NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Simulation of Running

by

William Scott Selbie

B.A.Sc. Engineering Physics
University of British Columbia, 1980

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OI
THE REQUIREMENTS FOR THE DEGREE OF
Doctor of Philosophy
in the School
of
Kinesiology

SIMON FRASER UNIVERSI
December 1989

ISBN 0-315-66237-9

APPROVAL

Name: William Scott Selbie

Degree: Doctor of Philosophy

Title of Thesis: SIMULATION OF RUNNING

Examining Committee:

Chairman: Dr. D. Goodman

Dr. A.E. Chapman
Senior Supervisor

Dr. T.W. Calvert

Dr. J.C. Dill
School of Engineering
Simon Fraser University
Internal Examiner

Dr. J. Wilhelms
Computing and Information Sciences
University of California, Santa Cruz
External Examiner

Date Approved: Dec 13th 1989

PARTIAL COPYRIGHT LICENSE

I hereby grant to Simon Fraser University the right to lend my thesis or dissertation (the title of which is shown below) to users of the Simon Fraser University Library, and to make partial or single copies only for such users or in response to a request from the library of any other university, or other educational institution, on its own behalf or for one of its users. I further agree that permission for multiple copying of this thesis for scholarly purposes may be granted by me or the Dean of Graduate Studies. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Title of Thesis/Dissertation:

SIMULATION OF RUNNING

Author:

(signature)

WILLIAM SCOTT SELBIE

(name)

JANUARY 1, 1990

(date)

Abstract

The primary focus of this research was the development of a computer simulation of bipedal running. Joint moments and the initial kinematic state are the variable inputs to the system. The objective was a reasonable computer generated reproduction of a human running performance. The simulation represents the fundamental basis for the development of a computer program that enables prediction of changes in a movement pattern which result from changes in magnitude and timing of muscular contraction. The applications range from the production of aesthetically interesting styles to mechanically desirable styles.

Emphasis has been placed on the strategic development of the model and the numerical solution of the differential equations of motion. Discussion of the data collection process focussed on the procedural simplifications and the associated assumptions. Although the model is restricted to planar motion, extension to three dimensional motion is feasible but not implemented.

A single male subject ran over ground while being filmed by a high speed camera. Segmental positional data were obtained by digitization of body markers, such data were manipulated to generate a set of angles consistent with the definition of the model. This involved the fabrication of data for the contralateral side of the body as this information was unavailable from film. The reference position and segment angles were smoothed and differentiated using a quintic spline routine.

The model of the human body comprised twelve linked rigid segments. Net joint moment profiles and the initial kinematic state were input to a system of ordinary differential equations. The only constraint was ground contact during stance and was represented by an analytic rolling constraint. The equations of motion were derived using a combination of Newtonian and Lagrangian dynamics and were integrated using the LSODI numerical integration routine. The recorded kinematics were reproduced by the solution of a series of boundary value problems.

The rudimentary interactive computer program developed is a research and teaching tool which provides the basis for simulation and modification of the motion of any linked rigid segment model. The computer program allows online modification of the moment profiles, and therefore the movement, during the simulation.

Acknowledgements

To my wife Connie and my daughters Adrienne and Haley. Their support, encouragement and friendship has made finishing this degree possible. To Rob Lonergan, who guided me through the first years of graduate school and introduced me to critical thinking. To Bob Herring, for his friendship and his interest in this research. To Mrs. Joan Madison, my Junior High School math teacher, for her encouragement in the development of my mathematical skills. To Arthur Chapman, for his patience and his support. To my immediate family, who have struggled through this with me. To Armin Bruderlin & Helen Woo for their invaluable assistance in generating the thesis text. To the many other people, too numerous to be individually listed, who have assisted me along the way I give my heartfelt thanks.

"It was the best of times, it was the worst of times,
it was the age of wisdom, it was the age of foolishness..."

Charles Dickens, ***"A Tale of Two Cities"***

Table of Contents

List of Figures

Chapter 1
Introduction

This research was initiated by the author's desire to improve the biomechanical analysis techniques currently used to quantify human running. In particular, this dissertation describes a mathematical simulation of running based on a mechanical model of the human body. Of underlying importance to the choice of the mathematical representation of the human body is the assumption that the principal determinants of running performance are the inertial properties of the body. This unverifiable assumption is based on the belief that since the human body is an inertial object moving in an inertial world, mechanics provides a sensible basis from which to describe human movement. This dissertation centres on the development of a mathematical simulation of a planar analogy to human running consisting of the movement of twelve interconnected rigid segments. The simulation is the basis for an interactive computer graphics programme which enables the user to modify selectively the movement technique by altering the driving forces. A rudimentary interactive programme is presented and used to examine the practicalities of this approach. This thesis suggests that the use of computer simulation need not be restricted to a group of mathematically competent researchers, but through interactive graphics can be made available to every one.

The introduction to this thesis consists of a general discussion of the various biomechanical analysis techniques commonly used to describe human movement. The development of a dynamic simulation of running is presented as an alternative to the existing approaches. Criteria for determining the appropriateness of the simulation will be presented in the conclusions.

Traditional methods of analysing gross human movement skills are based primarily on visual information. The visual assessment of performance places a tremendous burden on all but highly skilled observers as it is difficult to both assimilate all of the information and to mentally compare successive performances. The implication of this dependence on human perception is that there are relatively few people proficient at assessing the mechanics of movement. The advent of quantitative analysis techniques has increased the number of skilled observers by presenting movement characteristics in many different ways. The function of quantitative analysis technique is not the prescription of optimal performance but the presentation of information.

The dominant prerequisite of a quantitative biomechanical analysis of human movement is the definition of a model that describes the fundamental functional characteristics of the human body. This functionality is necessarily dependent on the movement being analysed. The following sections will discuss the various general analysis techniques that employ mathematical models.

1.1. Mathematical Models For the Study of Human Movement

The principal reason for the development of a mathematical representation of the dynamic characteristics of the human body is the reduction of the natural system to a general formulation. A model is a symbolic abstraction of a system. A general formulation allows one to concentrate on the mathematical task rather than the specific system. It is possible to economise time and effort in the study of movement by using mathematical descriptions of the human body which are equivalent to systems that have already been theoretically analysed. This minimises the necessity of conducting physical experiments. One can capitalise on the multitude of research results produced by other studies of natural or robotic systems. The philosophical importance of the modelling and simulation of the movement of the human body is the resulting capability of making predictions concerning novel tasks. The use of predictions may range from the identification of the principles determining why the human body behaves in a specific manner, to the modification of movement patterns.

The choice of the type of model to be used for simulation is dependent on the resolution of several issues such as, the requisite degree of complexity and the type of mathematical control implemented.

Mathematical modelling necessarily involves a simplification of the original system. A determining factor for the ultimate usefulness of a model is the initial selection of the principal relationships which are to be included. The requisite degree of complexity of the model of the human body varies in accordance with the movement intended to be simulated. Care must be taken in this selection process, as redundant complexity may mask the inherent elegance of the system, while oversimplification may result in the omission of fundamental properties. If the model is overly simplistic, it will be incapable of performing the desired movement. On the other hand, excessive complexity is equally unconstructive in that it may not produce any valuable information (e.g. modelling the hands will not provide meaningful information for the study of a running stride). Furthermore, an excessive degree of complexity makes interpretation of the model results extremely difficult because much of the information is superfluous. In light of these objections, along with the high computing costs associated with excessive model complexity, it is submitted that the complexity of a model should be restricted to those principal characteristics of the original system which affect the movement to be analysed.

The value of a model is related to its predictive capability, and this can not be determined a priori. The selection of the degree of complexity of the model is of fundamental importance to its ultimate predictive capability, but is, perhaps unfortunately, a subjective decision. The researcher is free to choose the desired level of complexity. However, it is suggested that there is an optimal degree of complexity which is dependent upon the movement to be studied. Unfortunately that optimal degree of complexity can not be specifically determined[1]. One approach to the issue of determining an appropriate degree of complexity is the utilization of one elaborate, all inclusive model (Hatze, 1983b). Such a model must necessarily be extremely complex if it is to be capable of performing all movements[2]. However, as noted above, there are a number of problems associated with extremely complex models.

Excessive model complexity may also necessitate the identification of several redundant parameters and inputs. This parameter identification problem can be a serious limitation of a complex model.

It was decided that for this simulation of human running, it would be most useful to develop a simple model. The model includes the principal mechanical relationships of running such as, the inertial properties of the limbs. This is arguably a severe limitation of this thesis (and every other biomechanical analysis as well) as there are many physiological characteristics that undoubtedly play a functional role in running. The next sections describe the methods by which mechanical models are utilised in a biomechanical analysis.

---

[1]As discussed in a later section this is due to the impossibility of validation.
[2]Hatze (1983b) uses a 17 segment hominoid with 44 degrees of freedom.

1.2. Descriptive Biomechanical Analyses of Human Movement.

A biomechanical analysis of human movement is normally defined as one of two general classifications of quantitative techniques. These two classes of analysis are commonly referred to as kinematic or kinetic. A kinematic analysis quantitatively describes the geometry of a movement pattern with respect to some variable, such as time (Ariel, 1980, Chapman & Medhurst, 1981). In this type of analysis little attempt is made to determine the cause of the movement studied. This dissertation does not focus on the many types of kinematic analyses that have been presented in the literature. Kinematics is an adequate way to describe the pattern of movement, but without a causal mechanism it offers little in the way of predictive capability. Kinematics would be more useful if the movement could be non-dimensionalised so as to eliminate the problem of similitude between people of different anatomical proportions.

A kinetic analysis assumes a causal relationship between a force producing mechanism and the observed kinematics. This type of analysis requires the derivation of equations of motion which represent a mathematical description of the model dynamics. These equations are commonly derived from Newton's law (F=ma). The derivation and formulation of the equations is discussed in greater detail in Chapter 2. There are two sub classes of a kinetic analysis of movement that will be discussed: inverse dynamics and forward dynamics.

An inverse dynamics analysis requires the specification of the kinematics. A set of force or torque profiles is then calculated from the algebraic equations. Discussion is centred on these force profiles, or on the mathematical transformations of the force profiles, such as work or power (Chapman & Caldwell, 1983a, Chapman & Caldwell, 1983b, Chapman, Lonergan & Caldwell, 1984, Chapman et al., 1985). A forward dynamics analysis requires specification of the driving forces. The resulting motion is determined by integration of a system of non-linear differential equations.

1.2.1. Inverse Dynamics

Inverse dynamics has a more traditional acceptance in the biomechanics community and will be discussed first. An obvious requirement for an inverse dynamic analysis of human movement is that the kinematics of a performance must be experimentally recorded. A typical experiment involves the collection of a time series of positional data for a specified set of body markers. Mathematical transformations then produce the necessary information for the analysis. An overview of the procedure will be discussed in this section. Emphasis will be on the experimental determination of the kinematics and the lack of predictive capability.

It is necessary that methods of data collection should affect the performance of the movement as little as possible. The most efficient non-invasive recording techniques currently in use record the movement of specific external body markers. High speed filming or real-time collection equipment, such as SELSPOT, collect data while allowing the performer to carry out the movement in familiar surroundings unencumbered by recording equipment. Unfortunately, because of the problems associated with existing digitization techniques, such as, the movement of body markers, perspective error, and other anomolies, a considerable amount of random variability is introduced by that type of recording process. Errors are also introduced to the kinematics by the inconsistency of human movement. Hatze (1986) gives an extensive overview of the causes of human motion variability. "They include initial perturbations of the skeletal, muscular, and neural systems as well as perturbations due to incremental changes, during motion execution, of external forces, muscular parameters (fatigue), afferent sensory inputs, and of the motor programmes controlling the execution of the movement". The result of this uncertainty in the data is that subtle changes in the pattern of the movement cannot be analysed. The necessity of studying distinct movement patterns, insuring differences greater than the statistical variability results in a disjointed analysis. Owing to the limitations of the accuracy of the data collection process, the researcher has a limited choice of acceptable mathematical representations of the human body. The models must necessarily be relatively simple in order that the data are reliably transformed to the model structure.

In addition to the problems created by the data collection process, there are a number of other areas in which traditional biomechanical analyses fall short of being the best possible approach to the analysis of human movement. More significantly, descriptive analyses of movement have limited predictive capabilities.

1.3. Predictive Biomechanical Analyses of Human Movement

A severe restriction of traditional descriptive analyses of human movement arises from the requirement that the movement must be performed prior to its analysis. Simulation of the movement of a human analogy would reduce the amount of experimentation required by providing predictions as to how the human body would perform under varying conditions. The researcher could determine mathematically a desired change in technique, with the benefit that the individual need not actually perform the movement. It is suggested that the ability to predict the effect of a change in performance without having to involve the subject is a useful objective.

As mentioned earlier, forward dynamics is the result of the specification of the driving forces and the mathematical determination of the movement. Once the researcher is committed to a particular model the theoretical decisions focus on the determination of a reasonable set of force inputs. The following sections discuss some of the possible approaches to identifying these forces.

Forward dynamics can be classified by the type of controls employed. In particular, whether the model is under autonomous adaptive control, or under rigid user supplied control. These specifications are not universal definitions but are convenient for this thesis. A brief review of two types of dynamic controls will be given, as the utilisation of each was considered in the formulation of the present simulation.

1.3.1. Autonomous Adaptive Control Models

The first type of dynamic model to be discussed is that which is most often applied in the field of robotics; autonomous adaptive control. One of the primary concerns in robotic locomotion is the search for a control strategy which efficiently produces a stable running or walking stride in a robot. Control theory implies that the system is internally forced to satisfy some objective criterion (e.g. running speed). Since the controls are essentially inherent to the system they are adaptive, and the system is blessed with predictive capabilities.

Despite the fact that one of the objectives in robotics research is the development of robots that are capable of walking and running, the information gained from this type of modelling may also be significant for the understanding of human motion, particularly as these models are usually autonomous. Unfortunately, there are only a few robotic developments that offer information that is of interest to the field of biomechanics. The following two sections present two of these developments.

1.3.2. Autonomous Hopping Robot

The development of an autonomous hopping robot in 1984 was a significant engineering development (Raibert, 1984, Raibert, 1986). Raibert produced a robot capable of a stable hopping motion at a user determined speed. He has suggested that the one legged hopping machine is the simplest analogy to human running. Locomotion of the robot is achieved by a combination of a stance and an airborne phase, with the distinction between hopping and running being the alternation of the support leg. It is not unreasonable to suggest that the robot controls are similar to the human neural counterpart, as the overall objective in both cases is unaided locomotion. This is not to suggest that there is either an equivalence or functional mapping between the two control systems. Theoretically, an understanding of the mechanics of hopping could give us an insight into how to approach the analysis of running. It is suggested that the relevant mechanical relationships would be similar in both cases.

The obvious strengths of Raibert's models are their simplicity and predictive capability. The user, with minimal intervention, can generate a variety of movements. The user can test the limitations of the model, modify the system parameters, and retest. In this heuristic manner, the researcher can develop an intuitive understanding of the system dynamics. A brief overview of the control strategy will be presented as an example of the simplicity and elegance of Raibert's design.

The control strategy was separated into two distinct phases: an airborne phase and a stance phase. During stance, a linear control torque was applied between the torso and the leg with the objective being to keep the torso vertical. When the spring of the stance leg was fully compressed a position actuator was turned on, which further compressed the spring. The position actuator was employed to offset the loss in momentum at impact and takeoff.

During the airborne phase, the leg was oriented so that ground contact was made at a desired angle. The choice of desired angle was based on the difference between the desired and current horizontal velocity, and the necessary position of the robot's centre of mass relative to the contact point. This last calculation required knowledge of the general behaviour of the robot during stance. The resulting autonomous robot was robust and capable of hopping over a wide range of speeds and changes in direction. The only user input required was the desired horizontal velocity.

During the research into this dissertation the author had the opportunity to reproduce the simulation of the hopping robot. The results of this research have not been included in this dissertation as its only functional significance for the thesis is the fundamental nature of the control system. In the author's research Raibert's control strategy was modified to control the movement of a hopping model consisting of a jointed leg. Subsequently, a second leg was attached to produce a running stride. It was found that, as the complexity of the model increases, the subsequent size of the control strategy must also increase, and therefore, at some level of model complexity, the control will become the dominant characteristic and the natural passive dynamics of the model are not discernible. Preliminary research indicated that a simple extension for Raibert's hopping robot did not result in movement qualitatively determined as natural for the human body.

This was the predominant reason for the current exclusion of a control strategy for this dissertation. Another less serious, disadvantage of this approach is that the use of control theory for the study of human motion often induces negative responses from many researchers. Despite the fact that no one has suggested that mathematical controls mimic the human motor control system, many criticise this supposed analogy. A reasonable alternative was to study those stable walking or running robots that do not have contrived controls. Their movement is a result of their inherent dynamics properties and the initial kinematic conditions.

1.3.3. Passive Walking Robot

Recently, McGeer has adopted a passive control approach in his research into walking and running robots (Mcgeer, 1989). His initial research was based on an analysis of a simple child's toy, which when placed at the top of a slope, "walked" down hill unaided by controlling forces (McMahon, 1984). The "walking" was a by-product of the natural dynamic characteristics of the toy. McGeer has provided a complete theoretical analysis of how and why the toy walked. He has also built an unpowered walking machine based upon the principles of the child's toy. The model of the toy was modified slightly to overcome the limitation of requiring a slope. For example, simply by shortening the swing leg before impact, and a subsequent lengthening of the leg during stance, the robot achieved a stable walking cycle on a horizontal surface, (although it is limited in its ability to walk uphill). Unlike most other walking robots in which the control strategy is the dominant characteristic of the system and the dynamic characteristics are natural obstacles, the dominant characteristics of McGeer's walking machine were the natural dynamics, making it a better analogy for the study of human motion.

McGeer hopes to discover the theoretical underpinnings of a passively stable running cycle. He would then introduce a minimal control strategy into his model, which would incorporate the natural dynamic characteristics, and by these means develop a running robot. It is submitted that this approach to the development of a robot that will be capable of running is preferable to the approach taken by Raibert, because the natural dynamics are retained in the passive control model.

The possible relevance of this passive control model research to the understanding of human running is particularly interesting. It is conceivable that an individual's preferred running stride is one which requires minimal control. Considerable biomechanical research has focussed on developing criteria for the identification of an efficient running stride (Chapman et al., 1985, Williams & Cavanaugh, 1983, Lonergan, 1988). Lonergan has demonstrated the difficulty of identifying an appropriate criterion cost function employed by the human motor control system. Hypothetically, the passive running cycle may provide a unique approach to this issue. It may be postulated that the preferred, or most efficient running style is that which is closest to passive running, and therefore, that which requires the fewest controlling inputs from the muscular system.

In summary, the analysis of stability and reproducibility are fundamental concepts for robotics. With respect to human movement, it is obviously true that both the system and the movement are stable, and that these are not fundamental issues for the modelling and simulation of human movement. The goal of simulating human movement as emphasised in this thesis, stretches beyond these studies of stability and reproducibility. We are interested in the gross properties of human movement directly observed, and are not in a search for motor control theories. One of the objectives of the simulation of human movement is to lead to an understanding of the modification of movement technique, and the development of many possible ways of performing the same task.

1.3.4. Rigid User Supplied Control Models

The second type of system dynamics models that were examined in the development of the author's simulation of human running, were those models that are controlled by rigid user supplied inputs. These models are not adaptively controlled and the user is required to specify all of the controlling forces. In effect, the simulation user is a replacement for the internal control system described in the previous section. The primary objective in the development of these models was to provide a better means of modifying human movement.

A number of physiological models of the human body have been developed. Some of these models are extremely complex, and claims have been made that they describe the neurophysiological system as well as the musculoskeletal system (Hatze, 1983b, Bauer, 1983). In spite of claims that the development of physiological analogues is the only reasonable direction for the biomechanical study of human motion to proceed, it is suggested here that such an argument is not valid. As indicated previously, there are a number of problems associated with complex models of the human body. In terms of controls for the model, parameter estimation in a complex model becomes unreliable because of the limitations in experimental techniques. Physiological analogue models may become so elaborate that the parameter estimation becomes in fact the major objective. These elaborate models can only be reasonably tested with such simple movements that nothing valuable is gained from the analysis. For example, the kicking simulation presented by Hatze (1984), does not provide any practical information other than a review of the mathematical task.

Another problem associated with current physiological models of the human body is that their performance is dependent upon several assumptions that are not directly justifiable from experimental results. Rather than be limited by these experimental difficulties, the limitations imposed by a less complex model were accepted in this research on the grounds that it would be better to have a simple model that provides useful information, rather than a complex model which might be incapable of offering any additional insight.

Following the development of a reasonable model, the movement of a human analogy can be simulated. Several researchers are currently working in the area of human movement simulation. Their work can be divided into two general areas: the first is the modelling of human performance, and the second is the modelling of the system dynamics of a movement.

1.4. Simulation of Human Performance

The majority of research reports on human movement simulations can be assigned to a group within which the major emphasis is on modelling human performance (Hill, 1927, Keller, 1974, Vaughan, 1983a, Ward-Smith, 1985a). These are not simulations of the mechanics of movement. The models presented in these publications emphasize human performance characteristics such as, speed, distance, and physiological capacity. In some papers, the author's have claimed to have made a mechanical analysis of the movement. For example, Vaughan (1983a, 1983b) modelled the body as a point mass propelled forward by a horizontal force. The horizontal force consisted of two components, a term representing the horizontal ground reaction force and a term representing wind resistance. The equation parameters were statistically determined from performance times for various running distances. Predictions of running time were made from these differential equations. However, little or no emphasis was placed on mechanical characteristics. Only the movement of the centre of mass was analyzed.

1.5. Dynamic Simulations of Human Movement

Several researchers have chosen to simulate the movement of the human body. Much of the early work in this area presented constrained simulations in which only a few of the mechanical degrees of freedom were considered variable. Studies of jumping, swimming, diving and pole vaulting were all documented in the literature in the 1970's (Passerello & Huston, 1971, Chao & Rim, 1973, Gallenstein & Huston, 1973, Ramey, 1973, Boysen, Francis & Thomas, 1977, Walker, 1973). These simulations can be separated into, firstly, those which represented the torque produced by muscular contraction, either as net joint torques or via muscle models, and secondly, those which kinematically described the relative movement of the individual segments. A similarity in all of this work is that the simulations were not dynamically autonomous. This does not imply that they required an external force, but rather that they required some predetermined kinematics, and that therefore, the movement was not entirely a dynamic simulation.

In the following two sections the concept of a constrained simulation will be expanded. Constrained simulations of human movement have been presented as either kinematically or dynamically constrained simulations.

1.5.1. Kinematically Constrained Simulations

Kinematically constrained simulations will be discussed first, as they are more numerous in the literature. The equations of motion in these studies were written to describe the movement of a multisegment model. Many of these models restricted the simulation to the airborne phase and the motion of many of the degrees of freedom were specified. Only a few of the variables were actually simulated. Commonly, the orientation of the torso was determined by the solution of the equations of motion. In many of the airborne simulations the equations were derived based on the conservation of angular momentum. As this represents only a single equation in planar motion and three equations in three dimensional motion, only a limited number of degrees of freedom could be considered to be variable. By restricting the unknowns it is easier to make predictions from these models since there is less information to assimilate, and it is therefore, computationally less expensive. For example, the mathematical model of pole vaulting by Walker (1973) predicted that trained athletes would be capable of pole vaulting over twenty feet, long before any one else considered it to be possible.

The diving simulations produced by Yeadon (1986) present the most significant predictive capability currently published in biomechanics literature. In spite of the fact that only three variables were actually simulated, Yeadon has introduced a new approach to the performance of twisting dives. Of particular significance is the fact that the predictions based on his simulations contradicted the prevailing views of the coaching community, and proved to be correct. It was commonly accepted in the diving community that in order to maximise the rotation of twist in a dive, the diver must reduce the moment of inertia about the longitudinal axis immediately after takeoff. The gist of the theory was that minimising the moment of inertia would maximise the angular velocity and consequently, the amount of rotation. Yeadon has shown that the twist is accompanied with rotation about another of the principle axis of rotation and that it is beneficial to delay the reduction of the moment of inertia. When the predictions based on Yeadon's simulations were put into practice, it was discovered that the rotation of the dive was in fact increased. More recently, external forces and torques, and environmental constraints have been added to this type of simulation (Hahn, 1988). This modification has resulted in greater flexibility and the capability of simulation more than just airborne movements.

A programme has recently been developed (Isaacs & Cohen, 1988) which introduces complex kinematic constraints into the dynamic simulation. The programme also includes the addition of inverse kinematics. One of the difficulties with this approach is that a poor definition of the constraints can produce an overdetermined solution. The benefit of such an approach is that the researcher is able to minimize the effective complexity of the model.

1.5.2. Dynamically Constrained Simulations

Published simulations of human movement in which torque inputs were implemented are few in number and unsophisticated. For example, an early running simulation (Chow & Jacobsen, 1971) required specification of both the hip movement and the ground reaction forces. Therefore, that simulation was not autonomous and was capable of making few predictions, as any change in system input would ultimately effect the specified variables.

The motion of the recovery leg in running was simulated using a two segment leg, in an attempt to quantify the non muscular reactions between the adjacent segments (Phillips, Roberts, Huang, 1983). The movement was simulated without reference to the knee torque and the results were compared with the human performance. The apparent differences in motion were attributed to muscle action. This explanation is misleading because muscular forces modify reactions between adjacent segments in an interactive way, suggesting that there is not the direct relationship that Phillips has proposed.

1.5.3. Optimization

Another method for constraining a simulation so that the control problem is minimised, is to require the simulation to satisfy some objective criterion such as minimisation of energy. In effect producing a local optimal solution.

It has been suggested in the literature (Hatze, 1984) that the primary impact of the successful simulation of human movement will lie in its utilisation as a tool for the optimisation of human movement. It has even been suggested that, at some point, the optimisation of human performance will be produced by the mathematical optimisation of the movement of a model (Hatze, 1984, Vaughan, 1984). Within this suggestion lies the implicit assumption that the mathematical model can be functionally representative of the human body.

Mathematical optimisation of a model is restricted to minimising a single objective function (e.g. the minimal mechanical energy required) within the limits of physical constraints. However, it is submitted that the human body does not similarly optimise over a single variable. The human body necessarily optimises over several variables, and this involves some intermediate combination of objective functions (Nelson, 1983). It may be reasonable to develop an objective function that consists of a weighted average of several objective criteria. The identification of such a function that reasonably explains the natural motion seems implausible to this author. Therefore, one cannot assume that the optimisation of the movements of a model will have the direct effect of optimising the movement of the subject.

Despite the fact that mathematical optimisation of the movement of a model cannot directly lead to the optimisation of human movement, it may nevertheless provide relevant information. For example, in one study (Nelson, 1983), a simple movement of the model was optimised for several objective functions and the resulting movements were plotted. It was submitted that the plots formed an envelope, and that the optimal human movement lay within it. The tighter the envelopes, the greater were the chances of predicting the best movement of the athlete.

Marshall, Wood & Jennings (1985b) examined the efficiency of seven optimisation criteria to predict the kinematics of walking. Differences were noted in the ability of the objective functions to predict stance leg or swing leg kinematics, which suggested to the authors the possibility that there may be more than one performance objective in a system at one time.

For simple movements, such as straight kicking, optimisation of the two segment model is considered to provide a reasonable prediction of the optimal human performance (Hatze, 1984). However, this is not the case for more complex human movements.

Mathematical optimisation of the movement of a model may provide a useful method for identifying the criteria determining how the human body naturally chooses to perform a given movement. Future development of this thesis will undoubtedly consider the use of optimisation as an aid in developing strategies for modifying technique.

1.5.4. Interactive Simulations

In a recent review article on the use of simulations in biomechanics, Vaughan (1984) suggested that "a possible limitation of computer simulation is that an advanced knowledge of mathematics and computers may be necessary. Indeed one of the greatest dangers is using the computer model as a "black box" without understanding its complexities, limitations or validity". This statement is typical of the view of much of the community of researchers in biomechanics who currently use simulation in their analysis of human movement. Hatze (1983b) has suggested that it is necessary to have an understanding of physiology, mathematics and physics before one can tackle the problem of simulating the movement of the human body. This elitist attitude has resulted in the development of models which are both difficult to understand and to use.

An attempt is made in this thesis to provide a simulation of human running that can be used as a research tool by both professional and lay individuals. The elements of the model are simple, with simple relationships of interaction. The user does not need to have direct access to the source code as the programme is interactive. Marshall, Wood & Jennings (1985a) have developed an interactive simulation of simple human movements, but the interactive aspect requires the user to edit and modify the torque profiles. It is suggested that a truly interactive model must provide the user with the ability to interrupt the simulation, modify the profiles and continue the simulation. The interactive programme Virya (Wilhelms, 1987) and more recently Kaya (Wilhelms, 1988) are examples of elaborate interactive computer programmes for the animation of human movement. In terms of the simulation of human movement, perhaps future programmes will allow the user to modify the force profiles without stopping the programme. Preliminary investigations, using the programme developed for this thesis, indicates that the movement is too quick for this to be a practical approach. The user would simply be able to modify the forces in advance of the simulation reaching a given point in time.

One of the concerns about an interactive simulation that involves several variables, is that there must be some movement to modify. The task of producing a human movement in its entirety is formidable (Armstrong, Green & Lake, 1987). Therefore, rather than leaving the whole task to the user of the interactive programme, several general movement patterns should be made available. It is suggested that requiring the user to make minor modifications of existing movements is a more reasonable task than having him/her generate novel tasks. It was therefore necessary to develop a procedure for reproducing a known movement. The obvious solution was to perform an inverse dynamic analysis on a recorded movement (Isaacs & Cohen, 1988) and then to use the calculated forces as inputs to the simulation. This method is commonly used to validate a simulation (Ju & Mansour, 1988), although it is known a priori, that the original movement will not be replicated. This is evident, because the inverse dynamic analysis calculates instantaneous forces, whereas the simulation uses average force over an iteration step. To overcome this difficulty, a boundary value problem was set up between successive states. The required forces were then calculated using this approach. The boundary value problem involved a shooting method in which the objective was to match the velocity state at the endpoint. Originally the problem was designed to match the position state (Chao & Rim, 1973), but unusually large fluctuations in forces over time became apparent. Matching velocity states produced a reasonable reproduction.

In summary, this dissertation presents the development of a dynamic simulation of human running. All relevant aspects of the simulation will be reproduced and discussed. Of particular significance is the autonomy of the simulation from external constraints. The simulation requires the input of generalized forces but is otherwise independent of the environment. The forces can either be derived by requiring the reproduction of a set of kinematics or by interactive modification. This autonomy suggests that it is practical to discuss the presentation of this simulation in an interactive computer graphics programme. A rudimentary interactive simulation has been developed as a basis for assessing the practicality of the approach. The purpose of the interactive programme is to provide the user with the benefits of dynamic simulation while insulating him/her from the mathematical detail. The simulation is intended as a teaching and research tool.

Chapter 2
Methods

This chapter has been divided into three distinct sections. The first section is devoted to an overview of the mechanics and the integration of the equations of motion. The second section describes the data collection process and focuses on the fundamental assumptions required for the specification of the model kinematics. The third section describes the interactive graphics programme.

2.1. The Mechanics

The model of the human body comprised twelve linked rigid segments. Net joint torque profiles and the initial kinematic state were input to a system of ordinary differential equations. The only external constraint was ground contact during stance and was represented by an analytic rolling constraint. This section expands on the mathematics involved in the development of the running simulation.

2.1.1. Derivation of the Equations of Motion

One essential component of the modeling process is the development of differential equations representing the system dynamics. Any one of the following five general methods, or several others not listed, can be used to derive the necessary equations of motion.

1. Newtonian Mechanics (Newton-Euler Formalism)
   (Marshall, Jenson & Woods, 1985a)
2. Lagrangian Dynamics
   (Bourassa & Morel, 1983, Chao & Rim 1973, Hatze, 1981)
3. Hamilton's Principle
   (Dapena, 1981, Yeadon, 1986, Passerello & Huston, 1971)
4. Kane's Method
   (Kane & Levinson, 1983, Ju & Mansour, 1988)

5. Gibbs-Appell Method

(Wilhelms, 1987)

The choice of method is of minor importance, as algebraic manipulation can reduce the formulations to identical equations. This is a natural consequence of all derivations being ultimately reduced to Newton's law (F=ma). It is acknowledged that some researchers prefer not to use algebraic manipulation on the derived equations (Hollerbach, 1980) and thus feel it is necessary to use a mathematical derivation which directly results in the form of the equations they desire. It is suggested that emphasis on the best method of derivation is unnecessary and the method chosen should be the one in which the researcher feels the most proficient. It is only the final form of the equations that is important, not the method of derivation. For much of this work, the equations of motion were derived using the Lagrange formalism. This method was chosen, firstly, because of the author's personal preference for algebraic manipulation over vector calculus, secondly, because the resulting equations are of minimal rank and the inertia matrix is symmetric and positive definite (this produces a desirable form for the numerical integrator) and thirdly there arises only one equation of motion for each degree of freedom of the model. The Lagrange method requires that the internal and external constraints of the model be holonomic. If non-holonomic constraints are necessary for a particular purpose (e.g. friction), the number of degrees of freedom must be increased to the extent that the resulting equations of motion implicitly include the constraint[1]. It is acknowledged, that under certain conditions it may be prudent to adopt another method. For example, impact conditions in this thesis are derived assuming an inelastic collision with no slipping and uses the principle of conservation of angular momentum about the instantaneous contact point. The rolling constraint is derived using Newtonian mechanics[2]. The equations for the twelve-segment model were derived using a combination of Newtonian and Lagrangian dynamics. The dynamics of the foot were described using the Newtonian method and the dynamics of the rest of the body were derived using the Lagrangian method. The two models were coupled through the reaction force at the ankle joint.

The construct of the final model was the end result of the development of a series of models. The development started from the analysis of a simple one segment model. The

---

[1]For example, through the introduction of Lagrange Multipliers.

[2]The foot is modeled as a finite radius arc and the ground constraint as a rolling constraint.

purpose of the first model was to test the mathematical methods. Subsequent models involved a successive increase in complexity. Each stage was an attempt to determine the construct of the succeeding model as well as producing a better model for the purposes of the simulation of running. All of the models were constructed using the following assumptions:

1. All models comprised a series of planar interconnected rigid segments.
2. The segments had constant mass and constant length.
3. The inertia tensor was assumed to be in dyadic form
   (e.g. the reference axes were considered principal axes).

The segment lengths were determined from the film data. The other anthropometric values (inertia, mass, distance to centre of mass) were obtained from the literature (Winter, 1979). The joints were holonomic constraints and were the points of articulation between segments. Coordinate axis for each segment were chosen at the proximal end of the segment. The segments were numbered so that joint *i* connects segments *i-1* and *i*. All joints were rotational and the joint variables q_i were referenced either to the vertical up (as shown in Figure 2-1) or to the vertical down.

The reference vector to the body $\bar{R}$ is dependent upon the choice of model. It was normally located at a mathematically convenient point (e.g. the point of contact of an external constraint). The reference vectors $\bar{r}_i$ extend from the origin of an external reference system to the centre of mass of segment i and were expressed as:

$$\bar{r}_i = \bar{q}_{i-1} + r_i\,(-\sin q_i, \cos q_i)^T$$

where:

$\bar{q}_{i-1}$ = Position vector at the distal end of segment i-1

r_i = Proximal distance between the joint and the centre of mass of segment i

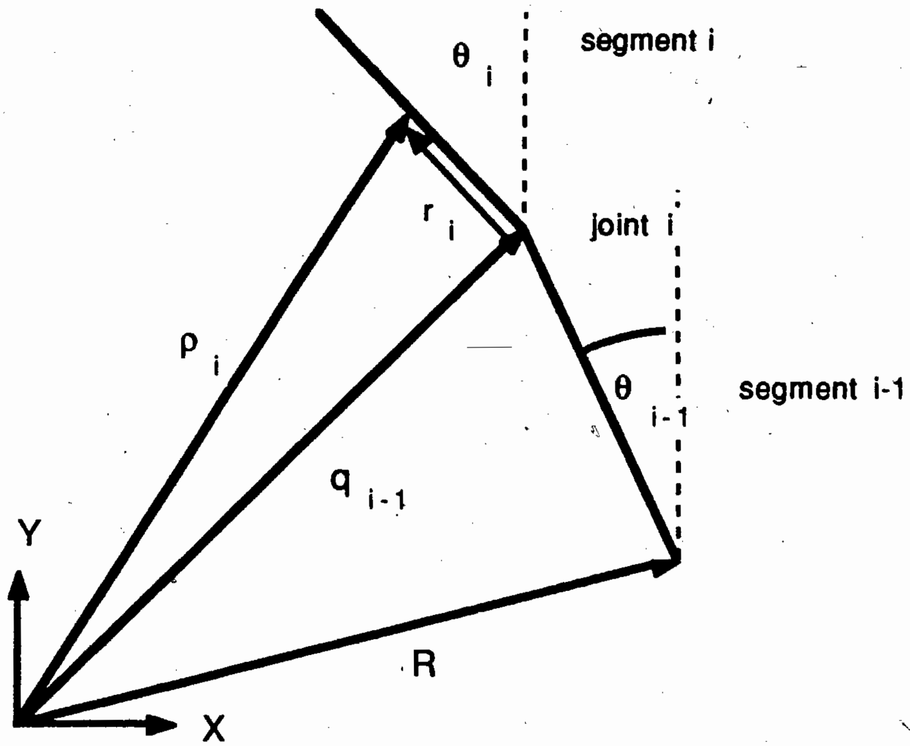


Figure 2-1 Specification of the variables and reference orientation for two segments of a planar model.

The angular velocity of segment *i* is expressed as:

$$\bar{w}_i = \dot{q}_i \, \bar{\mathbf{k}}$$

The Lagrangian L is defined by :

$$L = \sum_{i=1}^{n} \left(\frac{1}{2} m_i \, \dot{\bar{r}}_i \cdot \dot{\bar{r}}_i + \frac{1}{2} I_i \, \dot{q}^2_i - m_i \, g \, \bar{r}_i \cdot \bar{\mathbf{j}}\right)$$

where :

$\bar{\mathbf{j}}$	= Unit vector in the vertical direction
m_i	= Mass of segment i
I_i	= Moment of Inertia of segment i

The Lagrange equation can be written:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_r} - \frac{\partial L}{\partial q_r} = F_{q_r} \qquad r = 1, 2, \ldots, n$$

where :

L	= Lagrangian = T-V
T	= Kinetic Energy

V	= Potential Energy
q_r	= Generalized Coordinate
n	= Number of Degrees of Freedom
F_{q_r}	= The Applied Generalized Forces

The differentiation can be carried out to yield a set of non-linear second order ordinary differential equations:

$$\frac{\partial^2 L}{\partial \dot{q}_i \partial \dot{q}_j} \ddot{q}_i + \frac{\partial^2 L}{\partial q \partial \dot{q}_j} \dot{q}_i - \frac{\partial L}{\partial q_j} = F_{q_j} \qquad i,j = 1, 2, ..., n$$

This equation can be written in the form:

$$A(q)\,\ddot{q} = B(q, \dot{q}) + F \quad i,j = 1, 2, ..., n$$

where :

A	= The Inertia Matrix
B	= The Vector of Transient Terms
F	= The Applied Generalized Forces

The task of differentiation can be quite tedious, but is not a limitation, as suggested in the literature (Pandy & Berme, 1988a). The equations of motion are symmetric and a recursive algorithm is tractable. An explicit declaration of the branching that would enable a computer generation of the equations of motion for the twelve segment model used in this thesis was not designed. An example of a recursive algorithmic formulation for a single kinematic chain is presented in Appendix F. It is also possible to use a simulation language (MACSYMA, 1977) to perform the differentiation analytically (Ju & Mansour, 1988).

With the exception of the simplest second order system: $\ddot{q} + w_0^2\, q = 0$ where a numerical second order integration method is possible (e.g. Numerov's method (Gladwell & Thomas, 1981)), it is usually necessary to transform the system of second order equations into a system of first order equations. This can be easily done as follows (Burden, Faires & Reynolds, 1981):

$$y_{2i-1}(t) = q_i(t)$$

$$y_{2i}(t) = \dot{y}_{2i-1}(t) = \dot{q}_i(t)$$

The equation can then be written in the form:

$$A(y)\,\dot{y} = B(y) + F$$

2.1.2. Introduction of an External Constraint

One of the natural consequences of simulating running is the necessity to incorporate the ground constraint. The contraction of the human muscles causes only a reorientation of the segments relative to each other, and in fact, it is only the interaction between the foot and the ground which propels the runner forward. Three methods of imposing the ground constraint have been considered. The first, and technically the simplest method of applying the constraint would have been to introduce a massless spring and damper at the point of contact between the foot and the ground. This method was rejected because it introduced artificial eigenvalues[1] and was computationally expensive.

The second method considered was to determine mathematically the force required to maintain the constraint (Hatze & Venter, 1981). This method would require the introduction of a new set of equations, but would not introduce artificial eigenvalues. This method would be particularly useful when several possible constraint violations exist both independently and simultaneously. For example, tripping the runner and having the runner land flat on the ground and start rolling. A similar method of constraining movement was used for the first of the two twelve-segment running simulations in this thesis. In this thesis the constraint forces were determined by introducing a boundary value problem which specified that the point of contact between the foot and the ground remain stationary.

The third method was to redefine the model such that the constraint would be incorporated into the equations of motion, and the number of degrees of freedom reduced. This involved separation of the movement into two distinct phases (e.g. airborne and stance). The ground constraint forces are inherent to the equations describing the stance phase. They are neither necessary nor calculated. This is an elegant approach for the simulation of movements with only a few independent states, but is algorithmically intractable when there are numerous constraints. Fortunately, the analysis of running need only consider the ground constraint, and in particular, only contact with one foot. This method required the researcher to indicate explicitly the impact and take-off conditions. An elaborate mathematical derivation of the impact conditions has been presented in the

[1] See the next section for an explanation of the importance of the eigenvalues.

literature (Zheng & Hemami, 1984, Moore & Wilhelms, 1988), but this complexity is unnecessary here, as only a single point of constraint and only one constraint at a time were imposed. The impact conditions were calculated using the fact that the angular momentum of the body about the constraint point is conserved on impact.

For planer motion the angular momentum about a unique point O can be expressed as:

$$\bar{H}_0 = \sum_{i=1}^{n} m_i k_i^2 w_i^2 + \bar{r}_i \times m_i \bar{v}_i$$

where :

$\bar{r}_i$ = A Vector for 0 to the Centre of Mass of Segment i

$\bar{v}_i$ = The velocity of the Centre of Mass of Segment i

w_i = The Angular Velocity of Segment i

k_i = The Radius of Gyration of Segment i

At Impact :

$$\bar{H}_0^+ = \bar{H}_0^-$$

A significant difficulty was presented by this type of representation of the constraint. During natural running the position of the force vector on the sole of the foot changes continuously from the time of foot strike to toe off. Mathematically constraining a single point may be inappropriate. Three methods of dealing with this problem were considered. The first method was to ignore the movement of the force vector, and assume a fixed constraint position. This is often done when the leg is represented as a single extensible segment (Pandy & Berme, 1988a, Pandy & Berme, 1988b). This method was used in hopping simulations, but was considered to be unnatural for the running simulations as there is a significant displacement of the force vector.

The second method was to represent the foot with a more elaborate model and introduce more than one point of constraint, (e.g. a combination of springs and dampers and fixed points). Bourassa & Morel (1983) simulated the ground constraint with three separate constraint points. The process they used for changing the constraint from heel to metatarsal to toe was not described. This method was studied, for this research, for two constraint points, both of which can be active at the same time, but was discontinued in favour of the third method, as follows.

The third method was to represent the plantar surface of the foot by an algebraic expression. Ju & Mansour (1988) represented the foot by a second order polynomial and McGeer (1989) represented the foot as a finite radius arc. In this thesis the foot was also modelled as a finite radius arc. This allows the force vector to move naturally with the movements of the foot via a rolling constraint. The equation of motion describing the model of Figure 2-2 is derived from Newton's laws (see Appendix G):

$$t = \frac{d\bar{H}_0}{dt} = \frac{d}{dt}(m\, k^2 w + \bar{r} \times m\bar{v})$$

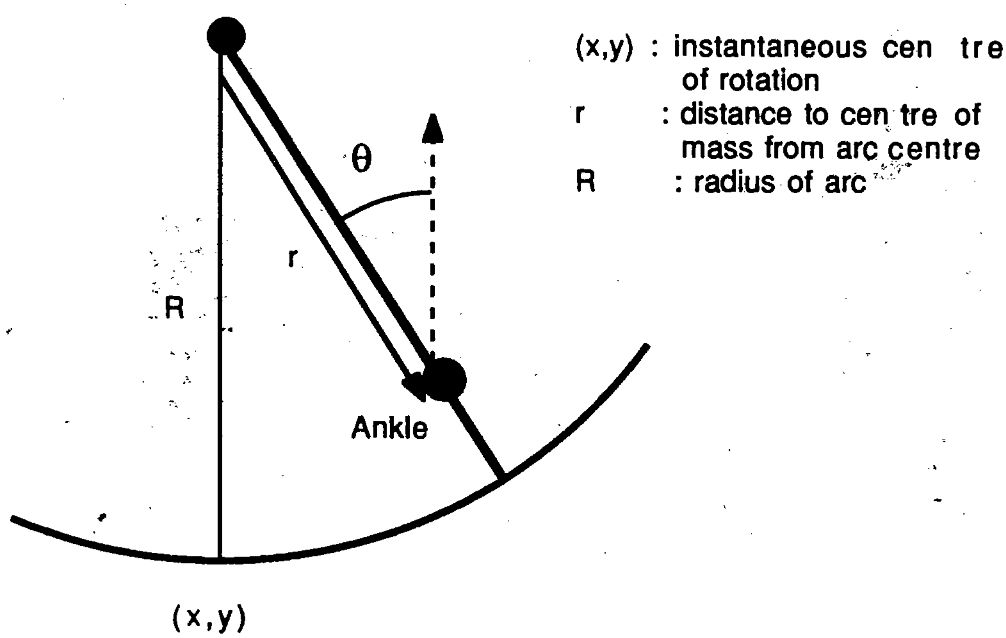


Figure 2-2 A description of the foot model.

The equation reduces to :

$$t = m\,(k^2 + R^2 + r^2 - 2rR\cos q)\ddot{q} + mgr\sin q$$

where:

R = The radius of the arc

r = The distance from the centre of the arc to the ankle

The movement of the constraint point instantaneous centre of rotation is described by:

$$x = x_c + R(q - q_c)$$

A foot comprising a fixed radius arc was employed in this simulation as it was considered to be more general, and easier to modify. Parameters were estimated by heuristically matching the movement of the ankle position of the model, to the recorded movement of the ankle. This was a relatively simple procedure, and is explained in greater detail in Chapter Three and Appendix G.

2.1.3 Integration of the Equations of Motion

The movement of the model is simulated by integrating a set of ordinary differential equations given the initial conditions and controlling forces. With the exception of simple physical models or linearised systems of equations, an analytical solution is unlikely.

Two distinct approaches to the solution of the equations of motion have been considered. One method is to consider a linearised form of the equations at a given state, and to use a standard technique such as Laplace transforms to derive a local solution. The second method involves the numerical integration of the differential equations and requires the selection of an integrator from a myriad of available numerical techniques. Both of these methods require knowledge of the system eigenvalues.

The eigenvectors represent the "normal modes" of the system, and in combination with the eigenvalues define the solution of the system of differential equations. The equations of motion for all of the models discussed in this thesis are non-linear and consequently, by definition, the eigenvalues are dependent on the instantaneous state of the system. Linearisation of the equations is reasonable only if the eigenvalues vary little over the range of states for a particular movement.

Standard numerical techniques can be used to calculate the eigenvalues for the system linearised about specific states[1]. The system of equations can be reduced to the form:

$$A\,y = l\,J\,y$$

where:

A = The Inertia Matrix

J = Jacobian of the Transient Terms

[1] IMSL routine EIGZF was used for this thesis.

y = The State Vector
l = The Eigenvalue Vector

The eigenvalues have been calculated for various states of the different models. With respect to all of the models studied in this thesis, the eigenvalues can range from a purely imaginary term to a purely real term. The non-linearity of the eigenvalues increases with the addition of distal segments. It is important to note however, that for a specific simulation, the eigenvalues may never traverse this extreme range, but it is suggested that the variability is sufficiently large that linearisation of the equations of motion for the simulation of running was inappropriate.

There has been a significant amount of research examining the integration of linear systems of differential equations. It has been shown that the solution of a linear system requires that the eigenvalues lie within the stability region of the numerical method chosen for integration (Numerical Analysis Course Notes, Payne, 1985). It is assumed that the same stability theory applies to nonlinear equations and therefore the eigenvalues, as linearised about a set of conditions, lie within the stability region of the numerical integration method over a reasonable step size.

With respect to the system eigenvalues, the two worst case scenarios are stiff problems and oscillatory problems. A stiff problem can be considered to be one in which one or more of the system eigenvalues are very large relative to the other eigenvalues. Integration with an inappropriate technique may result in a meaningless solution. An oscillatory problem is one in which the eigenvalues lie on the imaginary axis (for example an undamped pendulum). Integration with an inappropriate technique results in either an artificial increase in energy or an artificial decrease in energy. Both of these scenarios are possible conditions for linked rigid segment models (See Appendix A).

In Biomechanics, researchers often use an explicit Runge_Kutta integration method which behaves poorly under both of the above conditions. Evidence of numerical instability for the simulation of jumping is documented by Marshall, Jensen & Woods (1985a). In that simulation, the authors had to adopt a piecewise integration of a simulation of a vertical jump. The jumping simulation was performed in groups of five steps and the average output from the last two steps was used as input for the first step of the succeeding group. It is suggested that this technique is undesirable, and that a capable integration scheme should be employed instead of this heuristic approach. However, since the existing simulations involve so few iterations the numerical problems may be minimal. The LSODI subroutine (Hindmarsh, 1980) was used for the running simulations in this paper. It provides a series of backward differentiation formulas which are appropriate for stiff problems. The integrator can be instructed to use an A-stable method to deal with oscillatory problems. The LSODI integrator contains a convergence monitor that is often useful for detecting mathematical inconsistencies in the equations or divergent solutions. Lastly, the LSODI integrator internally modifies the integration step size and the order of the method to maximise the mathematical efficiency and therefore minimise the computational cost.

Having chosen a robust numerical integrator one can be satisfied that the simulation of the movement of a physical model will be adequate. However, one cannot ignore the possibility that under certain dynamic conditions the system will be unstable. In conclusion, no integrator is capable of handling all possible situations. The user should learn to recognise poor conditions and monitor the system eigenvalues when in doubt. It is essential that the researcher use caution when incorporating artificial eigenvalues via some control strategy or constraint application.

2.1.4. The Boundary Value Problem

The reproduction of the recorded performance was achieved by introducing the following boundary value problem.

$$A(y)\,\ddot{y} = B(y, \dot{y}) + F$$

$$\text{given} : y(0),\ \dot{y}(0),\ \dot{y}(t)$$

$$\text{unknown} : F$$

Note, that the final endpoint condition requires only the specification of velocity and not position as proposed by Chao & Rim (1973). The original formulation used final end position, but the estimated force profiles showed unreasonably high fluctuations with time. Matching the velocity profiles produced a reasonable match in position profiles as well as reasonable force profiles. It is suggested that the use of the boundary value problem is essentially an inverse dynamic analysis. The major differences between the two being that inverse dynamics calculates instantaneous forces and the boundary value problem calculates average forces over a time step.

The basic approach taken to solving the boundary value problem was to employ a shooting method. This technique involves, first, estimating the required forces, and then integrating the equations. If the final conditions are satisfied the task is finished. Otherwise, the forces must be modified and the integration is performed again.

Two methods were used to modify the forces . The first method is a quasi-Newtonian method (Burden, Faires & Reynolds, 1981)(see Appendix B)[1]. This method shows a fast convergence, but is extremely sensitive to the starting conditions. It is necessary for the initial guess to be very close to the correct solution. In fact, the stability region is too small for this method to be of practical use on its own. Therefore, it was necessary to choose a second method, which modified the forces until they were within the stability region of the Quasi-Newton method. The method chosen was similar to a bisection method. Forces were modified by an amount relative to the disagreement between desired final state and achieved final state. This method is very slow to converge, but is very stable. The two methods were used in combination, and the relative stability region of the Quasi-Newton method was heuristically determined, and varied with the conditions. An interesting restriction of this method was imposed by the nature of the numerical integrator. Since the integrator internally determined the method order and the step size, it has happened that the solution was on the borderline of the integrator switching. Consequently, the predetermined error bound was never satisfied. The author subjectively interrupted the infinite loop created, and gave up hope of matching that rare condition. The problem was always associated with very large force values. Subtle modification of the kinematics was found to be a practical solution. Although it is possible to input random force values to the boundary value problem, it is computationally more efficient to use the output forces from an inverse dynamic analysis.

[1]The quasi-Newton method employed was a modification of the classic procedure. Traditionally the approximation to the jacobian matrix was determined by repeating the integration with changes in parameter of the order of $\varepsilon = 0.001$ but it was discovered that the convergence region was much larger if ε was made much greater e.g. $\varepsilon = 10.0$.

2.1.5. Validation

It is suggested that validation of the model is a two step procedure. The first stage is the validation of the mechanical model and the second stage is the validation of the movement of the model as an analogue to the natural movement.

The mechanical model can be conditionally validated by verifying conservative movements (e.g. conservation of angular momentum in free flight) and limited cases of non-conservative movements. These tests are used to study the mathematical consistency of the dynamics. The LSODI integration subroutine used in this thesis includes considerable convergence monitoring. A divergent solution for these models is normally an indication of a mathematical inconsistency in the equations. This is a significant advantage of the LSODI solver over other numerical integrators. An experimental method of validation of the mechanical model could be provided by the development of a physical model. However, this technique is reasonable only for the simplest of models.

Validation of the movement of the model as an analogue to the natural movement is a complex problem (Panjabi, 1979). Assessment of the degree of verisimilitude is subjective and therefore open to debate. The model is necessarily a simplification of the natural system and therefore is limited in scope. The model is qualitatively validated by human assessment. Future work will attempt a more rigorous validation by examining the model's predictive capabilities. It is acknowledged that the model can never be absolutely validated.

2.2. The Data Collection Process

Having developed a method for mathematically describing the movement of a linked rigid segment planar model, the significant task of verifying a human analogy began. The initial objective was to obtain data from high speed cine analysis. The simulation task was then to adequately represent this motion. The use of the boundary value problem to reproduce kinematics suggests that any continuous set of kinematics is reproducible. One of the primary tasks of the research is therefore to generate the kinematics.

It is important to recognise that human running is three dimensional, and a result of the movement of a deformable body composed of hundreds of elements, none of which are connected by holonomic constraints. It is therefore unreasonable to assume that the recorded human performance will be uniquely translated to our 12 segment rigid body model. The data require interpretation before they can be used. For example, in planar recording the segments appear to change length due to perspective changes.

This thesis focussed on the simulation of running and not a replication of the human performance. A simple method for producing movement was considered desirable and sufficient since future work will be directed at modification of the simulated performance and only indirectly the modification of the human performance. It was suggested that a two dimensional high speed film recording would be sufficient to provide the required data as the analysis of a three dimensional film recording is an unnecessarily complex task.

A single male subject ran overground across a force plate while being filmed by a camera (sampling rate for both instruments = 180 Hz). The line of sight of the camera was perpendicular to the sagittal plane. The subject performed runs with his own preferred comfortable style. The film speed of 180 Hz was chosen to allow accurate identification of cycle events (e.g. foot strike), although digitisation of body markers was performed on every third frame (time interval 0.01676 seconds). The positional data were used to generate a set of angles consistent with the definition of the model, as well as a predetermined reference marker. The recorded positional data were used to estimate segment angles and not joint position. The raw kinematics, therefore did not results in a changing of the segment length as would be seen if one used the joint positional data. The reference markers and angles, not joint position, were smoothed using a quintic spline smoothing routine. One of the problems associated with using the quintic spline is that the smoothed data only poorly represent the original signal at the start and end frames. To overcome this difficulty, artificial data were created at both ends of the data array. This was done by assuming that the running stride was cyclic and producing a part of the stride at both the beginning and end of the digitised cycle. Having done this it became apparent that the additional data were slightly discontinuous with the original data. The new data were then subjectively modified so that the profiles were smooth. This technique was used only to overcome the problem of the quintic spline smoothing. Once the data were smoothed these extra data were rejected. The amount of smoothing required is determined by explicitly stating a smoothing factor. This factor reflects the root mean square difference between the smoothed signal and the original signal. This factor is not normalised and so it is inappropriate to predetermine the value of the smoothing factor. The choice of smoothing factors was subjective. Sufficient smoothing was achieved when the resulting acceleration profiles were both smooth and subjectively reasonable for the smallest value of smoothing factor.

It was necessary to reproduce the kinematics of the contralateral side of the body. This was achieved by assuming that the recorded running style was both symmetric and cyclic. The original data were shifted in time by a half cycle so that the running style was reasonable. The movement of the contralateral hip and shoulder were assumed to be the same as the ipsilateral hip and shoulder. This is the same as 180 degrees out of phase. The final data array contained information for a half cycle; from contralateral toe off to ipsilateral toe off.

It is suggested that the purpose of the foot for the simulation of running was to adequately represent the movement of the ankle and the relative position of the plantar surface of the foot at heel strike and toe off. Therefore, the size and shape of the foot were determined by examining the recorded positional data of the ankle. A computer programme was developed which graphically presented the original ankle kinematics and allowed the user to modify interactively the radius of the arc and location of the ankle. The parameters were heuristically determined using this programme. It is submitted that digitisation of the ankle markers is sufficiently noisy that the mid-stance difference between the recorded movement and the simulated movement can be ignored. The data collection process at first glance appears quite artificial since so much data was inferred from so little information. As the simulated movement profile will never look exactly like the recorded movement because of physical differences and because of random noise associated with the data collection process, the simplest data collection process possible was employed. It is suggested that the recorded movement is only a general basis for the simulation and that any reasonable movement would suffice.

2.3. The Interactive Graphics Programme

In a process analogous to the development of the model, the interactive programme was designed in distinct phases. The first programme consisted of a simulation of the one segment model. This implementation is of minimal simulation complexity thus the task focussed on the interactive algorithm. The second programme introduced the running stride and was used to test the practicality of modifying technique by modifying interactively the force profiles. Photographs of the three interactive graphics screens can be seen in Figures 2-3a, 2-3b and 2-3c.

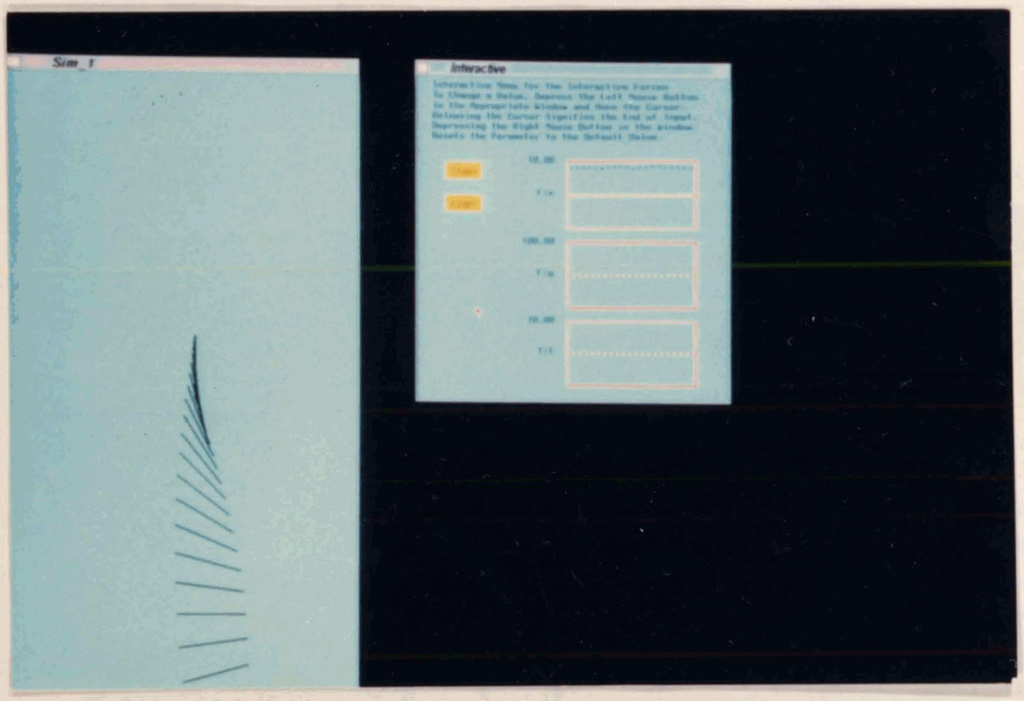


Figure 2-3a Interactive Graphics Screen I

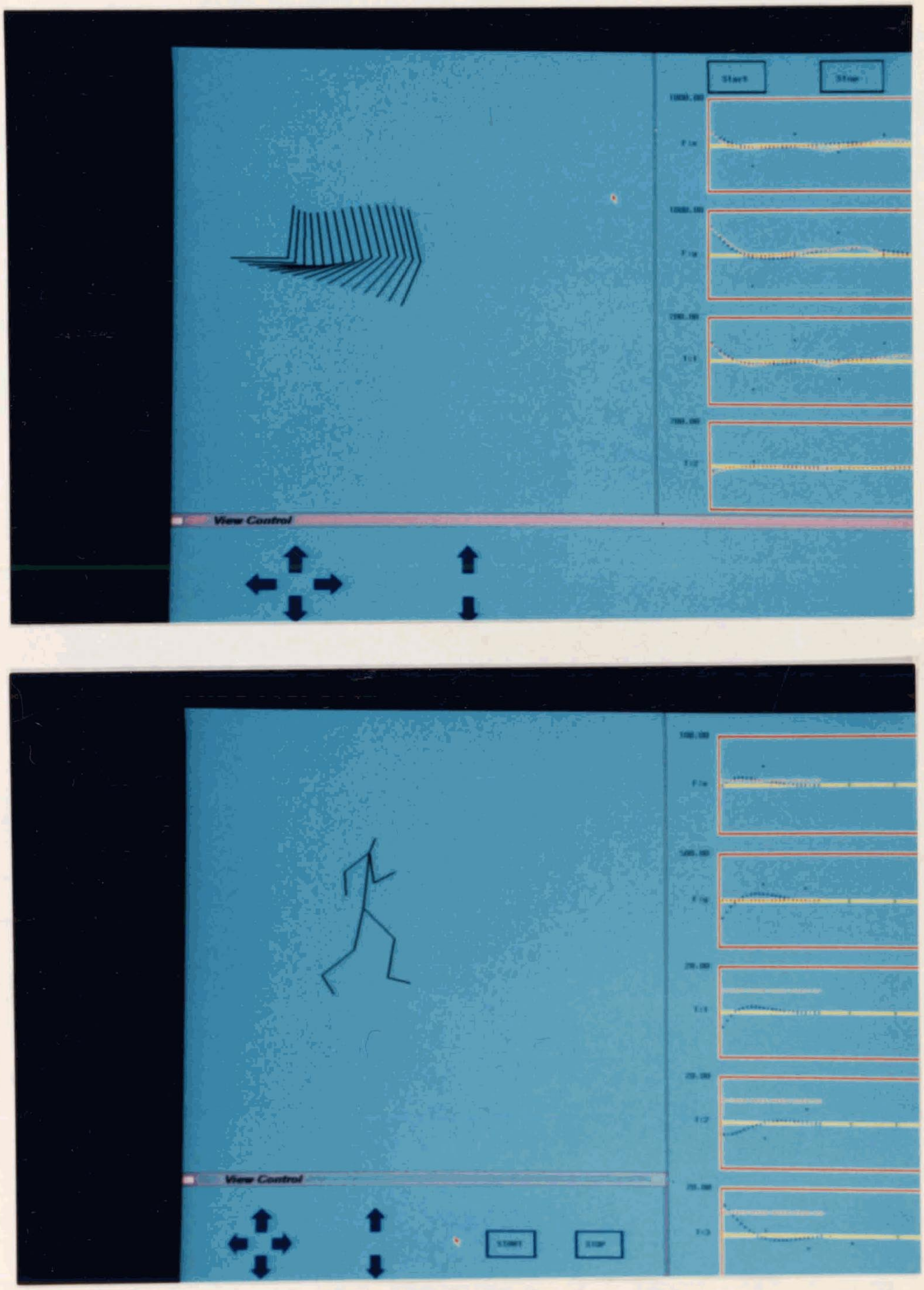


Figure 2-3b Interactive Graphics Screen II

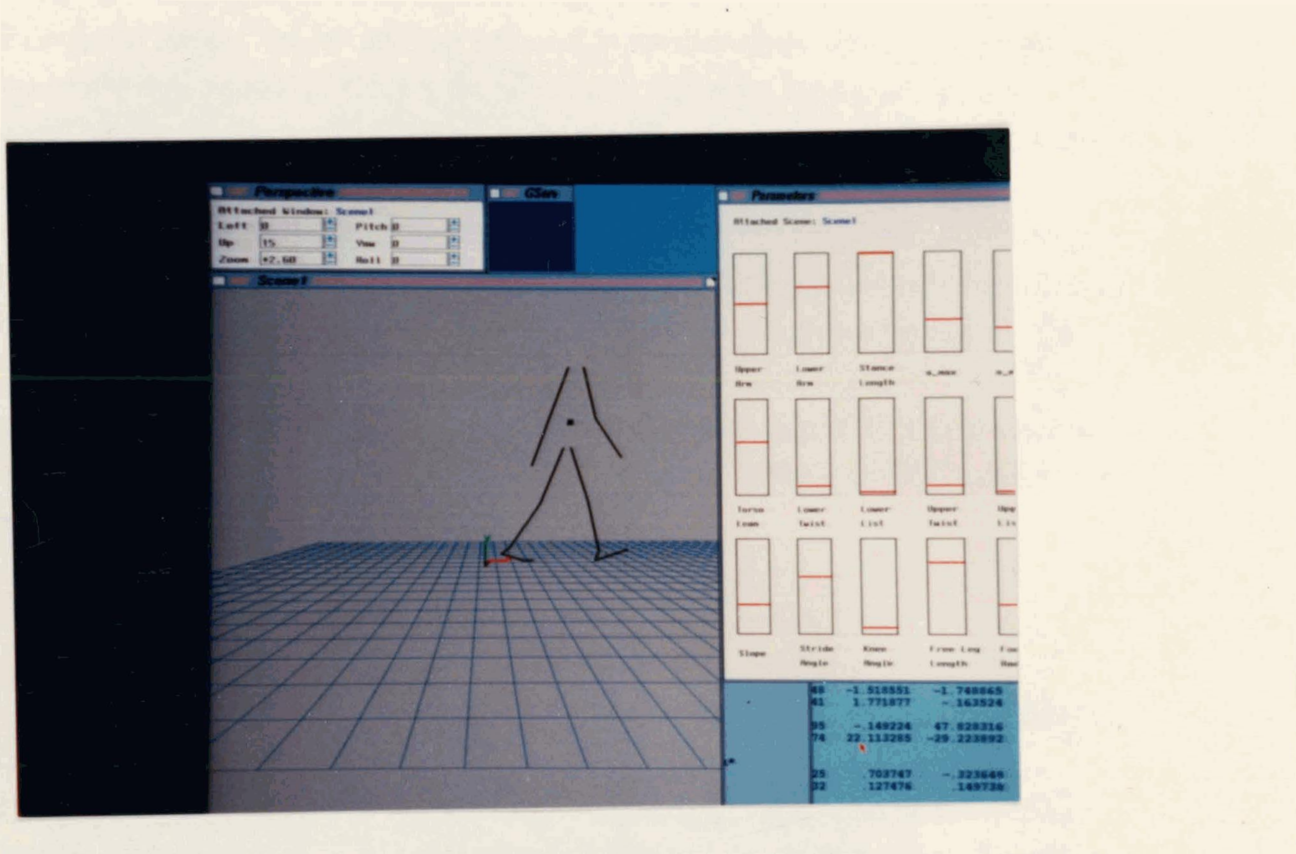


Figure 2-3c Interacitve Graphics Screen III

2.3.1. The Basic Programme

The final set of interactive programmes were developed on a Personal Iris Graphics Workstation. With the exception of the LSODI integrator, the programme has been written in C, using Silicon Graphics 4sight graphics windows and GL graphics library. The LSODI integrator was written in Fortran and was modified only slightly for adaptation to this thesis[1]. The programme consisted of two graphics windows situated side by side on the computer screen. The left window was used as the display window. The user was presented with the option of clearing the screen after redrawing the model. It was determined that it is easier to understand the interactive procedure by not clearing the screen between redraws. The computational task was minimal and there was no delay in the presentation of the simulation, the simulation proceeded too fast for modification during the simulation time. It was found to be more practical to wait until the simulation was finished before the force profiles were modified. The programme was organised to identify the location of the left mouse button when it was depressed. The location of the cursor determined the magnitude of the force value for that particular time. Note that the user can modify only the magnitude and not the timing of the forces. When the user is satisfied with the force modifications the simulation can be restarted.

2.3.2. The Two Segment Simulation

The major changes to the second interactive programme were the introduction of the forces calculated from the boundary value problem and the optional use of a bezier polynomial to represent each force profile. The simulated movement was therefore, that of a running stride. The modification of the force profiles had the obvious effect of modifying the running stride. Of significance, was the fact that only minor changes in forces resulted in simulated movements that were not representative of running. Running seems to be sensitive to the distribution of force inputs, and the modification of a single force requires the modification of all the rest of the forces. A major improvement to the user friendliness of the programme was the implementation of a representation of the force profiles by bezier polynomials. The user then need only modify the bezier control points. Since these are far fewer in number, as low as six points for 60 force values, and ensure continuity of the force profiles, it was easier to produce sensible movements. The bezier control points can

[1]The programme was modified to include a user defined procedure for identification of the violation of the ground constraint.

be seen as the black points in the force profile windows. Close examination of the two photographs of the two segment interactive programme reveal that the six bezier control points have been modified in the second picture. It was a simple process to modify the control points to produce an approximation to the original force profile[1]. A more accurate representation of the forces would require a greater number of bezier control points but no significant change in complexity. A third window was introduced which provided controls to modify the display screen (e.g. shift the focal point and zoom).

2.3.3. The Twelve Segment Simulation

The algorithm for the twelve segment simulation was identical to the algorithm for the previous two programmes. The major distinction between the programmes is that the twelve segment interactive programme displays only five of the force profiles at one time. The user must click the mouse in the window to scroll through all of the force windows. The programme was sufficiently fast on the Personal Iris Workstation that the limiting factor is the time to make force modifications, not the length of time to simulate the movement.

In summary, the interactive programmes are easy to use and provide the user with a menu interface for choosing the representation of the force profiles, either individual forces, bezier polynomials or predetermined forces from a file. The user is also able to modify the initial kinematic state. The inclusion of kinematic constraints is currently being implemented.

---

[1] In fact, it only took two minutes to generate the four profiles shown.

Chapter 3
Results

This chapter presents the results of the simulated performance of five different models:

1. A Compound Pendulum.
2. A One Segment Rigid Body.
3. A Two Segment Rigid Body.
4. A Twelve Segment Rigid Body with a Simple Foot.
5. A Twelve Segment Rigid Body with Rolling Constraint.

Four major aspects of the model development were addressed:

1. The physical construct of the model.
2. Derivation of the equations of motion.
3. Integration of the equations of motion.
4. Validation.

The first stage in the development of the running model was identification of the dynamic properties of the simplest component in the model. This simple model had relatively little practical value, but was a reasonable theoretical starting point for the development of the running simulation. Using this simple model several stages of the simulation process were examined unencumbered by algebraic complexity. The advantage of testing the analytical procedure on a simple model is apparent, as errors can be more easily identified and corrected at this stage.

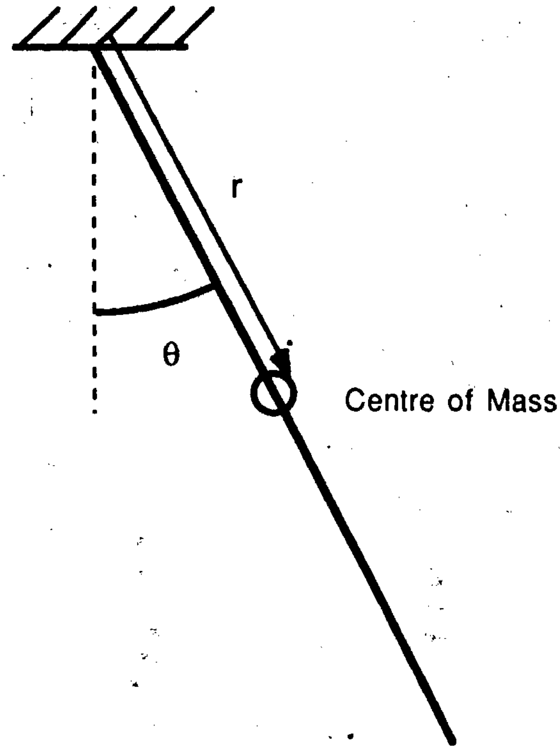


Figure 3-1 Reference Orientation for the Compound Pendulum

3.1. Compound Pendulum

For a model of the human body the least complex component is a compound pendulum. The equation of motion for the model of Figure 3-1 can be written as:

$$F_q = (I+mr^2)\ddot{q}+mgrsinq$$

where:

I = Moment of Inertia about the Centre for Mass
m = Mass of the Pendulum
r = Proximal Distance to the Centre of Mass
F_q = Applied Torque

In this section the segmental parameters are arbitrary and were chosen for convenience. The equation of motion was linearised about 32 evenly spaced angular positions covering a range of 2 pi radians with the eigenvalues subsequently calculated. The eigenvalues were calculated from the following formalism:

A y = l J y

where:

A = The Inertia Matrix

J = The Jacobian of the Transient Terms

y = The State Vector

l = The Eigenvalue Vector

Note that there is no velocity dependence in either of these terms. The eigenvalues have been calculated for several positions and are listed in Figure 3-2. The large range of eigenvalues, both imaginary and real, signify a potential instability for the numerical integrator. The LSODI integrator used in this thesis limits the effect of a numerical instability as an instability will likely produce a recognizable divergent solution. The resulting error message and system interrupt from the integrator will prevent the output of useless information.

Orientation	Eigenvalue	
0	$\pm i$	3.836
0.196	$\pm i$	3.799
0.393	$\pm i$	3.687
0.589	$\pm i$	3.498
0.785	$\pm i$	3.226
0.982	$\pm i$	2.859
1.178	$\pm i$	2.373
1.374	$\pm i$	1.694
1.571	$\pm i$	0.002
1.767	$\pm$	1.694
1.963	$\pm$	2.373
2.160	$\pm$	2.859
2.356	$\pm$	3.226
2.553	$\pm$	3.498
1.749	$\pm$	3.687
2.943	$\pm$	3.799

Figure 3-2 The eigenvalues for the Simple Pendulum of Figure 3-1.

3.1.1. Validation

Once the model was defined and the equations of motion written, movements of the model were validated[1].

The movement of the one segment pendulum was simulated over several different initial conditions using the LSODI integration subroutine. The initial kinematic state was supplied to the initial value problem. Figure 3-3 shows the movement of the pendulum for three different initial conditions. Included in the same figures are the solutions for the mathematical pendulums for the same conditions. Note that the analytic solution and the numerical solution are similar for the smallest angle but are less reasonable for the larger angles. Since running shows large variations in angle it is suggested that a linearised model for running is inappropriate. The period of oscillation is seen to vary with the amplitude. There was no artificial gain in energy and no phase delay for these simulations. The phase delay was assumed to be negligible on the basis that successive periods of the movement were identical. A phase shift would cause a gradual change in the period of the cycle. As can be seen in Appendix A, explicit numerical methods introduce artificial gains in energy. The instantaneous energy of the simulation of the conservative movement was calculated at specified intervals of time. The maximum deviation was less than 0.001%.

Several other conditions were tested to ensure that the numerical method was convergent and that the solutions were qualitatively reasonable. These conditions consisted of various initial states of the pendulum, and various force profiles. As an indication of the amount of non-linearity Figure 3-3 includes the solution to a linearised pendulum for the same initial conditions. An initial validation test for a compound pendulum consisted of a comparison of the simulation results with the solution of a mathematical pendulum.

---

[1] The validation tests were discussed in the previous section.

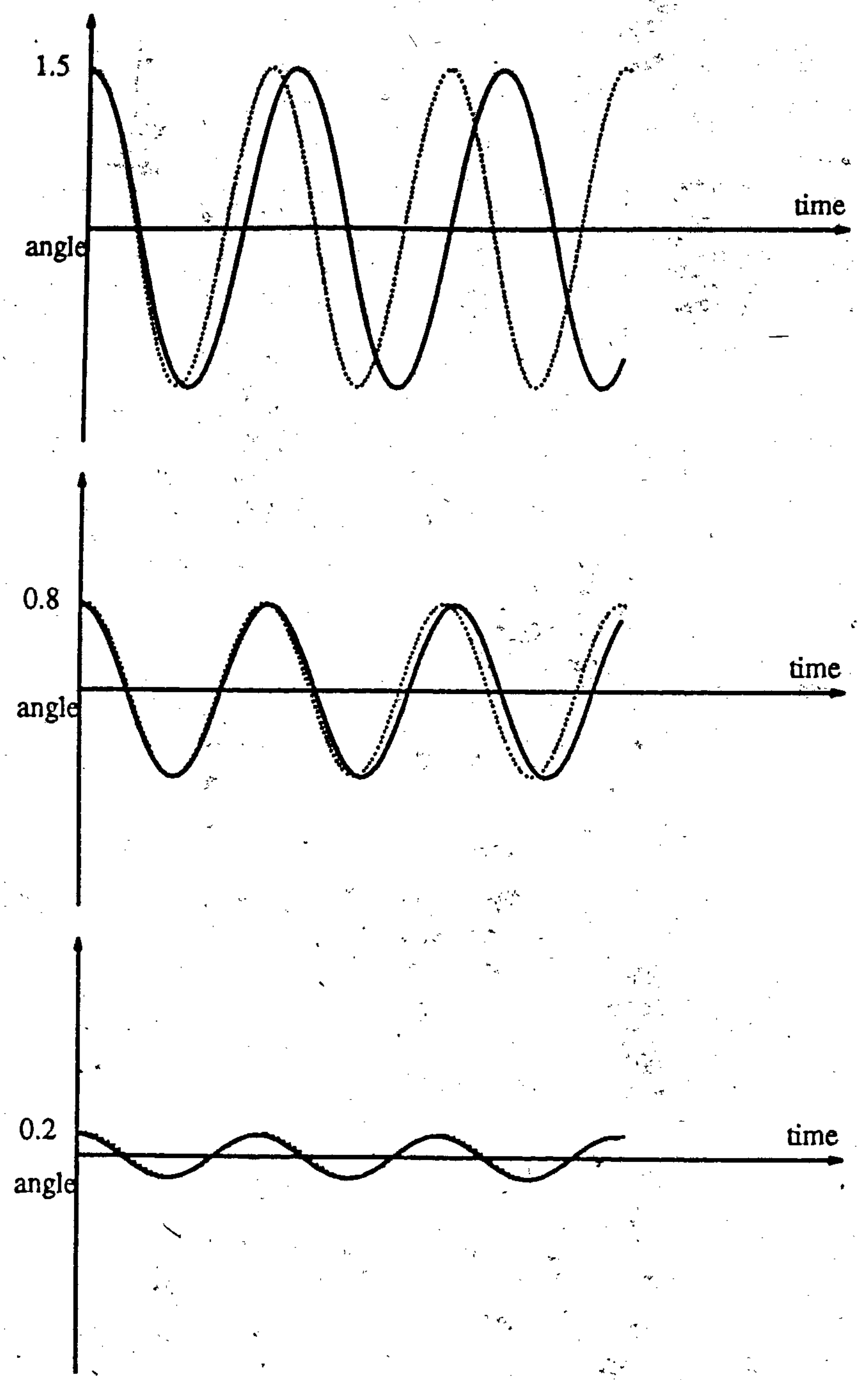


Figure 3-3 Time Course Solution for the Movement of a Simple Pendulum for three different Initial Conditions: q_0 = 1.5 , 0.80 , 0.20 radians. The dotted line shows the mathematical pendulum. The solid line shows the simulated pendulum.

The mathematical pendulum is a result of linearising about $q=0$ as follows:
for $q \ll 1$ for $\sin q \approx 1$ the equation of motion for the pendulum is approximated by:

$$F_q = (I+mr^2)\ddot{q} + mgrq = 0$$

Given the initial conditions:

$$q(0) = 1 \qquad \dot{q}(0) = 0$$

The solution is:

$$q(t) = \cos wt \quad \text{where:} \quad w^2 = \frac{mgr}{I+mr^2}$$

Although absolute validation is impossible, it is submitted that the results from the modelling of the single segment pendulum provide justification for increasing the complexity of the model. The simulation of the compound pendulum verified that methods of deriving and integrating the equations of motion were reasonable and that the resulting numerical simulations were consistent with the results of the mathematical pendulum. Comparison with a mathematical pendulum is by no means an adequate validation of the model but does indicate that the simulated movement is reasonable.

3.2. A One Segment Planar Object

The second stage in the development of a full body model involved an increase in the number of degrees of freedom for the one segment pendulum. This was done to allow the object to move freely through space with three degrees of freedom (two translation and one rotation). The model was then used to study impact conditions when a free body suddenly becomes constrained.

3.2.1. Equations of Motion

The equations of motion were derived using Lagrangian dynamics. The equations of motion for the model described in Figure 3-4 can be written as:

$$F_x = m\ddot{x} - mr(\ddot{q}\cos q - \dot{q}^2 \sin q)$$

$$F_y = m(\ddot{y}+g) - mr(\ddot{q}\sin q + \dot{q}_1{}^2 \cos q)$$

$$F_q = (I+mr^2)\ddot{q} - mr(\ddot{x}\cos q + (\ddot{y}+g)\sin q)$$

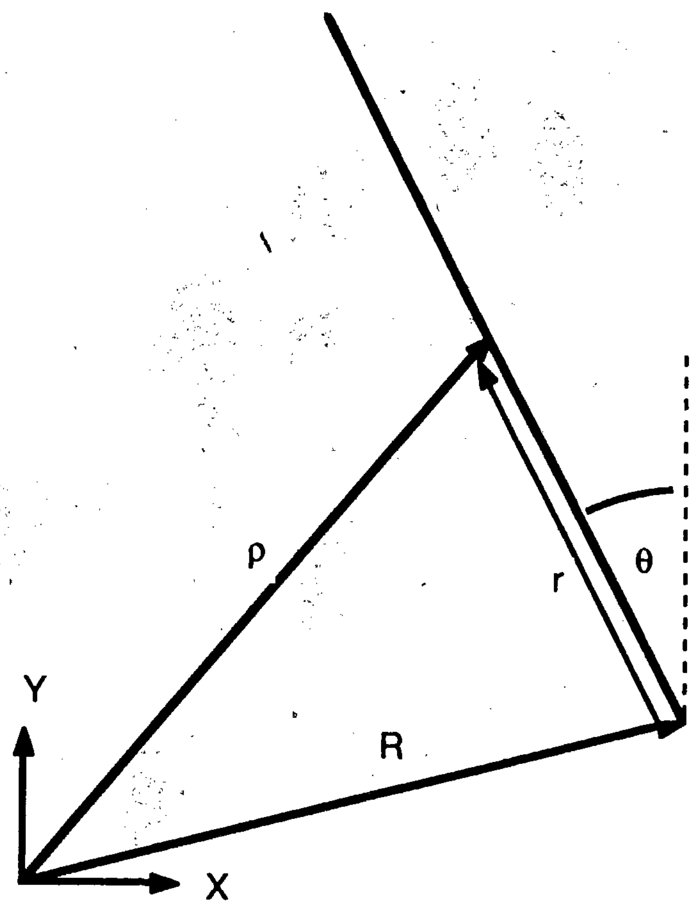


Figure 3.4. Definition of the One Segment Model with Three Degrees of Freedom. The Proximal End is Constrained.

These three second order differential equations were then transformed into six first order equations, using the method describe in Chapter 2. The resulting system of equations and the representation of these equations for the numerical integrator are listed in Appendix D.

3.2.2. Validation

The first validation test involved allowing the model to drop freely in space with different orientations of the segments. The initial conditions were various positions from zero velocity.

Theoretically, the system should fall vertically with a constant acceleration of 9.81 ms^{-2}, with no reorientation of the segments. There should be no movement in the horizontal direction. Using the equations $y=\frac{1}{2}gt^2$ and $v=-gt$ it is possible to calculate the distance which the centre of mass of the system will fall in a given time, and the final velocity it will attain[1]. If t= 1.0 seconds is substituted into the equations of projectile motion the displacement calculated is 4.905 metres and the final velocity is -9.81 ms^{-1}.

The equations were integrated using the LSODI subroutine. The simulation yielded a distance of 4.905 metres and a velocity of -9.81 ms^{-1} and was therefore consistent with the theoretical results.

The second validation test involved the study of the rotation of the model under free-fall conditions. The segment was given an initial angular velocity and no external forces were applied. The object was given an initial w= 1.0 s^{-1}, after two seconds of simulation w= 1.0 s^{-1}. Note that the angular velocity remained constant as expected.

The reference point was then constrained stationary. The resulting equations of motion are:

$$F_q = (I+mr^2)\ddot{q} - mgr\sin q$$

This derivation of the pendulum is only superficially different from that described in the previous section. The reference orientation is chosen as a matter of convenience. For

[1] With respect to the computer algorithm the reference point is a more convenient point to follow than the centre of mass and its movement is analogous to that of the centre of mass if there is no rotation of the segment thus constraining every point to fall with exactly the same acceleration and velocity.

comparison with a mathematical pendulum it is convenient to use the zero angle down for the sake of linearisation. For the study of impact it was desired to consider the zero angle directed up. In the final running model the reference angle position is specified in one of two reference positions, either up or down depending on the segment. It was considered useful to have the movement of a segment confined to the first positive and first negative quadrant. The direction of increasing angle does not however change. It is submitted that the initial validation tests were successful. The model could easily be defined in both the constrained and unconstrained states. The next task was to develop equations to account for the transition between states.

3.2.3. Impact Conditions

At impact the angular momentum about the instantaneous point of constraint is conserved during the state change. The linear momentum is obviously not conserved as an instantaneous impulse is applied. The mathematical conditions at impact are:

before impact:

$$H_A^- = (I+mr^2)\dot{q}^- + mr(\dot{x}^- \cos q + \dot{y}^- \sin q)$$

after impact:

$$H_A^+ = (I+mr^2)\dot{q}^+$$

At Impact

$$H_A^+ = H_A^-$$

$$\dot{q}^+ = \dot{q}^- + \frac{mr}{I+mr^2}(\dot{x}^- \cos q + \dot{y}^- \sin q)$$

Figure 3-5 shows the result of dropping the object from an initial height letting it fall for a specific amount of time and suddenly constraining the reference point.

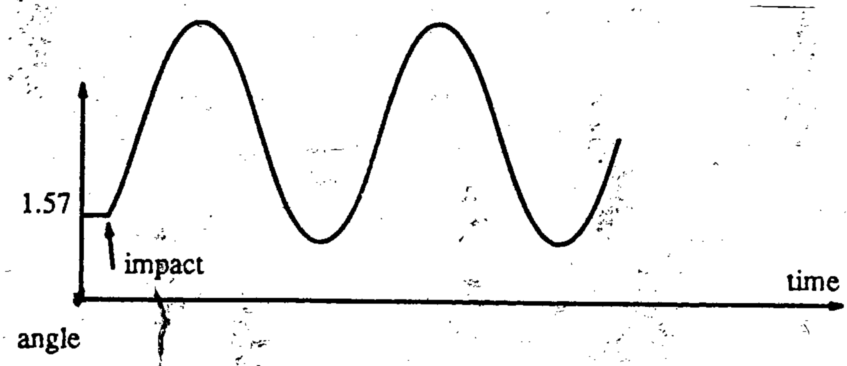


Figure 3-5 Impact experiment: The object is constrained after falling for 0.25 s

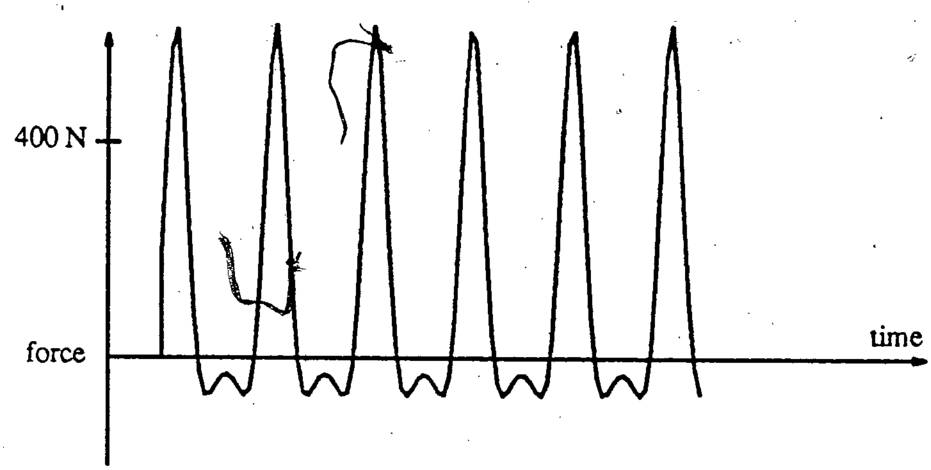


Figure 3-6 Showing vertical constraint force profile. The object is constrained after falling for 0.5 seconds

One can also study the release conditions if one assumes that the constraint force acts only in one direction (e.g. a ground constraint). Release occurs when the constraint force becomes directed down rather than up. Figure 3-6 shows the calculation of the vertical force while the pendulum is constrained. Note that a negative force would indicate a release of the ground constraint. Obviously the constraint was not released in the movement described in Figure 3-6.

3.3. Two Segment Planar Object

The next stage in the development of a full body model involved an increase in the number of segments. The second model consisting of a two segment object which comprised four degrees of freedom, two translational (movement of the reference) and two rotational. The model was used to simulate the movement of a two segment leg during a running stride. It was given the anthropometric measurements equivalent to the thigh and the shank & foot segments of the subject.

3.3.1. Equations of Motion

The equations of motion were derived using Lagrangian dynamics. The equations of motion (see Figure 3-7 for reference orientations) can be written as follows:

$$F_x = (m_1 + m_2)\ddot{x} + (m_1 r_1 + m_2 l_1)(\ddot{q}_1 \cos q_1 - \dot{q}_1^2 \sin q_1)$$
$$+ m_2 r_2(\ddot{q}_2 \cos q_2 - \dot{q}_2^2 \sin q_2)$$

$$F_y = (m_1 + m_2)(\ddot{y} + g) + (m_1 r_1 + m_2 l_1)(\ddot{q}_1 \sin q_1 + \dot{q}_1^2 \cos q_1)$$
$$+ m_2 r_2(\ddot{q}_2 \sin q_2 + \dot{q}_2^2 \cos q_2)$$

$$F_{q1} = (I_1 + m_1 r_1^2 + m_2 l_1^2)\ddot{q}_1 + (m_1 r_1 + m_2 l_1)(\ddot{x}\cos q_1 + (\ddot{y} + g)\sin q_1)$$
$$+ m_2 r_2 l_1(\ddot{q}_2 \cos(q_2 - q_1) - \dot{q}_2^2 \sin(q_2 - q_1))$$

$$F_{q2} = (I_2 + m_2 r_2^2)\ddot{q}_2 + m_2 r_2(\ddot{x}\cos q_2 + (\ddot{y} + g)\sin q_2)$$
$$+ m_2 r_2 l_1(\ddot{q}_1 \cos(q_2 - q_1) + \dot{q}_1^2 \sin(q_2 - q_1))$$

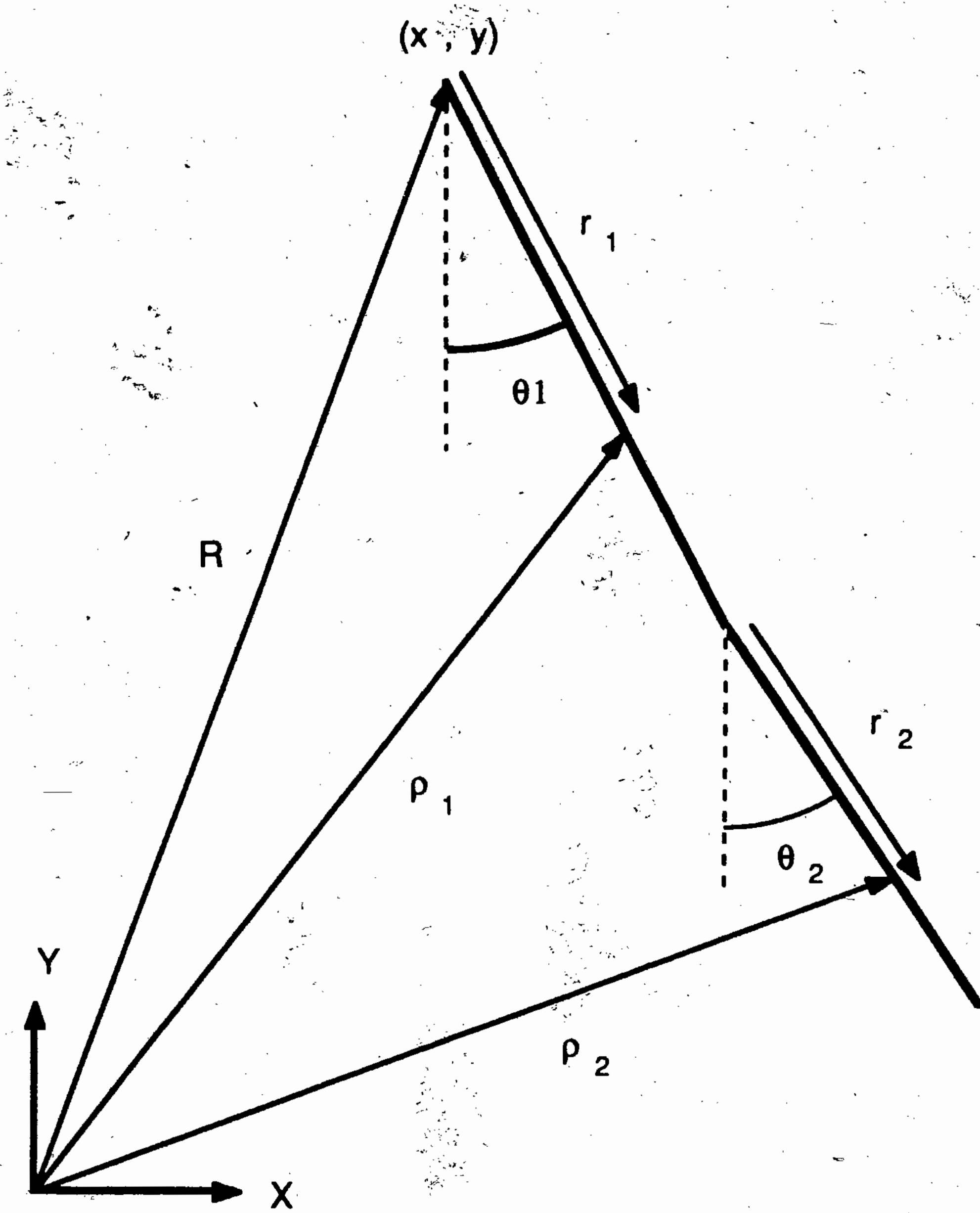


Figure 3-7 Reference Orientation for the Two Segment Rigid Body.

These four second order differential equations were then transformed into eight first order equations as listed in Appendix D. The eigenvalues were calculated for various states of the system. The wide range of possible values was similar to the range for the one segment model. A change in velocity results only in a change in the magnitude of the eigenvalues. It was not necessary to consider the movement of the reference vector as it had no effect on the eigenvalues. There was a high degree of nonlinearity in the system equations, for the arbitrary states examined. The orientations near the horizontal were the most nonlinear. It is interesting to note that the eigenvalues for conservative states always lay either on the imaginary or real axis, and as discussed in the methods, purely imaginary eigenvalues require an A-stable numerical integration method.

3.3.2. Validation

Validation tests similar to the ones used for the one segment object were examined. The results for these tests were analogous to the one segment tests and it was determined that the mathematics were consistent with theory and that it was necessary to test the model as an analogy to the human leg.

3.3.3 Simulation of a Running Stride

The movement of the free leg during a running stride was then simulated. The model represented the thigh and the shank & foot segments. The data collection procedure was explained in Chapter 2. As an example only, the recorded data were not manipulated to improve the smoothing. In particular, no extra data were added to the beginning and end of the data array before smoothing. The position and velocity profiles were input to the boundary value problem for the calculation of the necessary forces. The simulation was then performed with the calculated forces. The results of the simulation and the recorded kinematics are displayed in Figure 3-8. Note that forces due to the ground constraint were not explicitly considered. For simplicity, they are included in the joint torques and in the reference forces. The joint torques calculated are therefore not the net torque contributions from the musculature during the support phase.

It is submitted that the resulting simulated movement is a reasonable reproduction of the original kinematics and as such is a reasonable analogy to the movement of the human leg. Note that the initial velocity vector is obviously not correct as there is a significant difference between the original and simulated profiles. It is suggested that modification of the initial kinematic state would improve the match between simulated and recorded kinematics. The discrepancy between the simulated and recorded movement has been attributed to the smoothing of the data. The quintic spline smoothing routine is particularly poor at defining the start and end of the data array. Manipulation of the data as explained in Chapter 2 rectified this problem. The resulting simulation from the new data cannot be graphically distinguished from the original data.

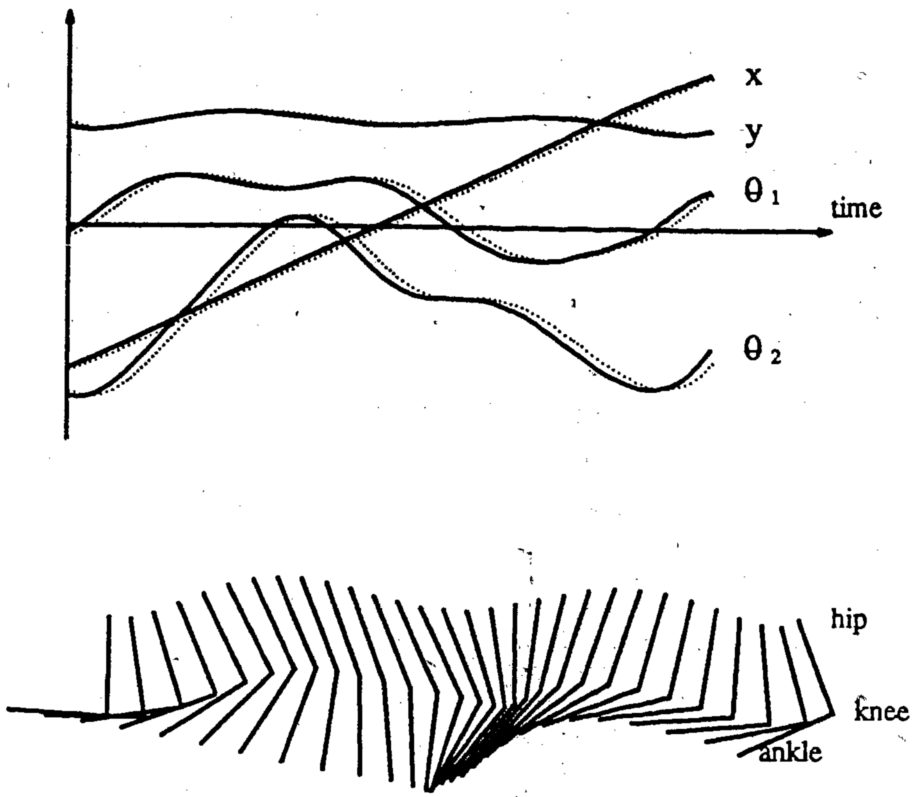


Figure 3-8 The simulated running movement for the two segment model. The original kinematics are the dotted line.

3.4. Twelve Segment Model Using an External Force

The model was extended to three segments, then to six segments and finally to twelve segments. Two twelve segment models were developed. They differed in the manner in which the ground constraint was represented.

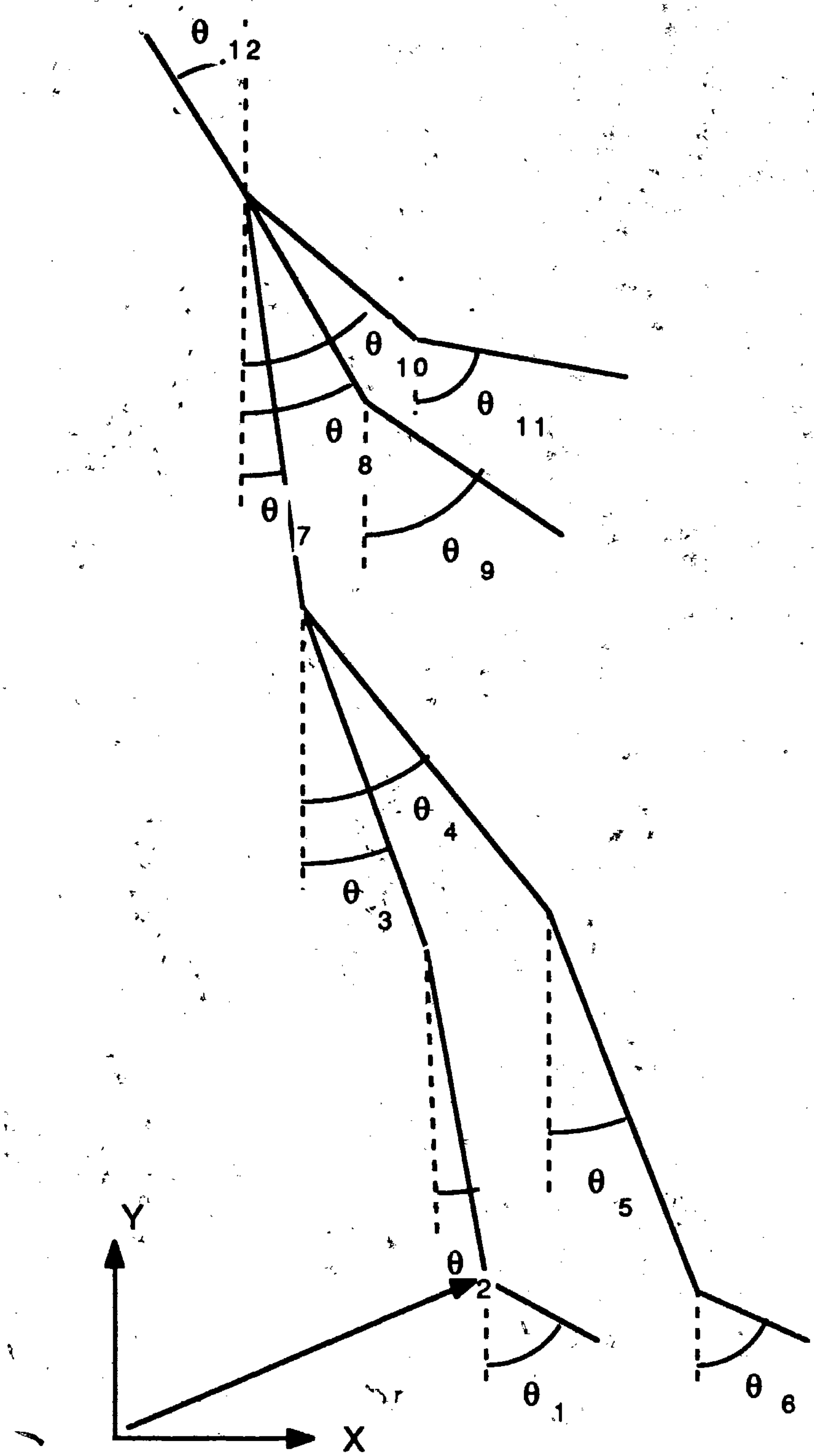


Figure 3-9 Description of the twelve segment model without a foot.

Intermediate models were also implemented but have not been presented as they were developed purely for mathematical reasons and not as endproducts in themselves. The twelve segment models were mathematically verified in the same manner as the other models. The equations of motion are presented in Appendices H and I. The running stride was simulated from Contralateral Toe Off to Ipsilateral Toe Off. The movement was unconstrained during the airborne phase. In the first model the ground constraint was imposed by introducing an external force at the ankle joint. The kinematic results are presented in Figure 3-9. The stance foot kinematics were not a meaningful addition to the model since they were the result of adding a second model at the reference point. This was added for completeness and was eliminated when the foot model was added later.

The constraint force was applied at the ankle, as this description of the model was best suited to the addition of a foot segment attached to the ankle. This force vector does not accurately represent the ground constraint as the properties of the foot are not included. It is however a reasonable representation of the joint reaction force at the ankle and therefore the joint torque profiles for the rest of the body are adequate. The simulation was useful as it showed that the model (with the exception of the foot) was in fact capable of reproducing the required kinematics. Figure 3-10 shows the comparison of the simulated kinematics and the recorded kinematics along with a stick figure representation of the simulated running stride.

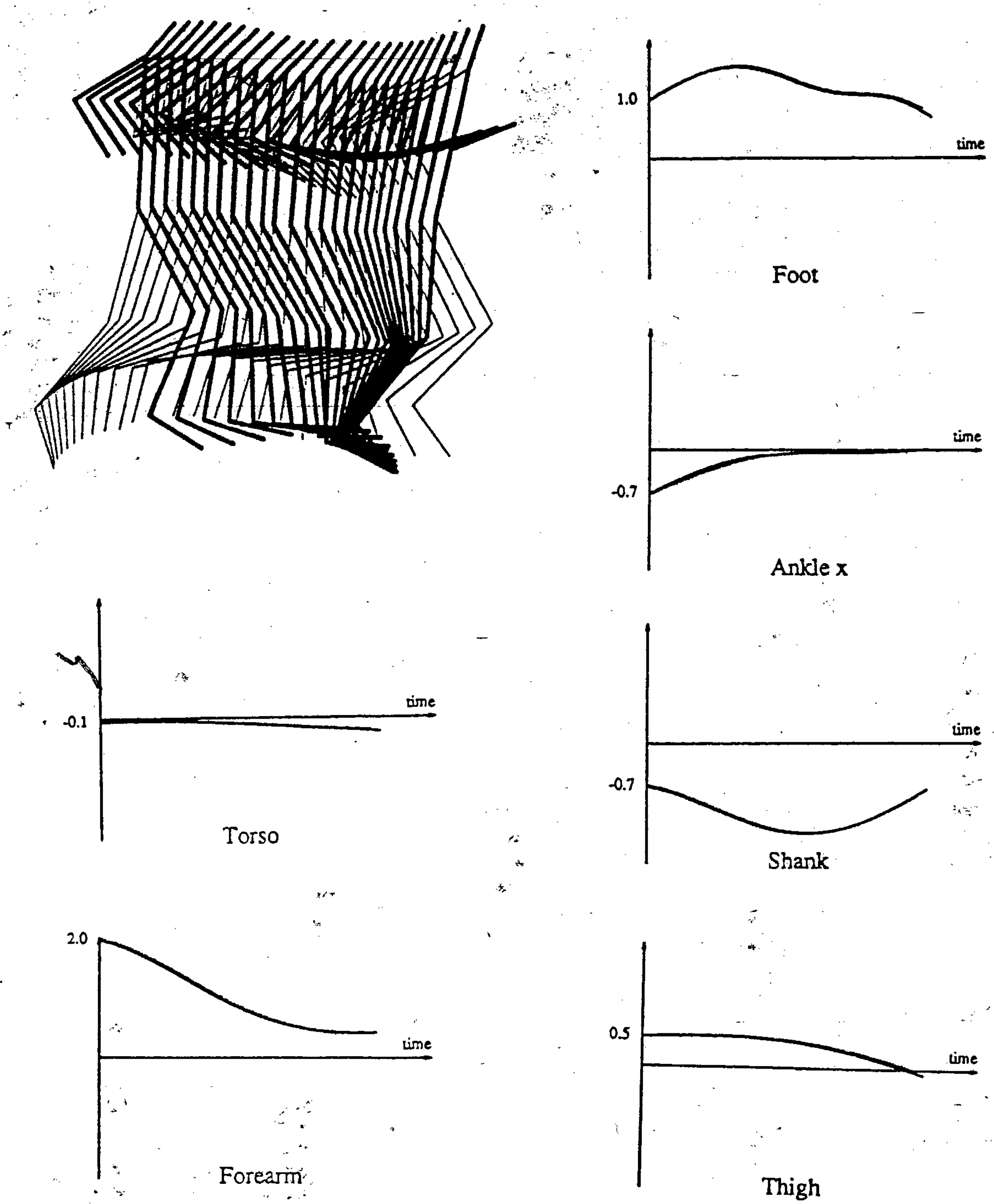


Figure 3-10 Simulation of the Running Stride from Contralateral Toe Off to Ipsilateral Toe Off using an external force as the ground constraint. The Ipsilateral side of the body is shown with the solid lines. The original kinematics are drawn in dotted lines.

3.5. Twelve Segment Model with a Foot

The inclusion of the foot into the model was a relatively straightforward task. The equation of motion of the foot was combined with the equation of motion of an eleven segment body whose reference vector was at the ankle. The two sets of equations were combined by introducing the reference force of the 11 segment model as a distal force to the foot model with the ankle kinematics being defined by the ankle kinematics of the foot. This method allowed the derivation of the foot by Newtonian mechanics and the derivation of the rest of the model by Lagrangian mechanics (see Appendix G). The equations of motion are listed in Appendix I. A graph of the angle of the foot during stance is shown in Figure 3-14. The kinematics of the rest of the angle are unchanged from Figure 3-10.

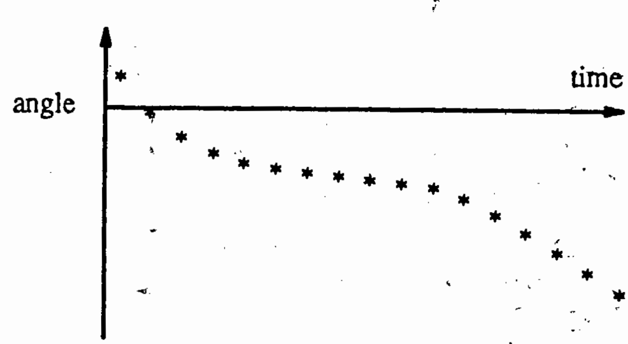


Figure 3-11 The simulated movement of the angle of the foot.

3.5.1. Representation of the Foot

This section describes the construct of the model of the foot. Ju & Mansour (1988) represented the plantar surface of the foot by a second order polynomial in the sagittal plane. The curve was constructed from a statistical fit to recorded data. This method is satisfactory for reproducing the movement but is suggested that this is inadequate for a general model. In this thesis the foot was represented as a finite radius arc with the ankle joint lying part way along the radial axis to the center of the arc, as seen in Figure 3-12. The ground constraint was then represented by a rolling constraint and no external force was required. With this representation the foot model can be changed by modifying only the radius of the arc. For a general model, it may also be possible to define the radius as a function of the running speed. It has been suggested (Personal communication with Tad McGeer) that the radius of the foot varies proportionally to the inverse of the running speed. This has not been tested but is a reasonable project for future research. The equations of motion representing the foot are:

$$F_q = [I+m(R^2-2rR\cos q+r^2)]\ddot{q} + mgr\sin q$$

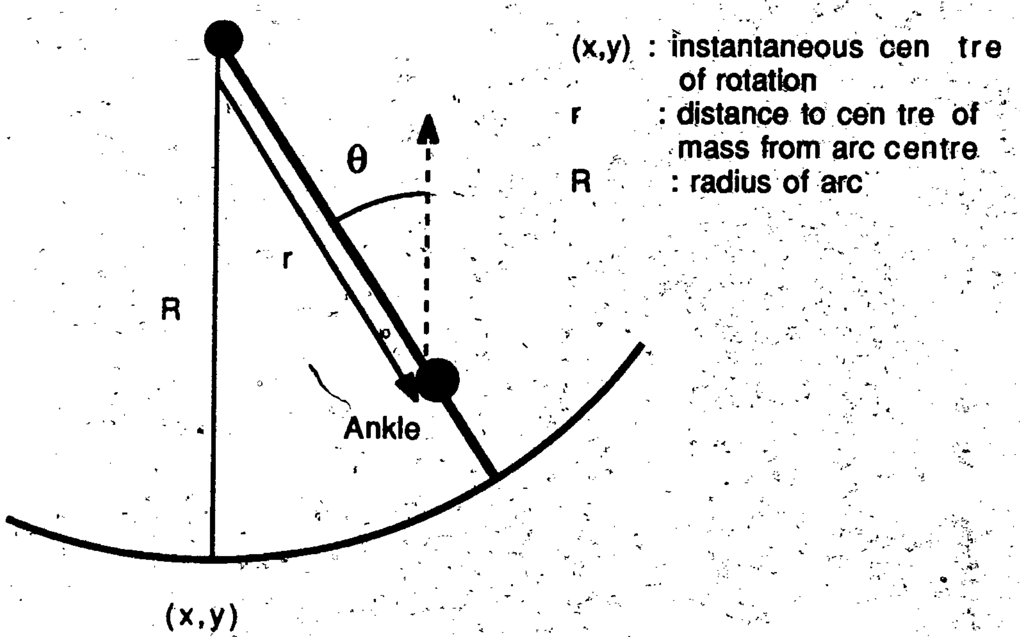


Figure 3-12 The description of the foot

The movement of the foot was simulated in isolation to validate the physical construct. Figure 3-13 shows the results of a simulated rocking movement of the foot. Rocking was initiated by rotating the foot to an angle q to the vertical and releasing it from rest. The centre of mass is located at the ankle joint and since it is closer to the ground than the centre of the arc, the foot should rock back and forth as observed.

The second simulation test confirmed that the recorded kinematics of the ankle joint[1] could be reasonable represented by the movement of this foot. Figure 3-14 shows the comparison of the recorded kinematics with the simulated movement of the ankle. It is suggested that the important attributes of the ankle kinematics are the initial and final velocity during stance. A computer programme was developed which graphically presented

[1]The choice of foot representation was based on an attempt to reproduce the ankle kinematics from a rolling constraint.

the original kinematics and allowed the user to modify interactively the radius of the arc and location of the ankle. The parameters were heuristically determined using this programme. It is submitted that digitization of the ankle markers is sufficiently noisy that the midstance difference between the recorded movement and the simulated movement can be ignored.

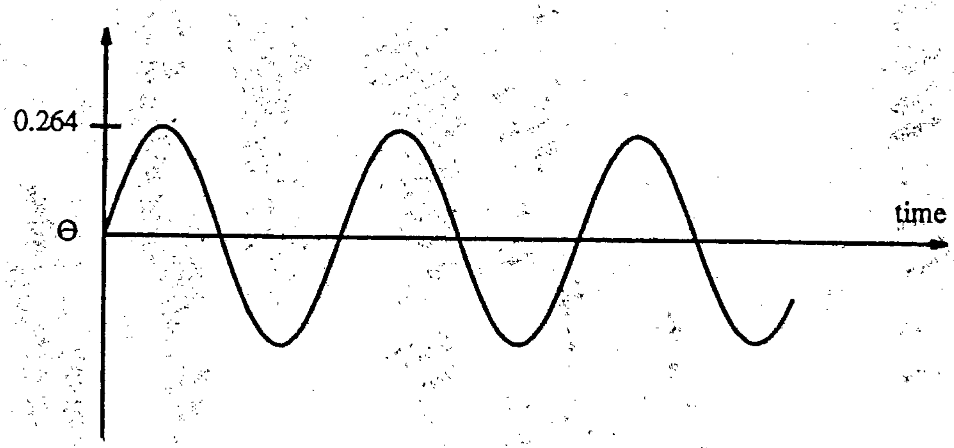


Figure 3-13 Simulation of the foot rocking back and forth. The foot is initially given an angular velocity of 1 s^{-1}

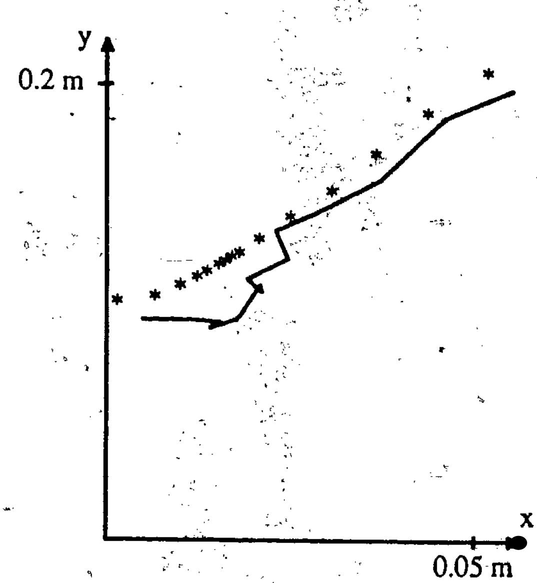


Figure 3-14 A comparison of the simulated movement of the ankle and the recorded ankle kinematics. The simulation results are plotted with a "*".

Chapter 4
Discussion

This discussion has been divided into two sections: a summary of key issues in the methods chapter and a critique of the fundamental structure of the prototype interactive graphics programme.

4.1. The Methods

This section addresses five aspects of the mathematical development of the running simulation:

1. Derivation of the equations of motion,
2. Integration of the equations of motion,
3. Definition of the boundary value problem,
4. Introduction of the ground constraint,
5. Representation of the foot.

4.1.1. Derivation of the Equations of Motion

The most notable aspect of the method of derivation was the use of both Lagrangian and Newtonian mechanics. Although it is more common for researchers to choose a single method of derivation, it is submitted that it was beneficial to combine the methods for this task. With the exception of the final model, the equations of motion were preferentially derived using Lagrange dynamics, and impact conditions were always based on the assumption that the angular momentum is conserved about the instantaneous contact point. The introduction of a foot model consisting of a finite radius arc rolling on the ground was a more complex task for Lagrange dynamics than the previous models consisting of holonomic constraints. It was considered straightforward to model the rolling constraint using Newtonian mechanics, thereby avoiding the introduction of Lagrange multipliers. Rather than redefining the rest of the model, as the equations of motion for the unconstrained state were already derived, the reaction force at the ankle joint was used to couple the foot to the rest of the body model. The equations describing the reaction forces at the ankle were algebraically eliminated. The system of differential equations was therefore of minimal rank. It is suggested that this approach was an improvement on using a single method of derivation. Several researchers have suggested that the method of deriving equations of motion is of the utmost importance to the efficiency of the simulation (Armstrong, Green & Lake, 1987, Wilhelms, 1988). It is suggested that because of the explicit definition of the model, the choice of derivation technique here was inconsequential, particularly since the equations were subsequently altered after derivation for algorithmic presentation to the numerical integrator. Any differences in the initial form of the equations could be eliminated at the algorithm stage. The presentation of the equations to the integrator is the stage at which the equations should be in an optimal form.

4.1.2. Integration of the Equations of Motion

Simulation is the result of the integration of the equations of motion given the initial kinematic state and the torque profiles. The consideration of several issues was prerequisite to the integration of the equations of motion. One of the key factors was the determination of the range of possible eigenvalues for the model. The eigenvalues are presented for only the one and two segment models to substantiate the claim that care must be taken when choosing an integrator. The eigenvalues were also calculated for several different models and it became apparent that they were similar for all undamped linked rigid segment models, varying only in magnitude. It was unnecessary to calculate them for every model. In summary, the eigenvalues range from purely real to purely imaginary values. It is suggested that there are no numerical integration methods with this large a stability region. It is therefore advisable to choose an integration method which incorporates a convergence monitor, a variable step size and variable order numerical methods. Confidence in the simulation results should be directly related to confidence in the integrator. It was observed that the overall time course of the dynamic simulations was reasonably short. With the exception of the Stiff problems any of the techniques would have been adequate. Since most of the dynamic simulations presented in the literature are of short duration it is suggested that numerical problems had little time to manifest themselves. Irrespective of the instability problem the computational efficiency of the integrator is evident even to simulations of short duration.

4.1.3. Specification of the Integration Parameters

This section discusses the specification of integrator inputs required in the main programme. The LSODI numerical integrator requires the algorithm to define several internal integration parameters. Most of the parameters have default values but these were modified to ensure that certain conservative movements were satisfied. If the default values of the integrator are used the model does not fall with an acceleration of gravity, nor is angular momentum conserved, nor is the energy of a pendulum conserved. The integrator parameters were heuristically determined based on the satisfaction of these three conditions. The relevant optional inputs were the tolerance specifications to the convergence monitor, the types of integration methods allowable, and the maximum internal step size. The Backward Differentiation Formulas were used because they contain both an A-Stable method and a higher order method suitable for Stiff problems. The Jacobian matrix was analytically defined as the numerically calculated Jacobian is significantly less efficient and less accurate.

4.1.4. Specification of the Boundary Value Problem

Several researchers have indicated that the forces calculated from an inverse dynamic analysis can be used to reproduce the original kinematics (Wilhelms, 1988, Isaacs & Cohen, 1988). It is submitted that the calculation of instantaneous forces using experimentally obtained kinematics will not replicate these kinematics. Integration necessarily uses average force over a time step and it is unreasonable to suggest that the average forces and the instantaneous forces are identical. Furthermore, the calculation of acceleration profiles from experimental data is more sensitive to noise in the experimental data than either position or velocity. It is submitted that the use of the boundary value problem calculates forces from kinematics and is therefore analogous, but not equivalent, to a traditional inverse dynamic analysis.

The desired kinematics of the model were reproduced by solving a boundary value problem. The boundary value problem was solved using a non-linear shooting method. Modification of the forces was based on a combination of two methods. The first method simply modified the value of the forces in the direction of the error between desired end velocity and achieved end velocity. This algorithm was robust but very slow to converge. The second method used was a modification of the quasi-Newtonian algorithm for calculating zero crossings of non linear systems of equations. This second method was considerably more efficient if the estimated forces were within a small tolerance of the actual forces. The algorithm was a modification of the classical text book algorithm (Burden, Faires & Reynolds, 1981). Rather than calculating an approximate Jacobian Matrix using a small increment in the force estimate, a large increment was used. A large increment imposes the use of an average gradient rather than an approximate instantaneous gradient. The kinematic effect of the modification of some of the estimated forces was dramatic. The use of an average gradient improved both the rate of convergence and the stability. The use of an inverse dynamic analysis to specify the initial force values was considered computationally beneficial.

The choice of defining the boundary value problem with respect to the velocity rather than the position state was pragmatic. When the programmes were initially run using boundary position states there were abnormally high fluctuations in successive force values. Specifying boundary velocities eliminated this problem. In retrospect, it is suggested that the fluctuations in forces may have been caused by the initial smoothing method. Originally the data were smoothed without the addition of artificial data at the beginning and end of the file. The data were therefore particularly poor for the first several steps. The unreliability of the first few forces was apparently unrecoverable and the forces oscillated in time. The original boundary value problem was constructed using kinematics without the addition of artificial data (as discussed in the methods). The final method of smoothing resulted in position profiles that were more reasonable and it is suggested that specifying the boundary value problem in terms of position rather than velocity is reasonable. It is submitted that boundary velocity specification is still preferential for the calculation of forces as only one integral separates acceleration and velocity while two integrals separate acceleration from position. Therefore, position may not be sensitive enough to changes in force values over the time steps used in this thesis.

4.1.5. Introduction of the Ground Constraint

The action of human muscles is such that limb segments rotate relative to each other. Translation of the human body is achieved only by interaction with the external environment. In running, the progression is due to the interaction between the foot and the ground. Representation of the ground constraint in this simulation was the most difficult technical problem encountered. The simplest method of mathematically constraining an object is the inclusion of an artificial spring damper combination between the point of contact of the model and the constraint. This method was rejected for several reasons. One of the reasons was the fact that the operation of the constraint introduces artificial eigenvalues. Artificial eigenvalues impose mathematical Stiffness on the equations of motion and render the integration computationally expensive. As well, they effect the quality of the results. Another reason for the rejection of this type of constraint was the fact that the spring damper constraint may require the user of the interactive computer programme to input additional information relating to the type of constraint parameters necessary. Redefinition of the spring constants is unlikely but is certainly theoretically reasonable. Choosing a single constraint parameter for all conditions is unnecessarily inefficient. The spring parameters may vary from task to task and this puts an unnecessary burden on the user. It was considered to be desirable that the user should specify as few details as possible in the simulation, and an analytical constraint was therefore implemented.

The analytic constraint employed in this thesis was inherent to the equations of motion, thus two sets of equations were required, one for the air borne phase and one for the stance phase. It would have been possible to use one set of equations and then introduce Lagrange multipliers to describe the constraint. Since there are only two states in running, it is computationally more efficient to use the methods employed in this thesis.

It is also possible to incorporate a pre-processor to provide the necessary forces to maintain a constraint (Wilhelms, 1988) but this is not a sensible way to deal with a system that has only a single constraint. The problem with this method is that the use of a pre-processor separates the task into two levels of control. First the constraint is imposed and then the user makes interactive modifications. It is likely that the user modification would necessitate the redefinition of the constraint.

In summary, the explicit definition of the constraint used in this article makes it more appropriate for an interactive graphics programme of running than any of the other methods mentioned. The principal limitation of the method of constraint used for this programme is that it is not generalizable. The constraint used is considered an appropriate way to represent the ground constraint for human running. It is suggested that it can also be used for walking, during single support, and jumping. This method is not considered appropriate when either the location of the constraint or the type of constraint is unknown. This precludes the use of this method of constraint for a general simulation package.

Determining a general constraint method is particularly difficult. It may be possible to generate a data base of procedures which represent the most common constraints. This is possible if the model has a limited number of constraints. All other constraint violations must be implemented using a more general but less efficient method.

4.1.6. The Representation of the Foot

The introduction of an analytic constraint actually reduced the computational cost with respect to the free system, as the number of degrees of freedom was reduced. However, this required the introduction of mathematically instantaneous impact and takeoff conditions. During ground contact in human running the force vector on the plantar surface of the foot is continually changing, usually progressing from heel to toe. This required a definition of the foot that would model this movement. The use of multiple descriptions of the foot was considered to be an inelegant solution. The inclusion of a foot of finite radius that employed a rolling constraint with respect to the ground appeared to be the most suitable method. Although an algorithm was not written to statistically determine the parameters of the foot, it is certainly possible to do that (Ju & Mansour, 1988). The parameters chosen were subjectively determined using an interactive graphics programme to give a reasonable approximation to the ankle kinematics.

4.2. The Interactive Graphics Programme

The discussion of the interactive graphics programme focuses on the practicality of modifying the force profiles. From a purely mechanistic point of view the modification of force profiles would seem to be the logical mathematical level at which to modify the movement. As will be presented in this section, it may not be the logical perceptual level. The levels referred to are related to the three levels of specifying and controlling behaviour presented by Zeltzer (1985):

1. The user specifies all of the information.
2. A special programming language interprets the user input and translates this information for the programme.
3. The programme implicitly responds to task level commands.

The approach taken in this theses was to study the simplest form of user interaction. The user was presented with the model and the ability to modify the force values. Forces represent the lowest level of input to the system as they are the causal mechanism for the movement. To simplify the task the user was presented with a sample movement and predetermined generalized force profiles. In using the interactive programme for the two segment model it was noted that a seemingly minor change in a single force value would disrupt the entire movement. A significant improvement in the ability to generate sensible movements by modifying the force profiles came as a result of representing the force profiles by a bezier polynomial. The user modifies the curve by modifying the bezier control points. The resulting force profiles were continuous and easy to modify.

The fundamental difficulty with this approach is the naivete of the user concerning the concept of a force. It is submitted that the idea of a force is not intuitively obvious to most people and that human perception of force is quite inaccurate as it can not be seen and the effect is manifested only over time. Furthermore, the modification of an individual force has a delayed effect upon the movement and modifies all of the kinematic variables in the system. It is conceivable that a user could develop the ability to produce desired movements of a two segment model using a heuristic approach. It is inconceivable to the author that a user could develop the mechanical insight necessary to produce a desired movement of a twelve segment model.

One of the problems concerns the number of degrees of freedom controlled by the user. The amount of control presented to the user must be restricted. A significant amount of information must be algorithmically specified allowing the user to modify only a few variables. Three or four may be sufficient. The constraint of many of the system variables can be accomplished by functional kinematic specification or by dynamic constraints. As mentioned in the introduction this is similar to the approaches advocated in the 1970's. This previous research had strict specification of the constrained variables. It is suggested here that a functional approach would have adaptive constraints. For example, the kinematics of the arms and the shoulders could be phasically linked to the movement of the legs. This requires the sensible assumption that the movement of the arms and shoulders are antisymmetric to the movement of the pelvis and legs. This approach has been used for the animation of human walking.

Recent research into the animation of human walking is considered particularly relevant (Bruderlin, 1988). Bruderlin's work simulated the basic dynamic characteristics of Raibert's Hopping Robot control strategy. The rest of the kinematics were produced by a set of kinematically defined relationships. Essentially, the movement of the body is phasically linked to the movement of the stance leg. This model is interesting because it provides an avenue for the production of novel human movements. A dynamic analysis can be performed on these simulations after the fact. In many ways this approach is analogous to the one presented in this thesis, since the user will modify the torque profile on a kinematic basis. The kinematic simulations may however produce movements which require torques that the human body is incapable of producing. This can be easily prevented in the dynamic simulations by constraining the maximal force. The inclusion of simple kinematic relationships reduces the internal control problem.

The recent work of Isaacs & Cohen(1988) has shown the simulations can combine elaborate kinematic and dynamic constraints into the dynamic simulation. The major drawback to the approach is that a poor specification of the constraints can lead to an over determined system of equations. A minor drawback is that the constraints must be twice differentiable and specified completely by the user. The constraints can therefore be algebraically complex.

The recent work of Hahn (1988) has shown the simulations based on the conservation of angular momentum and user specification of internal kinematics can include external forces and constraints. As discussed earlier, in this type of simulation the user modifies the internal kinematics of the system and dynamics of the body as a whole is calculated. The appeal of this approach is that most users have a better intuitive feel for kinematic modification as opposed to force modification. This type of dynamic simulation is also computationally more efficient than the dynamic simulation presented in this thesis. The major drawback of this approach is that it is possible to specify kinematics that are physiologically unreasonable.

The dynamic simulation presented in this thesis is the base level for future simulations. Regardless of the control strategies employed in the future, the task of these strategies is to provide the forces. The fundamental mechanics of this thesis is easily adaptable to the inclusion of a control strategy. The inclusion of kinematic constraints is manageable through the introduction of Lagrange multipliers.

Chapter 5
Conclusion

The conclusions focus on three fundamental issues:

1. appropriateness of the simulation,
2. functionality of the interactive simulation,
3. possible future research.

5.1. Appropriateness Of The Simulation

One of the difficult philosophical aspects of this dissertation is identifying the criteria for determining if the simulation has been successful. The following section will discuss the issue of validation of the model. It is submitted that the appropriateness of a simulation should be judged according to the criteria of validity and utility. The first criterion for the judgement of the success of a simulation is the validation of the model and the simulated movement. One obvious difficulty in the task of validation is determining the necessary degree of corroboration between the simulated movement of the model and the human movement. It is recognised that neither a theoretical representation nor the measurement of a natural system can be accurate. Absolute validation is impossible and therefore, a model will be invalidated only when it is clearly unrepresentative of the natural movement, or when a better model is developed. It is suggested that the kinematics presented in this work adequately described the running movement, and as the simulation replicated these kinematics, the model is conditionally valid.

The second criterion for judging the success of the simulation relates to its utility. The fundamental issue to be determined is the predictive capability of the model. It is apparent that modification of the driving forces produces a modification of the movement. The simulation is therefore capable of making predictions of novel performance.

In light of the facts that absolute validation is impossible, and that functional simplicity is necessary for general use, it is submitted that the simulation is appropriate.

5.2. Possible Future Directions

The genesis of the research was a suggestion that the development of an interactive computer simulation would be beneficial for the modification and understanding of human running. The objective was to develop a computer simulation that could be used by anyone. This entailed the development of a model of minimal complexity with intuitively obvious control strategies. Within the field of research into the simulation of human movement, there are a number of researchers who have espoused the elitist view that simulations should be used only by a select few individuals with exceptional mathematical skills. As well, it is asserted that these "qualified" researchers must collaborate with professional computer programmers and engineers if they hope to produce effective simulations (Hatze, 1983b, Vaughan, 1984). They also suggest that the researcher must have significant biomechanical insight. This researcher contends that such a restrictive view of simulation is shortsighted. The potential advantages of simulating human motion should be made available to all, and should not be put into the "protective custody" of a few individuals who feel that they are uniquely qualified to handle such "sensitive" material.

The objective of the research is to produce a simulation tool that can be used by both novel and experienced users. Therefore, the focus of future research should be on the development of a better computer interface.

Four possible directions for further research will be briefly discussed. Firstly, the user of the interactive programme is currently restricted to changing the input state of the system and modifying the torque profiles of the default running stride or producing the entire movement. A major improvement in the model would be achieved by the inclusion of a data base of various kinematic patterns. This would allow the user to be aware of various force profiles and their relationship to different movements. This would make it easier to determine the necessary modifications to achieve a desired result.

Secondly, the system may be modified to incorporate a mechanical model of human muscle. This would allow the user to obtain a more accurate estimate of how muscles produce movement. The major attributes of such a muscle model would be a contractile component and a series elastic component. It is likely that muscle elasticity has a significant effect on the kinematics. It is possible that the force profile of the contractile component of muscle is far less complex than the external joint torque profile. Modelling the elasticity may provide a better understanding of human motor control. The muscle models would likely represent functional muscle groups rather than individual muscles. The logical muscle groups for the study of running would be the flexors and extensors of the hips, knees, ankles and metatarsals.

Thirdly, the system may be modified to contain default kinematic relationships that would minimise the number of variables that the user would be required to specify. Bruderlin (1988) has developed a simulation of walking that minimally requires a single input parameter from the user, such as walking speed, and the programme calculates an appropriate movement based on internal default relationships.

Fourthly, a control strategy could be introduced. This possibly would have the same effect as would the inclusion of default kinematic relationships. The result would be that it would reduce the amount of information required by the user. The two processes are functionally analogous but the mathematical task is quite different.

One of the intentions in carrying out this project was to provide the biomechanics research community with an interactive simulation of human running. It should be emphasized that the author assumes that human conceptual understanding is sufficiently developed so that it is desirable to allow intuition to play a major role in modifying a movement. An understanding of how a human user manipulates control variables to produce a movement may lead to the development of an expert system capable of the same task. It is suggested that there is currently insufficient information available to mimic human perception and learning by a mathematical analogy. Considerable attention has been given to the question of how to control the inputs to the model and in particular, how to affect changes in its performance. It was decided that the goal of the research was to produce a tool for understanding human running. The emphasis is clearly on human running, and the ability to modify a simulated performance using an interactive computer graphics model.

In conclusion, an interactive computer graphics simulation of human running was successfully developed. An understanding of computers or mathematics is not a prerequisite for the user of this programme. This programme represents a significant step towards providing the use of simulation techniques to users with a broad range of skills. It is also acknowledged that the choice of force as the interactive variable in the simulation may not be the most appropriate interactive variable. The use of force is a perceptually difficult task for most people. The average user has a much better intuitive knowledge of kinematic information. Future interactive simulations must address either the problem of presenting the forces in a more appropriate manner or by presenting the user with the ability to modify other information.

It is obvious that the programme was not only developed for the naive user and as such the requirement of specifying force information is not always a problem. The researcher may be introducing the force profiles using some external algorithm and as such is not intimidated by the force profiles in any way.

References

Ariel, G. (1980)
Can Computers Win Gold Medals?
New Scientist, 87, 282-284

Armstrong, W.W., Green, M., Lake, R. (1987)
Near-Real-Time Control of Human Figure Models.
IEEE Computer Graphics and Applications, 7, 52-61

Bauer, W.L. (1983)
Mathematical Modeling and Optimization and their Influence on Sports Movements - Possibilities and Limitations.
In Baumann, W. (Ed) *Biomechanics and Performance in Sport*
Bundesinstitut fur Sportwissenschaft

Bourassa, P., Morel, Y. (1983)
A Mathematical Model for Computer Simulation of Human Running.
Paper presented at the International Congress of Biomechanics IX,
Meeting of the International Society of Biomechanics

Boysen, J.P., Francis, P.R., Thomas, R.A. (1977)
Interactive Computer Graphics in the Study of Human Body Planar Motion Under Free Fall Conditions.
Journal of Biomechanics, 10, 783-787

Bruderlin, A. (1988)
Goal-Directed, Dynamic Animation of Bipedal Locomotion.
Unpublished Master's Thesis, Simon Fraser University

Burden, R.L., Faires, J.D., Reynolds, A.C. (1981)
Numerical Analysis,
Boston, Massachusetts, Prindle, Weber and Schmidt

Chao, E.Y., Rim, K. (1973)
Application of Optimization Principles in Determining the Applied Moments in Human Leg Joints During Gait.
Journal of Biomechanics, 6(1), 497-510

Chapman, A.E., Medhurst, C.W. (1981)
Cyclographic Identification of Fatigue in Sprinting.
Journal of Human Movement Studies, 7, 255-272

Chapman, A.E., Caldwell, G.E.C. (1983)
Factors Determining Changes in Lower Limb Energy During Swing in Sprinting.
Journal of Biomechanics ,16, 69-77

Chapman, A.E., Caldwell, G.E.C. (1983)
Kinetic Limitations of Maximal Sprinting Speed.
Journal of Biomechanics ,16, 79-83

Chapman, A.E., Lonergan, R.L., Caldwell, G.E.C. (1984)
Kinetic Sources of Lower-Limb Angular Displacement in the Recovery Phase of Sprinting.
Medicine and Science in Sports and Exercise ,16(4), 382-388

Chapman, A.E., Caldwell, G.E.C., Herring, R.N., Lonergan, R.L., Selbie, W.S. (1985)
Mechanical Energy and the Preferred Running Style.
Paper presented at the International Congress of Biomechanics X, Meeting of the International Society of Biomechanics

Chow, C.K., Jacobsen, D.H. (1971)
Studies of Human Locomotion via Optimal Programming.
Mathematical Biosciences,10, 239-306

Dapena, J. (1981)
Simulation of Human Airborne Movements.
Journal of Biomechanics ,114(1), 81-89

Gallenstein, J., Huston, R.L. (1973)
Analysis of Swimming Motions.
Human Factors ,15, 91-98

Gladwell, I., Thomas, R.M. (1981)
Damping and Phase Analysis for Some Methods for Solving Second Order Ordinary Differential Equations.
Numerical Analysis Report ,(59)

Hahn, J.K. (1988)
Realistic Animation of Rigid Bodies.
Computer Graphics, 22(4), 299,308

Hatze, H., Venter, A. (1981)

Practical Activation and Retention of Locomotion Constraints in Neuromusculoskeletal Control System Models.

Journal of Biomechanics ,14(12), 873-877

Hatze, H. (1981)

A Comprehensive Model for Human Motion Simulation and its Applications to the Take-Off Phase of the Long Jump.

Journal of Biomechanics ,14(3), 135-142

Hatze, H. (1983)

Myocybernetic Optimization of Sports Motions: Model, Methods for Parameter Identification, Practical Implementation.

In Baumann, W. (Ed) *Biomechanics and Performance in Sport*

Bundesinstitut fur Sportwissenschaft

Hatze, H. (1984)

Quantitative Analysis, Synthesis and Optimization of Human Motion.

Human Movement Studies ,3, 5-25

Hatze, H. (1986)

Motion Variability--its Definition, Quantification and Origin.

Journal of Motor Behaviour ,18(1), 5-16

Hill, A.V. (1927)

The Air-Resistance to a Runner.

Proceedings of the Royal Society , B102, 380-385

Hindmarsh, A.C. (1980)

LDOSE and LSODI, Two New Initial Value Ordinary Differential Equation Solvers.

ACM-SIGNUM Newsletter 15(4), 10-11

Hollerbach, J.M. (1980)

A Recursive Lagrangian Formulation of Manipulator Dynamics.

IEEE Transactions on Systems, Man and Cybernetic ,SMC-10(11)

Isaacs, P.M., Cohen, F.C. (1988)

Mixed Methods For Complex Kinematic Constraints in Dynamic Figure Animation.

The Visual Computer ,4, 296-305

Ju, M.S., Mansour, J.M. (1988)
Simulation of the Double Support Phase of Human Gait.
Journal of Biomechanical Engineering ,110, 223-229

Kane, R.T., Levinson, D.A. (1983)
The Use of Kane's Dynamical Equations in Robotics.
The International Journal of Robotics Research ,2(3), 3-21

Keller, J.B.. (1974)
Optimal Velocity in a Race.
American Mathematics Monthly,Vol 5

Lonergan, R.L. (1988)
A Biomechanical Rational for Preferred Running Style.
Unpublished Master's Thesis, Simon Fraser University

The Math Lab Group, Laboratory for Computer Science
MACSYMA Reference Manual,
MIT, Cambridge, MA

Marshall, R.N., Jensen, R.K. and Wood, G.A. (1985)
A General Newtonian Simulation of an N-Segment Open Chain Model.
Journal of Biomechanics ,16(1), 359-369

Marshall, R.N., Wood, G.A., Jennings, L.S. (1985)
Performance Criteria in Normal Human Locomotion.
Presentation at the American Society of Biomechanics

McGeer, T. (1989)
Passive Dynamic Walking.
Journal of Robotics, In press

McMahon, T.A. (1984)
Mechanics of Locomotion.
The International Journal of Robotics Research ,3(2), 4-28

Moore, M., Wilhelms, J. (1988)
Collision Detection and Response for Computer Animation.
Presentation at ACM Siggraph 88

Nelson, W.L. (1983)
Physical Principles for Economies of Skilled Movements.
Biological Cybernetics , (46), 135-147

Pandy, M.G., Berme, N. (1988)

A Numerical Method for Simulating the Dynamics of Human Walking.

Journal of Biomechanics ,21, 1043-1051

Pandy, M.G., Berme, N. (1988)

Synthesis of Human Walking: A Planar Model for Single Support.

Journal of Biomechanics ,21, 1053-1060

Panjabi, M. (1979)

Validation of Mathematical Models.

Journal of Biomechanics ,12, 238

Passerello, C.E., Huston, R.L. (1971)

Human Attitude Control.

Journal of Biomechanics ,4, 369-378

Phillips, S.J., Roberts, E.M., Huang, T.C. (1983)

Quantification of Intersegmental Reactions During Rapid Swing Motion.

Journal of Biomechanics ,16(6), 411-417

Raibert, M.H. (1984)

Hopping in Legged Systems-Modeling and Simulation for the Two-Dimensional One-Legged Case.

IEEE Transactions on Systems, Man and Cybernetics ,SMC-14(3), 451-463

Raibert, M.H. (1986)

Legged Robots that Balance

The MIT Press, Cambridge, Massachusetts

Ramey, M.R. (1973)

A Simulation of the Running Long Jump.

in J.L.Bleuskin (Ed.) *Mechanics and Sports*, N.Y. ASME

Vaughan, C.L.. (1984)

Computer Simulation of Human Motion in Sports Biomechanics.

Exercise and Sports Sciences Reviews , D.C. Heath and Company

Vaughan, C.L. (1985)

Simulation of a Sprinter Part I. Development of a Model.

International Journal of Biomedical Computing , 14,65-74

Walker, H.S., Kirmsmer, P.G.. (1973)

Computer Modeling of Pole Vaulting.

in J.L.Bleuskin (Ed.) *Mechanics and Sports*, N.Y. ASME

Ward-Smith, A.J. (1985)
A Mathematical Theory of Running, Based on the First Law of Thermodynamics.
Journal of Biomechanics , 18, 337-349

Wilhelms, J. (1987)
Using Dynamic Analysis for Realistic Animation for Articulated Bodies.
IEEE Computer Graphics and Applications, 7, 12-27

Williams, K.R., Cavanagh, P.R. (1983)
A Model for the Calculation of Mechanical Power During Distance Running
Journal of Biomechanics , 16, 115-128

Winter, D.A. (1979)
Biomechanics of Human Movement
John Wiley and Sons, New York

Yeadon, M.R. (1986)
The Biomechanics of Twisting Somersaults.
Presented at a meeting of the Canadian and American Society of Biomechanics

Zeltzer, D. (1985)
Towards an Integrated View of 3-D Computer Animation.
The Visual Computer , 1(4), 249-259

Zheng, Y.F., Hemami, H. (1984)
Impact Effects of Biped Contact with the Environment.
IEEE Transactions on Systems, Man and Cybernetics, Vol SMC-14(3)

Appendix A

Artificial gain in energy created by an explicit integrator

This appendix considers the artificial gain in energy introduced by using an explicit numerical integration method to solve a periodic system of equations. For simplicity this section will consider the application of Euler's method to Simple Harmonic Motion.

Consider the simplified oscillatory problem

$$\frac{d^2y}{dt^2} + w_0^2\, y = 0$$

Transform this second order system into two first order systems

$$(1) \qquad u_1 = \frac{dy}{dt}$$

$$(2) \qquad u_2 = \frac{du_1}{dt} = -w_0^2\, y$$

Using Euler's method discretize these equations

$$(3) \qquad y(t+Dt) - y(t) = u_1\, Dt$$

$$(4) \qquad u_1(t+Dt) - u_1(t) = -w_0^2\, y\, Dt$$

Consider solutions to the discretized equations of (1) and (2)

$$y = A\, e^{igt}$$

$$u_1 = A\, i\, g\, e^{igt}$$

As this problem is purely oscillatory we can make the assumption that only the Real part of g be considered

$$u_1 = A\, i\, w_0\, e^{igt}$$

Substituting into equations (3) and (4)

$$A\, e^{ig\,(t+Dt)} - A\, e^{igt} = A\, i\, w_0\, e^{igt}\, Dt$$

$$iw_0(A\, e^{ig\,(t+Dt)} - A\, e^{igt}) = A\, i\, w_0\, e^{igt}\, Dt$$

both equations reduce to

$$e^{igDt} - 1 \qquad = i\, w_0\, Dt$$

Rewrite this equation as

$$i\, g\, Dt \;= \ln(\, 1 + i\, w_0\, Dt\,)$$

$$i\, g\, Dt \;= i\, w_0\, Dt - \frac{1}{2}(i\, w_0\, Dt)^2 + \ldots$$

$$g\, Dt \;= w_0\, Dt - \frac{1}{2}\, i\, w^2{}_0\, Dt^2 + \ldots$$

The imaginary term (i.e. second term of the expansion) means $y = Re(A\, e^{igt})$ has exponential growth.

These conclusions are representative of all explicit integration techniques (e.g. Runge-Kutta methods) and the Predictor part of Predictor-Corrector algorithms (e.g. Adams methods). Care must be taken if purely imaginary eigenvalues must be considered.

Appendix B

Solution To The Boundary Value Problem

$A(y)\ddot{y} = B(y, \dot{y}) + F$

given : $y(0)$, $\dot{y}(0)$, $y(t)$

unknown : F

The basic approach taken to solving the boundary value problem was to employ a shooting method. This technique involves first, estimating the required forces, and then integrating the equations. If the final conditions are satisfied the task is finished. Otherwise, the forces must be modified and the integrations performed again.

Two methods were used to modify the forces. The first method is a quasi-Newtonian method (Burden, Faires and Reynolds 1981). This method shows a fast convergence, but is extremely sensitive to the starting conditions. It is necessary that the initial guess be very close to the correct solution. In fact, the stability region is too small for this method to be of practical use on its own.

given : $F(x) = 0$. Solve for x.

$$J(x) = \begin{pmatrix} \frac{\partial f_0}{\partial x_0} & \frac{\partial f_0}{\partial x_1} & \cdots & \frac{\partial f_0}{\partial x_n} \\ \frac{\partial f_1}{\partial x_0} & \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f_n}{\partial x_0} & \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}$$

define the vector : $G(x) = x - J^{-1}(x)\,F(x)$

the new value of x is defined by : $x^{(k)} = G(x^{(k-1)})$

It is not always possible to adequately define : $J(x)^{-1}$

Instead the following two step procedure is used.

$$J(x^{(k)})\, y = -F(x^{(k)}),$$

$$x^{(k+1)} = x^{(k)} + y$$

Newtons method can only be used if there is an analytic expression of the Jacobian $J(x)$. The equations of motion are non-linear and it is necessary to use a quasi-Newton method which uses an approximate Jacobian. The Jacobian is approximated by integrating the same time step several times, each time the force is changed by some amount.

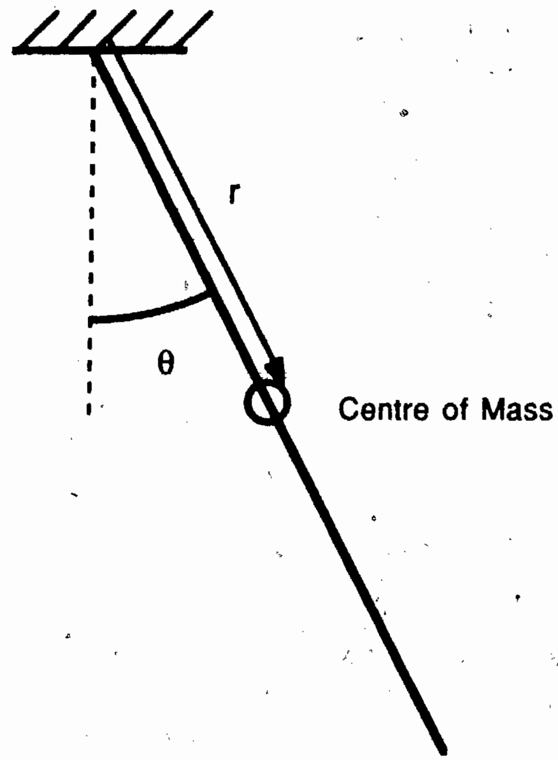
For example:

$$\frac{\partial f_j}{\partial x_k}(x^{(i)}) = \frac{f_j(x^{(i)}+h) - f_j(x^{(i)})}{h}$$

In theory the value of h should be small since an approximation to the derivative is desired. It was discovered that the convergence was better for a large value of h. (In this thesis of value of h=10.0 was used.)

Appendix C

Compound Pendulum



Equation of Motion

$$F_q = (I+mr^2)\ddot{q}+mgrsinq$$

Specification of the subroutines for the LSODI integrator

adda:
$a[1][1] \mathrel{+}= I+mr^2$

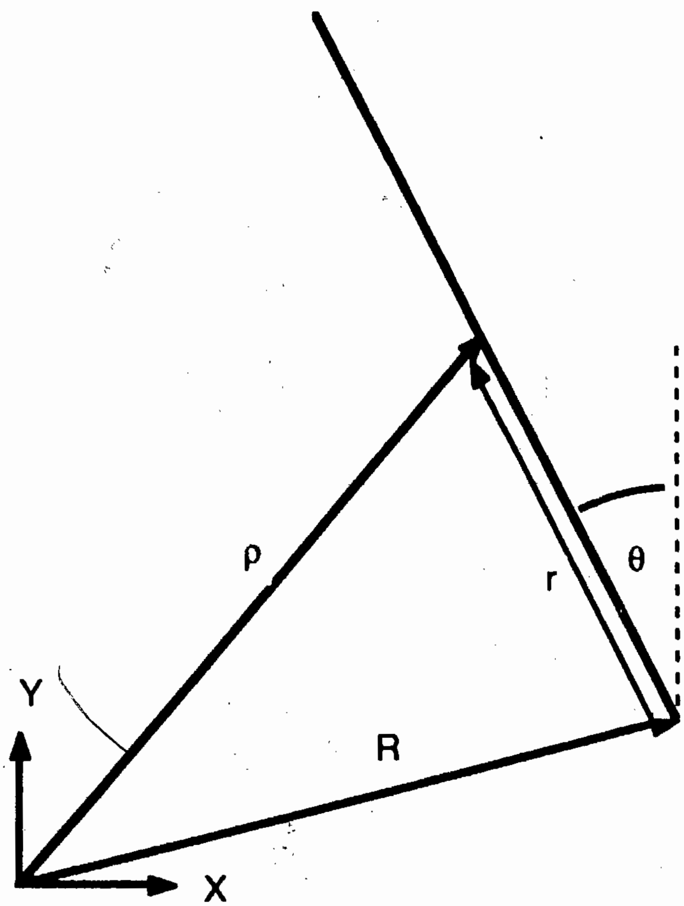
res :
$p[1]= F_q - (I+mr^2)s[5] - mgrsinq$

jac :
$b[1][0]= -mrgcosq$

Appendix D

One segment planar rigid body

This appendix describes the equations of motion for a one segment rigid body. Both the constrained (one degree of freedom) and unconstrained (three degrees of freedom) states are specified. In the representation for the LSODI integrator : y is a vector of the generalized coordinates after expansion to a first order system. s is a vector defining the approximate derivative to y.



r : Proximal distance to the centre of mass
m: Mass of the segment
I : mk^2

Unconstrained (three degrees of freedom):

$$F_x = m\ddot{x} - mr(\ddot{q}\cos q - \dot{q}^2 \sin q)$$

$$F_y = m(\ddot{y}+g) - mr(\ddot{q}\sin q + \dot{q}_1{}^2 \cos q)$$

$$F_q = (I+mr^2)\ddot{q} - mr(\ddot{x}\cos q + (\ddot{y}+g)\sin q)$$

Constrained (one degree of freedom)

$$F_q = (I+mr^2)\ddot{q} - mrg\sin q$$

Impact Conditions

$$H_A{}^- = (I+mr^2)\dot{q}^- - mr(\dot{x}^-\cos q + \dot{y}^-\sin q)$$

$$H_A{}^+ = (I+mr^2)\dot{q}^+$$

At Impact

$$H_A{}^+ = H_A{}^-$$

$$\dot{q}^+ = \dot{q}^- - \frac{mr}{I+mr^2}(\dot{x}^-\cos q + \dot{y}^-\sin q)$$

Specification of the subroutines for the LSODI integrator

Constrained State

adda:

a[1][1] += $I+mr^2$

res :

p[1]= F_q - $(I+mr^2)$s[5] + mgrsinq

jac :

b[1][0]= mgrcosq

Unconstrained State

adda:

a[1][1] += m
a[1][5] -= mrcosq

a[3][3] += m
a[3][5] -= mrsinq

a[5][1] -= mrcosq
a[5][3] -= mrsinq
a[5][5] += I+mr^2

res :

p[1]= F_x - ms[1] + mr(s[5]cosq-y[5]^{2}sinq)

p[3]= F_y - m(s[3]+g) + mr(s[5]sinq+y[5]^{2}cosq)

p[5]= F_q - (I+mr^2)s[5] + mr(s[1]cosq+(s[3]+g)sinq)

jac :

b[1][4]= -mr(s[5]sinq+y[5]^{2}cosq)
b[1][5]= -2.0 mr y[5]sinq

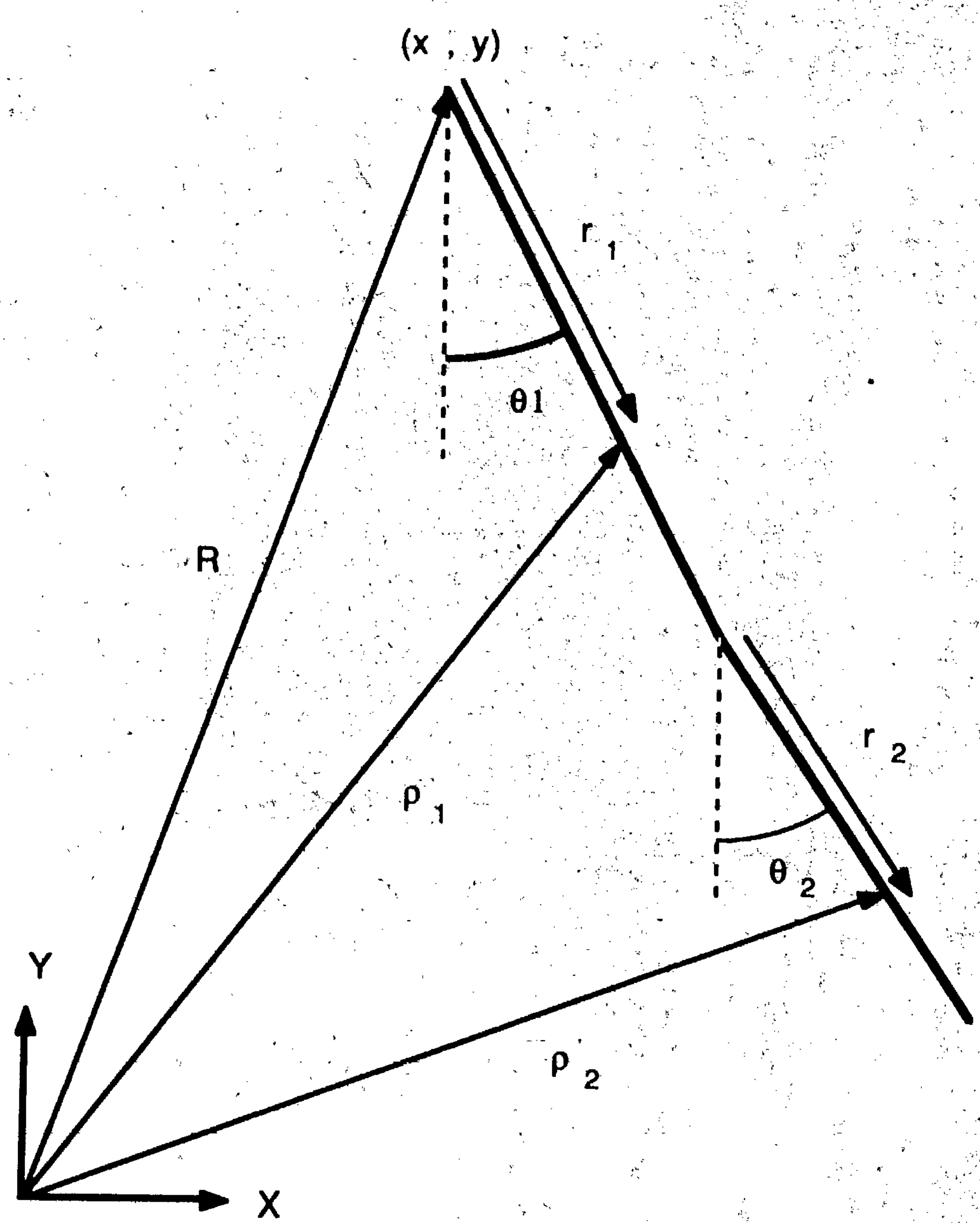
b[3][4]= mr(s[5]cosq-y[5]^{2}sinq)
b[3][5]= 2.0 mr y[5]cosq

b[5][4]= -mr(s[1]sinq-(s[3]+g)cosq)

Appendix E

Two Segment Planar Rigid Body

This appendix describes the equations of motion for a two segment rigid body. Both the constrained (two degree of freedom) and unconstrained (four degrees of freedom) states are specified. In the representation for the LSODI integrator : y is a vector of the generalized coordinates after expansion to a first order system: s is a vector defining the approximate derivative to y.



Equations of Motion:

$$F_x = (m_1+m_2)\ddot{x} + (m_1r_1+m_2l_1)(\ddot{q}_1\cos q_1-\dot{q}_1^2\sin q_1)$$
$$+m_2r_2(\ddot{q}_2\cos q_2-\dot{q}_2^2\sin q_2)$$

$$F_y = (m_1+m_2)(\ddot{y}+g) + (m_1r_1+m_2l_1)(\ddot{q}_1\sin q_1+\dot{q}_1^2\cos q_1)$$
$$+m_2r_2(\ddot{q}_2\sin q_2+\dot{q}_2^2\cos q_2)$$

$$F_{q1} = (I_1+m_1r_1^2+m_2l_1^2)\ddot{q}_1+ (m_1r_1+m_2l_1)(\ddot{x}\cos q_1+(\ddot{y}+g)\sin q_1)$$
$$+m_2r_2l_1(\ddot{q}_2\cos(q_2-q_1)-\dot{q}_2^2\sin(q_2-q_1))$$

$$F_{q2} = (I_2+m_2r_2^2)\ddot{q}_2+ m_2r_2(\ddot{x}\cos q_2+(\ddot{y}+g)\sin q_2)$$
$$+m_2r_2l_1(\ddot{q}_1\cos(q_2-q_1)+\dot{q}_1^2\sin(q_2-q_1))$$

Specification of the subroutines for the LSODI integrator

adda:

a[1][1] += m_1+m_2
a[1][5] += $(m_1r_1+m_2l_1)\cos q_1$
a[1][7] += $m_2r_2\cos q_2$

a[3][3] += m_1+m_2
a[3][5] += $(m_1r_1+m_2l_1)\sin q_1$
a[3][7] += $m_2r_2\sin q_2$

a[5][1] += $(m_1r_1+m_2l_1)\cos q_1$
a[5][3] += $(m_1r_1+m_2l_1)\sin q_1$
a[5][5] += $I_1+m_1r_1^2+m_2l_1^2$
a[5][7] += $m_2r_2l_1\cos(q_2-q_1)$

a[7][1] += $m_2r_2\cos q_2$
a[7][3] += $m_2r_2\sin q_2$
a[7][5] += $m_2r_2l_1\cos(q_2-q_1)$
a[7][7] += $I_2+m_2r_2^2$

res :

p[1]= $F_x - (m_1+m_2)$s[1] $- (m_1r_1+m_2l_1)($s[5]$\cos q_1-$y[5]$^2\sin q_1)$
$- m_2r_2($s[7]$\cos q_2-$y[7]$^2\sin q_2)$

$$p[3]= F_y - (m_1+ m_2)(s[3]+g) - (m_1r_1+m_2l_1)(s[5]\sin q_1+y[5]^2\cos q_1) - m_2r_2(s[7]\sin q_2+y[7]^2\cos q_2)$$

$$p[5]= F_{q1} - (I_1+m_1r_1{}^2+m_2l_1{}^2)s[5] - (m_1r_1+m_2l_1)(s[1]\cos q_1+(s[3]+g)\sin q_1) - m_2r_2l_1(s[7]\cos(q_2-q_1)-y[7]^2\sin(q_2-q_1))$$

$$p[7]= F_{q2} - (I_2+m_2r_2{}^2)s[7] - m_2r_2(s[1]\cos q_2+(s[3]+g)\sin q_2) - m_2r_2l_1(s[5]\cos(q_2-q_1)+y[5]^2\sin(q_2-q_1))$$

jac :

$b[1][4]= (m_1r_1+m_2l_1)(s[5]\sin q_1+y[5]^2\cos q_1)$
$b[1][5]= 2.0\,(m_1r_1+m_2l_1)y[5]\sin q_1$
$b[1][6]= m_2r_2(s[7]\sin q_2+y[7]^2\cos q_2)$
$b[1][7]= 2.0\,m_2r_2y[7]\sin q_2$

$b[3][4]= -(m_1r_1+m_2l_1)(s[5]\cos q_1-y[5]^2\sin q_1)$
$b[3][5]= -2.0\,(m_1r_1+m_2l_1)y[5]\cos q_1$
$b[3][6]= -m_2r_2(s[7]\cos q_2-y[7]^2\sin q_2)$
$b[3][7]= -2.0\,m_2r_2y[7]\cos q_2$

$b[5][4]= (m_1r_1+m_2l_1)(s[1]\sin q_1-(s[3]+g)\cos q_1) - m_2r_2l_1(s[7]\sin(q_2-q_1)+y[7]^2\cos(q_2-q_1))$
$b[5][6]= m_2r_2l_1(s[7]\sin(q_2-q_1)+y[7]^2\cos(q_2-q_1))$
$b[5][7]= 2.0\,m_2r_2l_1y[7]\sin(q_2-q_1)$

$b[7][4]= -m_2r_2l_1(s[5]\sin(q_2-q_1)-y[5]^2\cos(q_2-q_1))$
$b[7][5]= -2.0\,m_2r_2l_1y[5]\sin(q_2-q_1)$
$b[7][6]= m_2r_2(s[1]\sin q_2-(s[3]+g)\cos q_2) - b[7][4]$

Appendix F

Recursive Formulation of the Equations of Motion

The equations of motion for planar objects can be readily adapted to a general algorithm formulation. The following algorithm is not intended to be the most efficient algorithm available but does work. The algorithm is specified for the LSODI numerical integrator. Modification of the algorithm to another numerical integrator is a straightforward task. This section describes a scheme for specifying the equations of a planar n-segment linked rigid body(n<19). The reference point is specified at the proximal end of the chain. The recursion is expressed in pseudo-code very similar to c-code. The equations have been written so the R and L positive defines the segment directed vertically up.

Define the number of segments

```
NSEG                    /* Number of rigid segments          */
NEQ = 2*(NSEG+2)        /* Number of differential equations  */
```

Define the inertial variables.

```
I[NSEG]                 /* Moment of Inertia                          */
M[NSEG]                 /* Mass                                       */
R[NSEG]                 /* Proximal Distance to Centre of Mass */
L[NSEG]                 /* Length of Segment                          */
```

Define the algorithm inertial parameters

```
for(i=0;i<NSEG;i++) {
        mhl[i]= 0;
        for(j=i;j<NSEG;j++)
                mhl[i] += M[j];
        }

for(i=0;i<NSEG;i++) {
        mhl[20+i]= M[i]*R[i] + mhl[i+1]*L[i];
        mhl[40+i]= I[i] + M[i]*R[i]*R[i] + mhl[i+1]*L[i]*L[i];
        }
```

Define the trigonometric variables

```
Cos_[NSEG], Sin_[NSEG], Cos[NSEG][NSEG], Sin[NSEG][NSEG];

for(i=0;i<NSEG;i++) {
        Cos_[i]= cos(y[2*(i+2)]);
        Sin_[i]= sin(y[2*(i+2)]);
        if(i) {
                for(j=0;j<i;j++) {
                        Cos[i][j] = Cos_[i]*Cos_[j]+Sin_[i]*Sin_[j];
                        Sin[i][j] = Sin_[i]*Cos_[j]-Cos_[i]*Sin_[j];
                        }
                }
        }
```

Define the Inertia Matrix

```
for(i=0; i<NEQ; i += 2)
        adda[i][i] += 1.0;

adda[1][1] += mhl[0];
adda[3][3] += mhl[0];

for(i=0; i<NSEG; i++) {
        adda[2*i+5][1] -= mhl[20+i]*Cos_[i];
        adda[1][2*i+5] -= mhl[20+i]*Cos_[i];

        adda[2*i+5][3] -= mhl[20+i]*Sin_[i];
        adda[3][2*i+5] -= mhl[20+i]*Sin_[i];

        adda[2*i+5][2*i+5] += mhl[40+i];

        for(j=0; j<i; j++)
                adda[2*j+5][2*i+5] += mhl[20+i]*L[j]*Cos[i][j];
                adda[2*i+5][2*j+5] += mhl[20+i]*L[j]*Cos[i][j];
                }
        }
```

Define the Residual Vector

```
for(i=0;i<NEQ;i += 2)
        b[i]= y[i+1];

b[1] = force[0] - mhl[0]*s[1];
b[3] = force[1] - mhl[0]*(s[3]+g);

for(i=0;i<NSEG;i++) {
        b[1] += mhl[20+i]*( s[2*i+5]*Cos_[i]-y[2*i+5]*y[2*i+5]*Sin_[i] );
        b[3] += mhl[20+i]*( s[2*i+5]*Sin_[i]+y[2*i+5]*y[2*i+5]*Cos_[i] );

        b[2*i+5] = force[i+2] - mhl[40+i]*s[2*i+5]
                        + mhl[20+i]*(s[1]*Cos_[i]+(s[3]+g)*Sin_[i]);

        for(j=0;j<i;j++)
                b[2*i+5] -= mhl[20+i]*L[j]*
                                ( s[2*j+5]*Cos[i][j]+y[2*j+5]*y[2*j+5]*Sin[i][j] )

        for(j=i+1;j<NSEG;j++)
                b[2*i+5] -= mhl[20+j]*L[i]*
                                ( s[2*j+5]*Cos[j][i]-y[2*j+5]*y[2*j+5]*Sin[j][i] );
        }
```

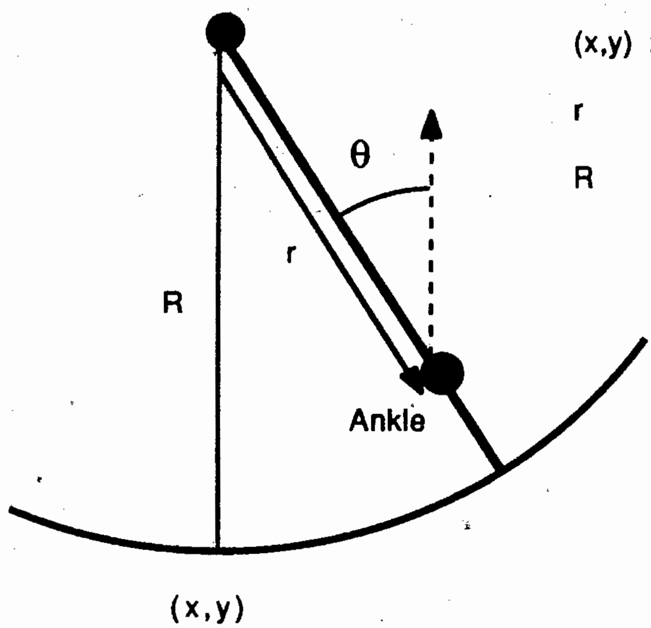
Define the Jacobian Matrix

```
for(i=0;i<NSEG;i++) {
        j[1][2*i+4]= -mhl[20+i]*(s[2*i+5]*Sin_[i]+y[2*i+5]*y[2*i+5]*Cos_[i]);
        j[1][2*i+5]= -2*mhl[20+i]*y[2*i+5]*Sin_[i];

        j[3][2*i+4]= mhl[20+i]*(s[2*i+5]*Cos_[i]-y[2*i+5]*y[2*i+5]*Sin_[i]);
        j[3][2*i+5]= 2*mhl[20+i]*y[2*i+5]*Cos_[i];

        for(j=0;j<i;j++) {
                j[2*i+5][2*j+4] = -mhl[20+i]*L[j]*
                                (s[2*j+5]*Sin[i][j]+y[2*j+5]*y[2*j+5]*Cos[i][j]);
                j[2*i+5][2*j+5] = -2*mhl[20+i]*L[j]*y[2*j+5]*Sin[i][j];mh
                }
        for(j=i+1;j<NSEG;j++) {
                j[2*i+5][2*j+4] = mhl[20+j]*L[i]*
                        (s[2*j+5]*Sin[j][i]+y[2*j+5]*y[2*j+5]*Cos[j][i]);
                j[2*i+5][2*j+5] = 2*mhl[20+j]*L[i]*y[2*j+5]*Sin[j][i];
                }

        j[2*i+5][2*i+4] = -mhl[20+i]*(s[1]*Sin_[i]-(s[3]+g)*Cos_[i]);

        for(j=0,j<i;j++)
                j[2*i+5][2*i+4] -= j[2*i+5][2*j+4];
}       for(j=i+1,j<NSEG;j++)
                j[2*i+5][2*i+4] -= j[2*i+5][2*j+4];
        }
```

Appendix G

One Segment Object with Curved Foot

This appendix contains an example of the method of combining Newtonian and Lagrangian Dynamics. The rolling constraint used for the foot is defined with Newtonian dynamics. The distal segment is defined using Lagrangian dynamics. This method was employed in the thesis for the final running model. For convenience the centre of mass of the foot was chosen to be at the ankle joint. This approximation is considered reasonable since the mass of the foot is so small.



(x,y) : instantaneous cen tre of rotation
r : distance to cen tre of mass from arc centre
R : radius of arc

$(x,y) = (x_c + R(q-q_c), y_c)$
(x_c, y_c) is the contact point at impact :

$r = (r\sin q, R-r\cos q)$

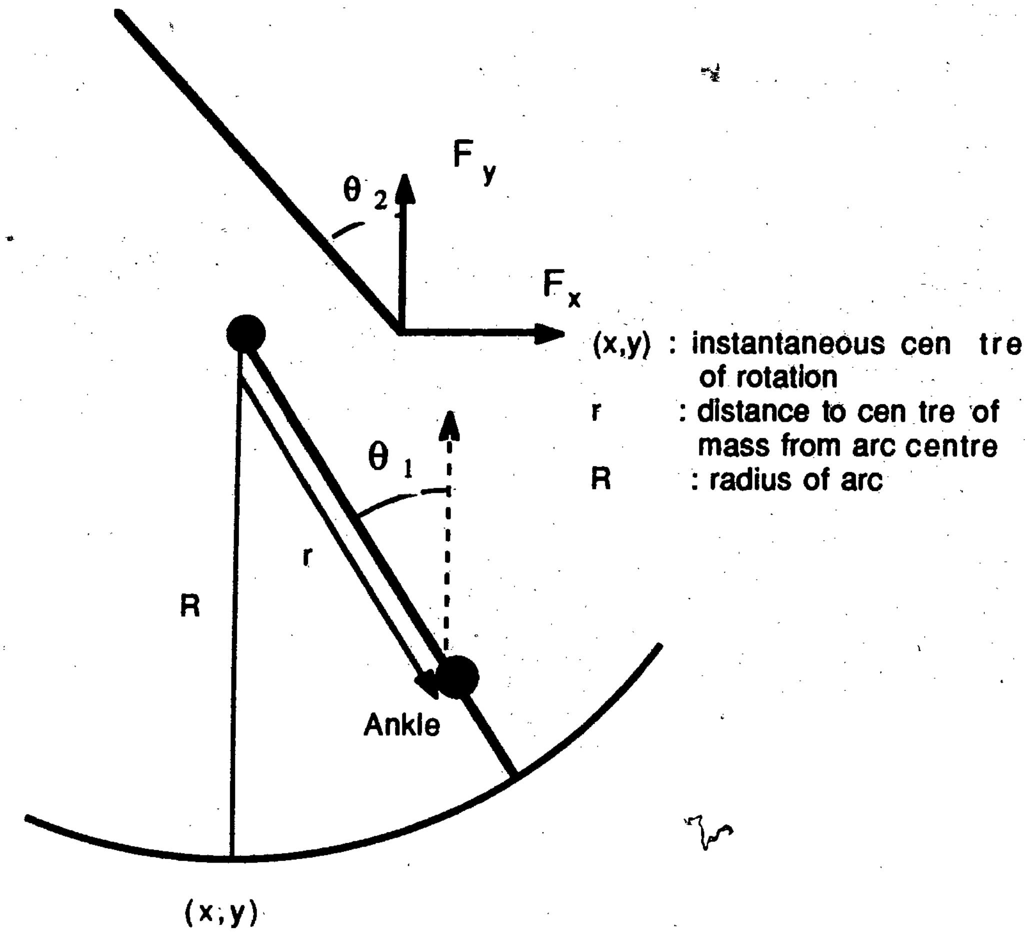
$v = w \times r = -\dot{q}\,(R-r\cos q, -r\sin q)$

$a = = a \times r + w \times w \times r = -\ddot{q}(R-r\cos q, -r\sin q) - \dot{q}^2(r\sin q, R-r\cos q)$

$(F_x, F_y) = ma = -m(\ddot{q}(R-r\cos q, -r\sin q) - \dot{q}^2(r\sin q, R-r\cos q)) + m(0,g)$

$F_q + F_x\,(R-r\cos q) - F_y\, r\sin q = I\ddot{q}$

$F_q = [I + m(R^2 - 2rR\cos q + r^2)]\,\ddot{q} + mgr\sin q$



The equations are combined through the joint force at the ankle. The following section presents the method of combination.

Equations of motion for the two models

Foot

$$F_q = [I+m(R^2-2rR\cos q+r^2)]\ddot{q} + mgr\sin q$$

Segment

$$F_x = m_2\ddot{x} - m_2r_2(\ddot{q}_2\cos q_2-\dot{q}_2^2\sin q_2)$$

$$F_y = m_2(\ddot{y}+g) - m_2r_2(\ddot{q}_2\sin q_2+\dot{q}_2^2\cos q_2)$$

$$F_{q2} = (I_2+m_2r_2^2)\ddot{q}_2-m_2r_2(\ddot{x}\cos q_2+(\ddot{y}+g)\sin q_2)$$

Combining the Two Sets of Equations

$$F_{q1} = [I_1+(m_1+m_2+m_3)(R^2-2r_1R\cos q_1+r_1^2)]\ddot{q}_1 + (m_1+m_2+m_3)r_1 g\sin q_1$$
$$+m_2r_2r_1(\ddot{q}_2\cos(q_2-q_1)-\dot{q}_2^2\sin(q_2-q_1)) - m_2r_2R(\ddot{q}_2\cos q_2-\dot{q}_2^2\sin q_2)$$

$$F_{q2} = (I_2+m_2r_2^2)\ddot{q}_2-m_2r_2(\ddot{x}\cos q_2+(\ddot{y}+g)\sin q_2)$$

substituting

$$(\ddot{x}, \ddot{y}) = -\ddot{q}(R-r\cos q,-r\sin q)-\dot{q}^2(r\sin q,R-r\cos q)$$

$$F_{q2} = (I_2+m_2r_2^2)\ddot{q}_2+m_2r_2R(\ddot{q}_1\cos q_2+\dot{q}_1^2\sin q_2) - m_2gr_2\sin q_2$$
$$-m_2r_2l_1(\ddot{q}_1\cos(q_2-q_1)+\dot{q}_1^2\sin(q_2-q_1))$$

Appendix H

Twelve Segment Planar Object

This appendix describes the equations of motion for a twelve segment rigid body. The ground constraint was represented by an external force introduced at the ankle joint. In the representation for the LSODI integrator : y is a vector of the generalized coordinates after expansion to a first order system. s is a vector defining the approximate derivative to y.

Define the number of segments

```
NSEG                 /* Number of rigid segments            */
NEQ = 2*(NSEG+2)     /* Number of differential equations    */
```

Define the inertial variables.

```
I[NSEG]              /* Moment of Inertia                   */
M[NSEG]              /* Mass                                */
R[NSEG]              /* Proximal Distance to Centre of Mass*/
L[NSEG]              /* Length of Segment                   */
```

Define some convenient inertial variables

```
mhl[0]=   M[0]+M[1]+mhl[1]
mhl[1]=   M[2]+mhl[2]
mhl[2]=   M[3]+M[4]+M[5]+M[6]+mhl[3]
mhl[3]=   M[7]+M[8]+M[9]+M[10]+M[11]

mhl[20]=  M[0]*R[0]
mhl[21]=  M[1]*R[1]        + mhl[1]*L[1]
mhl[22]=  M[2]*R[2]        + mhl[2]*L[2]
mhl[23]=  M[3]*R[3]        + (M[4]+M[5])*L[3]
mhl[24]=  M[4]*R[4]        + M[5]*L[4]
mhl[25]=  M[5]*R[5]
mhl[26]=  M[6]*R[6]        + mhl[3]*L[6]
mhl[27]=  M[7]*R[7]        + M[8]*L[7]
mhl[28]=  M[8]*R[8]
mhl[29]=  M[9]*R[9]        + M[10]*L[9]
mhl[30]=  M[10]*R[10]
mhl[31]=  M[11]*R[11]

mhl[40]=  I[0]   + M[0]*R[0]*R[0]
mhl[41]=  I[1]   + M[1]*R[1]*R[1]    + mhl[1]*L[1]*L[1]
mhl[42]=  I[2]   + M[2]*R[2]*R[2]    + mhl[2]*L[2]*L[2]
mhl[43]=  I[3]   + M[3]*R[3]*R[3]    + (M[4]+M[5])*L[3]*L[3]
mhl[44]=  I[4]   + M[4]*R[4]*R[4]    + M[5]*L[4]*L[4]
mhl[45]=  I[5]   + M[5]*R[5]*R[5]
mhl[46]=  I[6]   + M[6]*R[6]*R[6]    + mhl[3]*L[6]*L[6]
mhl[47]=  I[7]   + M[7]*R[7]*R[7]    + M[8]*L[7]*L[7]
mhl[48]=  I[8]   + M[8]*R[8]*R[8]
mhl[49]=  I[9]   + M[9]*R[9]*R[9]    + M[10]*L[9]*L[9]
mhl[50]=  I[10]  + M[10]*R[10]*R[10]
```

mhl[51]= I[11] + M[11]*R[11]*R[11]

F_x = mhl[0] $\ddot{x}$ + mhl[20] $(\ddot{q}_1 \cos q_1 - \dot{q}_1^2 \sin q_1)$
-mhl[21]$(\ddot{q}_2 \cos q_2 - \dot{q}_2^2 \sin q_2)$ - mhl[22]$(\ddot{q}_3 \cos q_3 - \dot{q}_3^2 \sin q_3)$
+mhl[23] $(\ddot{q}_4 \cos q_4 - \dot{q}_4^2 \sin q_4)$ + mhl[24]$(\ddot{q}_5 \cos q_5 - \dot{q}_5^2 \sin q_5)$
+mhl[25] $(\ddot{q}_6 \cos q_6 - \dot{q}_6^2 \sin q_6)$ - mhl[26]$(\ddot{q}_7 \cos q_7 - \dot{q}_7^2 \sin q_7)$
+mhl[27] $(\ddot{q}_8 \cos q_8 - \dot{q}_8^2 \sin q_8)$ + mhl[28]$(\ddot{q}_9 \cos q_9 - \dot{q}_9^2 \sin q_9)$
+mhl[29] $(\ddot{q}_{10} \cos q_{10} - \dot{q}_{10}^2 \sin q_{10})$ + mhl[30]$(\ddot{q}_{11} \cos q_{11} - \dot{q}_{11}^2 \sin q_{11})$
-mhl[31] $(\ddot{q}_{12} \cos q_{12} - \dot{q}_{12}^2 \sin q_{12})$

F_y = mhl[0] $(\ddot{y} + g)$ + mhl[20] $(\ddot{q}_1 \sin q_1 + \dot{q}_1^2 \cos q_1)$
-mhl[21] $(\ddot{q}_2 \sin q_2 + \dot{q}_2^2 \cos q_2)$ - mhl[22] $(\ddot{q}_3 \sin q_3 + \dot{q}_3^2 \cos q_3)$
+mhl[23] $(\ddot{q}_4 \sin q_4 + \dot{q}_4^2 \cos q_4)$ + mhl[24] $(\ddot{q}_5 \sin q_5 + \dot{q}_5^2 \cos q_5)$
+mhl[25] $(\ddot{q}_6 \sin q_6 + \dot{q}_6^2 \cos q_6)$ - mhl[26] $(\ddot{q}_7 \sin q_7 + \dot{q}_7^2 \cos q_7)$
+mhl[27] $(\ddot{q}_8 \sin q_8 + \dot{q}_8^2 \cos q_8)$ + mhl[28] $(\ddot{q}_9 \sin q_9 + \dot{q}_9^2 \cos q_9)$
+mhl[29] $(\ddot{q}_{10} \sin q_{10} + \dot{q}_{10}^2 \cos q_{10})$ + mhl[30]$(\ddot{q}_{11} \sin q_{11} + \dot{q}_{11}^2 \cos q_{11})$
-mhl[31] $(\ddot{q}_{12} \cos q_{12} + \dot{q}_{12}^2 \sin q_{12})$

F_{q1} = mhl[40] $\ddot{q}_1$ + mhl[20] $(\ddot{x} \cos q_1 + (\ddot{y} + g) \sin q_1)$

F_{q2} = mhl[41] $\ddot{q}_2$ - mhl[21] $(\ddot{x} \cos q_2 + (\ddot{y} + g) \sin q_2)$ +
L[1](mhl[22] $(\ddot{q}_3 \cos(q_3 - q_2) - \dot{q}_3^2 \sin(q_3 - q_2))$ -
mhl[23] $(\ddot{q}_4 \cos(q_4 - q_2) - \dot{q}_4^2 \sin(q_4 - q_2))$ -
mhl[24] $(\ddot{q}_5 \cos(q_5 - q_2) - \dot{q}_5^2 \sin(q_5 - q_2))$ -
mhl[25] $(\ddot{q}_6 \cos(q_6 - q_2) - \dot{q}_6^2 \sin(q_6 - q_2))$ +
mhl[26] $(\ddot{q}_7 \cos(q_7 - q_2) - \dot{q}_7^2 \sin(q_7 - q_2))$ -
mhl[27] $(\ddot{q}_8 \cos(q_8 - q_2) - \dot{q}_8^2 \sin(q_8 - q_2))$ -
mhl[28] $(\ddot{q}_9 \cos(q_9 - q_2) - \dot{q}_9^2 \sin(q_9 - q_2))$ -

$$mhl[29]\,(\ddot{q}_{10}\cos(q_{10}-q_2)-\dot{q}_{10}^2\sin(q_{10}-q_2))\ -$$
$$mhl[30]\,(\ddot{q}_{11}\cos(q_{11}-q_2)-\dot{q}_{11}^2\sin(q_{11}-q_2))\ +$$
$$mhl[31]\,(\ddot{q}_{12}\cos(q_{12}-q_2)-\dot{q}_{12}^2\sin(q_{12}-q_2))\ )$$

$$F_{q3} = mhl[42]\,\ddot{q}_3 - mhl[22]\,(\ddot{x}\cos q_3+(\ddot{y}+g)\sin q_3\ -$$
$$L[1]\,(\ddot{q}_2\cos(q_3-q_2)+\dot{q}_2^2\sin(q_3-q_2))\ -$$
$$L[2](\ mhl[23]\,(\ddot{q}_4\cos(q_4-q_3)-\dot{q}_4^2\sin(q_4-q_3))\ +$$
$$mhl[24]\,(\ddot{q}_5\cos(q_5-q_3)-\dot{q}_5^2\sin(q_5-q_3))\ +$$
$$mhl[25]\,(\ddot{q}_6\cos(q_6-q_3)-\dot{q}_6^2\sin(q_6-q_3))\ -$$
$$mhl[26]\,(\ddot{q}_7\cos(q_7-q_3)-\dot{q}_7^2\sin(q_7-q_3))\ +$$
$$mhl[27]\,(\ddot{q}_8\cos(q_8-q_3)-\dot{q}_8^2\sin(q_8-q_3))\ +$$
$$mhl[28]\,(\ddot{q}_9\cos(q_9-q_3)-\dot{q}_9^2\sin(q_9-q_3))\ +$$
$$mhl[29]\,(\ddot{q}_{10}\cos(q_{10}-q_3)-\dot{q}_{10}^2\sin(q_{10}-q_3))\ +$$
$$mhl[30]\,(\ddot{q}_{11}\cos(q_{11}-q_3)-\dot{q}_{11}^2\sin(q_{11}-q_3))\ -$$
$$mhl[31]\,(\ddot{q}_{12}\cos(q_{12}-q_3)-\dot{q}_{12}^2\sin(q_{12}-q_3))\ )$$

$$F_{q4} = mhl[43]\,\ddot{q}_4 + mhl[23]\,(\ddot{x}\cos q_4+(\ddot{y}+g)\sin q_4\ +$$
$$L[1]\,(\ddot{q}_2\cos(q_4-q_2)+\dot{q}_2^2\sin(q_4-q_2))\ +$$
$$L[2]\,(\ddot{q}_3\cos(q_4-q_3)+\dot{q}_3^2\sin(q_4-q_3))\ +$$
$$L[3](\ mhl[24]\,(\ddot{q}_5\cos(q_5-q_4)-\dot{q}_5^2\sin(q_5-q_4))\ +$$
$$mhl[25]\,(\ddot{q}_6\cos(q_6-q_4)-\dot{q}_6^2\sin(q_6-q_4))\ )$$

$$F_{q5} = mhl[44]\,\ddot{q}_5 + mhl[24]\,(\ddot{x}\cos q_5+(\ddot{y}+g)\sin q_5\ +$$
$$L[1]\,(\ddot{q}_2\cos(q_5-q_2)+\dot{q}_2^2\sin(q_5-q_2))\ +$$
$$L[2]\,(\ddot{q}_3\cos(q_5-q_3)+\dot{q}_3^2\sin(q_5-q_3))\ -$$
$$L[3]\,(\ddot{q}_4\cos(q_5-q_4)+\dot{q}_4^2\sin(q_5-q_4))\ +$$

$$L[4](\ mhl[25]\ (\ddot{q}_6\cos(q_6-q_5)-\dot{q}_6^{\,2}\sin(q_6-q_5))\ )$$

$$F_{q6} = mhl[45]\ \ddot{q}_6 + mhl[25]\ (\ddot{x}\cos q_6+(\ddot{y}+g)\sin q_6 +$$

$$L[1]\ (\ddot{q}_2\cos(q_6-q_2)+\dot{q}_2^{\,2}\sin(q_6-q_2)) +$$

$$L[2]\ (\ddot{q}_3\cos(q_6-q_3)+\dot{q}_3^{\,2}\sin(q_6-q_3)) -$$

$$L[3]\ (\ddot{q}_4\cos(q_6-q_4)+\dot{q}_4^{\,2}\sin(q_6-q_4)) -$$

$$L[4]\ (\ddot{q}_5\cos(q_6-q_5)+\dot{q}_5^{\,2}\sin(q_6-q_5))\ )$$

$$F_{q7} = mhl[46]\ \ddot{q}_7 - mhl[26]\ (\ddot{x}\cos q_7+(\ddot{y}+g)\sin q_7 -$$

$$L[1]\ (\ddot{q}_2\cos(q_7-q_2)+\dot{q}_2^{\,2}\sin(q_7-q_2)) -$$

$$L[2]\ (\ddot{q}_3\cos(q_7-q_3)+\dot{q}_3^{\,2}\sin(q_7-q_3))\ ) -$$

$$L[6](\ mhl[27]\ (\ddot{q}_8\cos(q_8-q_7)-\dot{q}_8^{\,2}\sin(q_8-q_7)) +$$

$$mhl[28]\ (\ddot{q}_9\cos(q_9-q_7)-\dot{q}_9^{\,2}\sin(q_9-q_7)) +$$

$$mhl[29]\ (\ddot{q}_{10}\cos(q_{10}-q_7)-\dot{q}_{10}^{\,2}\sin(q_{10}-q_7)) +$$

$$mhl[30]\ (\ddot{q}_{11}\cos(q_{11}-q_7)-\dot{q}_{11}^{\,2}\sin(q_{11}-q_7)) -$$

$$mhl[31]\ (\ddot{q}_{12}\cos(q_{12}-q_7)-\dot{q}_{12}^{\,2}\sin(q_{12}-q_7))\ )$$

$$F_{q8} = mhl[47]\ \ddot{q}_8 + mhl[27]\ (\ddot{x}\cos q_8+(\ddot{y}+g)\sin q_8 +$$

$$L[1]\ (\ddot{q}_2\cos(q_8-q_2)+\dot{q}_2^{\,2}\sin(q_8-q_2)) +$$

$$L[2]\ (\ddot{q}_3\cos(q_8-q_3)+\dot{q}_3^{\,2}\sin(q_8-q_3)) +$$

$$L[6]\ (\ddot{q}_7\cos(q_8-q_7)+\dot{q}_7^{\,2}\sin(q_8-q_7))\ ) +$$

$$L[7](\ mhl[28]\ (\ddot{q}_9\cos(q_9-q_8)-\dot{q}_9^{\,2}\sin(q_9-q_8))\ )$$

$$F_{q9} = mhl[48]\ \ddot{q}_9 + mhl[28]\ (\ddot{x}\cos q_9+(\ddot{y}+g)\sin q_9 +$$

$$L[1]\ (\ddot{q}_2\cos(q_9-q_2)+\dot{q}_2^{\,2}\sin(q_9-q_2)) +$$

$$L[2]\ (\ddot{q}_3\cos(q_9-q_3)+\dot{q}_3^{\,2}\sin(q_9-q_3)) +$$

$$L[6]\ (\ddot{q}_7\cos(q_9-q_7)+\dot{q}_7^{\,2}\sin(q_9-q_7)) -$$

$$L[7]\ (\ddot{q}_8\cos(q_9-q_8)+\dot{q}_8^{\,2}\sin(q_9-q_8))\ )$$

$$F_{q10} = mhl[49]\,\ddot{q}_{10} + mhl[29]\,(\ddot{x}\cos q_{10}+(\ddot{y}+g)\sin q_{10} +$$
$$L[1]\,(\ddot{q}_2\cos(q_{10}-q_2)+\dot{q}_2{}^2\sin(q_{10}-q_2)) +$$
$$L[2]\,(\ddot{q}_3\cos(q_{10}-q_3)+\dot{q}_3{}^2\sin(q_{10}-q_3)) +$$
$$L[6]\,(\ddot{q}_7\cos(q_{10}-q_7)+\dot{q}_7{}^2\sin(q_{10}-q_7)) +$$
$$L[9]\,)\; mhl[28]\,(\ddot{q}_{11}\cos(q_{11}-q_{10})-\dot{q}_{11}{}^2\sin(q_{11}-q_{10})) +$$

$$F_{q11} = mhl[50]\,\ddot{q}_{11}+ mhl[30]\,(\ddot{x}\cos q_{11}+(\ddot{y}+g)\sin q_{11} +$$
$$L[1]\,(\ddot{q}_2\cos(q_{11}-q_2)+\dot{q}_2{}^2\sin(q_{11}-q_2)) +$$
$$L[2]\,(\ddot{q}_3\cos(q_{11}-q_3)+\dot{q}_3{}^2\sin(q_{11}-q_3)) +$$
$$L[6]\,(\ddot{q}_7\cos(q_{11}-q_7)+\dot{q}_7{}^2\sin(q_{11}-q_7)) -$$
$$L[9]\,(\ddot{q}_{10}\cos(q_{11}-q_{10})+\dot{q}_{10}{}^2\sin(q_{11}-q_{10}))\,)$$

$$F_{q12} = mhl[51]\,\ddot{q}_{12}- mhl[31]\,(\ddot{x}\cos q_{12}+(\ddot{y}+g)\sin q_{12} -$$
$$L[1]\,(\ddot{q}_2\cos(q_{12}-q_2)+\dot{q}_2{}^2\sin(q_{12}-q_2)) -$$
$$L[2]\,(\ddot{q}_3\cos(q_{12}-q_3)+\dot{q}_3{}^2\sin(q_{12}-q_3)) +$$
$$L[6]\,(\ddot{q}_7\cos(q_{12}-q_7)+\dot{q}_7{}^2\sin(q_{12}-q_7))\,)$$

Define the trigonometric variables

Cos1=	cos(y[4])	Sin1=	sin(y[4])
Cos2=	cos(y[6])	Sin2=	sin(y[6])
Cos3=	cos(y[8])	Sin3=	sin(y[8])
Cos4=	cos(y[10])	Sin4=	sin(y[10])
Cos5=	cos(y[12])	Sin5=	sin(y[12])
Cos6=	cos(y[14])	Sin6=	sin(y[14])
Cos7=	cos(y[16])	Sin7=	sin(y[16])
Cos8=	cos(y[18])	Sin8=	sin(y[18])
Cos9=	cos(y[20])	Sin9=	sin(y[20])
Cos10=	cos(y[22])	Sin10=	sin(y[22])
Cos11=	cos(y[24])	Sin11=	sin(y[24])
Cos12=	cos(y[26])	Sin12=	sin(y[26])
Cos32=	Cos3*Cos2+Sin3*Sin2	Sin32=	Sin3*Cos2-Cos3*Sin2
Cos42=	Cos4*Cos2+Sin4*Sin2	Sin42=	Sin4*Cos2-Cos4*Sin2
Cos52=	Cos5*Cos2+Sin5*Sin2	Sin52=	Sin5*Cos2-Cos5*Sin2
Cos62=	Cos6*Cos2+Sin6*Sin2	Sin62=	Sin6*Cos2-Cos6*Sin2
Cos72=	Cos7*Cos2+Sin7*Sin2	Sin72=	Sin7*Cos2-Cos7*Sin2
Cos82=	Cos8*Cos2+Sin8*Sin2	Sin82=	Sin8*Cos2-Cos8*Sin2
Cos92=	Cos9*Cos2+Sin9*Sin2	Sin92=	Sin9*Cos2-Cos9*Sin2
Cos102=	Cos10*Cos2+Sin10*Sin2	Sin102=	Sin10*Cos2-Cos10*Sin2
Cos112=	Cos11*Cos2+Sin11*Sin2	Sin112=	Sin11*Cos2-Cos11*Sin2
Cos122=	Cos12*Cos2+Sin12*Sin2	Sin122=	Sin12*Cos2-Cos12*Sin2
Cos43=	Cos4*Cos3+Sin4*Sin3	Sin43=	Sin4*Cos3-Cos4*Sin3
Cos53=	Cos5*Cos3+Sin5*Sin3	Sin53=	Sin5*Cos3-Cos5*Sin3
Cos63=	Cos6*Cos3+Sin6*Sin3	Sin63=	Sin6*Cos3-Cos6*Sin3
Cos73=	Cos7*Cos3+Sin7*Sin3	Sin73=	Sin7*Cos3-Cos7*Sin3
Cos83=	Cos8*Cos3+Sin8*Sin3	Sin83=	Sin8*Cos3-Cos8*Sin3
Cos93=	Cos9*Cos3+Sin9*Sin3	Sin93=	Sin9*Cos3-Cos9*Sin3
Cos103=	Cos10*Cos3+Sin10*Sin3	Sin103=	Sin10*Cos3-Cos10*Sin3
Cos113=	Cos11*Cos3+Sin11*Sin3	Sin113=	Sin11*Cos3-Cos11*Sin3
Cos123=	Cos12*Cos3+Sin12*Sin3	Sin123=	Sin12*Cos3-Cos12*Sin3
Cos54=	Cos5*Cos4+Sin5*Sin4	Sin54=	Sin5*Cos4-Cos5*Sin4
Cos64=	Cos6*Cos4+Sin6*Sin4	Sin64=	Sin6*Cos4-Cos6*Sin4
Cos65=	Cos6*Cos5+Sin6*Sin5	Sin65=	Sin6*Cos5-Cos6*Sin5
Cos87=	Cos8*Cos7+Sin8*Sin7	Sin87=	Sin8*Cos7-Cos8*Sin7
Cos97=	Cos9*Cos7+Sin9*Sin7	Sin97=	Sin9*Cos7-Cos9*Sin7
Cos107=	Cos10*Cos7+Sin10*Sin7	Sin107=	Sin10*Cos7-Cos10*Sin7
Cos117=	Cos11*Cos7+Sin11*Sin7	Sin117=	Sin11*Cos7-Cos11*Sin7
Cos127=	Cos12*Cos7+Sin12*Sin7	Sin127=	Sin12*Cos7-Cos12*Sin7
Cos98=	Cos9*Cos8+Sin9*Sin8	Sin98=	Sin9*Cos8-Cos9*Sin8
Cos1110=	Cos11*Cos10+Sin11*Sin10	Sin1110=	Sin11*Cos10-Cos11*Sin10

Define the Inertia Matrix

```
adda[1][1]    += mhl[0];
adda[1][7]    -= mhl[21] Cos2
adda[1][11]   += mhl[23] Cos4
adda[1][15]   += mhl[25] Cos6
adda[1][19]   += mhl[27] Cos8
adda[1][23]   += mhl[29] Cos10
adda[1][27]   -= mhl[31] Cos12
adda[1][5]    += mhl[20] Cos1
adda[1][9]    -= mhl[22] Cos3
adda[1][13]   += mhl[24] Cos5
adda[1][17]   -= mhl[26] Cos7
adda[1][21]   += mhl[28] Cos9
adda[1][25]   += mhl[30] Cos11

adda[3][3]    += mhl[0];
adda[3][7]    -= mhl[21] Sin2
adda[3][11]   += mhl[23] Sin4
adda[3][15]   += mhl[25] Sin6
adda[3][19]   += mhl[27] Sin8
adda[3][23]   += mhl[29] Sin10
adda[3][27]   -= mhl[31] Sin12
adda[3][5]    += mhl[20] Sin1
adda[3][9]    -= mhl[22] Sin3
adda[3][13]   += mhl[24] Sin5
adda[3][17]   -= mhl[26] Sin7
adda[3][21]   += mhl[28] Sin9
adda[3][25]   += mhl[30] Sin11

adda[5][1]    += mhl[20] Cos1
adda[5][5]    += mhl[40]
adda[5][3]    += mhl[20] Sin1

adda[7][1]    -= mhl[21] Cos2
adda[7][7]    += mhl[41]
adda[7][11]   -= mhl[23] L[1] Cos42
adda[7][15]   -= mhl[25] L[1] Cos62
adda[7][19]   -= mhl[27] L[1] Cos82
adda[7][23]   -= mhl[29] L[1] Cos102
adda[7][27]   += mhl[31] L[1] Cos122
adda[7][3]    -= mhl[21] Sin2
adda[7][9]    += mhl[22] L[1] Cos32
adda[7][13]   -= mhl[24] L[1] Cos52
adda[7][17]   += mhl[26] L[1] Cos72
adda[7][21]   -= mhl[28] L[1] Cos92
adda[7][25]   -= mhl[30] L[1] Cos112

adda[9][1]    -= mhl[22] Cos3
adda[9][7]    += mhl[22] L[1] Cos32
adda[9][11]   -= mhl[23] L[2] Cos43
adda[9][15]   -= mhl[25] L[2] Cos63
adda[9][19]   -= mhl[27] L[2] Cos83
adda[9][23]   -= mhl[29] L[2] Cos103
adda[9][27]   += mhl[31] L[2] Cos123
adda[9][3]    -= mhl[22] Sin3
adda[9][9]    += mhl[42]
adda[9][13]   -= mhl[24] L[2] Cos53
adda[9][17]   += mhl[26] L[2] Cos73
adda[9][21]   -= mhl[28] L[2] Cos93
adda[9][25]   -= mhl[30] L[2] Cos113

adda[11][1]   += mhl[23] Cos4
adda[11][7]   -= mhl[23] L[1] Cos42
adda[11][11]  += mhl[43]
adda[11][15]  += mhl[25] L[3] Cos65
adda[11][3]   += mhl[23] Sin4
adda[11][9]   -= mhl[23] L[2] Cos43
adda[11][13]  += mhl[24] L[3] Cos64

adda[13][1]   += mhl[24] Cos5
adda[13][7]   -= mhl[24] L[1] Cos52
adda[13][11]  += mhl[24] L[3] Cos54
adda[13][15]  += mhl[25] L[3] Cos65
adda[13][3]   += mhl[24] Sin5
adda[13][9]   -= mhl[24] L[2] Cos53
adda[13][13]  += mhl[44]

adda[15][1]   += mhl[25] Cos6
adda[15][7]   -= mhl[25] L[1] Cos62
adda[15][11]  += mhl[25] L[3] Cos64
adda[15][15]  += mhl[45]
adda[15][3]   += mhl[25] Sin6
adda[15][9]   -= mhl[25] L[2] Cos63
adda[15][13]  += mhl[25] L[4] Cos65
```

```
adda[17][1]    -= mhl[26] Cos7              adda[17][3]    -= mhl[26] Sin7
adda[17][7]    += mhl[26] L[1] Cos72        adda[17][9]    += mhl[26] L[1] Cos72
adda[17][17]   += mhl[46]                   adda[17][19]   -= mhl[27] L[6] Cos87
adda[17][21]   -= mhl[28] L[6] Cos97        adda[17][23]   -= mhl[29] L[6] Cos107
adda[17][25]   -= mhl[30] L[6] Cos117       adda[17][27]   += mhl[31] L[6] Cos127

adda[19][1]    += mhl[27] Cos8              adda[19][3]    += mhl[27] Sin8
adda[19][7]    -= mhl[27] L[1] Cos82        adda[19][9]    -= mhl[27] L[1] Cos82
adda[19][17]   -= mhl[27] L[6] Cos87        adda[19][19]   += mhl[47]
adda[19][21]   += mhl[28] L[7] Cos98

adda[21][1]    += mhl[28] Cos9              adda[21][3]    += mhl[28] Sin9
adda[21][7]    -= mhl[28] L[1] Cos92        adda[21][9]    -= mhl[28] L[1] Cos92
adda[21][17]   -= mhl[28] L[6] Cos97        adda[21][19]   += mhl[28] L[7] Cos98
adda[21][21]   += mhl[48]

adda[23][1]    += mhl[29] Cos10             adda[23][3]    += mhl[29] Sin10
adda[23][7]    -= mhl[29] L[1] Cos102       adda[23][9]    -= mhl[29] L[1] Cos103
adda[23][17]   -= mhl[29] L[6] Cos107       adda[23][19]   += mhl[49]
adda[23][21]   += mhl[30] L[9] Cos1110

adda[25][1]    += mhl[30] Cos11             adda[25][3]    += mhl[30] Sin11
adda[25][7]    -= mhl[30] L[1] Cos112       adda[25][9]    -= mhl[30] L[1] Cos113
adda[25][17]   -= mhl[30] L[6] Cos117       adda[25][23]   += mhl[30] L[9] Cos1110
adda[25][25]   += mhl[50]

adda[27][1]    -= mhl[31] Cos12             adda[27][3]    -= mhl[31] Sin12
adda[27][7]    += mhl[31] L[1] Cos122       adda[27][9]    += mhl[31] L[1] Cos123
adda[27][17]   += mhl[31] L[6] Cos127       adda[27][27]   += mhl[51]

res :

p[1] = F[0]     - mhl[0] s[1]               - mhl[20] (s[5] Cos1  -y[5]y[5]   Sin1) +
      mhl[21] (s[7]  Cos2  -y[7]  y[7]  Sin2)  + mhl[22] (s[9]  Cos3  -y[9]y[9]   Sin3) -
      mhl[23] (s[11]Cos4  -y[11]y[11]Sin4)  - mhl[24] (s[13]Cos5  -y[13]y[13]Sin5) -
      mhl[25] (s[15]Cos6  -y[15]y[15]Sin6)  + mhl[26] (s[17]Cos7  -y[17]y[17]Sin7) -
      mhl[27] (s[19]Cos8  -y[19]y[19]Sin8)  - mhl[28] (s[21]Cos9  -y[21]y[21]Sin9) -
      mhl[29] (s[23]Cos10-y[23]y[23]Sin10) - mhl[30] (s[25]Cos11-y[25]y[25]Sin11)+
      mhl[31] (s[27]Cos12-y[27]y[27]Sin12)

p[3] = F[1]     - mhl[0] (s[3]+g)           - mhl[20] (s[5]  Sin1  +y[5]y[5]   Cos1)
+
      mhl[21] (s[7]  Sin2  +y[7]  y[7]  Cos2)  + mhl[22] (s[9]  Sin3  +y[9]y[9]   Cos3) -
      mhl[23] (s[11]Sin4  +y[11]y[11]Cos4)  - mhl[24] (s[13]Sin5  +y[13]y[13]Cos5) -
      mhl[25] (s[15]Sin6  +y[15]y[15]Cos6)  + mhl[26] (s[17]Sin7  +y[17]y[17]Cos7) -
      mhl[27] (s[19]Sin8  +y[19]y[19]Cos8)  - mhl[28] (s[21]Sin9  +y[21]y[21]Cos9) -
      mhl[29] (s[23]Sin10+y[23]y[23]Cos10) - mhl[30] (s[25]Sin11+y[25]y[25]Cos11)+
      mhl[31] (s[27]Sin12+y[27]y[27]Cos12)

p[5] = F[3]     - mhl[40] s[5]              - mhl[20] (s[1]Cos1+(s[3]+g)Sin1)
```

```
p[7] = F[4]    - mhl[41] s[7]         + mhl[21] (s[1]Cos2+(s[3]+g)Sin2) -
  L[1] *(
  mhl[22] (s[9]Cos32-y[9]y[9]Sin32)        - mhl[23] (s[11]Cos42-y[11]y[11]Sin42)    -
  mhl[24] (s[13]Cos52-y[13]y[13]Sin52)    - mhl[25] (s[15]Cos62-y[15]y[15]Sin62)    +
  mhl[26] (s[17]Cos72-y[17]y[17]Sin72)    - mhl[27] (s[19]Cos82-y[19]y[19]Sin82)    -
  mhl[28] (s[21]Cos92-y[21]y[21]Sin92)    - mhl[29] (s[23]Cos102-y[23]y[23]Sin102) -
  mhl[30] (s[25]Cos112-y[25]y[25]Sin112)+ mhl[31] (s[27]Cos122-y[27]y[27]Sin122))

p[9]= F[5]     - mhl[42] s[9]  +
  mhl[22] (s[1]Cos3+(s[3]+g)Sin3 -          L[1](s[7]Cos32+y[7]y[7]Sin32) ) +
  L[2]*(
  mhl[23] (s[11]Cos43-y[11]y[11]Sin43)    +mhl[24] (s[13]Cos53-y[13]y[13]Sin53)   +
  mhl[25] (s[15]Cos63-y[15]y[15]Sin63)    -mhl[26] (s[17]Cos73-y[17]y[17]Sin73)   +
  mhl[27] (s[19]Cos83-y[19]y[19]Sin83)    +mhl[28] (s[21]Cos93-y[21]y[21]Sin93)   +
  mhl[29] (s[23]Cos103-y[23]y[23]Sin103) +mhl[30] (s[25]Cos113-y[25]y[25]Sin113) -
  mhl[31] (s[27]Cos123-y[27]y[27]Sin123) )

p[11]= F[6]    - mhl[43] s[11]        - mhl[23] (s[1]Cos4+(s[3]+g)Sin4 -
  L[1](s[7]Cos42+y[7]y[7]Sin42) ) - L[2](s[9]Cos43+y[9]y[9]Sin43) ) -
  L[3]*(
  mhl[24] (s[13]Cos54-y[13]y[13]Sin54) - mhl[25] (s[15]Cos64-y[15]y[15]Sin64) )

p[13]= F[7]    - mhl[44] s[13]        - mhl[24] (s[1]Cos5+(s[3]+g)Sin5 -
  L[1](s[7]Cos52+y[7]y[7]Sin52)  - L[2](s[9]Cos53+y[9]y[9]Sin53) +
  L[3](s[11]Cos54+y[11]y[11]Sin54) ) -
  L[4]( mhl[25] (s[15]Cos65-y[15]y[15Sin65) )

p[15]= F[8]    - mhl[45] s[15]        - mhl[25] (s[1]Cos6+(s[3]+g)Sin6 -
  L[1](s[7]Cos62+y[7]y[7]Sin62)       - L[2](s[9]Cos63+y[9]y[9]Sin63)  +
  L[3](s[11]Cos64+y[11]y[11]Sin64) + L[4](s[13]Cos65+y[13]y[13]Sin65) )

p[17]= F[9]    - mhl[46] s[17]        + mhl[26] (s[1]Cos7+(s[3]+g)Sin7 -
  L[1](s[7]Cos72+y[7]y[7]Sin72) )  - L[2](s[9]Cos73+y[9]y[9]Sin73) ) +
  L[6]*(
  mhl[27] (s[19]Cos87  -y[19]y[19]Sin87)  -mhl[28] (s[21]Cos97  -y[21]y[21]Sin97) -
  mhl[29] (s[23]Cos107-y[23]y[23]Sin107) -mhl[30] (s[25]Cos117-y[25]y[25]Sin117)-
  mhl[31] (s[27]Cos127-y[27]y[27]Sin127) )

p[19]= F[10]  - mhl[47] s[19]         - mhl[27] (s[1]Cos8+(s[3]+g)Sin8 -
  L[1](s[7]Cos82+y[7]y[7]Sin82) - L[2](s[9]Cos83+y[9]y[9]Sin83) -
  L[6](s[17]Cos87+y[17]y[17]Sin87) ) -
  L[7]( mhl[28] (s[21]Cos98-y[21]y[21]Sin98) )

p[21]= F[11]  - mhl[48] s[21]         - mhl[28] (s[1]Cos9+(s[3]+g)Sin9 -
  L[1](s[7]Cos92+y[7]y[7]Sin92)       - L[2](s[9]Cos93+y[9]y[9]Sin93) -
  L[6](s[17]Cos97+y[17]y[17]Sin97) + L[7](s[19]Cos98+y[19]y[19]Sin98) )

p[23]= F[12]  - mhl[49] s[23]         - mhl[29] (s[1]Cos10+(s[3]+g)Sin10 -
  L[1](s[7]Cos102+y[7]y[7]Sin102) - L[2](s[9]Cos103+y[9]y[9]Sin103)  -
  L[6](s[17]Cos107+y[17]y[17]Sin107) )  -
  L[9]( mhl[30] (s[25]Cos1110-y[25]y25]Sin1110) )
```

p[25]= F[13] - mhl[50] s[25] - mhl[30] (s[1]Cos11+(s[3]+g)Sin11 -
L[1](s[7]Cos112+y[7]y[7]Sin112) - L[2](s[9]Cos113+y[9]y[9]Sin113) -
L[6](s[17]Cos117+y[17]y[17]Sin117) + L[9](s[23]Cos1110+y[23]y[23]Sin1110))

p[27]= F[14] - mhl[51] s[27] + mhl[31] (s[1]Cos12+(s[3]+g)Sin12 -
L[1](s[7]Cos122+y[7]y[7]Sin122) - L[2](s[9]Cos123+y[9]y[9]Sin123) -
L[6](s[17]Cos127+y[17]y[17]Sin127))

jac :

b[1][4]= mhl[10](s[5]Sin1+y[5]y[5]Cos1)
b[1][5]= 2.0 mhl[10] y[5]Sin1
b[1][6]= -mhl[11](s[7]Sin2+y[7]y[7]Cos2)
b[1][7]= -2.0 mhl[11] y[7]Sin2
b[1][8]= -mhl[12](s[9]Sin3+y[9]y[9]Cos3)
b[1][9]= -2.0 mhl[12] y[9]Sin3
b[1][10]= mhl[13](s[11]Sin4+y[11]y[11]Cos4)
b[1][11]= 2.0 mhl[13] y[11]Sin4
b[1][12]= mhl[14](s[13]Sin5+y[13]y[13]Cos5)
b[1][13]= 2.0 mhl[14] y[13]Sin5
b[1][14]= mhl[15](s[15]Sin6+y[15]y[15]Cos6)
b[1][15]= 2.0 mhl[15] y[15]Sin6
b[1][16]= -mhl[16](s[17]Sin7+y[17]y[17]Cos7)
b[1][17]= -2.0 mhl[16] y[17]Sin7
b[1][18]= mhl[17](s[19]Sin8+y[19]y[19]Cos8)
b[1][19]= 2.0 mhl[17] y[19]Sin8
b[1][20]= mhl[18](s[21]Sin9+y[21]y[21]Cos9)
b[1][21]= 2.0 mhl[18] y[21]Sin9
b[1][22]= mhl[19](s[23]Sin10+y[23]y[23]Cos10)
b[1][23]= 2.0 mhl[19] y[23]Sin10
b[1][24]= mhl[20](s[25]Sin11+y[25]y[25]Cos11)
b[1][25]= 2.0 mhl[20] y[25]Sin11
b[1][26]= -mhl[21](s[27]Sin12+y[27]y[27]Cos12)
b[1][27]= -2.0 mhl[21] y[27]Sin12

b[3][4]= -mhl[10](s[5]Cos1+y[5]y[5]Sin1)
b[3][5]= -2.0 mhl[10] y[5]Cos1
b[3][6]= mhl[11](s[7]Cos2+y[7]y[7]Sin2)
b[3][7]= 2.0 mhl[11] y[7]Cos2
b[3][8]= mhl[12](s[9]Cos3+y[9]y[9]Sin3)
b[3][9]= 2.0 mhl[12] y[9]Cos3
b[3][10]= -mhl[13](s[11]Cos4+y[11]y[11]Sin4)
b[3][11]= -2.0 mhl[13] y[11]Cos4
b[3][12]= -mhl[14](s[13]Cos5+y[13]y[13]Sin5)
b[3][13]= -2.0 mhl[14] y[13]Cos5
b[3][14]= -mhl[15](s[15]Cos6+y[15]y[15]Sin6)
b[3][15]= -2.0 mhl[15] y[15]Cos6
b[3][16]= mhl[16](s[17]Cos7+y[17]y[17]Sin7)
b[3][17]= 2.0 mhl[16] y[17]Cos7
b[3][18]= -mhl[17](s[19]Cos8+y[19]y[19]Sin8)
b[3][19]= -2.0 mhl[17] y[19]Cos8
b[3][20]= -mhl[18](s[21]Cos9+y[21]y[21]Sin9)
b[3][21]= -2.0 mhl[18] y[21]Cos9
b[3][22]= -mhl[19](s[23]Cos10+y[23]y[23]Sin10)

b[3][23]=	-2.0 mhl[19] y[23]Cos10
b[3][24]=	-mhl[20](s[25]Cos11+y[25]y[25]Sin11)
b[3][25]=	-2.0 mhl[20] y[25]Cos11
b[3][26]=	mhl[21](s[27]Cos12+y[27]y[27]Sin12)
b[3][27]=	2.0 mhl[21] y[27]Cos12
b[5][4]=	mhl[20] (s[1] Sin1-(s[3]+g) Cos1)
b[7][8]=	mhl[22] L[1] (s[9]Sin32+y[9]y[9]Cos32)
b[7][9]=	2.0 mhl[22] L[1] y[9] Sin32
b[7][10]=	-mhl[23] L[1] (s[11]Sin42+y[11]y[11]Cos42)
b[7][11]=	-2.0 mhl[23] L[1] y[11] Sin42
b[7][12]=	-mhl[24] L[1] (s[13]Sin52+y[13]y[13]Cos52)
b[7][13]=	-2.0 mhl[24] L[1] y[13] Sin52
b[7][14]=	-mhl[25] L[1] (s[15]Sin62+y[15]y[15]Cos62)
b[7][15]=	-2.0 mhl[25] L[1] y[15] Sin62
b[7][16]=	mhl[26] L[1] (s[17]Sin72+y[17]y[17]Cos72)
b[7][17]=	2.0 mhl[26] L[1] y[17] Sin72
b[7][18]=	-mhl[27] L[1] (s[19]Sin82+y[19]y[19]Cos82)
b[7][19]=	-2.0 mhl[27] L[1] y[19] Sin82
b[7][20]=	-mhl[28] L[1] (s[21]Sin92+y[21]y[21]Cos92)
b[7][21]=	-2.0 mhl[28] L[1] y[21] Sin92
b[7][22]=	-mhl[29] L[1] (s[23]Sin102+y[23]y[23]Cos102)
b[7][23]=	-2.0 mhl[29] L[1] y[23] Sin102
b[7][24]=	-mhl[30] L[1] (s[25]Sin112+y[25]y[25]Cos112)
b[7][25]=	-2.0 mhl[30] L[1] y[25] Sin112
b[7][26]=	mhl[31] L[1] (s[27]Sin122+y[27]y[27]Cos122)
b[7][27]=	2.0 mhl[31] L[1] y[27] Sin122
b[7][6]=	-mhl[21] (s[1] Sin2-(s[3]+g) Cos2) - b[7][4] - b[7][8] - b[7][10] - b[7][12] - b[7][14] - b[7][16] - b[7][18] - b[7][20] - b[7][22] - b[7][24] - b[7][26]
b[9][6]=	-mhl[23] L[1] (s[7]Sin32-y[7]y[7]Cos32)
b[9][7]=	-2.0 mhl[23] L[1] y[7] Sin32
b[9][10]=	-mhl[23] L[2] (s[11]Sin43+y[11]y[11]Cos43)
b[9][11]=	-2.0 mhl[23] L[2] y[11] Sin43
b[9][12]=	-mhl[24] L[2] (s[13]Sin53+y[13]y[13]Cos53)
b[9][13]=	-2.0 mhl[24] L[2] y[13] Sin53
b[9][14]=	-mhl[25] L[2] (s[15]Sin63+y[15]y[15]Cos63)
b[9][15]=	-2.0 mhl[25] L[2] y[15] Sin63
b[9][16]=	mhl[26] L[2] (s[17]Sin73+y[17]y[17]Cos73)
b[9][17]=	2.0 mhl[26] L[2] y[17] Sin73
b[9][18]=	-mhl[27] L[2] (s[19]Sin83+y[19]y[19]Cos83)
b[9][19]=	-2.0 mhl[27] L[2] y[19] Sin83
b[9][20]=	-mhl[28] L[2] (s[21]Sin93+y[21]y[21]Cos93)
b[9][21]=	-2.0 mhl[28] L[2] y[21] Sin93
b[9][22]=	-mhl[29] L[2] (s[23]Sin103+y[23]y[23]Cos103)
b[9][23]=	-2.0 mhl[29] L[2] y[23] Sin103
b[9][24]=	-mhl[30] L[2] (s[25]Sin113+y[25]y[25]Cos113)
b[9][25]=	-2.0 mhl[30] L[2] y[25] Sin113
b[9][26]=	mhl[31] L[2] (s[27]Sin123+y[27]y[27]Cos123)
b[9][27]=	2.0 mhl[31] L[2] y[27] Sin123

b[9][8]=	-mhl[22] (s[1] Sin3-(s[3]+g) Cos3) - b[9][4] - b[9][6] - b[9][10] - b[9][12] - b[9][14] - b[9][16] - b[9][18] - b[9][20] - b[9][22] - b[9][24] - b[9][26]
b[11][6]=	mhl[23] L[1] (s[7]Sin42-y[7]y[7]Cos42)
b[11][7]=	2.0 mhl[23] L[1] y[7] Sin42
b[11][8]=	mhl[23] L[2] (s[9]Sin43-y[9]y[9]Cos43)
b[11][9]=	2.0 mhl[23] L[2] y[9] Sin43
b[11][12]=	mhl[24] L[3] (s[13]Sin54+y[13]y[13]Cos54)
b[11][13]=	2.0 mhl[24] L[3] y[13] Sin54
b[11][14]=	mhl[25] L[3] (s[15]Sin64+y[15]y[15]Cos64)
b[11][15]=	2.0 mhl[25] L[3] y[15] Sin64
b[11][10]=	mhl[23] (s[1] Sin4-(s[3]+g) Cos4) - b[11][6] - b[11][8] - b[11][12] - b[11][14]
b[13][6]=	mhl[24] L[1] (s[7]Sin52-y[7]y[7]Cos52)
b[13][7]=	2.0 mhl[24] L[1] y[7] Sin52
b[13][8]=	mhl[24] L[2] (s[9]Sin53-y[9]y[9]Cos53)
b[13][9]=	2.0 mhl[24] L[2] y[9] Sin53
b[13][10]=	-mhl[24] L[3] (s[11]Sin54-y[11]y[11]Cos54)
b[13][11]=	-2.0 mhl[24] L[3] y[11] Sin54
b[13][14]=	mhl[25] L[4] (s[15]Sin65+y[15]y[15]Cos65)
b[13][15]=	2.0 mhl[25] L[4] y[15] Sin65
b[13][12]=	mhl[24] (s[1] Sin5-(s[3]+g) Cos5) - b[13][6] - b[13][8] - b[13][10] - b[13][14]
b[15][6]=	mhl[25] L[1] (s[7]Sin62-y[7]y[7]Cos62)
b[15][7]=	2.0 mhl[25] L[1] y[7] Sin62
b[15][8]=	mhl[25] L[2] (s[9]Sin63-y[9]y[9]Cos63)
b[15][9]=	2.0 mhl[25] L[2] y[9] Sin63
b[15][10]=	-mhl[25] L[3] (s[11]Sin64-y[11]y[11]Cos64)
b[15][11]=	-2.0 mhl[25] L[3] y[11] Sin64
b[15][12]=	-mhl[25] L[4] (s[13]Sin65-y[13]y[13]Cos65)
b[15][13]=	-2.0 mhl[25] L[4] y[13] Sin65
b[15][14]=	mhl[25] (s[1] Sin6-(s[3]+g) Cos6) - b[15][6] - b[15][8] - b[15][10] - b[15][12]
b[17][6]=	-mhl[26] L[1] (s[7]Sin72-y[7]y[7]Cos72)
b[17][7]=	-2.0 mhl[26] L[1] y[7] Sin72
b[17][8]=	-mhl[26] L[2] (s[9]Sin73-y[9]y[9]Cos73)
b[17][9]=	-2.0 mhl[26] L[2] y[9] Sin73
b[17][18]=	-mhl[27] L[6] (s[19]Sin87+y[19]y[19]Cos87)
b[17][19]=	-2.0 mhl[27] L[6] y[19] Sin87
b[17][20]=	-mhl[28] L[6] (s[21]Sin97+y[21]y[21]Cos97)
b[17][21]=	-2.0 mhl[28] L[6] y[21] Sin97
b[17][22]=	-mhl[29] L[6] (s[23]Sin107+y[23]y[23]Cos107)
b[17][23]=	-2.0 mhl[29] L[6] y[23] Sin107
b[17][24]=	-mhl[30] L[6] (s[25]Sin117+y[25]y[25]Cos117)
b[17][25]=	-2.0 mhl[30] L[6] y[25] Sin117
b[17][26]=	mhl[31] L[6] (s[27]Sin127+y[27]y[27]Cos127)
b[17][27]=	2.0 mhl[31] L[6] y[27] Sin127

b[17][16]= -mhl[26] (s[1] Sin7-(s[3]+g) Cos7) - b[17][6] - b[17][8] - b[17][18] - b[17][20] - b[17][22] - b[17][24] - b[17][26]

b[19][6]= mhl[27] L[1] (s[7]Sin82-y[7]y[7]Cos82)
b[19][7]= 2.0 mhl[27] L[1] y[7] Sin82
b[19][8]= mhl[27] L[2] (s[9]Sin83-y[9]y[9]Cos83)
b[19][9]= 2.0 mhl[27] L[2] y[9] Sin83
b[19][16]= mhl[27] L[6] (s[17]Sin87-y[17]y[17]Cos87)
b[19][17]= 2.0 mhl[27] L[6] y[17] Sin87
b[19][20]= mhl[28] L[7] (s[21]Sin98+y[21]y[21]Cos98)
b[19][21]= 2.0 mhl[28] L[7] y[21] Sin98

b[19][18]= mhl[27] (s[1] Sin8-(s[3]+g) Cos8) - b[19][6] - b[19][8] - b[19][16] - b[19][20]

b[21][6]= mhl[28] L[1] (s[7]Sin92-y[7]y[7]Cos92)
b[21][7]= 2.0 mhl[28] L[1] y[7] Sin92
b[21][8]= mhl[28] L[2] (s[9]Sin93-y[9]y[9]Cos93)
b[21][9]= 2.0 mhl[28] L[2] y[9] Sin93
b[21][16]= mhl[28] L[6] (s[17]Sin97-y[17]y[17]Cos97)
b[21][17]= 2.0 mhl[28] L[6] y[17] Sin97
b[21][18]= -mhl[28] L[7] (s[19]Sin98-y[19]y[19]Cos98)
b[21][19]= -2.0 mhl[28] L[7] y[19] Sin98

b[21][20]= mhl[28] (s[1] Sin9-(s[3]+g) Cos9) - b[21][6] - b[21][8] - b[21][16] - b[21][18]

b[23][6]= mhl[29] L[1] (s[7]Sin102-y[7]y[7]Cos102)
b[23][7]= 2.0 mhl[29] L[1] y[7] Sin102
b[23][8]= mhl[29] L[2] (s[9]Sin103-y[9]y[9]Cos103)
b[23][9]= 2.0 mhl[29] L[2] y[9] Sin103
b[23][16]= mhl[29] L[6] (s[17]Sin107-y[17]y[17]Cos107)
b[23][17]= 2.0 mhl[29] L[6] y[17] Sin107
b[23][24]= mhl[30] L[9] (s[25]Sin1110+y[25]y[25]Cos1110)
b[23][25]= 2.0 mhl[30] L[9] y[25] Sin1110

b[23][22]= mhl[29] (s[1] Sin10-(s[3]+g) Cos10) - b[23][6] - b[23][8] - b[23][16] - b[23][24]

b[25][6]= mhl[30] L[1] (s[7]Sin112-y[7]y[7]Cos112)
b[25][7]= 2.0 mhl[30] L[1] y[7] Sin112
b[25][8]= mhl[30] L[2] (s[9]Sin113-y[9]y[9]Cos113)
b[25][9]= 2.0 mhl[30] L[2] y[9] Sin113
b[25][16]= mhl[30] L[6] (s[17]Sin117-y[17]y[17]Cos117)
b[25][17]= 2.0 mhl[30] L[6] y[17] Sin117
b[25][22]= -mhl[30] L[9] (s[23]Sin1110-y[23]y[23]Cos1110)
b[25][23]= -2.0 mhl[30] L[9] y[123] Sin1110

b[25][24]= mhl[30] (s[1] Sin11-(s[3]+g) Cos11) - b[25][6] - b[25][8] - b[25][16] - b[25][22]

b[27][6]= mhl[31] L[1] (s[7]Sin122-y[7]y[7]Cos122)
b[27][7]= 2.0 mhl[31] L[1] y[7] Sin122
b[27][8]= mhl[31] L[2] (s[9]Sin123-y[9]y[9]Cos123)
b[27][9]= 2.0 mhl[31] L[2] y[9] Sin123
b[27][16]= mhl[31] L[6] (s[17]Sin127-y[17]y[17]Cos127)
b[27][17]= 2.0 mhl[31] L[6] y[17] Sin127

b[27][26]= -mhl[31] (s[1] Sin12-(s[3]+g) Cos12) - b[27][6] - b[27][8] - b[27][16]

Appendix I

Twelve Segment Planar Object Including the Foot Segment

This appendix describes the equations of motion for a twelve segment rigid body. The foot has been included using the method explained in Appendix G. In the representation for the LSODI integrator : y is a vector of the generalized coordinates after expansion to a first order system. s is a vector defining the approximate derivative to y. There are 24 degrees of freedom in this model.

Define the number of segments

```
NSEG                    /* Number of rigid segments           */
NEQ = 2*(NSEG+2)        /* Number of differential equations   */
```

Define the inertial variables.

```
I[NSEG]                 /* Moment of Inertia                  */
M[NSEG]                 /* Mass                               */
R[NSEG]                 /* Proximal Distance to Centre of Mass*/
L[NSEG]                 /* Length of Segment                  */
```

Define some convenient inertial variables

```
mhl[0]=    M[0]+mhl[1]
mhl[1]=    M[1]+mhl[2]
mhl[2]=    M[2]+mhl[3]
mhl[3]=    M[3]+M[4]+M[5]+M[6]+mhl[4]
mhl[4]=    M[7]+M[8]+M[9]+M[10]+M[11]

mhl[20]=   mhl[0]*R[0]
mhl[21]=   M[1]*R[1]        + mhl[2]*L[1]
mhl[22]=   M[2]*R[2]        + mhl[3]*L[2]
mhl[23]=   M[3]*R[3]        + (M[4]+M[5])*L[3]
mhl[24]=   M[4]*R[4]        + M[5]*L[4]
mhl[25]=   M[5]*R[5]
mhl[26]=   M[6]*R[6]        + mhl[3]*L[6]
mhl[27]=   M[7]*R[7]        + M[8]*L[7]
mhl[28]=   M[8]*R[8]
mhl[29]=   M[9]*R[9]        + M[10]*L[9]
mhl[30]=   M[10]*R[10]
mhl[31]=   M[11]*R[11]

mhl[40]=   I[0]  + mhl[0]*(Rad*Rad+R[0]*R[0])
mhl[41]=   I[1]  + M[1]*R[1]*R[1]     + mhl[2]*L[1]*L[1]
mhl[42]=   I[2]  + M[2]*R[2]*R[2]     + mhl[3]*L[2]*L[2]
mhl[43]=   I[3]  + M[3]*R[3]*R[3]     + (M[4]+M[5])*L[3]*L[3]
mhl[44]=   I[4]  + M[4]*R[4]*R[4]     + M[5]*L[4]*L[4]
mhl[45]=   I[5]  + M[5]*R[5]*R[5]
mhl[46]=   I[6]  + M[6]*R[6]*R[6]     + mhl[3]*L[6]*L[6]
mhl[47]=   I[7]  + M[7]*R[7]*R[7]     + M[8]*L[7]*L[7]
mhl[48]=   I[8]  + M[8]*R[8]*R[8]
mhl[49]=   I[9]  + M[9]*R[9]*R[9]     + M[10]*L[9]*L[9]
```

mhl[50]= I[10] + M[10]*R[10]*R[10]
mhl[51]= I[11] + M[11]*R[11]*R[11]

$$F_{q1} = (mhl[40] - 2mhl[20]Rad\cos q_1)\ddot{q}_1 + mhl[20]\, g\sin q_1$$

$$+mhl[21](\, Rad(\ddot{q}_2\cos q_2 - \dot{q}_2^2\sin q_2) \quad - R[0](\ddot{q}_2 \cos(q_2-q_1) - \dot{q}_2^2\sin(q_2-q_1)\,)$$

$$+mhl[22](\, Rad(\ddot{q}_3\cos q_3 - \dot{q}_3^2\sin q_3) \quad - R[0](\ddot{q}_3 \cos(q_3-q_1) - \dot{q}_3^2\sin(q_3-q_1)\,)$$

$$-mhl[23](\, Rad(\ddot{q}_4\cos q_4 - \dot{q}_4^2\sin q_4) \quad - R[0](\ddot{q}_4 \cos(q_4-q_1) - \dot{q}_4^2\sin(q_4-q_1)\,)$$

$$-mhl[24](\, Rad(\ddot{q}_5\cos q_5 - \dot{q}_5^2\sin q_5) \quad - R[0](\ddot{q}_5 \cos(q_5-q_1) - \dot{q}_5^2\sin(q_5-q_1)\,)$$

$$-mhl[25](\, Rad(\ddot{q}_6\cos q_6 - \dot{q}_6^2\sin q_6) \quad - R[0](\ddot{q}_6 \cos(q_6-q_1) - \dot{q}_6^2\sin(q_6-q_1)\,)$$

$$+mhl[26](\, Rad(\ddot{q}_7\cos q_7 - \dot{q}_7^2\sin q_7) \quad - R[0](\ddot{q}_7 \cos(q_7-q_1) - \dot{q}_7^2\sin(q_7-q_1)\,)$$

$$-mhl[27](\, Rad(\ddot{q}_8\cos q_8 - \dot{q}_8^2\sin q_8) \quad - R[0](\ddot{q}_8 \cos(q_8-q_1) - \dot{q}_8^2\sin(q_8-q_1)\,)$$

$$-mhl[28](\, Rad(\ddot{q}_9\cos q_9 - \dot{q}_9^2\sin q_9) \quad - R[0](\ddot{q}_9 \cos(q_9-q_1) - \dot{q}_9^2\sin(q_9-q_1)\,)$$

$$-mhl[29](\, Rad(\ddot{q}_{10}\cos q_{10}-\dot{q}_{10}^2\sin q_{10})-R[0](\ddot{q}_{10}\cos(q_{10}-q_1)-\dot{q}_{10}^2\sin(q_{10}-q_1))$$

$$-mhl[30](\, Rad(\ddot{q}_{11}\cos q_{11}-\dot{q}_{11}^2\sin q_{11})-R[0](\ddot{q}_{11}\cos(q_{11}-q_1)-\dot{q}_{11}^2\sin(q_{11}-q_1))$$

$$+mhl[31](\, Rad(\ddot{q}_{12}\cos q_{12}-\dot{q}_{12}^2\sin q_{12})-R[0](\ddot{q}_{12}\cos(q_{12}-q_1)-\dot{q}_{12}^2\sin(q_{12}-q_1))$$

$$F_{q2} = mhl[41]\, \ddot{q}_2$$

$$+ mhl[21](\, Rad(\ddot{q}_1\cos q_2 + \dot{q}_1^2\sin q_2) \; - R[0](\ddot{q}_1 \cos(q_2-q_1) + \dot{q}_1^2\sin(q_2-q_1)\,)\,) +$$

$$L[1](\, mhl[22]\, (\ddot{q}_3\cos(q_3-q_2)-\dot{q}_3^2\sin(q_3-q_2)) -$$

$$mhl[23]\, (\ddot{q}_4\cos(q_4-q_2)-\dot{q}_4^2\sin(q_4-q_2)) -$$

$$mhl[24]\, (\ddot{q}_5\cos(q_5-q_2)-\dot{q}_5^2\sin(q_5-q_2)) -$$

$$mhl[25]\, (\ddot{q}_6\cos(q_6-q_2)-\dot{q}_6^2\sin(q_6-q_2)) +$$

$$mhl[26]\, (\ddot{q}_7\cos(q_7-q_2)-\dot{q}_7^2\sin(q_7-q_2)) -$$

$$mhl[27]\, (\ddot{q}_8\cos(q_8-q_2)-\dot{q}_8^2\sin(q_8-q_2)) -$$

$$mhl[28]\, (\ddot{q}_9\cos(q_9-q_2)-\dot{q}_9^2\sin(q_9-q_2)) -$$

$$mhl[29]\, (\ddot{q}_{10}\cos(q_{10}-q_2)-\dot{q}_{10}^2\sin(q_{10}-q_2)) -$$

$$mhl[30]\, (\ddot{q}_{11}\cos(q_{11}-q_2)-\dot{q}_{11}^2\sin(q_{11}-q_2)) +$$

$$mhl[31]\,(\ddot{q}_{12}\cos(q_{12}-q_2)-\dot{q}_{12}^2\sin(q_{12}-q_2))\,)$$

$$F_{q3} = mhl[42]\,\ddot{q}_3$$

$$+\,mhl[22](\,Rad(\ddot{q}_1\cos q_3 + \dot{q}_1^2\sin q_3) \quad - R[0](\ddot{q}_1\,\cos(q_3-q_1) + \dot{q}_1^2\sin(q_3-q_1) -$$

$$L[1]\,(\ddot{q}_2\cos(q_3-q_2)+\dot{q}_2^2\sin(q_3-q_2)) -$$

$$L[2](\,mhl[23]\,(\ddot{q}_4\cos(q_4-q_3)-\dot{q}_4^2\sin(q_4-q_3)) +$$

$$mhl[24]\,(\ddot{q}_5\cos(q_5-q_3)-\dot{q}_5^2\sin(q_5-q_3)) +$$

$$mhl[25]\,(\ddot{q}_6\cos(q_6-q_3)-\dot{q}_6^2\sin(q_6-q_3)) -$$

$$mhl[26]\,(\ddot{q}_7\cos(q_7-q_3)-\dot{q}_7^2\sin(q_7-q_3)) +$$

$$mhl[27]\,(\ddot{q}_8\cos(q_8-q_3)-\dot{q}_8^2\sin(q_8-q_3)) +$$

$$mhl[28]\,(\ddot{q}_9\cos(q_9-q_3)-\dot{q}_9^2\sin(q_9-q_3)) +$$

$$mhl[29]\,(\ddot{q}_{10}\cos(q_{10}-q_3)-\dot{q}_{10}^2\sin(q_{10}-q_3)) +$$

$$mhl[30]\,(\ddot{q}_{11}\cos(q_{11}-q_3)-\dot{q}_{11}^2\sin(q_{11}-q_3)) -$$

$$mhl[31]\,(\ddot{q}_{12}\cos(q_{12}-q_3)-\dot{q}_{12}^2\sin(q_{12}-q_3))\,)$$

$$F_{q4} = mhl[43]\,\ddot{q}_4$$

$$-\,mhl[23](\,Rad(\ddot{q}_1\cos q_4 + \dot{q}_1^2\sin q_4) \quad - R[0](\ddot{q}_1\,\cos(q_4-q_1) + \dot{q}_1^2\sin(q_4-q_1) -$$

$$L[1]\,(\ddot{q}_2\cos(q_4-q_2)+\dot{q}_2^2\sin(q_4-q_2)) +$$

$$L[2]\,(\ddot{q}_3\cos(q_4-q_3)+\dot{q}_3^2\sin(q_4-q_3)) +$$

$$L[3](\,mhl[24]\,(\ddot{q}_5\cos(q_5-q_4)-\dot{q}_5^2\sin(q_5-q_4)) +$$

$$mhl[25]\,(\ddot{q}_6\cos(q_6-q_4)-\dot{q}_6^2\sin(q_6-q_4))\,)$$

$$F_{q5} = mhl[44]\,\ddot{q}_5$$

$$-\,mhl[24](\,Rad(\ddot{q}_1\cos q_5 + \dot{q}_1^2\sin q_5) \quad - R[0](\ddot{q}_1\,\cos(q_5-q_1) + \dot{q}_1^2\sin(q_5-q_1) -$$

$$L[1]\,(\ddot{q}_2\cos(q_5-q_2)+\dot{q}_2^2\sin(q_5-q_2)) +$$

$$L[2]\,(\ddot{q}_3\cos(q_5-q_3)+\dot{q}_3^2\sin(q_5-q_3)) -$$

$$L[3]\,(\ddot{q}_4\cos(q_5-q_4)+\dot{q}_4^2\sin(q_5-q_4)) +$$

$L[4](\,mhl[25]\,(\ddot{q}_6\cos(q_6-q_5)-\dot{q}_6{}^2\sin(q_6-q_5))\,)$

$F_{q6} = mhl[45]\,\ddot{q}_6$

$-\,mhl[25](\,Rad(\ddot{q}_1\cos q_6 + \dot{q}_1{}^2\sin q_6) \quad - R[0](\ddot{q}_1\cos(q_6-q_1) + \dot{q}_1{}^2\sin(q_6-q_1) -$

$L[1]\,(\ddot{q}_2\cos(q_6-q_2)+\dot{q}_2{}^2\sin(q_6-q_2)) +$

$L[2]\,(\ddot{q}_3\cos(q_6-q_3)+\dot{q}_3{}^2\sin(q_6-q_3)) -$

$L[3]\,(\ddot{q}_4\cos(q_6-q_4)+\dot{q}_4{}^2\sin(q_6-q_4)) -$

$L[4]\,(\ddot{q}_5\cos(q_6-q_5)+\dot{q}_5{}^2\sin(q_6-q_5))\,)$

$F_{q7} = mhl[46]\,\ddot{q}_7$

$+\,mhl[26](\,Rad(\ddot{q}_1\cos q_7 + \dot{q}_1{}^2\sin q_7) \quad - R[0](\ddot{q}_1\cos(q_7-q_1) + \dot{q}_1{}^2\sin(q_7-q_1) -$

$L[1]\,(\ddot{q}_2\cos(q_7-q_2)+\dot{q}_2{}^2\sin(q_7-q_2)) -$

$L[2]\,(\ddot{q}_3\cos(q_7-q_3)+\dot{q}_3{}^2\sin(q_7-q_3))\,) -$

$L[6](\,mhl[27]\,(\ddot{q}_8\cos(q_8-q_7)-\dot{q}_8{}^2\sin(q_8-q_7)) +$

$mhl[28]\,(\ddot{q}_9\cos(q_9-q_7)-\dot{q}_9{}^2\sin(q_9-q_7)) +$

$mhl[29]\,(\ddot{q}_{10}\cos(q_{10}-q_7)-\dot{q}_{10}{}^2\sin(q_{10}-q_7)) +$

$mhl[30]\,(\ddot{q}_{11}\cos(q_{11}-q_7)-\dot{q}_{11}{}^2\sin(q_{11}-q_7)) -$

$mhl[31]\,(\ddot{q}_{12}\cos(q_{12}-q_7)-\dot{q}_{12}{}^2\sin(q_{12}-q_7))\,)$

$F_{q8} = mhl[47]\,\ddot{q}_8$

$-\,mhl[27](\,Rad(\ddot{q}_1\cos q_8 + \dot{q}_1{}^2\sin q_8) \quad - R[0](\ddot{q}_1\cos(q_8-q_1) + \dot{q}_1{}^2\sin(q_8-q_1) -$

$L[1]\,(\ddot{q}_2\cos(q_8-q_2)+\dot{q}_2{}^2\sin(q_8-q_2)) +$

$L[2]\,(\ddot{q}_3\cos(q_8-q_3)+\dot{q}_3{}^2\sin(q_8-q_3)) +$

$L[6]\,(\ddot{q}_7\cos(q_8-q_7)+\dot{q}_7{}^2\sin(q_8-q_7))\,) +$

$L[7](\,mhl[28]\,(\ddot{q}_9\cos(q_9-q_8)-\dot{q}_9{}^2\sin(q_9-q_8))\,)$

$$F_{q9} = mhl[48]\,\ddot{q}_9$$

$$- mhl[28](\,Rad(\ddot{q}_1 \cos q_9 + \dot{q}_1{}^2 \sin q_9) \quad - R[0](\ddot{q}_1 \cos(q_9-q_1) + \dot{q}_1{}^2 \sin(q_9-q_1) -$$

$$L[1]\,(\ddot{q}_2 \cos(q_9-q_2)+\dot{q}_2{}^2 \sin(q_9-q_2)) +$$

$$L[2]\,(\ddot{q}_3 \cos(q_9-q_3)+\dot{q}_3{}^2 \sin(q_9-q_3)) +$$

$$L[6]\,(\ddot{q}_7 \cos(q_9-q_7)+\dot{q}_7{}^2 \sin(q_9-q_7)) -$$

$$L[7]\,(\ddot{q}_8 \cos(q_9-q_8)+\dot{q}_8{}^2 \sin(q_9-q_8))\,)$$

$$F_{q10} = mhl[49]\,\ddot{q}_{10}$$

$$-mhl[29](\,Rad(\ddot{q}_1 \cos q_{10} + \dot{q}_1{}^2 \sin q_{10})-R[0](\ddot{q}_1 \cos(q_{10}-q_1) + \dot{q}_1{}^2 \sin(q_{10}-q_1) -$$

$$L[1]\,(\ddot{q}_2 \cos(q_{10}-q_2)+\dot{q}_2{}^2 \sin(q_{10}-q_2)) +$$

$$L[2]\,(\ddot{q}_3 \cos(q_{10}-q_3)+\dot{q}_3{}^2 \sin(q_{10}-q_3)) +$$

$$L[6]\,(\ddot{q}_7 \cos(q_{10}-q_7)+\dot{q}_7{}^2 \sin(q_{10}-q_7)) +$$

$$L[9](\,mhl[28]\,(\ddot{q}_{11} \cos(q_{11}-q_{10})-\dot{q}_{11}{}^2 \sin(q_{11}-q_{10})) +$$

$$F_{q11} = mhl[50]\,\ddot{q}_{11}$$

$$-mhl[50](\,Rad(\ddot{q}_1 \cos q_{11} + \dot{q}_1{}^2 \sin q_{11})-R[0](\ddot{q}_1 \cos(q_{11}-q_1) + \dot{q}_1{}^2 \sin(q_{11}-q_1) -$$

$$L[1]\,(\ddot{q}_2 \cos(q_{11}-q_2)+\dot{q}_2{}^2 \sin(q_{11}-q_2)) +$$

$$L[2]\,(\ddot{q}_3 \cos(q_{11}-q_3)+\dot{q}_3{}^2 \sin(q_{11}-q_3)) +$$

$$L[6]\,(\ddot{q}_7 \cos(q_{11}-q_7)+\dot{q}_7{}^2 \sin(q_{11}-q_7)) -$$

$$L[9]\,(\ddot{q}_{10} \cos(q_{11}-q_{10})+\dot{q}_{10}{}^2 \sin(q_{11}-q_{10}))\,)$$

$$F_{q12} = mhl[51]\,\ddot{q}_{12}$$

$$+mhl[51](\,Rad(\ddot{q}_1 \cos q_{12} + \dot{q}_1{}^2 \sin q_{12})-R[0](\ddot{q}_1 \cos(q_{12}-q_1) + \dot{q}_1{}^2 \sin(q_{12}-q_1) -$$

$$L[1]\,(\ddot{q}_2 \cos(q_{12}-q_2)+\dot{q}_2{}^2 \sin(q_{12}-q_2)) -$$

$$L[2]\,(\ddot{q}_3 \cos(q_{12}-q_3)+\dot{q}_3{}^2 \sin(q_{12}-q_3)) +$$

$$L[6]\,(\ddot{q}_7 \cos(q_{12}-q_7)+\dot{q}_7{}^2 \sin(q_{12}-q_7))\,)$$

Define the trigonometric variables

Cos1=	cos(y[0])	Sin1=	sin(y[0])
Cos2=	cos(y[2])	Sin2=	sin(y[2])
Cos3=	cos(y[4])	Sin3=	sin(y[4])
Cos4=	cos(y[6])	Sin4=	sin(y[6])
Cos5=	cos(y[8])	Sin5=	sin(y[8])
Cos6=	cos(y[10])	Sin6=	sin(y[10])
Cos7=	cos(y[12])	Sin7=	sin(y[12])
Cos8=	cos(y[14])	Sin8=	sin(y[14])
Cos9=	cos(y[16])	Sin9=	sin(y[16])
Cos10=	cos(y[18])	Sin10=	sin(y[18])
Cos11=	cos(y[20])	Sin11=	sin(y[20])
Cos12=	cos(y[22])	Sin12=	sin(y[22])
Cos21=	Cos2*Cos1+Sin2*Sin1	Sin21=	Sin2*Cos1-Cos2*Sin1
Cos31=	Cos3*Cos1+Sin3*Sin1	Sin31=	Sin3*Cos1-Cos3*Sin1
Cos41=	Cos4*Cos1+Sin4*Sin1	Sin41=	Sin4*Cos1-Cos4*Sin1
Cos51=	Cos5*Cos1+Sin5*Sin1	Sin51=	Sin5*Cos1-Cos5*Sin1
Cos61=	Cos6*Cos1+Sin6*Sin1	Sin61=	Sin6*Cos1-Cos6*Sin1
Cos71=	Cos7*Cos1+Sin7*Sin1	Sin71=	Sin7*Cos1-Cos7*Sin1
Cos81=	Cos8*Cos1+Sin8*Sin1	Sin81=	Sin8*Cos1-Cos8*Sin1
Cos91=	Cos9*Cos1+Sin9*Sin1	Sin91=	Sin9*Cos1-Cos9*Sin1
Cos101=	Cos10*Cos1+Sin10*Sin1	Sin101=	Sin10*Cos1-Cos10*Sin1
Cos111=	Cos11*Cos1+Sin11*Sin1	Sin111=	Sin11*Cos1-Cos11*Sin1
Cos121=	Cos12*Cos1+Sin12*Sin1	Sin121=	Sin12*Cos1-Cos12*Sin1
Cos32=	Cos3*Cos2+Sin3*Sin2	Sin32=	Sin3*Cos2-Cos3*Sin2
Cos42=	Cos4*Cos2+Sin4*Sin2	Sin42=	Sin4*Cos2-Cos4*Sin2
Cos52=	Cos5*Cos2+Sin5*Sin2	Sin52=	Sin5*Cos2-Cos5*Sin2
Cos62=	Cos6*Cos2+Sin6*Sin2	Sin62=	Sin6*Cos2-Cos6*Sin2
Cos72=	Cos7*Cos2+Sin7*Sin2	Sin72=	Sin7*Cos2-Cos7*Sin2
Cos82=	Cos8*Cos2+Sin8*Sin2	Sin82=	Sin8*Cos2-Cos8*Sin2
Cos92=	Cos9*Cos2+Sin9*Sin2	Sin92=	Sin9*Cos2-Cos9*Sin2
Cos102=	Cos10*Cos2+Sin10*Sin2	Sin102=	Sin10*Cos2-Cos10*Sin2
Cos112=	Cos11*Cos2+Sin11*Sin2	Sin112=	Sin11*Cos2-Cos11*Sin2
Cos122=	Cos12*Cos2+Sin12*Sin2	Sin122=	Sin12*Cos2-Cos12*Sin2
Cos43=	Cos4*Cos3+Sin4*Sin3	Sin43=	Sin4*Cos3-Cos4*Sin3
Cos53=	Cos5*Cos3+Sin5*Sin3	Sin53=	Sin5*Cos3-Cos5*Sin3
Cos63=	Cos6*Cos3+Sin6*Sin3	Sin63=	Sin6*Cos3-Cos6*Sin3
Cos73=	Cos7*Cos3+Sin7*Sin3	Sin73=	Sin7*Cos3-Cos7*Sin3
Cos83=	Cos8*Cos3+Sin8*Sin3	Sin83=	Sin8*Cos3-Cos8*Sin3
Cos93=	Cos9*Cos3+Sin9*Sin3	Sin93=	Sin9*Cos3-Cos9*Sin3
Cos103=	Cos10*Cos3+Sin10*Sin3	Sin103=	Sin10*Cos3-Cos10*Sin3
Cos113=	Cos11*Cos3+Sin11*Sin3	Sin113=	Sin11*Cos3-Cos11*Sin3
Cos123=	Cos12*Cos3+Sin12*Sin3	Sin123=	Sin12*Cos3-Cos12*Sin3
Cos54=	Cos5*Cos4+Sin5*Sin4	Sin54=	Sin5*Cos4-Cos5*Sin4
Cos64=	Cos6*Cos4+Sin6*Sin4	Sin64=	Sin6*Cos4-Cos6*Sin4
Cos65=	Cos6*Cos5+Sin6*Sin5	Sin65=	Sin6*Cos5-Cos6*Sin5
Cos87=	Cos8*Cos7+Sin8*Sin7	Sin87=	Sin8*Cos7-Cos8*Sin7
Cos97=	Cos9*Cos7+Sin9*Sin7	Sin97=	Sin9*Cos7-Cos9*Sin7

Cos107= Cos10*Cos7+Sin10*Sin7	Sin107= Sin10*Cos7-Cos10*Sin7
Cos117= Cos11*Cos7+Sin11*Sin7	Sin117= Sin11*Cos7-Cos11*Sin7
Cos127= Cos12*Cos7+Sin12*Sin7	Sin127= Sin12*Cos7-Cos12*Sin7
Cos98= Cos9*Cos8+Sin9*Sin8	Sin98= Sin9*Cos8-Cos9*Sin8
Cos1110= Cos11*Cos10+Sin11*Sin10	Sin1110= Sin11*Cos10-Cos11*Sin10

Define the Inertia Matrix

adda[1][1]	+= mhl[40] - 2mhl[20]Rad Cos1
adda[1][3]	+= mhl[21](Rad Cos2- R[0]Sin21)
adda[1][5]	+= mhl[22](Rad Cos3- R[0]Sin31)
adda[1][7]	-= mhl[23](Rad Cos4- R[0]Sin41)
adda[1][9]	-= mhl[24](Rad Cos5- R[0]Sin51)
adda[1][11]	-= mhl[25](Rad Cos6- R[0]Sin61)
adda[1][13]	+= mhl[26](Rad Cos7- R[0]Sin71)
adda[1][15]	-= mhl[27](Rad Cos8- R[0]Sin81)
adda[1][17]	-= mhl[28](Rad Cos9- R[0]Sin91)
adda[1][19]	-= mhl[29](Rad Cos10-R[0]Sin101)
adda[1][21]	-= mhl[30](Rad Cos11-R[0]Sin111)
adda[1][23]	+= mhl[31](Rad Cos12-R[0]Sin121)

adda[3][3]	+= mhl[41]	adda[3][5]	+= mhl[22] L[1] Cos32
adda[3][7]	-= mhl[23] L[1] Cos42	adda[3][9]	-= mhl[24] L[1] Cos52
adda[3][11]	-= mhl[25] L[1] Cos62	adda[3][13]	+= mhl[26] L[1] Cos72
adda[3][15]	-= mhl[27] L[1] Cos82	adda[3][17]	-= mhl[28] L[1] Cos92
adda[3][19]	-= mhl[29] L[1] Cos102	adda[3][21]	-= mhl[30] L[1] Cos112
adda[3][23]	+= mhl[31] L[1] Cos122		

adda[5][3]	+= mhl[22] L[1] Cos32	adda[5][5]	+= mhl[42]
adda[5][7]	-= mhl[23] L[2] Cos43	adda[5][9]	-= mhl[24] L[2] Cos53
adda[5][11]	-= mhl[25] L[2] Cos63	adda[5][13]	+= mhl[26] L[2] Cos73
adda[5][15]	-= mhl[27] L[2] Cos83	adda[5][17]	-= mhl[28] L[2] Cos93
adda[5][19]	-= mhl[29] L[2] Cos103	adda[5][21]	-= mhl[30] L[2] Cos113
adda[5][23]	+= mhl[31] L[2] Cos123		

adda[7][3]	-= mhl[23] L[1] Cos42	adda[7][5]	-= mhl[23] L[2] Cos43
adda[7][7]	+= mhl[43]	adda[7][9]	+= mhl[24] L[3] Cos64
adda[7][11]	+= mhl[25] L[3] Cos65		

adda[9][3]	-= mhl[24] L[1] Cos52	adda[9][5]	-= mhl[24] L[2] Cos53
adda[9][7]	+= mhl[24] L[3] Cos54	adda[9][9]	+= mhl[44]
adda[9][11]	+= mhl[25] L[3] Cos65		

adda[11][3]	-= mhl[25] L[1] Cos62	adda[11][5]	-= mhl[25] L[2] Cos63
adda[11][7]	+= mhl[25] L[3] Cos64	adda[11][9]	+= mhl[25] L[4] Cos65
adda[11][11]	+= mhl[45]		

adda[13][3]	+= mhl[26] L[1] Cos72	adda[13][5]	+= mhl[26] L[1] Cos72
adda[13][13]	+= mhl[46]	adda[13][15]	-= mhl[27] L[6] Cos87
adda[13][17]	-= mhl[28] L[6] Cos97	adda[13][19]	-= mhl[29] L[6] Cos107
adda[13][21]	-= mhl[30] L[6] Cos117	adda[13][23]	+= mhl[31] L[6] Cos127

```
adda[15][3]   -= mhl[27] L[1] Cos82       adda[15][5]   -= mhl[27] L[1] Cos82
adda[15][13]  -= mhl[27] L[6] Cos87       adda[15][15]  += mhl[47]
adda[15][17]  += mhl[28] L[7] Cos98

adda[17][3]   -= mhl[28] L[1] Cos92       adda[17][5]   -= mhl[28] L[1] Cos92
adda[17][13]  -= mhl[28] L[6] Cos97       adda[17][15]  += mhl[28] L[7] Cos98
adda[17][17]  += mhl[48]

adda[19][3]   -= mhl[29] L[1] Cos102      adda[19][5]   -= mhl[29] L[1] Cos103
adda[19][13]  -= mhl[29] L[6] Cos107      adda[19][15]  += mhl[49]
adda[19][17]  += mhl[30] L[9] Cos1110

adda[21][3]   -= mhl[30] L[1] Cos112      adda[21][5]   -= mhl[30] L[1] Cos113
adda[21][13]  -= mhl[30] L[6] Cos117      adda[21][19]  += mhl[30] L[9] Cos1110
adda[21][21]  += mhl[50]

adda[23][3]   += mhl[31] L[1] Cos122      adda[23][5]   += mhl[31] L[1] Cos123
adda[23][13]  += mhl[31] L[6] Cos127      adda[23][23]  += mhl[51]

res :

p[1] = F[0]   - (mhl[40] - 2mhl[20]Rad Cos1) s[1]        + mhl[20] gSin1
+mhl[21]( R(s[3]Cos2   - y[3]y[3]Sin2)       - L[0](s[3]Cos21    - y[3]y[3]Sin21) )
+mhl[22]( R(s[5]Cos3   - y[5]y[5]Sin3)       - L[0](s[5]Cos31    - y[5]y[5]Sin31) )
-mhl[23]( R(s[7]Cos4   - y[7]y[7]Sin4)       - L[0](s[7]Cos41    - y[7]y[7]Sin41) )
-mhl[24]( R(s[9]Cos5   - y[9]y[9]Sin5)       - L[0](s[9]Cos51    - y[9]y[9]Sin51) )
-mhl[25]( R(s[11]Cos6  - y[11]y[11]Sin6)  - L[0](s[11]Cos61  - y[11]y[11]Sin61) )
+mhl[26]( R(s[13]Cos7  - y[13]y[13]Sin7)  - L[0](s[13]Cos71  - y[13]y[13]Sin71) )
-mhl[27]( R(s[15]Cos8  - y[15]y[15]Sin8)  - L[0](s[15]Cos81  - y[15]y[15]Sin81) )
-mhl[28]( R(s[17]Cos9  - y[17]y[17]Sin9)  - L[0](s[17]Cos91  - y[17]y[17]Sin91) )
-mhl[29]( R(s[19]Cos10 - y[19]y[19]Sin10) - L[0](s[19]Cos101 - y[19]y[19]Sin101) )
-mhl[30]( R(s[21]Cos11 - y[21]y[21]Sin11) - L[0](s[21]Cos111 - y[21]y[21]Sin111) )
+mhl[31]( R(s[23]Cos12 - y[23]y[23]Sin12) - L[0](s[23]Cos121 - y[23]y[23]Sin121) )

p[3] = F[1]    - mhl[41] s[3]         + mhl[21] ( gSin2
      - Rad(s[1]Cos2    + y[1]y[1]Sin2)   + L[0](s[1]Cos21    + y[1]y[1]Sin21) -
  L[1] *(
  mhl[22] (s[5]Cos32-y[5]y[5]Sin32)        - mhl[23] (s[7]Cos42-y[7]y[7]Sin42)   -
  mhl[24] (s[9]Cos52-y[9]y[9]Sin52)         - mhl[25] (s[11]Cos62-y[11]y[11]Sin62)
+
  mhl[26] (s[13]Cos72-y[13]y[13]Sin72)   - mhl[27] (s[15]Cos82-y[15]y[15]Sin82)   -
  mhl[28] (s[17]Cos92-y[17]y[17]Sin92)   - mhl[29] (s[19]Cos102-y[19]y[19]Sin102) -
  mhl[30] (s[21]Cos112-y[21]y[21]Sin112)+ mhl[31] (s[23]Cos122-y[23]y[23]Sin122))

p[5]= F[2]     - mhl[42] s[5]  + mhl[22] (gSin3
      - Rad(s[1]Cos3    + y[1]y[1]Sin3)  + L[0](s[1]Cos31     + y[1]y[1]Sin31) -
  L[1](s[3]Cos32+y[3]y[3]Sin32) ) +
  L[2]*(
  mhl[23] (s[7]Cos43-y[7]y[7]Sin43)    +mhl[24] (s[9]Cos53-y[9]y[9]Sin53)   +
  mhl[25] (s[11]Cos63-y[11]y[11]Sin63)    -mhl[26] (s[13]Cos73-y[13]y[13]Sin73)   +
  mhl[27] (s[15]Cos83-y[15]y[15]Sin83)    +mhl[28] (s[17]Cos93-y[17]y[17]Sin93)   +
  mhl[29] (s[19]Cos103-y[19]y[19]Sin103) +mhl[30] (s[21]Cos113-y[21]y[21]Sin113) -
  mhl[31] (s[23]Cos123-y[23]y[23]Sin123) )
```

```
p[7]= F[3]    - mhl[43] s[7]          - mhl[23] (gSin4
        - Rad(s[1]Cos4    + y[1]y[1]Sin4)  + L[0](s[1]Cos41    + y[1]y[1]Sin41) -
   L[1](s[3]Cos42+y[3]y[3]Sin42) ) - L[2](s[5]Cos43+y[5]y[5]Sin43) ) -
   L[3]*(
   mhl[24] (s[9]Cos54-y[9]y[9]Sin54) - mhl[25] (s[11]Cos64-y[11]y[11]Sin64) )

p[9]= F[4]    - mhl[44] s[9]          - mhl[24] (gSin5
        - Rad(s[1]Cos5    + y[1]y[1]Sin5)  + L[0](s[1]Cos51    + y[1]y[1]Sin51) -
   L[1](s[3]Cos52+y[3]y[3]Sin52)  - L[2](s[5]Cos53+y[5]y[5]Sin53) +
   L[3](s[7]Cos54+y[7]y[7]Sin54) ) -
   L[4]( mhl[25] (s[11]Cos65-y[11]y[11]Sin65) )

p[11]= F[5]    - mhl[45] s[11]          - mhl[25] (gSin6
        - Rad(s[1]Cos6    + y[1]y[1]Sin6)  + L[0](s[1]Cos61    + y[1]y[1]Sin61) -
   L[1](s[3]Cos62+y[3]y[3]Sin62)    - L[2](s[5]Cos63+y[5]y[5]Sin63)  +
   L[3](s[7]Cos64+y[7]y[7]Sin64)    + L[4](s[9]Cos65+y[9]y[9]Sin65) )

p[13]= F[6]    - mhl[46] s[13]          + mhl[26] (gSin7
        - Rad(s[1]Cos7    + y[1]y[1]Sin7)  + L[0](s[1]Cos71    + y[1]y[1]Sin71) -
   L[1](s[3]Cos72+y[3]y[3]Sin72) )  - L[2](s[5]Cos73+y[5]y[5]Sin73) ) +
   L[6]*(
   mhl[27] (s[15]Cos87  -y[15]y[15]Sin87)  -mhl[28] (s[17]Cos97  -y[17]y[17]Sin97) -
   mhl[29] (s[19]Cos107-y[19]y[19]Sin107) -mhl[30] (s[21]Cos117-y[21]y[21]Sin117)-
   mhl[31] (s[23]Cos127-y[23]y[23]Sin127) )

p[15]= F[7]    - mhl[47] s[15]          - mhl[27] (gSin8
        - Rad(s[1]Cos8    + y[1]y[1]Sin8)  + L[0](s[1]Cos81    + y[1]y[1]Sin81) -
   L[1](s[3]Cos82+y[3]y[3]Sin82) - L[2](s[5]Cos83+y[5]y[5]Sin83) -
   L[6](s[13]Cos87+y[13]y[13]Sin87) ) -
   L[7]( mhl[28] (s[17]Cos98-y[17]y[17]Sin98) )

p[17]= F[8]    - mhl[48] s[17]          - mhl[28] (gSin9
        - Rad(s[1]Cos9    + y[1]y[1]Sin9)  + L[0](s[1]Cos91    + y[1]y[1]Sin91) -
   L[1](s[3]Cos92+y[3]y[3]Sin92)     - L[2](s[5]Cos93+y[5]y[5]Sin93) -
   L[6](s[13]Cos97+y[13]y[13]Sin97) + L[7](s[15]Cos98+y[15]y[15]Sin98) )

p[19]= F[9]    - mhl[49] s[19]          - mhl[29] (gSin19
        - Rad(s[1]Cos10    + y[1]y[1]Sin10)  + L[0](s[1]Cos101    + y[1]y[1]Sin101) -
   L[1](s[3]Cos102+y[3]y[3]Sin102) - L[2](s[5]Cos103+y[5]y[5]Sin103)  -
   L[6](s[13]Cos107+y[13]y[13]Sin107) )  -
   L[9]( mhl[30] (s[21]Cos1110-y[21]y21]Sin1110) )

p[21]= F[10]   - mhl[50] s[21]          - mhl[30] (gSin11
        - Rad(s[1]Cos11    + y[1]y[1]Sin11)  + L[0](s[1]Cos111    + y[1]y[1]Sin111) -
   L[1](s[3]Cos112+y[3]y[3]Sin112)     - L[2](s[5]Cos113+y[5]y[5]Sin113) -
   L[6](s[13]Cos117+y[13]y[13]Sin117) + L[9](s[19]Cos1110+y[19]y[19]Sin1110) )

p[23]= F[11]   - mhl[51] s[23]          + mhl[31] (gSin12
        - Rad(s[1]Cos12    + y[1]y[1]Sin12)  + L[0](s[1]Cos121    + y[1]y[1]Sin121) -
   L[1](s[3]Cos122+y[3]y[3]Sin122) - L[2](s[5]Cos123+y[5]y[5]Sin123) -
   L[6](s[13]Cos127+y[13]y[13]Sin127) )
```

jac :

```
b[1][0]   = 2mhl[20]RSin1 +mhl[20] gCos1 - L[0](
 mhl[21](s[3]Sin21 + y[3]y[3]Cos21)       +mhl[22](s[5]Sin31 + y[5]y[5]Cos31)
-mhl[23](s[7]Sin41 + y[7]y[7]Cos41)       -mhl[24](s[9]Sin51 + y[9]y[9]Cos51)
-mhl[25](s[11]Sin61 + y[11]y[11]Cos61)  +mhl[26](s[13]Sin71 + y[13]y[13]Cos71)
-mhl[27](s[15]Sin81 + y[15]y[15]Cos81)  -mhl[28](s[17]Sin91 + y[17]y[17]Cos91)
 mhl[29](s[19]Sin101 + y[19]y[19]Cos101)-mhl[20](s[21]Sin111 + y[21]y[21]Cos111)
+mhl[31](s[23]Sin121 + y[23]y[23]Cos121))


b[1][2]   = -mhl[21]( Rad(s[3]Sin2 + y[3]y[3]Cos2) - L[0](s[3]Sin21 + y[3]y[3]Cos21))
b[1][3]   = -2 mhl[21]y[3](RadSin2 - L[0]Sin21)
b[1][4]   = -mhl[22]( Rad(s[5]Sin3 + y[5]y[5]Cos3) - L[0](s[5]Sin31 + y[5]y[5]Cos31))
b[1][5]   = -2 mhl[22]y[5](RadSin3 - L[0]Sin31)
b[1][6]   = mhl[23]( Rad(s[7]Sin4 + y[7]y[7]Cos4) - L[0](s[7]Sin41 + y[7]y[7]Cos41))
b[1][7]   = 2 mhl[23]y[7](RadSin4 - L[0]Sin41)
b[1][8]   = mhl[24]( Rad(s[9]Sin5 + y[9]y[9]Cos5) - L[0](s[9]Sin51 + y[9]y[9]Cos51))
b[1][9]   = 2 mhl[24]y[9](RadSin5 - L[0]Sin51)
b[1][10]  = mhl[25]( Rad(s[11]Sin6 + y[11]y[11]Cos6) -
                                  L[0](s[11]Sin61 + y[11]y[11]Cos61))
b[1][11]  = 2 mhl[25]y[11](RadSin6 - L[0]Sin61)
b[1][12]  = -mhl[26]( Rad(s[13]Sin7 + y[13]y[13]Cos7) -
                                  L[0](s[13]Sin71 + y[13]y[13]Cos71))
b[1][13]  = -2 mhl[26]y[13](RadSin7 - L[0]Sin71)
b[1][14]  = mhl[27]( Rad(s[15]Sin8 + y[15]y[15]Cos8) -
                                  L[0](s[15]Sin81 + y[15]y[15]Cos81))
b[1][15]  = 2 mhl[27]y[15](RadSin8 - L[0]Sin81)
b[1][16]  = mhl[28]( Rad(s[17]Sin9 + y[17]y[17]Cos9) -
                                  L[0](s[17]Sin91 + y[17]y[17]Cos91))
b[1][17]  = 2 mhl[28]y[17](RadSin9 - L[0]Sin91)
b[1][18]  = mhl[29]( Rad(s[19]Sin10 + y[19]y[19]Cos10) -
                                  L[0](s[19]Sin101 + y[19]y[19]Cos101))
b[1][19]  = 2 mhl[29]y[19](RadSin10 - L[0]Sin101)
b[1][20]  = mhl[20]( Rad(s[21]Sin11 + y[21]y[21]Cos11) -
                                  L[0](s[21]Sin111 + y[21]y[21]Cos111))
b[1][21]  = 2 mhl[30]y[21](RadSin11 - L[0]Sin111)
b[1][22]  = -mhl[31]( Rad(s[23]Sin12 + y[23]y[23]Cos12) -
                                  L[0](s[23]Sin121 + y[23]y[23]Cos121))
b[1][23]  = -2 mhl[31]y[23](RadSin12 - L[0]Sin121)


b[3][0] =       -mhl[21]L[0](s[1]Sin21 - y[1]y[1]Cos21)
b[3][1] =       -2 mhl[21]y[3]L[0]Sin21
b[3][4] =       mhl[22] L[1] (s[5]Sin32+y[5]y[5]Cos32)
b[3][5] =       2.0 mhl[22] L[1] y[5] Sin32
b[3][6] =       -mhl[23] L[1] (s[7]Sin42+y[7]y[7]Cos42)
b[3][7] =       -2.0 mhl[23] L[1] y[7] Sin42
b[3][8] =       -mhl[24] L[1] (s[9]Sin52+y[9]y[9]Cos52)
b[3][9] =       -2.0 mhl[24] L[1] y[9] Sin52
b[3][10] =      -mhl[25] L[1] (s[11]Sin62+y[11]y[11]Cos62)
b[3][11] =      -2.0 mhl[25] L[1] y[11] Sin62
b[3][12] =      mhl[26] L[1] (s[13]Sin72+y[13]y[13]Cos72)
b[3][13] =      2.0 mhl[26] L[1] y[13] Sin72
```

b[3][14] =	-mhl[27] L[1] (s[15]Sin82+y[15]y[15]Cos82)
b[3][15] =	-2.0 mhl[27] L[1] y[15] Sin82
b[3][16] =	-mhl[28] L[1] (s[17]Sin92+y[17]y[17]Cos92)
b[3][17] =	-2.0 mhl[28] L[1] y[17] Sin92
b[3][18] =	-mhl[29] L[1] (s[19]Sin102+y[19]y[19]Cos102)
b[3][19] =	-2.0 mhl[29] L[1] y[19] Sin102
b[3][20] =	-mhl[30] L[1] (s[21]Sin112+y[21]y[21]Cos112)
b[3][21] =	-2.0 mhl[30] L[1] y[21] Sin112
b[3][22] =	mhl[31] L[1] (s[23]Sin122+y[23]y[23]Cos122)
b[3][23] =	2.0 mhl[31] L[1] y[23] Sin122
b[3][2]=	mhl[21] gCos2 - b[3][0] - b[3][4] - b[3][6] - b[3][8] - b[3][10] - b[3][12] - b[3][14] - b[3][16] - b[3][18] - b[3][20] - b[3][22]
b[5][0] =	-mhl[22]L[0](s[1]Sin31 - y[1]y[1]Cos31)
b[5][1] =	-2 mhl[22]y[1]L[0]Sin31
b[5][2]=	-mhl[23] L[1] (s[3]Sin32-y[3]y[3]Cos32)
b[5][3]=	-2.0 mhl[23] L[1] y[3] Sin32
b[5][6]=	-mhl[23] L[2] (s[7]Sin43+y[7]y[7]Cos43)
b[5][7]=	-2.0 mhl[23] L[2] y[7] Sin43
b[5][8]=	-mhl[24] L[2] (s[9]Sin53+y[9]y[9]Cos53)
b[5][9]=	-2.0 mhl[24] L[2] y[9] Sin53
b[5][10]=	-mhl[25] L[2] (s[11]Sin63+y[11]y[11]Cos63)
b[5][11]=	-2.0 mhl[25] L[2] y[11] Sin63
b[5][12]=	mhl[26] L[2] (s[13]Sin73+y[13]y[13]Cos73)
b[5][13]=	2.0 mhl[26] L[2] y[13] Sin73
b[5][14]=	-mhl[27] L[2] (s[15]Sin83+y[15]y[15]Cos83)
b[5][15]=	-2.0 mhl[27] L[2] y[15] Sin83
b[5][16]=	-mhl[28] L[2] (s[17]Sin93+y[17]y[17]Cos93)
b[5][17]=	-2.0 mhl[28] L[2] y[17] Sin93
b[5][18]=	-mhl[29] L[2] (s[19]Sin103+y[19]y[19]Cos103)
b[5][19]=	-2.0 mhl[29] L[2] y[19] Sin103
b[5][20]=	-mhl[30] L[2] (s[21]Sin113+y[21]y[21]Cos113)
b[5][21]=	-2.0 mhl[30] L[2] y[21] Sin113
b[5][22]=	mhl[31] L[2] (s[23]Sin123+y[23]y[23]Cos123)
b[5][23]=	2.0 mhl[31] L[2] y[23] Sin123
b[5][4]=	mhl[22] g Cos3 - b[5][0] - b[5][2] - b[5][6] - b[5][8] - b[5][10] - b[5][12] - b[5][14] - b[5][16] - b[5][18] - b[5][20] - b[5][22]
b[7][0] =	mhl[23]L[0](s[1]Sin41 - y[1]y[1]Cos41)
b[7][1] =	2 mhl[23]y[1]L[0]Sin41
b[7][2]=	mhl[23] L[1] (s[3]Sin42-y[3]y[3]Cos42)
b[7][3]=	2.0 mhl[23] L[1] y[3] Sin42
b[7][4]=	mhl[23] L[2] (s[5]Sin43-y[5]y[5]Cos43)
b[7][5]=	2.0 mhl[23] L[2] y[5] Sin43
b[7][8]=	mhl[24] L[3] (s[9]Sin54+y[9]y[9]Cos54)
b[7][9]=	2.0 mhl[24] L[3] y[9] Sin54
b[7][10]=	mhl[25] L[3] (s[11]Sin64+y[11]y[11]Cos64)
b[7][11]=	2.0 mhl[25] L[3] y[11] Sin64
b[7][6]=	=mhl[23] g Cos4 - b[7][0] - b[7][2] - b[7][4] - b[7][8] - b[7][10]

b[9][0] =	mhl[24]L[0](s[1]Sin51 - y[1]y[1]Cos51)
b[9][1] =	2 mhl[24]y[1]L[0]Sin51
b[9][2]=	mhl[24] L[1] (s[3]Sin52-y[3]y[3]Cos52)
b[9][3]=	2.0 mhl[24] L[1] y[3] Sin52
b[9][4]=	mhl[24] L[2] (s[5]Sin53-y[5]y[5]Cos53)
b[9][5]=	2.0 mhl[24] L[2] y[5] Sin53
b[9][6]=	-mhl[24] L[3] (s[7]Sin54-y[7]y[7]Cos54)
b[9][7]=	-2.0 mhl[24] L[3] y[7] Sin54
b[9][10]=	mhl[25] L[4] (s[11]Sin65+y[11]y[11]Cos65)
b[9][11]=	2.0 mhl[25] L[4] y[11] Sin65
b[9][10]=	-mhl[24] g Cos5 - b[9][0] - b[9][2] - b[9][4] - b[9][6] - b[9][10]
b[11][0] =	mhl[25]L[0](s[1]Sin61 - y[1]y[1]Cos61)
b[11][1] =	2 mhl[25]y[1]L[0]Sin61
b[11][2]=	mhl[25] L[1] (s[3]Sin62-y[3]y[3]Cos62)
b[11][3]=	2.0 mhl[25] L[1] y[3] Sin62
b[11][4]=	mhl[25] L[2] (s[5]Sin63-y[5]y[5]Cos63)
b[11][5]=	2.0 mhl[25] L[2] y[5] Sin63
b[11][6]=	-mhl[25] L[3] (s[7]Sin64-y[7]y[7]Cos64)
b[11][7]=	-2.0 mhl[25] L[3] y[7] Sin64
b[11][8]=	-mhl[25] L[4] (s[9]Sin65-y[9]y[9]Cos65)
b[11][9]=	-2.0 mhl[25] L[4] y[9] Sin65
b[11][10]=	-mhl[25] g Cos6) - b[11][0] - b[11][2] - b[11][4] - b[11][6] - b[11][8]
b[13][0] =	-mhl[26]L[0](s[1]Sin71 - y[1]y[1]Cos71)
b[13][1] =	-2 mhl[26]y[1]L[0]Sin71
b[13][2]=	-mhl[26] L[1] (s[3]Sin72-y[3]y[3]Cos72)
b[13][3]=	-2.0 mhl[26] L[1] y[3] Sin72
b[13][4]=	-mhl[26] L[2] (s[5]Sin73-y[5]y[5]Cos73)
b[13][5]=	-2.0 mhl[26] L[2] y[5] Sin73
b[13][14]=	-mhl[27] L[6] (s[15]Sin87+y[15]y[15]Cos87)
b[13][15]=	-2.0 mhl[27] L[6] y[15] Sin87
b[13][16]=	-mhl[28] L[6] (s[17]Sin97+y[17]y[17]Cos97)
b[13][17]=	-2.0 mhl[28] L[6] y[17] Sin97
b[13][18]=	-mhl[29] L[6] (s[19]Sin107+y[19]y[19]Cos107)
b[13][19]=	-2.0 mhl[29] L[6] y[19] Sin107
b[13][20]=	-mhl[30] L[6] (s[21]Sin117+y[21]y[21]Cos117)
b[13][21]=	-2.0 mhl[30] L[6] y[21] Sin117
b[13][22]=	mhl[31] L[6] (s[23]Sin127+y[23]y[23]Cos127)
b[13][23]=	2.0 mhl[31] L[6] y[23] Sin127
b[13][12]=	mhl[26] g Cos7 - b[13][0] - b[13][2] - b[13][4] - b[13][14] - b[13][16] - b[13][18] - b[13][20] - b[13][22]
b[15][0] =	mhl[27]L[0](s[1]Sin81 - y[1]y[1]Cos81)
b[15][1] =	2 mhl[27]y[1]L[0]Sin81
b[15][2]=	mhl[27] L[1] (s[3]Sin82-y[3]y[3]Cos82)
b[15][3]=	2.0 mhl[27] L[1] y[3] Sin82
b[15][4]=	mhl[27] L[2] (s[5]Sin83-y[5]y[5]Cos83)
b[15][5]=	2.0 mhl[27] L[2] y[5] Sin83
b[15][12]=	mhl[27] L[6] (s[13]Sin87-y[13]y[13]Cos87)
b[15][13]=	2.0 mhl[27] L[6] y[13] Sin87

b[15][16]=	mhl[28] L[7] (s[17]Sin98+y[17]y[17]Cos98)
b[15][17]=	2.0 mhl[28] L[7] y[17] Sin98
b[15][14]=	-mhl[27] g Cos8 - b[15][0] - b[15][2] - b[15][4] - b[15][12] - b[15][16]
b[17][0] =	mhl[28]L[0](s[1]Sin91 - y[1]y[1]Cos91)
b[17][1] =	2 mhl[28]y[1]L[0]Sin91
b[17][2]=	mhl[28] L[1] (s[3]Sin92-y[3]y[3]Cos92)
b[17][3]=	2.0 mhl[28] L[1] y[3] Sin92
b[17][4]=	mhl[28] L[2] (s[5]Sin93-y[5]y[5]Cos93)
b[17][5]=	2.0 mhl[28] L[2] y[5] Sin93
b[17][12]=	mhl[28] L[6] (s[13]Sin97-y[13]y[13]Cos97)
b[17][13]=	2.0 mhl[28] L[6] y[13] Sin97
b[17][14]=	-mhl[28] L[7] (s[15]Sin98-y[15]y[15]Cos98)
b[17][15]=	-2.0 mhl[28] L[7] y[15] Sin98
b[17][16]=	-mhl[28] g Cos9 - b[17][0] - b[17][2] - b[17][4] - b[17][12] - b[17][14]
b[19][0] =	mhl[29]L[0](s[1]Sin101 - y[1]y[1]Cos101)
b[19][1] =	2 mhl[29]y[1]L[0]Sin101
b[19][2]=	mhl[29] L[1] (s[3]Sin102-y[3]y[3]Cos102)
b[19][3]=	2.0 mhl[29] L[1] y[3] Sin102
b[19][4]=	mhl[29] L[2] (s[5]Sin103-y[5]y[5]Cos103)
b[19][5]=	2.0 mhl[29] L[2] y[5] Sin103
b[19][12]=	mhl[29] L[6] (s[13]Sin107-y[13]y[13]Cos107)
b[19][13]=	2.0 mhl[29] L[6] y[13] Sin107
b[19][20]=	mhl[30] L[9] (s[21]Sin1110+y[21]y[21]Cos1110)
b[19][21]=	2.0 mhl[30] L[9] y[21] Sin1110
b[19][18]=	-mhl[29] g Cos10 - b[19][0] - b[19][2] - b[19][4] - b[19][12] - b[19][20]
b[21][0] =	mhl[30]L[0](s[1]Sin111 - y[1]y[1]Cos111)
b[21][1] =	2 mhl[30]y[1]L[0]Sin111
b[21][2]=	mhl[30] L[1] (s[3]Sin112-y[3]y[3]Cos112)
b[21][3]=	2.0 mhl[30] L[1] y[3] Sin112
b[21][4]=	mhl[30] L[2] (s[5]Sin113-y[5]y[5]Cos113)
b[21][5]=	2.0 mhl[30] L[2] y[5] Sin113
b[21][12]=	mhl[30] L[6] (s[13]Sin117-y[13]y[13]Cos117)
b[21][13]=	2.0 mhl[30] L[6] y[13] Sin117
b[21][18]=	-mhl[30] L[9] (s[19]Sin1110-y[19]y[19]Cos1110)
b[21][19]=	-2.0 mhl[30] L[9] y[19] Sin1110
b[21][20]=	-mhl[30] g Cos11 - b[21][0] - b[21][2] - b[21][4] - b[21][12] - b[21]18]
b[23][0] =	-mhl[31]L[0](s[1]Sin121 - y[1]y[1]Cos121)
b[23][1] =	-2 mhl[31]y[1]L[0]Sin121
b[23][2]=	mhl[31] L[1] (s[3]Sin122-y[3]y[3]Cos122)
b[23][3]=	2.0 mhl[31] L[1] y[3] Sin122
b[23][4]=	mhl[31] L[2] (s[5]Sin123-y[5]y[5]Cos123)
b[23][5]=	2.0 mhl[31] L[2] y[5] Sin123
b[23][12]=	mhl[31] L[6] (s[13]Sin127-y[13]y[13]Cos127)
b[23][13]=	2.0 mhl[31] L[6] y[13] Sin127

b[23][22]= mhl[31] g Cos12 - b[23][0] - b[23][2] - b[23][4] - b[23][12]