Some aspects of the use of MADYMO in general biomechanics analysis

Citation for published version (APA):

Document status and date:
Published: 01/01/1989

Publisher Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:
• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license above, please follow below link for the End User Agreement:
www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:
openaccess@tue.nl
providing details and we will investigate your claim.

Download date: 20. Oct. 2020
SOME ASPECTS OF THE USE OF MADYMO IN GENERAL BIOMECHANICS ANALYSIS

PART I: ORGANIZATION

Ir. D.K.I. Hoekstra

december 1989
SOME ASPECTS OF THE USE OF MADYMO IN GENERAL BIOMECHANICS ANALYSIS

PART I : ORGANIZATION

Ir. D.K.I. HOEKSTRA

DELFT, DECEMBER 1989
CONTENTS

Preface 1

Chapter 1. Introduction 2

1.1 What is a combination project? 2
1.2 The project at the TNO Road-Vehicles Research Institute 3
1.3 Overview of the contents of the report 4

Chapter 2. Project management 5

2.1 Introduction 5
2.2 Overall management of the project at TNO 6
2.3 Interaction with TNO 7
2.3.1 TNO management 7
2.3.2 TNO staff 7
2.4 Interaction with the Eindhoven University of Technology 7
2.4.1 TUE management 7
2.4.2 TUE staff and students 8
2.5 External relations 8

Chapter 3. Discussion, conclusion and recommendations 9

References 12
In October, 1987, I started my study in the postgraduate program "Mathematics in Industry" at the Eindhoven University of Technology (TUE). The program participates in the European Consortium for Mathematics in Industry (ECMI). In modern industry, mathematical models play an increasingly important role in research and development for production, distribution and management. The central aim of the program is to provide mathematicians and other university graduates in mathematically oriented fields with the training, necessary to work successfully as a mathematician in industry. A secondary aim of the program is to establish a European network of groups that covers the whole area of mathematical applications in industry and to use that network to stimulate the use of mathematics in industry. Each student concludes the postgraduate program with a final project in an industrial environment. In my case this environment was the Netherlands Organization for Applied Scientific Research (TNO) in Delft. There, at the Injury Prevention Department of the TNO Road-Vehicles Research Institute, the Crash Victim Simulation program MADYMO is developed. MADYMO stands for MAThematical DYnamic MOdelling. This computer program can simulate occupant behavior in car crashes.

I was leader of a so-called combination project in which also two graduate students took part. The organization of the project is described in this part (i.e. Part I) of the final report. The technical realization of the project and my individual project contribution on muscle modelling can be found in Part II and Part III of the report, respectively. The first graduate student (Pieter Vosbeek) has written a Master's thesis in which he presents two contributions to the modelling of the human body in MADYMO. The second student (Rob Vermazeren) is still working on MADYMO pre- and postprocessing facilities. His Master's thesis will appear in the beginning of 1990.

I thank Pieter and Rob for their efforts and the pleasant cooperation. On behalf of both of them I further thank their supervisors Fons van de Ven and Kees van Overveld and TNO staff Willy Koppens and Harry Heinen for their help. The project as a whole was supervised by Jos Jansen and Jac Wismans, whom I thank for their suggestions and for the fruitful discussions that we have had. A final acknowledgment goes to the TNO Road-Vehicles Research Institute, that provided the opportunity to carry out the first combination project of this type and that was so kind to employ me for an additional three months, so that I could extend this interesting project.
CHAPTER 1
INTRODUCTION

1.1 What is a combination project?

In September 1987 the postgraduate program "Mathematics in Industry" was started at the Eindhoven University of Technology. Each participant in the program concludes the two year training with a project in an industrial environment. In this final project, which lasts at least six months, the postgraduate student is expected to design or extend a mathematical model for a (problem) situation in a company, to solve the problem using mathematical techniques and to interpret the results. The final project can be an individual project, in which the postgraduate student carries out the research all by himself. It is also possible that this project is a combination project. The research in such a combination project is carried out by a team, which consists of

- the postgraduate student,
- two or more graduate students,
- supervisors of the students,
- one staff member of the Eindhoven University of Technology
- one staff member of the company involved.

The postgraduate student acts as the project leader and is responsible for the progress of the project and the reporting of the project results. The graduate students (not necessarily all from the Eindhoven University of Technology) work on parts of the project that are related to the field of their studies. Some of them are in the Department of Mathematical Engineering, others are from different disciplines, like computing science, mechanical or electrical engineering, etcetera. Each student is being supervised by a faculty staff member. The project contributions of the students yield as their final projects needed for graduation. The staff member of the Eindhoven University of Technology supervises the project as a whole. He gives advise and he makes sure that scientific standards are maintained. Further, he consults with the faculty members that act as supervisors of the students. The company staff member supplies the topic(s) of the research and may adapt the topic(s) in the course of the project. At the end, he judges whether the project results are useful to the company.

The project can be carried out at the company or at the university, depending on the facilities that are available. In the case of a combination project, it is also possible that one or more participants are located at the company and others are at the university. The company and the university enter into a contract, in which costs, confidentiality and other conditions are specified.

About twelve years ago the first combination projects were
started in the United States of America (University of Clairmont). The projects are called clinics in the United States. They have been very successful. Companies pay approximately sixty thousand dollars for each clinic. A few years ago clinics were first introduced in Europe (Trondheim and Bari). The combination project at the TNO Road-Vehicles Research Institute, which started in March of this year, is the first clinic in the Netherlands.

1.2 The project at the TNO Road-Vehicles Research Institute

At the Injury Prevention Department of the TNO Road-Vehicles Research Institute the computer program MADM0 is developed for simulations of occupant behavior in car crashes. In other words, MADM0 (MAthematical DYnamic MOdelling) is a Crash Victim Simulation program. Two- and three-dimensional simulations are possible with the 2D and 3D version of MADM0. In MADM0 the human body is modelled as a system of rigid bodies, which are connected by joints. The purpose of this research was to find out to what extent MADM0 is applicable in general biomechanics analysis (e.g. in sports or working postures studies). Therefore, a more realistic modelling of the human body in MADM0 had to be achieved. The research had to be facilitated by using improved pre- and postprocessing features of MADM0, which had to be developed in the project.

Postgraduate student ir. D.K.I. Hoekstra, student of mathematics P.H.J. Vosbeek and computing science student R.C.J.G. Vermazeren took part in the combination project at TNO. Hoekstra was project leader. He has conducted a literature study on muscle modelling and has developed subroutines for MADM0 that allow the user to prescribe forces and torques that are caused by muscles and act on body segments. Vosbeek has written a computer program (GEBMAD) that converts output from the anthropometry generator GEBOD into MADM0 input. This way accurate body description data for human bodies of arbitrary shape and size can be used in the model (anthropometry - the study of human body dimensions). Further, he has developed MADM0 routines allowing certain parts of the human body to be represented by massless deformable bodies. Vermazeren’s postprocessor can graphically illustrate the motion of the human body model. At the moment he is working on a MADM0 preprocessor, that can be used to position the human body model in its environment. The individual contributions to the project are described in Hoekstra (1989b, Part III of this report), Vosbeek (1989) and Vermazeren (1989 and 1990).

The project as a whole was supervised by dr. ir. J.K.M. Jansen of the Eindhoven University of Technology (TUE) and prof. dr. ir. J.S.H.M. Wismans of the TNO Road-Vehicles Research Institute. Vosbeek was supervised by dr. ir. A.A.F. van de Ven, who is a faculty member of the Department of Mathematical Engineering (TUE). Dr. ir. C.W.A.M. van Overveld of the Department of Computing Science (TUE) acted as supervisor of Vermazeren. Both students did their work at the Eindhoven University of
Technology, while Hoekstra fulfilled his part of the project at TNO in Delft.

The project serves as a test case for both TNO and the Eindhoven University of Technology to see how a combination project performs in practice. In this part of the report (on the organization of the project) the benefits and problems that we have experienced will be discussed. Recommendations are given that may be useful in future combination projects. The postgraduate program "Mathematics in Industry" at the Eindhoven University of Technology takes part in the European Consortium for Mathematics in Industry (ECMI). The intention of this report is also to help those ECMI members that want to participate in a combination project in the future.

1.3 Overview of the contents of the report

In the next chapter, first some general aspects of the management of a project are given. Then, the activities that were involved with leading the project at TNO will be described. Hereby, a distinction is made between activities regarding the overall management of the project and activities specifically concerned with TNO, the Eindhoven University of Technology (TUE) and external relations.

In the final chapter the problems experienced in the combination project at TNO will be described and recommendations are given that may help to avoid those problems in the future. Further, the benefits of a project like this will be described.
CHAPTER 2
PROJECT MANAGEMENT

2.1 Introduction

Before describing some aspects of leading the combination project at TNO, I will discuss some general aspects of project management. One can distinguish between two sorts of project activities: activities concerned with the content of the project and control activities. The first type consists of activities that are necessary to establish the desired project results. The second type controls the activities of the first type.

It is believed (Wijnen et al., 1984) that there are five groups of control activities: control of time, costs, quality, information and organization. These groups are defined as follows:

- **time control**: to make sure all project activities can be carried out in time, so that the desired project results are achieved on schedule.

- **costs control**: to make sure all project activities are carried out in a financial sensible way, so that the project results are profitable.

- **quality control**: to make sure all project activities can be carried out well (purposeful), so that the project result is of good quality.

- **information control**: to make sure all project activities can be carried out unambiguously, so that the project results will be unambiguous and reproducible.

- **organization control**: to make sure it is possible for the responsible and authorized persons to carry out all project activities, so that the project results will be formally accepted.

The end time and the budget of the project have to be specified (and approved of) before the project starts. The demands on the quality of the project results have to be testable, so that at the end of the project it can be concluded, whether those demands were met or not. The information is focussed on the technical project aspects, and has to be identified, registered and distributed to all persons involved in the project. Control of information is useful in case of changes in the aims, demands or constraints of the project. Control of organization encompasses division of tasks, authorization and responsibility (both in relation to the existing company organization and within the project), maintaining means of communication, and the definition of the process of decision-making.

The activities concerned with the content of the project can be
found in Part II and Part III of this report and in Vosbeek (1989) and Vermazeren (1989 and 1990). The control activities that I have carried out, being the leader of the project, are described in this part of the report (i.e. Part I). First, the activities concerned with the overall management of the project will be discussed (section 2.2). Then, I will describe the activities regarding interactions between myself (being the project leader) and the other participants in the project. A distinction is made between interactions concerning TNO (section 2.3) and interactions concerning the Eindhoven University of Technology (section 2.4). Finally, activities concerned with external relations will be discussed (section 2.5). All control activities can be classified in one of the five groups mentioned above.

2.2 Overall management of the project at TNO

Before the project started, the time-schedule for the different research activities has been discussed. It was decided that the project would start on March 1, 1989, and that it would end on January 1, 1990. Hoekstra would participate in the project from March 1 until October 1. He would spend the first three months of his research on a literature study on muscle modelling and the rest of his time on the incorporation of muscle activity in MADYMO and on the reporting of the project results. Vosbeek would start his project activities on March 1 and he would finish his work on August 31. He would write an interface between the anthropometry generator GEBOD and MADYMO. Further, he would incorporate massless deformable bodies in MADYMO. On each of the two project contributions he would work three months. Vermazeren would work part-time on the project from March 1 until June 1 and he would work full-time on the project from July 1, 1989, until January 1, 1990. He would spend seven months on improving MADYMO pre- and postprocessing facilities (the first month being spread over March, April and May).

This time-schedule has been adapted during the project. I have worked on the project for a period of ten instead of seven months, so I could spend more time on each of my activities. Vosbeek has carried out his work according to the time-schedule explained above. Vermazeren, however, has started his six-month graduation assignment two months later than was planned initially. Therefore, he will not finish his project contribution before March 1, 1990.

In the contract between TNO and the Eindhoven University of Technology, it was specified which of the two parties was responsible for certain costs involved with the various project activities. During the project no problems raised as far as the financial aspects of the project were concerned. Therefore, I have not carried out any noticeable cost control activities.

The organization of the project was established before the start of the project. Tasks, authorization and responsibilities are described in Chapter 1.
2.3 Interaction with TNO

2.3.1 TNO management

I have provided a description of the project, which was included in the contract between TNO and the TUE. The description contained information on the tasks, authorization and responsibilities of the participants in the project and on the time-schedule of the various research activities in the project.

Funds had to be arranged to finance my extra stay of three months at TNO. Therefore, I have submitted a request for a grant for extension of the project to the board of the TNO Road-Vehicles Research Institute, in which I have explained background, set-up, contents, estimated costs and duration of the project.

TNO management has provided the topics that were to be investigated. We have discussed the quality demands for the project results. These demands have been adapted a few times, because during the project new insights were gained on their feasibility. Approximately every two weeks I have informed the management on the progress of the research carried out by me and by the two students. In these meetings the direction in which the research activities should continue were also discussed.

2.3.2 TNO staff

TNO staff members have supplied part of the expertise to be used in the research. Moreover, they have assisted in formulating the tasks and demands concerning the individual project contributions. I have informed them on project results that were of interest to them. Further, I have organized and attended meetings between TNO staff and the students and their supervisors in which the progress of the research was discussed.

2.4 Interaction with the Eindhoven University of Technology

2.4.1 TUE management

The combination project at TNO was a test case for the postdoctoral program "Mathematics in Industry". I have informed the supervisor of the program (prof. dr. P.L. Cijsouw) regularly on the functionality of the combination project. On his request I have described my experiences concerning the management of the project in this part of the report. Further, I was asked to give some recommendations that might avoid future problems in organizing a project like this.

Another aspect of my interaction with TUE management was helping to establish the contract between TNO and the TUE in which costs, confidentiality and other conditions were laid down.
2.4.2 TUE staff and students

The part of the research that had to be carried out by the students was established in meetings between TNO staff, the students and their supervisors and me. TNO staff provided the assignment for the student and the supervisor judged, whether this assignment was appropriate and whether it could be carried out in time. Later, similar meetings were held to discuss the progress of the research. In mutual agreement the research activities could be adapted during the project. My role in this was that of a contact, who also gives advise.

Both students needed to work with MADYMO, so we have installed the program at the Eindhoven University of Technology. I helped the students, when they had problems concerning the use of MADYMO. Further, I tried to motivate them, I edited their written reports and I made sure that they did what they were supposed to do. Moreover, I was a member of the examination board that judged the work of Vosbeek, who graduated in August.

Regularly, I have discussed the progress of the research carried out by the students and me with the supervisor of the project as a whole (Jansen). Furthermore, we have discussed the (scientific) quality of the project results that are presented in Part II of this report.

2.5 External relations

In March Wismans (of TNO) and I have visited the Department of Mechanical Engineering of the University of Twente, where we discussed the participation of one of their students in the combination project. Unfortunately, no student was available at the time.

Vosbeek has used the anthropometry generator GEBOD in his research. I ordered the program from Wright-Patterson Air Force Base in Ohio and helped Vosbeek to get it running on the SUN at the Eindhoven University of Technology. Further, I have supplied GEBOD output to A. Huson of the University of Leiden. He wanted to compare the anthropometric data rendered by GEBOD with body description data obtained by means of the Magnetic Resonance Imaging (MRI) technique. This way, we could get a validation of GEBOD, which is important, because the anthropometry generator will be part of MADYMO preprocessing. However, his measurements were not done yet, so we could not compare results within the framework of this project.

Vermazeren’s discipline, computer graphics, was new to me. In order to get some feeling for the subject, I visited SCAN (Stichting Computer Animatie Nederland). SCAN represents the state of the art as far as computer animation in the Netherlands is concerned.
CHAPTER 3
DISCUSSION, CONCLUSION AND RECOMMENDATIONS

The management task has taken a considerable amount of my time. Approximately once every two weeks there was a meeting at the Eindhoven University of Technology, where I discussed the progress of the research with Jansen and Wismans or with the students and their supervisors. Further, all aspects of the supervision of the students took its time (even after Vosbeek’s graduation we have still worked on improvements of the theory in his report and on the MADYMO subroutines that he had written). Finally, I have spent more time than expected on writing the final report on the organization and the technical aspects of the project as a whole and on my individual contribution to the project. This fact should not be underestimated by future leaders of a combination project.

When one of the students is writing his final report, it should be anticipated by the project leader, that he will have to dedicate time to the student. He has to read the manuscript and to assist the student whenever possible. Moreover, he has to be present at the student’s oral presentation of the results and he has to be a member of the examination committee, that judges the work of the student.

Since I am a mathematical engineer myself I had no difficulties advising Vosbeek, whose work was in the field of mechanics. Vermazeren’s discipline, on the other hand, was new to me. When one knows the field of one of the participants in the project well, danger exists that one is tempted to get too much involved and that one goes into too much detail trying to solve the other’s problems. This situation should be avoided. Everyone in the project should stick to his own tasks and cooperation should consist of giving suggestions, not of explicitly solving problems. Maybe I have made that mistake when helping Vosbeek. In his work I have been much more involved than in the work of Vermazeren.

This project has shown, that in practice it is very difficult to organize a combination project. It requires a postgraduate student and two or more graduate students (preferably all from different disciplines) to be available during the same period of time. Further, there must be a company that is interested in a combination project and that provides topics of possible research that appeal to all of the students. Hoekstra and Vosbeek have finished their part of the research, while Vermazeren is still working on his part. His work could therefore not be used in the project and his results could not be included in Part II of this report.

In the project at TNO the time-sharing problem was anticipated by defining the tasks of the three persons carrying out the research in such manner that these tasks could be done more or less independently. This autonomy is desired, because that way
the students do not depend on the results of the other participants in the project. They can work on their own project contribution and write their results in a report. It is the task of the project leader to integrate the various individual project contributions.

In the last six months of the two-year program "Mathematics in Industry" the postgraduate student has to fulfill his final project in industry. In case of a combination project, I believe that six months is too short. The graduate students spend approximately six months on their final assignment. Therefore, the postgraduate student should be available at least nine months to lead the project. That way, there is time for a good (one month?) preparation of the project and at the end all project results can be integrated. Moreover, all project results and their integration can be included in the final report.

There are two ways to extend the postgraduate student's involvement in the project. First, the postgraduate program may be changed in order to allow the student to spend nine instead of six months on his final project. I do not think that it will be harmful to reduce the curriculum with a few months. In the three additional months that I have worked on the project I have learned more than I would have learned doing courses at university. The second possibility is that the company provides means to extend the project with a certain amount of time. This was done in my case, since TNO employed me for an additional three months.

As a final recommendation I would like to advise every postgraduate student, who will be leading a combination project, to take a course in project management at university or to read books on the subject (e.g. Wijnen et al., 1984). Maybe this should be compulsory in the curriculum of the postgraduate program "Mathematics in Industry".

This chapter will be concluded with a discussion on the benefits of a combination project for the postgraduate student, the university and the company.

The postgraduate student gets experience in leading a project and in working in a team of specialists from different disciplines. His communicative skills are developed, due to the many interactions of the project leader with the rest of the participants in the project. Examples of such interactions were described in the previous chapter.

For the Eindhoven University of Technology it is useful to know, that in the project at TNO the cooperation between the postgraduate student and the two graduate students has worked out well. This means, that, in the future, postgraduate students can assist faculty staff in supervising graduate students, who are working on their final assignment needed for graduation.

Also for the company the combination project is beneficial. Instead of a single (post) graduate student, a team consisting
of three or more persons is carrying out research that is relevant to the company. Thus, more work can be done in the same period of time. Besides, it would be more expensive for the company to engage the participants in the combination project individually.
REFERENCES

Some aspects of the use of MADYMO in general biomechanics analysis. Part II: Technical realization
Final report, postgraduate program "Mathematics in Industry", Eindhoven University of Technology

Some aspects of the use of MADYMO in general biomechanics analysis. Part III: Muscle modelling
Final report, postgraduate program "Mathematics in Industry", Eindhoven University of Technology

TNOPost: A postprocessor for MADYMO 3D
Report, Department of Computer Science, Eindhoven University of Technology

Vermazeren, R.C.J.G. (in preparation)
Title unknown (subject: a preprocessor for MADYMO 3D)
Master's thesis, Department of Computer Science, Eindhoven University of Technology

Two contributions to the modelling of the human body in MADYMO
Master's thesis, Department of Mathematical Engineering, Eindhoven University of Technology

Projectmatig werken
Het Spectrum B.V., Utrecht
SOME ASPECTS OF THE USE OF MADYMO IN GENERAL BIOMECHANICS ANALYSIS

PART II: TECHNICAL REALIZATION

ir D.K.I. Hoekstra

University supervisor: dr ir J.K.M. Jansen, Technische Universiteit, Eindhoven

Industrial supervisor: prof.dr ir J.S.H.M. Wismans, Instituut voor Wegtransportmiddelen TNO, Delft

December 1989
Pt. II: Technical realization. - Ill.
ISBN 90-5282-043-0
SISO 517.3 UDC [519.24:681.3.06]:67/68
Trefw. wiskunde voor de industrie / MADYMO (computerprogramma).


Niets uit deze uitgave mag worden vermenigvuldigd en/of openbaar gemaakt door middel van druk, fotokopie, microfilm of op welke andere wijze dan ook zonder voorafgaande schriftelijke toestemming van de auteur.

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, recording, or any information storage and retrieval system, without permission from the copyright owner.

ISBN 90 5282 043 0
## CONTENTS

**Preface**  
1

**Chapter 1. Introduction**  
2

**Chapter 2. Individual contributions to the project**  
3

2.1 Introduction  
3
2.2 Muscle modelling  
3
2.3 Anthropometry  
5
2.4 Massless deformable bodies in MADYMO  
7
2.5 Pre- and postprocessing of MADYMO  
8
2.5.1 Preprocessing  
8
2.5.2 Postprocessing  
9

**Chapter 3. Examples of MADYMO simulations using human body modelling**  
11

3.1 Introduction  
11
3.2 A dart-throwing arm  
11
3.2.1 The model  
11
3.2.2 The pulley model  
13
3.2.3 The MADYMO model  
13
3.2.4 Results  
15
3.3 A subject doing a push-up  
18
3.3.1 A simple model  
18
3.3.2 An improved model  
21
3.3.3 The MADYMO model  
22
3.3.4 Results  
22
3.4 A swinging leg  
27
3.4.1 The model  
27
3.4.2 The MADYMO model  
28
3.4.3 Results  
29

**Chapter 4. Discussion, conclusion and recommendations**  
32

4.1 Introduction  
32
4.2 Separate project contributions  
32
4.3 MADYMO simulations  
35
4.4 General conclusion  
36

**References**  
38

**Appendix A. MADYMO input files for the simulations in Chapter 3**  
40

A1. A dart-throwing arm  
40
A2. A subject doing a push-up  
42
A3. A swinging leg  
45
PREFACE

In October, 1987, I started my study in the postgraduate program "Mathematics in Industry" at the Eindhoven University of Technology (TUE). The program participates in the European Consortium for Mathematics in Industry (ECMI). In modern industry, mathematical models play an increasingly important role in research and development for production, distribution and management. The central aim of the program is to provide mathematicians and other university graduates in mathematically oriented fields with the training, necessary to work successfully as a mathematician in industry. A secondary aim of the program is to establish a European network of groups that covers the whole area of mathematical applications in industry and to use that network to stimulate the use of mathematics in industry. Each student concludes the postgraduate program with a final project in an industrial environment. In my case this environment was the Netherlands Organization for Applied Scientific Research (TNO) in Delft. There, at the Injury Prevention Department of the TNO Road-Vehicles Research Institute, the Crash Victim Simulation program MADYMO is developed. MADYMO stands for Mathematical Dynamic Modelling. This computer program can simulate occupant behavior in car crashes.

I was leader of a so-called combination project in which also two graduate students took part. The technical realization of the project is described in this part (i.e. Part II) of the final report. The organization of the project and my individual project contribution on muscle modelling can be found in Part I and Part III of the report, respectively. The first graduate student (Pieter Vosbeek) has written a Master's thesis in which he presents two contributions to the modelling of the human body in MADYMO. The second student (Rob Vermazeren) is still working on MADYMO pre- and postprocessing facilities. His Master's thesis will appear in the beginning of 1990.

I thank Pieter and Rob for their efforts and the pleasant cooperation. On behalf of both of them I further thank their supervisors Fons van de Ven and Kees van Overveld and TNO staff Willy Koppens and Harry Heinen for their help. The project as a whole was supervised by Jos Jansen and Jac Wismans, whom I thank for their suggestions and for the fruitful discussions that we have had. A final acknowledgment goes to the TNO Road-Vehicles Research Institute, that provided the opportunity to carry out the first combination project of this type and that was so kind to employ me for an additional three months, so that I could extend this interesting project.
CHAPTER 1

INTRODUCTION

At the Injury Prevention Department of the TNO Road-Vehicles Research Institute the computer program MADYMO has been developed for simulations of occupant behavior in car crashes. In other words, MADYMO (MAthematical DYnamic MOdelling) is a Crash Victim Simulation program. In MADYMO the human body is modelled as a system of rigid bodies (elements) which are connected by joints. The systems must be tree structures, which do not contain closed loops of elements. Two- and three-dimensional simulations are possible with the 2D and 3D version of MADYMO. In MADYMO 2D the connections (joints) between the elements are of the hinge type, while in MADYMO 3D ball and socket type joints are used to link the elements. In the MADYMO User's Manuals (1988) more detailed information about the program can be found.

The global purpose of this research was to find out to what extent MADYMO is applicable in general biomechanics analysis, like sports activities or studies of working postures. Two specific objectives were set that should allow us to be able to give a better judgement on the applicability of MADYMO in this field. The first objective was the development of an improved model of the human body in MADYMO. The improvements include the incorporation of muscle activity and massless deformable bodies and a more realistic description of body anthropometry. Muscle force will allow the model to move all by itself. Using an anthropometry generator will lead to more accurate body parameters, like length, mass and moments of inertia of body segments. Deformable bodies can be used to model certain flexible parts of the body, like the neck, more accurately. The second objective was the development of userfriendly pre- and postprocessing facilities for MADYMO. The postprocessor should be able to display the motion of the tree structure representing the body. The preprocessor should be able to position the tree structure in its environment and should simplify the finding of an initial equilibrium situation for the body.

In Chapter 2 of this report the several subprojects conducted in this study will be summarized, when muscle modelling (2.1), anthropometry (2.2), massless deformable bodies (2.3) and pre- and postprocessing results (2.4) are reviewed. Detailed presentations on these subprojects can be found in Part III of this report (muscle modelling), Vosbeek (1989) (anthropometry and massless deformable bodies) and Vermazern (1989, 1990) (pre- and postprocessing). Chapter 3 shows some examples of MADYMO simulations in which the improved model of the human body is used. Simulations of a dart-throwing arm, of a subject doing a push-up and of a swinging leg will be presented. In the final chapter the project results and recommendations for future research will be given.
CHAPTER 2
INDIVIDUAL CONTRIBUTIONS TO THE PROJECT

2.1 Introduction

In this chapter the results of the different project parts are described. The modelling of the human body in MADYMO and pre- and postprocessing facilities of MADYMO were investigated. As far as human body modelling is concerned, muscle activity, anthropometry and massless deformable bodies were incorporated in MADYMO. Further, a prototype of a postprocessor for MADYMO has been developed, whereas a prototype of a MADYMO preprocessor is still being developed.

The project was carried out by three persons. Two of them have written a report in which the details of their contribution to the project are described, while the third person is still preparing his report. Hoekstra has done a literature study on muscle modelling and has developed user-defined subroutines that allow the user to prescribe time-dependent (muscle) forces and torques in MADYMO (see Part III of the report). Vosbeek (1989) has developed an interface between an anthropometry generator (GEBOD) and MADYMO that allows the MADYMO user to model a human body of arbitrary size and shape. Moreover, he has written user-defined subroutines that allow the introduction of massless deformable bodies in MADYMO. Some parts of the human body, like the neck and the spine, can be modelled more precisely using these deformable bodies. Vermazeren (1989) has developed a prototype of a MADYMO3D postprocessor with which simulation results can be visualized. At the moment he is working on a preprocessor for MADYMO3D that enables the user to interactively manipulate a model of the human body in its environment (See Vermazeren, 1990).

2.2 Muscle modelling

The purpose of this study was to improve the modelling of the human body in MADYMO, in order to make the computer simulation program suitable for applications in general biomechanics analysis. Until recently it was not possible to incorporate muscle activity in the MADYMO human body model. The reason for this is that the package was developed to describe occupant behavior in car crash simulations. In an accident the impact forces on the human body are so large that muscle forces are negligible. Thus, there was no reason to include muscle modelling in MADYMO. But in other applications, like human movement in sports or simulations of working postures, it is obligatory to be able to include muscle activity.

The modelling of muscle activity can be described at different levels. A distinction can be made between musculo-skeletal models, in which the muscle is seen as a black box, and muscle models, in which the internal behavior of the muscle is
described. In the first case, the muscle is merely a force or torque generator. In the second case, the relation between the internal muscle behavior and the force or torque, that is caused by that behavior, is specified. Both ways of modelling muscle activity can be further distinguished.

In musculo-skeletal modelling, the most global way to describe muscle activity is by considering only the resultant torque around a joint caused by all muscular, ligament and friction forces acting on the joint. How all of the forces interact is not considered. At a more detailed level, the force caused by a group of lumped muscles or by an individual muscle is considered. The single muscle or group of muscles is viewed as a force generator.

In muscle modelling, a distinction can be made between macroscopic and microscopic muscle models. Most macroscopic models are Hill-type muscle models. In these models muscle length and contraction velocity and the concentration of calcium ions \([Ca^{2+}]\) in the muscle are related to the force generated by the muscle. The most refined level at which muscle activity is considered is that of the microscopic muscle models. In these models the structure of the muscle, i.e. fiber length, pennation angle, physiological cross-sectional area, number of motor units, etc., is considered.

Both macro- and microscopic muscle models describe muscular behavior as a result of electric stimulations from the central nervous system. In case of the macroscopic model the central nervous system determines the active state of the muscle. This is a measure of the muscle activity, which is related to the concentration of calcium ions. In the microscopic model an electrical pulse is send to a motor unit, which thereupon contracts. A motor unit consists of a group of homogeneous muscle fibers and an associated motor neuron that passes the pulse to the fibers.

Part III of this report contains a literature study on muscle modelling, in which all models described above were analyzed. From this study and information from persons, who are experts in the field of muscle modelling, it was concluded that in this phase it was not appropriate to incorporate macro- or microscopic muscle models in MADYMO. It was decided to consider muscle modelling at the level of resulting torque around joints and muscle force caused by groups of lumped muscles or by individual muscles.

User-defined subroutines were developed, which allow the user to define time-dependent (muscle) forces and torques in MADYMO (both 2D and 3D). When the user wants to prescribe a certain force acting on an element (body segment) he has to specify the point of application of the force, the coordinate system with respect to which the force components are given and the time-histories of the force components. The coordinate system can be that of inertial space, the local coordinate system of the element on which the force is applied or the local coordinate system of any other element in the model. In Chapter 3 examples of these three
possibilities will be given. When prescribing a torque, the user has the option to let the torque act around a joint or to let it act on an element. In the three-dimensional case the user has to specify in addition a coordinate system with respect to which the three torque-time histories are given.

2.3 Anthropometry

As was mentioned before, MADYMO is developed to describe the behavior of occupants in car crash simulations. The occupants in full-scale car crash tests are dummies of various sizes, which are used as physical models of the human body. It is desired to use MADYMO for general biomechanics analysis, and therefore it should be possible to efficiently model an arbitrary human body.

![Local coordinate systems in 2D and 3D human body models that are obtained using GEBMAD](image)

Figure 2.3.1: Local coordinate systems in 2D and 3D human body models that are obtained using GEBMAD
The anthropometry generator GEBOO (GEnerator of BOdy Data, see Baughman, 1983) produces body description data sets corresponding to specific types and sizes of human bodies. GEBOO uses a set of body measurements, which have to be supplied by the user, to segments GEBOO calculates its mass and its principal moments of produce a model of the human body consisting of a system of fifteen joint-connected rigid bodies. For each of these body inertia. The principal axes are parallel to the local coordinate axes. (Note, that for the head this is not true). The local coordinate systems for the 2D and 3D human body models are shown in Figure 2.3.1. Further, GEBOO generates a contact ellipsoid for each body segment that gives shape to the segment and that provides an interaction surface between the segment and its environment. Finally, GEBOO calculates the locations of the joints that connect the segments.

Vosbeek (1989) has developed a computer program (written in FORTRAN '77) called GEBMAO, which converts the body description data calculated by the program GEBOO into a database for MADYMO. The database contains configuration, geometry, inertia and ellipsoids (ellipses in the 2D case) of a tree structure of rigid bodies that represent the human body. In section 3.3 a two-dimensional example is given in which a human body model is created using GEBMAO.

GEBMAO produces only part of a MADYMO input file for a simulation in which the human body plays a part. To complete the MADYMO input data set additional information of the human body is needed, such as joint characteristics, segment surface compliance (stiffness) and initial conditions (initial orientations and angular velocities of the rigid bodies).

GEBMAO can provide both 2D and 3D databases for MADYMO. In the three-dimensional case GEBMAO produces a tree structure that consists of fifteen elements. In the two-dimensional case the human body is projected in its sagittal plane and it is viewed from the right side. The elements representing the arms, leg and feet are lumped together, so five elements represent the upper and lower arms, upper and lower legs and the feet. This results in a tree structure consisting of ten elements. The 2D and 3D tree structures are shown in figure 2.3.1 and their element names are listed in tables 2.3.1 and 2.3.2. A user's guide to the program GEBMAO can be found in Appendix B of Vosbeek (1989).

It is planned that GEBMAO will be part of a general preprocessor for MADYMO that will be developed in the near future.
Table 2.3.1: The elements in the 2D human body model

<table>
<thead>
<tr>
<th>Element number</th>
<th>Element name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>lower torso</td>
</tr>
<tr>
<td>2</td>
<td>spine</td>
</tr>
<tr>
<td>3</td>
<td>upper torso</td>
</tr>
<tr>
<td>4</td>
<td>neck</td>
</tr>
<tr>
<td>5</td>
<td>head</td>
</tr>
<tr>
<td>6</td>
<td>upper arms</td>
</tr>
<tr>
<td>7</td>
<td>lower arms</td>
</tr>
<tr>
<td>8</td>
<td>upper legs</td>
</tr>
<tr>
<td>9</td>
<td>lower legs</td>
</tr>
<tr>
<td>10</td>
<td>feet</td>
</tr>
</tbody>
</table>

Table 2.3.2: The elements in the 3D human body model

<table>
<thead>
<tr>
<th>Element number</th>
<th>Element name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>lower torso</td>
</tr>
<tr>
<td>2</td>
<td>spine</td>
</tr>
<tr>
<td>3</td>
<td>upper torso</td>
</tr>
<tr>
<td>4</td>
<td>neck</td>
</tr>
<tr>
<td>5</td>
<td>head</td>
</tr>
<tr>
<td>6</td>
<td>upper arm left</td>
</tr>
<tr>
<td>7</td>
<td>lower arm left</td>
</tr>
<tr>
<td>8</td>
<td>upper arm right</td>
</tr>
<tr>
<td>9</td>
<td>lower arm right</td>
</tr>
<tr>
<td>10</td>
<td>upper leg left</td>
</tr>
<tr>
<td>11</td>
<td>lower leg left</td>
</tr>
<tr>
<td>12</td>
<td>foot left</td>
</tr>
<tr>
<td>13</td>
<td>upper leg right</td>
</tr>
<tr>
<td>14</td>
<td>lower leg right</td>
</tr>
<tr>
<td>15</td>
<td>foot right</td>
</tr>
</tbody>
</table>

2.4 Massless deformable bodies in MADYMO

As was mentioned previously, in MADYMO the human body is represented by a number of joint-connected rigid bodies. In order to describe the motion of some parts of the human body (e.g. the neck and the spine) more accurately, these parts could be modelled by a large number of rigid bodies. However, this leads to a large system of equations of motion and it takes a considerable amount of computer time to solve such a system. An alternative (and more powerful) way of modelling these flexible parts is to represent them by deformable bodies. Using deformable bodies the extension of human body parts can be taken into account as well.

Vosbeek (1989) has written user-defined subroutines that allow the user to introduce massless deformable bodies in MADYMO (only 3D). It is assumed that each deformable body connects two rigid bodies. The deformable body is considered massless for reasons
of simplicity. In the massless case the only contribution of a deformable body to the equations of motion is a contribution to the external forces and torques that act on the rigid bodies. A massless body does not introduce any inertial forces or torques into the equations of motion. The mass of the part of the human body that is represented by a massless deformable body may be taken into account by adding it to the masses of the two rigid bodies to which it is attached.

Vosbeek introduces six generalized coordinates to describe the relative position and orientation of the two rigid bodies connected by a deformable body. The forces and torques that are exerted on the two rigid bodies by the intermediate body and vice versa (Newton's third law), are described as functions of the generalized coordinates and their first time derivatives. These functions form the constitutive equations for the deformable body. They depend, for instance, on the shape and the material of the body. For each deformable body in the model the user has to supply two user-defined subroutines, which are based on the constitutive equations for that specific deformable body. These subroutines read material properties from the MADYMO input file and calculate loads due to relative displacements, orientations and velocities. They are called by Vosbeek's routines. The details on the incorporation of massless deformable bodies can be found in Vosbeek (1989).

An example is given by Vosbeek for the simple case of the motion of two identical spheres, which are connected by a linear elastic rod of circular cross-section. In the constitutive equations for the rod only relative displacement and orientation of the spheres appear, not their first time derivatives. The user-defined input and the source codes of all user-defined subroutines can be found in Appendices D and E of Vosbeek's report.

2.2 Pre- and postprocessing of MADYMO

2.2.1 Preprocessing

In MADYMO the human body can be represented by a tree structure which will be called dummy here. At the moment all MADYMO input has to be supplied by the user in a data input file. For dummies consisting of a lot of elements it is a tedious job to make sure that its elements have the right initial position and orientation. A preprocessor, allowing interactive manipulations of a tree structure, would simplify the positioning of a dummy in its environment. Another problem in manually creating the correct input data file is that it is difficult to find an equilibrium position for the dummy at the start of the simulation. In a car crash simulation the dummy needs to be positioned in the car seat in such manner that the dummy’s weight is compensated by the contact force exerted on the dummy by the car seat. Until now, the correct initial position is found by a process of trial and error. A preprocessor can speed up this process by allowing MADYMO to interactively calculate a few time steps of the simulation to see whether the system is in
equilibrium or not.

Vermazeren (1990) is developing a prototype of a preprocessor that will help the MADYMO user with the positioning of a dummy in its environment. The program will be called PREPOS and it is set up for MADYMO 3D input files. PREPOS will be able to load an input file, which contains exactly one dummy consisting of an arbitrary number of elements. The environment of the dummy is restricted to be constructed merely of planes. The ellipsoids and planes that represent the dummy and its environment are visualized on screen by means of wire-frame or solid rendering of a still image. In the wire-frame case the user can specify whether the ellipsoids should be visualized by contour lines or by their semi-axes. Scaling, rotation and translation will enable the user to look at the dummy from various viewpoints.

The actual positioning of the dummy, using a mouse or a lightpen, can be done in a local or global way. Individual elements or chains of elements can be rotated around a joint, that connects them to another element, by explicitly prescribing the joint angle or by a step by step change of the joint angle. Joints cannot be translated by dragging, only by rotation. Further, it will be possible to rotate and translate the dummy as a whole. After each change of the model configuration the changed input file can be saved.

Finally, the PREPOS user can interactively start a one-step MADYMO simulation of the current input file. From the MADYMO output files it can be seen whether the dummy is in an equilibrium position or not. Once the user decides that the actual input data set is satisfactorily, a complete MADYMO simulation can be carried out.

PREPOS will be implemented on a SUN workstation. A user's manual will be included in the final report of Vermazeren (1990). Since the prototype PREPOS is still in development, it was not possible to use it in the simulations described in Chapter 3.

2.5.2 Postprocessing

At the TNO Road-Vehicles Research Institute the postprocessors TNOPOST4 and MADPOST are used for visualization of MADYMO 2D and 3D output. An Evans & Sutherland PS390 graphics system is used for TNOPOST4. This postprocessor is capable of animation of wire-frame models, view point selection, zooming and time-history plotting. Shading operations (solid rendering) can be performed on static images. MADPOST produces wire-frame animation of the simulations kinematics as well as variable-time and variable-variable XY-plots.

Vermazeren (1989) has developed a prototype of a postprocessor for MADYMO 3D output files, called TNOPOST. The program can handle both KINEMA and KINTWO output files. Only the ellipsoids in the files are displayed. As a result planes or safety belt segments will not be seen on the screen. TNOPOST is capable of
both wire-frame and shading animations. The latter feature is not offered by the existing TNO postprocessors. The solid rendering animation is not performed real-time. The user can specify how accurately the ellipsoids should be rendered by means of two variables. One variable specifies the number of contour ellipses used for the representation of the ellipsoids, while the other variable defines the number of polygons by which each contour ellipse is represented. In case of a KINEMA file the user is allowed to change the degree of the ellipsoids. Finally, TNOPOST offers the possibility to change the viewpoint by scaling (zooming), rotation or translation.

The TNOPOST software was implemented on a SUN work station at the Eindhoven University of Technology. At TNO this computer is not available. The postprocessor prototype was tested on output files from car crash simulations. It was never used, however, to visualize any results from simulations that were done in the framework of the project. TNOPOST4 was used for this purpose.
CHAPTER 3
EXAMPLES OF MADYMO SIMULATIONS USING HUMAN BODY MODELLING

3.1 Introduction

Three MADYMO simulations in which human movement is modelled will be presented. These simulations make use of the individual contributions to the project that are described in the previous chapter. In all of the simulations muscle activity is incorporated in the model of the human body. In one case anthropometric input data were obtained using the interface GEBMAD. In the two other cases these data were acquired from literature. No simulations involving also massless deformable bodies were carried out. Further, pre- and postprocessing contributions to the project have not been used because they were not (yet) available.

The first simulation is one of a dart-throwing arm as modelled by Hoekstra (1987). The arm consists of two joint-connected rigid bodies representing upper and lower arm. The triceps force is modelled as well as the resultant torque around the shoulder joint. Next, the simulation of a subject doing a push-up is considered. A simple model, which can also be found in section 3.4 of part III of this report, is described together with an improved model in which more elements are used to represent the subject. Anthropometric data for the improved model were obtained using GEBMAD. In both models the motion of the subject is caused by a torque acting around the elbow joint. Finally, a simulation of the human leg during the swing phase of gait was done.

The MADYMO input file for each simulation is given in Appendix A. MADYMO 2D input is described in Chapter 4 of the MADYMO User's Manual 2D. The user-defined input used to prescribe muscle activity can be found in Appendix A of part III of this report.

3.2 A dart-throwing arm

3.2.1 The model

Hoekstra (1987) has developed a two-dimensional model of the human arm, which is depicted in Figure 3.2.1.1. The arm is modelled as a system of two rigid bodies representing the upper and lower arm. The wrist was assumed to be fixed, thus the hand holding the dart did not need to be modelled by a separate rigid body. The two segments are connected by a joint of the hinge type, which represents the elbow. The upper arm is attached to the trunk by means of another hinge type joint. This shoulder joint is assumed to be fixed in space.

Geometry (segment lengths and positions of joints and centers of gravity), inertia (masses and moments of inertia) and initial conditions of the system are known.
The system has two degrees of freedom. Two angles $\theta_1$ and $\theta_2$ are defined to describe the position of the arm. They denote the angle between upper arm and vertical axis, respectively, the angle between lower arm and vertical axis, as shown in Figure 3.2.1.1. Note that they have an opposite orientation. The elbow joint angle is equal to $\theta_1 + \theta_2$.

Muscle activity is modelled by a resulting torque $T_1$ (with magnitude $T_1$) around the shoulder and a triceps force $K_2$ (magnitude $K_2$). The triceps force acts on the tip of the lower arm, which lies at a distance $u_1$ from the elbow joint. The orientation of $T_1$ is in a counter clockwise direction, equal to the orientation of $\theta_1$. The triceps force is assumed to act parallel to the upper arm, in the direction of the shoulder. In two-dimensional space, the direction of the torque vector is determined. Only its magnitude is variable. Therefore the torque will be denoted by $T_1$ from now on.

The darts' position was taken to be at the tip of the lower arm. It was assumed that during a throw the dart moved along a straight line as long as it was held in the hand. The angle between the line and the horizontal axis ($\beta$) is known.

The model was used to simulate the motion of the human arm throwing a dart. The model gives rise to two equations of motion, in which the angles $\theta_1$ and $\theta_2$, the triceps force $K_2$ and the torque $T_1$ appear as unknown functions of time. The straight line renders a relation between the two angles. Hoekstra has written a computer program called DARTS, which is based on the two equations of motion and the relation that follows from the straight line condition. It calculates $\theta_1$, $\theta_2$, and $T_1$ as a function of time, given $K_2$ at every instant of the motion and given the initial conditions of the system. Numerical simulation results show good agreement with experimental kinematics data that were obtained from experiments.

The approach used is a mixture between direct and inverse dynamics. In direct dynamics the kinetics of a system is given...
and the system's kinematics is calculated. In inverse dynamics the kinetics of a system is computed from the kinematics of the system. In the model of a dart-throwing arm both kinetics \( K_2 \) and kinematics (the straight line condition) are given and both kinematics \( T_1 \) and kinetics \( \theta_1 \) and \( \theta_2 \) are produced.

### 3.2.2 The pulley model

Hoekstra has also considered an alternative model that differed in only one aspect from the model described in the previous section. In this so-called pulley model, depicted in Figure 3.2.2.1, the triceps force is supposed to act in a direction perpendicular to the lower arm instead of parallel to the upper arm. In the pulley model, the triceps muscle is thought to be guided over a pulley with its center at the elbow joint and with a radius equal to the distance \( u_1 \) between the elbow joint and the tip of the lower arm. Now, the torque around the elbow caused by the triceps force is greater than in the case of the original model. There, the torque is equal to \( K_2 u_1 \sin(\theta_1 + \theta_2) \), whereas in the pulley model the moment arm of force does not depend on the elbow angle. The moment arm is constant and equal to the pulley radius \( u_1 \), resulting in a torque \( K_2 u_1 \). The change in orientation of \( K_2 \) causes different equations of motion, thus requiring also an adaption of the original program DARTS used to simulate the motion of a dart-throwing arm.

![Figure 3.2.2.1: The pulley model](image)

In the next sections both models will be referred to as the original model and the pulley model, respectively.

### 3.2.3 The MADYMO model

In MADYMO the human arm is represented by a system of two elements that are connected by a hinge type joint. The shoulder is supposed to be fixed in the two-dimensional space. However, in MADYMO it is not possible to prescribe kinematical constraints. A third element is defined representing the trunk to which the arm is connected by means of a (shoulder) joint.
(The trunk is thought to be attached rigidly to the floor). This element is given a very large mass (of one billion kilograms), so that the reaction forces in the shoulder joint caused by the arm are too small to give the huge mass an acceleration. The center of mass of the heavy element is located at the shoulder joint.

MADYMO uses direct dynamics in its calculations. It produces kinematics output from kinetics input. Forces and torques have to be supplied. Positions, velocities and accelerations are calculated. To simulate the motion of a dart-throwing arm using MADYMO a different approach was used than Hoekstra (1987). Both force $K_2(t)$ and torque $T_1(t)$ are prescribed together with the initial conditions, as obtained from Hoekstra’s results. MADYMO calculates $\Theta_1(t)$ and $\Theta_2(t)$. It is expected that the dart (which is at the tip of the lower arm) moves along a straight line, which makes an angle $\beta$ with the horizontal axis.

In Figure 3.2.3.1 the local coordinate systems for the elements representing upper and lower arm are shown. Since the element numbers in the MADYMO configuration are 2 and 3, their axes are denoted by superscripts (2) and (3), respectively. Further, the inertial y and z axes are shown. The local coordinate system of element 1, representing the trunk, is rotated over an angle $\pi$ with respect to inertial space and is not shown in the figure. A simulation based on the original model ($K_z$ parallel to the upper arm), requires the prescription of the force $K_2$ acting in the direction of the negative $Z^{(2)}$ axis. Thus, the force is given with respect to the local coordinate system of an element different from the element on which the force is applied. On the other hand, the pulley model ($K_z$ perpendicular to the lower arm) requires the specification of a time-dependent $K_2$ in the direction of the negative $y^{(3)}$ axis. In this case the force is expressed in the local coordinate system of the element on which the force acts.

Figure 3.2.3.1: Inertial axes and local coordinate systems for the upper and lower arm.
$T_1$ is the resulting torque around the shoulder. If $T_1$ is prescribed to act around the shoulder joint this results in a torque $T_1$ acting on the element representing the upper arm and an opposite torque $-T_1$ acting on the element that represents the fixed trunk. Since the user-defined subroutines allow the prescription of a torque acting on an element as well as on a joint, the torque $T_1$ can also be prescribed to act on the upper arm. That way, no torques will act on the element representing the trunk. Therefore, it will not rotate and its moment of inertia can be given an arbitrary positive value.

The MADYMO input data file for a simulation of a dart-throwing arm that uses the original model is given in Appendix A. The input file for a simulation using the pulley model is similar. The two simulations describe respectively the eighth and sixth throw in Hoekstra's report. Both throws last 87 milliseconds. The triceps force is taken to be constant throughout the motion. The values for $K_2$ in the simulations are 1180 N and 550 N, respectively. The value for the torque $T_1$ is given every millisecond, starting at $t=0$ and ending at $t=0.087s$. The timestep in the MADYMO simulations was 1 millisecond.

### 3.2.4 Results

The MADYMO simulations of both arm motions resulted in following output files: REPRINT, DEBUG, KINTWO, ANGLE and ANGVEL. Both REPRINT files showed a correct reproduction of the input for the user-defined subroutines. Inspection of the input values of $T_1(t)$ and $K_2(t)$ shows that they are stored correctly in the DEBUG files at every time point of the simulations.

![Figure 3.2.4.1](image)

**Figure 3.2.4.1**: Stickdiagrams of the motion of the throw using the original model in the simulation (a), and of the throw using the pulley model in the simulation (b)
The other three output files contain kinematics results. Starting at \( t=0 \), the position of the arm is stored in KINTWO every 8.7 ms, which is one tenth of the duration of the motion. The coordinates of shoulder, elbow and the two ends of the lower arm are given in four decimals. Figure 3.2.4.1 shows stick diagrams for the eleven time points of the motion for both throws. On the left the motion belonging to the simulation using the original model is depicted, while on the right the result of the simulation in which the pulley model was used can be seen. It seems in both cases that the dart indeed moves along a straight line, as was expected. Comparison of the values of the coordinates of elbow and dart from the KINTWO files with their values obtained by Hoekstra using the program DARTS shows, that they are exactly the same (up to four decimals). In other words, the MADYMO simulation results in the same motion of the dart-throwing arm.

The KINTWO files contain the position of the dart at different times. From these positions the angle \( \beta \) between the straight line and the horizontal axis can be calculated. For both simulations \( \beta \) was practically equal to the value that Hoekstra used as an input parameter in his program DARTS. The discrepancy was in the order of \( 10^{-4} \) rad and is due to round-off errors. From the coordinates of the shoulder joint in the KINTWO files it was seen that the shoulder does not move.

![Figure 3.2.4.2: Time-histories of the elbow angle in the simulation using the original model (a) and in the simulation using the pulley model (b)]](Image)

From the files ANGLE the time-histories of \( \Theta_1, \Theta_2 \), and \( \Theta_1+\Theta_2 \), as calculated by MADYMO, can be read. Moreover, from ANGVEL the computed angular velocities \( \omega_1(t) \) and \( \omega_2(t) \) can be obtained. The values of these variables agreed up to five significant digits with the values calculated by the program DARTS. Plots of the time-histories of the elbow angle, as calculated by MADYMO, can be seen in Figures 3.2.4.2 a and b for the case of the original and the pulley model. Note, that actually the graph of \( (\Theta_1+\Theta_2)(t)+\pi \) is shown. Using the current configuration of the
tree structure representing the arm, it is not possible to obtain $(\theta_1+\theta_2)(t)$ via the user-defined subroutine ANGOU2.

\[ \text{HUMAN ARM} \quad \text{UPPER ARM} \]

\[ \text{HUMAN ARM} \quad \text{LOWER ARM} \]

(a) \hspace{1cm} (b)

**Figure 3.2.4.3**: Time-histories of the angular velocities of upper arm (a) and lower arm (b) in the simulation using the original model.

\[ \text{HUMAN ARM} \quad \text{UPPER ARM} \]

\[ \text{HUMAN ARM} \quad \text{LOWER ARM} \]

(a) \hspace{1cm} (b)

**Figure 3.2.4.4**: Time-histories of the angular velocities of upper arm (a) and lower arm (b) in the simulation using the pulley model.

In Figure 3.2.4.3 the graphs of the angular velocities of both elements are shown for the simulation in which the original model was used. Figure 3.2.4.4 shows similar graphs for the case of the pulley model. Due to the orientation of MADYMO inertial space, which is opposite to the orientation of $\theta_2$, the angular velocity of the lower arm from ANGVEL is negative, whereas $\omega_2$ is positive.
3.3 A subject doing a push-up

3.3.1 A simple model

In section 3.4 of part III of this report a MADYMO 2D simulation of a subject doing a push-up is demonstrated. The subject is modelled as a system of five rigid bodies connected by four hinge type joints (see Figure 3.3.1.1). The five elements represent the feet (1), head-neck-trunk-legs (2), upper arms (3), lower arms (4) and hands (5). Wrist and ankle joints are considered fixed in space, while shoulder and elbow joints are free to move. The hands and feet rest on the floor and do not move. The other three elements form a chain with fixed ends. This chain has one degree of freedom. If, for instance, the elbow joint angle is known, the position of the chain is determined.

![Figure 3.3.1.1: Simple model of a subject doing a push-up](image)

MADYMO uses direct dynamics in its calculations. It has been shown (see Part III of this report) that the user-defined subroutines that allow the prescription of time-dependent (muscle) forces and torques can be used to let MADYMO follow an inverse dynamics approach. By introducing a torque generator with linear feedback (conform Van den Bogert, 1988) a desired joint angle and angular velocity can be used to create a control torque that causes the actual joint angle and angular velocity to approach their desired values. Thus, a kinematical variable is prescribed in order to achieve a kinetic variable, namely the torque that is necessary to cause a certain desired joint angle and angular velocity.

The torque generator suggested by Van den Bogert yields

\[
M(\phi, \omega, t) = -A(\phi(t) - \phi_d(t)) - B(\omega(t) - \omega_d(t)),
\]

where \(M\) is the total (muscle) torque, \(\phi\) is the actual joint angle, \(\phi_d\) is the desired joint angle that acts as a control function, \(\omega\) and \(\omega_d\) are the first time derivatives of \(\phi\) and \(\phi_d\), and \(A\) and \(B\) are the feedback parameters. At time \(t\) the first (elastic) term produces a torque that drives \(\phi\) towards \(\phi_d\), while the second (damping) term decreases (increases) the torque, if the angle \(\phi\) is moving towards (away from) \(\phi_d\). The torque generator can also be written as

\[
M(\phi, \omega, t) = -A\phi(t) + A\phi_d(t) - B\omega(t) + B\omega_d(t) = -A\phi(t) - B\omega(t) + C(t)
\]
PUSH-UP

Time 0.00  Time 100.00

Time 200.00  Time 300.00

Time 400.00  Time 500.00

Time 600.00  Time 700.00
Figure 3.3.1.2: The position of the subject doing a push-up at different time points

Figure 3.3.1.3: Time-histories of the desired and actual elbow angles

The first two terms on the right can be modelled in standard MADYMO (elastic and damping torques in a joint). The third term can be modelled using the user-defined subroutines.
In the MADYMO simulation of the push-up the desired elbow joint angle $\phi_d(t)$ was chosen to be a cubic polynomial function that satisfied certain initial and end conditions. It was assumed that the subject initially was at rest with an elbow angle of 30 degrees. At the end of the simulation ($t=1$s) the subject again was supposed to be at rest, this time with arms completely stretched, thus requiring a final $\phi_d$ value of 180 degrees. The inertia and geometry of the elements were roughly estimated. The wrist and the ankle were kept fixed in space by giving the feet and the hands a mass of $10^3$ kg and by taking their center of gravity at the joint. That way no torques will act on the feet and the hands. After a process of trial and error it was found that $A = 900 \text{ Nmrad}^{-1}$ and $B = 35 \text{ Nmsrad}^{-1}$ gave the best results, when a timestep of 1 millisecond was used in the simulation.

Figure 3.3.1.2 shows the position of the subject at different time points. In Figure 3.3.1.3 time-histories of the desired and actual angles in the elbow joint are depicted.

3.3.2 An improved model

The above model can be used in an example of the integration of GEBMAD, the interface between GEBOD and MADYMO (see Vosbeek, 1989), and the user-defined subroutines for muscle activity. In the two-dimensional case GEBMAD creates part of a MADYMO input file in which geometry, inertia and size of ten elements are specified. This information can be used in a simulation of a subject doing a push-up. This way, geometrical and inertial values of the system will be more realistic. The improved model is shown in Figure 3.3.2.1. There are more elements than the five considered previously and therefore the number of joints and thus the number of degrees of freedom is also greater. To do the same type of simulation requires the "locking" of the "new" joints. It is not possible to actually lock joints in MADYMO. Therefore an elastic torque and a damping torque, that will not allow much flexion or extension in these joints, was prescribed. In fact, all that was done here, was to apply the linear feedback torque generator from above with a desired joint angle and angular velocity equal to zero. The term $C(t)$ in equation (3.3.1.2) vanishes, when $\phi = \phi_d = 0$, leaving the elastic term $-A\phi$ and the damping term $-B\omega$.

![Figure 3.3.2.1: The improved model](image-url)
Element 2 in the previous model represents head, neck, torso and legs. In the new model these segments are represented by the following elements: head, neck, upper torso, spine, lower torso, upper legs and lower legs. Further, feet, upper arms and lower arms are represented by an element. This means that the hands are not modelled separately, but that they are part of the lower arms. This has no big consequences, because the hands played a passive role in the original model. Since in the new model the hands are part of the lower arm, it was implicitly assumed that the subject is doing a push-up on his fingertips. The lower arm is connected to the ground by means of a hinge type joint, which is located at the fingertips.

3.3.3 The MADYMO model

A MADYMO input file based on the input file of the simple model was constructed. The latter input file can be found in Appendix C of part III of this report. In this input data set the data obtained from GEBMAD were incorporated.

The function of the fixed wrist is now fulfilled by an imaginary "joint" between the lower arm and an extra element representing the floor. The latter element is taken so heavy that the joint will stay fixed. This extra element causes a slight change in the configuration of the tree structure that is produced by the anthropometry interface. Again, the mass of the feet is taken to be large, so that the ankle joint will stay in its place.

The configuration of the tree structure representing the human body obtained using GEBMAD is such that the subject is standing with his feet pointing to the right. The subject was maneuvered into the initial push-up position by specifying the orientations of the elements in the MADYMO input file.

For reasons of convenience the directions of the local coordinate systems of the elements representing lower and upper arms were changed. This caused changes for these elements in the data blocks under the keywords GEOMETRY and ELLIPSES.

In the elbow joint the same torque generator as in the simulation of section 3.3.1 (A = 900 Nmrad\(^{-1}\) and B = 35 Nmsrad\(^{-1}\)) was used. In the joints that should stay locked an elastic and a damping torque with coefficients A = 1000 Nmrad\(^{-1}\) and B = 1 Nmsrad\(^{-1}\) was prescribed.

The MADYMO simulation uses a timestep of 1 millisecond. Its input file is shown in Appendix A. Note the difference between the data block gathered from GEBMAD and the one actually used in the simulation.

3.3.4 Results

In Figures 3.3.4.1 and 3.3.4.2 the results of the MADYMO simulation are shown.
PUSH-UP

Time 300.00

Time 350.00

Time 400.00

Time 450.00

Time 500.00

Time 550.00
PUSH-UP

Time 600.00

Time 650.00

Time 700.00

Time 750.00

Time 800.00

Time 850.00
The plots in Figure 3.3.4.1 were obtained from the data stored in the KINTWO file, whereas the graph of the elbow angle in the second figure was retrieved from the output file ANGLE. Let us first take a look at the diagrams of the subject in Figure 3.3.4.1, which are shown every 50 milliseconds. In the first phase of the motion, until $t = 350$ ms, sagging of the trunk and legs occurs. This is due to gravity. Then, the torque generators in the joints that were desired to be locked cause the subject to move towards a position in which the actual joint angles approach their desired value of zero radians. This situation is reached at $t = 550$ ms. After that (at $t = 550, 600$ and $650$ ms), the subject initially is bent in a convex shape with his head slightly tilted forward and some flexion in the hip. In the following situations there is flexion in the knee, while the other elements are aligned in the desired way. Apparently, vibration occurs, resulting in a situation of buckle. This phenomenon is probably caused by numerical instability, and should be investigated further.
Figure 3.3.4.2 : Time-histories of the desired and actual elbow angles

Figure 3.3.4.2 shows that the behavior of the elbow angle in this simulation is similar to that in the simulation of the simple model, as can be seen in Figure 3.3.1.3. So, even though the movement of the subject is not the desired one (all "new" joints locked), the torque generator in the elbow joint still performs well.

3.4 A swinging leg

3.4.1 The model

The last example of a MADYMO simulation using modelling of the human body, that is presented in this chapter, is the two-dimensional simulation of a swinging leg. The leg is modelled as a system of three rigid bodies, which represent thigh, shank and foot (see Figure 3.4.1.1). The three rigid bodies are connected by hinge type joints representing knee and ankle. Moreover, the thigh is connected to the rest of the body by a hip joint. All inertial properties (mass and moment of inertia), geometrical properties (positions of joints and centers of gravity) and initial conditions of the system are known. All the necessary input data for the simulation were obtained from Matthijsse (1989), who got these data from literature.

During the swing phase, which lasts 385 milliseconds, the velocity of the hip \( v_h(t) = (v_{hy}(t), v_{hz}(t)) \) is prescribed by means of analytical periodic functions in horizontal and vertical direction:

\[
\begin{align*}
v_{hy}(t) &= a_1 + a_2 \cos(a_3 t + a_4) \\
v_{hz}(t) &= a_5 + a_6 \cos(a_7 t + a_8)
\end{align*}
\] (3.4.1.1)
In the simulation the following values for the parameters $a_i$ ($i = 1, \ldots, 8$) were used:

$$
\begin{align*}
    a_1 &= 1.4416667, & a_2 &= 0.1452987, & a_3 &= 11.6238928, \\
    a_4 &= -0.1, & a_5 &= 0, & a_6 &= 0.4588379, \\
    a_7 &= 15.2945958, & a_8 &= 5.1123890.
\end{align*}
$$

Horizontal movement is governed by a period of $2\pi/a_3$ and vertical motion by a period of $2\pi/a_7$ (the periods are approximately 0.54 s and 0.41 s, respectively). The horizontal velocity of the hip lies between 1.30 and 1.59 m/s (approximate values of $a_1 - a_2$ and $a_1 + a_2$), whereas the vertical component $v_{hz}$ has values between -0.46 and +0.46 m/s.

Further, time-dependent torques around hip, knee and ankle are prescribed during the motion. The torque-time histories are piecewise linear functions. For hip and knee torque values are specified at seven time points, while for the ankle ten time-torque coordinate pairs are given.

From the initial conditions and the hip velocity the position of the hip is known at all time points. Thus, the system representing the leg has three degrees of freedom.

3.4.2 The MADYMO model

As was mentioned previously, in MADYMO it is not possible to prescribe kinematical constraints. Thus, the hip velocity, which is necessary in the simulation of the swinging leg, can not be prescribed. Again, the method of defining an extra element with a large mass is used. The center of gravity of this element defined to be located at the hip joint. The heavy element has to move with a certain desired velocity $v_h$. The element that represents the thigh is connected to the heavy element by means of the hip joint.
Since the hip velocity $v_h$ is known analytically, the desired hip acceleration $a_h(t) = (a_{h_y}(t), a_{h_z}(t))$ can also be calculated. Differentiating $v_h$ yields:

$$a_{h_y}(t) = -a_2a_3\sin(a_3t + a_4)$$
$$a_{h_z}(t) = -a_6a_7\sin(a_7t + a_8)$$

(3.4.2.1)

The values of the parameters $a_i$ can be given in section 3.4.1. The periods of the horizontal and vertical oscillations are the same as in the case of the hip velocity, namely about 0.54 s and 0.41 s. The amplitude of the oscillation of the horizontal acceleration is equal to $a_2a_3$, and that of $a_{h_z}$ equals $a_6a_7$ (the values of the amplitudes are approximately 1.69 and 7.02, respectively).

The hip velocity $v_h$ is prescribed as follows. In MADYMO it is possible to define an acceleration field model. This force-interaction model calculates the forces at the centers of gravity of elements in a homogeneous acceleration field $a$. These forces cause the elements to move with an acceleration $a$ if no other forces act on the elements. Thus, an acceleration field $a_h$ can be prescribed to the heavy element positioned at the hip. Given the right initial hip velocity $v_h(0)$, the hip will move with the desired velocity $v_h$, thanks to the forces caused by the acceleration field. The components of the prescribed acceleration $a_h$ are defined as a function of time by means of 66 coordinate pairs. The motion lasts 385 milliseconds.

The MADYMO input file for a simulation of the swinging leg is shown in Appendix A. Gravity is modelled by an acceleration field $g$ of magnitude $10 \, \text{m/s}^2$ acting downward (this field acts only on the three elements that represent the leg, not on the heavy element positioned at the hip). The time step is taken to be 1 millisecond. Simulations in which a stepsize of $10^{-4}$ s or a variable stepsize (starting with an initial time step of one microsecond) were used rendered exactly the same results (no significant differences were found in positions, velocities or accelerations of the four elements). Thus, a time step of 1 millisecond is sufficiently small.

3.4.3 Results

The time-histories of the components of the hip velocity in the simulation are shown in Figure 3.4.3.1. They agree with the functions $v_{h_y}(t)$ and $v_{h_z}(t)$ of equations (3.4.1.1). From LINVEL it was seen that the horizontal velocity component agreed with its desired value $v_{h_y}$ up to six significant digits. The vertical components agree in four to six significant digits. This means that the error in $v_{h_z}$ is less than 1 mm/s at every instant of the simulation.

Figure 3.4.3.2 shows stick diagrams of the motion of the leg during the swing phase. At 21 time points the position of the leg is depicted. Comparison with results from Matthijsse (1989)
yields, that there is a deviation in the position and orientation of the foot. Matthijsse has used a multi-bond graph method for her simulation. After several discussions about the possible reasons for the different results, it was concluded that it is most likely that certain parameters values used in her simulation differed from the ones used in the MADYMO simulation. In practice it is very difficult to exactly reproduce a simulation that was carried out by someone else.

Figure 3.4.3.1: Time-histories of the y-component (a) and the z-component (b) of the hip velocity

Figure 3.4.3.2: Stickdiagrams of the motion of the swinging leg
The foot is sensitive to small input data variations. It was seen that small changes in the prescribed ankle torque induced changes in the foot motion that were visible in the stick diagrams. The foot, which is the lightest element in the model, is at the end of the chain of three links representing the leg. Thus, any disturbances in the position and orientation of the thigh and the shank highly affect the kinematics of the foot.

Figure 3.4.1.1: Time-histories of the angle between the upper leg and the vertical axis (a), the knee joint angle (b) and the angle between the foot and the vertical axis (c).

In Figure 3.4.3.3 the time-histories of the angle between the upper leg and the vertical axis (θ₁ in Figure 3.4.1.1), the knee joint angle and the angle θ₃ between the foot and the vertical axis are depicted. The graphs are similar to those found in other studies on the swing phase in human gait. The graphs for the hip and knee angle are almost equal to the ones found by Matthijsse, while the graph that shows the orientation of the foot as a function of time differ (especially in the initial and end phase of the motion).
CHAPTER 4
DISCUSSION, CONCLUSION AND RECOMMENDATIONS

4.1 Introduction

The aim of the research that was done in the project was twofold. First, it was desired to obtain a better model of the human body by incorporating muscle activity, anthropometry and massless deformable bodies in the existing MADYMO model. Second, MADYMO pre- and postprocessing facilities had to be improved. In Chapter 2 these individual contributions to the project were presented. In the next section, each of them will be discussed separately and some suggestions for future research will be given. In section 4.3 the MADYMO simulations of the previous chapter will be discussed. Finally, a general conclusion and recommendations are given.

4.2 Separate project contributions

As far as muscle modelling is concerned, user-defined subroutines were developed that allow the user to prescribe muscle forces and torques in a MADYMO simulation. These user-defined subroutines were validated by means of some simple examples (see part III of this report). The simulations in the previous chapter, which are more elaborate, yield as a further indication that those subroutines perform well. Forces were prescribed where the force components were expressed relative to the local coordinate system of the element on which the force was applied (pulley model of a dart-throwing arm) or relative to the local coordinate system of an element different from the element on which the force was applied (original model of a dart-throwing arm). In all simulations the user-defined subroutines performed well. All simulations carried out up to now were two-dimensional. Three-dimensional simulations in which muscle activity is involved were not done. These will be necessary as a further validation of the extension of the MADYMO 3D package.

So far, muscle activity is only considered at the level of torques and forces caused by individual muscles or groups of muscles. At the moment the user has to supply muscle force and resulting torque in the model of the human body. In the future, Hill-type muscle models can be included in the MADYMO human body model. MADYMO has to be extended with user-defined subroutines that calculate the force generated by a muscle given the active state of that muscle. The routines must be based on force-length and force-velocity characteristics of the muscle. Length and contraction velocity of the muscle can be obtained from the relative positions and angular velocities of the segments that affect the path of the muscle. In the human body many types of muscles exist. Each individual muscle has different properties. Attention should be paid to muscle parameter estimation and to using the appropriate muscle characteristics in a macroscopic model.
Microscopic muscle models go into even more detail. They allow for the input of electrical pulses provided by the central nervous system. The micro model (possibly integrated in a macro model) then renders the force that is developed by the muscle due to muscular contraction caused by the neural pulses. At the moment, mathematical modelling of the central nervous system is a hot item in the field of neuroscience. Should reliable models of the central nervous system ever be established, then (theoretically) these models could be connected to muscle models, thus allowing for muscular control. It is expected that these integrated models will not be realized in the near future. In many studies of human motion such complicated models are not necessary, however.

GEBMAD, the interface between the anthropometry generator GEBOD and MADYMO, produces body description data sets corresponding to human bodies of different size and type. These data sets contain information on the configuration, the geometry and the inertial properties of the tree structure that represents the human body in MADYMO. GEBMAD output was used in the two-dimensional simulation of a subject doing a push-up. The interface can also render three-dimensional data sets, which can be used in MADYMO 3D simulations. (In fact, MADYMO customers in the automobile industry already use GEBMAD in the modelling of car occupants in crash victim simulations).

In the future, an anthropometry generator that also produces human joint characteristics is desirable. In MADYMO joint models an external load (e.g. joint torque) is applied to a joint. This external load is a function of the relative orientation and the relative angular velocity of the two elements connected by the joint. The MADYMO user can define elastic, damping and friction torques.

Further, it would be useful when anthropometric muscle parameters, like mass and origin and insertion, for certain muscles in the human body could be obtained by means of an anthropometry generator. These muscle data can then be used in any future macroscopic muscle models in MADYMO. Brand et al. (1982) have measured origin and insertion of all 47 muscles in the lower limb that cross the hip, knee and ankle joint for three cadavers (six limbs). Experimental data like these could be used in an anthropometry generator that calculates origin and insertion of muscles for an arbitrary human body using regression analysis.

Vosbeek (1989) has developed user-defined subroutines that allow the user to introduce massless deformable bodies in MADYMO 3D. These deformable bodies allow a more precise modelling of, for instance, the human neck and spine. The deformable body exerts forces and torques on the (two) rigid bodies it connects. These loads are functions of the relative orientation and position and the relative linear and angular velocity of the rigid bodies. The functions form the constitutive equations of the deformable body. The user-defined subroutines were tested only for the simple case in which the deformable body consisted of a linear elastic
material. In the future, the routines must be validated using constitutive equations in which also relative velocities occur. This is, for example, the case if the deformable body is made of a linear visco-elastic material, such as a Kelvin-Voight material. Later, non-linear materials can be considered.

Once the user-defined subroutines are tested rigorously, they can be used to actually model certain segments of the human body by a massless deformable body.

Next, an example of the possible use of massless deformable bodies in human body modelling is given.

In some biomechanical studies of spine and neck problems (especially in working postures) it is desired to have a more detailed model of the human body than the fifteen segments model currently used in most MADYMO 3D simulations. In the latter model the neck is represented by one element and the trunk by three elements: upper torso, spine and lower torso. The model is not suitable for accurate studies of spine and neck behavior.

Using massless deformable bodies a detailed model of the human spine can be obtained by representing, for instance, each vertebra by a rigid body and the (flexible) intervertebral disks by massless deformable bodies. (Note, that each rigid body (vertebra) is represented by a tree structure that consists of one element). This spine model could be incorporated in the tree structure consisting of fifteen elements that represents the human body, thus allowing for a better description of the bending and rotation of the spine and the neck.

![Figure 4.2.1: Model of the human neck using massless deformable bodies](image)

The detailed modelling described above may be applied to the part of the neck only, leaving the other fourteen elements in the tree structure unchanged. Suppose the neck is represented by n deformable and n-1 rigid bodies and let the mass and of the neck be m. It should be investigated, whether it is allowed to add the mass and the moments of inertia of the parts of the neck that is represented by deformable bodies to those of the surrounding rigid bodies. If so, a mass \(m/n\) should be prescribed to the n-1
rigid bodies. Further, an extra mass \( m/2n \) should be prescribed to the elements representing head and the trunk. The moments of inertia of the massless bodies should be distributed in a similar way. In Figure 4.2.1 the situation is depicted for \( n=5 \).

In the future, it is desirable that deformable bodies with a certain mass and certain moments of inertia can be defined in MADYMO. Then, the problem of distributing masses and moments of inertia, which occurs when massless deformable bodies are modelled, will not occur.

Further, the use of finite element methods in modelling the deformable bodies in a MADYMO simulation should be investigated. This requires the coupling of a finite element package with MADYMO. At the moment a finite element package is incorporated in MADYMO to describe airbag deformation in two-dimensional car crash simulations.

Apart from obtaining more realistic human body modelling, it was the purpose of the project to improve MADYMO pre- and postprocessing facilities. A prototype of a postprocessor was developed at the Eindhoven University of Technology. This prototype will not be used to visualize future MADYMO simulations, but some of its parts are used in the development of a preprocessor. This development is still in progress. When it is finished, the preprocessor can be used to position a tree structure, that represents the human body, in its environment. This will simplify the time-consuming task of manually constructing a MADYMO input file. The tree structure can be rotated and translated as a whole and elements can rotate around the joints to which they are attached. Further, the user can interactively give a command to let MADYMO calculate a few time steps of the simulation, so it can be seen whether the system is in equilibrium or not.

In the future, this preprocessor can be made more userfriendly still. Dragging of joints, adapting also the environment, changing geometry or inertia of the elements or changing joint characteristics are only a few improvements that can be thought of. Maybe it will even be possible to let the preprocessor establish an equilibrium by using some kind of control mechanism. Finally, anthropometry generators (like GEBOD) have to be integrated in the MADYMO preprocessor.

4.3 MADYMO simulations

In the previous chapter three examples of MADYMO simulations using human body modelling were presented. In the case of the subject doing a push-up two individual project contributions were combined, namely muscle activity and anthropometry. This simulation is especially suitable for the use of a preprocessor, because it was a tedious job to move the subject from the 'GEBMAD position' into the initial push-up position. Also in the simulations of the arm and the leg it would have been helpful if the preprocessor would have been available.
The MADYMO simulation of the dart-throwing arm rendered exactly the same results as in the work of Hoekstra (1987).

In the simulation of the push-up there was a problem, because some joints in the tree structure representing the subject could not be locked. This caused the subject to vibrate, leading to a final situation in which there was buckle in the knee joint. A way to solve this problem was not found, due to a lack of time. In the future, the problem has to be further investigated.

In the model an analytical function was chosen for the desired elbow joint angle and angular velocity that control the torque generator. It would be nice, if experimental values for the angle and angular velocity of the shoulder or elbow joint could be obtained. This would allow for an experimental verification of the model. In literature no relevant data seemed to be available.

The simulation of the swinging leg showed results similar to results found in literature. However, they were not exactly equal to those of Matthijsse (1989). The reason for this is probably that in the MADYMO model different input data were used. Matthijsse has done a simulation of the dart-throwing arm as well using the multi-bond graph method. She obtained the same results as were found in the MADYMO simulation. This indicates that different results in the simulations of the swing leg motion were caused by different input data.

In the future, MADYMO simulations of human gait in which also the stance phase is modelled are desired. Human standing is an unstable process, which is controlled by the central nervous system. Muscles are continuously activated to keep the body from collapsing. The modelling of the stance phase requires the use of some control strategy, together with a detailed model of the leg and its musculature. Further, it must be investigated how ground-foot interaction should be modelled. In case of the double stance phase one should be careful not to create any closed loops in the system representing the human body.

4.4 General conclusion

MADYMO is a Crash Victim Simulation program. The purpose of the research was to find out, whether MADYMO could be applied in general biomechanics analysis as well. Tools were developed to improve human body modelling in MADYMO. The program GEBMAD can be used to provide anthropometric data for an human body of arbitrary shape and size. Further, muscle activity and massless deformable bodies can now be incorporated in the model of the human body. This report shows that MADYMO is quite suitable for simulations in general biomechanics analysis, when the features of human body modelling that were developed in this project are applied.

Another purpose of the research was to improve MADYMO pre- and postprocessing facilities. A postprocessor allowing the display of MADYMO 3D simulations has been developed. Further, a
A preprocessor that will facilitate the use of MADYMO is still being developed.

Each of the separate contributions to MADYMO human body modelling and pre- and postprocessing can be further improved. Suggestions for future research were given in section 4.2.

In the simulation of the push-up a feedback control strategy was used to generate a torque at the elbow joint. After a simulation the user wants to know the value of this control torque. At the moment the user has to compute the torque himself. It is equal to the sum of an unknown term \(-A\phi - B\omega\) and a user-supplied term \(A\phi_d + B\omega_d\). The first term depends on the actual joint motion as calculated by MADYMO and can be obtained from the output file DEBUG. In the future, MADYMO should be extended in such manner that the values of each control torque in a simulation can be stored in an output file, from which they can be read after the simulation.

At the moment it is not possible to prescribe the motion of a point of an element or the orientation of an element in a MADYMO simulation. In the simulations of Chapter 3 desired joint motion was prescribed in an indirect way. It is desirable that in the future it will be possible to prescribe kinematical constraints in MADYMO.
REFERENCES

Baughman, L.D. (1983)
Development of an interactive computer program to produce body description data
Report AFAMRL-TR--83-058, Air Force Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio 45433

Simulation of quadrupedal locomotion using a rigid body model
J. Biomechanics, 22, 33-41

A model of lower extremity muscular anatomy
J. of Biomechanical Engineering, 104, 304-310

De mechanica van het dartwerpen
Master’s thesis, Department of Mathematical Engineering, University of Groningen

Some aspects of the use of MADYMO in general biomechanics analysis. Part I: Organization
Final report, postgraduate program "Mathematics in Industry", Eindhoven University of Technology

Some aspects of the use of MADYMO in general biomechanics analysis. Part III: Muscle modelling
Final report, postgraduate program "Mathematics in Industry", Eindhoven University of Technology

Report TNO Road-Vehicles Research Institute

Report TNO Road-Vehicles Research Institute

Matthijsse, P.M. (1989)
Modellvorming en simulatie met behulp van multibondgrafen toegepast op het looppatroon van de mens
Report, Department of Electrical Engineering, University of Twente

TNOPOST: A postprocessor for MADYMO 3D
Report, Department of Computer Science, Eindhoven University of Technology

Vermazeren, R.C.J.G. (in preparation)
Title unknown (subject: a preprocessor for MADYMO 3D)
Master’s thesis, Department of Computer Science, Eindhoven University of Technology
Two contributions to the modelling of the human body in MADYMO
Master’s thesis, Department of Mathematical Engineering,
Eindhoven University of Technology

Wittenburg, J. (1977)
Dynamics of systems of rigid bodies
BG Teubner, Stuttgart
APPENDIX A

MADYMO INPUT FILES FOR THE SIMULATIONS IN CHAPTER 3

A1. A dart-throwing arm

ARM.DAT
THROW E8
SEPTEMBER 1989
0. 0.087 0.0087
0. 0.001
0. 0.5 0.01 0.1
INERTIAL SPACE
THROWSPACE
PLANES
-1 -0.05 0.05 0.0 0 0 0 0 SUPPORT
-1 -0.05 -0.05 0.0 0 0 0 0 SUPPORT
-999
END INERTIAL SPACE
SYSTEM1
HUMAN ARM
CONFIGURATION
3 2 1
-999
GEOMETRY
0. 0. 0. 0. SHOULDER
0. 0. 0. 0.14 UPPER ARM
0. 0.32 0. 0.19 LOWER ARM
-999
INERTIA
10000000000 1
2.5 0.0217
1.7 0.0298
-999
PLANES
2 0. 0. 0.32 0 0 0 UPPER ARM
3 0. -0.02 0. 0.42 0 0 0 LOWER ARM
-999
INITIAL CONDITIONS
0. 0. 0. 0.
ORIENTATIONS
1 -1 3.14159265
2 -1 4.05459265
3 -1 0.335
-999
ANGULAR VELOCITY
1 0.
2 0.5965
3 -0.87
-999
END SYSTEM1
FORCE MODELS
ACCELERATION FIELDS
0 0 0 1
1 1 0 0
FUNCTIONS
2
0. -9.80 0.1 -9.80
END

FORCE MODELS

OUTPUT
1 0 0.0087 1

ANGVEL
1 2
1 3

END

OUTPUT

USER INPUT
EXT FOR
1 3 0. -0.02 1 2 0 1 'TRICEPS FORCE'
1
2 0. -1180. 0.1 -1180.

EXTMOM
1 1 2 -1 'SHOULDER'
1
88 0.000 27.12787 0.001 27.23329 0.002 27.34802 0.003
27.47206 0.004 27.60436 0.005 27.74596 0.006 27.89583 0.007
28.05501 0.008 28.22245 0.009 28.39816 0.010 28.58318 0.011
28.77479 0.012 28.98008 0.013 29.19093 0.014 29.41006 0.015
29.63848 0.016 29.87621 0.017 30.12117 0.018 30.37647 0.019
30.64004 0.020 30.91188 0.021 31.19302 0.022 31.48346 0.023
31.78217 0.024 32.09121 0.025 32.40853 0.026 32.73618 0.027
33.07314 0.028 33.41939 0.029 33.77598 0.030 34.14291 0.031
34.52017 0.032 34.90778 0.033 35.30674 0.034 35.71605 0.035
36.13776 0.036 36.57084 0.037 37.01529 0.038 37.47317 0.039
37.94346 0.040 38.42822 0.041 38.92641 0.042 39.43908 0.043
39.96621 0.044 40.50885 0.045 41.06907 0.046 41.64478 0.047
42.23910 0.048 42.85202 0.049 43.48458 0.050 44.13679 0.051
44.81070 0.052 45.50838 0.053 46.22879 0.054 46.97402 0.055
47.74715 0.056 48.54716 0.057 49.37817 0.058 50.24200 0.059
51.13633 0.060 52.06760 0.061 53.03815 0.062 54.04901 0.063
55.10432 0.064 56.20717 0.065 57.36067 0.066 58.56895 0.067
59.83614 0.068 61.16845 0.069 62.57001 0.070 64.04703 0.071
65.60776 0.072 67.25842 0.073 69.00934 0.074 70.86982 0.075
72.85123 0.076 74.96701 0.077 77.23266 0.078 79.66473 0.079
82.28387 0.080 85.11387 0.081 88.18262 0.082 91.52219 0.083
95.17182 0.084 99.17909 0.085 103.59876 0.086 108.49906 0.087
113.96474

ANGLES
1 2 1 1
1 2 1 3

END

INPUT DATA
A2. A subject doing a push-up

A data block obtained using GEBMAD is shown, followed by the MADYMO input file of the simulation of a subject doing a push-up, in which the GEBMAD output is incorporated.

**GEBMAD output**

```
SYSTEM
DICK HOEKSTRA
CONFIGURATION
5 4 3 2 1
7 6 3 2 1
10 9 8 1
-999
GEOMETRY
0.000 0.000 0.000 -0.102 LOWER TORSO
0.000 0.000 0.000 0.040 SPINE
0.000 0.081 0.000 0.163 UPPER TORSO
0.000 0.364 0.000 0.023 NECK
0.000 0.045 0.000 0.147 HEAD
0.000 -0.288 0.000 -0.145 UPPER ARMS
0.000 -0.287 0.000 -0.213 LOWER ARMS
0.000 -0.107 0.000 -0.222 UPPER LEGS
0.000 -0.473 0.000 -0.181 LOWER LEGS
0.000 -0.388 0.070 -0.074 FEET
-999
INERTIA
13.667 0.0883
4.701 0.0162
20.149 0.2375
1.406 0.0030
5.385 0.0342
4.221 0.0334
4.863 0.0694
17.141 0.3569
7.724 0.0970
1.730 0.0085
-999
ELLIPSES
1 0.116 0.099 0.000 -0.099 2 0 0 0 0 LOWER TORSO
2 0.106 0.101 0.000 0.040 2 0 0 0 0 SPINE
3 0.119 0.182 0.000 0.182 2 0 0 0 0 UPPER TORSO
4 0.060 0.083 0.000 0.023 2 0 0 0 0 NECK
5 0.101 0.147 0.000 0.147 2 0 0 0 0 HEAD
6 0.048 0.193 0.000 -0.145 2 0 0 0 0 UPPER ARMS
7 0.045 0.263 0.000 -0.213 2 0 0 0 0 LOWER ARMS
8 0.077 0.313 0.000 -0.222 2 0 0 0 0 UPPER LEGS
9 0.058 0.244 0.000 -0.181 2 0 0 0 0 LOWER LEGS
10 0.143 0.074 0.107 -0.074 2 0 0 0 0 FEET
-999
END SYSTEM
```

**MADYMO input file**

```
PUSH-UP
```
ONE OF
OCTOBER 1989
0.  1.0  0.01
0 0.001
0.  0.5  0.01  0.1
INERTIAL SPACE
FLOOR
PLANES
-1  -0.20  0.  1.5  0.  0  0  0  FLOOR
-999
END INERTIAL SPACE
SYSTEM
DICK HOEKSTRA
CONFIGURATION
5 4 3 2 1
8 7 6 3 2 1
11 10 9 1
-999
GEOMETRY
0.000  0.000  0.000 -0.102  LOWER TORSO
0.000  0.000  0.000  0.040   SPINE
0.000  0.081  0.000  0.163  UPPER TORSO
0.000  0.364  0.000  0.023   NECK
0.000  0.045  0.000  0.147   HEAD
0.000  0.288  0.000  0.145  UPPER ARMS
0.000  0.287  0.000  0.213  LOWER ARMS
0.000  0.476  0.000  0.000   FLOOR
0.000 -0.107  0.000 -0.222  UPPER LEGS
0.000 -0.473  0.000 -0.181  LOWER LEGS
0.000 -0.388  0.000 -0.000   FEET
-999
INERTIA
13.667  0.0883
4.701  0.0162
20.149  0.2375
1.406  0.0030
5.385  0.0342
4.221  0.0334
4.863  0.0694
10000000  100
17.141  0.3569
7.724  0.0970
10000000  100
-999
JOINTS
2 2 0 0 0 1 0 0. 0.
3 2 0 0 0 1 0 0. 0.
4 2 0 0 0 1 0 0. 0.
5 2 0 0 0 1 0 0. 0.
7 1 0 0 0 35 0 0. 0.
8 2 0 0 0 1 0 0. 0.
9 2 0 0 0 1 0 0. 0.
-999
FUNCTIONS
2
-3.14159265  0
0  2827.4334
2
-1. -1000.
  1.  1000.
-999

ELLIPSES
1  0.116  0.099  0.000 -0.099  2  0  0  0  LOWER TORSO
2  0.106  0.101  0.000  0.040  2  0  0  0  SPINE
3  0.119  0.182  0.000  0.182  2  0  0  0  UPPER TORSO
4  0.060  0.083  0.000  0.023  2  0  0  0  NECK
5  0.101  0.147  0.000  0.147  2  0  0  0  HEAD
6  0.048  0.193  0.000  0.145  2  0  0  0  UPPER ARMS
7  0.045  0.263  0.000  0.213  2  0  0  0  LOWER ARMS
8  0.001  0.001  0.000  0.000  2  0  0  0  FLOORHINGE
9  0.077  0.313  0.000 -0.222  2  0  0  0  UPPER LEGS
10 0.058  0.244  0.000 -0.181  2  0  0  0  LOWER LEGS
11 0.143  0.074  -0.107 -0.074  2  0  0  0  FEET
-999

INITIAL CONDITIONS
0.26666634  0.24906225  0.  0.

ORIENTATIONS
1  -1  1.571765
2   1  0.
3   1  0.
4   1  0.
5   1  0.
6   1  4.015338
7   1  -2.6179939
8   1  0.
9   1  0.
10  1  0.
11  -1  1.5707963
-999

ANGULAR VELOCITY
1  0.
2  0.
3  0.
4  0.
5  0.
-999

END SYSTEM
FORCE MODELS
ACCELERATION FIELDS
0  0  0  1
1  8  0  0
1  11  0  0
-999

FUNCTIONS
2
0.0  -9.8066  2.0  -9.8066
-999

END FORCE MODELS
OUTPUT
0  0  0.05  1

END OUTPUT
USER INPUT
EXTMOM

1
2 1 7 -1 'ELBOW'
1
56
0.0000 471.2389221 0.0200 484.8042908 0.0400 503.3585815
0.0600 526.6754761 0.0800 554.5288696 0.1000 586.6925049
0.1200 622.9401855 0.1400 663.0457764 0.1600 706.7830200
0.1800 753.9257202 0.2000 804.2478027 0.2200 857.5228882
0.2400 913.5249634 0.2600 972.0276489 0.2800 1032.8049316
0.3000 1095.6304932 0.3200 1160.2781982 0.3400 1226.5219727
0.3600 1294.1352539 0.3800 1362.8922119 0.4000 1432.5665283
0.4200 1502.931762 0.4400 1573.7620850 0.4600 1644.8312988
0.4800 1715.9129639 0.5000 1786.7810059 0.5200 1857.2092285
0.5400 1926.9713135 0.5600 1995.8415527 0.5800 2063.5930176
0.6000 2130.0000000 0.6200 2194.8361816 0.6400 2257.8754883
0.6600 2318.8913574 0.6800 2377.6582031 0.7000 2433.9492188
0.7200 2487.5385742 0.7400 2538.1997070 0.7600 2585.7070313
0.7800 2629.8337402 0.8000 2670.3547363 0.8200 2707.0419922
0.8400 2739.6704102 0.8600 2768.0136719 0.8800 2791.8457031
0.9000 2810.9406738 0.9200 2825.0717773 0.9400 2834.0124512
0.9600 2837.5375977 0.9800 2835.4196777 1.0000 2827.4338379
1.0200 2813.3532715 1.0400 2792.9523926 1.0600 2766.0029297
1.0800 2732.2817383 1.1000 2691.5598145

ANGLES
1 7 1 6
-999
END INPUT DATA

A3. A swinging leg

BEEN1.DAT
SWINGPHASE
JUNE 1989
0. 0.385 0.0385
0 0.001
0. 0.5 0.01 0.1
INERTIAL SPACE
SWING SPACE
PLANES
-1 -0.50 -.7879 1.0 -.7879 0 0 0 FLOOR
-999
END INERTIAL SPACE
SYSTEM 1
HUMAN LEG
CONFIGURATION
4 3 2 1
-999
GEOMETRY
0. 0. 0. 0. HIP
0. 0. 0. 0.188 UPPER LEG
0. 0.40 0. 0.186 LOWER LEG
0. 0.40 0. 0.074 FOOT
-999
**INERTIA**

| 100000000.0 | 1.00000000 |
| 7.02 | 0.084 |
| 3.37 | 0.051 |
| 1.67 | 0.001 |
| -999 |

**JOINTS**

| 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0.0 |
| 3 | 2 | 0 | 0 | 0 | 0 | 0 | 0.0 |
| 4 | 3 | 0 | 0 | 0 | 0 | 1.57079630 | 0.0 |
| -999 |

**FUNCTIONS**

| 1.524 | -1000 | -0.524 | 0.192 | 0.292 | 1000 |
| -2.92 | -1000 | -1.92 | 0.0349 | 0.10349 | 1000 |
| -2.047 | -1000 | -1.047 | 0.349 | 0.1349 | 1000 |
| -999 |

**PLANES**

| 2 | 0.0 | 0.0 | 0.0 | 0.4 | 0 | 0 | 0 | 0 | 0 | 0 | UPPER LEG |
| 3 | 0.0 | 0.0 | 0.40 | 0 | 0 | 0 | 0 | LOWER LEG |
| 4 | 0.0 | 0.0 | 0.15 | 0 | 0 | 0 | FOOT |
| -999 |

**INITIAL CONDITIONS**

| 0.0 | 0.0 | 1.58623950 | 0.1786799 |

**ORIENTATIONS**

| 1 | 0 | 3.14159265 |
| 2 | 0 | 3.193952531 |
| 3 | -1 | 2.21656815 |
| 4 | -1 | 3.16125579 |
| -999 |

**ANGULAR VELOCITY**

| 1 | 0 | 0.0 |
| 2 | 3.246312409 |
| 3 | -0.890117918 |
| 4 | -2.530727415 |
| -999 |

**END SYSTEM 1**

**FORCE MODELS**

**ACCELERATION FIELDS**

| 0 | 0 | 0 | 1.0 |
| 1 | 1 | -2 | -3 |
| -999 |

**FUNCTIONS**

| 2 | 0.0 | -10.0 | 1.0 | -10.0 |
| 66 | 0.000000 | 0.1686123 | 0.00625 | 0.0461878 | 0.01250 |
| -0.0764804 | 0.01875 | -0.1987450 | 0.02500 | -0.3199612 |
| 0.03125 | -0.4394894 | 0.03750 | -0.5566989 | 0.04375 |
| -0.6709716 | 0.05000 | -0.7817046 | 0.05625 | -0.8883134 |
| 0.06250 | -0.9902360 | 0.06875 | -1.08609346 | 0.07500 |
| -1.1778986 | 0.08125 | -1.2626488 | 0.08750 | -1.3407376 |
| 0.09375 | -1.4117533 | 0.10000 | -1.47532110 | 0.10625 |
| -1.5311056 | 0.11250 | -1.5788128 | 0.11875 | -1.6181906 |
- 47 -

0.12500 -1.6490316 0.13125 -1.6711730 0.13750 -1.6844640
-1.6844978 0.14375 -1.6889360 0.15000 -1.6849307 0.15625 -1.6889307
0.15625 -1.6711054 0.15625 -1.6649307 0.20000 -1.5309086
-1.3404535 0.20625 -1.2623386 0.21250 -1.1775641
0.21875 -1.0865769 0.22500 -0.9898577 0.23125 -0.8879164
-0.8879164 0.23750 -0.7812905 0.24375 -0.6705431
0.25000 -0.5562579 0.25625 -0.4390384 0.26250 -0.3195028
-0.3195028 0.26875 -0.1982815 0.27500 -0.0760135
0.28125 0.0466548 0.28750 0.1690770 0.29375 0.2906727
0.30000 0.4106042 0.30625 0.5284359 0.31250 0.6434789
0.31875 0.7551273 0.32500 0.8627919
0.33125 0.9659047 0.33750 1.0639224
0.34375 1.1563268 0.35000 1.2426304
1.3223790 0.36250 1.3223790
0.37500 1.5182694 0.38125 1.5679662
1.5938286 0.38750 1.6093907
0.40000 1.6665951

66
0.00000 6.4637661 0.00625 6.1734195 0.01250 5.8267035
0.01250 5.4267874 0.01875 5.2953207 0.02500 4.9773207
0.03125 4.4824047 0.03750 3.9465642 0.04375 3.746889
0.05000 2.7719967 0.05625 2.1439974 0.06250 1.4964222
0.06875 0.8351800 0.07500 0.1663153
0.08125 -0.5040679 0.08750 -1.1698519
0.09375 -1.8249511 0.10000 -4.2215042
-3.0793338 0.11250 -3.6671607 0.11875 -4.2215042
0.12500 -4.7373047 0.13125 -5.2098475 0.13750 -5.6348209
0.13750 -6.0083461 0.15000 -6.3270092
0.15625 -6.5879021 0.16250 -7.0686438 0.16875
-6.9273992 0.17500 -7.0029025 0.18125 -7.0144644
0.18750 -6.9619794 0.19375 -6.8459263 0.20000
-6.6673641 0.20625 -6.4279265 0.21250 -6.1297951
0.21875 -5.7756934 0.22500 -5.3688593 0.23125
-4.9130001 0.23750 -4.4122806 0.24375 -3.8712795
0.25000 -3.2949250 0.25625 -2.6884856 0.26250
-2.0575047 0.26875 -1.4077307 0.27500 -0.7451030
0.28125 0.0756720 0.28750 0.5944432 0.29375
1.2591375 0.30000 1.9123282 0.30625 2.5480647
0.31250 3.1605353 0.31875 3.7441421 0.32500
4.2935681 0.33125 4.8037901 0.33750 5.2701492
0.34375 5.6883845 0.35000 6.0546846 0.35625
6.3656983 0.36250 6.6185908 0.36875 6.8110504
0.37500 6.9413204 0.38125 7.0082097 0.38500
7.0176435 0.38750 7.0111098 0.39375 6.9499941
0.40000 6.8254189

END FORCE MODELS
OUTPUT
0 0 0.01925 1
LINVEL
1 1 0. 0 HIP

LINACC
1 1 0. 0 0 0 HIP
-999
END OUTPUT
USER INPUT
EXTMOM
3
1 1 2 1 'HIP'
2 1 3 2 'KNEE'
2 1 4 3 'ANKLE'
3
12 0.0 27.0 0.07999 10.0 0.080 10.0 0.16599 5.0 0.166
5.0 0.18 0.0 0.181 0.0 0.21089 -28.5 0.2109 -28.5
0.3199 -26.5 0.320 -26.5 0.45 -28.5
12 0.0 4.5 0.015 2.5 0.01501 2.5 0.079 5.75 0.0791
5.75 0.170 0.025 0.1701 0.025 0.23099 -18.0 0.231 -18.0
0.38 -19.5 0.3801 -19.5 0.45 -22.0
18 0.0 2.8 0.034 1.8 0.0341 1.8 0.068 1.4 0.0681
1.4 0.06999 1.4 0.070 1.4 0.1599 1.2 0.16 1.2
0.2 1.4 0.2001 1.4 0.27999 1.0 0.280 1.0 0.3199
0.2 0.32 0.2 0.35 0.125 0.351 0.125 0.45 0.0
ANGLES
1 1 1 1
1 2 1 1
1 3 1 2
1 4 1 3
-999
END INPUT DATA
SOME ASPECTS OF THE USE OF MADYMO IN GENERAL BIOMECHANICS ANALYSIS

PART III: MUSCLE MODELLING

Ir. D.K.I. Hoekstra

December 1989
SOME ASPECTS OF THE USE OF MADYMO IN GENERAL BIOMECHANICS ANALYSIS

PART III : MUSCLE MODELLING

Ir. D.K.I. HOEKSTRA

DELMFT, DECEMBER 1989
CONTENTS

Preface 1

Chapter 1. Introduction 2
1.1 Purpose of the research 2
1.2 Overview of the contents of the report 2

Chapter 2. Literature study of muscle modelling 3
2.1 Introduction 3
2.2 Review of biomechanics literature 4
2.2.1 Preliminaries 4
2.2.2 Musculo-skeletal models 7
2.2.3 Muscle models 10
2.2.4 Incorporation of muscle models in musculo-skeletal models 14
2.3 Conclusions 16

Chapter 3. Muscle forces and torques in MADYMO 17
3.1 Introduction 17
3.2 What is MADYMO 17
3.3 User-defined subroutines 18
3.4 Validation of the results 20
3.4.1 Simulation of a 2D simple analytical example 20
3.4.2 Simulation of a 3D simple analytical example 24
3.4.3 Simulation of a subject doing a push-up 32

Chapter 4. Discussion, conclusion and recommendations 37

References 39

Appendix A. Muscle modelling activities in the Netherlands 44

Appendix B. User's guide to the subroutines TIMFIN, TIMMIN and ANGIN2 45
B1. The subroutine TIMFIN (2D) 45
B2. The subroutine TIMMIN (2D) 47
B3. The subroutine ANGIN2 (2D) 49
B4. The subroutine TIMFIN (3D) 50
B5. The subroutine TIMMIN (3D) 51

Appendix C. Source codes of the user-defined subroutines 54
C1. The 2D routines USERSY, USERIN, TIMFIN, TIMMIN,
ANGIN2, USERFO, TIMFOR, TIMMOM, USEROU and ANGOU2

C2. The 3D routines USERSY, USERIN, TIMFIN, TIMMIN, USERFO, TIMFOR and TIMMOM

Appendix D. MADYMO input files for the simulations in Chapter 3

D1. 2D analytical example
D2. 3D analytical example
D3. A subject doing a push-up
PREFACE

In October 1987 I started my study in the postgraduate program "Mathematics in Industry" at the Eindhoven University of Technology (TUE). The program participates in the European Consortium for Mathematics in Industry (ECMI). In modern industry, mathematical models play an increasingly important role in research and development for production, distribution and management. The central aim of the program is to provide mathematicians and other university graduates in mathematically oriented fields with the training, necessary to work successfully as a mathematician in industry. A secondary aim of the program is to establish a European network of groups that covers the whole area of mathematical applications in industry and to use that network to stimulate the use of mathematics in industry. Each student concludes the postgraduate program with a final project in an industrial environment. In my case this environment was the Netherlands Organization for Applied Scientific Research (TNO) in Delft. There, at the Injury Prevention Department of the TNO Road-Vehicles Research Institute, the Crash Victim Simulation program MADYMO is developed. MADYMO stands for MAthematical DYnamic MOdelling. This computer program can simulate occupant behavior in car crashes.

I was leader of a so-called combination project in which also two graduate students took part. My individual project contribution on muscle modelling is described in this part (i.e. Part III) of the final report. The organization and the technical realization of the project can be found in Part I and Part II of the report, respectively. The first graduate student (Pieter Vosbeek) has written a Master's thesis in which he presents two contributions to the modelling of the human body in MADYMO. The second student (Rob Vermazeren) is still working on MADYMO pre- and postprocessing facilities. His Master's thesis will appear in the beginning of 1990.

I thank Pieter and Rob for their efforts and the pleasant cooperation. On behalf of both of them I further thank their supervisors Fons van de Ven and Kees van Overveld and TNO staff Willy Koppens and Harry Heinen for their help. The project as a whole was supervised by Jos Jansen and Jac Wismans, whom I thank for their suggestions and for the fruitful discussions that we have had. A final acknowledgment goes to the TNO Road-Vehicles Research Institute, that provided the opportunity to carry out the first combination project of this type and that was so kind to employ me for an additional three months, so that I could extend this interesting project.
CHAPTER 1
INTRODUCTION

1.1 Purpose of the research

At the Injury Prevention Department of the TNO Road-Vehicles Research Institute the computer program MADYMO has been developed for simulations of occupant kinematics in car crashes. In other words, MADYMO (MAthematical DYnamic MOdelling) is a Crash Victim Simulation program. In MADYMO the human body is modelled as a system of rigid bodies (elements), which are connected by joints. The systems must be tree structures, which do not contain closed loops of elements. The model of the human body is passive in the sense that it can only move when an external force is applied to it. Two- and three-dimensional simulations are possible with the 2D and 3D version of MADYMO, respectively. In MADYMO 2D the connections (joints) between the elements are of the hinge type, while in MADYMO 3D ball and socket type joints are used to link the elements. In the MADYMO User's Manuals (1988) more detailed information about the program can be found.

The objective of this research was to find out to what extent MADYMO is applicable in general biomechanical analysis, like sports activities or studies of working postures. For this purpose muscle activity was incorporated in the model, so that the human body can move actively.

1.2 Overview of the contents of the report

The incorporation of muscle activity in the MADYMO human body model is described in this report. In the following chapter a review of literature on the modelling of muscle activity is given. A distinction is made between studies of musculo-skeletal models and those of muscle models. In the first type of modelling the muscle is seen as a black box, which causes a certain force or torque. The other models describe the relation between muscular properties and produced muscle force. Both ways of modelling can be further differentiated, as will be shown later. The selection of the level of complexity of the muscle model to be incorporated in MADYMO within this study is based on the findings from literature and on discussions with experts in the field of muscle modelling (see Chapter 2).

In Chapter 3 the user-defined subroutines, that allow the MADYMO user to include muscular behavior will be presented (both 2D and 3D). These routines were tested using simulations of simple two- and three-dimensional motions that are analytically known. An example of its use is given by the simulation of a subject doing a push-up (section 3.4).

In the final chapter the results of the research will be discussed and some suggestions for future research will be given.
2.1 Introduction

The modelling of muscle activity can be considered at various levels. One can distinguish between musculo-skeletal models, in which the muscle is treated as a black box, and muscle models, in which the internal behavior of the muscle is described. In the first case the muscle is merely a force or torque generator. In the second case the relation between the internal muscle behavior and the force or torque that is caused by that behavior is specified. Both ways of modelling muscle activity can be further subdivided.

In musculo-skeletal modelling, the most global way to include muscle activity is by describing the resultant torque around a joint caused by all muscular, ligament and friction forces acting on the joint. At a more detailed level, the force caused by a group of lumped muscles or by an individual muscle is prescribed. In this case, the muscles are viewed as force generators. Musculo-skeletal models are used to describe gross motion of (part of) the human body caused by muscular and external forces.

In muscle modelling, a distinction can be made between macroscopic and microscopic models. Macroscopic muscle models imply the use of springs and dampers with adjustable, nonlinear characteristics. By tuning these properties carefully a wide range of functional behavior can be simulated, including force-length and force-velocity characteristics, which depend on the stimulation rate. The stimulation rate is a measure of the active state of the muscle and is related to the concentration of calcium ions in the muscle. Macroscopic muscle models are mostly based on Hill's model (Hill, 1938 and Hof, 1980). They do not give any insight in the physical processes within the muscle. Therefore more detailed models are necessary.

In microscopic models the actual muscle structure is taken into account. They describe the behavior of single muscle fibers or groups of fibers. Some microscopic models consider individual or small groups of motor units. A motor unit consists of a number of homogeneous muscle fibers and a motor neuron. The motor neuron, which is situated in the spinal cord, transports electrical pulses from the central nervous system to so-called motor end plates on the muscle fibers. Thereupon, the fibers contract as a result of the neural stimulation. From the mechanical behavior of the motor units that constitute the muscle one can predict the spatial process of contraction and force development of the whole muscle.

Clearly the microscopic approach offers more than the macroscopic one. Its numerical simulation however, demands much more time and it requires many more parameters to be defined.
The aim of this study is to gain more insight in the various ways of modelling muscle activity, in order to select a suitable way to incorporate muscular behavior in the MADYMO human body model. Should musculo-skeletal models be considered only, or is it needed to build in macroscopic or even microscopic models? To answer this question we need to know, whether Hill-like or microscopic muscle models have ever been used in models of the human body, and whether it is possible to use a microscopic model to predict parameter values of a macroscopic model.

In section 2.2 a detailed review of various muscle models will be given with the objective to answer the above questions. Appendix A summarizes the work carried out in the Netherlands in this field. Conclusions relevant for future inclusions of muscle activity in MADYMO are given in section 2.3.

2.2 Review of biomechanics literature

2.2.1 Preliminaries

First, some of the terminology used in biomechanics literature will be explained. When muscle activity is considered at the level of musculo-skeletal models, one is interested in the motion of the musculo-skeletal system caused by external and muscular forces. The kinematics of the system define its geometrical and topological characteristics. By contrast, the kinetics or dynamics of movement is concerned with its Newtonian mechanical properties. That is, given the mass and the moment of inertia of each rigid body in the system, and the presence of external forces and torques, dynamics defines the relationship between the applied forces and torques and the linear and angular acceleration and velocity and the position and orientation of the rigid body. The movement of the system is governed by the equations of motion (or by the equations of equilibrium in the static case. Note, that the equilibrium equations are a special case of the equations of motion). The equations of motion of the system can be applied in two ways. Either a forward or direct dynamics or an inverse dynamics approach can be followed. The computation of the kinematics (given initial position and velocity of the system) from the kinetics of the system is called the direct dynamics problem. Solving this problem allows us to simulate the dynamics of the system using given input forces and torques. Computing the kinetics from the kinematics of motion, on the other hand, is called the inverse dynamics problem. Here, the forces and torques, that are necessary for a certain desired motion, are calculated from the kinematics of that motion.

Since both methods are each other's inverse, one can be used to check the other (Figure 2.2.1). Let's assume that forces and torques are calculated from the kinematics of a certain motion using the inverse dynamics method. These forces and torques can be used as input in a direct dynamics model. The kinematics output of this model should then be equal to the original kinematics. Ju and Mansour (1988) used this strategy in their simulation of the double limb support phase of human gait.
Sometimes a mixture of direct and inverse dynamics is used. Hoekstra (1987), for example, used a time-dependent force and a kinematical constraint (a desired relationship between joint angles) as input of his two-dimensional model of a dart-throwing arm. From this mixed input a torque and the kinematics of the motion of the arm are calculated. Thus, the output is also a mixture of kinetics and kinematics. (The model is described in Chapter 3 of part II of this report).

The major joints in the human body are controlled by several muscles, which are active simultaneously during most movements (synergistic muscle action). For example, at least three muscles contribute to knee flexion. Strictly speaking, one muscle would be sufficient to realize such a motion. This is called "anatomical redundancy" in the musculo-skeletal system. Anatomical redundancy leads to an indeterminate problem: there are more unknown muscle forces than equations of motion. A possible solution to this problem can be obtained by formulating an objective function and utilizing an optimization technique (such as the simplex method in the case of a linear objective function). Various objective functions have been defined in literature: minimizing the forces in the muscles, minimizing the total work done by the muscles, minimizing vertical reaction forces in the joints, minimizing muscle fatigue, etcetera. Dul and co-workers (1984a) and Dul (1985) have given an extensive review of optimization criteria that are used in biomechanics studies and they compare the results of several linear and nonlinear criteria reported in literature.

Muscle fibers are activated by neural pulses that travel from the central nervous system via the motor neurons to the motor end plates of the fibers. This electrical excitation of the muscle can be measured using an electromyogram (EMG), in which electrical activity in the muscle is shown as a function of time. Hof (1980 and 1984) has given a clear description of the electromyogram and its use. He has indicated how EMG measurements may be used in relation to force, mechanical work, energy consumption and activity of the muscle. In his Ph.D. thesis Hof used the EMG signal as input of his Hill-based muscle model. He assumed that the EMG signal is a measure of the active state of the muscle. Muscular force is not directly measurable. Therefore one would like to determine the relationship between (measurable) myoelectric activity and muscle force. Hof (1980), Ray and Guha
(1983) and Dul (1984) have claimed that there exists a linear relationship between EMG and muscle tension. Others, who used less complicated models came up with nonlinear relations. Agarwal and Gottlieb (1982) give a review of literature on studies of the electrical properties of the muscle fibers and on studies of the relation between the EMG and the force that is produced by the excited muscle.

It is not yet understood how the central nervous system controls muscular activity by sending electrical pulses that are delivered to the muscle fibers via the spinal cord. For some people the communication between the brain and the skeletal muscles is blocked. The functional mobility of these paralyzed people can be restored by means of artificial stimulation of the muscles. An external controller could generate appropriate command signals that stimulate the muscles. The controller takes over the function of the central nervous system. This is called functional neuromuscular stimulation (FNS). Control and stimulation of human movement is described by FitzHugh (1977), Zheng et al. (1984) and Khang (1988). All of them have used a Hill-like muscle model, in which the active state of the muscle is related to the stimulus parameter (motor signal). Once there exists a reliable model of the central nervous system, this model can be used to provide a muscle model with electrical pulses that are necessary for a certain contraction of the muscle. Hemami (1985) has given a detailed review of works on modelling of the central nervous system and of research on human movement control.

According to Hof (1984) there are two ways of controlling muscle force: changing the number of active motor units ('recruitment') and changing the firing frequency of the active units ('rate coding'). When someone increases the force of a muscle, the motor units are recruited one by one in order of magnitude ('size principle'). Motor units are considered to be on-off units. There exist only models of a single motor unit, in which the unit is activated for one hundred percent. A muscle does not always need to be activated maximally, because not all of its motor units need to be activated at the same time.

One of the most challenging aspects of micro- and macroscopic muscle modelling is parameter estimation. One can develop a very detailed and ingenious model to describe muscular behavior, but when the muscle parameters that are needed in the model can not be determined accurately, the advantage of the detailed model is cancelled out by the inaccuracies. It is an elaborate task to obtain muscle parameters from experiments. A large amount of anthropometric data have been gathered over the years (for instance by the US Airforce), but very little measurements were done to assess anthropometric muscle data. Many researchers use data from literature. Brand et al. (1982) have measured origin and insertion (the positions where a muscle is attached to the skeleton) of 47 muscles in the lower limb crossing the hip, knee and ankle joints at three cadavers (six limbs). The values of the physiological cross-sectional area (PCSA) of the 47 muscles for the three data sets are presented in Brand et al. (1986). Winters and Stark (1988) considered 46 muscles involved in flexion and
extension of the elbow, wrist, knee and ankle joint. They provide estimations of parameter values that define the nonlinear torque-velocity relations within the Hill-based muscle model structure. Further, they give origin and insertion and average fiber length, mass, pennation angle and physiological cross-sectional area of each muscle. Other authors that present parameter estimations of muscle or muscle fiber characteristics are referred to by Otten (1988).

In the following section an overview is given of musculo-skeletal models. First, models considering human gait are treated. Next, some models in which other human movements occur are discussed. Then, literature on microscopic and macroscopic muscle models is reviewed (section 2.2.3). This chapter is concluded with some examples of studies in which muscle models are incorporated in musculo-skeletal models.

2.2.2 Musculo-skeletal models

One of the basic movements of humans is walking. Therefore, first attention is paid to the musculo-skeletal modelling of human locomotion. After that, studies in which other types of human movement is modelled are discussed.

First, two-dimensional human gait studies are considered. Onyshko and Winter (1980), Mena et al. (1981) and Pandy and Berme (1988a,b) all used a direct dynamics approach to study human gait. In their models initial conditions and joint torques are prescribed. Mena considered the swing phase of human walking with the torques acting on the three elements representing thigh, shank and foot. The other authors described a complete cycle of human locomotion. In these models the whole body is represented by a system of seven rigid bodies: three for each leg and one for the head-arms-trunk system. A distinction is made between different phases of the motion (single or double stance; foot flat or foot pivoting around the toes). Onyshko and Winter considered four phases requiring two different mathematical representations, while Pandy and Berme distinguished three phases involving three different mathematical representations. From both studies it seems that the coupling of the phases is a complicated matter. Kinematical constraints for the stance feet keep the body from sinking in the ground. Further, a stiff spring-damper actuator is modelled at the hip to allow for a smooth landing of the foot on the ground. Since joint torques are not directly measurable, one may wonder how the torques used in the direct dynamics approach were obtained. Onyshko and Winter and Mena and co-workers achieved the joint torques through inverse dynamics. Pandy and Berme, however, chose their joint moments of force on the basis of trial and error.

Most of the effort related to gait studies has been directed to solving for the forces and torques that are applied to a system based on motion information of the system from laboratory measurements, thus to the inverse dynamics problem. Studies in which an inverse dynamics approach is used will be discussed
Cappozzo et al. (1975) have recorded movement and ground reaction forces of a single leg during one complete cycle of gait, both for level walking and walking up stairs. They extensively describe a statistical method used to reduce measurement errors in the experimental data. The filtered data were used in their mathematical model for the leg to calculate resultant torques around hip, knee and ankle joint. Further, the relative range of uncertainty of the torque values were calculated. In a later study (Cappozzo et al., 1976) a seven link model for the human body was used to simulate walking. An analysis was carried out of the energy absorption and energy generation by the ankle, knee and hip muscles using the product of muscle moment of force and joint angular velocity. Muscular phasic activity was assessed by means of electromyographic (EMG) recordings. The calculated joint torques showed a qualitative correlation to the muscle activity. Winter and Robertson (1978) proceeded in a similar way. They also used an energy analysis in their simulations of level walking, in which the leg was modelled by three links. Apart from the absorption and generation of energy by the joint muscles, they further considered the flow of energy between the body segments. From measured motion kinematics the resultant torques around the hip, knee and ankle joint were calculated. Special attention was given to the knee joint. They found that during the swing phase three forces (gravitational, muscle and knee joint acceleration) are responsible for shank rotation, and that these forces act together (synergism) during both acceleration and deceleration of the swing phase.

The discussion of two-dimensional gait will be concluded with a mixed forward-inverse dynamics method. Van den Bogert (1988) simulated the (quadrupedal) locomotion of the horse using a mathematical model in which the horse is represented by a tree structure consisting of twenty joint-connected rigid bodies. Muscle activity is considered only at the level of resultant torques around the joints, thus individual muscle forces are not taken into account. Part of the joints are constrained to follow a certain kinematics pattern (prescribed rotations as a function of time), while the movement of the other joints is controlled by a torque generator. This torque generator produces a torque that causes the joint angle and angular velocity to approach a desired time-dependent joint angle and angular velocity. (In other words, the desired joint angle acts as the control function). The torque generator suggested by Van den Bogert yields:

\[
M(\phi, \omega, t) = -A(\phi(t) - \phi_d(t)) - B(\omega(t) - \omega_d(t))
\]

Ground-hoof interaction forces are approximated by a viscoelastic model and pseudo-Coulomb friction in vertical and horizontal directions, respectively. This approximation of interaction with the ground seems more realistic than the one used by Onyshko and Winter and Pandy and Berme in their direct dynamics models of human locomotion.
Next, three-dimensional studies of human gait will be discussed. Apkarian et al. (1989) have developed a model in which the lower limb is modelled as a sequence of four rigid bodies (pelvis, thigh, shank, foot) connected by three universal rotary joints representing the hip, knee and ankle joints. Each joint is modelled as a sequence of three single axis rotational joints, thus in each joint there are three degrees of freedom. A method was presented to measure the gait variables so that all nine angles can be computed based on the positions of nine markers placed on the subject during a gait study. The gait variables were used in a Newton-Euler inverse dynamics formulation to compute the torques exerted about the three axes of each joint during gait (thus, a total of nine torques was assessed). The authors claimed that a three-dimensional analysis of gait was required, because considerable movements outside the sagittal plane were observed.

Seireg and Arvikar (1975) were the first to attack the problem of redundant muscle forces (indeterminacy). Using seven body segments to simulate walking, they had 104 unknowns (31 muscles on each side of the sagittal plane, three joint reaction and movement components at six joints, and three patellar reactions on each side), but only 42 equations of motion. To solve for the 104 unknowns, they used an objective of minimizing a weighted sum of forces in all the significant muscles in the lower extremities and the torques at all the joints. According to the authors the calculated muscle forces showed excellent agreement with typical EMG patterns for all the major muscles involve in level walking.

Dul and Shiavi (1985) have developed a model of the human ankle and its musculature. They described the ankle as a joint with two rotational degrees of freedom corresponding to the gross motions of the foot relative to the shank. Twelve muscles crossing both rotational axes were included in the analysis. For a given (instantaneous) position and orientation of the foot and a given magnitude and direction of the ground reaction force, the model provides two moment equilibrium equations. The indeterminate problem is solved with a minimum-fatigue optimization technique (Dul et al., 1984a,b). Muscle forces were computed for standing (static) and walking (quasi-static). Data on foot positions and ground reaction forces were obtained from literature.

Others, who obtained muscle forces using inverse dynamics and optimization were Brand et al. (1986). They determined hip, knee and ankle torques from measured ground reaction forces and gait analysis. From these torques muscle forces of all 47 muscles crossing the three joints were achieved. The objective function to be minimized was the sum of the cubed muscle stresses. Muscle stress was defined to be muscle force divided by physiological cross-sectional area (PCSA) of the muscle.

Chen et al. (1986) used a mixed forward-inverse dynamics approach similar to that of Van den Bogert for two dimensions. Their control torque at the joints is nonlinear and is not just based on the desired joint kinematics, but also on the desired ground
reaction forces. They used a seven-link model to simulate a right turn in human walking. From the control torques, which cause a motion that approaches the desired motion, 22 muscle forces in the lower extremities were computed using the objective of minimizing the sum of the squared forces.

After this overview of musculo-skeletal models in human gait, models of other human movement are discussed. Otten (1987) developed a myocytbernetic model of the jaw system of the rat in which also micro- and macroscopic muscle models are incorporated. The muscle models produce force that are used as input for the musculo-skeletal model. This work will be discussed in section 2.2.4. In two recent articles Dul (1988) and Buchner et al. (1988) used inverse dynamics to obtain individual muscle forces from 2D motion kinematics. Dul described a biomechanical model to predict forces in shoulder muscles (deltoid and supraspinatus) and the reaction force at the shoulder joint in static elevated arm postures. Furthermore, the model estimated the endurance time of the given arm position. This is the period that elapses until the shoulder muscles become fatigued. There are more unknowns (two muscle and two joint reaction forces) than equilibrium equations (three). Therefore, an optimization technique that minimizes muscular fatigue is used to compute the four forces.

Buchner, Hines and Hemami have developed a dynamic model for finger interphalangeal coordination. From trajectories of desired finger motion eight muscle forces are derived using optimization. The objective was to minimize the sum of the squared muscle stresses.

The final inverse dynamics musculo-skeletal model that will be discussed here is that of static-isometric knee flexion, which is presented in Dul et al. (1984a) and Dul (1985). The leg was modelled by two rigid bodies connected by a hinge type joint representing the knee. Three groups of lumped muscles were included in the model: long and short hamstrings and gastrocnemius. Moment arms for the forces caused by the three muscle groups were estimated from literature. Optimization techniques were used to derive the forces.

Sol (1980) gives an extensive review of literature on musculo-skeletal models.

2.2.3 Muscle models

Muscle models are subdivided into macroscopic and microscopic models. Most macroscopic muscle models are based on the model of Hill (1938) (see Figure 2.2.3.1). Hill’s muscle model consists of three components. The parallel elastic component (PEC) represents the elasticity of the passive muscle and the ligaments. The behavior of the active muscle is determined by the contractile component (CC) and the series elastic component (SEC).
Figure 2.2.3.1: Hill's muscle model

The standardized length of the muscle \( x = x_c + x_e \) depends on the angle of the joint between the elements to which the muscle is attached. The standardized length is defined as: \( x = l/l_0 \), where \( l \) and \( l_0 \) are actual and rest length of the muscle. The length of the PEC is equal to the length of the muscle. The muscle force \( F \) is equal to the sum of \( F_c \) and \( F_p \). The forces in CC and SEC are both equal to \( F_c \). The active state of the muscle \( F_0 \) is a measure of muscle activity. If the muscle is not active, then \( F_0 = 0 \), and when the muscle activity is maximal, \( F_0 = F_m \). The total force \( F \) that is produced by the muscle can be derived from the following relations for the three components in the Hill model:

- The passive force \( F_p \) of the PEC is expressed as a (nonlinear) function of whole muscle length \( x \).

- For the CC there is a force-length relation \( F_c = F_0 f(x_e) \) in case the contraction is isometric (muscle length \( l \) is constant).

- In case of contraction of the CC the force-velocity relation \( F_c = F_0 g(x_c, v_c) \) holds, where \( v_c \) is the contraction velocity of the CC.

- The relationship between the length of the SEC \( x_e \) and the applied force \( F_c \) is of a similar nature as the one between \( F_p \) and \( x \).

The force-velocity relationship is known as Hill's relation. The functions \( f \) and \( g \) are chosen such, that \( g(x_c, 0) = f(x_c) \). When \( x_c \) equals its rest value, then \( f(x_c) = 1 \) and \( F_c = F_0 \). Thus, the active state is equal to the tension that is developed, when the CC is neither lengthening nor shortening, while its length is equal to its rest length. The model is described in more detail by Hill (1938) and Hof (1980, 1987).

It is assumed that the active state \( F_0 \) in the Hill model is not affected by the mechanical events: force, length and velocity.
From both relations for the CC it can be seen, that muscle force is a linear function of the active state. Recent studies (Ray and Guha, 1983; Dul, 1984) show a linear relationship between muscle force and electromyographic activity. This implies a linear relation between the active state and the myoelectric activity of the muscle. In other words, the EMG signal may be used as a (qualitative) measure of the active state in Hill's muscle model. This was already assumed by Hof (1980), who used EMG recordings as input for his model of the human calves.

In his study of optimal voluntary muscle control, FitzHugh (1977) used a simplified Hill-type model to describe the mechanics of a single muscle. The active state of the muscle was represented by one input variable. A mathematical optimization technique was applied to establish the input signal that minimizes the total energy expended by the muscle during shortening or lengthening. The work done by the muscle equals muscle force times contraction or lengthening velocity. The model may also be used for time optimization in motions in which speed rather than endurance is important.

Sometimes (e.g. Hof, 1980; Baildon and Chapman, 1983) the model of Hill is used as a basis for models that produce torque around a joint caused by the muscle instead of muscle force. In these models torque-angle and torque-velocity relations are used. The advantage of using muscle torques and joint angles is that they can be measured directly.

In section 2.2.4 examples of studies, in which Hill-like muscle models are incorporated in musculo-skeletal models, are given.

The definition of microscopic muscle models that was given in the introduction is somewhat broad. It covers every model that goes into more detail than a Hill-type model. In microscopic muscle models the actual muscle structure is taken into account. Figure 2.2.3.2 shows the structure of a skeletal muscle. The muscle consists of bundles of muscle fibers. In combination with a motor neuron a bundle of fibers forms a motor unit. The motor neuron, which is situated in the spinal cord, transports electric signals from the central nervous system to the muscle fibers. The fibers contract as a result of this neuro-stimulation. More than one group of fibers is connected to the motor neuron, so that several groups are activated by one pulse coming from the motor neuron. The myofibrils, which constitute a single muscle fiber, consist of myofilaments. Actine- and myosine filaments can be distinguished. Muscle structure is extensively described by Bernards and Bouman (1983) and Hof (1987).

Since the behavior of muscle fibers is similar to that of the muscle, Hill-like models can also be used to describe the dynamics of single fibers or groups of fibers. An example of this type of modelling will be given later, when the work of Zheng et al. (1984) is discussed. Huxley (1957) developed the so-called sliding filament theory, which explains the shortening and lengthening of muscle fibers caused by changes in the concentration of calcium ions in the fibers. The theory can be
used to interpret the force-length relation of the contractile component (CC) of Hill's model.

Figure 2.2.3.2: Structure of a skeletal muscle (from Bernards and Bouman, 1983)

A critical review of twelve publications on functional models of skeletal muscle microstructure is given by Otten (1988). He proposes revisions of a number of concepts and models, which concern in particular:

- principles of the change of shape of muscles during shortening and lengthening,
- the relation between muscle microstructure and force-length diagrams,

- the relation between muscle microstructure and intramuscular pressure.

The review is clarifying to those, who are new to the subject of microscopic muscle modelling.

Other reviews are given by Hemami (1985) and Agarwal and Gottlieb (1982). They contain 341, respectively 288 references, in which more details on muscle modelling can be found.

2.2.4 Incorporation of muscle models in musculo-skeletal models

Two-dimensional models of human standing, in which macroscopic muscle models are used, are described by Hof (1980), Andrews (1985) and Khang (1988). The first author has developed a model that predicts the ankle torque produced by the human calf muscles from a given EMG signal and the angle of the ankle joint. The essential part of the model is an electrical analogon of the Hill muscle model. It was assumed that the intensity of the EMG signal is a measure for the muscle activity or (more specific) of the active state of the muscle. Muscle length and contraction velocity are computed from the ankle joint angle and angular velocity. The method is suitable for the assessment of muscle torque and work in movement studies like walking, running and cycling. This is illustrated in Hof et al. (1983).

The work of Khang (1988) is a nice illustration of the use of a macroscopic muscle model in a musculo-skeletal model, where also muscular control is taken into account. He has developed a computer model to investigate standing induced by functional neuromuscular stimulation. The model consists of musculo-tendon dynamics, body-segmental dynamics, and an output feedback control law. Hill's muscle model was modified into a musculo-tendon actuator model, in which apart from muscle activity also tendon dynamics was taken into account. This musculo-tendon actuator was used to describe the behavior of thirteen muscles in the lower extremity. The feedback control strategy computes activation signals for the muscles, in such manner that the muscular forces cause the body to attain a certain desired position. Two types of body motion were simulated. First, the segmental orientations were initially perturbed from the vertical, while all muscles were assumed to be inactive at $t = 0$. Once the upright position was recovered, an external force was applied to the body to investigate how the muscles acted to overcome this second disturbance.

Hatze (1977) presents a three-dimensional model of the total human musculo-skeletal system in which equations concerning the body segment dynamics are coupled with those that describe muscle dynamics. The muscle model used is an extension of the Hill model. It also contains a (linear) parallel damping component to describe the relation between the damping force and the
contraction velocity of the muscle. Seventeen body segments and a large number of muscles are modelled, leading to a system with almost 300 (!) degrees of freedom. The model can be used to simulate human gait and other gross body motions. The huge amount of parameters make the model very complicated and therefore many researchers in the field of biomechanics remain sceptical about its performance and its usefulness. The model of Koopman (1989) is much more simple and is developed to simulate human gait only. It consists of seven joint-connected rigid bodies, while in each leg twelve muscles are represented by the same muscle model that was used by Hatze.

Another example of the use of macro models in human movement is the work of Winters and Stark (1985). They considered groups of lumped muscles that cause joint rotation. The resultant torque around a joint caused by the muscle groups is obtained using a Hill-type model similar to that of Hof (1980). Muscle parameters were obtained from literature. Simulations of flexion and extension of the knee, ankle, wrist and elbow joints and wrist, head and eye rotation showed good agreement with results found in literature.

In the final two studies, which will be discussed here, microscopic muscle models were incorporated in models of the musculo-skeletal system. Zheng et al. (1984) used a Hill-based model to describe the dynamics of a single motor unit. Contraction of the motor unit is caused by neural stimulations. They also presented a model for the whole muscle consisting of a number of motor units. Muscle contraction is caused by the contraction of the individual motor units. The authors propose a model for the recruitment of the motor units, which serves as input for the muscle model. This muscle model has been implemented in a model that describes the rotational movement of the elbow joint caused by the biceps and triceps muscles.

The work of Otten (1987) concerns a myocybernetic model of the jaw system of the rat, which is used to simulate chewing movements in rats. The muscle model consists of the following system of nested units:

- a static muscle fiber model, relating force with length, based on fiber structure information.

- a static skeletal muscle model, relating muscle force with muscle length, based on muscle structure information and the static fiber model.

- a dynamic skeletal muscle model, relating muscle force with muscle length, contraction velocity and stimulation rate, based on fiber type data, the concentration of calcium ions and force-velocity relations and on the static skeletal muscle model.

Measured EMG data of chewing rats and data of muscle properties are put into the dynamic model resulting in muscle force. Thirteen muscles on each side of the head cause the motion of the
lower jaw and the hyoid of the rat. The muscle model, a kinematic model of the jaw system and an optimization technique are used to compute the kinematics of the chewing movement. The optimization algorithm is based on the minimization of the total energy in the system. This energy consists of the strain energy in the muscles, the potential (gravitational) energy of all muscles and of bony and cartilaginous elements, the kinetical energy of the elements and the energy taken up by the deforming food.

2.3 Conclusions

Muscle activity is being modelled at various levels. Human movement is studied using musculo-skeletal models, in which muscular torques and forces are applied to the skeletal system. In these models the muscle is seen as a black box. The functional role of the muscle itself is described by macro- and microscopic muscle models. In macro models springs and dampers are used to simulate the mechanical behavior of the muscle, while micro models describe the actual structure of the muscle. Both types of muscle models can be used to predict the force that a muscle produces. It is possible to use both macro and micro models to provide muscle forces or torques in musculo-skeletal models.

Hof, Otten and Dul have stressed the fact of the parameter estimation problem. Very little reliable data of muscle properties are available. Therefore muscle models should be used with care.

MADYMO is a computer program with which gross motion of systems of rigid bodies can be simulated. The purpose of this study is to improve the model of the human body in MADYMO. From the literature study and from discussions with experts in the field of muscle modelling insight was gained in the various possibilities for the modelling of muscle activity. It was decided to consider only musculo-skeletal modelling in MADYMO at the present stage. (Due to a lack of time a Hill-type muscle model was not yet considered). Muscle activity will be restricted to muscle forces and torques. In the next chapter it is shown how MADYMO is extended, so that it is possible to define time-dependent (muscular) forces and torques, which act on certain elements or around certain joints of the human body model.
CHAPTER 3
MUSCLE FORCES AND TORQUES IN MADYMO

3.1 Introduction

It is desired to improve the modelling of the human body in MADYMO in order to make it suitable for application in general biomechanics analysis. One way to get a more realistic model is to incorporate muscle activity in the model. Therefore, user-defined subroutines, that allow the user to prescribe muscle forces and torques acting on the model of the human body, have to be added to MADYMO. After a survey of the features of the MADYMO package in the next section, the user-defined subroutines will be presented in section 3.3. In section 3.4 the routines are validated. In the first two paragraphs of that section forces and torques are prescribed acting on single elements and causing the elements to move in an analytically known way. By comparing analytical and numerical (MADYMO) results it can be seen, whether the user-defined subroutines perform well. The examples of MADYMO simulations using human body modelling in Chapter 3 of part II of this report render a further validation of the routines.

MADYMO is a direct dynamics simulation program. It calculates the kinematics of the motion of a system of rigid bodies from the prescribed forces and torques that cause the motion. In 3.4.3 it is shown how the extended MADYMO program can be used to calculate resulting joint torques from given kinematical data.

3.2 What is MADYMO?

MADYMO (MAthematical DYnamic MOdelling) is a computer simulation program for multi-body dynamics. In other words, it is a program for simulating the dynamic motion of one or more linkage systems that undergo large displacements (gross motion). Such systems are also known as tree structures. A tree structure can not contain any closed loop of elements. Two- and three-dimensional simulations are possible with the 2D and 3D version of MADYMO. The number of elements (rigid bodies) in each linkage system can be selected freely. In MADYMO 2D the connections (joints) between the elements in a linkage system are of the hinge type. In MADYMO 3D ball and socket type joints are used to link the elements. The MADYMO program has been developed especially for the simulation of the motion of the human body during an impact and it has special features for occupant behavior in car crash simulations.

MADYMO is a forward or direct dynamics program. It calculates the kinematics of tree structures of joint-connected elements from the external forces that are acting on the elements. An element can be in force interaction with another element or with the surroundings. Force interactions can consist of, for example, impact forces, belt forces, spring-damper forces or acceleration...
The program automatically generates and solves the equations of motion for the structures of rigid bodies. The generation of the equations of motion is based on the Lagrange method (see Wittenburg, 1977). The equations of motion form a set of nonlinear second order differential equations. For each tree structure such a set of equations is generated. The solution of these equations is obtained by numerical integration.

In order to carry out a simulation, MADYMO needs an input file in which the model is described. A MADYMO input file contains information about the number of systems and the number of elements in each system. Each rigid body element requires the specification of its geometry, the position of its center of mass, its mass and its moment(s) of inertia. (The system of units used in MADYMO is the International System (SI). The basic units in MADYMO are mass (kg), length (m), and time (s). Angles are expressed in radians). The characteristics of the joints, the force-interaction properties and the initial conditions of the systems are also defined in the input file, where keywords are used to identify data blocks. The data in the input file completely describe the simulation model of which MADYMO generates and solves the set of nonlinear equations of motion.

Output control parameters in the input file specify the output requested from a MADYMO simulation. This output can consist of time-histories of (relative) displacements, linear and angular velocities and accelerations, forces, torques, etcetera. Results of a simulation are stored in a number of output files, which can be used by post-processing programs to produce graphs, tables or animations.

For special modelling purposes it is possible to add user-defined subroutines to the MADYMO program. With these the user can, for instance, incorporate his own force interaction models or special input or output routines. User-defined subroutines are described in the MADYMO Programmer's Manual. In the next section the user-defined subroutines that allow the MADYMO user to include muscle modelling in MADYMO are described.

### 3.3 User-defined subroutines

MADYMO is written in standard FORTRAN 77. It consists of a small main program that performs calls to subroutines followed by all these subroutines. The user-defined modules are supplied as dummy routines. The user is allowed to program and link his own routines to the MADYMO package.

The purpose of this study is to achieve a better model of the human body in MADYMO. Until now the human body is modelled as a tree structure consisting of rigid bodies (representing the various body segments), which are linked by joints. A tree structure is a passive system in the sense that it only moves, when a force is applied to it. A tree structure representing a
standing human body will immediately collapse, when a gravitational acceleration field is applied to it, because no internal muscle forces can be activated to prevent this. The model of the human body in MADYMO will be more realistic, when it is possible to incorporate muscle activity in the simulation program.

As was mentioned in the previous chapter, muscle activity will only be considered at the level of musculo-skeletal modelling. Thus, only resultant torques around joints, forces caused by groups of lumped muscles or forces caused by individual muscles will be considered. This means that user-defined subroutines are needed in MADYMO that allow the user to prescribe time-dependent forces and torques to act on certain elements or around certain joints in a tree structure.

Appendix A contains all the information needed to supply the user-input data (both 2D and 3D). The input required for a force model that generates a time-dependent force acting on a certain element is described, as well as the input for a torque model, which produces a time-dependent torque that acts on certain element or around a certain joint. The user has to supply both point of application on the element on which the force is applied and the coordinate system relative to which the force components are expressed. The coordinate system can be that of inertial space, the local coordinate system of the element on which the force acts or the local coordinate system of any other element in the model.

The user has the option of prescribing a torque on an element or around a joint. In the three-dimensional case a local coordinate system has to be specified with respect to which the three torque-time histories are expressed.

Further, it is explained in Appendix B, which input should be supplied to a 2D input file in order to obtain an output file containing the angles between certain elements. This latter facility is useful, because in biomechanics one is usually interested in the joint angles, in other words, in the angles between two joint-connected elements. The notation used in the appendix is conform the MADYMO input description, which can be found in Chapter 4 of the MADYMO User’s Manuals 2D and 3D.

The source codes of the user-defined subroutines are given in Appendix C. The subroutines TIMFIN and TIMMIN are called by USERIN. TIMFIN and TIMMIN read all the data concerning the user-defined force-time histories, respectively torque-time histories. Further, the two subroutines print these data in the REPRINT file, which is generated automatically during every MADYMO simulation and which contains a self-explanatory reprint of the MADYMO input file. USERFO calls the subroutines TIMFOR and TIMMOM, which calculate the user-supplied forces and torques at the actual time of the simulation and store them in appropriate force and torque tables. These tables are used in the generation of the equations of motion. Two- and three-dimensional versions of the four routines mentioned above are available.
In MADMOMO 2D, the subroutine ANGIN2 is called by USERIN. ANGIN2 reads and prints all the pairs of elements for which the relative angles and the orientation angles are desired. ANGOU2, which is called by USEROU, writes these desired angles to the output file ANGLE.

3.4 Validation of the results

3.4.1 Simulation of a 2D simple analytical example

The translation and rotation of a uniform rod of length $l$ and mass $m$ will be considered. Its moment of inertia around the center of mass is then given by $I = ml^2/12$. A constant force $F = (F_y, F_z)$ and a constant torque $M$ are prescribed to act on the rod. $F$ is applied to the center of gravity of the rod. $F_y$ and $F_z$ are the force components in $y$ and $z$ direction in MADMOMO inertial space, where the $y$-axis is the horizontal axis and the $z$-axis is directed upward. Position, linear velocity and linear acceleration of the rod's center of mass will be denoted by $r_{cm} = (r_y, r_z)$, $v_{cm} = (v_y, v_z)$, respectively $a_{cm} = (a_y, a_z)$. Further, orientation, angular velocity and angular acceleration of the rod will be denoted by $\phi$, $\omega$ and $\alpha$. The orientation $\phi$ is the angle between the rod and the positive $y$-axis.

At $t = 0$ let the center of mass be at the origin of inertial space and let the rod be parallel to the horizontal axis. The rod is assumed to be at rest initially. This implies the following initial conditions:

\[
\begin{align*}
    r_y(0) &= 0 \\
    r_z(0) &= 0 \\
    v_y(0) &= 0 \\
    v_z(0) &= 0 \\
    \phi(0) &= 0 \\
    \omega(0) &= 0
\end{align*}
\]  

(3.4.1.1)

Using Newton's second law in linear and angular form, we have:

\[
\begin{align*}
    a_y(t) &= \frac{F_y(t)}{m} \\
    a_z(t) &= \frac{F_z(t)}{m} \\
    \alpha(t) &= \frac{M(t)}{I}
\end{align*}
\]  

(3.4.1.2)

These three initial value problems can be solved by integrating (3.4.1.2) and using (3.4.1.1). Integrating once yields the velocities $v_y$, $v_z$ and $\omega$. The next integration renders $r_y$, $r_z$ and $\phi$. Since $a_y$, $a_z$ and $\alpha$ are constants, we have:
\[ r_y(t) = a_y t^2 \]
\[ v_y(t) = a_y t \]
\[ a_y(t) = a_y \]
\[ r_z(t) = a_z t^2 \]
\[ v_z(t) = a_z t \]
\[ a_z(t) = a_z \]

\[ \phi(t) = \alpha t^2 \]
\[ \omega(t) = \alpha t \]
\[ \alpha(t) = \alpha \]

In the MADYMO simulation \( l = 0.6 \) m, \( m = 0.7 \) kg, \( F_y = F_z = 140 \) N and \( M = 6.3 \) Nm. For the moment of inertia of the rod yields:
\[ I = 0.021 \text{ kgm}^2. \] From (3.4.1.2), \( a_y = a_z = 200 \) m/s\(^2\) and \( \alpha = 300 \) rad/s\(^2\). Substituting the values for \( a_y, a_z \) and \( \alpha \) in (3.4.1.3) yields:
\[ r_y(t) = 100t^2 \]
\[ v_y(t) = 200t \]
\[ a_y(t) = 200 \]
\[ r_z(t) = 100t^2 \]
\[ v_z(t) = 200t \]
\[ a_z(t) = 200 \]

\[ \phi(t) = 150t^2 \]
\[ \omega(t) = 300t \]
\[ \alpha(t) = 300 \]

The MADYMO simulation lasts 100 milliseconds and uses a constant time step of 1 millisecond. The MADYMO input file for the simulation is shown in Appendix D.

In the REPRINT file the user input is printed in the right way, thus TIMFIN, TIMMIN and ANGIN perform satisfactorily. From the DEBUG file it can be seen that at every instant of the simulation the right force and torque values are stored in the force and torque tables. This indicates that the routines TIMFOR and TIMMOM work well.

In Figure 3.4.1.1 the motion of the rod is depicted. Its center of mass is visualized by a small circle. It is clear that the center of gravity moves along a straight line. This line makes a forty five degree angle with the horizontal axis, indicating that \( r_y \) equals \( r_z \). The inertial coordinate axes are not shown in the figure for reasons of visibility.

From equations (3.4.1.4) it can be seen that at the end of the simulation, we should have: \( r_y = r_z = 1 \) m, \( \phi = 1.5 \) rad, \( v_y = v_z = 20 \) m/s, and \( \omega = 30 \) rad/s. The figure shows that the final orientation of the rod is indeed almost \( \pi/2 \) radians. The MADYMO output files LINDIS, ANGLE, LINVEL and ANGVEL tell us that the end values for the positions, orientation, linear velocities and angular velocity in the simulation are correct (they are equal to their analytical values within single-precision accuracy).
Figure 3.4.1.1: The motion of the rod
Figure 3.4.1.2: Time-histories of position (a), velocity (b) and acceleration (c) of the rod's center of gravity.

In LINDIS, LINVEL and LINACC the time-histories of position, velocity and acceleration of the rod's center of gravity, as calculated by MADYMO, are stored. Figure 3.4.1.2 shows these time-histories. Only the graphs of the y-components are depicted, because the graphs of the z-components were identical. From ANGLE, ANGVEL and ANGACC the graphs of orientation, angular velocity and angular acceleration of the rod can be obtained as a function of time. These graphs are shown in Figure 3.4.1.3. These last two figures show that the time-histories calculated by MADYMO are equal to the time-dependent functions of (3.4.1.4), which were derived analytically.
3.4.2 Simulation of a 3D simple analytical example

In this section the rotational motion of two rigid bodies will be considered. The first rigid body consists of two homogeneous spheres of mass m and radius R, which are joined by a massless rod of length 2a. This body will be called "bar-bell" from now on. It is made to rotate at constant angular velocity about a fixed axis through the center of mass and orientated at an angle θ from the rod. The second rigid body is a homogeneous sphere also of mass m and radius R. It has the same rotational axis as the bar-bell and moves along the same circle as one of the spheres of the bar-bell. The angular velocities of both bodies are the same. This angular velocity is represented by the vector \( \omega \), which has magnitude \( \omega \). The single sphere and that of the bar-bell are on opposite sides of the midpoint of their circular orbit. Further, the sphere rotates about its axis with the same velocity \( \omega \), so that at any instant of the motion the same point on the sphere is turned toward the center of the sphere's orbit. Therefore the second rigid body will be called "moon".

This system is pictured in Figure 3.4.2.1 at an instant in its rotation, when the rod coincides with the yz-plane of inertial space. The center of the moon also lies in this plane. The inertial x-axis is directed toward the viewer. The center of gravity of the bar-bell is chosen at the origin of inertial space. (The inertial coordinate system is not shown in the figure for reasons of visibility). Note that the axis of rotation coincides with the y-axis of inertial space at every instant, and that the center of mass of the bar-bell remains at the origin of inertial space. Local coordinate systems are defined to be fixed.
to bar-bell and moon and their axes are denoted with superscripts 1 and 2, respectively. Because of the motion of the rigid bodies, these local coordinate systems are not fixed in inertial space. The origins of both coordinate systems are chosen at the center of gravity of the rigid bodies. The $z^{(1)}$-axis is chosen to coincide with the rod. Further, the $y^{(1)}$-axis is taken perpendicular to the rod in the plane determined by the rod and $\omega$. The $x^{(1)}$-axis (not shown) then extends outward toward the viewer at the instant pictured. The orientations of the axes of the local coordinate system of the moon coincide with those of inertial space in the situation of Figure 3.4.2.1. The moon moves along a circle in the $xz$-plane of inertial space and the direction of its $y^{(2)}$-axis is always equal to that of the inertial $y$-axis. Since the moon rotates around the $y^{(2)}$-axis with constant angular velocity $\omega$, its $z^{(2)}$-axis is directed towards the center of its orbit at every instant of the motion. In other words, its North pole is always closest to the orbit's center.

\begin{equation}
J = I_x \omega_x e_x^{(1)} + I_y \omega_y e_y^{(1)} + I_z \omega_z e_z^{(1)} \tag{3.4.2.1}
\end{equation}

Figure 3.4.2.1: Rotating system consisting of a sphere and a rod connecting two spheres

For a MADYMO simulation of the rotational movements described above, the external forces and torques acting on the two bodies that cause these motions need to be known.

First, consider the bar-bell. The choice of its local coordinate system is not arbitrarily. The $x^{(1)}$, $y^{(1)}$- and $z^{(1)}$-axes are the principal axes for the rigid body and thus its products of inertia vanish. Let the angular velocity vector be expressed as:

$$
\omega = \omega_x e_x^{(1)} + \omega_y e_y^{(1)} + \omega_z e_z^{(1)}
$$

where the unit coordinate vectors are along the principal axes and the values of the components of $\omega$ relate to these axes. The angular momentum vector in terms of components pertaining to these axis will be simply:
Here, $I_x$, $I_y$ and $I_z$ are the moments of inertia of the bar-bell around its principal axes. According to the equation of motion the torque vector $\mathbf{M}$, which gives rise to the changing angular momentum vector, equals the first time derivative of $J$. Expressing this derivative of $J$ the equation of motion can be written as the following set of three equations known as Euler's equations for the motion of a rigid body.

\begin{align*}
I_x \alpha_x - (I_y - I_z) \omega_y \omega_z &= M_x \\
I_y \alpha_y - (I_z - I_x) \omega_z \omega_x &= M_y \\
I_z \alpha_z - (I_x - I_y) \omega_x \omega_y &= M_z
\end{align*}

(3.4.2.2)

Here, the acceleration vector $\mathbf{a} = (\alpha_x, \alpha_y, \alpha_z)$ is the first time derivative of $\omega$. Further, the components of the torque vector $\mathbf{M}$ are expressed relative to the principal axes of the bar-bell.

In the case of the bar-bell we have:

\begin{align*}
I_x &= I_y = 4mR^2/5 + 2ma^2, \\
I_z &= 4mR^2/5 \\
\omega_x &= 0, \\
\omega_y &= \omega \sin \theta, \\
\omega_z &= \omega \cos \theta
\end{align*}

(3.4.2.3)

Then, from (3.4.2.1):

\begin{align*}
J &= (4mR^2/5 + 2ma^2) \omega \sin \theta e_y^{(1)} + 4mR^2/5 \omega \cos \theta e_z^{(1)}
\end{align*}

The angular momentum vector lies in the plane determined by the rod and the angular velocity vector $\omega$, which lies along the inertial $y$-axis.

Since $\omega$ is considered to be constant, from Euler's equations we have:

\begin{align*}
M_x &= (I_z - I_y) \omega_y \omega_z = -2ma^2 \omega^2 \sin \theta \cos \theta, \\
M_y &= M_z = 0
\end{align*}

Thus, the torque vector $\mathbf{M}$ that causes the desired motion only has a component along the $x^{(1)}$-axis, and it rotates with the bar-bell. It is not hard to see the reason for this, if one recognizes the centripetal forces $ma^2 \omega^2 \sin \theta \cos \theta$ required to hold the two spheres in their circular motion around the inertial $y$-axis. The sum of these two centripetal forces times the distance $2a \cos \theta$ between them make up the torque $\mathbf{M}$. (Note, that the bar-bell does not rotate around its longitudinal axis).

The rotational movement of the single sphere is a much simpler case. A centripetal force $F$ applied to its center of gravity and acting along the positive $z^{(2)}$-axis with magnitude $ma^2 \omega^2 \sin \theta \cos \theta$ will hold the moon in its circular orbit. An initial velocity $\omega$ of the moon around the $y^{(2)}$-axis will be maintained throughout the motion, because no other forces or torques are prescribed to act on this rigid body.

In the MADYMO simulation the following constants were chosen: $a = 0.4 \text{ m}$, $R = 0.05 \text{ m}$, $m = 0.15 \text{ kg}$, $\theta = \pi/6 \text{ rad}$ and $\omega = 20\pi \text{ rad/s}$. The $x^{(1)}$-component of the torque that has to be
applied to the bar-bell yields $M_z = -328.2174$ Nm. The $z^{(2)}$-component of the centripetal force that should be applied to the moon's center of gravity yields $F_z = 118.43525$ N. Values of $I_x$, $I_y$, $I_z$, $\omega_x$, $\omega_y$ and $\omega_z$ for the bar-bell can be calculated from (3.4.2.3). Since the end time of the simulation is taken to be $0.1$ s, the system should go through one rotation and its end position should be the same as its initial position. The constant time step for the fourth-order Runge-Kutta integration method is taken to be 1 millisecond. The MADYMO input file for the simulation can be found in Appendix D.

From the MADYMO output files it can be seen, whether the results of the simulation agree with the theoretical expectations. From the REPRINT file it was seen that TIMFIN and TIMMIN handle the user-defined input correctly. The file DEBUG shows that at every instant of the simulation the correct torque (applied to the barbell) is stored in the torque table. The same is true for the force, which acts on the center of mass of the moon. This is an indication that TIMFOR and TIMMOM perform well.

From ANGVEL it was seen that the angular velocity of the barbell has $x$- and $z$-components with respect to inertial space, which are of the order $10^{-5}$ rad/s. The $y$-component is very close to $20\pi$.

Since the angular velocity is constant, the angular acceleration of the rod should be zero. ANGACC shows that the calculated acceleration components are of the order of $10^{-3}$ rad/s$^2$.

The discrepancies between the actual and theoretical values of velocities and accelerations are due to round-off errors. (MADYMO uses single precision arithmetic in its calculations).

To be able to compare the kinematics results of the MADYMO simulation with analytical results, a closer look is taken at the circular motion of the moon and the sphere at the right end of the bar-bell in Figure 3.4.2.1. Analytical expressions for the positions, linear velocity and linear acceleration of the centers of both spheres will be derived. The MADYMO output files LINDIS, LINVEL and LINACC contain calculated time-histories of the three variables mentioned above.

In Figure 3.4.2.2 the circle that is the projection of the orbit of both spheres on the $xz$-plane is shown. Note, that the viewer looks in the direction of the negative $y$-axis. Consider a point $P$ moving at constant angular velocity $\omega$ along a circle with radius $r$ (here $r = a\sin\theta$). Let $r(t) = (r_x(t), r_z(t))$, $v(t) = (v_x(t), v_z(t))$ and $a(t) = (a_x(t), a_z(t))$ denote position, linear velocity, respectively linear acceleration of point $P$. Further, let $\phi(t)$ be the angle between $r(t)$ and the positive $z$-axis. We have $\phi(t) = \phi_0 + \omega t$, where $\phi_0$ is the initial value of $\phi$. For the bar-bell $\phi_0 = 0$ and for the moon $\phi_0 = \pi$. Furthermore,

\[ r_x(t) = r\sin\phi(t) \]
\[ r_z(t) = r\cos\phi(t) \]
Differentiation yields:

\[ v_x(t) = \omega r \cos \phi(t) \]
\[ v_z(t) = -\omega r \sin \phi(t) \]
\[ a_x(t) = -\omega^2 r \sin \phi(t) \]
\[ a_z(t) = -\omega^2 r \cos \phi(t) \]

It can be seen that \( a = -\omega^2 r \) and \( v \) is perpendicular to both \( r \) and \( a \). Moreover, the resultant velocity \( v \) is equal to \( \omega r \) and the resultant acceleration equals \( \omega^2 r \). These are all well known results from elementary physics.

![Figure 3.4.2.2: Particle moving along a sphere](image)

In the case of the sphere that is part of the barbell \( \phi(t) = \omega t \). We can substitute this relation in the above expressions for \( r_x, r_z, v_x, v_z, a_x \) and \( a_z \). In the other case \( \phi(t) = \pi + \omega t \). Position, velocity and acceleration of the moon are opposite to those of the other sphere, since \( \sin(\pi + \omega t) = -\sin \omega t \) and \( \cos(\pi + \omega t) = -\cos \omega t \).

Summarizing, for the sphere that is attached to the rod the following relationships yield:

\[ r_x(t) = r \sin \omega t \]
\[ r_z(t) = r \cos \omega t \]
\[ v_x(t) = \omega r \cos \omega t \]
\[ v_z(t) = -\omega r \sin \omega t \]
\[ a_x(t) = -\omega^2 r \sin \omega t \]
\[ a_z(t) = -\omega^2 r \cos \omega t \]
In the case of the moon every right hand side in the above equations should be replaced by its opposite value.

As was mentioned above, the amplitudes of displacement, velocity and acceleration of the centers of both spheres are \( r \), \( \omega r \) and \( \omega^2 r \). For the simulation considered here yields: \( r = \cos \theta = 0.2 \, \text{m} \), \( \omega r = 4\pi (\approx 12.57) \, \text{m/s} \) and \( \omega^2 r = 80\pi^2 (\approx 789.6) \, \text{m/s}^2 \). The y-component of the position vector of the the centers of the two spheres is equal to \( \cos \theta \approx 0.3464 \).

![Graphs showing time-histories of displacement components of sphere center](image)

**Figure 3.4.2.3**: Time-histories of the displacement components of the center of the sphere that is attached to the rod.

From LINDIS it was seen that the center of mass of the bar-bell remains indeed at the origin of inertial space. LINDIS, LINVEL
and LINACC show that the values of the x- and z-components of the computed displacements, velocities and accelerations of the center of the sphere at the end of the rod are opposite to those of the center of the moon. Moreover, the values of all variables at the end of the simulation are equal to their initial values. Figures 3.4.2.3, 3.4.2.4 and 3.4.2.5 show time-histories of position, velocity and acceleration of the center of the sphere that is attached to the rod. Similar graphs for the moon are not shown. It can be seen that the graphs in the three figures agree with the analytical functions of equations (3.4.2.4).

\[ \begin{align*}
\text{Figure 3.4.2.4} & : \text{Time-histories of the velocity components of the center of the sphere that is attached to the rod.}
\end{align*} \]
Figure 3.4.2.5: Time-histories of the acceleration components of the center of the sphere that is attached to the rod.

The results calculated in the MADYMO 3D simulation are equal (within numerical accuracy) to analytical results. Thus, it can be concluded that the user-defined subroutines that were developed to prescribe forces and torques perform well.

No figures of the actual motion were shown here, because it is hard to visualize three-dimensional motion in two-dimensional space.
3.4.3 Simulation of a subject doing a push-up

In this section a different way to prescribe torque around a joint is described. The discussion will be restricted to the two-dimensional case only. Generalization to three dimensions is straightforward. The user-defined subroutines allow the MADYMO user to define torques, that act on a joint or on an element, as a function of time. So in simulating human movement one has to know the torque-time characteristics beforehand. In most studies of human movement however, only kinematic data (joint angles, etc.) are available.

It will be shown that MADYMO can be used for the simulation of a motion of which only the kinematics are known. By introducing a torque generator with linear feedback (conform Van den Bogert, 1988) a desired joint angle and angular velocity can be used to create a control torque that causes the actual joint angle and angular velocity to approach their desired values. In other words, this torque generator allows us to control joint movement. A kinematical variable is prescribed in order to achieve a kinetic variable, namely the torque that is necessary to cause a certain desired joint angle and angular velocity.

The torque generator suggested by Van den Bogert yields:

\[ M(\phi, \omega, t) = - A(\phi(t) - \phi_d(t)) - B(\omega(t) - \omega_d(t)) \]  

(3.4.3.1)

where \( M \) is the total (muscle) torque, \( \phi \) is the actual joint angle, \( \phi_d \) is the desired joint angle that acts as a control function, \( \omega \) and \( \omega_d \) are the first time derivatives of \( \phi \) and \( \phi_d \), and \( A \) and \( B \) are the feedback parameters. At time \( t \) the first (elastic) term produces a torque that drives \( \phi \) towards \( \phi_d \), while the second (damping) term decreases (increases) the torque, if the angle \( \phi \) is moving towards (away from) \( \phi_d \).

The model is equivalent to a three-element spring-damper-actuator as can be seen after rearranging the terms:

\[ M(\phi, \omega, t) = - A\phi(t) + A\phi_d(t) - B\omega(t) + B\omega_d(t) \]

\[ = - A\phi(t) - B\omega(t) + C(t) \]  

(3.4.3.2)

The first two terms on the right can be modelled in standard MADYMO (elastic and damping torques in a joint). The third term can be modelled using the user-defined subroutines (time-dependent torque acting around a joint).

Being able to incorporate the above model in MADYMO allows the user to carry out simulations of human movement of which only kinematic data are known. Certain desired joint angles and angular velocities can be prescribed by the user. The model produces a joint torque that drives joint angle and angular velocity to their desired values. This will be illustrated with an example of a MADYMO simulation of human movement of which no measured kinematics were available. The kinematics were chosen by the user.
Figure 3.4.3.1 : Model of a subject doing a push-up

Consider a subject doing a push-up in the sagittal plane. The subject is modelled as a system of five rigid bodies connected by four hinge type joints (see Figure 3.4.3.1). The five elements represent the feet (1), head-neck-trunk-legs (2), upper arms (3), lower arms (4) and hands (5). Wrist and ankle joints are considered fixed in space, while shoulder and elbow joints are free to move. The hands and feet rest on the floor and do not move. The other three elements form a chain with fixed ends. This chain has one degree of freedom. If, for instance, the elbow joint angle is known, the position of the chain is determined. This means that the position of the subject can be controlled by controlling the elbow angle. The torque generator will be used to control the elbow joint.

In this two-dimensional model feet, upper and lower arms, hands, ankles, shoulders, elbows and wrists are modelled by one element or by one joint. The elements will subsequently be denoted by their plural form (e.g. feet), while a singular form will be used for the joints (e.g. ankle, elbow joint).

The desired elbow joint angle \( \phi_d(t) \) during the push-up is chosen to be a cubic polynomial function that satisfies certain initial and end conditions. The subject initially is at rest with an elbow angle of 30 degrees. The motion lasts one second and at the end the subject is again supposed to be at rest, this time with the arms completely stretched, thus requiring a final \( \phi_d \) value of 180 degrees. The desired elbow angle and angular velocity then yield:

\[
\phi_d(t) = -5nt^3/3 + 5nt^2/2 + \pi/6 \\
\omega_d(t) = -5nt^2 + 5nt
\]

At the end the shoulder joint is supposed to be located straight above the fixed wrist. Given the length of the body segments, the desired position of the subject is now determined. Before the MADYMO simulation the initial angles in the ankle, shoulder and wrist joints have to be calculated, because the initial orientation of every element has to be specified in the MADYMO input file. The inertia and geometry of the elements are roughly estimated.
PUSH-UP

Time 0.00

Time 100.00

Time 200.00

Time 300.00

Time 400.00

Time 500.00

Time 600.00

Time 700.00
The wrist and the ankle are kept fixed in space by giving the feet and the hands a mass of one billion kilograms. The forces exerted on the heavy elements by the other elements in the tree structure will be too small to move the heavy elements. By taking their center of gravity at the joint no torques will act on the feet and the hands. After a process of trial and error $A = 900 \text{ Nm rad}^{-1}$ and $B = 35 \text{ Nm rad}^{-1}$ gave the best results for the computed elbow angle. A timestep of 1 millisecond was used in the simulation. The MADYMO input file is shown in Appendix D.

Figure 3.4.3.2 shows the position of the subject at different time points, and in Figure 3.4.3.3 the computed elbow angle $\phi$ and the desired joint angle $\phi_d$ are shown as a function of time. Initially the shoulder moves downward and the elbow joint angle decreases. This is accounted for by the fact that the initial values for elbow angle and angular velocity were chosen equal to their desired values. This means that initially the torque generated at the elbow joint is zero (see equation (3.4.3.1)). Gravity acts on the body and causes the subject to flex his arms rather than extend them. In other words, $\phi$ and $\omega$ become less than $\phi_d$ and $\omega_d$. Therefore the control torque $M$ will be positive after the first step of the simulation and then the control of the elbow joint angle starts. After approximately 200 milliseconds
\( \phi \) starts to increase and at \( t = 1 \) s the subject has completed the push-up. The arms are indeed fully extended (\( \phi = \pi \)).

![Diagram](image.png)

**Figure 3.4.3.3**: Time-histories of the desired and actual elbow angles

The motion of the subject can be illustrated by the following situation. Suppose that the subject is supported in such manner that he is in the initial push-up position, while relaxing his arms. Thus, at \( t = 0 \) no muscles are active in the arms. The subject is told to do a push-up as soon as the support is removed. When the support is taken away suddenly, the subject will initially flex his arms, but then he will react, tense his arm muscles and do the push-up.
A literature study on the modelling of muscle activity in the human body was conducted. Also experts in the field of muscle modelling were consulted. The purpose of this was to gain insight in the various levels at which muscle activity can be modelled. This way, a sensible decision could be made concerning the incorporation of muscle activity in the computer simulation program MADYMO. So far, the human body had been modelled as a passive tree structure of joint-connected rigid bodies, which moves only as a consequence of applied external forces. The integration of muscular behavior allows the tree structure to move "all by itself".

It was decided to extend MADYMO in such manner, that it would be possible to define muscular forces and torques to act on certain elements or joints of the human body model. Muscle activity is considered at the level of forces and torques that are generated by the muscle. In other words, the improved human body model is a musculo-skeletal model. Micro- and macroscopic muscle models, which describe the relation between muscle properties and produced force, were not taken into account.

The extension of MADYMO (both 2D and 3D) consists of user-defined subroutines, which allow the user to prescribe time-dependent forces and torques. These loads can be exerted on elements or joints of a tree structure that represents the human body. For each force the point of application has to be specified. Further, the user must specify the coordinate system relative to which the components of each torque (only 3D) and force are expressed.

The user-defined subroutines were tested by means of a 2D and 3D simulation of an analytically known motion. It was shown that the numerical and analytical results agreed, indicating that the routines performed satisfactory. Moreover, the routines were validated by the simulation of a subject doing a push-up. Desired joint movement was achieved by means of a control torque at the joint. This control torque requires the prescription of a time-dependent torque in MADYMO, which is possible thanks to the user-defined subroutines. A further validation of the routines is given by the simulations, which are described in Chapter 3 of part II of the report.

This report on muscle modelling will be concluded with some suggestions for future research.

The user-defined subroutines perform well in MADYMO 2D simulations of human body movement in which muscle forces and torques occur. The extended MADYMO 3D program has only been tested for a simple analytically known motion.

In the simulation of the push-up a feedback control strategy was used to generate a torque at the elbow joint that caused the
motion of the human body model. The control torque consists of an unknown torque $A\dot{\phi} - B\dot{\omega}$ and a user-supplied torque $A\dot{\phi}_d + B\dot{\omega}_d$. The first term depends on the actual joint motion as computed by MADYMO. At the end of a simulation the user wants to know the value of control torque. At the moment the control torque has to be computed by the user himself. The unknown torque can be obtained from the MADYMO output file DEBUG and has to be added to the user-supplied torque. In the future the torque should be rendered by a user-defined subroutine, which is called by USEROU. So far, muscle activity is only considered at the level of torques and forces caused by individual muscles or groups of muscles. A next step in achieving more realistic muscle modelling is to incorporate macroscopic muscle models in MADYMO. At the moment the user has to supply muscle force and resulting torque as a function of time. In the future, Hill-type muscle models can be included in the MADYMO human body model. Given the active state of the muscle, which has to be supplied by the user, the force that is produced by the muscle must be computed by user-defined subroutines. These routines must be based on force-length and force-velocity characteristics of the muscle. The length and contraction velocity of the muscle can be calculated from the relative positions and angular velocities of the segments that affect the path of the muscle. In the human body many types of muscles exist. Each individual muscle has different properties. Attention should be paid to muscle parameter estimation and to using the appropriate muscle characteristics in a macroscopic model. For parameters needed in a Hill-type model see Brand et al. (1982 and 1986) and Winters and Stark (1988). Microscopic muscle models go into even more detail. They allow for the input of electrical pulses provided by the central nervous system. The micro model (possibly integrated in a macro model) then renders the force that is developed by the muscle due to muscular contraction caused by the neural pulses. At the moment, mathematical modelling of the central nervous system is a hot item in the field of neuroscience. Should reliable models of the central nervous system ever be established, then (theoretically) these models could be connected to muscle models, thus allowing for muscular control. It is expected however, that a long way has to be gone before this goal will be achieved. It should be noted that for many types of problems such complicated models are not required.
REFERENCES

Mathematical modelling and simulation of the postural control loop: part I  
Critical Reviews in Biomedical Engineering, 8, 93-134

Alberts, L.K. (1985)  
Modelvorming en simulatie van een 2D spier-skelet systeem  
Verslag 65-middagen opdracht, Department of Electrical Engineering, University of Twente

Biomechanical considerations in the modelling of muscle function  

A general method for determining the functional role of a muscle  
Transactions of the ASME, 107, 348-353

A three-dimensional kinematic and dynamic model of the lower limb  
J. Biomechanics, 22, 27-50

A new approach to the human muscle model  
J. Biomechanics, 16, 803-809

Bernards, J.A. and Bouman, L.N. (1983)  
Fysiologie van de mens  
Utrecht/Antwerpen: Bohn, Sheltema and Holkema, ISBN 90-313-0590-1

Simulation of quadrupedal locomotion using a rigid body model  
J. Biomechanics, 22, 33-41

A model of lower extremity muscular anatomy  
J. of Biomechanical Engineering, 104, 304-310

The sensitivity of muscle force predictions to changes in physiological cross-sectional area  
J. Biomechanics, 19, 589-596

A dynamic model for finger interphalangcal coordination  
J. Biomechanics, 21, 459-468

Cappozzo, A., Leo, T. and Pedotti, A. (1975)  
A general computing method for the analysis of human locomotion  
J. Biomechanics, 8, 307-320

The interplay of muscular and external forces in human ambulation
J. Biomechanics, 9, 35-43

Gait analysis methodology
Human Movement Science, 3, 27-50

Dynamic modeling and implementation of a right turn in walking
J. Biomechanics, 19, 195-206

De EMG-spierkracht relatie
Nederlands Tijdschrift voor Fysiotherapie, 94, 145-148

Muscular Synergism - I. On criteria for load sharing between synergistic muscles
J. Biomechanics, 17, 663-673

Muscular Synergism - II. A minimum-fatigue criterion for load sharing between synergistic muscles
J. Biomechanics, 17, 675-684

A kinematic model of the human ankle
J. Biomedical Engineering, 7, 137-143

Dul, J. and Shiavi, R. (1985)
Simulation of tendon transfer surgery
Engineering in Medicine, 14, 31-38

Dul, J. (1985)
Muscular Coordination in Working Postures

Dul, J. (1986)
The biomechanical prediction of muscle forces
Clinical Biomechanics, 1, 27-30

Dul, J. (1988)
A biomechanical model to quantify shoulder load at the work place
Clinical Biomechanics, 3, 124-128

FitzHugh, R. (1977)
A model of optimal voluntary muscular control
J. Mathematical Biology, 4, 203-236

Harlaar, J. (1985)
Bepaling van enkele mechanische eigenschappen van de kuitmusculatuur, alsmede 3D modelvorming
Master's thesis, Department of Electrical Engineering, University of Twente

Hatze, H. (1977)
A complete set of control equations for the human musculo-skeletal system
J. Biomechanics, 10, 799-805

Hemami, H. (1985)
Modeling, control and simulation of human movement
Critical Reviews in Biomedical Engineering, 13, 1-34

Hill, A.V. (1938)
The heat of shortening and the dynamic constants of muscle
Proceedings of the Royal Society, B126, 136-195

De mechanica van het dartswerpen
Master's thesis, Department of Mathematical Engineering, University of Groningen

Some aspects of the use of MADYMO in general biomechanics analysis. Part I: Organization
Final report, postgraduate program "Mathematics in Industry", Eindhoven University of Technology

Some aspects of the use of MADYMO in general biomechanics analysis. Part II: Technical realization
Final report, postgraduate program "Mathematics in Industry", Eindhoven University of Technology

An electrical analogue of the Hill muscle model EMG to force processing
Ph.D. thesis, University of Groningen

Calf muscle moment, work and efficiency in level walking: role of series elasticity
J. of Biomechanics, 16, 523-537

EMG and muscle force: an introduction
Human Movement Science, 3, 119-153

Biomechanica, Chapter 2: Spiermechanica (p 29-48), editor Huiskes Samson-Stafleu, Alphen a/d Rijn

Huxley, A.F. (1957)
Muscle structure and theories of contraction
Progr. Biophysics, 7, 255-318

Simulation of the double limb support phase of human gait
J. Biomechanical Engineering, 110, 223-229

Paraplegic standing controlled by functional neuromuscular stimulation: computer model, control-system design and simulation studies

The three-dimensional analysis and prediction of human walking
Ph.D. thesis, University of Twente

Report TNO Road-Vehicles Research Institute

Report TNO Road-Vehicles Research Institute

Analysis and syntheses of human swing leg motion during gait and its clinical applications
J. Biomechanics, 14, 823-832

A mathematical model for the dynamics of human locomotion
J. Biomechanics, 13, 361-368

Otten, E. (1987)
A myocybernetic model of the jaw system of the rat
J. of Neuroscience Methods, 21, 287-302

Otten, E. (1988)
Concepts and models of functional architecture in skeletal muscle
Exercise and sport Sciences Reviews, 16, 89-137

An experimental and analytical study of impact forces during human jumping
J. Biomechanics, 21,1061-1066

Pandy, M.G. and Berme, N. (1988)
A numerical method for simulating the dynamics of human walking
J. Biomechanics, 21, 1043-1051

Pandy, M.G. and Berme, N. (1988)
Synthesis of human walking: a planar model for single support
J. Biomechanics, 21, 1053-1060

Relationship between the surface EMG and muscular force
Medical & Biological Engineering & Computing, 21, 579-568

Seireg, A. and Arvikar, R.J. (1975)
The prediction of muscular load sharing and joint forces in the lower extremities during walking
J. Biomechanics, 8, 89-102

Sol, E.J. (1980)
Literatuurverslag over spier-skelet-modellen,
Report, Eindhoven University of Technology

Townsend, M.A. and Seireg, A. (1972)
The synthesis of bipedal locomotion
J. Biomechanics, 5, 71-83

TNOPOST: A postprocessor for MADYMO 3D
Report, Department of Computer Science,
Eindhoven University of Technology

Vermazeren, R.C.J.G. (in preparation)
Title unknown (subject: a preprocessor for MADYMO 3D)
Master's thesis, Department of Computer Science,
Eindhoven University of Technology

Two contributions to the modelling of the human body in MADYMO
Master's thesis, Department of Mathematical Engineering,
Eindhoven University of Technology

Joint torque and energy patterns in normal gait
Biological Cybernetics, 29, 137-142

Analysis of fundamental human movement patterns through the use
of in-depth antagonistic muscle models
IEEE Transactions on Biomedical Engineering, 32, 826-839

Estimated mechanical properties of synergistic muscles involved
in movements of a variety of human joints
J. Biomechanics, 21, 12, 1027-1041

Wittenburg, J. (1977)
Dynamics of systems of rigid bodies
BG Teubner, Stuttgart

Muscle dynamics, size principle and stability
IEEE Transactions on Biomedical Engineering, 31, 489-497
APPENDIX A

MUSCLE MODELLING ACTIVITIES IN THE NETHERLANDS

This is an overview of a number of persons in the Netherlands, who are working in the field of muscle modelling.

A.L. Hof investigates the application of the muscle model of his Ph.D. thesis (Hof, 1980) in human walking and running. He works at the Department of Medical Physics of the University of Groningen.

E. Otten, who is working for the Orofacial Research Group at the University of Groningen, studies the coupling of muscle models with the central nervous system. The modelling of the central nervous system is a hot topic in the field of neuroscience. Researchers are especially interested in the functionality of the central nervous system with respect to control of muscle activity.

J. Dul is working in the field of ergonomics of working postures at the Netherlands Institute for Preventive Health Care-TNO in Leiden. This institute has facilities for three-dimensional motion analysis. Musculo-skeletal models are used to predict muscle fatigue in (mostly) static working postures.

At the Faculty of Human Movement Science of the Free University of Amsterdam a mechanical device (called Jumping Jack) is used to simulate human jumping and walking. A.J. van Soest and G.J. van Ingen Schenau are developing a musculo-skeletal model of the lower extremities in which groups of synergetic muscles are represented by a Hill-type muscle model. The mathematical model will be used to simulate movements of the physical model. The "muscle" parameters for the physical model can be obtained easier and more accurate than parameters human body muscles.

Several persons from different faculties at the University of Twente carry out research on the simulation of human movement. At the Electrical Engineering Department research is carried out on functional neuromuscular stimulation. At the Mechanical Engineering Department the research is focussed on the modelling of human walking.

At the Department of Veterinary Anatomy of the University of Utrecht muscle activity in animals is modelled. An example is the simulation of the locomotion of a horse by A.J. Van den Bogert.
APPENDIX B

USER'S GUIDE TO THE SUBROUTINES TIMFIN, TIMMIN AND ANGIN2

In MADYMO there are three user-defined subroutines: USERIN, USERFO and USEROU. USERIN calls all routines that read user-defined input. The input data must be placed at the end of the MADYMO input file DATA.DAT after the keyword "USER INPUT", but before the keyword "END INPUT" (see MADYMO User's Manuals 2D and 3D, 1988, Chapter 4). The order in which the keywords in the user-defined input are given can be chosen arbitrarily by the user. User-defined joint, force and torque models can be placed in separate subroutines, which are called by the subroutine USERFO. The routine USEROU calls all user-defined output modules.

In this appendix the input that has to be supplied by the user for each subroutine is described. The description is conform the lay-out of the MADYMO Programmer’s Manual (1988). For TIMFIN and TIMMIN a distinction is made between the 2D and the 3D case, while ANGIN2 is written for the two-dimensional case only.

TIMFIN and TIMMIN read all data concerning the time-dependent force, respectively torque, that the user prescribes. ANGIN2 reads the pairs of elements for which the orientation angles and the relative angle will be written to the output file ANGLE. All these subroutines print the data that they have read at the end of the MADYMO output file REPRINT. The subroutines are called by the MADYMO routine USERIN.

B1. The subroutine TIMFIN (2D)

- Ext for (keyword)
- NO
- SYS1 EL1 Y Z SYS2 EL2 FY FZ TEXT

(This data record must be repeated NO times. If NO = 0 this data block and the remaining part must be omitted.)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Units</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>Integer</td>
<td></td>
<td>Number of applied forces</td>
</tr>
<tr>
<td>Variable</td>
<td>Type</td>
<td>Units</td>
<td>Definition</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>-------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SYS1</td>
<td>Integer</td>
<td></td>
<td>System number on which force is applied</td>
</tr>
<tr>
<td>EL1</td>
<td>Integer</td>
<td></td>
<td>Element number on which force is applied</td>
</tr>
<tr>
<td>Y</td>
<td>Real</td>
<td>m</td>
<td>Coordinates point of application, expressed in the local coordinate system of element EL1 of system SYS1</td>
</tr>
<tr>
<td>Z</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYS2</td>
<td>Integer</td>
<td></td>
<td>FY and FZ are the function numbers for force-time histories in Y and Z direction of the local coordinate system of element EL2 of system SYS2 in the function table to be defined here after. A negative value for FY (FZ) means that spline approximation is used for the force-time history in Y (Z) direction (note 1)</td>
</tr>
<tr>
<td>EL2</td>
<td>Integer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FY</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FZ</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TEXT</td>
<td>Character</td>
<td></td>
<td>Identifier (max. char. = 15) (note 2)</td>
</tr>
</tbody>
</table>

NFUN

N Ti F1 ........... TN FN

(This data record must be repeated NFUN times. If NFUN = 0 this data block must be omitted.)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Units</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NFUN</td>
<td>Integer</td>
<td></td>
<td>Number of functions defined</td>
</tr>
<tr>
<td>N</td>
<td>Integer</td>
<td></td>
<td>Number of coordinate pairs defining the considered function</td>
</tr>
<tr>
<td>Ti</td>
<td>Real</td>
<td>s</td>
<td>Time points at which the considered function is specified</td>
</tr>
<tr>
<td>(i=1,N)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fi</td>
<td>Real</td>
<td>N</td>
<td>Force at time Ti of the considered function</td>
</tr>
<tr>
<td>(i=1,N)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
NOTES:

1. If the force-time histories are given relative to inertial space: SYS2 = -1. If the force-time histories are given relative to null system EL2: SYS2 = 0 and EL2 > 0.

2. The character string should be placed between apostrophes according to the standard FORTRAN specifications.

B2. The subroutine TIMMIN (2D)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Units</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPAR</td>
<td>Integer</td>
<td>-</td>
<td>Selection parameter</td>
</tr>
<tr>
<td>NO</td>
<td>Integer</td>
<td>-</td>
<td>Number of applied torques</td>
</tr>
<tr>
<td>SYS</td>
<td>Integer</td>
<td>-</td>
<td>System number on which torque is applied</td>
</tr>
<tr>
<td>EL</td>
<td>Integer</td>
<td>-</td>
<td>Element number on which torque is applied (IPAR = 1) Joint number on which torque is applied (IPAR = 2)</td>
</tr>
<tr>
<td>MOM</td>
<td>Integer</td>
<td>-</td>
<td>Function number for torque-time history in the function table to be defined here after. A negative value for MOM means that spline approximation is used for the torque-time history (note 1)</td>
</tr>
</tbody>
</table>
This data record must be repeated NFUN times. If NFUN = 0 this data block must be omitted.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Units</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NFUN</td>
<td>Integer</td>
<td>-</td>
<td>Number of functions defined</td>
</tr>
<tr>
<td>N</td>
<td>Integer</td>
<td>-</td>
<td>Number of coordinate pairs defining the considered function</td>
</tr>
<tr>
<td>Ti</td>
<td>Real</td>
<td>s</td>
<td>Time points at which the considered function is specified</td>
</tr>
<tr>
<td>(i=1,N)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mi</td>
<td>Real</td>
<td>Nm</td>
<td>Torque at time Ti of the considered function</td>
</tr>
<tr>
<td>(i=1,N)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

NOTES:

1. If IPAR = 1, the torque specified by MOM will be applied on element EL. If IPAR = 2, the torque specified by MOM will be applied on element EL, while the negative value of this torque will be applied on the preceeding element NPREC(EL,SYS). (Note, that joint EL (EL > 1) connects the two elements EL and NPREC(EL,SYS) in system SYS). A torque with a positive value acting on an element will cause the element to rotate in a counter clockwise direction.

2. The character string should be placed between apostrophes according to the standard FORTRAN specifications.
B3. The subroutine ANGIN2 (2D)

ANGLES (keyword)

SYS1 EL1 SYS2 EL2

(This data record must be repeated for each pair of system -
element combinations for which angles are calculated; see notes
1 and 2).

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Units</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYS1</td>
<td>Integer</td>
<td>-</td>
<td>Number of the first system</td>
</tr>
<tr>
<td>EL1</td>
<td>Integer</td>
<td>-</td>
<td>Number of the element in system SYS1 or number of the null system if SYS1 = 0</td>
</tr>
<tr>
<td>SYS2</td>
<td>Integer</td>
<td>-</td>
<td>Number of the second system</td>
</tr>
<tr>
<td>EL2</td>
<td>Integer</td>
<td>-</td>
<td>Number of the element in system SYS2 or number of the null system if SYS2 = 0</td>
</tr>
</tbody>
</table>

NOTES:

1. The angle of the system-element combinations SYS1-EL1 and
SYS2-EL2 relative to the inertial coordinate system and the
angle of the system-element combination SYS1-EL1 relative to
SYS2-EL2 will be calculated.

2. The total number of system-element combinations should not
exceed QNPOUT (see Appendix A of the MADYMO Programmer’s
Manual).
B4. The subroutine TIMFIN (3D)

EXTFOR (keyword)

NO

SYS1 EL1 X Y Z SYS2 EL2 FX FY FZ TEXT

(This data record must be repeated NO times. If NO = 0 this data block and the remaining part must be omitted.)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Units</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>Integer</td>
<td>-</td>
<td>Number of applied forces</td>
</tr>
<tr>
<td>SYS1</td>
<td>Integer</td>
<td>-</td>
<td>System number on which force is applied</td>
</tr>
<tr>
<td>EL1</td>
<td>Integer</td>
<td>-</td>
<td>Element number on which force is applied</td>
</tr>
<tr>
<td>X</td>
<td>Real</td>
<td>m</td>
<td>Coordinates point of application, expressed in the local coordinate system of element EL1 of system SYS1</td>
</tr>
<tr>
<td>Y</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYS2</td>
<td>Integer</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>EL2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FX</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FY</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FZ</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TEXT</td>
<td>Character</td>
<td>-</td>
<td>Identifier (max. char. = 15) (note 2)</td>
</tr>
</tbody>
</table>

NFUN
N T1 F1 ........... TN FN

(This data record must be repeated NFUN times. If NFUN = 0 this data block must be omitted.)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Units</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NFUN</td>
<td>Integer</td>
<td>-</td>
<td>Number of functions defined</td>
</tr>
<tr>
<td>N</td>
<td>Integer</td>
<td>-</td>
<td>Number of coordinate pairs defining the considered function</td>
</tr>
<tr>
<td>Ti</td>
<td>Real</td>
<td>s</td>
<td>Time points at which the considered function is specified</td>
</tr>
<tr>
<td>(i=1,N)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fi</td>
<td>Real</td>
<td>N</td>
<td>Force at time Ti of the considered function</td>
</tr>
<tr>
<td>(i=1,N)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

NOTES:

1. If the force-time histories are given relative to inertial space: SYS2 = -1. If the force-time histories are given relative to null system EL2: SYS2 = 0 and EL2 > 0.

2. The character string should be placed between apostrophes according to the standard FORTRAN specifications.

B5. The subroutine TIMMIN (3D)

<table>
<thead>
<tr>
<th>EXTMOM (keyword)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>NO</th>
</tr>
</thead>
</table>

| IPAR SYS1 EL1 SYS2 EL2 FX FY FZ TEXT |

(This data record must be repeated NO times. If NO = 0 this data block and the remaining part must be omitted.)
### Variable Types and Definitions

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Units</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPAR</td>
<td>Integer</td>
<td>-</td>
<td>Selection parameter</td>
</tr>
<tr>
<td>NO</td>
<td>Integer</td>
<td>-</td>
<td>Number of applied torques</td>
</tr>
<tr>
<td>SYS</td>
<td>Integer</td>
<td>-</td>
<td>System number on which torque is applied</td>
</tr>
<tr>
<td>EL</td>
<td>Integer</td>
<td>-</td>
<td>Element number on which torque is applied (IPAR = 1) Joint number on which torque is applied (IPAR = 2)</td>
</tr>
<tr>
<td>SYS2</td>
<td>Integer</td>
<td>-</td>
<td>FZ, FY and FZ are the function numbers for torque-time histories in X, Y and Z direction of the local coordinate system of element EL2 of system SYS2 in the function table to be defined here after. A negative value for FX (FY) (FZ) means that spline approximation is used for the torque-time history in X (Y) (Z) direction (notes 1 and 2)</td>
</tr>
<tr>
<td>EL2</td>
<td>Integer</td>
<td>-</td>
<td>TEXT Character</td>
</tr>
</tbody>
</table>

### NFUN

N T1 M1 ........... TN MN

(This data record must be repeated NFUN times. If NFUN = 0 this data block must be omitted.)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Units</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NFUN</td>
<td>Integer</td>
<td>-</td>
<td>Number of functions defined</td>
</tr>
<tr>
<td>N</td>
<td>Integer</td>
<td>-</td>
<td>Number of coordinate pairs defining the considered function</td>
</tr>
<tr>
<td>Ti</td>
<td>Real</td>
<td>s</td>
<td>Time points at which the considered function is specified</td>
</tr>
</tbody>
</table>

(i-1,N)
NOTES:

1. If the torque-time histories are given relative to inertial space: SYS2 = -1. If the torque-time histories are given relative to null system EL2: SYS2 = 0 and EL2 > 0.

2. If IPAR = 1, the torque specified by FX, FY and FZ will be applied on element EL1. If IPAR = 2, the torque specified by FX, FY and FZ will be applied on element EL1, while the negative value of this torque will be applied on the preceding element NPREC(EL1, SYS1). (Note, that joint EL1 (EL1 > 1) connects the two elements EL1 and NPREC(EL1, SYS1) in system SYS1).

3. The character string should be placed between apostrophes according to the standard FORTRAN specifications.
APPENDIX C

SOURCE CODES OF THE USER-DEFINED SUBROUTINES

C1. The 2D routines USERSY, USERIN, TIMFIN, TIMMIN, ANGIN2, USERFO, TIMFOR, TIMMOM, USEROU and ANGOU2

SUBROUTINE USERSY
IMPLICIT INTEGER (L)
COMMON /UNITS1/ LUN1, LUN2, LUN3, LUN4, LUN5, LUN6, LUN7, 
  LUN8, LUN9, LUN10, LUN11, LUN12, LUN13, 
  LUN14, LUN15, LUN30
COMMON /UNITS2/ LUN16, LUN17, LUN18, LUN19, LUN20, LUN21, 
  LUN22
COMMON /UNITS3/ LUN23
COMMON /UNITS4/ LUNUS1

C Unit numbers used under the VMS operating system
LUN1 = 11
LUN2 = 12
LUN3 = 13
LUN4 = 14
LUN5 = 15
LUN6 = 16
LUN7 = 17
LUN8 = 18
LUN9 = 19
LUN10 = 20
LUN11 = 21
LUN12 = 22
LUN13 = 23
LUN14 = 24
LUN15 = 25
LUN30 = 40
LUN16 = 26
LUN17 = 27
LUN18 = 28
LUN19 = 29
LUN20 = 30
LUN21 = 31
LUN22 = 32
LUN23 = 33

C Logical unit numbers for extra files to be opened by
C the user
LUNUS1 = 99
RETURN
END

SUBROUTINE USERIN
IMPLICIT INTEGER (L)
PARAMETER (NKEY=3)
CHARACTER KEY(NKEY)*6
INTEGER LNKEY
COMMON /UNITS1/ LUN1, LUN2, LUN3, LUN4, LUN5, LUN6, LUN7, 
  LUN8, LUN9, LUN10, LUN11, LUN12, LUN13, 
  LUN14, LUN15, LUN30

DATA KEY/'EXTFOR','EXTMOM','ANGLES'/, LNKEY/NKEY*6/
WRITE (LUN3,10)
10 FORMAT(/,' '',132('**'),/,' USER-DEFINED PART',' ','
$ 132('**')
20 CALL GETKEY(IKEY,KEY,NKEY,6,LNKEY,1)
IF ( IKEY .EQ. 1 ) THEN
   CALL TIMFIN
ELSE IF ( IKEY .EQ. 2 ) THEN
   CALL TIMMIN
ELSE IF ( IKEY .EQ. 3 ) THEN
   CALL ANGIN2
ELSE
   RETURN
END IF
GO TO 20
END

SUBROUTINE TIMFIN
INTEGER QNTFOR, QRFUN, QCFUN
PARAMETER (QNTFOR=15, QRFUN=201, QCFUN=2*QNTFOR)
CHARACTER*15 TFID
COMMON /TIMF1/ NTFOR,TFOR(8,QNTFOR)
COMMON /TIMF2/ TFID(QNTFOR)
COMMON /TIMF3/ FUN(QRFUN,QCFUN)
COMMON /UNITS1/ LUN1,LUN2,LUN3,LUN4,LUN5,LUN6,LUN7,
$ LUN8,LUN9,LUN10,LUN11,LUN12,LUN13,
$ LUN14,LUN15,LUN30
READ (LUN1,*) NTFOR
IF (NTFOR .EQ. 0)
   RETURN
ELSE IF (NTFOR .GT. QNTFOR)
   WRITE (LUN3,*) 'QNTFOR IN SUBROUTINE TIMFIN IS TOO SMALL'
   STOP
END IF
WRITE (LUN3,10)
10 FORMAT(/,' ** USER DEFINED FORCE MODEL TIMFOR **',/,
$ ' NO | SYS1| EL1| POINT OF APPLICATION |',
$ ' SYS2| EL2| FUNCTION NUMBERS |',/,
$ ' | | | Y Z |',
$ ' | | | Y Z |')
DO 20 LV = 1,NTFOR
   READ (LUN1,*) TFOR(I,LV),I=1,8),TFID(LV)
   ISN1 = NINT(TFOR(1,LV))
   IEN1 = NINT(TFOR(2,LV))
   ISN2 = NINT(TFOR(5,LV))
   IEN2 = NINT(TFOR(6,LV))
   NY = NINT(TFOR(7,LV))
   NZ = NINT(TFOR(8,LV))
   WRITE (LUN3,30) LV,ISN1,IEN1,(TFOR(J,LV),J=3,4),
$ ISN2,IEN2,NZ,TFID(LV)
20 CONTINUE
30 FORMAT(I3,' ',I3,' | ','I3,' | ',F8.3,F11.3,' | ',
$ I3,' ',I3,' | ','I6,I8,' | ',A15)
READ (LUN1,*) NFUN
IF (NFUN .GT. QCFUN) THEN
WRITE (LUN3,*) 'QCFUN IN SUBROUTINE TIMFIN IS TOO SMALL'
STOP
ELSE IF (NFUN .GT. 0) THEN
   WRITE (LUN3,40)
   FORM(A,'** FORCE-TIME CHARACTERISTICS **')
   DO 100 LV = 1,NFUN
      READ (LUN1,*) NP,(FUN(I,LV),I=2,NP+NP+1)
      IF ((2*NP+1).GT. QRFUN) THEN
         WRITE (LUN3,*) 'QRFUN IN SUBROUTINE TIMFIN IS TOO SMALL'
      END IF
   END IF
   RETURN
END IF
READ (LUN1,*) NTMOM
IF (NTMOM .EQ. 0) THEN
   RETURN
ELSE IF (NTMOM .GT. QNTMOM) THEN
   WRITE (LUN3,*) 'QNTMOM IN SUBROUTINE TIMMIN IS TOO SMALL'
END IF
STOP
READ (LUN1,*) NTMOM
$ '
READ (LUN1,*) NFUN
IF (NFUN .GT. QCFUN) THEN
   WRITE (LUN3,*) 'QCFUN IN SUBROUTINE TIMMIN IS TOO SMALL'
END IF
SUBROUTINE TIMMIN
   INTEGER QNTMOM, QRFUN, QCFUN, TMOM
   PARAMETER (QNTMOM=15, QRFUN=201, QCFUN=QNTMOM)
   CHARACTER*15 TMID
   COMMON /TIMM1/ NTMOM,TMOM(4,QNTMOM)
   COMMON /TIMM2/ TMID(QNTMOM)
   COMMON /TIMM3/ FUN(QRFUN,QCFUN)
   COMMON /UNITS1/ LUN1,LUN2,LUN3,LUN4,LUN5,LUN6,LUN7,
                    LUN8,LUN9,LUN10,LUN11,LUN12,LUN13,
                    LUN14,LUN15,LUN30
END
STOP
ELSE IF (NFUN .GT. 0) THEN
    WRITE (LUN3,40)
    40 FORMAT(/,** TORQUE-TIME CHARACTERISTICS **')
    DO 100 LV = 1,NFUN
       READ (LUN1,*) NP,(FUN(I,LV),I=2,NP+NP+1)
    IF ((2*NP+1) .GT. QRFUN) THEN
       WRITE (LUN3,*) 'QRFUN IN SUBROUTINE TIMMIN ',
       'IS TOO SMALL'
       STOP
    END IF
    WRITE (LUN3,50) NP,(FUN(I,LV),I=2,NP+NP+1)
    50 FORMAT(/,I4,/,6(F8.4,F10.2))
    FUN(1,LV) = FLOAT(NP)
    CONTINUE
END IF
RETURN
END

SUBROUTINE ANGIN2
IMPLICIT INTEGER (Q)
PARAMETER (QNPOUT=100, NV=4)
CHARACTER*1 SFLD
INTEGER ANGR, ITYP(NV), LOC(NV), LENF(NV), IFLD(NV)
REAL FLD(NV)
COMMON /LINE1/ SFLD(132)
COMMON /OUTP15/ NANGR, ANGR(4,QNPOUT)
COMMON /UNITS1/ LUN1,LUN2,LUN3,LUN4,LUN5,LUN6,LUN7,
                 LUN8,LUN9,LUN10,LUN11,LUN12,LUN13,
                 LUN14,LUN15,LUN30
SAVE ITYP
DATA ITYP/NV*0/
WRITE (LUN3,10)
10 FORMAT(/,** SYSTEM ELEMENT COMBINATIONS FOR WHICH THE',
     ' RELATIVE ANGLE WILL BE CALCULATED **',/,
     ' ANGLE | RELATIVE TO | ',/,
     ' SYS | EL | SYS | EL | ')
20 CALL RDLINE(NV,ITYP,LOC,LENF,FLD,IFLD,SFLD,IERR)
    IF ( IERR .NE. 2 ) THEN
       IF ( IERR .LT. 0 ) THEN
          CALL WRIPAR(0,'QNPOUT')
       ELSE
          NANGR = NANGR + 1
          CALL CPYINT(IFLD(1),ANGR(1,NANGR),NV)
          ISN1 = ANGR(1,NANGR)
          IEN1 = ANGR(2,NANGR)
          ISN2 = ANGR(3,NANGR)
          IEN2 = ANGR(4,NANGR)
          CALL CHESEL(ISN1,IEN1)
          CALL CHESEL(ISN2,IEN2)
       END IF
       IF ( (ISN1 .EQ. 0) .AND. (IEN1 .LE. 0) ) THEN
          ANGR(2,NANGR) = 1
       END IF
IF ( (ISN2 .EQ. 0) .AND. (IEN2 .LE. 0) ) THEN
  ANGR(4,NANGR) = 1
END IF
WRITE (LUN3,30) (ANGR(J,NANGR),J=1,4)
FORMAT(2(I4, ' |'),I4, ' |',I4, ' |')
END IF
GO TO 20
END IF
RETURN
END

SUBROUTINE USERFO
CALL TIMFOR
CALL TIMMOM
RETURN
END

SUBROUTINE TIMFOR
INTEGER QNTFOR, QRFUN, QCFUN
PARAMETER (QNTFOR=15, QRFUN=201, QCFUN=2*QNTFOR)
LOGICAL EXTR
COMMON /MAIN1/ T0,T,TS,TE,TSR,TRST
COMMON /TIMF1/ NTFOR,TFOR(8,QNTFOR)
COMMON /TIMF3/ FUN(QRFUN,QCFUN)
IF (NTFOR .EQ. 0) THEN
  RETURN
END IF
DO 100 LV = 1,NTFOR
  ISN1 = NINT(TFOR(1,LV))
  IEN1 = NINT(TFOR(2,LV))
  YP = TFOR(3,LV)
  ZP = TFOR(4,LV)
  ISN2 = NINT(TFOR(5,LV))
  IEN2 = NINT(TFOR(6,LV))
  NY = NINT(TFOR(7,LV))
  NZ = NINT(TFOR(8,LV))
  C Calculate force at time T
  EXTR = .FALSE.
  CALL FUNCTI(FUN,QRFUN,QCFUN,NY,T,FYA,EXTR)
  CALL FUNCTI(FUN,QRFUN,QCFUN,NZ,T,FZA,EXTR)
  C Express force components in local coordinate system of
  C element IEN1 of system ISN1 and store force in table FF
  IF (ABS(FYA)+ABS(FZA) .GT. 0.000000 ) THEN
    MODEL = 10
    IF (.NOT.((ISN1.EQ.ISN2).AND.(IEN1.EQ.IEN2))) THEN
      CALL TRANS2(2,ISN2,IEN2,FYA,FZA,ISN1,IEN1,FY,FZ)
      CALL FILFF2(MODEL,LV,ISN1,IEN1,YP,ZP,FY,FZ)
    ELSE
      CALL FILFF2(MODEL,LV,ISN1,IEN1,YP,ZP,FYA,FZA)
    END IF
  END IF
100 CONTINUE
RETURN
END
SUBROUTINE TIMMOM
INTEGER QNTMOM, QRFUN, QCFUN, QMIEL, QNST, TMOM
PARAMETER (QNTMOM=15, QRFUN=201, QCFUN=QNTMOM, QMIEL=30, QNST=30)
LOGICAL EXTR
COMMON /MAIN1/ T0, T, TS, TE, TSR, TRST
COMMON /TIMM1/ NTMOM, TMOM(4, QNTMOM)
COMMON /TIMM3/ FUN(QRFUN, QCFUN)
COMMON /PRECE1/ NPREC(QMIEL, QNST)
IF (NTMOM .EQ. 0) THEN
    RETURN
END IF
DO 100 LV = 1, NTMOM
    IPAR = TMOM(1, LV)
    ISN = TMOM(2, LV)
    IEN = TMOM(3, LV)
    NMOM = TMOM(4, LV)
    IENPRE = NPREC(IEN, ISN)
    C Calculate torque at time T
    EXTR = .FALSE.
    CALL FUNCTI(FUN, QRFUN, QCFUN, NMOM, T, FMOM, EXTR)
    C Apply torque FMOM on element IEN of system ISN and store
    C torque in table MM. If IPAR=2 then also apply torque
    C -FMOM on the element preceding element IEN (IENPRE) and
    C store both torques in table MM
    MODEL = 10
    CALL FILMM2(MODEL, LV, ISN, IEN, FMOM)
    IF (IPAR .EQ. 2) THEN
        CALL FILMM2(MODEL, LV, ISN, IENPRE, -FMOM)
    END IF
END IF
100 CONTINUE
RETURN
END

SUBROUTINE USEROU
CALL ANGOU2
RETURN
END

SUBROUTINE ANGOU2
IMPLICIT INTEGER (Q)
PARAMETER (QNST=30, QIEL=60, QNNS=15, QNPOUT=100)
PARAMETER (QISYS=2*QNST+QIEL)
INTEGER ST, ANGR
CHARACTER*50 RUNNO, RUNID1, RUNID2
CHARACTER*15 INERID, NULLID, SYSID, IELID, ID1, ID2
LOGICAL LSTART
COMMON /CONFI1/ NST, ST(5, QNST)
COMMON /MAIN1/ T0, T, TS, TE, TSR, TRST
COMMON /NULLS4/ DOR(4, QNNS), ANGNUL(QNNS), ANNULL(4, QNNS),
                $ FANGN(QNNS), FANULL(4, QNNS)
COMMON /OUTP15/ NANGR, ANGR(4, QNPOUT)
COMMON /SIMU11/ RUNNO, RUNID1, RUNID2
COMMON /SOLDI1/ PCC(QISYS), SCC(QISYS), VCC(QISYS)
COMMON /SYSTIO/ INERID, NULLID(QNNS)
COMMON /SYST1/ SYSID(QNST),IELID(QIEL)
COMMON /UNITS4/ LUNUS1
SAVE LSTART
DATA LSTART /=.TRUE./
IF ( NANGR .EQ. 0 ) RETURN
IF ( LSTART ) THEN
  OPEN (UNIT= LUNUS1,FILE='ANGLE',STATUS='NEW')
  REWIND LUNUS1
  WRITE ( LUNUS1,10) RUNID1,RUNNO,NANGR,3
  FORMAT( ' ',A50,'/',',A50,'/,214)
  DO 25 I = 1,NANGR
      ISN1 = ANGR(1,I)
      IEN1 = ANGR(2,I)
      ISN2 = ANGR(3,I)
      IEN2 = ANGR(4,I)
      IF ( ISN1 .GT. 0 ) THEN
          IT = ST(3,ISN1) + IEN1
          ID1 = IELID(IT)
      ELSE IF ( ISN1 .EQ. 0 ) THEN
          ID1 = NULLID(IEN1)
      ELSE
          ID1 = INERID
      END IF
      IF ( ISN2 .GT. 0 ) THEN
          IT = ST(3,ISN2) + IEN2
          ID2 = IELID(IT)
      ELSE IF ( ISN2 .EQ. 0 ) THEN
          ID2 = NULLID(IEN2)
      ELSE
          ID2 = INERID
      END IF
      WRITE (LUNUS1,20) 101,102
  FORMAT( ',A15,' - ',A15)
  25 CONTINUE
  WRITE (LUNUS1,60) 1000.*T
  DO 50 I = 1,NANGR
      ISN1 = ANGR(1,I)
      IEN1 = ANGR(2,I)
      ISN2 = ANGR(3,I)
      IEN2 = ANGR(4,I)
      C Determine the angle of ISN1/IEN1
      IF ( ISN1 .GT. 0 ) THEN
          IROOT = ST(5,ISN1)
          THETA1 = PCC(IROOT + 2 + IEN1)
      ELSE IF ( ISN1 .EQ. 0 ) THEN
          THETA1 = ANGNUL(IEN1)
      ELSE
          THETA1 = 0.
END IF
C Determine the angle of ISN2/IEN2
IF ( ISN2 .GT. 0 ) THEN
   IROOT = ST(5,ISN2)
   THETA2 = PCC(IROOT + 2 + IEN2)
ELSE IF ( ISN2 .EQ. 0 ) THEN
   THETA2 = ANGNUL(IEN2)
ELSE
   THETA2 = 0.
END IF
C Determine the relative angle of ISN1/IEL1 w.r.t. ISN2/IEL2
THETAR = THETA1 - THETA2
WRITE ( LUNUS1,60 ) THETA1,THETA2,THETAR
50 CONTINUE
60 FORMAT(3(1PE14.6))
RETURN
END

C2. The 3D routines USERSY, USERIN, TIMFIN, TIMMIN, USERFO, TIMFOR and TIMMOM

SUBROUTINE USERSY
IMPLICIT INTEGER (L)
COMMON /UNITS1/ LUN1,LUN2,LUN3,LUN4,LUN5,LUN6,LUN7,
$ LUN8,LUN9,LUN10,LUN11,LUN12,LUN13,
$ LUN14,LUN15,LUN30
COMMON /UNITS2/ LUN16,LUN17,LUN18,LUN19
COMMON /UNITS3/ LUN23
COMMON /UNITS4/ LUNUS1
C Unit numbers used under the VMS operating system
LUN1 = 11
LUN2 = 12
LUN3 = 13
LUN4 = 14
LUN5 = 15
LUN6 = 16
LUN7 = 17
LUN8 = 18
LUN9 = 19
LUN10 = 20
LUN11 = 21
LUN12 = 22
LUN13 = 23
LUN14 = 24
LUN15 = 25
LUN30 = 40
LUN16 = 26
LUN17 = 27
LUN18 = 28
LUN19 = 29
LUN23 = 33
C Logical unit numbers for extra files to be opened by the user
LUNUS1 = 99
RETURN
END

SUBROUTINE USERIN
IMPLICIT INTEGER (L)
PARAMETER (NKEY=2)
CHARACTER KEY(NKEY)*6
INTEGER LNKEY(NKEY)
COMMON /UNITS1/ LUN1,LUN2,LUN3,LUN4,LUN5,LUN,LUN7,
$ LUN8,LUN9, LUN10,LUN11,LUN12,LUN13,LUN14,
$ LUN15,LUN30
DATA KEY/'EXTFOR','EXTMOM'/,
$ LNKEY/NKEY*6/
WRITE (LUN3,10)
10 FORMAT(/,'** USER-DEFINED PART**',//,
$ ',132('**'),/,' USER-DEFINED PART',/,
$ ',132('**'))
20 CALL GETKEY(IKEY,KEY,NKEY,6,LNKEY,1)
IF ( IKEY .EQ. 1 ) THEN
 CALL TIMFIN
ELSE IF ( IKEY .EQ. 2 ) THEN
 CALL TIMMIN
ELSE
 RETURN
END IF
GO TO 20
END

SUBROUTINE TIMFIN
INTEGER QNTFOR, QRFUN, QCFUN
PARAMETER (QNTFOR=15, QRFUN=201, QCFUN=3*QNTFOR)
CHARACTER*15 TFID
COMMON /TIMFO1/ NTOR,TFOR(10,QNTFOR)
COMMON /TIMFO2/ TFID(QNTFOR)
COMMON /TIMFO3/ FUN(QRFUN,QCFUN)
COMMON /UNITS1/ LUN1,LUN2,LUN3,LUN4,LUN5,LUN6,LUN7,
$ LUN8,LUN9, LUN10,LUN11,LUN12,LUN13,
$ LUN14,LUN15,LUN30
READ (LUN1,*) NTOR
IF (NTFOR .EQ. 0) THEN
 RETURN
ELSE IF (NTFOR .GT. QNTFOR) THEN
 WRITE (LUN3,*) 'QNTFOR IN SUBROUTINE TIMFIN IS',
$ ' TOO SMALL'
 STOP
 END IF
 WRITE (LUN3,10)
10 FORMAT(/
$ ',** USER DEFINED FORCE MODEL TIMFOR **',//,
$ ', NO | SYS1| EL1| POINT OF APPLICATION |',
$ ', SYS2| EL2 | FUNCTION NUMBERS |',/,
$ ', | | | X Y Z |',
$ ', | | | X Y Z |')
DO 20 LV = 1,NTFOR
 READ (LUN1,*) (TFOR(I,LV),I=1,10),TFID(LV)
 ISN1 = NINT(TFOR(1,LV))
 IEN1 = NINT(TFOR(2,LV))
 ISN2 = NINT(TFOR(6,LV))
 IEN2 = NINT(TFOR(7,LV))

NX = NINT(TFOR(8,LV))
NY = NINT(TFOR(9,LV))
NZ = NINT(TFOR(10,LV))
WRITE (LUN3,30) LV,ISN1,IEN1,(TFOR(J,LV),J=3,5),
$ ISN2,IEN2,NX,NY,NZ,TFID(LV)
$ 20 CONTINUE
$ 30 FORMAT(I3,' |',I3,' |',I3,' |',3F7.3,' |',3(I4,1X),' |',A15)
READ (LUN1,*) NFUN
IF (NFUN .GT. QCFUN) THEN
WRITE (LUN3,*) 'QCFUN IN SUBROUTINE TIMFIN IS TOO SMALL'
STOP
ELSE IF (NFUN .GT. 0) THEN
WRITE (LUN3,40)
$ 40 FORMAT(/,*,** FORCE-TIME CHARACTERISTICS ***)
DO 100 LV = 1,NFUN
READ (LUN1,*) NP,(FUN(I,LV),I=2,NP+NP+1)
IF ((2*NP+1) .GT. QRFUN) THEN
WRITE (LUN3,*) 'QRFUN IN SUBROUTINE TIMFIN ',
$ 'IS TOO SMALL'
STOP
END IF
WRITE (LUN3,50) NP,(FUN(I,LV),I=2,NP+NP+1)
$ 50 FORMAT(/,I4,/,6(F8.4,F10.2))
FUN(1,LV) = FLOAT(NP)
100 CONTINUE
END IF
RETURN
END

SUBROUTINE TIMMIN
INTEGER QNTMOM, QRFUN, QCFUN, TMOM
PARAMETER (QNTMOM=15, QRFUN=201, QCFUN=3*QNTMOM)
CHARACTER*15 TMID
COMMON /TIMMO1/ NTMOM,TMOM(8,QNTMOM)
COMMON /TIMMO2/ TMID(QNTMOM)
COMMON /TIMMO3/ FUN(QRFUN,QCFUN)
COMMON /UNITSl/ LUN1,LUN2,LUN3,LUN4,LUN5,LUN6,LUN7,
$ LUN8,LUN9, LUN10,LUN11,LUN12,LUN13,
$ LUN14,LUN15,LUN30
READ (LUN1,*) NTMOM
IF (NTMOM .EQ. 0) THEN
RETURN
ELSE IF (NTMOM .GT. QNTMOM) THEN
WRITE (LUN3,*) 'QNTMOM IN SUBROUTINE TIMMIN IS',
$ 'TOO SMALL'
STOP
END IF
WRITE (LUN3,10)
$ 10 FORMAT(/,* USER DEFINED TORQUE MODEL TIMMOM **',/,,
$ ' NO | IPAR| SYS1| EL1| SYS2| EL2| FUNCTION NUMBERS',
$ ' |'/'',
$ ' | | | | | X Y Z |')
DO 20 LV = 1,NTMOM
READ (LUN1,*) (TMOM(I,LV),I=1,8),TMID(LV)
IPAR = TMOM(1,LV)
ISN1 = TMOM(2,LV)
IEN1 = TMOM(3,LV)
ISN2 = TMOM(4,LV)
IEN2 = TMOM(5,LV)
NX = TMOM(6,LV)
NY = TMOM(7,LV)
NZ = TMOM(8,LV)
WRITE (LUN3,30) LV,IPAR,ISN1,IEN1,ISN2,IEN2,NX,NY,$
NZ,TMID(LV)
20 CONTINUE
30 FORMAT(I3,' |',I3,' |',I3,' |',I3,' |',I3,' |',I3,' |',I3,$
' |',3(I4,1X),' | ',A15)
READ (LUN1,*) NFUN
IF (NFUN .GT. QCFUN) THEN
WRITE (LUN3,*) 'QCFUN IN SUBROUTINE TIMMIN IS TOO SMALL'
STOP
ELSE IF (NFUN .GT. 0) THEN
WRITE (LUN3,40)
40 FORMAT(/,' ** TORQUE-TIME CHARACTERISTICS **')
DO 100 LV = 1,NFUN
READ (LUN1,*) NP,(FUN(I,LV),I=2,NP+NP+1)
IF ((2*NP+1) .GT. QRFUN) THEN
WRITE (LUN3,*) 'QRFUN IN SUBROUTINE TIMMIN IS TOO SMALL'
ELSE IF (NFUN .GT. 0) THEN
WRITE (LUN3,40)
40 FORMAT(/,' ** TORQUE-TIME CHARACTERISTICS **')
DO 100 LV = 1,NFUN
READ (LUN1,*) NP,(FUN(I,LV),I=2,NP+NP+1)
IF ((2*NP+1) .GT. QRFUN) THEN
WRITE (LUN3,*) 'QRFUN IN SUBROUTINE TIMMIN IS TOO SMALL'
END IF
STOP
END IF
WRITE (LUN3,50) NP,(FUN(I,LV),I=2,NP+NP+1)
50 FORMAT(/,I4,/,6(F8.4,F10.2))
FUN(1,LV) = FLOAT(NP)
100 CONTINUE
END IF
RETURN
END

SUBROUTINE USERFO
CALL TIMFOR
CALL TIMMOM
RETURN
END

SUBROUTINE TIMFOR
INTEGER QNTFOR, QRFUN, QCFUN
PARAMETER (QNTFOR=15, QRFUN=201, QCFUN=3*QNTFOR)
LOGICAL EXTR
COMMON /MAIN1/ T0,T,TS,TE,TSR,TRST
COMMON /TIMF01/ NTFOR,TFOR(10,QNTFOR)
COMMON /TIMF03/ FUN(QRFUN,QCFUN)
IF (NTFOR .EQ. 0) THEN
RETURN
END IF
DO 100 LV = 1,NTFOR
ISN1 = NINT(TFOR(1,LV))
IEN1 = NINT(TFOR(2,LV))
XP = TFOR(3,LV)
YP = TFOR(4,LV)
ZP = TFOR(5,LV)
ISN2 = NINT(TFOR(6, LV))
IEN2 = NINT(TFOR(7, LV))
NX = NINT(TFOR(8, LV))
NY = NINT(TFOR(9, LV))
NZ = NINT(TFOR(10, LV))

C Calculate force at time T
EXTR = .FALSE.
CALL FUNCTI(FUN, QRFUN, QCFUN, NX, T, FXA, EXTR)
CALL FUNCTI(FUN, QRFUN, QCFUN, NY, T, FYA, EXTR)
CALL FUNCTI(FUN, QRFUN, QCFUN, NZ, T, FZA, EXTR)

C Express force components in local coordinate system of
element IEN1 of system ISN1 and store force in table FF
IF (ABS(FXA)+ABS(FYA)+ABS(FZA) .GT. 0.000000) THEN
MODEL = 10
IF (.NOT. (ISN1.EQ.ISN2).AND.(IEN1.EQ.IEN2)) THEN
CALL TRANS(2, ISN2, IEN2, FXA, FYA, FZA, ISN1, IEN1, FX, FY, FZ)
ELSE
CALL FILLFF(MODEL, LV, ISN1, IEN1, XP, YP, ZP, FX, FY, FZ)
ENDIF
ENDIF
100 CONTINUE
RETURN
END

SUBROUTINE TIMMOR
INTEGER QNTMOM, QRFUN, QCFUN, QMIEL, QNST, TMOM
PARAMETER(QNTMOM=15, QRFUN=201, QCFUN=3*QNTMOM, QMIEL=30, $ QNST=30)
LOGICAL EXTR
COMMON /MAIN1/ T0, T, TS, TE, TSR, TRST
COMMON /TIMM01/ NTMOM, TMOM(8, QNTMOM)
COMMON /TIMM03/ FUN(QRFUN, QCFUN)
COMMON /PRECE1/ NPREC(QMIEL, QNST)
IF (NTMOM .EQ. 0) THEN
RETURN
ENDIF
DO 100 LV = 1, NTMOM
IPAR = TMOM(1, LV)
ISN1 = TMOM(2, LV)
IEN1 = TMOM(3, LV)
ISN2 = TMOM(4, LV)
IEN2 = TMOM(5, LV)
NX = TMOM(6, LV)
NY = TMOM(7, LV)
NZ = TMOM(8, LV)
IENPRE = NPREC(IEN1, ISN)
C Calculate torque at time T
EXTR = .FALSE.
CALL FUNCTI(FUN, QRFUN, QCFUN, NX, T, FXA, EXTR)
CALL FUNCTI(FUN, QRFUN, QCFUN, NY, T, FYA, EXTR)
CALL FUNCTI(FUN, QRFUN, QCFUN, NZ, T, FZA, EXTR)
C Express torque components in local coordinate system of
element IEN1 of system ISN1 and store torque in table
C MM. If IPAR=2 then also express torque components in local coordinate system of element IENPRE of system ISN1 and store negative value of torque in table MM.

IF (ABS(FXA)+ABS(FYA)+ABS(FZA) .GT. 0.000000 ) THEN
  MODEL = 10
  IF (.NOT.((ISN1.EQ.ISN2).AND.(IEN1.EQ.IEN2))) THEN
    CALL TRANS(2,ISN2,IEN2,FXA,FYA,FZA,ISN1,IEN1,FX,FY,FZ)
    CALL FILLMM(MODEL,LV,ISN1,IEN1,FX,FY,FZ)
  ELSE
    CALL FILLMM(MODEL,LV,ISN1,IEN1,FXA,FYA,FZA)
  END IF
  IF (IPAR .EQ. 2) THEN
    IF (.NOT.((ISN1.EQ.ISN2).AND.(IENPRE.EQ.IEN2))) THEN
      CALL TRANS(2,ISN2,IEN2,FXA,FYA,FZA,ISN1,IENPRE,FX,FY,FZ)
      CALL FILLMM(MODEL,LV,ISN1,IENPRE,-FX,-FY,-FZ)
    ELSE
      CALL FILLMM(MODEL,LV,ISN1,IENPRE,-FXA,-FYA,-FZA)
    END IF
  END IF
END IF
RETURN
END
APPENDIX D

MADYMO INPUT FILES FOR THE SIMULATIONS IN CHAPTER 3

D1. 2D analytical example

2D EXAMPLE
LINACCEL
OCTOBER 1989
0. 0.1 0.01
0 0.001
0. 0.5 0.01 0.1
SYSTEM 1
ROD
CONFIGURATION
1
-999
GEOMETRY
0. 0. 0. 0. ROD
-999
INERTIA
0.7 0.021
-999
ELLIPSES
1 0.0075 0.0075 0. 0. 2 0 0 0 0. CENTRE
-999
PLANES
1 -0.3 0.3 0. 0 0 0 ROD
-999
INITIAL CONDITIONS
0. 0. 0. 0.
ORIENTATIONS
1 -1 0.
-999
ANGULAR VELOCITY
1 0.
-999
END SYSTEM 1
OUTPUT
1 0 0.01 1
END OUTPUT
USER INPUT
EXTFOR
1
1 1 0. 0. -1 0 1 1 'FORCE'
1
2
0. 140.
0.25 140.
EXTMOM
1
1 1 1 1 'TORQUE'
1
2
0. 6.3
D2. 3D analytical example

3D EXAMPLE

3D EXAMPLE

ROTATIONS

OCTOBER 1989

0. 0.1 0.01
0 0.001
0. 0.5 0.01 0.1

INERTIAL SPACE

Y AXIS

ELLIPSOIDS

1 0.0075 0.0075 0.0075 0. 0.34641 0. 2 0 0 0 EARTH
1 0.0075 0.0075 0.0075 0. -0.34641 0. 2 0 0 0 SPHERE

-999

PLANES

0 0. -0.34641 0.0025 0. -0.34641 -0.0025 0. 0.34641 -0.0025 +
0 0 0 Y AXIS

-999

END INERTIAL SPACE

SYSTEM 1

BARBELL

CONFIGURATION 1

-999

GEOMETRY

0. 0.0 0. 0. 0. 0. BARBELL

-999

INERTIA

0.15 0.1923 0.1923 0.0003

-999

ELLIPSOIDS

1 0.0075 0.0075 0.0075 0. 0. 0. 2 0 0 0 CENTER
1 0.05 0.05 0.05 0. 0. -0.4 2 0 0 0 LEFTEND
1 0.05 0.05 0.05 0. 0. 0.4 2 0 0 0 RIGHTEND

-999

PLANES

1 0. 0.0025 -0.4 0. -0.0025 -0.4 0. 0.0025 0.4 0 0 0 ROD

-999

INITIAL CONDITIONS

0. 0. 0. 0. 0. 0.

ORIENTATIONS

1 -1 1 1 -1.04719756

-999

ANGULAR VELOCITY

1 0. 31.4159265 54.413981

-999

END SYSTEM 1

SYSTEM 2

MOON

CONFIGURATION 1

-999

GEOMETRY
0. 0. 0. 0. 0. 0. SPHERE

INERTIA
0.15 1.5E-4 1.5E-4 1.5E-4

ELLIPSOIDS
1 0.05 0.05 0.05 0. 0.0. 2 0 0 0 CENTER

PLANES
1 0.05 0.05 0.05 -0.05 0. -0.05 -0.05 0. 0 0 0 MIDPLANE

INITIAL CONDITIONS
0. 0.3464102 -0.2 -12.566371 0. 0.

ORIENTATIONS
1 -1 1 1 0.

ANGULAR VELOCITY
1 0. 62.831853 0.

END SYSTEM 2

OUTPUT
1 0 0.01 1

LINDIS
1 1 0. 0. 0.4 -1 0 SPHERE CENTER
1 1 0. 0. -1 0 ORIGIN
2 1 0. 0. -1 0 MOON CENTER

LINVEL
1 1 0. 0. 0.4 0 SPHERE CENTER
2 1 0. 0. 0 MOON CENTER

LINACC
1 1 0. 0. 0.4 0 0 0 0 0 SPHERE CENTER
2 1 0. 0. 0 0 0 0 0 0 MOON CENTER

ANGVEL
1 1 0.

ANGACC
1 1 0.

END OUTPUT

USER INPUT

EXTFOR
1
2 1 0. 0. 2 1 0 0 1 'CENTRIPETAL'
1
2
0. 118.43525
0.25 118.43525

EXTMOM
1
1 1 1 1 1 1 0 0 'TORQUE'
1
2
0. -328.2174
D3. A subject doing a push-up

PUSH-UP
ONE DF
OCTOBER 1989
0. 1. 0.01
0 0.001
0. 0.5 0.01 0.1
INERTIAL SPACE
FLOOR
PLANES
-1 -0.20 -.02 1.5 -0.02 0 0 0 FLOOR
-999
END INERTIAL SPACE
SYSTEM 1
HUMAN BODY
CONFIGURATION
5 4 3 2 1
-999
GEOMETRY
0. 0. 0. 0. FEET
0. 0. 0. 0.90 HTL
0. 1.40 0. 0.14 UPPER ARMS
0. 0.32 0. 0.13 LOWER ARMS
0. 0.30 0. HANDS
-999
INERTIA
1000000000. 100.
70 7.5
5.0 0.035
3.0 0.05
10000000. 100.
-999
JOINTS
4 1 0 0 0 35 0 0. 0.
-999
FUNCTIONS
2
-3.14159265 0
0 2827.4334
-999
ELLIPSES
1 0.036 0.09 0. -0.05 4 0 0 0 0 0. FEET
5 0.02 0.08 0. 0.07 4 0 0 0 0 0. HANDS
-999
PLANES
2 0. 0. 0. 1.4 0 0 0 HTL
3 0. 0. 0.32 0 0 0 UPPER ARMS
4 0. 0. 0.30 0 0 0 LOWER ARMS
-999
INITIAL CONDITIONS
1.3076697 0.12 0. 0.
ORIENTATIONS
1 -1 0.
2 -1 1.5617266
3 1 3.3766548
4 1 -2.6179939
5 -1 1.5707963
-999
ANGULAR VELOCITY
1 0.
2 0.
3 0.
4 0.
5 0.
-999
END SYSTEM 1
FORCE MODELS
ACCELERATION FIELDS
0 0 0 1
1 1 0 0
1 5 0 0
-999
FUNCTIONS
2
0.0 -9.8066 2.0 -9.8066
-999
END FORCE MODELS
OUTPUT
1 0 0.1 1
END OUTPUT
USER INPUT
EXTMOM
1
2 1 4 -1 'ELBOW'
1
56
0.0000 471.2389221 0.0200 484.8042908 0.0400 503.3585815
0.0600 526.6754761 0.0800 554.5288696 0.1000 586.6925049
0.1200 622.9401855 0.1400 663.0457764 0.1600 706.7830200
0.1800 753.9257202 0.2000 804.2478027 0.2200 857.5228882
0.2400 913.5249634 0.2600 972.0276489 0.2800 1032.8049316
0.3000 1095.6304932 0.3200 1160.2782192 0.3400 1226.5219727
0.3600 1294.1352539 0.3800 1362.8922119 0.4000 1432.5665283
0.4200 1502.9317627 0.4400 1573.7620850 0.4600 1644.8312988
0.4800 1715.9129639 0.5000 1786.7810059 0.5200 1857.2092285
0.5400 1926.9713135 0.5600 1995.8415527 0.5800 2063.5930176
0.6000 2130.0000000 0.6200 2194.8361816 0.6400 2257.8754883
0.6600 2318.8913574 0.6800 2377.6582031 0.7000 2433.9492188
0.7200 2487.5385742 0.7400 2538.1997070 0.7600 2585.7070313
0.7800 2629.8337402 0.8000 2670.3547363 0.8200 2707.0419922
0.8400 2739.6704102 0.8600 2768.0136719 0.8800 2791.8457031
0.9000 2810.9406738 0.9200 2825.0717773 0.9400 2834.0124512
0.9600 2837.5375977 0.9800 2835.4196777 1.0000 2827.4338379
1.0200 2813.3532715 1.0400 2792.9523926 1.0600 2766.0029297
1.0800 2732.2817383 1.1000 2691.5598145
ANGLES
1 4 1 3
-999
END INPUT DATA