	0	0	-10	0	0	1	.34
Pin	1500	-90	0	0	-70	0	0		
Head	1500	0	0	0	0	0	0		

Step Size	.01
Final Time	20 s

The position and velocity gains for each joint (swing, boom and stick) are also calculated off line. The desired angle for the pin (shown in a shaded box) results from Eq. (6.24) as the pendulum should remain vertical at steady state due to gravity.

$$q_{4_d} = -\left(\tfrac{\pi}{2} + q_{2_d} + q_{3_d}\right) \tag{6.24}$$

where, the subscript d denotes the desired condition.

Table 6.3 shows the output data for the simulation, including the time step used.

Table 6.3: Output Data for the 5 dof System.

	Calculated Gains		Calculated Torque	
	Position	Velocity	Initial $N.m$	Feedforward $N.m$
Swing	91590	194360	0	0
Boom	552740	517480	140825	136544
Stick	167314	156640	58800	55608

Simulation Step dt	1.5 ms

Figure 6.6 shows how the swing, boom and stick move from initial position to the desired position. From Figure 6.7 we observe that the velocity for the three powered links settles to zero as per the desired condition.

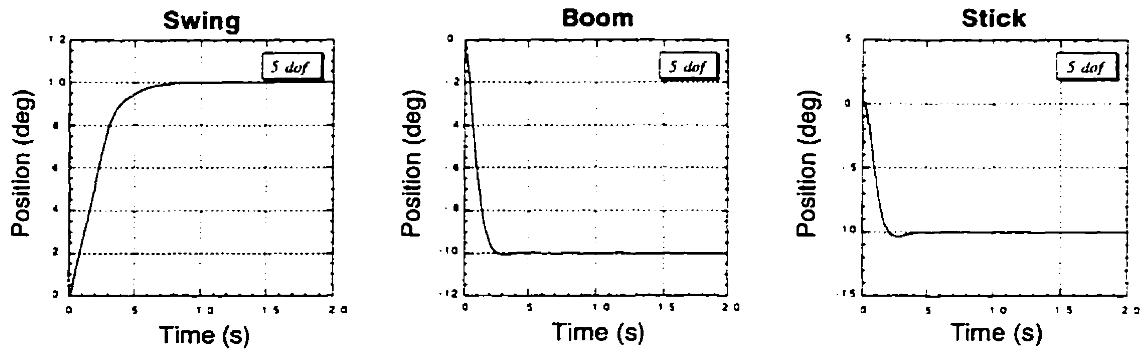


Figure 6.6: Positions of Powered Links.

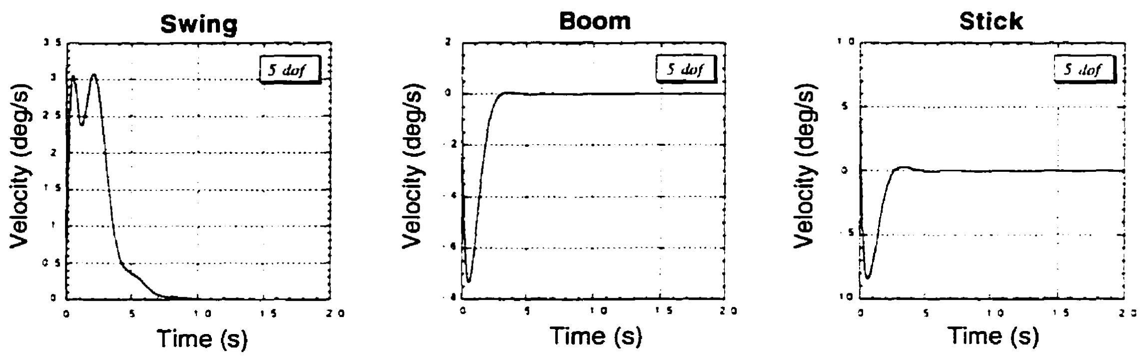


Figure 6.7: Velocities of Powered Links.

The pin automatically settles down to its stable position and end-effector will always settle to zero angle as the base is horizontal. The two pendulums do not swing forever, as joint friction, incorporated into the model, slows them down. Figure 6.8 shows the motion of pin and end-effector after getting command from the controller.

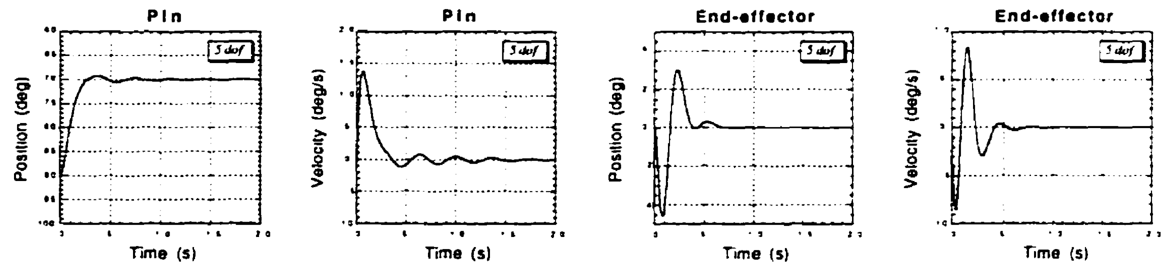


Figure 6.8: The Motion of Gimbals.

Figure 6.9 depicts joint tracking error performance. The observed overshoot occurs because of dynamic coupling, and because the feedforward gravity term is computed at the desired final position only. Again for the simulator design there must be a trade off, if both the gains are increased then we need a smaller time step in numerical integration routine for stability reasons. In such case, it may be difficult to achieve real-time simulation.

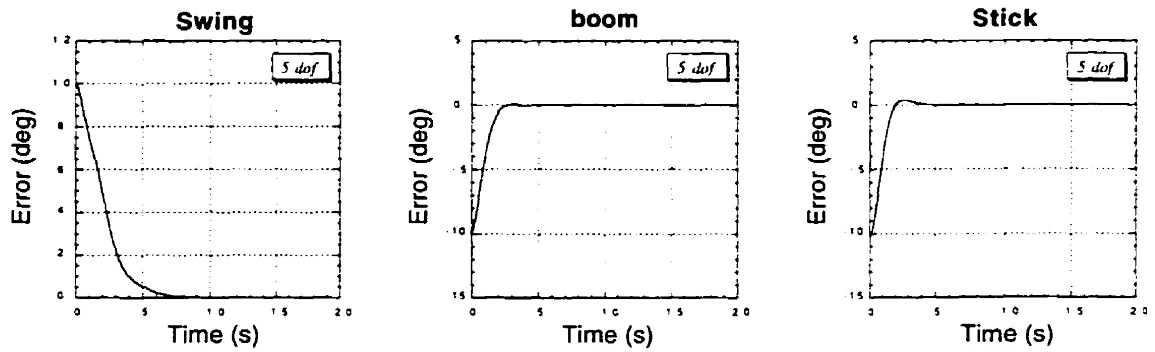


Figure 6.9: Errors Dynamics in Powered Links.

Figure 6.10 shows typical actuator applied torques for a set-point command in the swing, boom, and stick angles, $\hat{\mathbf{q}}_d$. Note that these are smooth, and therefore valid actuator torques.

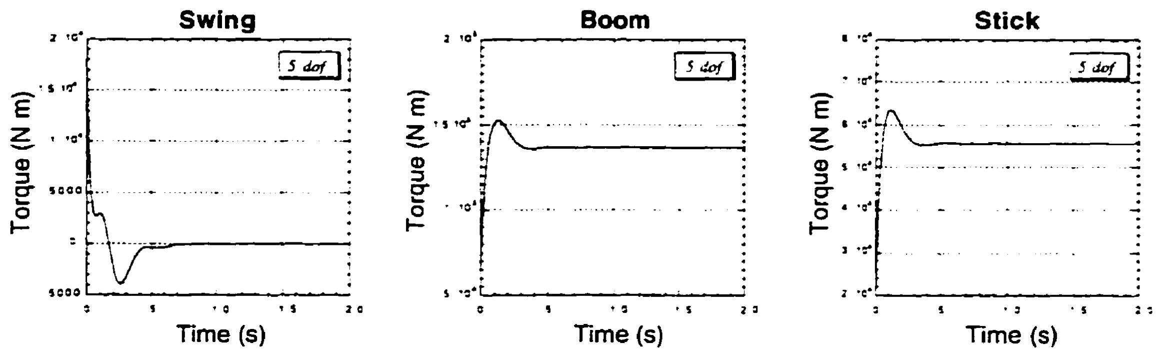


Figure 6.10: Applied Actuator Torque for the 5 dof System.

6.3.2 Simulation for the 8 dof System

In this section, the model includes tire compliance. The compliance effect is described by roll, pitch and bounce. The parameters for stiffness and damping are obtained by experiments and discussed in Chapter 8. The input data for 8 dof system is given in Table 6.4.

Table 6.4: Input Data for Simulation

	b	Initial Conditions			Desired Conditions			ζ	ν
	$^{N\,m\,s}/_{rad}$	q_t (°)	$\dot{q}_t$(°/s)	$\ddot{q}_t$ (°/s²)	q_d (°)	$\dot{q}_d$ (°/s)	$\ddot{q}_d$ (°/s²)		Hz
Swing	0	0	0	0	10	0	0	1	.15
Boom	0	0	0	0	-10	0	0	1	.34
Stick	0	0	0	0	-10	0	0	1	.34
Pin	1500	-90	0	0	-70	0	0		
Head	1500	0	0	0	0	0	0		
Bounce	-	-.104	0	0					
Roll	-	0	0	0					
Pitch	-	0	0	0					
Step Size	.01								
Final Time	20 s								

Table 6.5: Output Data for the 8 dof System.

	Calculated Gains		Calculated Torque $(N.m)$	
	Position	Velocity	Initial	Feedforward
Swing	91593	194366	0	0
Boom	552742	517480	140825	136544
Stick	167314	156640	58800	55608

Simulation Step dt	11.87 ms

As we are not controlling the bounce, roll and pitch, a desired conditions for the compliant base do not apply. The initial conditions for the base are chosen such that the system is initially at equilibrium. From the initial configuration the forces and torques at the base are calculated, and the deflections in bounce, roll and pitch are calculated and supplied as initial conditions. Another approach is to let the simulation run for a while with any initial condition for compliance and observe the steady state for a particular manipulator configuration. In the case of successive simulation runs, the initial conditions are updated automatically.

Figure 6.11 and Figure 6.12 show the response for the three powered links.

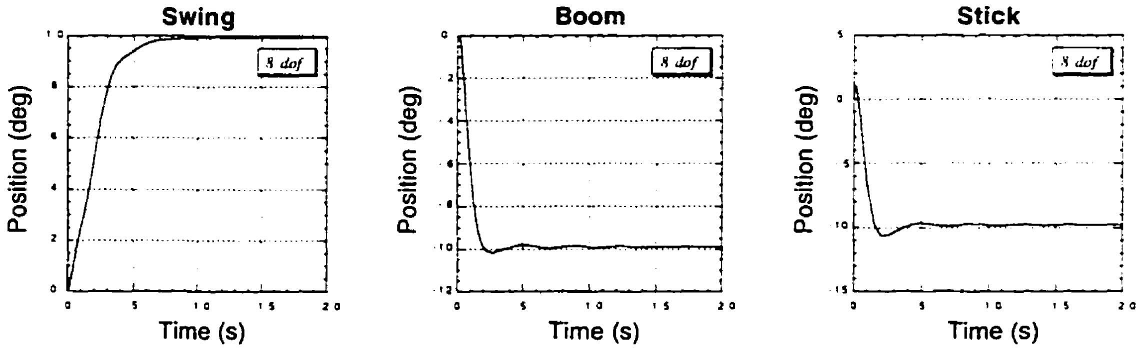


Figure 6.11: Positions of Powered Links.

For the swing there is a steady state error, as gravity compensation can not be applied to the base compliance. This kind of behavior is absent in the case of 5 dof system. The steady state error can be decreased by increasing the position gain. This can be done by increasing the bandwidth of the corresponding controller.

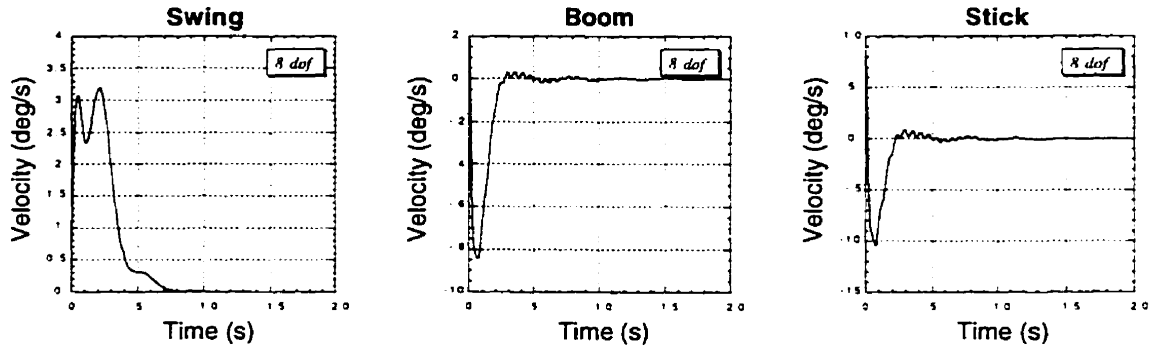


Figure 6.12 Velocities of Powered Links.

The angle histories of the Hooke assembly are shown in Figure 6.13. As expected, since these links are not actuated, their response is quite oscillatory. However, eventually this oscillation dies out due to friction at the joints. The desired position of the pin will come automatically from the desired position of boom and stick. Due to compliance effect desired position for the pin will be changed from 5 dof case as described in Eq. (6.24). The modified equation is shown in Eq. (6.25). The swing movement affects the end-effector movement, if the swing does not move the end-effector would not move (planar case).

$$q_{4_d} = -\left(\tfrac{\pi}{2} + q_{2_d} + q_{3_d} - q_{p_i}\right) \tag{6.25}$$

where, q_{p_i} is the initial pitch angle.

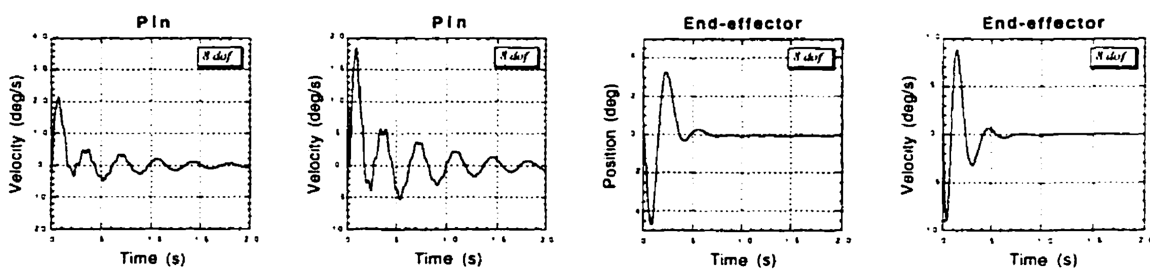


Figure 6.13: The Motion of Gimbals.

Figure 6.14 and Figure 6.15 depict the base pitch, roll and bounce motion. Although these are relatively small, their effect at the end-point is not negligible. This is due to the length of the manipulator arm.

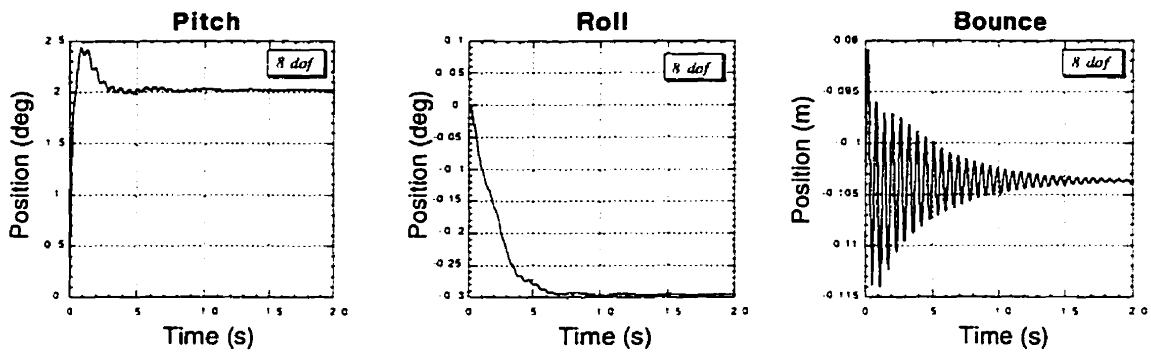


Figure 6.14: Base Position & Orientation Due to Compliance.

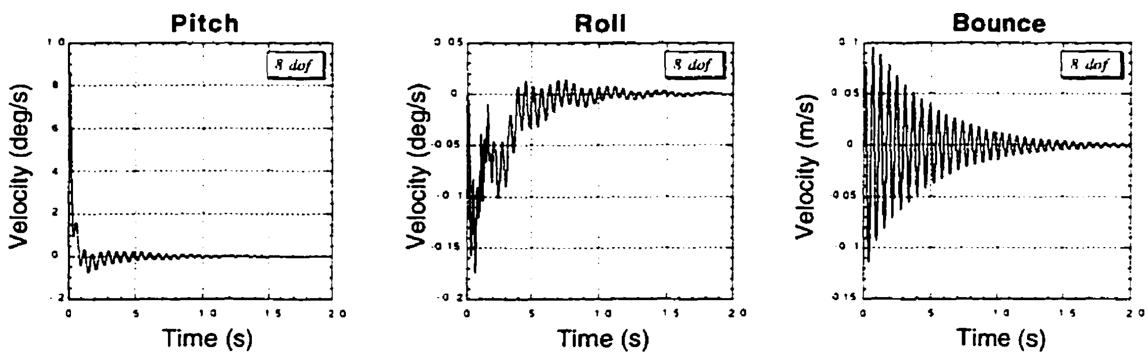


Figure 6.15: Base Velocities Due to Compliance.

Figure 6.16 shows joint tracking error performance for the 8 dof system. Unlike the 5 dof system, there are small non-zero steady state errors, due to the effects of base compliance, and to the lack of compensation for it. The applied actuator torque are depicted in Figure 6.17. Since the controller uses feedback from the joint sensors, the boom and stick torque profiles include higher frequency components, compared to Figure 6.10. These frequencies are due to the coupling between base compliance and joint motion, an effect missing from the 5 dof system.

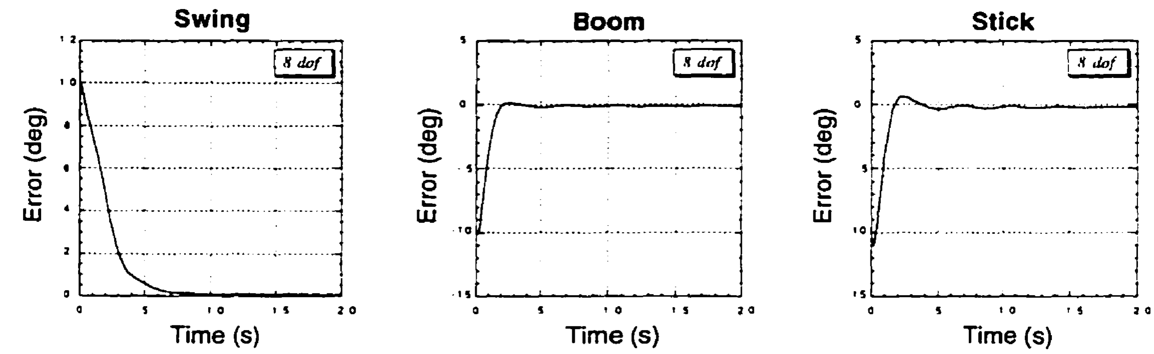


Figure 6.16: Errors Dynamics in Powered Links.

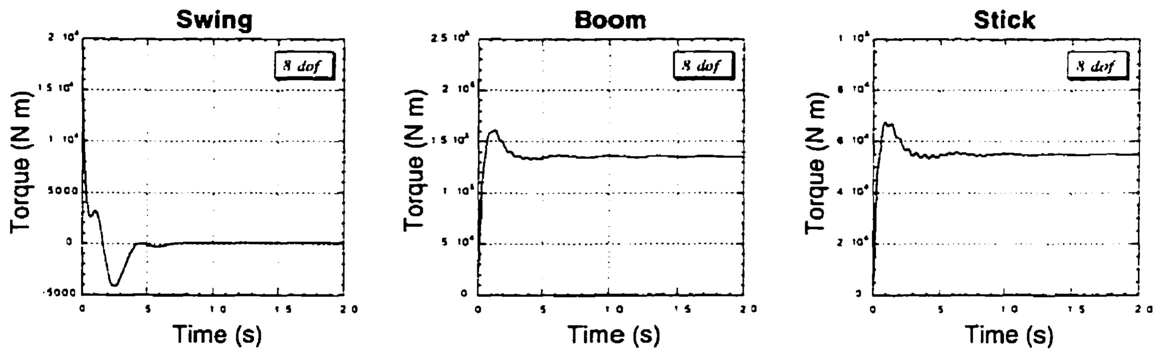


Figure 6.17: Applied Actuator Torque for the 8 dof System

7. Conclusions and Future Work

This thesis is primarily concerned with the dynamic modeling of a forestry vehicle. The dynamic models have been used to develop a training simulator, to size system valves, and to develop control algorithms. The training simulator is a visual graphic simulator, developed in Silicon graphics workstation interfaced with a joystick for operation. The actuator valves are sized based on the dynamics of a simplified system. A computed torque controller was designed to observe dynamic response of the system (a mathematical model of the hydraulic subsystem is necessary to design an actual controller) .

The complete dynamic model is developed in three stages of increasing complexity namely, (a) a three-degrees-of-freedom system, (b) a five-degrees-of-freedom system, and finally (c) an eight-degrees-of-freedom system. In the three-degrees-of-freedom system the links are the swing, the boom and the stick. In the five-degrees-of-freedom system two pendulums (gimbals) namely the pin and the end-effector, were attached at the end of the stick. The final model is an eight-degrees-of-freedom system where, the five-degrees-of-freedom system is mounted on a vehicle which rests on four tires. The tires are modeled as spring-damper systems. The tires introduce compliance to the base, and result in base motion such as roll, pitch, and bounce. The yaw effect is neglected as the swing does not rotate at a high speed.

The dynamics were developed using a Newton-Euler iterative algorithm. This algorithm is chosen as it is easy to implement by computer program and it requires a smaller number of computations. In the forward iteration, manipulator kinematics were employed while the dynamics were developed in backward iteration. The dynamics were found in a symbolic closed form solution of a manipulator.

To get the dynamic response of a system the system dynamics of a manipulator may be implemented either in numeric iterative form or in symbolic closed form. For a fixed-base manipulator in which all degrees-of-freedom are actuated both forms (numeric or symbolic) can be applied. The numeric iterative form, can only be applied in case of a fixed-base manipulator, where the desired trajectories for all joints, or degrees-of-freedoms, are used to calculate numerically the forces and torques necessary to cause the desired motion. The numeric iterative form can not be applied in the case of a manipulator mounted on a compliant base, since the base is not actuated, and its position, velocity and acceleration will depend on how fast the arm moves, the load being manipulated, etc. However, if this formulation is applied symbolically, then it results in a closed set of symbolic equations of motions, which is not subject to this problem.

Joint friction was assumed to be described by a viscous friction model, and appropriate terms were added to the velocity terms in the equation of motion. The dynamics of all three models were optimized for execution time by applying trigonometric identities, and were written in matrix form.

To carry out simulations, parameter values are needed. Parameters like lengths, masses and angles were found from industrial drawings or by simple direct measurements (like weighing, measurement by scale etc.) and trigonometric calculations. But parameters like the location of a center of mass, link moments of inertia, products of inertia of a link could not be estimated from blueprints. In the case of the boom and stick, pendulum experiments were carried out to measure moments of inertia. However moments of inertia are very sensitive to the time period of oscillation (they are proportional to the square of the time period of oscillation). Therefore, errors in obtaining the period of oscillation result in substantial errors in calculating the moment of inertia. Moreover, swinging a body with respect to a single axis is also a difficult task. Fortunately, solid modeling techniques can be used to compute moments and products of inertia very accurately. We used Auto CAD with the Advanced Modeling Extension package to estimate mass properties of all the links and the location of the center of mass. Note that solid models were refined to the point that both the estimated and measured total mass and moment of inertia were in agreement.

Another set of parameters is required for the base compliance model, attributed mostly to the tires. The tires are modeled as a spring and damper system, and their stiffness and damping ratio were found by static load-deflection and drop experiments respectively. It has been found that the tire behaves like a linear spring in the region of loads of interest. In the case of drop experiments, the tire must be in contact with a slab throughout the test, otherwise obtained results will not be valid due to the physics of collisions.

As all the links are driven by hydraulic power the valve characteristic and its sizing become important. Some typical trajectories (trapezoidal, cubic and quintic) were planned, and simulations using inverse dynamics were carried out. The flow through a valve, and the pressure drop across it (obtained from the inverse dynamics calculations) were plotted. This plots were compared with the provided valve characteristic curve to test if the valve under consideration is capable of performing the task. This methodology resulted in useful valve sizing results. The total power consumed for the system was also simulated for typical trajectories.

The model becomes more complicated as the degrees-of-freedom increase, and simulations take substantial amount of time. As a remedy, the codes were converted in compiled C code, and were integrated using a customized numerical integration routine (Runge-Kutta-Nyström algorithm). To get the accelerations of all the links, we need to invert the system's mass matrix. Due to symmetry and positive definiteness of the mass matrices involved in the dynamic equations of all models, matrix inversion is avoided by applying Cholesky's method. A customized source code for Cholesky's method was developed and written in C. As all the codes are on line (no libraries are called), the execution time became substantially faster compared to Simulink execution, even with the CMEX option.

In order to observe the dynamic behavior of all the models of the system a computed torque controller was developed. The position and velocity gains were chosen by specifying a closed-loop frequency and damping ratio for all the powered joints. Since a detailed friction model of the pendulum joints was not available, the damping coefficients for these joints were chosen by observing the decay of pendulums' oscillations. Obtained simulation results show good agreement with observed vehicle response.

7.1 Future Work

The complete base kinematics was developed in Chapter 2, and it allows calculations of the bogie angles from the two linear gauge readings. However, the bogie angles are not incorporated in the dynamic models. The model can be extended by introducing one degree-of-freedom between the platform and the swing and adding a transformation matrix from platform to swing, whose elements should be a function of bogie angle. Another important factor is the forward motion of the vehicle; this can be very easily incorporated in the Newton-Euler iterative dynamic formulation. At the first link frame, an initial condition should be given depending on the speed of the vehicle. The provision is made in the computer program but in the presented simulations, the velocity of the vehicle is taken as zero.

As all the codes are written in C in order to develop a simulator of the forestry vehicle the codes can be directly interfaced with graphic routines in Open GL or Open Inventor on Silicon Graphics workstations. Also, the code can be optimized further by removing terms that are small in magnitude. Finally, all solid models are generated in Auto-CAD and stored in the DXF format. The DXF files can be converted to .iv files in order to generate animation in Open Inventor format.

References

[1] Abdalla, E., Pu, H. J., Mueller, M., Tantawy, A. A., Abdelatif, L., and Nour, E. H., "Novel Parallel Recursive Newton-Euler Algorithm for Modelling and Computation of Robot Dynamics," *Mathematics and Computers in Simulation*, Vol 37, No. 2, 1994, pp. 227-240.

[2] Angeles, J., "On the Numerical Solution of the Inverse Kinematic Problem," *The Int. Journal of Robotics Research*, Vol. 4, No. 2, 1985, pp. 21-37.

[3] Armstrong, B., Khatib, O., Burdick, J. W., "The Explicit Dynamic Model and Inertial Parameters of the PUMA 560 Arm," *Proc. of the 1986 IEEE Int. Conf. on Robotics and Automation*, 1986, pp. 510-518.

[4] Ashley, S., "Underground Mining from Above," *Mechanical Engineering*, Vol. 117, No. 5, 1995, pp. 78-81.

[5] Book, W. J., "Recursive Lagrangian Dynamics of Flexible Manipulator Arms," *The Int. Journal of Robotics Research*, Vol. 3, No. 3, 1984, pp. 87-101.

[6] Burdick, J. W., "An Algorithm for Generation of Efficient Manipulator Dynamic Equations," *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 1986, pp. 212-218.

[7] Carlton, R. E., and Bartholet, S. J., "The Evolution of the Application of Mobile Robotics to Nuclear Facility Operations and Maintenance," *Proc. 1987 IEEE Int. Conf. on Robotics and Automation*, 1987, pp. 720-726.

[8] Chace, M. A., "Analysis of the Time Depedence of Multi-Freedom Mechanical System in Relative Coordinates," *Trans. of the ASME, Journal of Engineering for Industry*, Feb., 1971, pp. 317-327.

[9] Char, B. W., Fee, G. J., Geddes, K. O., and Monagan, M. B., Tutorial Introduction to Maple, University of Waterloo, 1985.

[10] Courteau, J.,"Robotics in Forestry: An Industrial Response," *Proc. of the Symposium of Robotics in Forestry,* Sep 1990, pp. 40-44.

[11] Craig, J. J., Introduction to Robotics, Second Edition, Addison-Wesley Publishing Company, 1989.

[12] Cusack, M. M., "Industrial Robots in the UK - A Review," *The 5th International Symposium on Robotics in Construction,* Tokyo, Japan, 1988, pp. 51-56.

[13] Davidson, J. K., and Schweitzer G., "A Mechanics-based Computer Algorithm for Displaying the Margin of Stability in Quadrupped Walking Machine," *Proc. ASME Design Automation Conference*, Monteral, Canada, 1989.

[14] Denavit, J., and Hartenberg, R. S., "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices," Journal of Applied Mechanics, 1955, pp. 215-221.

[15] Dubowsky, S., Paul, I., and West, H., "An Analytical and Experimental Program to Develop Control Algorithms for Mobile Manipulators," *Proc. VII Symp. Theory and Practice of Robots and Manipulators,* Udine, Italy, September, 1988.

[16] Dubowsky, S., and Tanner, A. B., "A Study of the Dynamics and Control of Mobile Manipulators Subjected to Vehicle Disturbances," Proc. IV Int. Symp. of Robotics Research, Santa Cruz, CA, USA, August, 1987, pp. 111-117.

[17] Dubowsky, S., and Vance, E. E., "Planning Mobile Manipulator Motions Considering Vehicle Dynamic Stability Constraints," *Proc. IEEE Int Conf. on Robotics and Automation*, Scottsdale, AZ, USA, May, 1989, pp. 1271-1276.

[18] Edan, Y., Engel, B. A., and Miles, G. E., "Intelligent Control System Simulation of an Agricultural Robot." *Journal of Intelligent and Robotic Systems,* Vol. 8, No. 2, 1993, pp. 267-284.

[19] Eskandarian, A., Beduri, N. E. and Kramer, B. M., "Dynamic Modelling of Robotic Manipulators using an Artificial Neural Network," *Journal of Robotic Systems*, Vol 11, No. 1, 1994, pp. 41-56.

[20] Featherstone, R., "The Calculation of Robot Dynamics Using Articulated-Body Inertias," *The Int. Journal of Robotics Research*, Vol 2, No. 1, 1983, pp. 13-30.

[21] Fijany, A., and Bejczy, A. K., "Efficient Jacobian Inversion for the Control of Simple Robot Manipulators," *Proc. of the IEEE Int Conf. on Robotics and Automation*, 1988, pp. 999-1007.

[22] Frenton, R. G., and Xi, F. "On the efficiency of Computations for Robot Kinematics, Dynamics and Control Using the Algebra of Rotations.", *Proc. IEEE Int. Conf. on Robotics and Automation*, Vol. 1, 1993, pp. 968-973.

[23] Fukuda, S., The Manipulator to Turn Raw Land into Building Lots, *The 5th International Symposium on Robotics in Construction*, Tokyo, Japan, 1988, pp. 309-316.

[24] Fukuda, T., Fujisawa, Y., Kosuge, K., Arai, F., "Manipulator/Vehicle System for Man-Robot Cooperation," *Proc of the IEEE Int. Conf. on Robotics and Automation*, May 1992, pp. 74-79.

[25] Gill, P. S., "The Flexible Agricultural Robotics Manipulator," pp. 344-349.

[26] Giordano, M., "Dynamic Model of Robots with a Complex Kinematic Chain," *Proc. of the 16th Int. Symposium on Industrial Robots*, Sep. 1986, pp. 377-388.

[27] Goldsmith, S., "It's a Dirty Job, But Something's Gotta Do It," Science & Technology, 1990.

[28] Gupta, K. C., "Kinematic Analysis of Manipulators Using the Zero Reference Position Description," *The Int. Journal of Robotics Research*, Vol. 5, No. 2, 1986, pp. 5-13.

[29] *The Int. Journal of Robotics Research*, Special Issue on Legged Locomotion, Vol 9, No. 2, Apr. 1990.

[30] Hiller, M. and Schmitz, T., "Kinematics and Dynamics of the Combined Legged and Wheeled Vehicle 'ROBOTRAC'," *CSME Mechanical Engineering Forum*, 1990.

[31] Hollerbach, J. M., "A Recursive Lagrangian Formulation of Manipulator Dynamics and a Comparative Study of Dynamics Formulation Complexity," *IEEE Trans. Syst., Man and Cybern.*, Nov. 1980, pp. 730-736.

[32] Hwang, H., and Sistler, F. E., "The Implementation of a Robotic Manipulator on a Pepper Transplanting Machine," *Proc. CAD/CAM, Robotics Automation Int. Conf.*, 1985, pp. 553-556.

[33] Johnson, P.C., Farm Inventions in the Making of America. Des Moines, IA: Wallace-Homestead Book Company, 1976

[34] Joshi, J., and Desrochers, A., "Modeling and Control of a Mobile Robot Subject to Disturbances," *Proc. IEEE Int. Conf. on Robotics and Automation*, San Francisco, CA, USA Vol. 3, March, 1986, pp. 1508-1513.

[35] Kakuzen, M., Araya, H., Kimura, N., Automatic Control Systems for Construction Machinery, *The 5th International Symposium on Robotics in Construction*, Tokyo, Japan, 1988, pp. 755-764.

[36] Kane, T. R., and Levinson, D. A., "The use of Kane's Dynamical Equations in Robotics," *The Int. Journal of Robotics Research*, Vol. 2, No. 3, 1983, pp. 3-21.

[37] Kent, S., "Outpost Service and Construction Robot (OSCR)". *The 5th International Symposium on Robotics in Construction*, Tokyo, Japan, 1988, pp. 1454-1463

[38] Key, S. J., "Productivity Modelling and Forecasting for Automated Shearing Machinery," *Proc. Agri-Mation Conf. Exposition*, 1985, pp. 200-209

[39] Kochan, A., "Robots in the Nuclear Industry,"

[40] Koplik, J., and Leu, M. C., "Computer Generation of Robot Dynamics Equations and the Related Issues," *Journal of Robotic Systems* Vol. 3, No. 3, 1986, pp. 301-319.

[41] Koskela, L., "The Current Status of Industrialized Construction and Construction Robotics in Finland," *The 5th International Symposium on Robotics in Construction,* Tokyo, Japan, 1988, pp. 31-37.

[42] Kreutz-Delagado, K., Long, M., and Seraji, H., "Kinematic Analysis of 7 DOF Manipulator," *The Int. Journal of Robotics Research,* Vol. 11, No. 5, 1992, pp. 469-481.

[43] Kulz, G. W., "Future use of Robots in Agriculture," *Proc. 1st Int. Conf. Robotics and Intelligent Machines in Agriculture,* 1983, pp. 15-28.

[44] Kung, D., Parsons, J., and Hannaford, B., "Visualization of Manipulability With Mathematica," *IASTED Int. Conf. on Control and Robotics,* Vancouver, Canada, 1992, pp. 223-228.

[45] Langreth, R., "Smart Shovel," *Popular Science*, June 1992, pp.82-84.

[46] Lawrence, P.,Sauder, B., Wallersteiner, U., and Wilson, J., "Teleoperation of Forest Harvesting Machines," *Proc. of the Symposium of Robotics in Forestry,* Sep 1990, pp. 36-39.

[47] Leahy, M. B., Nugent, L. M., Saridis, G. N., and Valvanis, K. P., "Efficient PUMA Manipulator Jacobian Calculations and Inversion," *Journal of Robotic Systems,* Vol 4, No. 2, 1987, pp. 185-197.

[48] Leu, M. C., and Hemati, N., "Automated Symbolic Derivation of Dynamic Equations of Motion for Robotic Manipulators," *Journal of Dynamic Systems, Measurement, and Control,* Vol. 108 No.3, 1986, pp. 172-179.

[49] Li, Y. and Frank, A. A., "A moving Base Robot," *Proc. of the American Control Conference*, Seattle, WA, USA, June, 1986.

[50] Lin, J., and Lewis, F. L., "A Symbolic Formulation of Dynamic Equations For a Manipulator With Rigid and Flexible Links," *The Int. Journal of Robotics Research*, Vol. 13, No. 5, 1994, pp. 454-466.

[51] Luh, J. Y. S., and Zheng, Y. F., "Computation of Input Generalized Forces for Robots With Closed Kinematic Chain Mechanism," *IEEE Journal of Robotics and Automation*, Vol RA-1, No. 2, 1985.

[52] Luh, J. Y. S., Walker, M. W., and Paul, R. P., "On-Line Computational Scheme for Mechanical Manipulators," *Transactions of the ASME Journal of Dynamic Systems, Measurement, and Control*, 1980.

[53] Lynch, R., "Analysis of the Dynamics and Control of a Two Degree of Freedom Robotic Manipulator Mounted on a Moving Base," *M.S. Thesis*, Department of Mechanical Engineering, MIT Cambridge, MA, USA, October, 1985.

[54] Macsyma Reference Manual, MIT.

[55] McCarthy, J. M., "Dual Orthogonal Matrices in Manipulator Kinematics," The Int. Journal of Robotics Research, Vol. 5, No. 2, 1986, pp. 45-51.

[56] McMillan, S., Orin, D. E., Sadayappan, P., "Real-Time Robot Dynamic Simulation on a Vector/Parallel Supercomputer," *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Vol. 2, 1992, pp. 1836-1841.

[57] Megahed, S. and Renaud, M., "Dynamic Modelling of Robots Containing Closed Kinematic Chains," *Advanced Software in Robotics*, Liege, Belgium, 1983.

[58] Messuri, D. A. and Klein, C. A., "Automatic Body Regulation for Maintaining Stability of a Legged Vehicle During Rough-Terrain Locomotion," *IEEE Journal of Robotics and Automation*. Vol RA-1, No. 3, Sep 1985, pp. 132-141.

[59] Minami, M., Nakano, Y., Ueda, S., and Sudare, M., "Autonomous Mobile Robot Guidance on Arbitrarily Curved Courses Using Homogeneous Matrices," *Japan-U.S.A. Symposium on Flexible Automation,* 1990, pp. 519-526.

[60] Minami, M., Tomikawa, H., Fujiwara, N., Kanbara, K., "Inverse Dynamics Calculation Method of Power Wheeled Mobile Manipulators," *Proc. of Int. Conf. on Intelligent Robots and Systems,* 1993, pp. 781-786.

[61] Moselhi, O., "Feasibility and Potential Implementation of Robots in Canadian Building Construction," *The 5th International Symposium on Robotics in Construction,* Tokyo, Japan, 1988, pp. 229-238.

[62] Muir, P. F., Neuman, C. P., "Kinematic Modelling of Wheeled Mobile Robots," *Journal of Robotic Systems,* 1987, pp. 281-340.

[63] Obayashi, S., "The surrounding of Construction Industry and Problems of Automatization and Robotization in Construction of Japan," *The 5th International Symposium on Robotics in Construction,* Tokyo, Japan, 1988, pp. 39-46.

[64] Okawa, Y., System proposal of an Automated Vehicle Considering Its Use in Construction Sites, *The 5th International Symposium on Robotics in Construction,* Tokyo, Japan, 1988, pp. 249-258.

[65] Okazaki, N., Conception of Idealized New Construction Robot in Civil Engineering Work, *The 5th International Symposium on Robotics in Construction,* Tokyo, Japan, 1988, pp. 195-201

[66] Orin, D. E., and Schrader, W. W., "Efficient Computation of the Jacobian for Robot Manipulators," *The Int. Journal of Robotics Research,* Vol. 3, No. 4, 1984, pp. 66-75.

[67] Papadopoulos, E. G., Rey, D. A., "A New Measure of Tipover Stability Margin for Mobile Manipulators," *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Minneapolis, USA, 1996.

[68] Paul, R. P., and Zhang, H., "Computationally Efficient Kinematics for Manipulators with Spherical Wrists Based on the Homogeneous Transformation Representation," *The Int. Journal of Robotics Research,* Vol. 5, No. 2, 1986, pp. 32-44.

[69] Pieper, D. L., and Roth, B., "The Kinematics of Manipulators Under Computer Control," *Proc. of 2nd Int. Conf. on the Theory of Machines and Mechanisms,* 1969.

[70] Qiu, Y., and Xiao, S., "Dynamic Analysis and Computer Implementation of Manipulators Based on Generalized D'Alembert's Principle," *Journal of Northwestern Polytechnical University,* Vol. 6, No. 4, Oct 1988, pp. 457-468

[71] Rahm, H., G., "Robot in the Swedish Construction Industry," *The 5th International Symposium on Robotics in Construction*, Tokyo, Japan, 1988, pp. 47-50.

[72] Rodriguez, G., Jain, A., and Kreutz-Delgado, K., "A Spatial Operator Algebra for Manipulator Modelling and Control," *The Int. Journal of Robotics Research,* Vol. 10, No 4, 1991, pp. 371-381.

[73] Schilling, R., "Telerobots in the Nuclear Industry," *Industrial Robots,* Vol. 19, No. 2, 1992, pp. 11-14.

[74] Seward, D., "LUCIE - The Autonomous Robot Excavator", *Industrial Robots,* Vol. 19, No. 1, 1992, pp. 14-18.

[75] Sheridan, T. B., Telerobotics, Automation, and Human Supervisory Control, The MIT Press, 1992.

[76] Shoemaker, C. M., "Integrated Army Robotics Thrust," *Proc. of the Int. Topical Meeting on Remote Systems and Robotics in Hostile Environments,* Pasco, Washington, USA, March 29 - April 2, 1987, pp. 371-375.

[77] Silver, W. M., "On the Equivalence of Lagrangian and Newton-Euler Dynamics for Manipulators," *The Int. Journal of Robotics Research,* Vol. 1 No. 2, 1982, pp. 60-70.

[78] Skibniewski, M. J., Robotics in Civil Engineering, *Computational Mechanics Publications*, Van Nostrand Reinhold, 1988, pp.10-13.

[79] Smidt, D., Blume, C., Wadle, M., "A Multi-Link Multi-Purpose Advanced Manipulator with a Large Handling for Out-Door Applications," *The 5th International Symposium on Robotics in Construction*, Tokyo, Japan, June 6-8, 1988, pp. 625.

[80] Sugano, S., Huang, Q.and Kato, I., "Stability Criteria in Controlling Mobile Robotic Systems," *Proc. of Int. Conf. on Intelligent Robots and Systems,* July 1993, pp. 832-838.

[81] Trevelyan, J.P. and Elford, D., "Sheep Handling and Manipulation for Automated Shearing," *Proc of the Int. Symposium and Exposition on Robots*, Nov. 1988, pp. 397-407.

[82] Vagenas, N., "Possibilities for Mining Appplications of Industrial Robots", *Industrial Automation Conference,* Montreal, Canada, June 1992.

[83] Valenti, M., "ROSEE cleans up after the Cold War," *Mechanical Engineering,* Vol. 116, No. 7, 1994, pp. 70-71.

[84] Varrasi, J., "ASME News, Vol. 11, No. 9, Feb 1992.

[85] Vicker, J.J., "Dynamic Behaviour of Spatial Linkages," *Trans. of the ASME, Journal of Engineering for Industry,* Vol 91, 1969, pp. 251-258.

[86] Vukobratovic, M., Kircanski, N., and Li, S. G., "Approach to Parallel Processing of Dynamic Robot Models," *The Int. Journal of Robotics Research,* Vol. 7, No. 2, 1988, pp. 64-71.

[87] Wanner, M. C., Results of the development of a Manipulator with a Very Large Reach, *The 5th International Symposium on Robotics in Construction*, Tokyo, Japan, 1988, pp. 653-660.

[88] Wanner, M., "Current Status of the Development and Implementation of the Construction Robots in the Federal Republic of Germany," *The 5th International Symposium on Robotics in Construction*, Tokyo, Japan, 1988, pp. 27-30.

[89] West, H., Hootsmans, N., Dubowsky, S., Stelman, N., "Experimental Simulation of Manipulator Base Compliance," *Proc. of the First Int. Symposium on Experimental Robotics*, Montreal, Canada, 1990, pp. 304-321.

[90] Wiens, G. J., "Effects of Dynamic Coupling in Mobile Robotic Systems," *Proc. of SME Robotics Research*, Gaitherberg, MI, USA, 1989, pp. 3.43-3.57.

[91] Wolfarm, S., *Mathematica*, 2nd Edition, Addison-Wesley Publishing Company, 1993.

[92] Wong, J. Y., *Theory of Ground Vehicles*, John Wiley and Sons, Inc., 1993, pp. 47.

[93] Yuan, J. S-C., Stovman, J., MacDonald, R., and Norgate, G., "An Experimental Program on Advanced Robotics", 1994.

Appendix A

Expressions of torques in Stick, Boom and Swing Joints

The following expressions describe the equations of motion for three degrees of freedom system. The detailed procedure is described in Chapter 3. The left hand side of the Eqs. A.1 to A.3 denotes the torques required to cause the motion of the manipulator. The right hand side of the above mentioned equations have the acceleration, Coriolis and centrifugal, and gravity terms.

$$
\begin{aligned}
\tau_3 = & -\left[m_3 z_3 (x_3 s_{23} + y_3 c_{23}) + I^c_{3xz} s_{23} + I^c_{3yz} c_{23}\right]\ddot{q}_1 + \left[m_3 ({x_3}^2 + {y_3}^2 + l_2 x_3 c_3 - l_2 y_3 s_3) + I^c_{3zz}\right]\ddot{q}_2 + \\
& \left[m_3 ({x_3}^2 + {y_3}^2) + I^c_{3zz}\right]\ddot{q}_3 + \left[m_3 \{l_1 x_3 s_{23} + l_2 x_3 c_2 s_{23} + l_1 y_3 c_{23} + l_2 y_3 c_2 c_{23} + x_3 y_3 ({c_{23}}^2 - {s_{23}}^2) + \right. \\
& \left. c_{23} s_{23} ({x_3}^2 - {y_3}^2)\} + I^c_{3xy} ({c_{23}}^2 - {s_{23}}^2) + c_{23} s_{23} (I^c_{3yy} - I^c_{3xx})\right]{\dot{q}_1}^2 + \left[m_3 l_2 (x_3 s_3 + y_3 c_3)\right]{\dot{q}_2}^2 + \\
& m_3 \left[x_3 c_{23} - y_3 s_{23}\right] g
\end{aligned} \tag{A.1}
$$

$$
\begin{aligned}
\tau_2 = & \left[-m_3 z_3 (x_3 s_{23} + y_3 c_{23} + l_2 s_2) + m_2 z_2 (x_2 s_2 + y_2 c_2) + I^c_{3xz} s_{23} + I^c_{3yz} c_{23} + I^c_{2xz} s_2 + I^c_{2yz} c_2\right]\ddot{q}_1 + \\
& \left[m_3 \{{x_3}^2 + {y_3}^2 + {l_2}^2 + 2 l_2 (x_3 c_3 - y_3 s_3)\} + m_2 ({x_2}^2 + {y_2}^2) + I^c_{3zz} + I^c_{2zz}\right]\ddot{q}_2 + \\
& \left[m_3 \{{x_3}^2 + {y_3}^2 + l_2 (x_3 c_3 - y_3 s_3)\} + I^c_{3zz}\right]\ddot{q}_3 + \left[m_3 \{l_1 (l_2 s_2 + x_3 s_{23} + y_3 c_{23}) + \right. \\
& l_2 y_3 (2 c_2 c_{23} - c_3) + x_3 y_3 ({c_{23}}^2 - {s_{23}}^2) + {l_2}^2 s_2 c_2 + l_2 x_3 (2 c_2 s_{23} - s_3) + s_{23} c_{23} ({x_3}^2 - {y_3}^2)\} + \\
& m_2 \{l_1 (x_2 s_2 + y_2 c_2) + x_2 y_2 ({c_2}^2 - {s_2}^2) + s_2 c_2 ({x_2}^2 - {y_2}^2)\} + I^c_{3xy} ({c_{23}}^2 - {s_{23}}^2) + s_{23} c_{23} (I^c_{3yy} - I^c_{3xx}) + \\
& \left. I^c_{2xy} ({c_2}^2 - {s_2}^2) - s_2 c_2 (I^c_{2xx} - I^c_{2yy})\right]{\dot{q}_1}^2 - 2 m_3 l_2 \left[x_3 s_3 + y_3 c_3\right]\dot{q}_2 \dot{q}_3 - m_3 l_2 \left[x_3 s_3 + y_3 c_3\right]{\dot{q}_3}^2 + \\
& \left[m_3 (l_2 c_2 + x_3 c_{23} - y_3 s_{23}) + m_2 (x_2 c_2 - y_2 s_2)\right] g
\end{aligned} \tag{A.2}
$$

$$
\begin{aligned}
\tau_1 = {} & \left[m_3\{l_1^2 + l_2^2 c_2^2 + x_3^2 c_{23}^2 + y_3^2 s_{23}^2 + 2 l_1 l_2 c_2 + 2 l_1 (x_3 c_{23} - y_3 s_{23}) + \right. \\
& l_2 (x_3 c_3 - y_3 s_3) + z_3^2 + l_2 x_3 (2 c_2 c_{23} - c_3) - l_2 y_3 (2 c_2 s_{23} - s_3) - 2 x_3 y_3 s_{23} c_{23}\} + \\
& m_2\{l_1^2 + y_2^2 s_2^2 + x_2^2 c_2^2 + z_2^2 + 2 l_1 (x_2 c_2 - y_2 s_2) - 2 x_2 y_2 s_2 c_2\} + \\
& \left. m_1\{x_1^2 + y_1^2\} + I^c_{1zz} + I^c_{2xx} s_2^2 + I^c_{2yy} c_2^2 - 2 I^c_{2xy} s_2 c_2 + I^c_{3xx} s_{23}^2 + I^c_{3yy} c_{23}^2 - 2 I^c_{3xy} s_{23} c_{23}\right] \ddot{q}_1 - \\
& \left[m_3 z_3 (x_3 s_{23} + y_3 c_{23} + l_2 s_2) + m_2 z_2 (x_2 s_2 + y_2 c_2) + I^c_{3xz} s_{23} + I^c_{3yz} c_{23} + I^c_{2xz} s_2 + I^c_{2yz} c_2\right] \ddot{q}_2 - \\
& \left[m_3 z_3 (x_3 s_{23} + y_3 c_{23}) + I^c_{3xz} s_{23} + I^c_{3yz} c_{23}\right] \ddot{q}_3 - \left[m_3\{2 l_1 (l_2 s_2 + x_3 s_{23} + y_3 c_{23}) + \right. \\
& 2 l_2 y_3 (2 c_2 c_{23} - c_3) + 2 x_3 y_3 (c_{23}^2 - s_{23}^2) + 2 l_2^2 s_2 c_2 + 2 l_2 x_3 (2 c_2 s_{23} - s_3) + \\
& 2 s_{23} c_{23} (x_3^2 - y_3^2)\} - 2 m_2\{l_1 (x_2 s_2 + y_2 c_2) + x_2 y_2 (c_2^2 - s_2^2) + s_2 c_2 (x_2^2 - y_2^2)\} - \\
& \left. 2 I^c_{2xy} (c_2^2 - s_2^2) - 2 s_2 c_2 (I^c_{2yy} - I^c_{2xx}) - 2 I^c_{3xy} (c_{23}^2 - s_{23}^2) - 2 s_{23} c_{23} (I^c_{3yy} - I^c_{3xx})\right] \dot{q}_1 \dot{q}_2 + \\
& \left[m_3\{2 l_1 (x_3 s_{23} + y_3 c_{23}) - l_2 (x_3 s_3 + y_3 c_3) - l_2 y_3 (2 c_2 c_{23} - c_3) - 2 x_3 y_3 (c_{23}^2 - s_{23}^2) - \right. \\
& l_2 y_3 (2 c_2 c_{23} - c_3) - 2 x_3 y_3 (c_{23}^2 - s_{23}^2) - l_2 x_3 (2 c_2 s_{23} - s_3) - 2 s_{23} c_{23} (x_3^2 - y_3^2)\} - \\
& \left. -2 I^c_{3xy} (c_{23}^2 - s_{23}^2) - 2 s_{23} c_{23} (I^c_{3yy} - I^c_{3xx})\right] \dot{q}_1 \dot{q}_3 + \\
& \left[2 m_3 z_3 (-x_3 c_{23} + y_3 s_{23}) - 2 I^c_{3xz} c_{23} + 2 I^c_{3yz} s_{23}\right] \dot{q}_2 \dot{q}_3 + \\
& \left[-m_3 z_3 (l_2 c_2 + x_3 c_{23} - y_3 s_{23}) - m_2 z_2 (x_2 c_2 - y_2 s_2) - I^c_{3xz} c_{23} + I^c_{3yz} s_{23} - I^c_{2xz} c_2 + I^c_{2yz} s_2\right] \dot{q}_2^2 + \\
& \left[m_3 z_3 (-x_3 c_{23} + y_3 s_{23}) - I^c_{3xz} c_{23} + I^c_{3yz} s_{23}\right] \dot{q}_3^2
\end{aligned}
\tag{A.3}
$$

Appendix B

This program generates the dynamic model in various files in the home directory. Mass matrix, Coriolis and centrifugal terms, and gravity terms are stored in separate files with their conventional names. Firstly, all the successive rotation matrices are defined, and inverted (*Inverse[]* command), then stored in separate variables. Then the link parameters (length and location of center of mass) are written in vector form. Velocity and acceleration propagation are written next and finally the backward propagation is done and actuator torques are equated.

After each matrix multiplication the resulting expressions are simplified with trigonometric flags on, and stored in a variable. Once usage of a variable is finished all the attributes of that variables are cleared (*ClearAll[]* command) and the name of the variable is removed (*Remove[]* command), thus freeing memory. The replacement of the variables are done by (/. operator), the recursive replacements are done by (//. operator). For big symbolic expressions the coefficients are collected (*Coefficient[expression, variable]* command) and simplified (*Simplify[]* command) step by step rather than simplifying the whole expression. The whole expression is written in a readable form by collecting all the acceleration, Coriolis and gravity terms, (*Collect[]* command).

Source Code in Mathematica to Generate the Dynamics for 8 DOF System

```
(* mass in kg length in meter *)
(*<<Ssonerule.m. It is a package to
substitute
Sin ^2 (theta) + Cos ^2 (theta) = 1
instantaneously *)
Off[General::spell];
Off[General::spell1];

r11 = 1;r12 =
0;r13=0;r21=0;r22=1;r23=0;r31=0;r32=0;r3
3=1;

(*Rotation Matrices*)
qz[t] = 0;(*yaw neglected*)
rotationz = {{Cos[qz[t]], -Sin[qz[t]], 0},
{Sin[qz[t]], Cos[qz[t]], 0},
{0, 0, 1}};
rotationy = {{Cos[qy[t]],0,Sin[qy[t]]},
{0,1,0},
{-Sin[qy[t]],0,Cos[qy[t]]}};
rotationx = {{1,0,0},
{0,Cos[qx[t]],-Sin[qx[t]]},
{0,Sin[qx[t]],Cos[qx[t]]}};
wr0p = rotationz.rotationy.rotationx;
(*wr0p=wr0prime*)
(*from AutoCad*)
zeropr0 =
{{r11,r12,r13},{r21,r22,r23},{r31,r32,r33}};
wr0 = wr0p.zeropr0;
(*Print["wr0=",wr0];*)
zerorw = SSonerule[Inverse[wr0]];
zeror1 = {{Cos[q1[t]], -Sin[q1[t]], 0},
{Sin[q1[t]], Cos[q1[t]], 0},
{0, 0, 1}};
oner0 = SSonerule[Inverse[zeror1]];
oner2 = {{Cos[q2[t]], -Sin[q2[t]], 0},
{0, 0, -1},
{Sin[q2[t]], Cos[q2[t]], 0}};
twor1 = SSonerule[Inverse[oner2]];
twor3 = {{Cos[q3[t]], -Sin[q3[t]], 0},
{Sin[q3[t]], Cos[q3[t]], 0},
{0, 0,1}};
threer2 = SSonerule[Inverse[twor3]];
threer4 = {{Cos[q4[t]], -Sin[q4[t]], 0},
{Sin[q4[t]], Cos[q4[t]], 0},
{0, 0,1}};
fourr3 = SSonerule[Inverse[threer4]];
fourr5 = {{Cos[q5[t]], -Sin[q5[t]], 0},
{0,0,1},
{-Sin[q5[t]],-Cos[q5[t]],0}};
fiver4 = SSonerule[Inverse[fourr5]];
zeror5 = zeror1.oner2.twor3.threer4.fourr5;

(*Link Parameters*)
wp0 = {{xw},{yw},{zw}};
(*Dynamics won't be dependent on this
parameters*)
zerop1 = {{x0},{y0},{z0}};
zerop1m = {{0,-z0,y0},{z0,0,-x0},{-y0,x0,0}};
zeropc0 = {{0},{0},{0}};
zeropc0m = {{0,0,0},{0,0,0},{0,0,0}};

onep2 = {{l1},{0},{0}};
onep2m = {{0,0,0},{0,0,-l1},{0,l1,0}};
onepc1 = {{x1},{y1},{z1}};
onepc1m = {{0,-z1,y1},{z1,0,-x1},{-y1,x1,0}};

twop3 = {{l2},{0},{0}};
twop3m = {{0,0,0},{0,0,-l2},{0,l2,0}};
twopc2 = {{x2},{y2},{z2}};
twopc2m = {{0,-z2,y2},{z2,0,-x2},{-y2,x2,0}};

threep4 = {{l3},{0},{0}};
threep4m = {{0,0,0},{0,0,-l3},{0,l3,0}};
threepc3 = {{x3},{y3},{z3}};
threepc3m = {{0,-z3,y3},{z3,0,-x3},{-
y3,x3,0}};

fourp5 = {{l4},{0},{0}};
fourp5m = {{0,0,0},{0,0,-l4},{0,l4,0}};
fourpc4 = {{x4},{y4},{z4}};
fourpc4m = {{0,-z4,y4},{z4,0,-x4},{-y4,x4,0}};

fivepc5 = {{x5},{y5},{z5}};
fivepc5m = {{0,-z5,y5},{z5,0,-x5},{-y5,x5,0}};

(*Velocity and Acceleration Propagation*)
wvw = {{0},{0},{0}};
(*it does not hamper manipulator dynamics*)
wvDw = {{0},{0},{g}};
www = {{0},{0},{0}};
wwDw = D[www,t];
wwwm = {{0,-
www[[3,1]],www[[2,1]]},{www[[3,1]],0,
-www[[1,1]]},{-www[[2,1]],www[[1,1]],0}};
wwDwm = D[wwwm,t];
```

```
zerow0 = zerorw.www +
{{D[qx[t],t]},{D[qy[t],t]},{0}};
zerow0m = {{0,-
zerow0[[3,1]],zerow0[[2,1]]},{zerow0[[3,1]],
0,
-zerow0[[1,1]]},{-
zerow0[[2,1]],zerow0[[1,1]],0}};
zerowD0 = D[zerow0,t];
zerowD0m = D[zerow0m,t];
zerov0 =
zerorw.(wvw+wwwm.wp0)+{{0},{0},{zd}};
(*only motion in z direction*)
zeroww = zerorw.www;
zerowwm = {{0,-
zeroww[[3,1]],zeroww[[2,1]]},{zeroww[[3,1]],
0,
-zeroww[[1,1]]},{-
zeroww[[2,1]],zeroww[[1,1]],0}};
zerovD0 =
zerorw.(wvDw+wwDwm.wp0+wwwm.wwwm.
wp0)+2*zerowwm.{{0},{0},{zd}}+{{0},{0},{zdd}
};
ClearAll[wvDw,zerorw];
Remove[wvDw,zerorw];
zerovDc0 =
zerovD0+zerowD0m.zeropc0+zerow0m.zer
ow0m.zeropc0;

onew1 = oner0.zerow0 +
{{0},{0},{D[q1[t],t]}};
onew1m = {{0,-
onew1[[3,1]],onew1[[2,1]]},{onew1[[3,1]],0,
-onew1[[1,1]]},{-
onew1[[2,1]],onew1[[1,1]],0}};
onewD1 = D[onew1,t];
onewD1 = Simplify[onewD1];
onewD1m = D[onew1m,t];
onev1 = oner0.(zerov0+zerow0m.zerop1);
onev1 = Simplify[onev1];
onevD1 = oner0.zerovD0;
onevD1 = Simplify[onevD1];
ClearAll[zerovD0,oner0];
Remove[zerovD0,oner0];
onevDc1 =
onevD1+onewD1m.onepc1+onew1m.one
w1m.onepc1;
onevDc1 = Simplify[onevDc1];

twow2 = twor1.onew1+{{0},{0},{D[q2[t],t]}};
twow2 = Simplify[twow2];
twow2m = {{0,-
twow2[[3,1]],twow2[[2,1]]},{twow2[[3,1]],0,
-twow2[[1,1]]},{-
twow2[[2,1]],twow2[[1,1]],0}};
twowD2 = D[twow2,t];
twowD2 = Simplify[twowD2];
twowD2m = {{0,-
twowD2[[3,1]],twowD2[[2,1]]},{twowD2[[3,1]
],0,
-twowD2[[1,1]]},{-
twowD2[[2,1]],twowD2[[1,1]],0}};
twov2 = twor1.(onev1+onew1m.onep2);
twov2 = Simplify[twov2];
twovD2 =
twor1.(onevD1+onewD1m.onep2+onew1m
.onew1m.onep2);
twovD2 = Simplify[twovD2];
ClearAll[twor1,onewD1m,onevD1];
Remove[twor1,onewD1m,onevD1];
twovDc2 =
twovD2+twowD2m.twopc2+twow2m.twow2
m.twopc2;
twovDc2 = Simplify[twovDc2];
ClearAll[twopc2];
Remove[twopc2];

threew3 =
threer2.twow2+{{0},{0},{D[q3[t],t]}};
threew3 = Simplify[threew3];
threew3m = {{0,-
threew3[[3,1]],threew3[[2,1]]},{threew3[[3,1
]],0,
-threew3[[1,1]]},{-
threew3[[2,1]],threew3[[1,1]],0}};
threewD3 = D[threew3,t];
threewD3 = Simplify[threewD3];
threewD3m = {{0,-
threewD3[[3,1]],threewD3[[2,1]]},{threewD3
[[3,1]],0,
-threewD3[[1,1]]},{-
threewD3[[2,1]],threewD3[[1,1]],0}};
threev3 = threer2.(twov2+twow2m.twop3);
threev3 = Simplify[threev3];
threevD3 =
threer2.(twovD2+twowD2m.twop3+twow2m
.twow2m.twop3);
threevD3 = Simplify[threevD3];
ClearAll[twop3,threer2,twowD2m,twovD2];
Remove[twop3,threer2,twowD2m,twovD2];
threevDc3 =
threevD3+threewD3m.threepc3+threew3m.
threew3m.threepc3;
threevDc3 = Simplify[threevDc3];

fourw4 =
fourr3.threew3+{{0},{0},{D[q4[t],t]}};
```

```
fourw4 = Simplify[fourw4];
fourw4m = {{0,-
fourw4[[3,1]],fourw4[[2,1]]},{fourw4[[3,1]],0,
-fourw4[[1,1]]},{-
fourw4[[2,1]],fourw4[[1,1]],0}};
fourwD4 = D[fourw4,t];
fourwD4 = Simplify[fourwD4];
fourwD4m = {{0,-
fourwD4[[3,1]],fourwD4[[2,1]]},{fourwD4[[3,
1]],0,
-fourwD4[[1,1]]},{-
fourwD4[[2,1]],fourwD4[[1,1]],0}};
fourv4 =
fourr3.(threev3+threew3m.threep4);
fourv4 = Simplify[fourv4];
fourvD4 =
fourr3.(threevD3+threewD3m.threep4+thre
ew3m.threew3m.threep4);
fourvD4 = Simplify[fourvD4];
fourvDc4 =
fourvD4+fourwD4m.fourpc4+fourw4m.four
w4m.fourpc4;
fourvDc4 = Simplify[fourvDc4];

fivew5 = fiver4.fourw4+{{0},{0},{D[q5[t],t]}};
fivew5 = Simplify[fivew5];
fivew5m = {{0,-
fivew5[[3,1]],fivew5[[2,1]]},{fivew5[[3,1]],0,
-fivew5[[1,1]]},{-
fivew5[[2,1]],fivew5[[1,1]],0}};
fivewD5 = D[fivew5,t];
fivewD5 = Simplify[fivewD5];
fivewD5m = {{0,-
fivewD5[[3,1]],fivewD5[[2,1]]},{fivewD5[[3,1]
],0,
-fivewD5[[1,1]]},{-
fivewD5[[2,1]],fivewD5[[1,1]],0}};
fivev5 = fiver4.(fourv4+fourw4m.fourp5);
fivev5 = Simplify[fivev5];
fivevD5 =
fiver4.(fourvD4+fourwD4m.fourp5+fourw4m
.fourw4m.fourp5);
fivevD5 = Simplify[fivevD5];
fivevDc5 =
fivevD5+fivewD5m.fivepc5+fivew5m.fivew5
m.fivepc5;
fivevDc5 = Simplify[fivevDc5];

I5 = {{I5xx,-I5xy,-I5xz},{-I5xy,I5yy,-I5yz},{-
I5xz,-I5yz,I5zz}};
I4 = {{I4xx,-I4xy,-I4xz},{-I4xy,I4yy,-I4yz},{-
I4xz,-I4yz,I4zz}};
I3 = {{I3xx,-I3xy,-I3xz},{-I3xy,I3yy,-I3yz},{-
I3xz,-I3yz,I3zz}};
I2 = {{I2xx,-I2xy,-I2xz},{-I2xy,I2yy,-I2yz},{-
I2xz,-I2yz,I2zz}};
I1 = {{I1xx,-I1xy,-I1xz},{-I1xy,I1yy,-I1yz},{-
I1xz,-I1yz,I1zz}};
I0 = {{I0xx,-I0xy,-I0xz},{-I0xy,I0yy,-I0yz},{-
I0xz,-I0yz,I0zz}};

fiveF5 = m5*fivevDc5;
ClearAll[fivevDc5];
Remove[fivevDc5];
fiveN5 = I5.fivewD5+fivew5m.I5.fivew5;
fiveN5 = Simplify[fiveN5];
ClearAll[I5,fivew5,fivew5m,fivewD5];
Remove[I5,fivew5,fivew5m,fivewD5];

fourF4 = m4*fourvDc4;
ClearAll[fourvDc4];
Remove[fourvDc4];
fourN4 = I4.fourwD4+fourw4m.I4.fourw4;
fourN4 = Simplify[fourN4];
ClearAll[I4,fourw4,fourw4m,fourwD4];
Remove[I4,fourw4,fourw4m,fourwD4];

threeF3 = m3*threevDc3;
ClearAll[threevDc3];
Remove[threevDc3];
threeN3 =
I3.threewD3+threew3m.I3.threew3;
threeN3 = Simplify[threeN3];
ClearAll[I3,threew3,threew3m,threewD3];
Remove[I3,threew3,threew3m,threewD3];

twoF2 = m2*twovDc2;
ClearAll[twovDc2];
Remove[twovDc2];
twoN2 = I2.twowD2+twow2m.I2.twow2;
twoN2 = Simplify[twoN2];
ClearAll[I2,twow2,twow2m,twowD2];
Remove[I2,twow2,twow2m,twowD2];

oneF1 = m1*onevDc1;
ClearAll[onevDc1];
Remove[onevDc1];
oneN1 = I1.onewD1+onew1m.I1.onew1;
oneN1 = Simplify[oneN1];
ClearAll[I1,onew1,onew1m,onewD1];
Remove[I1,onew1,onew1m,onewD1];

zeroF0 = m0*zerovDc0;
ClearAll[zerovDc0];
Remove[zerovDc0];
zeroN0 = I0.zerowD0+zerow0m.I0.zerow0;
zeroN0 = Simplify[zeroN0];
ClearAll[I0,zerow0,zerow0m,zerowD0];
```

```
Remove[I0,zerow0,zerow0m,zerowD0];

precision = .00001;(*below this value will be
treated as zero*)
decimal = 4;(*in output after decimal 4 digits
will be printed*)

(*Backward Iteration for Dynamics*)
(****************** for torque5 ***************)
fivef5 = fiveF5;
fiven5 = fiveN5+fivepc5m.fiveF5;
fiven5 = Simplify[fiven5];
ClearAll[fivepc5m,fiveF5,fiveN5];
Remove[fivepc5m,fiveF5,fiveN5];

torque5 = fiven5[[3,1]];
torque5 = SSonerule[torque5];
torque5 = torque5 /.{q1[t]->q1,q2[t]-
>q2,q3[t]->q3,q4[t]->q4,q5[t]->q5,qx[t]-
>qx,qy[t]->qy,qz[t]->qz,
q1'[t]->q1d,q2'[t]->q2d,q3'[t]->q3d,q4'[t]-
>q4d,q5'[t]->q5d,qx'[t]->qxd,qy'[t]-
>qyd,qz'[t]->qzd,
q1"[t]->q1dd,q2"[t]->q2dd,q3"[t]-
>q3dd,q4"[t]->q4dd,q5"[t]->q5dd,qx"[t]-
>qxdd,qy"[t]->qydd,qz"[t]->qzdd};

torque5 = torque5/.{q4d q5d->q4dq5d,q3d
q5d->q3dq5d,q3d q4d->q3dq4d,q2d q5d-
>q2dq5d,
q2d q4d->q2dq4d,q2d q3d->q2dq3d,q1d
q5d->q1dq5d,
q1d q4d->q1dq4d,q1d q3d->q1dq3d,q1d
q2d->q1dq2d,q5d^2->q5dsq,q4d^2-
>q4dsq,q3d^2->q3dsq,
q2d^2->q2dsq,q1d^2->q1dsq,q5d qxd-
>q5dqxd,q5d qyd->q5dqyd,q4d qxd-
>q4dqxd,q4d qyd->q4dqyd,
q3d qxd->q3dqxd,q3d qyd->q3dqyd,q2d
qxd->q2dqxd,q2d qyd->q2dqyd,
q1d qxd->q1dqxd,q1d qyd->q1dqyd,
qxd^2->qxdsq,qyd^2->qydsq,qxd qyd-
>qxdqyd};

(*TAKE ALL mass matrix element*)

torque5 = Expand[torque5];
torque5 = torque5//.{q4d q5d->q4dq5d,q3d
q5d->q3dq5d,q3d q4d->q3dq4d,q2d q5d-
>q2dq5d,
q2d q4d->q2dq4d,q2d q3d->q2dq3d,q1d
q5d->q1dq5d,
q1d q4d->q1dq4d,q1d q3d->q1dq3d,q1d
q2d->q1dq2d,q5d^2->q5dsq,q4d^2-
>q4dsq,q3d^2->q3dsq,
q2d^2->q2dsq,q1d^2->q1dsq,q5d qxd-
>q5dqxd,q5d qyd->q5dqyd,q4d qxd-
>q4dqxd,q4d qyd->q4dqyd,
q3d qxd->q3dqxd,q3d qyd->q3dqyd,q2d
qxd->q2dqxd,q2d qyd->q2dqyd,
q1d qxd->q1dqxd,q1d qyd->q1dqyd,
qxd^2->qxdsq,qyd^2->qydsq,qxd qyd-
>qxdqyd};

torque5 = Chop[torque5,precision];
torque5 = SetAccuracy[torque5,decimal];
torque5 = torque5//.{2.0000->2,1.0000-
>1};

file[1]="q4dq5d5";file[2]="q3dq5d5";file[3]
="q3dq4d5";file[4]="q2dq5d5";file[5]="q2d
q4d5";file[6]="q2dq3d5";file[7]="q1dq5d5"
;
file[8]="q1dq4d5";file[9]="q1dq3d5";file[10
]="q1dq2d5";file[11]="q5dsq5";file[12]="q4
dsq5";file[13]="q3dsq5";
file[14]="q2dsq5";file[15]="q1dsq5";file[16]
="q5dqxd5";file[17]="q5dqyd5";file[18]="q
4dqxd5";file[19]="q4dqyd5";
file[20]="q3dqxd5";file[21]="q3dqyd5";file[
22]="q2dqxd5";file[23]="q2dqyd5";file[24]
="q1dqxd5";file[25]="q1dqyd5";
file[26]="qxdsq5";file[27]="qydsq5";file[28]
="qxdqyd5";file[29]="m88";file[30]="m78";fi
le[31]="m68";file[32]="m58";file[33]="m48"
;
file[34]="m38";file[35]="m28";file[36]="m18
";file[37]="G5";

variable[1]=q4dq5d;variable[2]=q3dq5d;vari
able[3]=q3dq4d;
variable[4]=q2dq5d;variable[5]=q2dq4d;vari
able[6]=q2dq3d;
variable[7]=q1dq5d;variable[8]=q1dq4d;vari
able[9]=q1dq3d;
variable[10]=q1dq2d;variable[11]=q5dsq;va
riable[12]=q4dsq;
variable[13]=q3dsq;variable[14]=q2dsq;vari
able[15]=q1dsq;
variable[16]=q5dqxd;variable[17]=q5dqyd;v
ariable[18]=q4dqxd;
variable[19]=q4dqyd;variable[20]=q3dqxd;v
ariable[21]=q3dqyd;
variable[22]=q2dqxd;variable[23]=q2dqyd;v
ariable[24]=q1dqxd;
```

```
variable[25]=q1dqyd;variable[26]=qxdsq;var
iable[27]=qydsq;
variable[28]=qxdqyd;variable[29]=q5dd;vari
able[30]=q4dd;variable[31]=q3dd;variable[3
2]=q2dd;variable[33]=q1dd;variable[34]=qy
dd;variable[35]=qxdd;variable[36]=zdd;

(*AUTOMATION FOR ALL TERMS*)
For[counter=1,counter<=36,counter++,var
name=variable[counter];
expr = Coefficient[torque5,varname];
filename = file[counter];
expr = Simplify[expr];
expr = expr//.{Sin[qx]->s6,Sin[qy]-
>s7,Sin[q1]->s1,Sin[q2]->s2,Sin[q3]-
>s3,Sin[q4]->s4,Sin[q5]->s5,
Cos[qx]->c6,Cos[qy]->c7,Cos[q1]-
>c1,Cos[q2]->c2,Cos[q3]->c3,Cos[q4]-
>c4,Cos[q5]->c5};
expr = expr//.{2.0000->2,1.0000->1,
2.0000-> 2, 1.0000-> 1};
expr =
Collect[expr,{s1,c1,s2,c2,s3,c3,s4,c4,s5,c5
,s6,c6,s7,c7}];
expr = varname*expr;
torque5 = torque5/.{varname->0};
stmp = OpenWrite[filename];
WriteString[stmp,filename];
PutAppend["=",filename];
PutAppend[expr,filename];
PutAppend[";",filename];Close[stmp];];
torque5 =
Collect[torque5,{s1,c1,s2,c2,s3,c3,s4,c4,s
5,c5,s6,c6,s7,c7}];
filename = file[37];
stmp = OpenWrite[filename];
WriteString[stmp,filename];
PutAppend["=",filename];
PutAppend[torque5,filename];
PutAppend[";",filename];Close[stmp];

time=Date[];
Print["Derivation of torque5 finished
at:",time];

(****************** for torque4 ***************)
fourf4 = fourr5.fivef5+fourF4;
fourf4 = Simplify[fourf4];
fourn4 =
fourN4+fourr5.fiven5+fourpc4m.fourF4+fou
rp5m.fourr5.fivef5;
fourn4 = Simplify[fourn4];
ClearAll[fourr5,fourp5m,fourpc4m,fourF4,fo
urN4,fivef5,fiven5];
Remove[fourr5,fourp5m,fourpc4m,fourF4,f
ourN4,fivef5,fiven5];

torque4 = fourn4[[3,1]];
torque4 = SSonerule[torque4];
torque4 = torque4 /.{q1[t]->q1,q2[t]-
>q2,q3[t]->q3,q4[t]->q4,q5[t]->q5,qx[t]-
>qx,qy[t]->qy,qz[t]->qz,
q1'[t]->q1d,q2'[t]->q2d,q3'[t]->q3d,q4'[t]-
>q4d,q5'[t]->q5d,qx'[t]->qxd,qy'[t]-
>qyd,qz'[t]->qzd,
q1"[t]->q1dd,q2"[t]->q2dd,q3"[t]-
>q3dd,q4"[t]->q4dd,q5"[t]->q5dd,qx"[t]-
>qxdd,qy"[t]->qydd,qz"[t]->qzdd};

torque4 = torque4/.{q4d q5d->q4dq5d,q3d
q5d->q3dq5d,q3d q4d->q3dq4d,q2d q5d-
>q2dq5d,
q2d q4d->q2dq4d,q2d q3d->q2dq3d,q1d
q5d->q1dq5d,
q1d q4d->q1dq4d,q1d q3d->q1dq3d,q1d
q2d->q1dq2d,q5d^2->q5dsq,q4d^2-
>q4dsq,q3d^2->q3dsq,
q2d^2->q2dsq,q1d^2->q1dsq,q5d qxd-
>q5dqxd,q5d qyd->q5dqyd,q4d qxd-
>q4dqxd,q4d qyd->q4dqyd,
q3d qxd->q3dqxd,q3d qyd->q3dqyd,q2d
qxd->q2dqxd,q2d qyd->q2dqyd,
q1d qxd->q1dqxd,q1d qyd->q1dqyd,
qxd^2->qxdsq,qyd^2->qydsq,qxd qyd-
>qxdqyd};

(*get rid of unnecessary mass matrix
element*)
torque4 = torque4/.{q5dd->0};

torque4 = Expand[torque4];
torque4 = torque4//.{q4d q5d->q4dq5d,q3d
q5d->q3dq5d,q3d q4d->q3dq4d,q2d q5d-
>q2dq5d,
q2d q4d->q2dq4d,q2d q3d->q2dq3d,q1d
q5d->q1dq5d,
q1d q4d->q1dq4d,q1d q3d->q1dq3d,q1d
q2d->q1dq2d,q5d^2->q5dsq,q4d^2-
>q4dsq,q3d^2->q3dsq,
q2d^2->q2dsq,q1d^2->q1dsq,q5d qxd-
>q5dqxd,q5d qyd->q5dqyd,q4d qxd-
>q4dqxd,q4d qyd->q4dqyd,
q3d qxd->q3dqxd,q3d qyd->q3dqyd,q2d
qxd->q2dqxd,q2d qyd->q2dqyd,
q1d qxd->q1dqxd,q1d qyd->q1dqyd,
qxd^2->qxdsq,qyd^2->qydsq,qxd qyd-
>qxdqyd};
```

```
torque4 = Chop[torque4,precision];
torque4 = SetAccuracy[torque4,decimal];
torque4 = torque4//.{2.0000->2,1.0000-
>1};

file[1]="q4dq5d4";file[2]="q3dq5d4";file[3]
="q3dq4d4";file[4]="q2dq5d4";file[5]="q2d
q4d4";file[6]="q2dq3d4";file[7]="q1dq5d4"
;
file[8]="q1dq4d4";file[9]="q1dq3d4";file[10
]="q1dq2d4";file[11]="q5dsq4";file[12]="q4
dsq4";file[13]="q3dsq4";
file[14]="q2dsq4";file[15]="q1dsq4";file[16]
="q5dqxd4";file[17]="q5dqyd4";file[18]="q
4dqxd4";file[19]="q4dqyd4";
file[20]="q3dqxd4";file[21]="q3dqyd4";file[
22]="q2dqxd4";file[23]="q2dqyd4";file[24]
="q1dqxd4";file[25]="q1dqyd4";
file[26]="qxdsq4";file[27]="qydsq4";file[28]
="qxdqyd4";file[29]="m77";file[30]="m67";fi
le[31]="m57";file[32]="m47";file[33]="m37"
;
file[34]="m27";file[35]="m17";file[36]="G4";

variable[1]=q4dq5d;variable[2]=q3dq5d;vari
able[3]=q3dq4d;
variable[4]=q2dq5d;variable[5]=q2dq4d;vari
able[6]=q2dq3d;
variable[7]=q1dq5d;variable[8]=q1dq4d;vari
able[9]=q1dq3d;
variable[10]=q1dq2d;variable[11]=q5dsq;va
riable[12]=q4dsq;
variable[13]=q3dsq;variable[14]=q2dsq;vari
able[15]=q1dsq;
variable[16]=q5dqxd;variable[17]=q5dqyd;v
ariable[18]=q4dqxd;
variable[19]=q4dqyd;variable[20]=q3dqxd;v
ariable[21]=q3dqyd;
variable[22]=q2dqxd;variable[23]=q2dqyd;v
ariable[24]=q1dqxd;
variable[25]=q1dqyd;variable[26]=qxdsq;var
iable[27]=qydsq;
variable[28]=qxdqyd;variable[29]=q4dd;vari
able[30]=q3dd;variable[31]=q2dd;variable[3
2]=q1dd;variable[33]=qydd;variable[34]=qx
dd;variable[35]=zdd;

(*AUTOMATION FOR ALL TERMS*)
For[counter=1,counter<=35,counter++,var
name=variable[counter];
expr = Coefficient[torque4,varname];
filename = file[counter];
expr = Simplify[expr];
expr = expr//.{Sin[qx]->s6,Sin[qy]-
>s7,Sin[q1]->s1,Sin[q2]->s2,Sin[q3]-
>s3,Sin[q4]->s4,Sin[q5]->s5,
Cos[qx]->c6,Cos[qy]->c7,Cos[q1]-
>c1,Cos[q2]->c2,Cos[q3]->c3,Cos[q4]-
>c4,Cos[q5]->c5};
expr = expr//.{2.0000->2,1.0000->1,
2.0000-> 2, 1.0000-> 1};
expr =
Collect[expr,{s1,c1,s2,c2,s3,c3,s4,c4,s5,c5
,s6,c6,s7,c7}];
expr = varname*expr;
torque4 = torque4/.{varname->0};

stmp = OpenWrite[filename];
WriteString[stmp,filename];
PutAppend["=",filename];
PutAppend[expr,filename];
PutAppend[";",filename];Close[stmp];];
torque4 =
Collect[torque4,{s1,c1,s2,c2,s3,c3,s4,c4,s
5,c5,s6,c6,s7,c7}];
filename = file[36];
stmp = OpenWrite[filename];
WriteString[stmp,filename];
PutAppend["=",filename];
PutAppend[torque4,filename];
PutAppend[";",filename];Close[stmp];

time=Date[];
Print["Derivation of torque4 finished
at:",time];
(****************** for torque3 ***************)
threef3 = threer4.fourf4+threeF3;
threef3 = Simplify[threef3];
threen3 =
threeN3+threer4.fourn4+threepc3m.threeF
3+threep4m.threer4.fourf4;
threen3 = Simplify[threen3];
ClearAll[threer4,threep4m,threepc3m,three
F3,threeN3,fourf4,fourn4];
Remove[threer4,threep4m,threepc3m,thre
eF3,threeN3,fourf4,fourn4];

torque3 = threen3[[3,1]];
torque3 = SSonerule[torque3];
torque3 = torque3 /.{q1[t]->q1,q2[t]-
>q2,q3[t]->q3,q4[t]->q4,q5[t]->q5,qx[t]-
>qx,qy[t]->qy,qz[t]->qz,
q1'[t]->q1d,q2'[t]->q2d,q3'[t]->q3d,q4'[t]-
>q4d,q5'[t]->q5d,qx'[t]->qxd,qy'[t]-
>qyd,qz'[t]->qzd,
```

```
q1''[t]->q1dd,q2''[t]->q2dd,q3''[t]-
>q3dd,q4''[t]->q4dd,q5''[t]->q5dd,qx''[t]-
>qxdd,qy''[t]->qydd,qz''[t]->qzdd};

torque3 = torque3/.{q4d q5d->q4dq5d,q3d
q5d->q3dq5d,q3d q4d->q3dq4d,q2d q5d-
>q2dq5d,
q2d q4d->q2dq4d,q2d q3d->q2dq3d,q1d
q5d->q1dq5d,
q1d q4d->q1dq4d,q1d q3d->q1dq3d,q1d
q2d->q1dq2d,q5d^2->q5dsq,q4d^2-
>q4dsq,q3d^2->q3dsq,
q2d^2->q2dsq,q1d^2->q1dsq,q5d qxd-
>q5dqxd,q5d qyd->q5dqyd,q4d qxd-
>q4dqxd,q4d qyd->q4dqyd,
q3d qxd->q3dqxd,q3d qyd->q3dqyd,q2d
qxd->q2dqxd,q2d qyd->q2dqyd,
q1d qxd->q1dqxd,q1d qyd->q1dqyd,
qxd^2->qxdsq,qyd^2->qydsq,qxd qyd-
>qxdqyd};

(*get rid of unnecessary mass matrix
element*)
torque3 = torque3/.{q5dd->0,q4dd->0};

torque3 = Expand[torque3];
torque3 = torque3//.{q4d q5d->q4dq5d,q3d
q5d->q3dq5d,q3d q4d->q3dq4d,q2d q5d-
>q2dq5d,
q2d q4d->q2dq4d,q2d q3d->q2dq3d,q1d
q5d->q1dq5d,
q1d q4d->q1dq4d,q1d q3d->q1dq3d,q1d
q2d->q1dq2d,q5d^2->q5dsq,q4d^2-
>q4dsq,q3d^2->q3dsq,
q2d^2->q2dsq,q1d^2->q1dsq,q5d qxd-
>q5dqxd,q5d qyd->q5dqyd,q4d qxd-
>q4dqxd,q4d qyd->q4dqyd,
q3d qxd->q3dqxd,q3d qyd->q3dqyd,q2d
qxd->q2dqxd,q2d qyd->q2dqyd,
q1d qxd->q1dqxd,q1d qyd->q1dqyd,
qxd^2->qxdsq,qyd^2->qydsq,qxd qyd-
>qxdqyd};

torque3 = Chop[torque3,precision];
torque3 = SetAccuracy[torque3,decimal];
torque3 = torque3//.{2.0000->2,1.0000-
>1};
file[1]="q4dq5d3";file[2]="q3dq5d3";file[3]
="q3dq4d3";file[4]="q2dq5d3";file[5]="q2d
q4d3";file[6]="q2dq3d3";file[7]="q1dq5d3"
;
file[8]="q1dq4d3";file[9]="q1dq3d3";file[10
]="q1dq2d3";file[11]="q5dsq3";file[12]="q4
dsq3";file[13]="q3dsq3";
file[14]="q2dsq3";file[15]="q1dsq3";file[16]
="q5dqxd3";file[17]="q5dqyd3";file[18]="q
4dqxd3";file[19]="q4dqyd3";
file[20]="q3dqxd3";file[21]="q3dqyd3";file[
22]="q2dqxd3";file[23]="q2dqyd3";file[24]
="q1dqxd3";file[25]="q1dqyd3";
file[26]="qxdsq3";file[27]="qydsq3";file[28]
="qxdqyd3";file[29]="m66";file[30]="m56";fi
le[31]="m46";file[32]="m36";file[33]="m26"
;
file[34]="m16";file[35]="G3";

variable[1]=q4dq5d;variable[2]=q3dq5d;vari
able[3]=q3dq4d;
variable[4]=q2dq5d;variable[5]=q2dq4d;vari
able[6]=q2dq3d;
variable[7]=q1dq5d;variable[8]=q1dq4d;vari
able[9]=q1dq3d;
variable[10]=q1dq2d;variable[11]=q5dsq;va
riable[12]=q4dsq;
variable[13]=q3dsq;variable[14]=q2dsq;vari
able[15]=q1dsq;
variable[16]=q5dqxd;variable[17]=q5dqyd;v
ariable[18]=q4dqxd;
variable[19]=q4dqyd;variable[20]=q3dqxd;v
ariable[21]=q3dqyd;
variable[22]=q2dqxd;variable[23]=q2dqyd;v
ariable[24]=q1dqxd;
variable[25]=q1dqyd;variable[26]=qxdsq;var
iable[27]=qydsq;
variable[28]=qxdqyd;variable[29]=q3dd;vari
able[30]=q2dd;variable[31]=q1dd;variable[3
2]=qydd;variable[33]=qxdd;variable[34]=zd
d;

(*AUTOMATION FOR ALL TERMS*)
For[counter=1,counter<=34,counter++,var
name=variable[counter];
expr = Coefficient[torque3,varname];
filename = file[counter];
expr = Simplify[expr];
expr = expr//.{Sin[qx]->s6,Sin[qy]-
>s7,Sin[q1]->s1,Sin[q2]->s2,Sin[q3]-
>s3,Sin[q4]->s4,Sin[q5]->s5,
Cos[qx]->c6,Cos[qy]->c7,Cos[q1]-
>c1,Cos[q2]->c2,Cos[q3]->c3,Cos[q4]-
>c4,Cos[q5]->c5};
expr = expr//.{2.0000->2,1.0000->1,
2.0000-> 2, 1.0000-> 1};
expr =
Collect[expr,{s1,c1,s2,c2,s3,c3,s4,c4,s5,c5
,s6,c6,s7,c7}];
expr = varname*expr;
torque3 = torque3/.{varname->0};
```

```
stmp = OpenWrite[filename];
WriteString[stmp,filename];
PutAppend["=",filename];
PutAppend[expr,filename];
PutAppend[";",filename];Close[stmp];];
torque3 =
Collect[torque3,{s1,c1,s2,c2,s3,c3,s4,c4,s
5,c5,s6,c6,s7,c7}];
filename = file[35];
stmp = OpenWrite[filename];
WriteString[stmp,filename];
PutAppend["=",filename];
PutAppend[torque3,filename];
PutAppend[";",filename];Close[stmp];

time=Date[];
Print["Derivation of torque3 finished
at:",time];
(***************** for torque2 **************)
twof2 = twor3.threef3+twoF2;
twof2 = Simplify[twof2];
twon2 =
twoN2+twor3.threen3+twopc2m.twoF2+tw
op3m.twor3.threef3;
twon2 = Simplify[twon2];
ClearAll[twor3,twop3m,twopc2m,twoF2,two
N2,threef3,threen3];
Remove[twor3,twop3m,twopc2m,twoF2,tw
oN2,threef3,threen3];

torque2 = twon2[[3,1]];
torque2 = SSonerule[torque2];
torque2 = torque2 /.{q1[t]->q1,q2[t]-
>q2,q3[t]->q3,q4[t]->q4,q5[t]->q5,qx[t]-
>qx,qy[t]->qy,qz[t]->qz,
q1'[t]->q1d,q2'[t]->q2d,q3'[t]->q3d,q4'[t]-
>q4d,q5'[t]->q5d,qx'[t]->qxd,qy'[t]-
>qyd,qz'[t]->qzd,
q1''[t]->q1dd,q2''[t]->q2dd,q3''[t]-
>q3dd,q4''[t]->q4dd,q5''[t]->q5dd,qx''[t]-
>qxdd,qy''[t]->qydd,qz''[t]->qzdd};

torque2 = torque2/.{q4d q5d->q4dq5d,q3d
q5d->q3dq5d,q3d q4d->q3dq4d,q2d q5d-
>q2dq5d,
q2d q4d->q2dq4d,q2d q3d->q2dq3d,q1d
q5d->q1dq5d,
q1d q4d->q1dq4d,q1d q3d->q1dq3d,q1d
q2d->q1dq2d,q5d^2->q5dsq,q4d^2-
>q4dsq,q3d^2->q3dsq,
q2d^2->q2dsq,q1d^2->q1dsq,q5d qxd-
>q5dqxd,q5d qyd->q5dqyd,q4d qxd-
>q4dqxd,q4d qyd->q4dqyd,
q3d qxd->q3dqxd,q3d qyd->q3dqyd,q2d
qxd->q2dqxd,q2d qyd->q2dqyd,
q1d qxd->q1dqxd,q1d qyd->q1dqyd,
qxd^2->qxdsq,qyd^2->qydsq,qxd qyd-
>qxdqyd};

(*get rid of unnecessary mass matrix
element*)
torque2 = torque2/.{q5dd->0,q4dd-
>0,q3dd->0};

torque2 = Expand[torque2];
torque2 = torque2//.{q4d q5d->q4dq5d,q3d
q5d->q3dq5d,q3d q4d->q3dq4d,q2d q5d-
>q2dq5d,
q2d q4d->q2dq4d,q2d q3d->q2dq3d,q1d
q5d->q1dq5d,
q1d q4d->q1dq4d,q1d q3d->q1dq3d,q1d
q2d->q1dq2d,q5d^2->q5dsq,q4d^2-
>q4dsq,q3d^2->q3dsq,
q2d^2->q2dsq,q1d^2->q1dsq,q5d qxd-
>q5dqxd,q5d qyd->q5dqyd,q4d qxd-
>q4dqxd,q4d qyd->q4dqyd,
q3d qxd->q3dqxd,q3d qyd->q3dqyd,q2d
qxd->q2dqxd,q2d qyd->q2dqyd,
q1d qxd->q1dqxd,q1d qyd->q1dqyd,
qxd^2->qxdsq,qyd^2->qydsq,qxd qyd-
>qxdqyd};

torque2 = Chop[torque2,precision];
torque2 = SetAccuracy[torque2,decimal];
torque2 = torque2//.{2.0000->2,1.0000-
>1};
file[1]="q4dq5d2";file[2]="q3dq5d2";file[3]
="q3dq4d2";file[4]="q2dq5d2";file[5]="q2d
q4d2";file[6]="q2dq3d2";file[7]="q1dq5d2"
;
file[8]="q1dq4d2";file[9]="q1dq3d2";file[10
]="q1dq2d2";file[11]="q5dsq2";file[12]="q4
dsq2";file[13]="q3dsq2";
file[14]="q2dsq2";file[15]="q1dsq2";file[16]
="q5dqxd2";file[17]="q5dqyd2";file[18]="q
4dqxd2";file[19]="q4dqyd2";
file[20]="q3dqxd2";file[21]="q3dqyd2";file[
22]="q2dqxd2";file[23]="q2dqyd2";file[24]
="q1dqxd2";file[25]="q1dqyd2";
file[26]="qxdsq2";file[27]="qydsq2";file[28]
="qxdqyd2";file[29]="m55";file[30]="m45";fi
le[31]="m35";file[32]="m25";file[33]="m15"
;
file[34]="G2";

variable[1]=q4dq5d;variable[2]=q3dq5d;vari
able[3]=q3dq4d;
```

```
variable[4]=q2dq5d;variable[5]=q2dq4d;vari
able[6]=q2dq3d;
variable[7]=q1dq5d;variable[8]=q1dq4d;vari
able[9]=q1dq3d;
variable[10]=q1dq2d;variable[11]=q5dsq;va
riable[12]=q4dsq;
variable[13]=q3dsq;variable[14]=q2dsq;vari
able[15]=q1dsq;
variable[16]=q5dqxd;variable[17]=q5dqyd;v
ariable[18]=q4dqxd;
variable[19]=q4dqyd;variable[20]=q3dqxd;v
ariable[21]=q3dqyd;
variable[22]=q2dqxd;variable[23]=q2dqyd;v
ariable[24]=q1dqxd;
variable[25]=q1dqyd;variable[26]=qxdsq;var
iable[27]=qydsq;
variable[28]=qxdqyd;variable[29]=q2dd;vari
able[30]=q1dd;variable[31]=qydd;variable[3
2]=qxdd;variable[33]=zdd;

(*AUTOMATION FOR ALL TERMS*)
For[counter=1,counter<=33,counter++,var
name=variable[counter];
expr = Coefficient[torque2,varname];
filename = file[counter];
expr = Simplify[expr];
expr = expr//.{Sin[qx]->s6,Sin[qy]-
>s7,Sin[q1]->s1,Sin[q2]->s2,Sin[q3]-
>s3,Sin[q4]->s4,Sin[q5]->s5,
Cos[qx]->c6,Cos[qy]->c7,Cos[q1]-
>c1,Cos[q2]->c2,Cos[q3]->c3,Cos[q4]-
>c4,Cos[q5]->c5};
expr = expr//.{2.0000->2,1.0000->1,
2.0000-> 2, 1.0000-> 1};
expr =
Collect[expr,{s1,c1,s2,c2,s3,c3,s4,c4,s5,c5
,s6,c6,s7,c7}];
expr = varname*expr;
torque2 = torque2/.{varname->0};
stmp = OpenWrite[filename];
WriteString[stmp,filename];
PutAppend["=",filename];
PutAppend[expr,filename];
PutAppend[";",filename];Close[stmp];];
torque2 =
Collect[torque2,{s1,c1,s2,c2,s3,c3,s4,c4,s
5,c5,s6,c6,s7,c7}];
filename = file[34];
stmp = OpenWrite[filename];
WriteString[stmp,filename];
PutAppend["=",filename];
PutAppend[torque2,filename];
PutAppend[";",filename];Close[stmp];
time=Date[];
Print["Derivation of torque2 finished
at:",time];

(****************** for torque1 **************)
onef1 = oner2.twof2+oneF1;
onef1 = Simplify[onef1];
onen1 =
oneN1+oner2.twon2+onepc1m.oneF1+on
ep2m.oner2.twof2;
onen1 = Simplify[onen1];
ClearAll[oner2,onep2,onep2m,onepc1,one
pc1m,oneN1,oneF1,twof2,twon2];
Remove[oner2,onep2,onep2m,onepc1,on
epc1m,oneN1,oneF1,twof2,twon2];

torque1 = onen1[[3,1]];
torque1 = SSonerule[torque1];
torque1 = torque1 /.{q1[t]->q1,q2[t]-
>q2,q3[t]->q3,q4[t]->q4,q5[t]->q5,qx[t]-
>qx,qy[t]->qy,qz[t]->qz,
q1'[t]->q1d,q2'[t]->q2d,q3'[t]->q3d,q4'[t]-
>q4d,q5'[t]->q5d,qx'[t]->qxd,qy'[t]-
>qyd,qz'[t]->qzd,
q1"[t]->q1dd,q2"[t]->q2dd,q3"[t]-
>q3dd,q4"[t]->q4dd,q5"[t]->q5dd,qx"[t]-
>qxdd,qy"[t]->qydd,qz"[t]->qzdd};

torque1 = torque1/.{q4d q5d->q4dq5d,q3d
q5d->q3dq5d,q3d q4d->q3dq4d,q2d q5d-
>q2dq5d,
q2d q4d->q2dq4d,q2d q3d->q2dq3d,q1d
q5d->q1dq5d,
q1d q4d->q1dq4d,q1d q3d->q1dq3d,q1d
q2d->q1dq2d,q5d^2->q5dsq,q4d^2-
>q4dsq,q3d^2->q3dsq,
q2d^2->q2dsq,q1d^2->q1dsq,q5d qxd-
>q5dqxd,q5d qyd->q5dqyd,q4d qxd-
>q4dqxd,q4d qyd->q4dqyd,
q3d qxd->q3dqxd,q3d qyd->q3dqyd,q2d
qxd->q2dqxd,q2d qyd->q2dqyd,
q1d qxd->q1dqxd,q1d qyd->q1dqyd,
qxd^2->qxdsq,qyd^2->qydsq,qxd qyd-
>qxdqyd};

(*get rid of unnecessary mass matrix
element*)
torque1 = torque1/.{q5dd->0,q4dd-
>0,q3dd->0,q2dd->0};

torque1 = Expand[torque1];
torque1 = torque1//.{q4d q5d->q4dq5d,q3d
q5d->q3dq5d,q3d q4d->q3dq4d,q2d q5d-
>q2dq5d,
```

```
q2d q4d->q2dq4d,q2d q3d->q2dq3d,q1d
q5d->q1dq5d,
q1d q4d->q1dq4d,q1d q3d->q1dq3d,q1d
q2d->q1dq2d,q5d^2->q5dsq,q4d^2-
>q4dsq,q3d^2->q3dsq,
q2d^2->q2dsq,q1d^2->q1dsq,q5d qxd-
>q5dqxd,q5d qyd->q5dqyd,q4d qxd-
>q4dqxd,q4d qyd->q4dqyd,
q3d qxd->q3dqxd,q3d qyd->q3dqyd,q2d
qxd->q2dqxd,q2d qyd->q2dqyd,
q1d qxd->q1dqxd,q1d qyd->q1dqyd,
qxd^2->qxdsq,qyd^2->qydsq,qxd qyd-
>qxdqyd};

torque1 = Chop[torque1,precision];
torque1 = SetAccuracy[torque1,decimal];
torque1 = torque1//.{2.0000->2,1.0000-
>1};
file[1]="q4dq5d1";file[2]="q3dq5d1";file[3]
="q3dq4d1";file[4]="q2dq5d1";file[5]="q2d
q4d1";file[6]="q2dq3d1";file[7]="q1dq5d1"
;
file[8]="q1dq4d1";file[9]="q1dq3d1";file[10
]="q1dq2d1";file[11]="q5dsq1";file[12]="q4
dsq1";file[13]="q3dsq1";
file[14]="q2dsq1";file[15]="q1dsq1";file[16]
="q5dqxd1";file[17]="q5dqyd1";file[18]="q
4dqxd1";file[19]="q4dqyd1";
file[20]="q3dqxd1";file[21]="q3dqyd1";file[
22]="q2dqxd1";file[23]="q2dqyd1";file[24]
="q1dqxd1";file[25]="q1dqyd1";
file[26]="qxdsq1";file[27]="qydsq1";file[28]
="qxdqyd1";file[29]="m44";file[30]="m34";fi
le[31]="m24";file[32]="m14";file[33]="G1";

variable[1]=q4dq5d;variable[2]=q3dq5d;vari
able[3]=q3dq4d;
variable[4]=q2dq5d;variable[5]=q2dq4d;vari
able[6]=q2dq3d;
variable[7]=q1dq5d;variable[8]=q1dq4d;vari
able[9]=q1dq3d;
variable[10]=q1dq2d;variable[11]=q5dsq;va
riable[12]=q4dsq;
variable[13]=q3dsq;variable[14]=q2dsq;vari
able[15]=q1dsq;
variable[16]=q5dqxd;variable[17]=q5dqyd;v
ariable[18]=q4dqxd;
variable[19]=q4dqyd;variable[20]=q3dqxd;v
ariable[21]=q3dqyd;
variable[22]=q2dqxd;variable[23]=q2dqyd;v
ariable[24]=q1dqxd;
variable[25]=q1dqyd;variable[26]=qxdsq;var
iable[27]=qydsq;
variable[28]=qxdqyd;variable[29]=q1dd;vari
able[30]=qydd;variable[31]=qxdd;variable[3
2]=zdd;

(*AUTOMATION FOR ALL TERMS*)
For[counter=1,counter<=32,counter++,var
name=variable[counter];
expr = Coefficient[torque1,varname];
filename = file[counter];
expr = Simplify[expr];
expr = expr//.{Sin[qx]->s6,Sin[qy]-
>s7,Sin[q1]->s1,Sin[q2]->s2,Sin[q3]-
>s3,Sin[q4]->s4,Sin[q5]->s5,
Cos[qx]->c6,Cos[qy]->c7,Cos[q1]-
>c1,Cos[q2]->c2,Cos[q3]->c3,Cos[q4]-
>c4,Cos[q5]->c5};
expr = expr//.{2.0000->2,1.0000->1,
2.0000-> 2, 1.0000-> 1};
expr =
Collect[expr,{s1,c1,s2,c2,s3,c3,s4,c4,s5,c5
,s6,c6,s7,c7}];
expr = varname*expr;
torque1 = torque1/.{varname->0};
stmp = OpenWrite[filename];
WriteString[stmp,filename];
PutAppend["=",filename];
PutAppend[expr,filename];
PutAppend[";",filename];Close[stmp];];
torque1 =
Collect[torque1,{s1,c1,s2,c2,s3,c3,s4,c4,s
5,c5,s6,c6,s7,c7}];
filename = file[33];
stmp = OpenWrite[filename];
WriteString[stmp,filename];
PutAppend["=",filename];
PutAppend[torque1,filename];
PutAppend[";",filename];Close[stmp];

time=Date[];
Print["Derivation of torque1 finished
at:",time];

(****************** for fz **************)
zerof0 = zeror1.onef1+zeroF0;
zerof0 = Simplify[zerof0];
zeron0 =
zeroN0+zeror1.onen1+zeropc0m.zeroF0+z
erop1m.zeror1.onef1;
zeron0 = Simplify[zeron0];
ClearAll[zeror1,zerop1,zerop1m,zeropc0,ze
ropc0m,zeroN0,zeroF0,onen1,onef1];
Remove[zeror1,zerop1,zerop1m,zeropc0,z
eropc0m,zeroN0,zeroF0,onen1,onef1];
```

```
wf0 = wr0.zerof0;
ClearAll[zerof0];
Remove[zerof0];

fz = wf0[[3,1]];
fz = SSonerule[fz];
fz = fz/.{q1[t]->q1,q2[t]->q2,q3[t]->q3,q4[t]-
>q4,q5[t]->q5,qx[t]->qx,qy[t]->qy,qz[t]->qz,
q1'[t]->q1d,q2'[t]->q2d,q3'[t]->q3d,q4'[t]-
>q4d,q5'[t]->q5d,qx'[t]->qxd,qy'[t]-
>qyd,qz'[t]->qzd,
q1''[t]->q1dd,q2''[t]->q2dd,q3''[t]-
>q3dd,q4''[t]->q4dd,q5''[t]->q5dd,qx''[t]-
>qxdd,qy''[t]->qydd,qz''[t]->qzdd};

fz = fz/.{q4d q5d->q4dq5d,q3d q5d-
>q3dq5d,q3d q4d->q3dq4d,q2d q5d-
>q2dq5d,
q2d q4d->q2dq4d,q2d q3d->q2dq3d,q1d
q5d->q1dq5d,
q1d q4d->q1dq4d,q1d q3d->q1dq3d,q1d
q2d->q1dq2d,q5d^2->q5dsq,q4d^2-
>q4dsq,q3d^2->q3dsq,
q2d^2->q2dsq,q1d^2->q1dsq,q5d qxd-
>q5dqxd,q5d qyd->q5dqyd,q4d qxd-
>q4dqxd,q4d qyd->q4dqyd,
q3d qxd->q3dqxd,q3d qyd->q3dqyd,q2d
qxd->q2dqxd,q2d qyd->q2dqyd,
q1d qxd->q1dqxd,q1d qyd->q1dqyd,
qxd^2->qxdsq,qyd^2->qydsq,qxd qyd-
>qxdqyd};

(*get rid of unnecessary mass matrix
element*)
fz = fz/.{q5dd->0,q4dd->0,q3dd->0,q2dd-
>0,q1dd->0};

fz = Expand[fz];
fz = fz//.{q4d q5d->q4dq5d,q3d q5d-
>q3dq5d,q3d q4d->q3dq4d,q2d q5d-
>q2dq5d,
q2d q4d->q2dq4d,q2d q3d->q2dq3d,q1d
q5d->q1dq5d,
q1d q4d->q1dq4d,q1d q3d->q1dq3d,q1d
q2d->q1dq2d,q5d^2->q5dsq,q4d^2-
>q4dsq,q3d^2->q3dsq,
q2d^2->q2dsq,q1d^2->q1dsq,q5d qxd-
>q5dqxd,q5d qyd->q5dqyd,q4d qxd-
>q4dqxd,q4d qyd->q4dqyd,
q3d qxd->q3dqxd,q3d qyd->q3dqyd,q2d
qxd->q2dqxd,q2d qyd->q2dqyd,
q1d qxd->q1dqxd,q1d qyd->q1dqyd,
qxd^2->qxdsq,qyd^2->qydsq,qxd qyd-
>qxdqyd};

fz = Chop[fz,precision];
fz = SetAccuracy[fz,decimal];
fz = fz//.{2.0000->2,1.0000->1};
file[1]="q4dq5dz";file[2]="q3dq5dz";file[3]=
"q3dq4dz";file[4]="q2dq5dz";file[5]="q2dq
4dz";file[6]="q2dq3dz";file[7]="q1dq5dz";
file[8]="q1dq4dz";file[9]="q1dq3dz";file[10]
="q1dq2dz";file[11]="q5dsqz";file[12]="q4
dsqz";file[13]="q3dsqz";
file[14]="q2dsqz";file[15]="q1dsqz";file[16]
="q5dqxdz";file[17]="q5dqydz";file[18]="q4
dqxdz";file[19]="q4dqydz";
file[20]="q3dqxdz";file[21]="q3dqydz";file[2
2]="q2dqxdz";file[23]="q2dqydz";file[24]="
q1dqxdz";file[25]="q1dqydz";
file[26]="qxdsqz";file[27]="qydsqz";file[28]
="qxdqydz";file[29]="m13";file[30]="m12";fi
le[31]="m11";file[32]="Gz";

variable[1]=q4dq5d;variable[2]=q3dq5d;vari
able[3]=q3dq4d;
variable[4]=q2dq5d;variable[5]=q2dq4d;vari
able[6]=q2dq3d;
variable[7]=q1dq5d;variable[8]=q1dq4d;vari
able[9]=q1dq3d;
variable[10]=q1dq2d;variable[11]=q5dsq;va
riable[12]=q4dsq;
variable[13]=q3dsq;variable[14]=q2dsq;vari
able[15]=q1dsq;
variable[16]=q5dqxd;variable[17]=q5dqyd;v
ariable[18]=q4dqxd;
variable[19]=q4dqyd;variable[20]=q3dqxd;v
ariable[21]=q3dqyd;
variable[22]=q2dqxd;variable[23]=q2dqyd;v
ariable[24]=q1dqxd;
variable[25]=q1dqyd;variable[26]=qxdsq;var
iable[27]=qydsq;
variable[28]=qxdqyd;variable[29]=qydd;vari
able[30]=qxdd;variable[31]=zdd;

(*AUTOMATION FOR ALL TERMS*)
For[counter=1,counter<=31,counter++,var
name=variable[counter];
expr = Coefficient[fz,varname];
filename = file[counter];
expr = Simplify[expr];
expr = expr//.{Sin[qx]->s6,Sin[qy]-
>s7,Sin[q1]->s1,Sin[q2]->s2,Sin[q3]-
>s3,Sin[q4]->s4,Sin[q5]->s5,
Cos[qx]->c6,Cos[qy]->c7,Cos[q1]-
>c1,Cos[q2]->c2,Cos[q3]->c3,Cos[q4]-
>c4,Cos[q5]->c5};
```

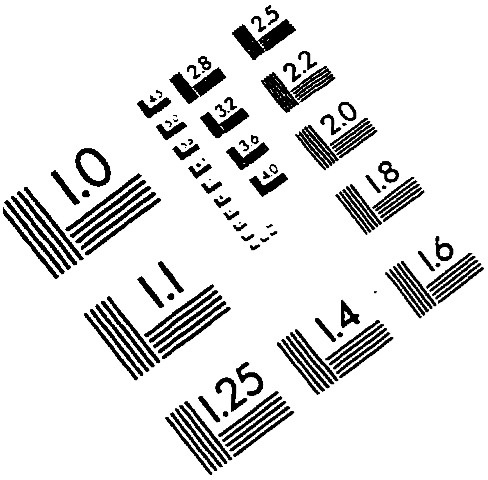
```
expr = expr//.{2.0000->2,1.0000->1,
2.0000-> 2, 1.0000-> 1};
expr =
Collect[expr,{s1,c1,s2,c2,s3,c3,s4,c4,s5,c5
,s6,c6,s7,c7}];
expr = varname*expr;
fz = fz/.{varname->0};
stmp = OpenWrite[filename];
WriteString[stmp,filename];
PutAppend["=",filename];
PutAppend[expr,filename];
PutAppend[";",filename];Close[stmp];];
fz =
Collect[fz,{s1,c1,s2,c2,s3,c3,s4,c4,s5,c5,s
6,c6,s7,c7}];
filename = file[32];
stmp = OpenWrite[filename];
WriteString[stmp,filename];
PutAppend["=",filename];
PutAppend[fz,filename];
PutAppend[";",filename];Close[stmp];

time=Date[];
Print["Derivation of fz finished at:",time];
(****************** for nx ***************)
time=Date[];
Print["time1=",time];
wn0 = wr0.zeron0;
time=Date[];
Print["time2=",time];
ClearAll[zeron0,wr0];
Remove[zeron0,wr0];

nx = wn0[[1,1]];
time=Date[];
Print["time3=",time];
ny = wn0[[2,1]];
ClearAll[wn0];
Remove[wn0];
time=Date[];
Print["time4=",time];
nx = SSonerule[nx];
time=Date[];
Print["time5=",time];

nx = nx/.{q1[t]->q1,q2[t]->q2,q3[t]-
>q3,q4[t]->q4,q5[t]->q5,qx[t]->qx,qy[t]-
>qy,qz[t]->qz,
q1'[t]->q1d,q2'[t]->q2d,q3'[t]->q3d,q4'[t]-
>q4d,q5'[t]->q5d,qx'[t]->qxd,qy'[t]-
>qyd,qz'[t]->qzd,
q1''[t]->q1dd,q2''[t]->q2dd,q3''[t]-
>q3dd,q4''[t]->q4dd,q5''[t]->q5dd,qx''[t]-
>qxdd,qy''[t]->qydd,qz''[t]->qzdd};
time=Date[];
Print["time6=",time];
nx = nx/.{q4d q5d->q4dq5d,q3d q5d-
>q3dq5d,q3d q4d->q3dq4d,q2d q5d-
>q2dq5d,
q2d q4d->q2dq4d,q2d q3d->q2dq3d,q1d
q5d->q1dq5d,
q1d q4d->q1dq4d,q1d q3d->q1dq3d,q1d
q2d->q1dq2d,q5d^2->q5dsq,q4d^2-
>q4dsq,q3d^2->q3dsq,
q2d^2->q2dsq,q1d^2->q1dsq,q5d qxd-
>q5dqxd,q5d qyd->q5dqyd,q4d qxd-
>q4dqxd,q4d qyd->q4dqyd,
q3d qxd->q3dqxd,q3d qyd->q3dqyd,q2d
qxd->q2dqxd,q2d qyd->q2dqyd,
q1d qxd->q1dqxd,q1d qyd->q1dqyd,
qxd^2->qxdsq,qyd^2->qydsq,qxd qyd-
>qxdqyd};
time=Date[];
Print["time7=",time];
(*get rid of unnecessary mass matrix
element*)
nx = nx/.{q5dd->0,q4dd->0,q3dd->0,q2dd-
>0,q1dd->0,zdd->0};
time=Date[];
Print["time8=",time];
nx = Expand[nx];
time=Date[];
Print["time9=",time];
nx = nx//.{q4d q5d->q4dq5d,q3d q5d-
>q3dq5d,q3d q4d->q3dq4d,q2d q5d-
>q2dq5d,
q2d q4d->q2dq4d,q2d q3d->q2dq3d,q1d
q5d->q1dq5d,
q1d q4d->q1dq4d,q1d q3d->q1dq3d,q1d
q2d->q1dq2d,q5d^2->q5dsq,q4d^2-
>q4dsq,q3d^2->q3dsq,
q2d^2->q2dsq,q1d^2->q1dsq,q5d qxd-
>q5dqxd,q5d qyd->q5dqyd,q4d qxd-
>q4dqxd,q4d qyd->q4dqyd,
q3d qxd->q3dqxd,q3d qyd->q3dqyd,q2d
qxd->q2dqxd,q2d qyd->q2dqyd,
q1d qxd->q1dqxd,q1d qyd->q1dqyd,
qxd^2->qxdsq,qyd^2->qydsq,qxd qyd-
>qxdqyd};
time=Date[];
Print["time10=",time];
nx = Chop[nx,precision];
time=Date[];
Print["time11=",time];
nx = SetAccuracy[nx,decimal];
nx = nx//.{2.0000->2,1.0000->1};
time=Date[];
Print["time11a=",time];
```

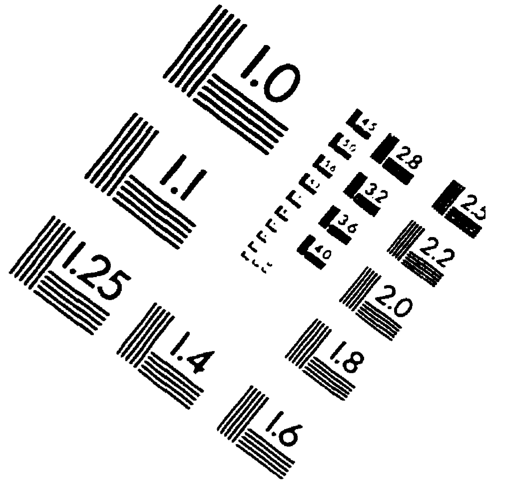
```
file[1]="q4dq5dx";file[2]="q3dq5dx";file[3]=
"q3dq4dx";file[4]="q2dq5dx";file[5]="q2dq
4dx";file[6]="q2dq3dx";file[7]="q1dq5dx";
file[8]="q1dq4dx";file[9]="q1dq3dx";file[10]
="q1dq2dx";file[11]="q5dsqx";file[12]="q4
dsqx";file[13]="q3dsqx";
file[14]="q2dsqx";file[15]="q1dsqx";file[16]
="q5dqxdx";file[17]="q5dqydx";file[18]="q4
dqxdx";file[19]="q4dqydx";
file[20]="q3dqxdx";file[21]="q3dqydx";file[2
2]="q2dqxdx";file[23]="q2dqydx";file[24]="
q1dqxdx";file[25]="q1dqydx";
file[26]="qxdsqx";file[27]="qydsqx";file[28]
="qxdqydx";file[29]="m23";file[30]="m22";fi
le[31]="Gx";

variable[1]=q4dq5d;variable[2]=q3dq5d;vari
able[3]=q3dq4d;
variable[4]=q2dq5d;variable[5]=q2dq4d;vari
able[6]=q2dq3d;
variable[7]=q1dq5d;variable[8]=q1dq4d;vari
able[9]=q1dq3d;
variable[10]=q1dq2d;variable[11]=q5dsq;va
riable[12]=q4dsq;
variable[13]=q3dsq;variable[14]=q2dsq;vari
able[15]=q1dsq;
variable[16]=q5dqxd;variable[17]=q5dqyd;v
ariable[18]=q4dqxd;
variable[19]=q4dqyd;variable[20]=q3dqxd;v
ariable[21]=q3dqyd;
variable[22]=q2dqxd;variable[23]=q2dqyd;v
ariable[24]=q1dqxd;
variable[25]=q1dqyd;variable[26]=qxdsq;var
iable[27]=qydsq;
variable[28]=qxdqyd;variable[29]=qydd;vari
able[30]=qxdd;

(*AUTOMATION FOR ALL TERMS*)
For[counter=1,counter<=30,counter++,var
name=variable[counter];
expr = Coefficient[nx,varname];
filename = file[counter];
expr = Simplify[expr];
expr = expr//.{Sin[qx]->s6,Sin[qy]-
>s7,Sin[q1]->s1,Sin[q2]->s2,Sin[q3]-
>s3,Sin[q4]->s4,Sin[q5]->s5,
Cos[qx]->c6,Cos[qy]->c7,Cos[q1]-
>c1,Cos[q2]->c2,Cos[q3]->c3,Cos[q4]-
>c4,Cos[q5]->c5};
expr = expr//.{2.0000->2,1.0000->1,
2.0000-> 2, 1.0000-> 1};
expr =
Collect[expr,{s1,c1,s2,c2,s3,c3,s4,c4,s5,c5
,s6,c6,s7,c7}];
expr = varname*expr;
nx = nx/.{varname->0};
stmp = OpenWrite[filename];
WriteString[stmp,filename];
PutAppend["=",filename];
PutAppend[expr,filename];
PutAppend[";",filename];Close[stmp];];
nx =
Collect[nx,{s1,c1,s2,c2,s3,c3,s4,c4,s5,c5,s
6,c6,s7,c7}];
filename = file[31];
stmp = OpenWrite[filename];
WriteString[stmp,filename];
PutAppend["=",filename];
PutAppend[nx,filename];
PutAppend[";",filename];Close[stmp];

time=Date[];
Print["Derivation of nx finished at:",time];

(****************** for ny ***************)
ny = SSonerule[ny];
time=Date[];
Print["time14=",time];
ny = ny/.{q1[t]->q1,q2[t]->q2,q3[t]-
>q3,q4[t]->q4,q5[t]->q5,qx[t]->qx,qy[t]-
>qy,qz[t]->qz,
q1'[t]->q1d,q2'[t]->q2d,q3'[t]->q3d,q4'[t]-
>q4d,q5'[t]->q5d,qx'[t]->qxd,qy'[t]-
>qyd,qz'[t]->qzd,
q1"[t]->q1dd,q2"[t]->q2dd,q3"[t]-
>q3dd,q4"[t]->q4dd,q5"[t]->q5dd,qx"[t]-
>qxdd,qy"[t]->qydd,qz"[t]->qzdd};
time=Date[];
Print["time15=",time];
ny = ny/.{q4d q5d->q4dq5d,q3d q5d-
>q3dq5d,q3d q4d->q3dq4d,q2d q5d-
>q2dq5d,
q2d q4d->q2dq4d,q2d q3d->q2dq3d,q1d
q5d->q1dq5d,
q1d q4d->q1dq4d,q1d q3d->q1dq3d,q1d
q2d->q1dq2d,q5d^2->q5dsq,q4d^2-
>q4dsq,q3d^2->q3dsq,
q2d^2->q2dsq,q1d^2->q1dsq,q5d qxd-
>q5dqxd,q5d qyd->q5dqyd,q4d qxd-
>q4dqxd,q4d qyd->q4dqyd,
q3d qxd->q3dqxd,q3d qyd->q3dqyd,q2d
qxd->q2dqxd,q2d qyd->q2dqyd,
q1d qxd->q1dqxd,q1d qyd->q1dqyd,
qxd^2->qxdsq,qyd^2->qydsq,qxd qyd-
>qxdqyd};
```

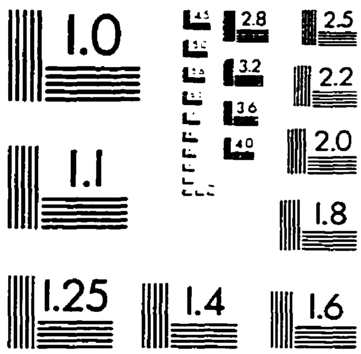
```
time=Date[];
Print["time16=",time];
(*get rid of unnecessary mass matrix
element*)
ny = ny/.{q5dd->0,q4dd->0,q3dd->0,q2dd-
>0,q1dd->0,zdd->0,qxdd->0};
time=Date[];
Print["time17=",time];
ny = Expand[ny];
time=Date[];
Print["time18=",time];
ny = ny//.{q4d q5d->q4dq5d,q3d q5d-
>q3dq5d,q3d q4d->q3dq4d,q2d q5d-
>q2dq5d,
q2d q4d->q2dq4d,q2d q3d->q2dq3d,q1d
q5d->q1dq5d,
q1d q4d->q1dq4d,q1d q3d->q1dq3d,q1d
q2d->q1dq2d,q5d^2->q5dsq,q4d^2-
>q4dsq,q3d^2->q3dsq,
q2d^2->q2dsq,q1d^2->q1dsq,q5d qxd-
>q5dqxd,q5d qyd->q5dqyd,q4d qxd-
>q4dqxd,q4d qyd->q4dqyd,
q3d qxd->q3dqxd,q3d qyd->q3dqyd,q2d
qxd->q2dqxd,q2d qyd->q2dqyd,
q1d qxd->q1dqxd,q1d qyd->q1dqyd,
qxd^2->qxdsq,qyd^2->qydsq,qxd qyd-
>qxdqyd};
time=Date[];
Print["time19=",time];
ny = Chop[ny,precision];
time=Date[];
Print["time20=",time];
ny = SetAccuracy[ny,decimal];
ny = ny//.{2.0000->2,1.0000->1};
time=Date[];
Print["time21=",time];
file[1]="q4dq5dy";file[2]="q3dq5dy";file[3]=
"q3dq4dy";file[4]="q2dq5dy";file[5]="q2dq
4dy";file[6]="q2dq3dy";file[7]="q1dq5dy";
file[8]="q1dq4dy";file[9]="q1dq3dy";file[10]
="q1dq2dy";file[11]="q5dsqy";file[12]="q4
dsqy";file[13]="q3dsqy";
file[14]="q2dsqy";file[15]="q1dsqy";file[16]
="q5dqxdy";file[17]="q5dqydy";file[18]="q4
dqxdy";file[19]="q4dqydy";
file[20]="q3dqxdy";file[21]="q3dqydy";file[2
2]="q2dqxdy";file[23]="q2dqydy";file[24]="
q1dqxdy";file[25]="q1dqydy";
file[26]="qxdsqy";file[27]="qydsqy";file[28]
="qxdqydy";file[29]="m33";file[30]="Gy";

variable[1]=q4dq5d;variable[2]=q3dq5d;vari
able[3]=q3dq4d;
variable[4]=q2dq5d;variable[5]=q2dq4d;vari
able[6]=q2dq3d;
variable[7]=q1dq5d;variable[8]=q1dq4d;vari
able[9]=q1dq3d;
variable[10]=q1dq2d;variable[11]=q5dsq;va
riable[12]=q4dsq;
variable[13]=q3dsq;variable[14]=q2dsq;vari
able[15]=q1dsq;
variable[16]=q5dqxd;variable[17]=q5dqyd;v
ariable[18]=q4dqxd;
variable[19]=q4dqyd;variable[20]=q3dqxd;v
ariable[21]=q3dqyd;
variable[22]=q2dqxd;variable[23]=q2dqyd;v
ariable[24]=q1dqxd;
variable[25]=q1dqyd;variable[26]=qxdsq;var
iable[27]=qydsq;
variable[28]=qxdqyd;variable[29]=qydd;

(*AUTOMATION FOR ALL TERMS*)

For[counter=1,counter<=29,counter++,var
name=variable[counter];
expr = Coefficient[ny,varname];
filename = file[counter];
expr = Simplify[expr];
expr = expr//.{Sin[qx]->s6,Sin[qy]-
>s7,Sin[q1]->s1,Sin[q2]->s2,Sin[q3]-
>s3,Sin[q4]->s4,Sin[q5]->s5,
Cos[qx]->c6,Cos[qy]->c7,Cos[q1]-
>c1,Cos[q2]->c2,Cos[q3]->c3,Cos[q4]-
>c4,Cos[q5]->c5};
expr = expr//.{2.0000->2,1.0000->1,
2.0000-> 2, 1.0000-> 1};
expr =
Collect[expr,{s1,c1,s2,c2,s3,c3,s4,c4,s5,c5
,s6,c6,s7,c7}];
expr = varname*expr;
ny = ny/.{varname->0};
stmp = OpenWrite[filename];
WriteString[stmp,filename];
PutAppend["=",filename];
PutAppend[expr,filename];
PutAppend[";",filename];Close[stmp];];
ny =
Collect[ny,{s1,c1,s2,c2,s3,c3,s4,c4,s5,c5,s
6,c6,s7,c7}];
filename = file[30];
stmp = OpenWrite[filename];
WriteString[stmp,filename];
PutAppend["=",filename];
PutAppend[ny,filename];
PutAppend[";",filename];Close[stmp];

time=Date[];
```

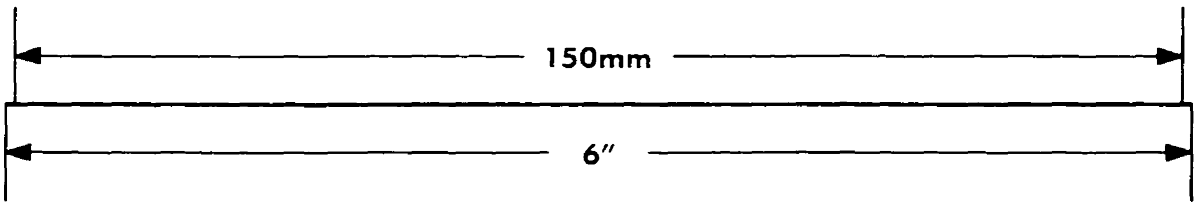
```
Print["Derivation of ny finished at:",time];
time=Date[];
Print["time22=",time];
Quit[];
```

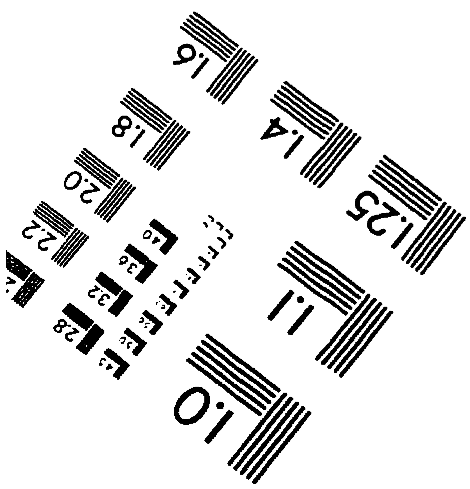
IMAGE EVALUATION TEST TARGET (QA-3)











APPLIED IMAGE, Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989