Biomimetic Motion Synthesis for Synthetic Humanoids

Joshua G. Hale

April 28, 2003

Submitted to the Computer Science Department of the
University of Glasgow for the degree of Doctor of Philosophy.

Abstract

The design and evaluation of synthetic motion production algorithms for synthetic humanoids.

Animation techniques are needed for robot and computer animated humanoids. Humanoid robots are an ever more populous reality and advances in rendering technologies have made it possible to display photo-realistic humanoid actors, but existing motion generation techniques including motion capture are often unable to maintain a similar standard of realism.

Dynamics based humanoid animation aims to solve the problem of realism by simulating a physical humanoid so that all motions are physically consistent. The burden of effort is the generation of realistic motion *control*. Dynamics based animation is well established and much knowledge has been accumulated regarding the many problematic issues of simulating and controlling the mechanics of a humanoid figure. However, existing research has not focused on the specific task of producing motion according to the same principles as humans themselves. This thesis adopts a novel approach by combining animation techniques with human motion production research to solve the problem of motion control.

Human motion production research is strongly related to biomechanics, muscle physiology and psychomotor research and has led to the development of many theories regarding the way humans plan and execute motions. These theories have been used to inspire a number of synthetic motion production models.

Fourteen synthetic motion models are developed according to existing theories of human motion planning and production, and by considering computational efficiency. Formal specifications are made and developed through

to algorithmic implementations requiring the development of new and innovative techniques. The models yield full body humanoid motions based around hand trajectories that may be point to point, multi-point or cyclic.

A unified motion production environment facilitates the generation of synthetic motions from knot point specifications describing the position of the humanoid's hand. The tasks of motion generation, editing, analysis and export for rendering or robot performance are handled in a unified manner by the motion production environment. A novel technique for minimising the N^{th} derivative of an arbitrary dimension spline trajectory is developed and used to solve motion models such as minimum jerk and minimum angular jerk. Dynamics based models such as torque and torque change minimisation are also implemented. Two models are developed with specific attention to computational efficiency in an attempt to capture the qualities of sophisticated models of human motion while reducing the computational burden. These are called the 'minimum torque postures', and 'minimum torque postures virtual trajectory' models.

The motion production models were validated and compared by computational and psychovisual testing. Synthetic motion clips including captured human motion were displayed to subjects who assessed the naturalness and similarities. The results indicate that certain easily implemented synthetic models such as minimum angular jerk can appear very natural and compare well with genuine human motion. The minimum torque postures and minimum torque postures virtual trajectory models also compare well with human motion and demonstrate a significant improvement in computation time with respect to other dynamics based optimisations.

Acknowledgements

I would like to thank the following people, for whose support and assistance I am indebted:

Nobuaki Aoyama, Dr Yukinobu Arata, Professor Chris Atkeson, Darrin Bentivegna, Sylvain Brugnot, Matthew Cairns, Professor Kunihiro Chihara, Dr Lim Chun Ren, Alasdair Coull, Alasdair Flemming, Stuart Ford, Sensei Toshikazu Fukunaga, Douglas Green, Paul Goddard, Mark Harris, Toshinari Higashiura, Dr Harold Hill, Svenja Hoff, Kaori Horikoshi, Tokiko Ichimoto, Takahiro Imashiro Sensei, Dr Xiangyang Ju, Keiko Kawamoto, Toshiyuki Kawasaki, Neil Ketley, Jenny Mao, Professor Mitsuo Kawato, Professor Shin'ya Kotosaka, Tim Langford, Edward Mann, Ayako Matsumoto, S. Miller, Professor Carroll Morgan, Helena Paterson, Dr John Patterson, Dr Nick Pitman, Dr Frank Pollick, Marcia Riley, Professor Stefan Schaal, Dr Tomohiro Shibata, Tal Shalif, Mitsuhiro Shunto, Kuratate Takaaki, Naka Takahito, Benjamin Tipping, Tassos Tombros, Makoto Toudou, Makiko Tatsumi, Ito Tsuneo, Dr Ales Ude, Noriko Uehara, Dr Colin Urquhart, Dr Ray Wan, Guy Wardrop, Dr R. Waterston, Mitsugu Watatani, Ciaran Wills, Hiroko Yamashita, Assistant Professor Yoshihiro Yasumuro, and Tao Yegang.

In addition, I thankfully acknowledge the support of the following groups: Human Information Science Laboratories and Shien foreign support staff at the Advanced Telecommunications Research Institute International in Kyoto, Japan, the Chihara Image Processing Laboratory at the Nara Institute of Technology, Japan, the British Council in Japan, and in particular the staff of the REES programme, the Engineering and Physical Sciences Research Council, the Department of Computing Science at Glasgow University and the 3D-Perception group.

Finally, this thesis was prepared in LaTeX 2_ε, for which I am also very grateful.

この論文を福永敏一先生に捧げます。かの人の寛容さは大きすぎてそれに報いる事が出来ないので唯一私が出来るのは彼を見習う事だけです。

This thesis is dedicated to Toshikazu Fukunaga Sensei, a man whose infinite generosity I can never repay and must therefore emulate.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Theme, objectives & outline

'Biomimetic motion synthesis' refers to the automatic production of motion that mimics a biological entity, in this case a synthetic humanoid.

This thesis will reveal that there is an increasing need for automatic animation techniques for both computer graphic and robotic humanoids.

A survey of dynamics based animation techniques will reveal that dynamic simulation according to Newtonian principles is both physically accurate, and highly akin to the task of controlling an actual robot.

A survey of human motion production research will reveal that much empirical and theoretical knowledge regarding the way humans plan and actuate motions has been amassed.

These research areas are combined, and a number of synthetic motion production models developed. This thesis describes the design and implementation of those models, and the design and implementation of a motion production environment of which they form the larger components. The synthetic models were tested computationally and psychovisually by means of perception experiments with human subjects. These tests, their results and conclusions are presented.

The content of this thesis may therefore be summarised as:

The design and evaluation of synthetic motion production algorithms for synthetic humanoids.

1.1 Theme

This thesis investigates biomimetic motion synthesis for synthetic humanoids. The focus is realistic human-like motion. The research is applicable to the computer animation of virtual humans, humanoid robotics, the perception of human motion (psychophysics), and the study of human motion production.

The areas from which this research draws a basis include dynamics based humanoid animation, dynamic simulation, empirical human motion research, human muscles, muscle models and human motion production research.

All of these areas are related to biomimetic motion synthesis as will be explained below. Following this motivation the research goals are formalised in Section 1.1.1

Dynamics based animation & human motion production

Dynamic analysis is a powerful tool for simulating realistic physical processes. Dynamic simulation automates the process of motion generation and greatly simplifies the animator's task of interacting with and controlling models [Armstrong and Green (1985)]. In the case of humanoids however, the actors themselves must generate convincing motion:

> *Dynamic correctness is a sufficient condition for realistic motion of non-living objects. In animating a self-actuated system, however, visual realism is another important, separate criterion for determining the success of a technique. Dynamic correctness is not a sufficient condition for this visual realism.* [Ko and Badler (1996)]

The difficulty in replicating human motion is due in part to the complex morphology of the skeletal system and the dynamic properties of the muscular

system. Indeed it has been argued that "the human body is the antithesis of shapes for which computer modelling techniques are best suited" [Badler (1982)].

The problem is exacerbated by the fact that humans are particularly expert in observing and understanding animal motion [Lake and Green (1991)]. For example, humans are capable of assessing the mass of carried weights, determining gender and perceiving emotions merely from a set of light points attached to humans' limbs [Cutting and Kozlowski (1977), Bingham (1993)]. There is also evidence to suggest the existence of 'mirror neurones' -motor neurones that actually fire in response to perceptual stimuli of motions involving the muscles they control [Atkeson et al. (2000)].

Animators have suggested modelling animal motor control in order to overcome these problems:

> *...the simulation of articulated animal motion must encompass both Newtonian mechanical laws and the articulated animal's sensory-motor coordination that operates within these laws.* [Girard (1991)]

Various animation systems have been developed in accordance with this idea. The so-called action-perception systems, for example, make simple control decisions based on computationally straightforward conditions on the state space [van de Panne and Fiume (1993), Boone and Hodgins (1997)]. Such systems have been based on simple sub-humanoid articulations and have involved models of motor planning designed without recourse to human motion planning research.

Human motion production however, is a complex dynamic process involving the physics of the human body (biomechanics) and the psychology of motion planning. This includes the mechanics of the skeletal system and the muscles as well as the dynamical properties of the nervous system involving control based in the spine and the brain. Motion *planning* is likewise complicated by issues such as context, practise, coordination, *etc.* Fortunately these areas have been, and are undergoing, research aimed at understanding

the mental and physical processes involved in motion production. It is perhaps surprising that there has been little animation research drawing directly from human motion production.

The literature demonstrates that a broad collection of empirical data have been gathered and analysed, and many theoretical models have been proposed to unify these data. The features of human motion based on empirical evidence provide a way of quantifiably assessing the quality of synthesised motion, and the motion production theories are an excellent basis for the computational simulation of human motion. There is a rich collection of motion production theories, some of which may be implemented directly, and some that would require complex optimisation such as the minimum torque change model.

Designing a set of motion production algorithms based on human motion production theories is a significant contribution to animation research. A broad range of firmly justified motion generation techniques may be developed. The connection itself, between human motion production and animation, is a new direction that promises further possibilities in animation.

While there are already a plethora of motion production models, new computational implementations explore the applicable range of a given model and may ultimately yield unified models capable of explaining all motion production phenomena. Computational implementations of motion production hypotheses are therefore a significant contribution to the field of psychophysics. Computational implementations offer insight into the computational complexity required by the brain, tools for model comparison, and an environment for further psychovisual testing of the models.

Psychovisual testing is an area of relevance to both animation and psychophysics, since it may be used to validate and compare models and their suitability for animation. Results may be analysed to determine which features of human movement are key to generating convincingly natural motion, and hence significant to human perception.

Biomechanics & robotics

Biomechanical research has aimed to understand the mechanical properties of the human body. This research is relevant to dynamics based animation, robotics and human motion production.

Anthropomorphic data, including limb sizes, masses, and tissue density contribute to the construction of dynamic models reflecting the dynamic characteristics of human bodies correctly. Statistical research into anthropomorphic characteristics facilitates the anthropometric generalisation of such data to arbitrary synthetic humanoids. An understanding of muscle properties including force generation, time varying and low-level reflex loops is fundamental to various motion production theories and humanoid animation methods.

Armstrong and Wilhelms were among the first animation authors to draw attention not only to biomechanics, but also to robotics as having direct relevance to dynamics based animation of human figures [Armstrong and Green (1985), Wilhelms (1987), Hodgins et al. (1995)]. More recently the concept was expressed concisely, "...the animated actor will be a simulated robot in a simulated world." [Cerezo et al. (1999)]. Generating animation using a dynamic humanoid requires finding suitable control signals in terms of forces and torques to bring about the desired motion, and simulating the physical results. Robots are also controlled by forces and torques, and the solutions are therefore mutually applicable. It comes as no surprise that much of the original work on dynamics based animation was inspired by dynamics formulations published in the robotics literature.

Previous work by the author has aimed to facilitate physical interactions with humanoid robots [Hale and Pollick (2000a,b)]. An accurate replication of human motion styles is central to the perception of robots as human-like entities rather than machines. This perception encourages people to interact with robots in a similar way as with other humans, thus tapping into a rich range of communicative abilities and encouraging a greater integration and cooperation between robots and humans in general.

1.1.1 Objectives

The aim of this research was to generate and analyse biomimetic motion for a synthetic humanoid. This was motivated by the need to realistically animate virtual humans and humanoid robots. This parallel application allows a comparison between the techniques suitable for, and requirements of the two media. The generative investigation of human motion hypotheses yields various conclusions relevant to psychophysics.

The phases of research following the literature survey include selecting motion production theories for implementation, building a development environment for motion synthesis, generating motions for a computer animated humanoid and a humanoid robot, investigating motions according to empirical features of human motion, investigating motion generation computationally and psychovisually, analysing the results, drawing conclusions and identifying weaknesses in the research conducted. The project goals are formalised below:

1. Model selection

   Choose a set of human motion production theories for implementation as motion production algorithms. Select models covering a range of implementation complexity, with coarse to accurate replication of human movement, representing the work of various research groups, and including both dynamics based and purely kinematic planning.

2. Software development

   Develop a motion production environment with a unified framework facilitating motion generation, editing, visualisation, recording and statistical evaluation. Motion generation will be an off-line process. The focal development aspect will be motion generation and will involve implementing the human motion production theories selected in phase (1). If possible, design computationally efficient motion generation techniques based on, but not limited by human motion production theories.

3. Motion generation

   Generate motions and render them using a computer animated humanoid. Perform the same motions with a humanoid robot and record them with video. Record high frequency motion logs of robot motion to facilitate subsequent analysis.

4. Investigation

   Investigate the models in terms of complexity of implementation and potential for on-line implementation. Investigate motion generation time requirements and compare robot and simulated replication of generated trajectories. Perform psychovisual experiments to assess the naturalness and similarity of motions. Compare human responses to robot and computer generated motion.

5. Analysis

   Analyse computational requirements of motion production models. Assess the perceived naturalness of each motion model developed, and assess the perceptual similarity of the models. Compare motion generation models and motion production theories quantifiably according to features of human motion. Assess the naturalness of motions with respect to human motion features and the psychovisual significance of these features.

6. Conclusions

   Identify the psychovisually significant features of motion. Identify sufficient characteristics for believably natural motion, and conversely, unnecessary computational effort. Draw conclusions about model selection for animation and robotics based on computational and psychovisual results. Contrast the applicability of models to computer animation and robotics in terms of motion quality, control issues and the visual impression on human subjects. Draw conclusions about the

coding complexity required of the human brain for each motion planning hypothesis.

Analyse experimental weaknesses and potential for better computational implementations of motion production theories.

1.2 Concessions

Rather than develop all the components needed for this research from scratch, a number of concessions were made by using existing software and technology. Likewise, a limited class of motions was chosen so that a thorough investigation could be performed.

The class of motions investigated was reaching targets for the right hand. This encompassed point-point reaching, as well as arbitrary hand trajectories. The whole upper body was free to move during motion, and while the specification class is thus limited the variety of motions satisfying a given specification is broad, as will be seen in Chapter 2. This choice is discussed along with other conceptual design decisions in Section 3.1.1.

The number of motion production models developed as algorithms was restricted to fourteen. This was thought to be a sufficiently broad investigation of various human motion production hypotheses, and to facilitate adequate comparison and experimental testing. Fourteen was also the largest number of models that could reasonably be investigated psychovisually, and indeed the psychovisual tests were quite time demanding for voluntary subjects (about 2 hours). Experimental design issues are discussed further in Section 5.1.1.

It was decided at the design stage that motion models should be implemented as off-line software processes. Although it is desirable to generate motion during animation or motion production because it facilitates real-time interaction, some of the motion models were definitely not possible in real-time and a consistent paradigm was required for comparison. Real-time algorithms *were* also investigated, but the first priority was to develop the

appropriate algorithms without the computational limitations of real-time generation. Design aspects of the motion production environment are discussed further in Section 3.1.1.

The majority of practical research was conducted at ATR[1] in Japan, and was supported by a wealthy research environment. A humanoid robot at ATR was used to investigate robotic motion production. This robot included a software control environment with a robot simulator. Some of the tools provided by the control and simulation software were incorporated into motion production software. The software components utilised were forward and inverse kinematics, forward and inverse dynamics and simple-Euler numerical integration. These are fundamental tools in robot control and dynamics based animation, and are discussed in Section 2.2.

The focus of this research was motion production rather than animation in general. Since visualisations of the developed motions were however essential, computer animations had to be rendered and robot motions recorded. Having generated and stored motions electronically, the robot control environment was also used to assist the performance of motions with the humanoid robot.

Besides the robot, motions were rendered using a commercial humanoid animation package: MetaCreation's Poser. This software is specifically designed for rendering high quality images and animations of humanoids. Motion files were converted into a suitable format for export to Poser, and thus rendered as `AVI` files.

1.3 Thesis structure

This thesis contributes to humanoid animation, humanoid robotics and psychophysics. It is likewise founded in dynamics based animation, robotics and psychophysics. The research was begun from a computer graphics perspective in the computer science department of Glasgow University.

[1] Appendix C gives a brief description of ATR and the considerations involved in working there.

The literature review therefore begins by contextualising dynamics based animation within the backdrop of computer graphics. Since the research background draws from multiple areas and some preparation is necessary for the detailed review of human motion production, the literature review and technical background is somewhat lengthy and some material has been deferred to the appendices. The survey is presented in Chapter 2. It begins with dynamics based animation techniques, and covers several significant factors in the development of dynamics based humanoid animation software. The must fundamental factor is dynamic simulation itself, which is subsequently reviewed. This leads the discussion to humanoid robotics, and its relationship with dynamics based animation. A technical introduction to practical issues of humanoid robotics is made by means of an overview of the humanoid robot 'DB' since it was also used in this research.

Human motion is then examined, and the the field of human motion production is introduced. The final component of Chapter 2 is a comprehensive review of human motion hypotheses, a number of which were used as inspiration for the synthetic motion production algorithms. In order to discuss the technical basis for the models and facilitate an analysis of human-like motion characteristics, a summary of empirical results about human motion is first presented. The biomechanical properties of the musculo-skeletal system are also relevant, and are also summarised. Particular attention is paid to properties of skeletal muscles, their operating characteristics and mathematical models thereof.

The completed research is presented in Chapters 3-5. These divide the research into theoretical, practical and empirical components respectively.

Chapter 3 begins by presenting the conceptual design of a motion production software environment. The synthetic motion models are then discussed along with their theoretical bases and significant characteristics. For each model a formal mathematical definition is established. Theoretical techniques for implementing the models are then expounded.

Chapter 4 presents the practical achievements of this research. The soft-

ware components of the motion production environment are described, including for example, the file format, and motion rendering techniques. Motion production algorithms are described, along with motion conversion and visualisation software. Human motion was recorded by motion capture, and this process is discussed at the hardware and software level.

Experiments are described in Chapter 5. This includes the experimental design for psychovisual testing of synthetic motions, practicalities, and results. Computational evaluations of motion software and theoretical conclusions regarding motion models are also made. Developmental experiments ratifying various design choices are also summarised.

Finally, in Chapter 6 experiments are analysed and conclusions are drawn. Criticisms regarding various aspects of this thesis are presented. The research achievements are summarised and possible directions for future research are illustrated.

There is also a compact disc accompanying this thesis which contains the source code of the various algorithms presented, `AVI` motion clips used for psychovisual testing and experimental testing software.

The remaining text is composed of Appendices. Appendix A summarises all of the formal definitions most of which are presented in the body of the thesis. This was primarily intended for easy reference. Appendix B contains a `BVH` format motion file, illustrating this file format, which is described in the main text. Appendix C describes the environment and practicalities of conducting research at ATR in Japan. Appendix D contains a number of discussions that were omitted from the main text but are nevertheless relevant, and may be seen as extensions to the reviews in Chapter 2. This includes descriptions of a number of dynamic simulation problems and possible solutions; an overview of the neural processes of muscle stimulation, spinal reflexes and the force generation characteristics of muscles; a description of motion production research apparatus, and arguments related to empirical results and theoretical models of human motion production.

1.4 Symbols & conventions

For ease of understanding, the following letter symbols have been used consistently to represent the following quantities throughout this thesis:

p	Cartesian coordinates	θ	angular quantities
α	angular accelerations	τ	torques
x	distances	l	lengths
f	forces or frequency	v	velocities
T	durations	t	time
β_0, β_1	arbitrary constants	$\mathcal{M}$, $\mathcal{P}$	compound matrices
$\mathbb{R}$	the real numbers	$\mathbb{R}^+$	the strictly positive real numbers
		$\mathbb{R}^3$	real coordinates in 3 dimensions
$\mathbb{Z}$	the integers	$\mathbb{Z}^+$	the strictly positive integers
k	knot points	$\mathbb{K}_d$	knot points with d dimensions
s	splines	$\mathbb{S}_d$	splines in d dimensions
		$\mathbb{M}_d$	motions in d dimensions

Types such as $\mathbb{K}_d$, $\mathbb{S}_d$ and $\mathbb{M}_d$ are defined formally prior to their usage. The following symbols have been used to construct mathematical and logical expressions:

$\Rightarrow$	implies	$\not\Rightarrow$	does not imply
$\Leftrightarrow$	is equivalent to	$\vee$	logical or
$\wedge$	logical and	$\mapsto$	maps to, *e.g.*, $x : \mathbb{R} \mapsto \mathbb{R}$
$\propto$	is proportional to		

$\{x_{ij}\}$	a set parameterised by i and j
$\overline{x}$	emphasises the vector/set nature of x
$[x_0, x_1]$	the closed interval of all x s.t. $x_0 \leq x \leq x_1$
(x_0, x_1)	the open interval of all x s.t. $x_0 < x < x_1$
$[x_0, x_1)$	the half-closed interval of all x s.t. $x_0 \leq x < x_1$
$\lfloor x \rfloor$	the floor of x, *i.e.*, largest integer y s.t. $y \leq x$
$\lceil x \rceil$	the ceiling of x, *i.e.*, smallest integer y s.t. $y \geq x$

$\frac{dx}{dt}$ or $\dot{x}$	first derivative of x w.r.t. t
$\frac{dx^2}{dt^2}$ or $\ddot{x}$	second derivative of x w.r.t. t
$x^{(n)}$	n^{th} derivative of x
$\frac{\partial x}{\partial t}$	partial derivative of x w.r.t. t
$\left(\frac{\partial x}{\partial t}\right)_y$	partial derivative of x w.r.t. t, also indicating that y is a free variable held fixed for differentiation
$O(f(x))$	f is an upper bound on the asymptotic time complexity of an algorithm in terms of x

In addition, a number of formal definitions are made is terms of equivalence sets, and are denoted with capitalised abbreviations, *e.g.*, $FK(\theta, p)$ denotes the forward kinematic relationship between joint angles and hand position (defined on page 28).

Chapter 2

Natural and synthetic humanoid motion

Literature survey & technical background

Synthetic humanoid motion is relevant to computer animation and humanoid robotics. The study of natural motion -the field of human motion production, provides congruent inspiration.

Within the field of computer graphics, animation researchers have sought to create realistic human actors. Dynamics based techniques have facilitated accurate animation of physically correct motions for both rigid and articulated bodies. The area of dynamics based humanoids explores the most realistic methods of generating motion for synthetic humanoids.

Robotics research has advanced to the stage where it is now possible to create fully articulated humanoid robots with human-like sensory capabilities. Such robots can mimic the motion of human beings, and respond to similar cues (verbal, visual, physical, *etc.*) as humans.

However, the task of generating realistic synthetic motion is particularly difficult because humans are capable of learning and performing an enormous range of physical skills. The biomechanical complexity of the body itself results from millions of years of evolution. A human being's potential for

motion is supported by a skeleton with over two hundred articulating links and around six hundred and seventy muscles. The range of motion includes articulation of the skeleton, deformation of the body, muscle contraction, *etc.*

The motions of the body are regulated by the central nervous system. This includes conscious control in the brain, as well as unconscious control such as balancing and reflexes that may act before the brain itself has a chance to respond. The way these processes interact while the brain plans and coordinates multiple interactions with the physical environment shapes the motions of the human body.

This chapter surveys each of these areas of research. Dynamics based humanoid animation and the underlying computational system of dynamic simulation are presented in Sections 2.1 & 2.2. Humanoid robotics is then discussed with a technical description of the humanoid robot 'DB' (used for this research) in Section 2.3. The study of human motion production is then described along with a summary of empirically deduced features of human motion in Sections 2.4 & 2.5. Human muscles, and theoretical muscle models are examined in Section 2.6. Finally, in Section 2.7 theories are presented regarding the way human motion is actually planned.

2.1 Dynamic humanoid animation

Computer graphics (CG) is a science that has developed almost from scratch over the last forty years Fley et al. (1995). Initially a highly specialist tool, it is now an ever present medium that appears on television, in films, on computers and in the gaming industry.

Dynamics based animation of human figures began in 1985 when Armstrong and Green (1985), and Wilhelms and Barsky (1985) both proposed rigid body dynamics of articulated figures for animation, drawing in part on the work of Hollerbach whose robotics based work on dynamics was established before the start of the 1980s.

The interest in generating realistically animated humanoids has been

steadily increasing. The twenty-first century has seen increasing use of computer generated special effects, fully computer generated movies, and the 2001 release of 'Final Fantasy' demonstrated consistently photo-realistic computer animated human characters. Computer entertainment has developed so that it is now comparable to the film industry. For example, "Metal Gear Solid 2", a Playstation 2 console game released in November 2001, was developed by Hideo Kojima with a budget worth ten million dollars and features a soundtrack by Hollywood composer Harry Gregson-Williams.

Now that the most advanced systems are capable of rendering photo-realistic humanoids the challenge is to automatically generate realistic human motion. Almost all of the high quality human motion used in films and games is motion captured from actors -a time consuming and awkward process incurring most of the overheads of conventional filming. Such motion is typically processed using motion editing software, adding an additional burden to the animator. Automatic motion generation can circumvent the motion capture process or facilitate higher level motion editing that automatically preserves the human-like qualities of motion. 'Final Fantasy', with its extremely realistic rendering revealed a further limitation of motion capture as the most significantly unrealistic element of its presentation. Since capture must be performed in as uncluttered environment as possible in order to maximise sensor visibility or resolution of sensor positions and many sets are designed virtually, it is in general impossible for the actors to have a good spatial awareness of their set. Likewise, the motions of large groups of people must be recorded discretely and subsequently reassembled. This leads to a sense that the characters are abstracted from each other and their environment.

Dynamics based motion generation is a key tool in generating realistic motion, since by the nature of the physical laws involved, such motion is inherently physically correct. Groups of virtual actors may also coexist in one virtual set and interact. The challenge is to find suitable methods of controlling dynamically simulated actors and mimicking the characteristics

of genuine human motion.

2.1.1 Dynamics based humanoids

The early computer simulated humanoids were mainly produced by corporate bodies investigating ergonomics [Dooley (1982), Badler (1982)]. For example, in 1969 Boeing developed Boeman, a 23 joint scalable humanoid. Boeman was used to assess reaching attempts and analyse the visual field, computer cards were necessary to specify the environment and output was generated on a graphic plotter. Boeman was used as a basis for the Car system also developed by Boeing, and used for analysing groups of humanoids. Chrysler developed Cyberman, a physically modelled 15 link humanoid, to model drivers, passengers and human activity around a car. Rockwell International in California developed Buford, a polygonal humanoid used for reach and clearance visualisation in CAD applications. Some early models were also developed by academic institutions such as the 35 link Combiman system at the University of Dayton's Aerospace Medical Research Laboratory, used for reach analysis. Bubbleman is a humanoid composed from spheres, and was developed at the University of Pennsylvania and used for reach and collision analysis -the spherical primitives facilitated easy collision tests. A 21 link polygonal humanoid, Sammie, was developed at the University of Nottingham and used for reach-space, visual field, and physical comfort analysis.

Dynamics based humanoid animation has been widely discussed in the literature and some long standing models have evolved. In particular Norman Badler's work on the Jack system [Badler et al. (1993), Ko and Badler (1996)], and Jane Wilhelms' models: Virya, Manikin and Kaya. The Jack system developed from a purely kinematic model to a dynamically simulated humanoid capable of standing, walking, manipulating objects with the hands and administrating control for complex coordinated tasks. Wilhelms' first system, Virya, was a dynamically simulated humanoid with limited control by means of torque functions for the joints of the figure. Manikin added inter-

active control by means of goal positions, or forces added into the simulation. Kaya incorporated an improved user interface, motion storage using key-frames and interchangeable dynamics formulations. McKenna, supervised by David Zeltzer, developed a humanoid model called Corpus [McKenna and Zeltzer (1996)] incorporating gravity, collisions, contact, friction, damping and joint limits (problems that are discussed in Appendix D.1). Lake and Green (1991) developed a ballroom dancing dynamic humanoid. In 1996 the space-time constraints paradigm was formulated using a fast recursive dynamics system facilitating the animation of a humanoid model with 38 degrees of freedom (DOFs) [Rose et al. (1996)]. Space-time constraints permit a number of motion constraints such as position goals, and the resultant motion is formulated as an optimisation problem using symbolic algebra solved *en masse* [Witkin and Kass (1988), Cohen (1992), Ngo and Marks (1993), Li et al. (1994), Gleicher (1997)]. Optimisation has also been used to generate control for dynamically simulated muscle actuated humanoid models using BFGS optimisation [Kōmura and Kunii (1997)], and according to the so-called sub-optimal control paradigm [Park et al. (1992)]. In fact, dynamic muscle models have also been used to analyse and modify human motion capture data [Kunii and Sun (1990)]. Highly specific motions have also been programmed for dynamic humanoids such as running, hand-spring vaulting and cycling [Hodgins et al. (1995)]. Balancing and walking tasks have also been accomplished by identifying and replicating cyclic patterns in motion data [Laszlo et al. (1996)].

There are several recurrent issues relevant to dynamic humanoids. These issues include model complexity (since highly articulated flexible models can replicate natural motion more realistically but increase the computational burden), obtaining dynamical data to realistically model the mass and mass distributions of human limbs, scaling such data in an anthropomorphically correct manner to facilitate generalisable humanoid models, maintaining balance, controlling figure posture, planning motions, generating natural motions and incorporating human-like emotional qualities.

2.1.2 Model complexity

Model complexity typically ranges from about 15-80 DOFs. The shoulder and spine are usually key factors since both are highly articulated structures that cannot be accurately modelled with a single articulation. Indeed, the body is not composed of rigid links and articulates according to complex bone and muscle interactions [Badler (1982)]. Simple models include a 19 DOF model [Rose et al. (1996)], and a cycling humanoid where the bicycle and human are combined into a single kinematic structure [Brogan et al. (1998)]. More complex models have involved 38 DOFs [Rose et al. (1996)], 77 DOFs [Isaacs and Cohen (1988)], and even a 17 link articulated spine [Ko and Badler (1996)]. Authors have also suggested merging body segments to parameterise complexity [Wilhelms et al. (1988)].

2.1.3 Anthropomorphic data

A number of sources of dynamic data have been identified. These include the weights and cross sectional muscle areas of dissected cadavers [Dempster and Gaughran (1967)], approximations from medical scans [McKenna and Zeltzer (1996)], biomechanical research [Nordin and Frankel (1989)], and various other published data [Brogan et al. (1998)]. Such data is generally applied to polyhedral limb models. A number of techniques exist for calculating inertial moments of polyhedra [Lien and Kajiya (1984), Vasilonikolidakis and Clapworthy (1991), Yang et al. (1997)] and these have been applied to limb characteristics [Hodgins et al. (1995), Lake and Green (1991)] and can likewise be applied to anthropometric scaling [Ko and Badler (1996)].

2.1.4 Balance

The first control task for a humanoid in a dynamic environment is to maintain a balanced posture. Unfortunately this is one of the most difficult problems in the field of human motion, especially when a humanoid experiences significant force interactions with its environment [Aydin and Nakajima (1999)].

Human beings have highly sophisticated methods of balancing, incorporating reflexes activated in the brain and spinal cord[1] [Roberts (1995)], pressure sensors distributed over the sole of the foot [Kavounoudias and Roll (1987)], and balance can take up to 30 seconds to recover from weak perturbations [Lauk et al. (1998)]. Computerised balance controllers have been designed for standing and tumbling [Wooten and Hodgins (1997)], using the so-called zero moment point -the point about which a falling body is rotating that transmits no torque [Ko and Badler (1996)], by simulating upside-down gravity to effectively hang a body upwards [van Overveld (1993)], using spring-damper systems [Lake and Green (1991)] and through detailed specification of supporting limbs throughout posture transitions [Girard (1991)]. The area of support, being the planar convex hull of points in contact with the ground, and the projection of the centre of mass (COM) onto the ground can be used to determine static balance -a structure is balanced if the COM projection is within the area of support. This technique has been used to analyse the stability of martial arts techniques [Shinagawa et al. (1997)], and to balance dynamic humanoids [Hodgins et al. (1995), McKenna and Zeltzer (1996)]. Iterative methods have also been used to calculate stable motions [Laszlo et al. (1996)] and balanced postures [Aydin and Nakajima (1999)], including an extended IK method incorporating static balance [Boulic et al. (1997)]. Specialised controllers have been used to balance simulated hopping robots and cyclists [Brogan et al. (1998)] and rescue reflexes for slipping and tripping [Boone and Hodgins (1997)].

2.1.5 Posture control & motion planning

Posture control and motion planning are broad problems. Posture control involves satisfying multiple kinematic targets [Aydin and Nakajima (1999)] and obstacle avoidance [Korein and Badler (1982)]. Motion planning involves the task of regulating target postures and has been tackled with finite state

[1]Spinal reflexes are faster than brain reflexes and less adaptable in general. They are discussed further in Section D.2.

machines [Laszlo et al. (1996)], and by identifying cyclic patterns in motion capture data [Stokes et al. (1999)]. Coordination for two armed manipulations involving releasing and re-grasping has also been tackled [Koga et al. (1994)]. Generalisable learning methods have been applied to observe and synthesise hand trajectories [Hale and Pollick (2000a)] (by the author), and tasks as complex as juggling have been solved using sets of action controllers with hand coded preconditions and termination conditions [Multon and Arnaldi (1999)]. Coordination in insects [Pearson (1991)] and humans [Turvey et al. (1991)] has been investigated although it has so far made little impact on animation research.

2.1.6 Natural motion & motion style

Natural motion synthesis is a particularly challenging aspect of dynamics based humanoid animation. The complexity of natural movements is due in part to the intricacy of the skeletal system and its articulations, the muscle system and its attachments, and the fact that people have an innate understanding of animal motion [Lake and Green (1991)]. Any system for generating realistic human motion is arguably a contribution to researching how to generate natural motion, and thus dynamic simulation itself is such a technique. For the case of dynamically simulated humanoids, techniques for synthesising natural motion have included minimisation of joint power (the product of muscle force and joint angular velocity) [Witkin and Kass (1988)], minimisation of torque [Cohen (1992)], filtering motions to yield smooth dynamic consistency [Yamane and Nakamura (2000)], capping the maximum muscle force in joint based movements using Hill's muscle model[2] [Kōmura et al. (1999)].

Emotional and stylistic characteristics have been analysed using Fourier techniques applied to joint angles [Bruderlin and Williams (1995), Unuma et al. (1995), Amaya et al. (1996)], by blending generalised emotional postures [Densley and Willis (1997)], and scaling kinematic parameters such

[2]Hill's muscle model is discussed in Section 2.6.1.

as velocity and movement amplitude [Badler and Chi (1999)]. High level emotional state models may be used to select an appropriate emotional state when an animator does not explicitly set it [Ho (1996), El-Nasr et al. (1999)].

2.2 Dynamic simulation

'Dynamics' is the science of matter and motion. The Newtonian model is suitable for most dynamics based animations unless atomic or stellar objects are to be modelled. Newton's model is defined by three laws:

1. A body experiencing a net force of zero remains at a constant velocity.
2. A body's acceleration is equal to the net force acting on the body divided by its mass.
3. For every action there is an equal and opposite reaction.

Realistic interactions between physical objects may be generated automatically using the differential equations that model dynamics. 'Forward dynamics' (FD) is the simulation of physical processes and requires iterative numerical integration of the dynamic equations, *i.e.*, a force accelerates a body proportionally to it mass, the acceleration is integrated to yield the body's velocity, and the velocity is integrated to yield its position. 'Inverse dynamics' (ID) is the opposite calculation whereby position functions are differentiated to find velocity and acceleration functions, and thus the corresponding force functions. ID is used in the control of robots and animation systems that mix simulation and control.

In order to simulate humanoid figures dynamically it is necessary to calculate FD for articulated figures. Unfortunately this is an extremely cumbersome computational problem requiring careful handling in a number of areas.

The basic dynamic equations of a rigid body [Wilhelms et al. (1988)] require a representation of the mass distribution of the body called the 'inertial

tensor' [Baraff (1995)] which is an integral over the volume of the body [Yang et al. (1997)]. Inertial tensors may be calculated for polyhedral bodies by reducing the dimensionality of this integral to the surface of the body [Lien and Kajiya (1984)]. The differential equations of motion may be expressed in various ways to facilitate for example, rapid calculation or aid numerical stability (*dynamics formulation*[3]). The equations require careful treatment and must be cautiously integrated using numerical methods to prevent inaccuracies precipitating invalid configurations (*numerical integration*).

The joint configuration must be represented, and there are a number of different schemes with various properties (*joint configuration*). Collisions between bodies may occur between integration time steps but must be isolated in time precisely (*collision detection*) to resolve the resultant motion (*collision resolution*). Velocities change discontinuously following collisions and also require careful handling during numerical integration.

Objects coming to rest in contact form supportive contact points that must be monitored and logged (*contact logging*) to ensure the appropriate forces are generated and objects to not inter-penetrate (*contact resolution*). Articulation points, similarly to contacts, must be monitored to ensure the connection points do not drift because of numerical inaccuracy (*joint constraints*). For both contact resolution and joint constraints the forces maintaining the correct condition must be calculated and incorporated into the simulation because simply correcting numerical drift kinematically precipitates huge forces and numerical instability.

Joints may be range limited so that for example, an elbow joint does not naturally bend backwards. Enforcing these constraints requires that the joint angle is monitored and appropriate forces generated at the limit angles (*joint limits*).

[3]Concepts introduced in italics are described further along with the range of appropriate solutions in Appendix D.1. This excludes the 'control' problem, 'inverse kinematics' and 'inverse dynamics' which are presented in the following sections.

Any of the conditions with force transmission points such as contacts and articulation points may become part of a closed loop, *i.e.* a humanoid figure with both hands together has a closed loop between hand, arm, torso, arm, and hand, similarly two blocks resting on a floor and leaning against each other form a force chain from floor to block to block to floor. Closed loop conditions are awkward to resolve because the force generated at any one force transmission point affects the forces generated at all the other force transmission points in the closed loop. The net force must maintain the correct condition without generating energy and requires specific numerical handling (*closed loops*).

Bodies sliding against each other generate friction opposing the motion. Calculating the appropriate forces requires calculation of the area of contact and the dynamics of friction are beyond exact modelling. It is generally necessary to incorporate some model of friction to prevent bodies sliding indefinitely (*friction*).

Finally, having constructed a dynamically simulated humanoid it is necessary to control the torques generated at the joints to produce animate movement (*control*). Two techniques are fundamental to the generation of control signals. Given a target position for any of the links of the humanoid, *e.g.* the hand, a set of joint angles that place the hand in this position may be calculated (*inverse kinematics*). Secondly, given a set of joint angles, angular velocities and desired angular accelerations, the required joint torques may be calculated (*inverse dynamics*).

The above issues constitute the basic minimum for convincing simulation of an articulated humanoid figure The breadth of problems indicates the complexity of simulating physical reality and depending on the application more modelling may be required. For example, a humanoid walking through wind may require exposed surface area resolution, air flow dynamics, *etc.*

2.2.1 Control

Motion control for humanoid figures involves the low level problems of achieving desired limb positions and orientations and the high level issue of coordinating complex motions [Wilhelms (1987)]. Low level control may be further divided into tightly coupled control and loosely coupled control whereby control processes are evaluated between every integration time-step of the simulator. Low level controllers examine the simulator state and determine force or torque responses for actuators. Such controllers are therefore intuitively physical but may not necessarily precipitate the desired motion. Tightly coupled control methods impose kinematic or dynamic constraints on bodies, and resolve the necessary forces automatically. Tightly coupled controllers are dependent upon computationally expensive force resolution methods.

Low level kinematic constraints have been imposed by attaching large volumeless masses to bodies thus generating forces to push or pull objects [Wilhelms and Forsey (1988)] or by blending FD and ID simultaneously in an elegant matrix solution [Isaacs and Cohen (1987)]. Human figures have been controlled by directly specifying joint torque functions or specifying the rest lengths of spring-damper systems [Wilhelms (1991)]. In fact the spring-damper system has been blended with ID to calculate spring parameters that are then linearly interpolated throughout motion [McKenna and Zeltzer (1996)]. Localised methods have been used to resolve joint torques independently according to so-called local motor programs [Zeltzer (1982)] and by distributed control [van Overveld (1993)]. 'Perception-action systems' have been designed to trigger slipping and tripping reflexes for simple articulated figures [Boone and Hodgins (1997)].

High level control for human figures has been tackled using a hierarchy of low level controllers such as limb orientation goals [Lake and Green (1991)], and by a hierarchical behaviour model based on concatenations of motion verbs [Green (1991)]. Highly specific controllers have been hand coded, such as whipping motions for n link chains [Bhat and Kearney (1996)], running, cycling and vaulting controllers [Hodgins et al. (1995)] and controllers for

tumbling, diving and standing [Wooten and Hodgins (1997)]. Such controllers are of limited application but may be generalised to an extent by morphing controllers onto different models [Hodgins and Pollard (1997)]. Motion capture has also been used to reverse engineer force control patterns [Kunii and Sun (1990)].

2.2.2 Inverse kinematics

For an articulated figure the relationship between the joint angles and link positions is in general non-linear. It is often necessary to place links (such as the hand) in specific Cartesian locations, and this requires the calculation of suitable joint angles.

Forward kinematics (FK) is the process of calculating the link positions given the joint angles. FK is a straightforward process of propagating the coordinate frames at each joint according to the appropriate translations and rotations parameterised in terms of the link lengths and joint angles.

Inverting this calculation is called inverse kinematics (IK). IK is an awkward problem. Analytical solutions are typically infeasible, and there is a redundancy issue since multiple joint configurations may yield the same Cartesian location for a given link.

Analytical solutions are desirable since they can find all solutions and are quick to calculate. Since such formulations are generally impossible numerical techniques have been adopted. Several matrix based methods have been developed as well as non-linear programming methods.

Matrix methods are generally derived from the intuitively straightforward Jacobian method [Watt and Watt (1995)]. The partial derivative of the Cartesian link position in terms of each joint angle is calculated analytically and assembled into a matrix called the Jacobian. By inverting this matrix the infinitesimal joint angle adjustments yielding an infinitesimal step towards the target link position can then be calculated. The Jacobian is dependent on the joint angles and the whole process must be iterated until the link hopefully converges on the target position.

The Jacobian matrix may be singular (usually because the redundancy precipitates a non-square matrix), necessitating the use of pseudo-inverses [Maciejewski and Klein (1985), Koga et al. (1994), Baerlocher and Boulic (1998), Aydin and Nakajima (1999)]. Pseudo-inverse methods typically incorporate an optimisation function such as a least mean squares approximation to overcome the redundancy that prevents the calculation of an exact inverse. This is advantageous because it allows the inclusion of optimisation functions such as minimum joint deviations *etc.* into the IK computation.

Tevatia and Schaal (2000) discussed pseudo-inverse methods and the so-called extended Jacobian method. By combining features of these methods they developed the 'improved extended Jacobian method' which is more computationally efficient than the extended Jacobian method. This IK method was used in this research. The extended Jacobian method is based on fixing the gradient of the optimisation function in the null space of the Jacobian and tracking this optimal solution. The null space is the vector space of configurations with the same solution (link position).

Finally, Zhao and Badler (1994) presented a non-linear programming method for solving IK. The technique was partially based on the BFGS quasi-Newton optimisation method. Such approaches treat IK as an optimisation problem over the space of joint angle configurations.

Since formal definitions of motions are presented in Chapter 3 it is useful to make a formal forward/inverse kinematic constraint. In stating this, and further definitions, a humanoid figure with $NDOFS \in \mathbb{Z}$ degrees of freedom is assumed. It is further assumed that all kinematic relationships will be between the joint angles and the placement of the right hand unless explicitly specified.

Definition 1 (Forward and inverse kinematic hand constraint)

$FK(\theta, p) \quad \Leftrightarrow$

$\theta \in \mathbb{R}^{NDOFS}, \quad p \in \mathbb{R}^3$

When the joint angles are θ the right hand position is p.

This is a many to one relationship since $FK(\theta, p) \wedge FK(\theta', p) \nRightarrow \theta = \theta'$ *but* $FK(\theta, p) \wedge FK(\theta, p') \Rightarrow p = p'$.

2.2.3 Inverse dynamics

ID is the process of calculating the forces that bring about a given motion. It is used to generate smooth control for articulated figures, and to satisfy kinematic and articulation constraints. The equations are summarised by Newton's second law $F = ma$, as are those of FD, but F is the unknown rather than a. The equations become more intricate for articulated figures when force, mass and acceleration are generalised to include their rotational equivalents torque, moment of inertia and angular acceleration respectively.

The most basic formulation requires $O(n^4)$ time for an n link articulation but can be expressed in such a way that velocity values are propagated through the linkages of an articulation from root to tips, and the forces then calculated in a reverse propagation from tips to root according to the so-called Newton-Euler recursive method [Girard (1991)]. This method is $O(n)$, as is the recursive Lagrangian method -an alternative that may be implemented using Cartesian tensors to improve the speed by a linear factor [Vasilonikolidakis and Clapworthy (1991), Rose et al. (1996)]. A limited solution to ID has also been formulated using a cost minimisation strategy [Breen (1997)].

The forward and inverse dynamics constraints may be formalised simultaneously as follows:

Definition 2 (Forward and inverse dynamics constraint)

$FD(\theta, \dot{\theta}, \tau, \alpha) \Leftrightarrow$

$\theta, \dot{\theta}, \tau, \alpha \in \mathbb{R}^{NDOFS}$

When the joint angles and velocities are θ and $\dot{\theta}$, if the joint torques are τ the resultant accelerations are α or equivalently, the accelerations α necessitate torques τ.

In a given posture, with given joint velocities, there is a one to one relationship between the joint torques and joint accelerations[4].

2.3 Humanoid robotics and 'DB'

Robots have existed for many decades and much research has already been completed. The image of the robot spray painting cars is a well established. In science fiction the humanoid robot is also a well established concept dating back in film as early as Fritz Lang's Metropolis (1927). The reality of humanoid robots capable of full articulation and human-like sensing however, has not been achieved until very recently.

Several humanoid robots have been manufactured in the last few years. For example, the Kawato Dynamic Brain Project developed the $80Kg$ 30 DOF robot DB, used in this research and described in Section 2.3.1. Honda's latest humanoid robot ASIMO weighs $43Kg$ and is capable of walking and climbing stairs. The Kawada Industrial Corporation and Tokyo University developed HRP-2P, a $58Kg$ robot with 30 DOFs capable of walking outside, and the Sony Dream Robot, SDR-3X weighing just $5Kg$ can walk, balance on one leg, dance, perceive coloured objects and respond to a small audio vocabulary.

Robotic humanoids are still extremely expensive but it is not unreasonable to expect that within the next 10 to 20 years they may play a significant

[4]This relationship may be expressed mathematically, but the equations are particularly involved and it was considered unnecessary to duplicate them here.

role in our daily lives. An important aspect of humanoid robots is that humans themselves be able to interact with them naturally. The robot must therefore move and respond in the same way as a human being, performing and responding to the same social cues. This problem is broad, encompassing for example, the psychology of communication, artificial intelligence and motion production. Automatic motion production has already received considerable research attention under the more popular heading: computer animation.

There exists an overlap between the fields of robotics and animation, and in particular, robot humanoids and dynamically simulated virtual actors. Under dynamic simulation the task of motion production involves calculating the forces and torques needed to bring about a required motion. Simulation guarantees physical consistency. Robots on the other hand are implicitly correct physically, being physical entities. The control problem is equivalent, *i.e.*, forces and torques must be calculated.

Contrasting animation and robotics, Wilhelms (1987) observed that:

> *In one sense, the animation problem is more difficult, as most robots are far simpler than the articulated bodies shown in computer graphics. ... But in another sense, the animation problem is simpler because only the appearance of realism is needed, not the actual physical reality.*

In general it is reasonable to expect that simulated humanoids may include more DOFs than the average of 20-40 used by most humanoid robots. Closed kinematic chains are handled implicitly by robots but care must be taken to ensure a robot does not damage itself. Collisions and contacts are also resolved automatically but care must likewise be taken to avoid damage [Yamane and Nakamura (2000)]. It is possible to interact physically with a robot [Hale and Pollick (2000a)] (research by the author). Virtual humans on the other hand, cannot without the aid of special force interaction mechanisms make physical contact with a human (though such devices further blur the boundary between virtual humanoids and humanoid robots).

Besides these comparisons, the most significant practical difference is that working with simulated humanoids requires a fast computer, whereas working with robotic humanoids requires a robotic humanoid. As a doctoral student, this equates to fortunate placement in a well funded laboratory with a team of research engineers maintaining the various component robotic systems.

2.3.1 Overview of DB

'DB' belongs to the Kawato Dynamic Brain project at the Human Information Science laboratories, ATR, Japan. Researchers working at ATR's CyberHuman project collaborated with SARCOS, a successful robot engineering company, to design DB, which was constructed by SARCOS at their facility in Salt Lake City, Utah.

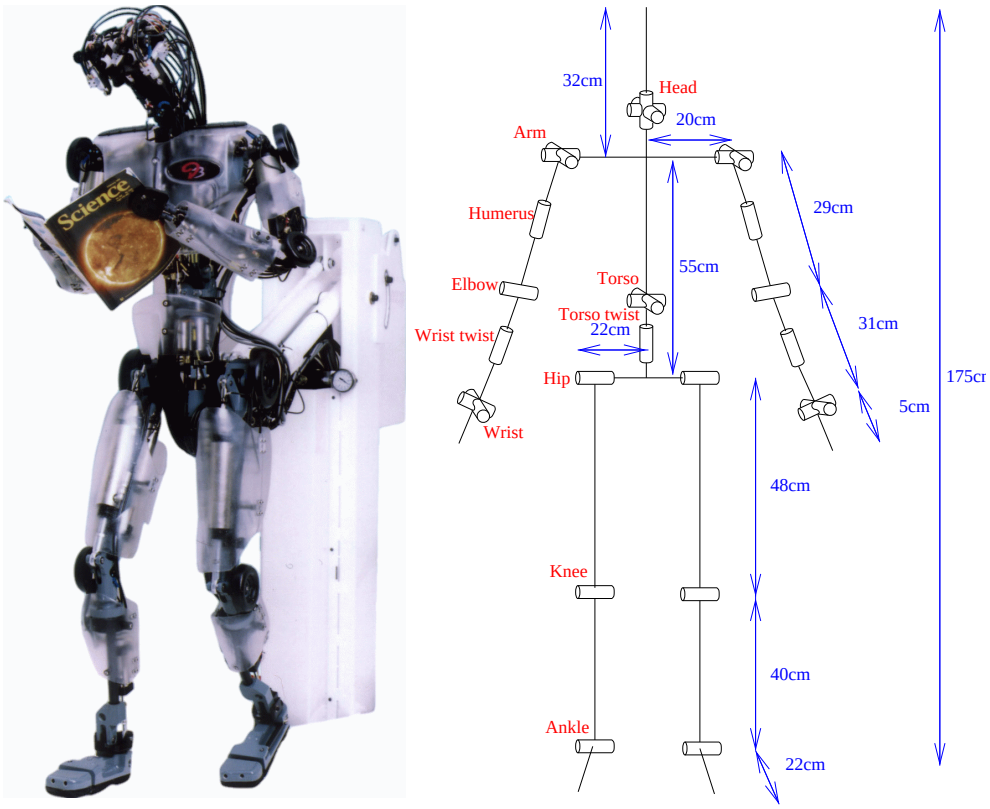

Figure 2.1: The humanoid robot DB

DB has 30 degrees of freedom as shown in Figure 2.1. There are an additional 8 DOFs that must be locked off-line. The locked DOFs set the shoulder shrug, hip abduction, ankle twist and ankle abduction. The head unit has four mobile cameras which account for the four DOFs not shown in the figure. Two of the cameras have wide angle lenses so that the robot may have a wide angle view but also fixate on a target. DB weighs $80Kg$ and stands 1.85 metres tall. The waist is attached to a mechanical support so that DB does not in fact stand, but hang from the waist. This circumvents the awkward problem of balancing, and facilitates the use of a non-portable hydraulic compressor to actuate DB's joints. Since DB is hydraulic it can generate motion compliantly so that physical interactions with humans are possible. Typically, it has been too dangerous to perform human/robot physical interactions with human-size robots because the rigidity and strength of robot motions places humans at risk. Besides the hydraulic actuators DB's motor system is equipped with torque and joint angle sensors.

The research accomplished by other Cyberhuman project team members has examined various aspects of humanoid robotics. Darrin Bentivegna devised a 'learning from observation' scheme and applied it to air hockey, allowing a human to play the table-top game with DB [Bentivegna and Atkeson (2000)]. Marcia Riley developed a catching system so that DB could respond in real-time to thrown objects and catch them [Riley and Atkeson (2002)]. Ales Ude has focused on vision processing applied to face location, human motion analysis and motion capture [Ude et al. (2000)]. This represents only a fraction of the research completed, and a general overview of recent achievements may be found in Atkeson et al. (2000).

2.3.2 DB's Control processes

Controlling DB's motion is a hierarchical process. Low level control ensures that target postures are attained in a stable fashion, while high level control tools assist the generation of postures and satisfaction of motion goals. All these features are available through the control interface which is discussed

in the next section.

The low level control process must constantly monitor the joint state and regulate the actuator forces appropriately. The fundamental component is a feed back spring-damper system. According to this process the torque τ generated at a joint is proportional to the difference between its target angle θ_T and actual angle θ, and negatively proportional to its velocity $\frac{d\theta}{dt}$.

$$\tau = \beta_0(\theta_T - \theta) - \beta_1 \frac{d\theta}{dt} \tag{2.1}$$

β_0 is the spring stiffness coefficient. As β_0 increases, the corrective forces increase. β_1 is the damping coefficient, without this component θ typically oscillates back and forth around the target position. The damping component acts against the joint velocity smoothing out the oscillations so the joint comes to rest near the target. Because the robot is affected by gravity the spring-damper system alone typically comes to rest at an angle offset from the target, where gravity balances the spring force.

Two solutions were available to satisfy exact joint placement. The first, integral gains, maintains an integral torque so that while a joint is offset from its target angle, the integral torque steadily increases until the joint reaches the target. This can be dangerous since if the robot's motion is impeded the integral force can build up until the obstruction is forced aside. The integral torque must therefore be strictly capped within safe limits. Alternatively ID may be used to calculate a feed forward torque term added to the feed back term, intended to exactly compensate for the gravity force.

IK and ID constitute DB's significant high level control features. IK may be used on-line to compute the joint angles satisfying link placement goals. The improved extended Jacobian IK algorithm was fast enough to iterate on-line, although convergence to the optimal posture satisfying a given link placement typically required several seconds. ID may be used to facilitate exact link positioning in the presence of gravity, and also to smooth motions and encourage rapid oscillation free motion. The recursive Newton-Euler ID method was available.

2.3.3 DB's Software environment

DB is supported by software programmed by Stefan Schaal. The software neatly encapsulates all the low and high level control processes into an integrated environment. It is possible to write C code for controlling DB using these libraries.

The control code is applicable to the robot itself, and equally, to a robot simulator used for testing. Since the robot is an expensive and sophisticated piece of hardware, it is extremely undesirable to damage it. All control software was therefore tested under simulation. Programming mistakes that could precipitate unpredictable motions dangerous to humans or the robot itself were thus resolved prior to physical testing.

Control interface

The DB control interface is at the programming language level. In order to operate the robot it is necessary to write a task specification in C. This specification must contain an initialisation function, a runtime function, and a few less significant functions.

The initialisation function must check that no other tasks are running, prepare the robot in a safe posture and prepare any variable space required by the runtime function. The initialisation function may temporarily allocate control to another task, moving to a default posture for example.

The runtime function is subject to real time requirements. It is called at $420Hz$ regardless of whether its previous execution has completed. A number of state variables are updated between task cycles and these are available to the initialisation and runtime functions. Likewise, a number of control variables may be set by these functions. The state variables contain sensor readings indicating the joint angles of the robot, the sensed torques, the feed back and feed forward force terms. The control variables set the desired angles, velocities and accelerations and any feed forward forces.

It is thus possible to fix the target joint angles exactly. It is dangerous however, to set joint angle targets more than a few degrees away from the

current posture because the forces and velocities generated can become very large. Instead, motions must be deconstructed into a series of small steps to ensure that movement is smooth, and performed at a reasonable rate.

The IK function may assist this process, converting a Cartesian hand target into a set of suitable joint angles. Usually, a Cartesian hand trajectory is itself sampled with a series of small Cartesian offsets and IK is performed on-line to meet these targets.

The ID function may be applied to a set of target joint angles, velocities and accelerations. The resultant torques can then be supplied as feed forward forces to the robot's actuators.

Due to the real-time requirements it is generally unsafe to print information to the control environment during task execution. Printing can be held up by operating system requirements and can cause unacceptable delays in the runtime function. Motion analysis and reporting must therefore be carried out off-line, using motion logs stored in the memory during task execution and subsequently dumped during non-critical execution time.

Simulation environment

The robot simulator is an extension of the control environment itself. The control environment necessitates a dynamic model of the robot, including the kinematics (link lengths and articulation topology) and dynamics (link masses, inertial properties and actuator strengths). These models are integral to the low level control system, and also required by the high level control processes of IK and ID. The control system likewise contains a FK model to facilitate the calculation of link positions based on joint angle sensors. Since the kinematic and dynamic models are such an integral part of the control software, there is considerable overlap between control and simulation code.

The simulator is based on a FD process as described above. FD iterations are integrated using the simple Euler method. Since joint task code should avoid joint angle violations, no explicit solution is included in the simulator. Joint angle violations are merely reported so that the programmer may

correct their task code. Collision detection is also not implemented. Since it requires considerable processing, and the exact kinematics of DB's outer plastic shell are subject to modification collision detection is left to the human eye. By watching the computer animation of motions under simulation it is possible to detect collisions, and indeed near collisions or unexpected behaviours.

2.4 Human motion production

The study of human motion production concerns itself with the question 'How do we, as human beings, move our bodies?'. Closely related topics include biomechanics, neurophysiology, and psychophysics. In their relation to human motion production these topics may be summarised as follows:

- Biomechanics - the physics of the musculo-skeletal system with respect to it's environment
- Neurophysiology - the neural systems and processes governing brain function and muscle control
- Psychophysics - how the brain interprets and responds to sensory input regarding the physical world

Goals of human motion production research include discovering how motions are planned and what processes modulate motion as it occurs. These questions are extremely deep and broad ranging. Factors influencing motion production make it difficult to gather generalisable empirical data, and some questions probe far into the mental control processes of movement. For example, the way motions are planned depends on the context of motion *e.g.* precise movements such as threading a needle, powerful movements, well-timed movements, repetitive movements, *etc.* The human body is capable of a huge range of motions that may require totally different control processes such as writing, pointing, punching, running, balancing, and lifting.

Some deeper questions include whether or not motions are planned purely spatially or with respect to dynamic loads (discussed further below), and whether planning is made in some representation of the external world, or in the internal space of joint angles or muscle lengths (also discussed below). Furthermore, the actual computations are complicated. Determining the necessary joint angles to place the hand at a given Cartesian position (IK) for example, and determining the muscle tensions required to maintain the joints of the body at given angles, velocities and accelerations (ID) are complex problems, as discussed in Sections 2.2.2 & 2.2.3.

Typical experimental apparatus for motor control includes the planar manipulandum, three dimensional motion capture and the nuclear magnetic resonance imager (MRI). The planar manipulandum is a mechanism by which forces controlling the hand may be investigated, and is described in Appendix D.3.1. Motion capture is discussed in depth in Section 4.5. The MRI scanner is used to determine which areas of the brain are active during motor tasks. The density of oxygen in the brain may be revealed by its nuclear density, and the rate of metabolism at a given site and hence brain activity thus determined.

The following sections present a review of research in three areas: features of human motion, muscles, and motion production hypotheses. The first deals with various empirical observations about human motion without seeking to explain them. These features are significant in understanding both the subtleties of motion and the accuracy of motion production hypotheses. Likewise, an appreciation of the complexity of muscles, and the level of simulation necessary to replicate features significant to perceivable limb motion is relevant to the comparison of motion production hypotheses. Finally, motion production hypotheses are discussed in terms of the motion features they replicate, and their standing among motion production research.

2.5 Features of human motion

Motion capture and manipulandum experiments have revealed empirically a number of specific features that human motion often displays. These features have been categorised and analysed in the literature, and there are wide ranging arguments about whether various features are specifically planned by the brain or result trivially from interactions with the physical world. In this section the various findings are reported.

2.5.1 Curvature of point-point motions

Experiments involving people moving their hands along point-point trajectories have revealed that the trajectory of the hand is in general straight, but often displays a gentle curvature [Abend et al. (1982), Wann et al. (1988)]. While an exactly straight path from start to end point would seem unlikely, the observed curvature is consistent among trials and across subjects. The nature of this curvature is also dependent upon the region of workspace in which the hand moves.

For example, in the horizontal plane at shoulder level transverse movements are curved outwardly while radial movements are relatively straight [Haggard and Richardson (1996), Uno et al. (1989)]. In the vertical plane up and down movements are curved outwards whereas back and forth movements are relatively straight [Atkeson and Hollerbach (1985)]. Numerous other studies have confirmed the consistency of the curvature in point-point reaching movements.

2.5.2 Bell shaped velocity profiles

Point-point reaching movements in general display bell shaped velocity profiles. That is to say, the trajectory of the hand accelerates gradually from rest, peaks at the midpoint of displacement, and decelerates smoothly to rest. This basic property has been observed by many authors [Morasso and Mussa-Ivaldi (1982), Flash and Hogan (1985), Hoff and Arbib (1993), Wolpert et al.

(1994)].

2.5.3 Multi-peaked and skewed velocity profiles

Morasso and Mussa-Ivaldi (1982) investigated the velocity profile of the hand in reaching movements and revealed a number of departures from the bell shaped profile. When the limbs are relaxed it has been observed that the velocity profile is skewed towards the onset of motion [Atkeson and Hollerbach (1985), Wing and Miller (1984)]. For motions where the goal trajectory is curved, such as point-point movements around an obstacle, the velocity profile may be double-peaked as the hand decelerates in the region of the obstacle or midpoint of the trajectory. Skewed single peaked velocity profiles have also been observed, so that one portion of the displacement is accomplished more quickly. Such movements generally take longer than point-point movements. Closed (looping) trajectories display two velocity peaks when the path is long and narrow, and have four or five less distinguishable peaks for rounded paths.

2.5.4 Smoothness

The trajectory of the hand in motion is smooth. In fact the physical environment constrains the hand to move in the mathematical sense of the word 'smooth', *i.e.* all derivatives exist and are continuous. However, the velocity tends to vary gradually, as does the acceleration and indeed third derivative, jerk [Wolpert et al. (1994), Todorov and Jordan (1998), Sejnowski (1998)]. Some discussion regarding the origin of this smoothness may be found in Appendix D.3.2.

2.5.5 Piecewise planarity

Soechting and Terzuolo (1987a,b) recorded three dimensional wrist movements, concluding that wrist motion is segmented into piecewise planar components. According to their theory, non-planar motions are possible but are

planned as concatenated planar segments. Planar segmentation has been disputed by Todorov and Jordan (1998) who claimed that "work on segmentation has focused on repetitive movements of long duration, which implied (at least visually) a rather obvious segmentation pattern". This objection reflects the *context* problem in human motion studies, *i.e.* the context of motion may have a significant effect on the nature of movements generated and motion capture is time consuming and restrictive. Nevertheless, the property has at least been observed in certain circumstances.

2.5.6 Isochrony

Isochrony refers to the tendency for people to perform similar movements in similar times [Viviani and Terzuolo (1982), Lacquaniti et al. (1983), Viviani and Schneider (1991)]. As early as 1893 it was observed that the average velocity of point-point movements increases with the distance between the points [Derwort (1938), Binet and Couriet (1893)]. This is now described as 'global isochrony' to distinguish it from 'local isochrony' according to which velocity is modulated with respect to the units of motor action. An example of local isochrony occurs when people describe a figure of eight with one loop larger than the other. They tend to increase the velocity for the larger loop to make the times for the two loops equal [Viviani and Flash (1995), Wolpert (1997)].

2.5.7 Isongany

Similar to the isochrony principle, isogany refers to the angular traversal of a movement. Viviani and Terzuolo (1982) observed that arcs over equal angles are performed in similar times even though the arc length itself may vary.

2.5.8 $\frac{1}{3}$ power law

Lacquaniti et al. (1983) proposed a concise mathematical description governing hand movements. The law incorporates smoothness, isochrony and

isogany to an extent and is formulated in the language of differential geometry. It states that for human hand trajectories, the magnitude of the tangential velocity is inversely proportional to the $\frac{1}{3}$ power of curvature[5].

In order to express the law formally let us begin with a space-line:

$$p : \mathbb{R} \mapsto \mathbb{R}^3 \tag{2.2}$$

The position at time $t \in \mathbb{R}$ is $p(t)$. The unit tangent vector u may be defined as follows:

$$u(t) = \frac{\dot{p}(t)}{|\dot{p}(t)|} \tag{2.3}$$

The curvature κ is the planar deviation from a straight line. It reflects the proportion of the velocity away from the immediate direction of motion and is defined according to the Serret-Frenet formulae as follows:

$$\kappa = \frac{|\frac{\partial u}{\partial t}|}{|\frac{\partial p}{\partial t}|} \tag{2.4}$$

$$= \frac{|\dot{u}|}{|\dot{p}|} \tag{2.5}$$

$$= \frac{|\dot{p}|^2|\ddot{p}|^2 - (\dot{p}.\ddot{p})^2}{|\dot{p}|^6} \tag{2.6}$$

We may now express the $\frac{1}{3}$ power law concisely:

$$|\dot{p}| \propto \frac{1}{\kappa^{\frac{1}{3}}} \tag{2.7}$$

The $\frac{1}{3}$ power law is discussed critically in Appendix D.3.3.

2.5.9 Arm-forearm phase law

The arm-forearm phase law observed by Soechting and Terzuolo (1986) applies to planar motions and states that the arm and forearm angular elevation are 180° out of phase. They demonstrated empirical coherence for planes of motion up to 45° off from vertical. They observed consistent distortions in

[5]Some authors prefer the term '$\frac{2}{3}$ power law', under a slightly different but equivalent formulation.

wrist motions when subjects were asked to draw circles and ellipses, such as a flattening of the portion of trajectory closer to the subject when drawing a circle in the sagittal plane. By modelling the shoulder as a 3 DOF point articulation, and the elbow as a 1 DOF hinge joint and utilising the phase law they were able to reproduce these distortions in simulation. Their simulation used the $\frac{1}{3}$ power law in trajectory generation.

2.5.10 Corrective hooks

It has been observed in many experiments involving point-point reaching and movements to a target, that for fast movements a corrective motion is generally included [Jordan et al. (1994)]. The trajectory generally accelerates rapidly towards the target, comes close to rest at some point near the target and a secondary motion converges on the target point. 'Corrective hooks' are often observed as the hand initially overshoots and returns to the target, and have been cited as qualities indicating the realism of certain motion production hypotheses [Flash and Hogan (1985), Osu et al. (1997)].

2.6 Human muscles

Muscles are the organs of motion in the human body. Striated muscle tissue consists of many aligned muscle fibres capable of contracting in the same direction. The muscles connecting the bones together and facilitating voluntary movement are striated, and are often called *skeletal* muscle or *red* muscle to distinguish them from *white* muscle. White or *smooth* muscle tissue consists of muscle fibres arranged in random directions and is capable of contracting as a unit. Smooth muscles occur inside the body, forming structures such as sphincters.

Skeletal muscles are generally arranged in opposition, so that each joint is controlled by a flexor/extensor pair. The flexor muscle reduces the angle of the joint while the extensor increases it. Many joints have several cooperating muscles, and some muscles attach across two joints. Using the basic

flexor/extensor pair it is possible to flex a joint, extend it, stiffen it at a given angle and relax it, as shown in Figure 2.2.

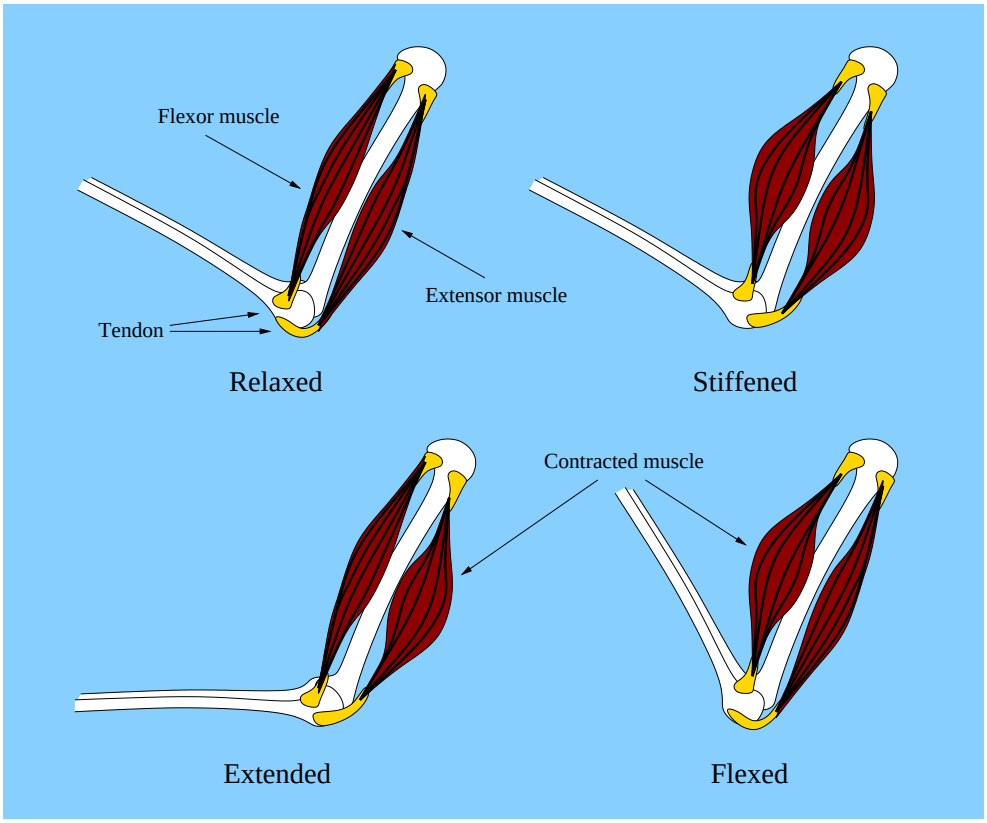

Figure 2.2: Flexor/extensor muscle pair

2.6.1 Muscle models

Human muscles vary in strength from about 30 to 100 Newtons per square centimetre of tissue. Their contraction times can range from $7.5ms$ to $100ms$, and the torque generated at a joint may be affected by the relative muscle activation level, the muscle length, the angle of the joint, the muscle contraction velocity, the temperature of the muscle, the level of fatigue, and the

visco-elastic properties of the tissue itself[6]

Given the complexity of muscle force generation it is impossible to accurately model all the properties of a muscle. Mathematical models range from spring systems [Shadmehr and Arbib (1992), McKenna and Zeltzer (1996)] at the most simplistic level, to second order differential models incorporating neural activation levels [Winters and Stark (1987)] and high order nonlinear systems with up to fifty parameters [Hatze (1978)]. Hill's three element model is a long standing and popular model that incorporates detailed muscle properties and is conceptually straightforward [Hill (1938)].

The simplistic linear spring model has a spring constant β_0 and a target length l. At a given length x the force generated is:

$$f = \beta_0(x - l) \qquad x > l \tag{2.8}$$

$$= 0 \qquad x \leq l \tag{2.9}$$

This model is of use in systems with many muscles, or where large scale optimisation is involved such as facial animation. However, the model does not capture the exponential characteristics due to the stretch reflex, and exponential springs were thus proposed. The exponential spring incorporates an index parameter β_1 which does not generally vary.

$$f = \beta_0 e^{\beta_1(x-l)} \qquad x > l \tag{2.10}$$

$$= 0 \qquad x \leq l \tag{2.11}$$

The force generated in these models may be regulated by changing the target length l, or the spring coefficient β_0. Both have been proposed, but Shadmehr and Arbib (1992) argue that the rest length should be varied based on the observation that the force/stiffness relationship of muscles is a non-linear function that does not vary with length. Variation of the spring coefficient is therefore not consistent with empirical findings.

Hill's model explicitly incorporates the complex elastic properties of muscles. Sometimes referred to as the 'three-element' model, it has the structure

[6] Appendix D.2 discusses both the neural pathways and feedback loops used to regulate muscle contraction and the characteristics of muscle force generation.

shown in Figure 2.3. The serial elastic element simulates the stretch that occurs in the tendons, and the parallel elastic element simulates the forces generated by connective tissues surrounding muscle fibres. The contractile element generates force and incorporates a model of the relationship between force generation and contractile velocity:

$$(f + \beta_0 f_0)(v + \beta_0 v_m) = \beta_0 f_0 v_m (1 + \beta_0) \tag{2.12}$$

f is the muscle force, v is contractile velocity, β_0 is a non-varying shape parameter, v_m is the maximum unloaded contractile velocity and f_0 is the maximum isometric force generation (muscle tension without change of length).

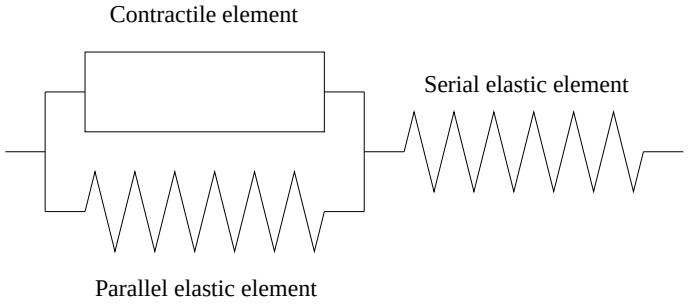

Figure 2.3: Hill three-element muscle model

Hill models have been widely used and developed. For example, an extended Hill muscle incorporated muscle pennation angle so that the angle of a joint affects the torque generation properties of the attached muscles, and the maximum force and moments of a joint in a given position may be calculated using linear systems solved with the simplex method [Kōmura et al. (1999)]. Hill's model was also used at the heart of an accurate single muscle simulation incorporating the actual shape of the muscle and its stress and strain properties. The shape and internal properties were modelled using an iterative finite element model [Kojic et al. (1998)].

Second order models were proposed in order reduce the complexity of solving models such as Hill's. The serial elastic element in particular makes

the Hill model difficult to solve. Some second order models incorporate a serial elastic element, but those that do not are represented by the following equation:

$$l\frac{d^2x}{dt^2} + m\frac{dx}{dt} + gx = hu \tag{2.13}$$

$x : \mathbb{R} \mapsto \mathbb{R}$ is a function representing joint angle or torque, and $u : \mathbb{R} \mapsto \mathbb{R}$ represents either neural activation or external loading. l, m, g and h typically represent inertia, viscosity, elasticity and gain respectively. While such models are relatively easy to solve, treating the muscle/joint system as a black box process, they are limited. The model only has one input so that secondary inputs such as muscle co-activation levels must be represented artificially by parameter manipulation, and the parameter values themselves change as a function of task and even for different operating ranges of the same task.

Huxley models are an extremely accurate and complicated description of muscle action. They represent the bonding (contraction) and unbonding (extension) rates of actin and myosin molecules in individual muscle fibres using partial differential equations with probability distributions:

$$\left(\frac{\partial n}{\partial t}\right)_x - v(t)\left(\frac{\partial n}{\partial x}\right)_t = g(x) - (g(x) + h(x))n(x,t) \tag{2.14}$$

n is a probability distribution representing the ratio of molecular bonding. The independent variables are time t, and length x. g and h represent the bonding and unbonding rate parameters. v represents the rate of contraction of an actin/myosin pair. The rate parameters are typically piecewise linear functions of x. This equation is at the heart of Huxley type models and once $n(x,t)$ has been calculated it may be used to calculate more useful properties such as the force generation in terms of activation level and cross sectional muscle area. Huxley type models explain the full range of lengthening muscle phenomena whereas even the Hill models cannot describe all aspects of the transient lengthening response seen in isolated muscle studies. Unfortunately however, Huxley models are too mathematically complex to be used in any practical problem involving human movement.

2.7 Motion production hypotheses

In this section research regarding how the brain transforms motion goals into trajectories and motor commands is reviewed. Before embarking on a survey however it is appropriate to make clear certain terminology.

The terms intrinsic/extrinsic clarify the space in which planning occurs. *e.g.* joint angle space is *intrinsic* to the humanoid, whereas the external three dimensional environment is *extrinsic*. A model that plans movements in terms of postures of the human body and paths in joint space is thus intrinsic, and a model that plans movements in terms of three dimensional trajectories in the real world is extrinsic. Extrinsic planning typically assumes that some on-line process or trivial mapping calculates the joint angles corresponding to specific Cartesian goals.

The terms kinematic/dynamic (in this context) refer again to the space in which planning occurs. Kinematic space incorporates position and it's derivatives, but does not include mass. Dynamic space incorporates mass and force. A model that plans movement without regard to the muscle activation or forces required to bring about the movement is thus kinematic, while a model that plans movement in terms of joint torques or muscle forces is dynamic. Kinematic planning typically assumes that some on-line feedback process takes care of the appropriate force generation.

2.7.1 Simple analytical optimality constraints

Among the simplest of motion theories are those examined by Nelson (1983). Nelson investigated extrinsic kinematic motion production, considering Cartesian point-point trajectory planning with respect to certain optimality constraints. Constraints included movement time, distance, peak velocity, kinematic energy, peak acceleration and jerk. Minimum velocity yields a piecewise constant velocity profile with an infinite acceleration profile. Motion is a linear interpolation between start and target points. Constraining the acceleration yields a pyramid shaped velocity profile with a piecewise con-

stant acceleration profile. The trajectory thus accelerates from zero velocity up to a maximum at a point halfway between start and target points, and decelerates symmetrically.

2.7.2 Minimum jerk

Beyond this purely mathematical approach and based on empirical evidence it had been noted by many authors that humans often produce smooth motions with bell shaped velocity profiles. This is natural given the mechanical requirement to accelerate and decelerate the limbs with continuous velocities that do no damage the musculo-skeletal system. By considering extrinsic kinematic trajectory planning, it has been proposed that trajectories with minimum of jerk (the derivative of acceleration) closely match human motion [Flash (1983), Hogan (1984), Flash and Hogan (1985)].

It is suggested that such motion eases the mechanical burden on the musculo-skeletal system and is in general consistent with torque change minimisation [Morasso (1981), Abend et al. (1982)], as discussed below. It also yields the isochrony principle according to which segmented components of human motions are performed in similar times and the $\frac{1}{3}$ power law that expresses a direct relationship between the velocity and the curvature of hand trajectories [Viviani and Flash (1995)]. Since motions are planned with respect to target points in Cartesian space, trajectories are workspace independent and point-point trajectories are straight. Hoff and Arbib (1993) formulated minimum jerk trajectory generation using an on-line feedback rule, demonstrating that it can also be accomplished without explicit planning.

One particular complaint against the minimum jerk model is that being extrinsic, point-point motions always yield straight line paths unrelated to the workspace region. Many explanations for this curvature assert that planning occurs as a straight line in some non-Euclidean space. *e.g.* joint angle space, kinaesthetic space and perceptually distorted visual space.

2.7.3 Straightness in joint angle space

It was observed that two joint movements in the vertical plan tend to use trajectories that are straight in joint-angle space [Soechting and Lacquaniti (1981)], and that drawing movements reveal angular covariation [Soechting et al. (1986)]. These observations lead to the hypothesis that motions are planned as straight lines in intrinsic joint angle space.

It has also been observed however, that a general kinematic property of two link mechanisms is the tendency of the joint angle ratios to approach a constant value at the boundary of the workspace [Hollerbach and Atkeson (1986)]. It was further asserted that a straight trajectory in joint angle space yields a complex end-point trajectory in Cartesian coordinates and cannot account for the roughly straight hand trajectories in front of the body [Hollerbach (1990)].

2.7.4 Minimum kinaesthetic jerk

The kinaesthetic model [Wann et al. (1988)] is a visco-elastic minimum jerk model. Instantaneous stretch is determined by the spring and damping characteristics of limb tissue itself, and reflects the internal deformation of the mass distribution. The optimal trajectory minimises jerk over a sum of instantaneous stretch and the position of the centre of mass. It may be regarded as a minimisation of both internal and external jerk. Since the internal deformation depends on limb mass and acceleration the model is dynamic. This model yields skewed bell shaped velocity profiles sometimes observed in human motions. Deviations from the $\frac{1}{3}$ power law due to velocity skewing towards the onset of movement observed in empirical data are replicated.

2.7.5 Straightness in visual space

Wolpert et al. (1994) carried out both visual experiments to determine perceived curvature, and motion experiments to measure the curvature of hand movements. Trajectories were displayed to subjects who were asked to clas-

sify the curvature as concave or convex. Their results confirmed a consistent trend also observed by Foley (1980) and referred to as the 'Apparent fronto-parallel plane': at far distances convex curves appear flat, and at near distances concave curves appear flat. They established a correlation with the curvature of hand motions and concluded that motions may be planned in visually distorted extrinsic kinematic space.

2.7.6 Minimum jerk virtual trajectory control

The minimum jerk virtual trajectory control (MJVT) hypothesis [Flash (1987)] addresses the workspace and curvature issues of the minimum jerk model by suggesting that motor planning occurs in extrinsic space while actual motion is brought about by PD-servo based control in a dynamic environment. The 'virtual trajectory' concept supposes that a *target* trajectory is used as a guide for actual motion production. A PD-servo is a spring and damper system where the acceleration is a linear combination of the displacement between actual and target positions, and the current velocity. In the MJVT model the hand virtual trajectory is derived according to minimum jerk, but motion occurs by generating forces linearly proportional to the offset between the actual hand position and the ideal position, and negatively proportional to the velocity of the hand. In reality, the configuration of the muscles and bones is thus supposed to yield the subtly curved trajectories, and the PD-servo control yields inaccuracies and corrective 'hooks' in trajectories characteristic of fast human motions. Simultaneously, the model is computationally less demanding than the comparable kinaesthetic minimum jerk model since motions are planned entirely extrinsically.

2.7.7 Minimum snap

The minimisation of snap -the derivative of jerk [Flash (1983)] yields trajectories with smooth acceleration, jerk *and* minimal snap profiles. It is unclear however, whether humans are able to perceive snap directly and Flash con-

cluded that nothing is gained over the minimum jerk model in the context of human motion production, and it is sufficient to minimise only the third derivative.

2.7.8 Minimum torque

While minimal derivatives may be consistent in many cases with human movements it does not follow that humans calculate and minimise these functions. For example, minimal acceleration yields minimal joint torque and corresponds well with minimal muscle activation. Minimum jerk likewise yields minimal torque change. It plausible that humans learn to minimise the effort of muscle usage and indirectly precipitate properties like minimal jerk. While torque minimisation has been used in computer animation to generate transitions between motion clips [Rose et al. (1996)], and to guide constraint based animations for extremely simple articulated figures [Cohen (1992)], it is not in general consistent with human motion because minimised torque tends to generate sharp discontinuities of acceleration [Kawato (1996)].

2.7.9 Minimum torque change

The minimal torque *change* model [Uno et al. (1989)] is consistent with minimum jerk and yields slightly curved paths dependent on the joint configuration and the area of the workspace in which motion occurs. Furthermore, the IK and ID problems of calculating postures and appropriate joint torques are incorporated because minimum torque change is an intrinsic dynamic model, *i.e.* motion planning occurs in intrinsic joint angle space and takes account of Newtonian dynamic properties.

Torque change minimisation closely relates to jerk minimisation because at zero velocity acceleration is proportional to torque. The model differs from minimum jerk most significantly by taking into account gravity, external forces and the complex dynamics of the arm. These differences are emphasised by Uno et al. (1989).

2.7.10 Minimum muscle tension change

In order to establish a closer correspondence with human motion the minimum torque model was extended to the minimum muscle tension change model [Uno et al. (1989)]. By considering muscles' tension sensors (the Golgi apparatus) and length/velocity sensors (spindle fibres) a model was developed in which the sensory control and muscle based actuators correspond more accurately with human biomechanics than a joint/torque model. The gently curved nature of actual hand trajectories are reproduced and the model corresponds well with conclusions based on a model of a monkey's arm incorporating seventeen muscles [Dornay et al. (1996)].

As a further step the minimum muscle tension change model was extended to the minimum motor command change model.

2.7.11 Minimum motor command change

According to this theory the neural signals activating the muscles of the body undergo minimal change throughout motion. Since motor command information is available directly in the brain it is perhaps more plausible that optimisation occurs at the motor command level rather than based on delayed sensory feedback from muscles. The minimum motor command change model [Kawato (1992, 1996)] supposes that smooth motion is generated by smoothing at the level of neuronal firing patterns. The motion may be characterised as the smoothest possible trajectory in motor command space (intrinsic) and since the ID solution is implicit the model is dynamic.

2.7.12 Optimisation over signal dependent noise

Harris and Wolpert (1998) proposed that planning occurs as an optimisation at the control signal level, modelling control with a noise term proportional to the activation signal and optimising end point positional error.

The model predicts bell shaped velocity profiles, reproduces the $\frac{1}{3}$ power law, yields inherently smooth trajectories and is biologically plausible since

there are no complicated integrations required of the brain.

2.7.13 Equilibrium point control

The equilibrium point hypothesis was first proposed by Feldman [Feldman (1966a,b)], and has received much attention in the literature [Bizzi et al. (1976), Abend et al. (1982), Feldman (1986), Latash and Gottlieb (1991)].

The essential hypothesis is that the limbs of the body are controlled in terms of equilibrium positions where the forces generated by the muscles attached to a given joint are in balance. While the position of a joint may not correspond to its current equilibrium position, it is assumed that the muscles responding to the target equilibrium position will bring the joint to rest at a target angle.

There are several models based on the equilibrium hypothesis. Feldman's original λ model assumes that muscles generate an exponential force response. For each muscle there is a corresponding λ value that represents the threshold value of the muscle's stretch reflex. If the muscle has a current length of x then the force generation f is exponential if the muscle is stretched beyond its target length:

$$f = \beta_0(e^{\beta_1(x-\lambda)} - 1) \qquad x > \lambda \tag{2.15}$$

$$= 0 \qquad x \leq \lambda \tag{2.16}$$

Each joint has at least two opposed muscles so that by setting the ratio of opposing muscles' λ parameters the length at which the forces balance may be selected, and by scaling their magnitudes the stiffness of the joint may also be controlled. According to Feldman's model the orientations of the joints are controlled by setting the λ parameters corresponding to desired postures and linearly interpolating the λ values. This intrinsic dynamic process causes the equilibrium position of the joints to translate from one posture to the next, while the actual position depends on the target joint angles, joint stiffnesses, the current joint angles and dynamic properties of the figure and its environment. The model thus generates curved hand paths and hooks at the

end of movements. A comparative discussion with other equilibrium point models may be found in Appendix D.3.9.

2.7.14 Conclusion

This concludes the discussion of human motion production research. The field is rich and well explored, with over fifty years of empirical and theoretical research. A great many unresolved issues still however remain. The question of whether motions are planned in intrinsic space related to figure kinematics or the extrinsic Cartesian space of the physical environment has not been agreed. Neither has it been firmly established whether planning takes account of dynamic characteristics and is inherently dynamic, or occurs entirely in kinematic space leaving the dynamic properties to be resolved during motion execution. Planning motions with respect to the dynamic properties of the body and the environment certainly necessitates the use of some internal dynamic model and an explicit solution to the inverse kinematics and inverse dynamics problems. Kawato (1996) argues that this is indeed the case.

A great deal of empirical evidence has been gathered, but while most of the models can be supported by some empirical evidence contradictory evidence also exists in most cases. It seems that the planning strategy used by the brain is highly adaptive, capable of automatic improvement, and dependent upon the context of motion generation. Even with the benefit of Magnetic Resonance imaging to identify the areas of the brain active during motion generation and planning very few concrete conclusions have been drawn.

It is interesting to note that the greater proportion of models involve some optimisation process: minimum Cartesian derivatives, minimum torque change, minimum motor command change, optimisation over signal dependent noise, *etc.* While the optimisation criteria may be unrelated, the popularity of optimisation models may be partially explained by the comment that "The use of optimisation techniques to model natural behaviour is appealing because of the analogy it bears to the optimisation presumed to occur as

a result of natural selection." [Flash and Hogan (1985)]. Alternatively, one might argue that optimisation models are a convenient way of specifying motion generation without the need to supply an explicit generation algorithm and have therefore been favoured in the literature.

Finally, a detailed discussion of the evidence and criticisms relating to the motion production theories described above was omitted for the sake of brevity. Appendix D.3 gives further details of such evidence and arguments, and presents some variations on the models described. Alternatively, Osu et al. (1997) provide a further survey of human motion production.

Chapter 3

Designing the motion production environment

Theoretical research

This chapter describes the theoretical aspects of the motion production environment. As such, the majority of theoretical research is summarised, and the reader is prepared for the implementation in Chapter 4.

This chapter begins with a discussion of various design decisions (Section 3.1), leading up to an abstract overview of the motion production environment. Motion generation models are then discussed in Section 3.2. For each model the theoretical basis is recalled, and the authoring research group, expected fidelity of natural motion reproduction, and expected computational complexity are stated. Each model is defined in the language of formal mathematics.

A general derivative minimisation technique is then presented (Section 3.3). This compound matrix method was used to find optimal splines satisfying a large group of motion models. On-line methods for implementing some motion models are then discussed.

3.1 Motion production environment

The generation of motion based on human motion production hypotheses, and subsequent testing psychovisually, computationally, and analytically is a broad problem. The generation of motion itself lies at the core, but a supporting framework is also required. A motion production environment was therefore designed.

The motion production environment was intended to support motion generation, editing, visualisation and rendering. This environment was ultimately implemented as three distinct software applications, as will be described in Chapter 4.

3.1.1 Conceptual design

It was decided that all motions would be based on movements of the hand. It was necessary to select a class of motions that is well represented in the human motion production literature, and also yields a wide and expressive range of possibilities. Motions based around hand reaching targets can include significant movement in the body itself, thus justifying and exploiting the use of a full humanoid figure.

A fundamental characteristic of the motion production environment is the method of motion specification. Since it was decided that all motions would be hand movements, it was thought that hand trajectories would constitute a suitable specification. The full spatial description of trajectories, and the joint angles of the body are variables explicitly controlled by some motion production models. However, another consideration is that many of the hypotheses in the human motion production literature have been formulated and tested according to certain repeatable motion primitives, the majority of which fall into one of three categories: point-point hand trajectories, via-point hand trajectories, and indefinitely repeating patterns. Via-point paths are similar to point-point motions except that they must also pass *through* a third point.

A natural choice for motion specification is therefore the knots representing interpolating splines. A pair of knots represents a point-point hand trajectory, three knots represents a via-point trajectory, and a cyclic spline may also be fitted to an arbitrary collection of knots. It was decided that the duration of motion should also be specified, since duration was thought to be a factor determined by the context of motion -and therefore requires either user specification or some justifiable artificially intelligent selection process beyond the scope of this research.

Since a number of the motion models were expected to require time consuming calculations it was decided to generate motion off-line. The generation of motions simultaneous with motion production on-the-fly is however, a desirable characteristic that facilitates interactive applications. On-line implementations therefore represent a related problem relevant to model evaluation and were investigated analytically where possible. Since the visual result is the same irrespective of whether motion is generated on-line or prepared off-line, it was not felt to be an issue relevant to implementation and the motion production environment was designed to operate off-line. A direct consequence is that motions must be stored in a persistent format.

A specific file format that will be described in Section 4.1.1 was used. It is possible to store motion according to the primitives used to specify the motion such as spline control points, or the internal representation strategy of motion generation algorithms. However, motions must be exported for performance by a humanoid robot, rendering using a commercial animation package, be analysed both mathematically and visually, and edited. It was therefore decided at the conceptual design stage that motion would be stored using discrete motion samples at a parameterised frequency. This simple format facilitates an arbitrary resolution of accuracy and straightforward interchange with the various applications providing the required facilities.

It was intended to both render motions with a computer graphic figure and perform them with a humanoid robot. The problem of rendering was considered outwith the scope of the research, and so it was necessary that the

motion production environment be capable of outputting motions in a format suitable for import into some commercial rendering package. It was likewise required to be compatible with the control mechanism of the humanoid robot.

The theoretical outline of the motion production environment is summarised in Figure 3.1.

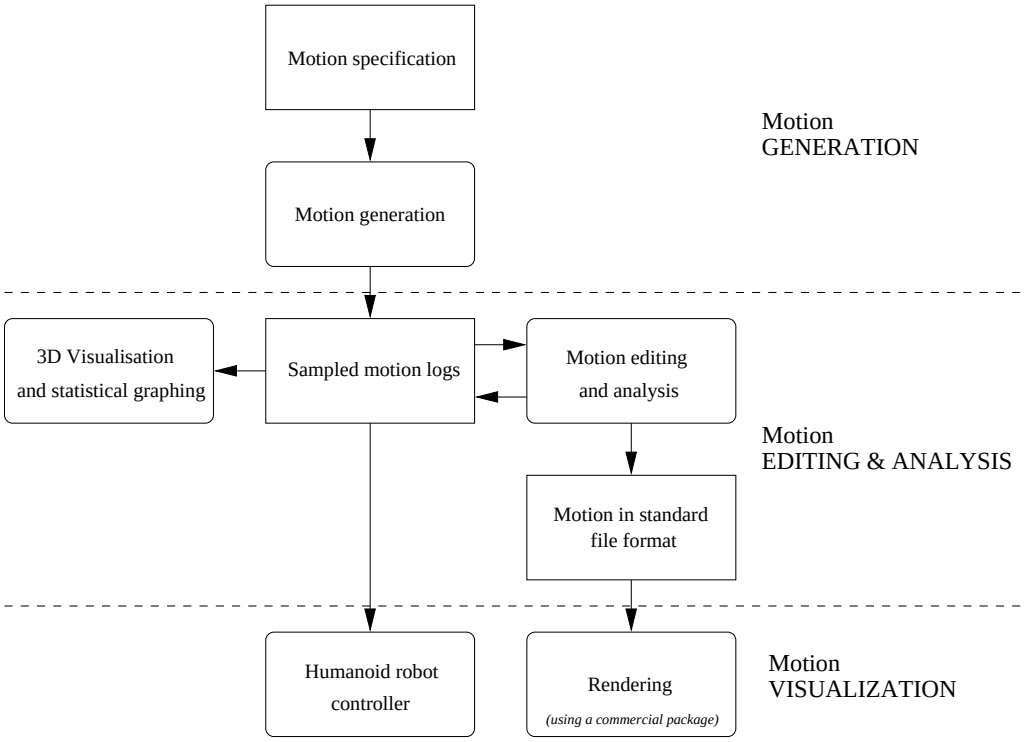

Figure 3.1: The Motion Production Environment

Many of the human motion hypotheses are formulated in terms of optimisation criteria. It is therefore natural that motions be generated using optimisation methods. As will be revealed in the following section, a broad class of models formulated using related optimisation criteria were solved using a general analytical method. Some of the models selected were purely kinematic, and thus required inverse kinematic solutions. A number of the models were based on dynamic optimisation criteria and thus required inverse dynamic solutions to evaluate the optimisation metrics. Finally, some of the

models were based on dynamic interaction and thus required full forward dynamic simulation to implement.

The inverse kinematics, forward and inverse dynamics problems are well established techniques and have been well researched in their own right. For the purposes of this research it was considered reasonable to make use of existing technology and these issues were solved using software components of a robot simulator. Likewise, the rendering problem was solved using a commercial package, MetaCreation's Poser. The practical issues are discussed elsewhere -in Chapter 4.

3.2 Motion generation models

Twelve motion production models were designed in accordance with the criteria stated in the first research goal (Section 1.1.1). Motion hypotheses were thus selected from a range of research groups to explore a range of implementation complexities, a range of human psychomotor fidelities, and utilised both purely kinematic and dynamic planning. All motions were specified in terms of knot points that should be interpolated by the trajectory of the hand according to the capabilities of the motion model.

Besides the psychomotor based models, two experimental motion production models were also designed in accordance with the second research goal. These experimental models were intended to replicate the properties of what were expected to be the most convincing psychomotor based motion production models. As such they exploit computational optimisations bearing little correlation with the actual motor planning strategies likely to occur in the brain.

The fourteen models are described in the following text. Formal specifications rigidly describe the motions each model should generate without constraining the computational method. Since all motions are specified using Cartesian knots, many are handled with splines, and all are output as discre-

tised posture sets, the following definitions are central to this formalisation[1].

Definition 3 (Knot specification)

$$(T, \{p_0, \ldots, p_n\}) \in \mathbb{K}_d \quad \Leftrightarrow \quad T \in \mathbb{R}^+, \quad n, d \in \mathbb{Z}^+ \\ \forall i \in [0, n] \quad p_i \in \mathbb{R}^d$$

$\mathbb{K}_d$ represents the set of valid knot specifications in d dimensions. T represents the duration of the trajectory intended to pass through the set of knots $\{p_0, \ldots, p_n\}$ in sequence.

Definition 4 (Spline specification)

$$(T, s) \in \mathbb{S}_d \quad \Leftrightarrow \quad T \in \mathbb{R}, \quad d \in \mathbb{Z}^+, \quad s : \mathbb{R} \mapsto \mathbb{R}^d \\ \forall t \in [0, T] \quad \forall \epsilon \in \mathbb{R}^+ \quad \exists \delta > 0 \quad \text{s.t.} \\ \forall t' \in [0, T] \quad \text{s.t.} \quad |t - t'| < \delta, \quad |s(t) - s(t')| < \epsilon$$

$\mathbb{S}_d$ represents the set of continuous functions defined over the interval $[0, T]$ that map into d dimensional space.

Definition 5 (Motion specification)

$$(T, f, \{p_0, \ldots, p_n\}) \in \mathbb{M}_d \quad \Leftrightarrow \quad T, f \in \mathbb{R}^+, \quad d \in \mathbb{Z}^+, \quad n = \left\lfloor \frac{T}{f} \right\rfloor \\ \forall i \in [0, n] \quad p_i \in \mathbb{R}^d$$

$\mathbb{M}_d$ represents the space of all valid motions in d dimensions. T and f represent the duration and sampling frequency respectively, and $\{p_0, \ldots, p_n\}$ represent the set of ordered samples. d should typically be equal to the number of DOFs, so that each sample is a posture.

Three fundamental types have thus been established. These roughly correspond to the motion specification type (knots), the internal representation (splines), and the output type (motions). In order to formalise their relationships with each other two further definitions are necessary:

[1] All definitions are repeated in Appendix A.

Definition 6 (Spline-knot correspondence)

$$SKC(s,k) \quad \Leftrightarrow$$

$$\exists d, n \in \mathbb{Z}^+ \quad \text{s.t.} \quad s = (T, s') \in \mathbb{S}_d, \quad k = (T, \{p_0, \ldots, p_n\}) \in \mathbb{K}_d$$
$$\exists (t_0, \ldots, t_n) \in \mathbb{R}^{n+1} \quad \text{s.t.} \quad \forall i, j \in [0, n] \quad \text{s.t.} \quad i < j,$$
$$t_i < t_j \quad \wedge \quad s'(t_i) = p_i \quad \wedge \quad t_0 = 0 \quad \wedge \quad t_n = T$$

The spline s correctly interpolates the knot set k in duration T.

Definition 7 (Motion sampling correspondence)

$$MSC(s,m) \quad \Leftrightarrow$$

$$\exists d, n \in \mathbb{Z}^+ \quad \text{s.t.} \quad s = (T, s') \in \mathbb{S}_d, \quad m = (T, f, \{p_0, \ldots, p_n\}) \in \mathbb{M}_d$$
$$\forall i \in [0, n], \quad p_i = s'\left(\frac{i}{f}\right)$$

The motion m is a valid sampling of the spline s at frequency f.

The formal definitions of motion in the following text assume that the specified motion is non-cyclic. Unfortunately it is awkward to handle the case of cyclic trajectories, and for the sake of brevity the extension of these definitions to the cyclic case is described in Appendix A.

3.2.1 Linear interpolation & minimum Cartesian acceleration

The simplistic trajectory investigations by Nelson (1983), working at Bell laboratories, were the inspiration for two motion production models: linear interpolation, and minimum Cartesian acceleration. Linear interpolation between two points in a fixed time is equivalent to a minimum Cartesian velocity interpolation.

According to these models, motions are planned in extrinsic kinematic space, *i.e.*, as explicit hand trajectories. The models represent a simplistic

initial step into the realm of Cartesian trajectory planning, and little evidence supports their replication of human motion.

In order to formalise these motion models two further definitions are required. The first incorporates an optimality criterion into the definition of an interpolating spline, and the second establishes a kinematic correspondence between a Cartesian trajectory and a joint angle trajectory.

Definition 8 (Optimal spline-knot correspondence)

$$OSKC(s,k,C) \quad \Leftrightarrow$$

$$\begin{aligned} &\exists d \quad \text{s.t.} \quad s \in \mathbb{S}_d, \quad C : \mathbb{S}_d \mapsto \mathbb{R} \\ &SKC(s,k) \quad \wedge \quad \forall s' \in \mathbb{S}_d \quad \text{s.t.} \quad SKC(s',k), \quad C(s) \leq C(s') \end{aligned}$$

The d dimensional spline s correctly interpolates the knot set k in duration T, and has minimal cost according to the function C over the space of all such splines that correctly interpolate k.

Definition 9 (Kinematic spline correspondence)

$$KSC(s_3, s_{NDOFS}) \quad \Leftrightarrow$$

$$\begin{aligned} &s_3 = (T, s_3') \in \mathbb{S}_3, \quad s_{NDOFS} = (T, s_{NDOFS}') \in \mathbb{S}_{NDOFS} \\ &\forall t \in [0,T], \quad FK(s_{NDOFS}'(t), s_3'(t)) \end{aligned}$$

The postures specified in s_{NDOFS} yield the hand positions specified in s_3 at every point on the spline.

Since linear interpolation is equivalent to minimum velocity interpolation, it may be specified as follows:

Definition 10 (Minimum Cartesian velocity motion)

$$MV(k,m,f) \quad \Leftrightarrow$$

$$k = (T, \overline{p}) \in \mathbb{K}_3, \quad m = (T, f, \overline{p'}) \in \mathbb{M}_{NDOFS}$$
$$\exists s_3 = (T, s_3') \in \mathbb{S}_3, \quad s_{NDOFS} \in \mathbb{S}_{NDOFS} \quad \text{s.t.}$$

$$OSKC\left(s_3, k, \int_o^T \left(\frac{ds_3'}{dt}\right)^2 dt\right) \quad \wedge$$
$$KSC(s_3, s_{NDOFS}) \quad \wedge \quad MSC(s_{NDOFS}, m)$$

Definition 11 (Minimum Cartesian acceleration motion)

$$MA(k,m,f) \quad \Leftrightarrow$$

$$k = (T, \overline{p}) \in \mathbb{K}_3, \quad m = (T, f, \overline{p'}) \in \mathbb{M}_{NDOFS}$$
$$\exists s_3 = (T, s_3') \in \mathbb{S}_3, \quad s_{NDOFS} \in \mathbb{S}_{NDOFS} \quad \text{s.t.}$$

$$OSKC\left(s_3, k, \int_o^T \left(\frac{d^2 s_3'}{dt^2}\right)^2 dt\right) \quad \wedge$$
$$KSC(s_3, s_{NDOFS}) \quad \wedge \quad MSC(s_{NDOFS}, m)$$

In order to output a motion generated using either of these models it is necessary sample the spline representing the hand coordinates in Cartesian space and perform inverse kinematics on each sample. Due to this burden of repeated inverse kinematic solutions the computational complexity is significant and may be regarded as *medium* relative to the other models.

3.2.2 Minimum Cartesian jerk

The minimum jerk model proposed by Flash (1983), working at the Massachusetts Institute of Technology, was the basis for the third motion production model.

The minimum jerk theory was discussed in detail in Section 2.7. Motions are planned kinematically as extrinsic trajectories. A number of criticisms have been levelled, in particular, that it does not take account of dynamic loads, and that trajectories are workspace independent [Uno et al. (1989), Osu et al. (1997)]. However, a large body of supportive evidence also testifies that this model is a faithful replication of human hand trajectories in a number of situations.

Solving for a minimum jerk trajectory requires complicated mathematical solution, but can be achieved rapidly. The motion generation process is the same as for minimum velocity or acceleration; as with those models, computation time is dominated by inverse kinematics calculations at each sample point on the trajectory, and the overall computational requirement is also *medium*.

Definition 12 (Minimum Cartesian jerk motion)

$$MJ(k,m,f) \quad \Leftrightarrow$$

$$k = (T, \overline{p}) \in \mathbb{K}_3, \quad m = (T, f, \overline{p'}) \in \mathbb{M}_{NDOFS}$$
$$\exists s_3 = (T, s_3') \in \mathbb{S}_3, \quad s_{NDOFS} \in \mathbb{S}_{NDOFS} \quad \text{s.t.}$$

$$OSKC\left(s_3, k, \int_o^T \left(\frac{d^3 s_3'}{dt^3}\right)^2 dt\right) \quad \wedge$$
$$KSC(s_3, s_{NDOFS}) \quad \wedge \quad MSC(s_{NDOFS}, m)$$

3.2.3 Minimum Cartesian snap

Minimum Cartesian snap is a motion planning method that represents an extension of the well established minimum jerk method. Also proposed by Flash (1983), it is subject to the same criticism and supported by the same evidence as the minimum jerk model. The computational method is the same and the time requirement is similarly *medium* by comparison to other models.

This model was selected in an attempt to support the minimum jerk model by demonstrating that jerk minimisation yields all the characteristics of human motion that arbitrary orders of derivative minimisation are capable of replicating.

Definition 13 (Minimum Cartesian snap motion)

$$MS(k,m,f) \quad \Leftrightarrow$$

$$k = (T, \overline{p}) \in \mathbb{K}_3, \quad m = (T, f, \overline{p'}) \in \mathbb{M}_{NDOFS}$$
$$\exists s_3 = (T, s_3') \in \mathbb{S}_3, \quad \exists s_{NDOFS} \in \mathbb{S}_{NDOFS} \quad \text{s.t.}$$

$$OSKC\left(s_3, k, \int_o^T \left(\frac{d^4 s_3'}{dt^4}\right)^2 dt\right) \quad \wedge$$
$$KSC(s_3, s_{NDOFS}) \quad \wedge \quad MSC(s_{NDOFS}, m)$$

3.2.4 Straightness in joint angle space

The work of Soechting and Lacquaniti (1981), of the Neurophysiology Laboratory at the University of Minnesota, led to the hypothesis that motions are planned as straight trajectories in joint angle space. This hypothesis is empirically supported, although criticisms have been raised as to whether straightness in joint angle space is explicitly planned or merely a function of

limb kinematics [Hollerbach and Atkeson (1986)]. For the purposes of realistic motion synthesis this criticism is irrelevant. While empirical evidence supports the curved nature of hand trajectories derived from straight paths in joint angle space, it is doubtful whether the velocity characteristics of the hand would correlate well with those observed empirically.

According to this model, motion is planned kinematically at the intrinsic level of the joint angles of the body. Hand positions are specified at the knots, and inverse kinematic computations are necessary to find corresponding postures, but straightness in joint angle space means that these postures may be interpolated linearly without further recourse to inverse kinematics. The relative computational complexity of the model is therefore *low*.

A formalisation of the model requires a law of kinematic correspondence between Cartesian and joint angle knots.

Definition 14 (Kinematic knot correspondence)

$$
\begin{aligned}
&KKC(k_3, k_{NDOFS}) \quad \Leftrightarrow \\
&\quad k_3 = (T, \{p_0, ..., p_n\}) \in \mathbb{K}_3, \quad k_{NDOFS} = (T, \{p'_0, ..., p'_n\}) \in \mathbb{K}_{NDOFS}, \\
&\quad \forall i \in [0, n], \quad FK(p'_i, p_i)
\end{aligned}
$$

The postures specified in k_{NDOFS} yield the hand positions specified in k_3 at every knot.

Definition 15 (Minimum angular velocity motion)

$$
\begin{aligned}
&MAV(k, m, f) \quad \Leftrightarrow \\
&\quad k = (T, \overline{p}) \in \mathbb{K}_3, \quad m = (T, f, \overline{p'}) \in \mathbb{M}_{NDOFS}, \\
&\quad \exists k_{NDOFS} \in \mathbb{K}_3, \quad \exists s_{NDOFS} = (T, s'_{NDOFS}) \in \mathbb{S}_{NDOFS} \quad \text{s.t.} \\
&\\
&\quad KKC(k, k_{NDOFS}) \quad \wedge \\
&\quad OSKC\left(s_{NDOFS}, k_{NDOFS}, \int_o^T \left(\frac{ds'_{NDOFS}}{dt}\right)^2 dt\right) \quad \wedge \\
&\quad MSC(s_{NDOFS}, m)
\end{aligned}
$$

3.2.5 Minimum angular derivatives

The hypothesis that motions are planned as straight paths in joint angle space does not make any explicit reference to hand velocity. It is therefore unreasonable to test this hypothesis, and its applicability to motion production by means of linear joint angle interpolation alone. While linear interpolation is the simplest implementation of the hypothesis it is desirable to combine joint angle interpolation with potentially more accurate velocity profiles.

Three further intrinsic kinematic models, analogous to the extrinsic Cartesian interpolations, were therefore developed: minimum angular acceleration, jerk and snap. The original hypothesis that minimum Cartesian jerk trajectories reduce wear and tear on the musculo-skeletal system is maintained even more strongly by the concept of minimum angular jerk trajectory planning. Minimum angular acceleration and snap were included because it was expected that minimum angular snap would prove indistinguishable from minimum angular jerk, thus ratifying the latter's sufficiency, and minimum angular acceleration would either also prove satisfactory, or demonstrate the requirement for derivative minimisation of at least the third order.

Computational complexity for all three models is comparable, but slightly higher than that of linear joint angle interpolation. The theoretical method is the same: inverse kinematics yields postures corresponding to each knot point, and these postures are interpolated. The dominant computational factor is trajectory interpolation with minimised derivatives, and the relative complexity is *low*.

Formal definitions for the three models are analogous to that of linear (minimum velocity) angular interpolation. The integrand in the optimal spline specification is a second, third and fourth derivative for the minimum acceleration, jerk and snap cases respectively. The definitions have been omitted here for the sake of brevity but may be found in Appendix A.

3.2.6 Minimum jerk virtual trajectory control

The minimum jerk virtual trajectory (MJVT) control hypothesis was intended to replicate the workspace dependent nature of trajectory curvature and inaccuracies of actual human motions [Flash (1987)]. It represents a significant development of Flash's original minimum jerk hypothesis.

Empirical evidence supports the fidelity of MJVT for various motions. Critics have observed that the model does not replicate human motion accurately for fast (under 0.5 seconds) point-point motions, and that it does not support movement refinement with practise [Kawato (1992)]. The latter complaint is irrelevant for animator specified motion synthesis, and it is generally accepted that the model has very good replication fidelity.

The model is borderline kinematic/dynamic since although motions are planned in extrinsic kinematic space, an on-line process governs motion production by regulating muscle forces with spring-damper systems, *i.e.*, intrinsic dynamics. Implementation thus requires the planning phase accomplished by the minimum jerk model, *and* forward dynamic simulation. Having generated the minimum jerk motion according to the method above, a forward dynamic simulation is initialised with the starting posture of the minimum jerk model. The simulation runs for a fixed duration and the torques at each joint $\tau_i(t)$ are defined in terms of the minimum Cartesian jerk trajectories $\phi_i(t)$ and the actual joint angles of the figure under simulation $\theta_i(t)$ according to stiffness and viscosity parameters β_0 and β_1 respectively.

$$\tau_i(t) \quad = \quad \beta_0(\phi_i(t) - \theta_i(t)) - \beta_1 \frac{d\theta_i}{dt}(t) \tag{3.1}$$

The relative computational complexity is thus *high* for the purposes of computer animation, but *medium* for human or robot motion production where forward dynamics occur implicitly.

Definition 16 (Minimum jerk virtual trajectory motion)

$$
\begin{aligned}
&MJVT(k,m,f) \quad \Leftrightarrow \\
&\quad k=(T,\overline{p}) \in \mathbb{K}_3, \quad m=(T,f,\overline{p'}) \in \mathbb{M}_{NDOFS} \\
&\quad \exists s_3=(T,s_3') \in \mathbb{S}_3, \quad \exists s_{VT}=(T,s_{VT}') \in \mathbb{S}_{NDOFS}, \\
&\quad \exists s_{NDOFS}=(T,s_{NDOFS}') \in \mathbb{S}_{NDOFS}, \quad \exists \tau : \mathbb{R} \mapsto \mathbb{R}^{NDOFS}, \quad \text{s.t.} \\
&\quad OSKC\left(s_3,k,\int_o^T \left(\frac{d^3 s_3'}{dt^3}\right)^2 dt\right) \quad \wedge \quad KSC(s_3,s_{NDOFS}) \quad \wedge \\
&\quad \forall t \in [0,T], \quad \forall i \in [0,NDOFS-1], \\
&\quad \tau_i(t)=\beta_0(s_{VTi}'(t)-s_{NDOFSi}'(t))-\beta_1 s_{NDOFSi}'(t) \quad \wedge \\
&\quad FD\left(s_{NDOFS}'(t),\frac{ds_{NDOFS}'}{dt}(t),\tau(t),\frac{d(s_{NDOFS}')^2}{dt^2}(t)\right) \quad \wedge \\
&\quad MSC(s_{NDOFS},m)
\end{aligned}
$$

The virtual trajectory s_{VT} is specified in the same way as the minimum Cartesian jerk trajectory. The dynamic constraint formulated with FD asserts that S_{NDOFS} is a dynamically consistent trajectory where the internal torques τ are defined by spring-damper systems between the actual angles and the angles in the virtual trajectory. The parameters of the spring-dampers are β_0 and β_1.

3.2.7 Equilibrium point λ-model

The equilibrium point λ hypothesis (EPH) regarding human muscular control was proposed by Feldman (1966a), working at the Institute of Biological Physics in Moscow.

The EPH model is based on empirical evidence about human muscle physiology and force response. It states that the brain controls motions in terms of equilibrium positions of the joints with respect to muscle tension. Supportive evidence exists, but it has also been shown that stiffness values for real human movements are not replicated by the EPH model. The fidelity of motion replication is therefore difficult to assess.

The actual motion is brought about by the interaction of the muscles' length/stiffness characteristics with the dynamic environment. As such the EPH is an intrinsic dynamic model, and requires dynamic planning and forward dynamic simulation. The dynamic planning requires the generation of dynamically stable postures at each specified knot, *i.e.*, inverse dynamics. These computations need only occur at the knots however, and computation time is dominated by forward dynamic simulation.

Motion generation proceeds by performing inverse kinematics to generate a posture satisfying each knot specification, *i.e.*, a set $\{\theta_{ij}\}$ for the i^{th} knot and the j^{th} joint angle. Each posture is then processed using inverse dynamics to find a set of stable torques that maintain it, yielding a set $\{\tau_{ij}\}$. These torques are then mapped into the λ domain. Recalling from Section 2.7, the muscle force f is related to the target muscle length λ and the actual muscle length x:

$$f = \beta_0 e^{\beta_1 (x-\lambda)} \tag{3.2}$$

$$\therefore \quad \lambda = x - \frac{1}{\beta_1} \ln \left(\frac{f}{\beta_0} + 1 \right) \tag{3.3}$$

Having mapped the torques $\{\tau_{ij}\}$ into the λ domain[2] using Equation 3.3, λ parameters at any given instant during forward dynamic simulation are found by linearly interpolating between the λ parameters at the adjacent knots. The joint torques at any given instant are then found by using Equation 3.2 where f represents the torque, x the actual joint angle and λ the interpolated λ value. There is no optimisation criterion by which to select the passage times of a series of knots. Rather than further complicate the model, it was decided that for this model the knot passage times should be specified by the animator.

The relative computational complexity is the same as that of MJVT, *high* for full motion synthesis (animation) and *medium* for robot or human motion production.

[2]The original λ model was formulated in terms of muscle force rather than joint torque, and the implicit assumption that these are equivalent is discussed in Section 6.2.

Definition 17 (Equilibrium point λ motion)

$$
\begin{aligned}
&EPH(k, m, f, \overline{t}) \quad \Leftrightarrow \\
&\quad \overline{t} = (t_0, \ldots, t_n) \in \mathbb{R}^{n+1}, \quad k = (T, \overline{p}) \in \mathbb{K}_{NDOFS}, \\
&\quad k_{NDOFS} = (T, k'_{NDOFS}) \in \mathbb{K}_{NDOFS}, \\
&\quad s_{NDOFS} = (T, s'_{NDOFS}) \in \mathbb{S}_{NDOFS} \\
&\quad \exists \lambda_0 \ldots \lambda_{NDOFS-1}, \quad \text{where} \quad \lambda_i = (\lambda_{i0}, \ldots, \lambda_{in}) \in \mathbb{R}^{n+1}, \\
&\quad \exists \tau, \tau' = (\tau'_0, \ldots, \tau'_n) \in \mathbb{R}^{(n+1)\times NDOFS}, \\
&\quad \exists \Lambda : \mathbb{R} \mapsto \mathbb{R}^{NDOFS} \quad \text{s.t.}
\end{aligned}
$$

$$
\begin{aligned}
&KKC(k, k_{NDOFS}) \quad \wedge \quad \forall i \in [0, n], \quad FD(k'_{NDOFSi}, \overline{0}, \tau'_i, \overline{0}), \\
&\forall j \in [0, NDOFS - 1], \quad \lambda_{ij} = k'_{NDOFSij} - \frac{1}{\beta_0} ln\left(\frac{\tau'_{ij}}{\beta_1} + 1\right), \\
&\forall t \in [0, T], \quad \exists i \quad \text{s.t.} \quad t_i \leq t \leq t_{i+1} \quad \wedge \\
&\Lambda(t) = \lambda_i \left(\frac{t_{i+1} - t}{t_{i+1} - t_i}\right) + \lambda_{i+1} \left(\frac{t - t_i}{t_{i+1} - t_i}\right) \quad \wedge \\
&\forall t \in [0, T], \quad FD\left(s'_{NDOFS}(t), \frac{ds'_{NDOFS}}{dt}(t), \tau(t), \frac{d(s'_{NDOFS})^2}{dt^2}(t)\right) \\
&\wedge \quad \tau(t) = \beta_0 e^{\beta_1(s'(t) - \Lambda(t))} \quad \wedge \quad MSC(s_{NDOFS}, m)
\end{aligned}
$$

The postures corresponding to each Cartesian knot are represented by k_{NDOFS}. τ' represents the torques necessary to maintain those postures with zero velocity and acceleration. The set of λ parameters that precipitate these torques in a static configuration are represented by λ itself, and Λ gives their linear interpolation over the duration of the whole motion. Finally s_{NDOFS} is a motion that is dynamically consistent when all the internal torques are generated according to the appropriate λ torque generation equation.

3.2.8 Minimum torque change

Minimum torque change was proposed by Uno et al. (1989). Uno was working in the Department of Mathematical Engineering and Information Physics at

the University of Tokyo, but the hypothesis is representative of a series of developments lead by Kawato at ATR in Kyoto.

Empirical evidence supports the model, which takes into account dynamic factors, yields workspace dependent curvature, and responds correctly to dynamic loads. Trajectories are planned in intrinsic joint angle space according to a dynamic metric. Motion planning therefore requires integrated inverse dynamics computations to evaluate the torque change cost of motions, and optimisation over the space of possible motions. The computational complexity is therefore *very high* and it is infeasible to calculate guaranteably exact solutions.

Various theoretical methods are applicable to the generation of minimum torque change motions by optimisation. The method chosen represented the motion using minimum jerk splines in joint angle space. A minimum jerk interpolation through the postures corresponding to the given knots was used as an starting trajectory. The density of these angle space knots controlling the trajectory was then increased by sampling the spline at multiple instants and inserting the new knots as control points in the spline description. The positions of the knots were then perturbed iteratively, aiming to reduce the cost of the trajectory. The trajectory cost was evaluated by sampling postures throughout the motion, performing inverse dynamics on each posture, calculating the torque change cost between sampled postures, and summing the results.

Definition 18 (Minimum torque change motion)

$$MTC(k, m, f) \quad \Leftrightarrow$$

$$k = (T, \overline{p}) \in \mathbb{K}_3, \quad m = (T, f, \overline{p'}) \in \mathbb{M}_{NDOFS},$$
$$\exists k_{NDOFS} \in \mathbb{K}_3, \quad \exists \tau = (\tau_0, \ldots, \tau_{NDOFS-1}) : \mathbb{R} \mapsto \mathbb{R}^{NDOFS},$$
$$\exists s_{NDOFS} = (T, s'_{NDOFS}) \in \mathbb{S}_{NDOFS} \quad \text{s.t.}$$

$$KKC(k, k_{NDOFS}) \quad \wedge \quad MSC(s_{NDOFS}, m) \quad \wedge$$

$$OSKC\left(s_{NDOFS}, k_{NDOFS}, \int_o^T \left[\sum_{i=0}^{NDOFS-1} \left(\frac{d\tau_i}{dt}\right)^2\right] dt \quad \text{where}\right.$$
$$\left.\forall t \in [0,T], \quad FD\left(s'_{NDOFS}(t), \frac{ds'_{NDOFS}}{dt}(t), \tau(t), \frac{d(s'_{NDOFS})^2}{dt^2}(t)\right)\right)$$

3.2.9 Minimum torque

Torque minimisation is a concept drawn from the animation literature [Cohen (1992), Rose et al. (1996)]. It is intuitively related to animal motion since torque corresponds to muscle activation, which in turn corresponds to effort. Minimum torque motions are thus intuitively natural, but it has not been proposed as a hypothetical criteria of human motion planning and was criticised since it tends to generate sharp discontinuities in motion [Kawato (1996)]. Nevertheless, it was thought that minimum torque motions would yield revealing comparisons with models based on approved hypotheses of human motion planning.

The computational complexity of torque minimisation is similar to that of torque change minimisation. The space of possible motions must be searched for motions that minimise the torque cost, which must in turn by calculated by integrating torques found using inverse dynamics. The only difference in fact, when compared to the minimum torque change optimisation process, is that having found the torques at each of the postures sampled over a candidate motion, the torques are simply squared and summed rather than compared with the torques of the preceding posture. The computational complexity is therefore also *very high*.

Definition 19 (Minimum torque motion)

$$
\begin{aligned}
&MT(k,m,f) \quad \Leftrightarrow \\
&\quad k=(T,\overline{p}) \in \mathbb{K}_3, \quad m=(T,f,\overline{p'}) \in \mathbb{M}_{NDOFS}, \\
&\quad \exists k_{NDOFS} \in \mathbb{K}_3, \quad \exists \tau=(\tau_0,\ldots,\tau_{NDOFS-1}) : \mathbb{R} \mapsto \mathbb{R}^{NDOFS}, \\
&\quad \exists s_{NDOFS}=(T,s'_{NDOFS}) \in \mathbb{S}_{NDOFS} \quad \text{s.t.} \\
&\quad KKC(k,k_{NDOFS}) \quad \wedge \quad MSC(s_{NDOFS},m) \quad \wedge \\
&\quad OSKC\left(s_{NDOFS},k_{NDOFS},\int_o^T \left[\sum_{i=0}^{NDOFS-1} \tau_i^2\right] dt \quad \text{where}\right. \\
&\quad \left.\forall t \in [0,T], \quad FD\left(s'_{NDOFS}(t),\frac{ds'_{NDOFS}}{dt}(t),\tau(t),\frac{d(s'_{NDOFS})^2}{dt^2}(t)\right)\right)
\end{aligned}
$$

3.2.10 Minimum torque posture interpolation

Both the torque and torque change minimisation models are extremely demanding computationally. In order to preserve some of their characteristics, while facilitating more rapid computation the minimum torque posture interpolation (MTP) method was designed.

This model is experimental and based on human motion production only in so far as it is related to the minimum torque change hypothesis. According to minimum torque posture interpolation, motions are derived by interpolating optimal postures. The burdens of the torque change minimisation model are the evaluation of torque change over the whole motion, and the broad search space of possible motions. Optimising postures individually reduces the search space to that of possible postures, *and* reduces the time required to evaluate a candidate solution's cost to one inverse dynamics computation. In fact the postures for all the knots were optimised simultaneously so the search space and cost evaluation is linearly related to the number of knots.

The optimisation metric consists of three summed costs: the sum squared torque necessary to maintain the figure *statically* in the posture, the distance of the hand from the target point, and the sum squared difference between the torque at each joint in consecutive postures. This metric therefore approximates both torque minimisation *and* torque change minimisation. Postures are interpolated in joint angle space using minimum jerk.

In general it is difficult to predict the convergence time of the optimisation procedure. The relative computational complexity may be estimated at *medium-high.*

Definition 20 (Minimum torque postures spline)

$$MTPS(k,s,d) \quad \Leftrightarrow$$

$$d \in \mathbb{R}, \quad k = (T, \{p_0, ..., p_n\}) \in \mathbb{K}_3, \quad s = (T, s') \in \mathbb{S}_{NDOFS}$$
$$\exists K_{NDOFS} \subset \mathbb{K}_{NDOFS}, \quad \exists k_{MTP} \in K_{NDOFS}, \quad \text{s.t.}$$

$$k_{NDOFS} = (T, \{p'_0, ..., p'_n\}) \in K_{NDOFS} \quad \Leftrightarrow \quad \forall i \in [0,n], \quad FK(p'_i, p_i)$$
$$\forall k' \in K_{NDOFS}, \quad C(k_{MTP}) \leq C(k') \quad \wedge$$
$$OSKC\left(s, k_{NDOFS}, \int_o^T \left(\frac{d^3 s'_3}{dt^3}\right)^2 dt\right)$$

$$\text{where} \quad C : K_{NDOFS} \mapsto \mathbb{R}^+, \quad C\left((T, \{q_0, ..., q_{(nd)}\}\right) =$$
$$\beta_0 \sum_{i=0}^{nd} \sum_{j=0}^{NDOFS-1} \tau_{ij}^2 \quad + \quad \beta_1 \sum_{i=1}^{nd} \sum_{j=0}^{NDOFS-1} (\tau_{ij} - \tau_{(i-1)j})^2, \quad \text{where}$$
$$\tau \in \mathbb{R}^{(nd+1)\times NDOFS} \quad \wedge \quad \forall i \in [0, nd], \quad FD(q_i, \overline{0}, \tau_i, \overline{0})$$

d is the number of knots that will be inserted into the optimal posture set for every knot in the specification k. k_{NDOFS} is the optimal posture set, and s is a minimum angular jerk spline through the optimal postures. C is the cost function for optimisation and incorporates both torque and torque change cost according to the scaling factors β_0 and β_1 respectively.

Definition 21 (Minimum torque postures motion)

$$MTP(k,m,f,d) \quad \Leftrightarrow$$
$$\exists s \in \mathbb{S}_{NDOFS} \quad \text{s.t.} \qquad MTSP(k,s,d) \quad \wedge \quad MSC(s,m,f)$$

3.2.11 Minimum torque posture virtual trajectory

The minimum torque posture virtual trajectory (MTPVT) model is an extension of the MTP model. Having generated optimal trajectories according to a computationally efficient optimal dynamic strategy the MTP generates motion by interpolation without further regard for dynamic properties. While the optimisation process itself is intended to yield motions of optimal dynamic smoothness that are inherently dynamically feasible, the posture interpolation process is not necessarily consistent dynamically. To counter this problem the MTPVT model uses the motion output by the MTP model as a virtual trajectory. The actual motion is brought about by spring-damper systems under dynamic simulation. The computational complexity is therefore *high*, but the incorporation of forward dynamics ensures full dynamic correctness.

Definition 22 (Minimum torque postures virtual trajectory motion)

$$MTPVT(k,m,f,d) \quad \Leftrightarrow$$
$$\exists s_{VT} = (T, s'_{VT}), s_{NDOFS} = (T, s'_{NDOFS}) \in \mathbb{S}_{NDOFS},$$
$$\exists \tau : \mathbb{R} \mapsto \mathbb{R}^{NDOFS}, \quad \text{s.t.}$$

$$MTSP(k, s_{VT}, d) \quad \wedge \quad \forall t \in [0,T], \quad \forall i \in [0, NDOFS-1]$$
$$\tau_i(t) = \beta_0(s'_{VTi}(t) - s'_{NDOFSi}(t)) - \beta_1 s'_{NDOFSi}(t) \quad \wedge$$
$$FD\left(s'_{NDOFS}(t), \frac{ds'_{NDOFS}}{dt}(t), \tau(t), \frac{d(s'_{NDOFS})^2}{dt^2}(t)\right) \quad \wedge$$
$$MSC(s,m,f)$$

s_{VT} is the virtual trajectory spline generated according to the minimum torque postures model. The constraint formulated using FD and τ states that S_{NDOFS} must be a dynamically consistent motion generated with internal torques calculated using the virtual trajectory as a target for a spring-damper system in each joint.

3.3 Optimal kinematic trajectories

A unifying principle describes all the trajectories that satisfy minimum Cartesian velocity, acceleration, jerk and snap, and also minimum angular velocity, acceleration, jerk and snap. They are all examples of minimum n^{th} derivative interpolations. Given a set of knot points specifying the trajectory, the minimum Cartesian velocity model is described by a three dimensional interpolating spline subject to minimisation of the 1^{st} derivative. Acceleration, jerk and snap are likewise minimum 2^{nd}, 3^{rd} and 4^{th} derivative interpolations respectively. Angular trajectories are interpolations in d dimensions, where d is the number of DOFs.

This section presents a general solution to all of these trajectory specifications. The solution requires iterated linear programming and yields an analytical description of the optimal trajectory. The total trajectory duration must be specified, but the passage times of any internal knots (not the initial or target knots) are found by iteration. Implementation issues are discussed in Section 4.3.1. The method presented was developed from scratch, but the mathematics of derivative minimisation are well known in functional data analysis [Ramsay and Silverman (1997)]. Arbitrary linear combinations of derivatives as an error metric for spline smoothing have been investigated, but no generalised method for arbitrary n^{th} derivative smoothing incorporating variable knot passage times was found in the literature.

3.3.1 General solution

Given an arbitrary number of ordered knot points in Cartesian or joint angle space (*i.e.*, of consistent but arbitrary dimensionality), and given the total duration, it is required to find the interpolating trajectory that minimises the integral of the squared n^{th} derivative. The spline's parameter will represent time, and the total duration will be fixed, but the passage times for each knot will not be given. It may also be required that the trajectory be cyclic in which case the total duration will be the period of one complete cycle.

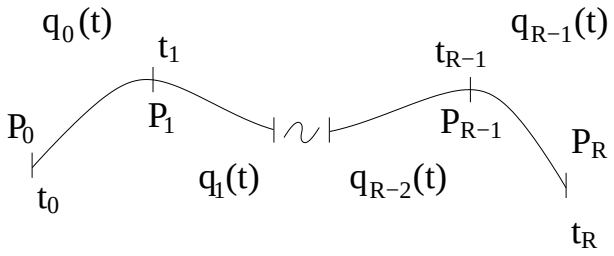

Figure 3.2: Polynomial spline

In fact handling cyclic trajectories is a trivial extension of the method presented below and for the sake of simplicity the non-cyclic case is first considered. Figure 3.2 shows a spline interpolating $R+1$ knots $\{P_0, \ldots, P_R\}$. The spline is segmented into R functions $\{q_0, \ldots, q_{R-1}\}$. The spline intersects P_i at time t_i and each spline segment q_i is parameterised over the range $[t_i, t_{i+1}]$. The complete spline q is thus defined:

$$q(t) = q_i(t) \text{ where } t_i \leq t < t_{i+1} \tag{3.4}$$

Let us first consider just one segment of this spline, q_i. P_i and P_{i+1} constrain the end points of q_i such that $q_i(t_i) = P_i$ and $q_i(t_{i+1}) = P_{i+1}$. The optimality constraint requires the minimisation of:

$$\int_{t_i}^{t_{i+1}} q_i^{(n)}(t)dt \tag{3.5}$$

In general this problem is solved by a polynomial of order $2n-1$. This may be proved but such was considered unnecessary since the solution itself

may be regarded as a constructive proof[3]. q_i is therefore a polynomial of degree $2n-1$ and the general problem must solve for the coefficients of all the segments $\{q_0, \ldots, q_{R-1}\}$ and the passage times $\{t_1, \ldots, t_{R-1}\}$ since only t_0 and t_R are known initially.

The solution, which is expounded formally below, proceeds as follows. The optimal knot passage times will be found by iteration, and it is assumed at each iteration that they are fixed. The coefficients of each polynomial q_i are expressed in terms of the function and its $n-1$ derivatives at the knots points, *i.e.* $\{P_i, P_i', \ldots, P_i^{(n-1)}\}$ and $\{P_{i+1}, P_{i+1}', \ldots, P_{i+1}^{(n-1)}\}$. The relationship is linear and can be expressed as a linear program. Expressing the coefficients in terms of the derivatives relates each adjacent segment because the spline and its derivatives must be continuous everywhere, including the knots, *i.e.* q_i and q_{i+1} share the same constraints at P_{i+1}. The cost function expressed in Equation 3.5 is continuous, and is also a continuous function of the polynomial coefficients, which are in turn continuous functions of the end point constraints. Hence, the partial derivatives of the cost function with respect to the end point constraints must be zero at the minimum value. These partial derivatives will be expressed algebraically as a linear program that may be solved for the end point constraints and thus the polynomial coefficients. The end point constraints affect not only one spline segment, but both adjacent segments, and in order to minimise the cost function over the whole spline it is necessary to compile the linear programs for each segment into a single amalgamated linear program. The amalgamated linear program expresses the knot values and derivatives in terms of non-linear functions of the knot passage times. It is therefore possible to constrain any subset of the knot values or derivatives prior to solving the amalgamated linear program by fixing their values in the linear program. The cost function itself may be evaluated quickly, and repeated solutions may be used to estimate the par-

[3]Hasan (1986) solved a related problem for the specific case of two point minimum jerk interpolations of fixed duration and included a proof according to the calculus of variations. The proof may be generalised to arbitrary order polynomials.

tial derivatives of the optimal cost with respect to the knot passage times. The optimal knot passage times may then be found by numerical gradient descent.

To begin a formal exposition, let us first consider an arbitrary polynomial segment p over an interval $[t_s, t_f]$ where $t_s < t_f \in \mathbb{R}$. For n^{th} order derivative minimisation p must be an $N = 2n-1$ order polynomial. p will be determined by $2n$ coefficients $\{c_0, \ldots, c_N\}$.

$$p(t) = \sum_{i=0}^{N} c_i t^{N-i} \tag{3.6}$$

$$\frac{d}{dt}p(t) = \sum_{i=0}^{N-1} (N-i) c_i t^{N-i-1} \tag{3.7}$$

$$\frac{d^k}{dt^k}p(t) = \sum_{i=0}^{N-k} \frac{(N-i)!}{(N-i-k)!} c_i t^{N-i-k} \tag{3.8}$$

$$\tag{3.9}$$

For convenience of notation let us define the following symbols:

$$\phi_{ki} = \frac{(N-i)!}{(N-i-k)!} \tag{3.10}$$

$$p_j = \frac{d^j}{dt^j}p(t_s) \tag{3.11}$$

$$p_{n+j} = \frac{d^j}{dt^j}p(t_f) \tag{3.12}$$

$$i, j, k \in [0, n-1] \tag{3.13}$$

ϕ_{ki} is the coefficient of $c_i t^{N-i}$ in $\frac{d^k}{dt^k}p(t)$ and the subscripted p terms are the values of p and its $n-1$ derivatives at the end points. Linear equations for the polynomial coefficients may be now be expressed in matrix form. The first n rows correspond to the initial constraints at t_s and the remaining n

rows correspond to the final constraints at t_f.

$$\underbrace{\begin{pmatrix} \phi_{00}(t_s)^N & \cdots & \phi_{0N} \\ \vdots & & \vdots \\ \phi_{(n-1)0}(t_s)^n & \cdots & \phi_{(n-1)N} \\ \phi_{00}(t_f)^N & \cdots & \phi_{0N} \\ \vdots & & \vdots \\ \phi_{(n-1)0}(t_f)^n & \cdots & \phi_{(n-1)N} \end{pmatrix}}_{T} \underbrace{\begin{pmatrix} c_0 \\ \\ \vdots \\ \vdots \\ \\ c_N \end{pmatrix}}_{C} = \underbrace{\begin{pmatrix} p_0 \\ \\ \vdots \\ \vdots \\ \\ p_N \end{pmatrix}}_{P} \tag{3.14}$$

$$TC = P \tag{3.15}$$

Having expressed a straightforward linear relationship between the polynomial coefficients and the end point constraints $\{p_0, \ldots, p_N\}$, the squared n^{th} derivative cost may be handled in terms of the end point constraints. The cost is:

$$F = \int_{t_s}^{t_f} \left[\frac{d^n}{dt^n} p(t)\right]^2 dt$$

The cost is a continuous polynomial function of the coefficients of p. Likewise $\{p_0, \ldots, p_N\}$ are continuous functions of the coefficients of p and hence the cost is also continuous with respect to the constraints. Since the cost is bounded below by zero it must have a minimum value with respect to $\{p_0, \ldots, p_N\}$. At the minimum value, the partial derivatives of the cost with respect to each constraint must be zero:

$$\frac{\partial F}{\partial p_k} = 0 \qquad k \in [0, N] \tag{3.16}$$

$$\tag{3.17}$$

Referring to Equation 3.14, the relationship of the cost to the end point constraints may be dealt with straightforwardly:

$$\frac{\partial F}{\partial p_k} = \sum_{i=0}^{N} \frac{\partial F}{\partial c_i} \frac{\partial c_i}{\partial p_k} \tag{3.18}$$

$$c_i = \sum_{j=0}^{N} T_{ij}^{-1} p_j \tag{3.19}$$

$$\therefore \quad \frac{\partial c_i}{\partial p_k} = T_{ik}^{-1} \tag{3.20}$$

This leaves $\frac{\partial F}{\partial c_i}$ to be determined:

$$F = \int_{t_s}^{t_f} \left[\sum_{j=0}^{N-n} \phi_{nj} c_j t^{N-n-j} \right]^2 dt \tag{3.21}$$

$$= \int_{t_s}^{t_f} \sum_{j=0}^{n-1} \sum_{k=0}^{n-1} \phi_{nj} \phi_{nk} t^{2n-2-j-k} c_j c_k dt \tag{3.22}$$

$$= \left[\sum_{j=0}^{n-1} \sum_{k=0}^{n-1} \frac{\phi_{nj} \phi_{nk}}{N-j-k} t^{N-j-k} c_j c_k \right]_{t_s}^{t_f}$$

This summation incorporates c_i only for $i \in [0, n-1]$ which is not surprising since the remaining n coefficients are eliminated by differentiation. N may therefore be replaced by $n-1$ in Equation 3.18. Defining m_{ij} as follows, the partial derivatives of F may be expressed concisely:

$$m_{ij} = \left[\frac{2\phi_{ni}\phi_{nj}}{N-i-j} t^{N-i-j} \right]_{t_s}^{t_f} \tag{3.23}$$

$$\frac{\partial F}{\partial c_i} = \sum_{j=0}^{n-1} m_{ij} c_j \tag{3.24}$$

Once again referring to Equation 3.14, and reconstructing the full system:

$$\frac{\partial F}{\partial c_i} = \sum_{j=0}^{n-1} m_{ij} \sum_{k=0}^{N} T_{jk}^{-1} p_k \tag{3.25}$$

$$\frac{\partial F}{\partial p_r} = \frac{\partial F}{\partial c_i} \frac{\partial c_i}{\partial p_r} \tag{3.26}$$

$$= \sum_{k=0}^{N} \left[\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} m_i j t_{jk}^{-1} t_{ir}^{-1} \right] p_k \tag{3.27}$$

Finally a matrix M may be defined, yielding a linear program with P as defined in Equation 3.15:

$$M_{rk} = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} m_{ij} t_{jk}^{-1} t_{ir}^{-1} \tag{3.28}$$

$$\frac{\partial F}{\partial p_r} = \sum_{k=0}^{N} M_{rk} p_k = 0 \tag{3.29}$$

$$MP = 0 \tag{3.30}$$

Inverting M and solving for P yields the end point constraints that minimise the squared n^{th} derivative. Some constraints will already be specified, such as the actual value of of polynomial at the start and end points, and possibly some or all of the derivatives. In order to constrain p_k the k^{th} row of M is replaced with a zero vector except for a 1 in the k^{th} column, and instead of a zero vector on the right hand side, the k^{th} element is set to the desired value of p_k. This trivially constrains p_k to take the desired value when the linear program is solved. So far all the constraints P_i^k have been scalar valued, and the interpolating spline has been one dimensional. It is straightforward to apply the method to splines of arbitrary dimension since the coefficients of the polynomial in each dimension are independent and may be found by treating each dimension separately.

Equation 3.30 is a linear program for the end point constraints of one polynomial. Each row corresponds to one partial derivative that has been equated with zero. In order to solve for all the polynomial spline segments it is required to minimise the squared n^{th} derivative over the *whole* spline, which has the following cost:

$$F_{total} = \sum_{r=0}^{R-1} \int_{t_r}^{t_{i+r}} q_r^{(n)}(t) dt \tag{3.31}$$

This expression may be solved in exactly the same way as for a single polynomial. The partial derivatives with respect to each knot constraint are equated with zero and the resulting equations form a linear program. If the costs for each spline segment are denoted $\{F_0, \ldots, F_{R-1}\}$ the partial derivative of the total cost with respect to knot constraint $P_i^{(k)}$ (the k^{th} derivative at the i^{th} knot) may be expressed as follows:

$$F_{total} = \sum_{r=0}^{R-1} F_r \tag{3.32}$$

$$\frac{\partial F_{total}}{\partial P_i^{(k)}} = \sum_{r=0}^{R-1} \frac{\partial F_r}{\partial P_i^{(k)}} = 0 \tag{3.33}$$

In fact $P_i^{(k)}$ appears in only F_{i-1} and F_i, the costs associated with the two spline segments adjacent to P_i. Thus,

$$\frac{\partial F_{total}}{\partial P_i^{(k)}} = \frac{\partial F_{i-1}}{\partial P_i^{(k)}} + \frac{\partial F_i}{\partial P_i^{(k)}} = 0 \tag{3.34}$$

All the partial derivatives may be expressed linearly as follows -note that $\overline{P_i}$ is a vector of knot constraints $(P_i \quad P_i' \quad \ldots \quad P_i^{(n-1)})^T$, and each derivative below is a vector partial:

$$\frac{\partial F_0}{\partial \overline{P_0}} = 0 \tag{3.35}$$

$$\frac{\partial F_0}{\partial \overline{P_1}} + \frac{\partial F_1}{\partial \overline{P_1}} = 0 \tag{3.36}$$

$$\frac{\partial F_1}{\partial \overline{P_2}} + \frac{\partial F_2}{\partial \overline{P_2}} = 0 \tag{3.37}$$

$$\ddots \ddots \tag{3.38}$$

$$\frac{\partial F_{R-2}}{\partial \overline{P_{R-1}}} + \frac{\partial F_{R-1}}{\partial \overline{P_{R-1}}} = 0 \tag{3.39}$$

$$\frac{\partial F_{R-1}}{\partial \overline{P_R}} = 0 \tag{3.40}$$

This formulation corresponds with the matrix in Equation 3.30. The i^{th} equation is the partial derivative of the total cost with respect to the vector of constraints at the i^{th} knot. Each derivative corresponds to the derivative expressed by Equation 3.30, at a particular knot and for a particular spline segment. Furthermore, since the derivatives are separable it is possible to combine all the terms into a single matrix expression as follows:

$$\mathcal{M}\mathcal{P} = 0 \tag{3.41}$$

$$\mathcal{P} \in \mathbb{R}^{n(R+1)} \tag{3.42}$$

$$\mathcal{M} \in \mathbb{R}^{n(R+1)} \times \mathbb{R}^{n(R+1)} \tag{3.43}$$

$$\mathcal{P}_{(in+k)} = P_i^{(k)} \tag{3.44}$$

$$\mathcal{M}_{(in+k)}\mathcal{P} = \left\{ \frac{\partial F}{\partial P_i^{(k)}} \right\} \in \mathbb{R}^{n(R+1)} \tag{3.45}$$

The format of the compound matrix $\mathcal{M}$ and compound vector $\mathcal{P}$ are shown in Figure 3.3. M_i is the matrix M from Equation 3.30 for the i^{th} polynomial segment. It is straightforward to build $\mathcal{M}$ progressively. After first initialising $\mathcal{M}$ to the zero matrix, $M_0, \ldots, M_{R-1}$ are calculated for each spline segment. The elements of each of these matrices are then added into $\mathcal{M}$ beginning at row and column rn for the r^{th} segment. If the spline is cyclic there is a spline segment from P_R to P_0, and the matrix M_R corresponding to the cost during this segment is also added into $\mathcal{M}$. In this case the indices into $\mathcal{M}$ must be treated modulo $n(R+1)$, because this spline segment relates the constraints at P_R and at P_0, which are not adjacent in $\mathcal{P}$. These elements wrap around into the four corners of $\mathcal{M}$ destroying its diagonal structure, so diagonal matrix methods may not be used in this case. In fact, it is sufficient to invert $\mathcal{M}$ using LU decomposition [Press (1992)]. As described above, in order to fix a given knot constraint the corresponding row of $\mathcal{M}$ is replaced with a unit vector, placing the unit in the appropriate column and the corresponding value in $\mathcal{P}$ set to the constraint value. The same pattern of constraints will be locked in each dimension since every dimension is a component of a position or a joint angle in a posture, so $\mathcal{M}$ may be constructed and inverted only once for the complete set of splines over all the dimensions.

The equations necessary to find the coefficients of the polynomial splines *given the knot times* have thus been expressed analytically. Unfortunately, it is infeasible to solve the resulting linear program analytically and it must be solved numerically. Likewise, it is not feasible to express the optimal passage times analytically. Fortunately, the solution space is very well behaved and gradient descent converges rapidly on the knot passage times. The time required to assemble and solve the compound matrix is minimal and the partial derivatives of the optimal cost with respect to the knot passage times may be estimated numerically. The total cost may be found by summing

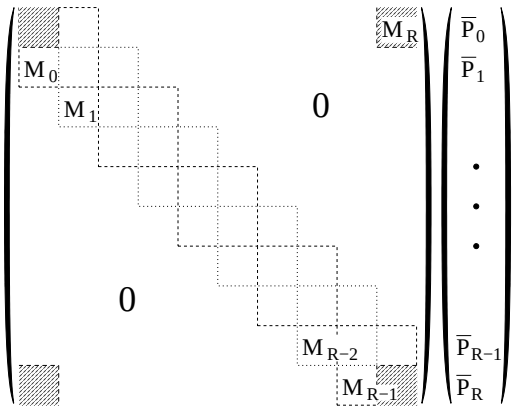

Figure 3.3: Compounded partial derivatives $\mathcal{MP}$

the cost over each spline segment, and for each dimension of the spline. The passage times were initialised by dividing the duration equally among them. Intuitively, the convergence is rapid because the n^{th} derivative minimisation constrains the spline to have as smooth and simple a trajectory as possible, and perturbing the optimal passage times increases the optimal cost monotonically. Convergence characteristics are discussed in Section 5.3.2.

3.3.2 On-line formulation

Minimal n^{th} derivative trajectories can in fact be generated on-line. Hypothetically, the general method above could be applied on-line were the driving processor fast enough. By far the most time consuming factor is the optimisation of knot passage times. If the passage times are fixed, for example, in the case of point-point movements of fixed duration, or according to some estimation process, then the compound matrix could be formulated and solved on-line during motion production. Alternatively, estimated passage times could be used to find the optimal trajectory, and the iterations necessary to find the optimal passage times performed in parallel to motion production. The trajectory synthesised during the first few moments would thus be specified by a non-optimal interpolating spline, but would converge

smoothly to the optimal spline during motion production.

Putting these high level solutions aside, it is possible to formulate all minimal n^{th} derivative point-point motions straightforwardly on-line. A $2n-1$ order polynomial is sufficient to yield a minimal n^{th} derivative. At any instant during motion production the immediate position, velocity, acceleration and higher derivatives are known through sensory feedback in the case of a robot, or by examining the state of a simulator. If the target position and trajectory derivatives at the target point are specified the optimal trajectory is fully specified and may be solved quickly using Equation 3.15. For point-point motions it is natural to assume that the trajectory come to rest smoothly at the target point, and thus all the derivatives at the target are zero. Other choices are straightforward to substitute. The target position may also be perturbed during motion yielding a continuous, minimal n^{th} derivative interpolation from the immediate point in the trajectory to the new target point. As an example, the minimum jerk point-point trajectory from a stationary initial position p_0 to a stationary final position p_1 in d seconds is described by the following quintic polynomial:

$$p(t) = p_0 + (p_1 - p_0)(6t'^5 - 15t'^4 + 10t'^3) \tag{3.46}$$

$$t' = \frac{t}{d} \tag{3.47}$$

The end point derivatives are all zero, but the motion amplitude $p_1 - p_0$ and duration d are easily parameterised.

Chapter 4

Implementing the motion production environment

Practical research

The motion production environment is a set of software tools for motion production, editing, analysis, and profiling. This chapter describes the environment itself (Section 4.1) and its component software tools (Sections 4.2-4.4). The significant algorithms and algorithmic optimisations are presented.

Since psychovisual testing required comparisons with genuine human motion, it was necessary to 'motion capture' human performances. The Optotrak capture system, data processing techniques, and motion post-processing involved in motion capture are also presented, in Section 4.5.

4.1 Motion production environment

The motion production environment was designed to accomplish all of the animation and research work in a coherent unified system. It was designed according to the specification in Figure 3.1 on page 59. According to this scheme the environment is divided into three areas: motion generation, editing & analysis, and visualisation.

The motion production environment comprises three software tools: `mctool`, `mltool` and `mdtool`. `mctool` was used for generating motion. `mltool` was used for manipulating motion log files and exporting motions to a commercial rendering package. `mdtool` was used for visualising motions and various statistics. These three software components and the most significant algorithms are described in this chapter.

User interaction with the software was achieved with a command line interface model. Apart from `mdtool` which required a graphical output window, it was considered unnecessary to provide a graphical user interface (GUI). It is typically a time consuming task writing GUIs, and since the focus of this research was the motions and algorithms themselves rather than animator interactions with the software, it was sufficient that the author be able to create and investigate motions efficiently. All commands were supplied in a single command line invocation of the relevant software tool. Motion specifications were made using short text files listing the knots constraining a hand trajectory, and these files were supplied as command arguments. Only `mdtool` provided an interactive control mechanism by means of keyboard shortcuts and mouse dragging operations to facilitate easy visualisation.

Interactions between each software tool were facilitated by storing motions in a special '`traj`' file format described below. Motion visualisation was accomplished using a commercial rendering package, Poser, or by controlling a humanoid robot so as to perform given `traj` file. The `traj` format, the export of motions for rendering, and the performance of `traj` files using the humanoid robot are described in the following sections.

4.1.1 `traj` file format

The `traj` file format was used to store generated motion files and to facilitate editing and interchange between the software tools of the motion production environment. Motions are stored as ordered sequences of discrete samples. Each sample contains at least a description of the relevant joint angles of the humanoid figure, and records the duration and sampling frequency of the

motion. A complete log-file contains for each joint: angle (rad), angular velocity (rad/s), angular acceleration (rad/s^2), force (N), feedback force (N), load (N), target angle (rad), target velocity (rad/s), and target acceleration (r/s^2). Data describing each link is also given: the link position in each axis (m), and the target hand position in each axis (m). The force term gives the total force generated at a linear joint actuator, and the feedback force term gives the force component generated by the PD-servo controller[1]. The subtractive difference is the feed-forward force component typically calculated by inverse dynamics, which yields a faster and smoother control than a PD-servo alone.

The data is stored in binary with a short ASCII header. The actual file layout is an extension of a simple file format designed at ATR for storing humanoid robot motions. As such, the motions generated using the software described in this thesis are compatible with existing software maintained at ATR.

The file header contains, in ASCII numerals separated by whitespace, the file size in bytes, the sample size in bytes, the number of samples, and the sampling frequency in Hertz. Following, also in ASCII, is a sequence of pairs of strings, each containing the name of a quantity and the units by which it is measured. The sample data is then stored with each sample in sequence, and each quantity given in the order specified in the header. All quantities are stored as binary floating point numbers. The 9.18 second `traj` file, `xpt_V_motion7.traj` (a motion used for psychovisual testing), for example, contains all the information described above sampled at $70Hz$ and is $284KB$.

4.1.2 Motion rendering

The motions generated with the techniques described in this thesis were rendered with the commercial package Poser. It was necessary to render

[1]The humanoid robot DB is typically controlled by setting target joint angles that are satisfied by a spring-damper system (PD-servo).

motions to facilitate visual analysis, psychovisual testing and demonstrate the functionality of the developed animation system.

Due to the animation focus of this research no decisions were taken that should limit the possible methods of rendering other than the involvement of articulated humanoids. MetaCreation's Poser package was chosen because it is specialised to rendering a variety of interchangeable humanoids. It is relatively straightforward to convert motion to a format compatible with Poser, as will be seen in Section 4.2.1. It is also possible to generate high quality images of humanoids without spending a large amount of time and money obtaining and manipulating a humanoid model. This was important because it has been shown psychovisually that humans' perception of computer animated human motion is influenced by the quality of model and rendering [Hodgins et al. (1998)]. The humanoids in Figure 4.1 on page 99 were generated using Poser, and animation stills are displayed in Section 5.1.

4.1.3 Humanoid robot motion production

Besides computer graphic rendering, motions were performed by a humanoid robot. Issues of robotic motion production were thus explored, and recordings were used for psychovisual testing.

A DB task capable of performing `traj` file motions was developed according to the task based control paradigm described in Section 2.3.3. The functionality and main algorithmic structures are summarised below.

- **Cartesian trajectory playback** of motion files that specify only the Cartesian hand position. This required on-line inverse kinematics meaning that optimal joint angles and kinematic consistency could not be guaranteed.

- **Kinematic playback** of motion files that specify the complete joint angle kinematics of the figure. Kinematic consistency could be guaranteed up to the limit of sensor and force actuator resolution.

- **Batch playback** of a series of motion files, thus facilitating the performance of the complete set of motions required for psychovisual testing. A log file indicating the sequence of motions performed, and the onset and completion times for each motion was also output.

- **Trajectory visualisation** including the actual and target trajectory during motion production. This was only available when the task was applied to the robot under simulation.

- **Smoothed motion** by means of inverse dynamic estimates of required forces. These estimates were supplied as feed-forward force terms added to the feedback force terms calculated according to spring-damper systems.

- **Simultaneous motion logging** during motion playback. Motion performances could thus be analysed off-line to assess the fidelity of motion production. Logged motions included the actual joint angles and link positions, the target angles and hand position, joint angles' first and second derivatives, force sensor readings, feedback and feed-forward force terms.

The source code, `kinematic_playback_task.c`, may be found on the CD accompanying this thesis. Recalling the specifications in Section 2.3.3, robot task code must include initialisation and runtime interrupt functions. These must be robust against overlapping executions since in particular, the runtime function is executed at $420Hz$ whether or not its previous iteration is complete. This frequency was the chief motivation in selecting $70Hz$ to store motion files. As a divisor of 420, samples may be taken at every fourth runtime iteration but are frequent enough to facilitate smooth stable motion performance.

The initialisation function checks that no other task is running, initialises the display routines if the task is running under simulation, prepares the robot in a default posture, and selects the motor control method, *i.e.*, with or without inverse dynamic smoothing.

The runtime function is guided by a six phase semaphore. Switching the phase control variable ensures that even if it does not complete in $\frac{1}{420}^{th}$ of a second, subsequent iterations will not lead to invalid states. A list of `traj` files to play in included in the task code itself. The phase sequence runs as follows:

1. **Completion test:** maintain a count of the number of motion files remaining to play, and signal task completion if this is zero.

2. **Load trajectory:** motion buffers are cleared and loaded from the appropriate `traj` file. The motion log buffer is cleared and prepared for input.

3. **Posture initialisation:** move the robot to the posture in the initial frame of motion and maintain the posture for a short period to ensure absolute mechanical stability -total time 4 seconds.

4. **Transition to playback phase:** signal change of phase into playback mode. Internal clocks are stored to facilitate accurate time correlation with the `traj` file.

5. **Motion playback**: calculate the target posture according to the `traj` file. If `traj` file contains only Cartesian hand coordinates, some inverse kinematic iterations are performed on the current posture to move the hand to the new target position, and thus calculate target joint angles.

6. **Playback completion:** the target posture is frozen, timing information and motion log file are output, and the completed playback counter is incremented. The phase counter then set back to 1.

Besides these serialised operations, the runtime function maintains estimates of each joint angle's velocity and acceleration by calculating differences. All of the target joint angles are tested against joint range limits and capped if necessary, and the feed-forward force is calculated by inverse dynamics if appropriate.

4.2 Motion logging tool

The motion logging tool, `mltool`, was used for manipulating and extracting information from `traj` files. An invocation with no arguments (or badly formatted arguments) yields the following command summary:

```
mltool {[-*]} <source file> [<dest file>]

 -Pj          Print all joint angle data from input file to stdout
 -Pjt[int]    Print target joint angle, velocity and accelerations
              Supplying an integer prints the data for the given
              joint index only
 -PT          Print instantaneous sum torque, and for each joint
              individually
 -PD          Print instantaneous sum torque change, and for each
              joint individually
 -Psj         Print a summary of joint range and velocity information
 -Pjv<float>  Print joint angle velocity smoothed over a window of
              given duration
 -Pt          Print trajectory data to stdout
 -Ptt         Print target trajectory data to stdout
 -Ptv<float>  Print trajectory velocity smoothed over a window of
              given duration
              use -f<float> to specify sampling frequency for smoothing
              o/w 30Hz
 -Ps          Print a summary of the input file
 -Pb[float]   Print the input file in BVH format at the given frequency
              If no frequency is specified 30 Hz is chosen by default
 -Pa          Print named and logged "arbitrary" data to stdout
 -Pf          Print a summary of signal magnitude according to frequency
              bands

 -v           Verify the input file
 -l           Impose joint limits on input file by clipping and store
              as output file
 -L           Impose joint range limits by linear resizing per DOF
 -p           Calculate link positions using forward kinematics and
              add to logfile
```

```
 -d          Duplicate source trajectory twice as a loop
 -c <float> <float>
             Cut out a subsection of the input log and store as output
 -b <float> <float>
             Buffer the sequence with a fixed width at the start and
             a fixed ending time
 -S <int> <int>
             Smooth a joint angle with a given window width
 -F <int>    Apply digital Chebyshev filtering to joint angles
             0 = 5Hz LP, 1 = 5Hz HP, 2 = LP 10Hz, 3 = HP 10Hz,
             4 = LP 20Hz, 5 = HP 20Hz

Flags
 -q          Quiet operation
 -r          Toggle reverse byte order of input and for output
 -ri         Toggle reverse byte order of input
 -ro         Toggle reverse byte order for output
 -f<float>   Resample output at given frequency
 -s<float>   Scale temporally
```

The features of this tool include several functions for extracting statistics from motion files and printing the results. Output is formatted in ASCII and printed to `stdout` (the interface window). Running in a UNIX type environment the output may be straightforwardly piped into any file for subsequent import into a graph plotting application, *etc.*

Print functions include: all joint angles, all or a single target joint angle and its two derivatives, sum squared torque or torque change for each joint in each sample, joint ranges and velocity ranges for each joint over the whole movement, Cartesian hand coordinates, Cartesian hand coordinates averaged over an arbitrary number of samples, all non-standard data included in the `traj` file, a spectral frequency analysis indicating the frequency bands containing most joint angle motion, a summary of motion file length and sampling frequency, *etc.*, and finally, a `BVH` file for the whole motion suitable for import into Poser. The conversion to `BVH` format is described in the following section.

The frequency analyses were useful in determining the approximate frequency of noise in motion captured movements and mechanical oscillations present in robot motion. Likewise, the smoothed trajectory output was useful for visualising the hand trajectory and its derivatives without this noise. The spectral analysis was approximated by comparing the energy outputs of Chebyshev filters. Low and high pass 5, 10 and 20 Hertz filters were constructed, each with fourteen coefficients. Chebyshev filters were selected for their relatively sharp frequency cut-off response. `mltool` also provides features to apply either of these six filters to the joint angles in a motion file, and also to smooth joint angles by averaging over a parameterised number of samples.

Other editing functions include enforcing joint angle limits, calculating link positions from joint angles and incorporating them into the `traj` file, duplicating the motion twice, cropping motion down to an arbitrary subsegment, and buffering the motion with a steady posture at the start and end of motion. Joint angle limits could be enforced by clipping, or linearly rescaling the range of joints exceeding their limits. The latter option was preferred since the former leads to discontinuities in angular velocity. Joint angle limiting was necessary since some motion production methods did not rigidly enforce the joint limits, and in particular, forward dynamic simulation may yield joint angle violations. Such violations can over-extend joints or cause high frequency oscillations that are mechanically damaging to a robot. Duplication, cropping and buffering were used to modify some motion sequences and standardise their durations for psychovisual tests.

Finally, motions could be re-sampled at arbitrary frequencies and scaled temporally facilitating a high degree of generality. The order of floating point byte interpretation or storage could be specified to overcome the lack of a coherent standard across processor architectures.

The source code, `mltool.c` was not included in this thesis since it is over a thousand lines. Furthermore, it is based heavily on a library of code for handling motion logs: `log_tools.c` and `log_tools.h` which together are

over three thousand lines. These files may however be found on the CD accompanying this thesis.

4.2.1 BVH format motion files

The BVH file format was designed by BioVision and is intended for describing motions of articulated figures. Since it has a straightforward specification and may be imported by Poser, it was used to facilitate the rendering of motions.

BVH files are in ASCII, and begin with a header that describes the kinematics of an articulated figure. A typical header begins as shown below, and contains a hierarchy of nested joint definitions. Each 'JOINT' contains a link length and the number of corresponding parameters, called 'CHANNELS'. At the end of the header the number of frames, and sampling period are given. Finally, each sample is output as a single line of ASCII floating point values ending in a carriage return. These numbers correspond sequentially to the channels specified in the header. The initial configuration (with all joint angles equal to zero) is determined by the unmodified offset vectors in each joint.

```
HIERARCHY
ROOT Hip
{
  OFFSET 0.00  0.00  0.00
  CHANNELS 6 Xposition Yposition Zposition Xrotation Zrotation Yrotation
  JOINT Abdomen
  {
    OFFSET 0.000000 2.159 0.000000
    CHANNELS 3 Xrotation Zrotation Yrotation
    JOINT Chest
    ...
    ...
  }
}
Frames: <number of frames>
```

```
Frame Time: <sampling period>
```

It is a straightforward process to output a `traj` motion file as a `BVH` file since each `traj` sample may be output as a corresponding `BVH` frame. However, to ensure compatibility with Poser, a standard Poser articulation most closely matching the kinematics of DB was selected. It was possible to modify the link length parameters but not the topology. Figure 4.1 shows how the link parameters were set in order to establish a correspondence with DB's kinematics. Since each of the joints in the `BVH` humanoid has three DOFs there was a considerable redundancy. The DOFs coloured red were simply set to zero in each frame.

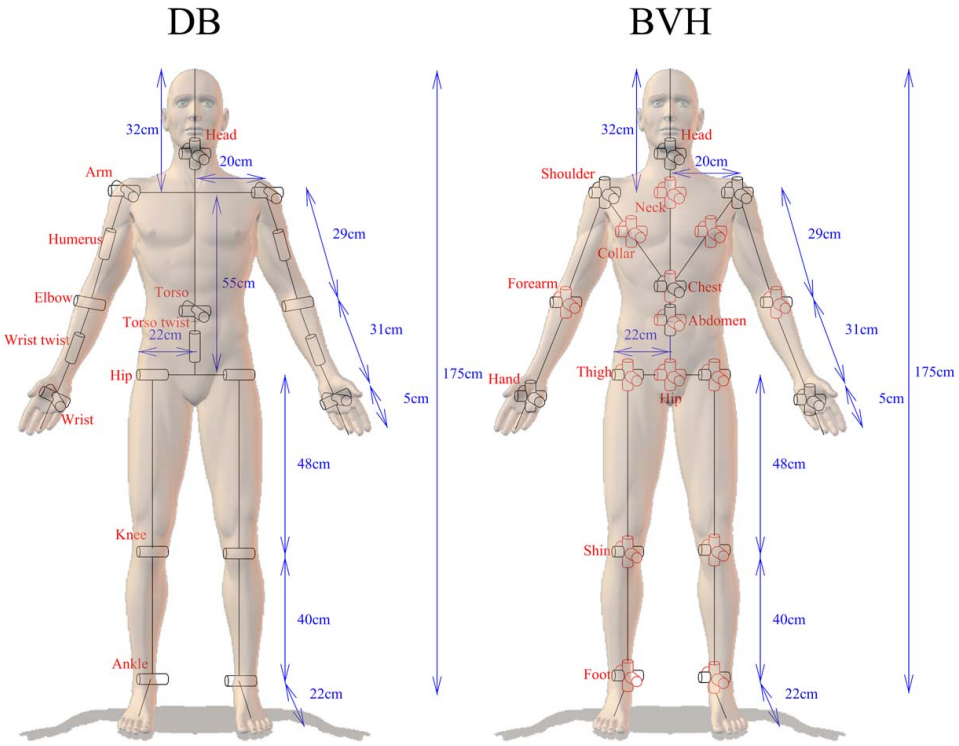

Figure 4.1: Kinematic specifications of DB and `BVH` humanoids

A further complication was that DB's joint angles are specified extrinsically, *i.e.*, in absolute angular coordinates irrespective of the joint. The `BVH`

joint angles however, are interpreted intrinsically by Poser, *i.e.*, in coordinates that depend on the joint, corresponding to anthropomorphic quantities such as flexion and adduction. It was therefore necessary to flip the orientation of sixteen joints: the torso and head twist, elbow, hip and ankle bend, shoulder flexion and wrist adduction, left shoulder adduction, right humeral twist, and right wrist flexion and twist.

An example `BVH` file with the complete joint hierarchy may be found in Appendix B.

4.3 Motion creation tool

The motion creation tool, `mctool`, was developed as an off-line tool. Motion specifications were loaded from text files, and synthesised motion was stored as `traj` files.

The command reference for `mctool` is long and is not duplicated here, although it may be found in the source code `mctool.c`. The command syntax requires a number of options specifying the required function, a source file, and optional destination file name.

Possible options include motion generation using any of the fourteen models already presented. In this case an ASCII file specifying a series of knot positions for the hand must also be given.

Other options include explicitly setting spring and damper parameters for virtual trajectory motions, testing the forward and inverse dynamics algorithms' speed, generating a randomised cluster of reachable hand targets in order to assess the workspace region, augmenting a `traj` file combining only hand positions with the corresponding joint angles using inverse kinematics, generating a sequence of postures corresponding to every iteration of the inverse kinematics algorithm to assess its convergence properties, setting the convergence rate for inverse kinematics, setting the maximum number of inverse kinematic iterations, setting the knot insertion ratio for dynamically optimal motions (see Section 3.2.8), setting the knot perturbation learning

rate, and toggling the byte order for loading and storing floating point values.

The most significant software components are those for generating motion. These methods were discussed theoretically in Chapter 3 and include N^{th} derivative minimisation, torque minimisation, torque change minimisation, and minimum torque/torque change posture generation. The practical research issues of implementation are discussed below for each of these problems.

4.3.1 Minimum N^{th} derivative splines

The general spline optimisation method was presented formally in Section 3.3.1. Two data structures with corresponding functions were used to implement the method. These were the 'general polynomial solver' (GPS) that handled the calculations required of a single polynomial spline segment, and the 'general vector solver' that handled an array of GPSs constituting all of the segments of a polynomial spline in arbitrary dimensions.

All the source code may be found in the files `numerical_computation.h` and `numerical_computation.c`, totalling just under four thousand lines. Evidence supporting the correct convergence, the adequacy of the numerical methods used, and timing results may be found in Section 5.3.

General polynomial solver

The GPS data structure is parameterised with respect to the order of the polynomial. It stores the polynomial coefficients, the end point constraints (function value and derivatives), the start and end times, and a few flags indicating the internal state of this data.

The following functions are based on the GPS structure. The polynomial or its derivative may be evaluated at a given time instant. The cost of the polynomial segment according to the squared N^{th} derivative metric may be evaluated. Given the end point constraints the polynomial coefficients may be calculated. This requires the formulation and solution of a linear

program, which was accomplished with the Numerical Recipes LU decomposition method using the published functions `ludcmp` and `lubksb`. Finally, the timing matrix M relating the optimal end point constraints, the fixed constraints, and the segment's start and end times may be calculated (see Equation 3.28). All calculations except the solution of the linear program were formulated analytically.

General vector solver

The GVS data structure is parameterised with respect to the order, N, of derivative minimisation, the number of spline segments, and the number of dimensions. It stores the knots, the knot passage times, the spline duration, the compound timing matrix and a set of GPSs as well as a few flags indicating the internal state of the data. There is a GPS for each spline segment in each dimension.

Besides the memory handling functions for creating and destroying GVS structures, the following functions are based on the GVS. The spline or its derivative may be evaluated at an arbitrary time instant. The knot values may be specified and an arbitrary set of knots may be locked -typically all of the function values and the initial and final derivatives. The solution for the optimal knot constraints and corresponding GPS polynomials may be calculated given the passage times for the knots. The optimal passage times and corresponding optimal polynomials may also be calculated.

The complete solution process begins by estimating the knot passage times (by dividing the duration equally between each segment). The optimal polynomials are solved and the N^{th} derivative cost evaluated. Each knot passage time is then perturbed by a small offset and the optimal solution and solution cost evaluated. In this way, the partial derivatives of each knot passage time with respect to the final cost may be found. By finding the derivative of the vector of knot times with respect to total solution cost, gradient descent may be used to find the optimal knot passage times.

For a given set of passage times the optimal spline is calculated as follows.

The appropriate start and end times are supplied to each GPS. The timing matrices for each GPS are calculated and assembled into the compound timing matrix stored in the GVS. The compound timing matrix forms a linear program for the optimal knot constraints. This is solved using LU decomposition and the appropriate knot constraints are supplied to each GPS. Each GPS is finally solved for its polynomial coefficients.

Algorithmic optimisations

Given fixed knot passage times the solution is almost entirely analytical, with the exception of two linear programs. The time required is dominated by these numerical methods. The time complexity of LU decomposition is $O(n^3)$ for an $n \times n$ matrix, and the solution of a set of linear equations given the decomposition is $O(n)$. For a quintic spline with five knots n is $6 \times (5+1)/2 = 18$ but the constant factor is very small and the process is very fast. However, the solution must be calculated many times in order to iteratively optimise the knot passage times. A number of algorithmic optimisations were therefore implemented.

The polynomial coefficients for a GPS were only recalculated if the knot values, or the start and end time were adjusted. In fact this may be exploited very often since the majority of solutions are calculated with only one knot time perturbed (during the calculation of the derivative of the passage time vector). This was implemented by requiring that changes to the knots or endpoint times be made through a special function that also logged the change and thus maintained an internal flag indicating whether or not the polynomial coefficients were in correspondence with the knot and timing parameters.

Coherence between subsequent compound timing matrices was exploited. While any change to the compound timing matrix necessitates re-computation of the LU decomposition the compound timing matrix itself need not be fully recomputed at every stage. Instead, only the timing sub-matrices corresponding to polynomial segments that have changed since the last construction

must be changed.

Finally, the LU decomposition method may be used to calculate a matrix inverse from scratch, or iteratively to improve an existing solution. If the compound timing matrix was only changed by one or two sub-matrices the iterative method was adopted, since it typically converged within two iterations.

Coherence between subsequent solutions in the same dimension was exploited. Especially when the spline had many dimensions (such as a joint angle trajectory) some of the dimensions' knot constraints did not change from one solution to the next. Provided the passage times do not change there is no need to recalculate any of the GPSs for such a dimension. Furthermore, the compound timing matrix need only be inverted once for the whole set of dimensions since they are all constrained by the same passage times.

4.3.2 Torque and Torque change minimisation

The production of full body human motions that are optimal with respect to dynamic metrics such as torque and torque change (MT & MTC) is an extremely difficult problem.

The solution space is huge since in general, every DOF may vary over the entire duration of motion. If motions are specified in terms of hand positions there is a high degree of kinematic redundancy that must be resolved by the cost minimisation. The space is also ill conditioned for numerical solution. Slight perturbations in a single DOF can lead to large changes in torque, making the solution space stiff and necessitating small learning parameters. Furthermore, the near optimal solutions form narrow filaments in solution space because dynamically optimal motions typically exploit the inertial characteristics of an articulated figure in motion, and thus require precise coordination between all the joints of a limb, and between limbs.

A hypothetical experiment reveals that dynamic optimisation over the complete range of DOFs is infeasible. The BFGS quasi-Newton optimisation

method is regarded as a reasonable approximation to, if not lower bound on the time required to solve problems of this nature. The BFGS algorithm typically converges within $30n$ iterations for an n parameter optimisation. Each iteration requires either 2 or 3 cost evaluations and 1 derivative evaluation. Since an analytical formulation of the cost function for the entire motion was not available, the derivative must be numerically calculated, requiring $6n$ cost evaluations. A typical solution thus requires $O(180n^2)$ cost evaluations.

The cost function must be calculated by integrating torque costs over the entire movement and thus depends on the resolution of numerical integration. The total time required by numerical integration may be estimated in the range $\frac{1}{100}^{th}$ to 1 second. Recalling that the humanoid has 30 DOFs, a single knot requires 30 optimisation parameters. Supposing that 1 knot is varied, and integration requires $\frac{1}{100}^{th}$ of a second the BFGS solution time is 1620 seconds or half an hour. Even with a limited number of knots, say 10, the time requirement is two days. Since the primary focus of this research was to compare the motion production models computationally and psychovisually, an overnight computation delay is undesirable but would be acceptable. The above estimate however, was based on generous assessments of the number of iterations required to solve the problem and the time required by numerical integrations, yet it presented a delay greater than that acceptable to a reasonable research and development schedule.

It was therefore necessary to make some concessions in order to facilitate convergence. Rather than optimise in joint angle space it is possible to optimise in Cartesian trajectory space. This reduces the dimensionality of each knot by a factor of 10 and according to the estimates above, facilitates computation of a trajectory specified with 10 knots in half an hour. However, this constrains the joint angles uniquely according to the hand position. This is undesirable since the resulting motion is not only highly dependent upon the inverse kinematics method, but cannot utilise kinematic redundancy to generate relaxed inertia guided motion.

It is necessary to reduce the dimensionality of the search space since it affects convergence time quadratically. Rather than drop to Cartesian dimensions, however it is possible to partition the joint angle search space and perform a two stage optimisation. This concession facilitated convergence but strictly speaking invalidates the absolute optimality of the final motion and is discussed along with other criticisms in Section 6.2.3.

The implementation of the torque and torque change minimisation models is discussed below, followed by the algorithmic optimisations that were necessary to obtain convergence in a reasonable time.

Implementation

Torque and torque change optimisation were both implemented using the same process. Given a set of Cartesian knots on a hand trajectory the optimal motion was derived as follows:

1. **Build kinematic postures** corresponding to each Cartesian knot using inverse kinematics.

2. **Fit a minimum angular jerk spline** through these postures.

3. **Increase the knot density** by sampling the joint angle spline a given number of times between each initial knot (the knot insertion density).

4. **Numerically optimise** the motion, supplying all the knots as optimisation parameters to a two stage numerical process.

It was necessary to represent the joint angles using some continuous representation based on knot values, *i.e.*, a spline. Linear splines have discontinuous velocity and acceleration profiles, and quadratic splines have discontinuous acceleration profiles. Both lead to infinite torque impulses and are therefore not a suitable representation for a torque sensitive motion. Quintic splines were therefore required. It was also hoped that the correspondence between minimum jerk and minimum torque observed by some authors in

the human motion production literature would mean that the minimum jerk splines were a good initial approximation to minimum torque.

Increasing the knot density facilitated a greater resolution of control than was feasible with the initial knot set corresponding directly with the motion specification. All of the knots in the increased knot density spline were optimisation parameters. This meant that the kinematic redundancy in hand positioning, and even the hand position itself were incorporated into the search space. It was therefore necessary to incorporate the hand position targets (the original Cartesian knots) into the cost function, so that cost minimisation would constrain the hand to pass through the specified targets. The sum squared distance between the hand and each target was therefore scaled and added to the cost function.

Unfortunately this constrained the passage times of the hand through each target since it was infeasible to calculate the instant of closest approach to each reaching target individually. However, the passage times were initialised optimally with respect to minimum jerk joint angle splines, and thus good approximations to the optimal passage times with respect to torque or torque change.

For a given joint angle knot set the cost function was calculated by first fitting a minimum angular jerk spline through the knots. Since the passage times were fixed this was a negligible computation. The postures along the spline were then sampled and inverse dynamics was used to calculate the torque at each joint given its angle, velocity and acceleration. These torques were squared and summed for torque minimisation, or subtracted from the torques in the previous sample, squared and summed in the case of torque change minimisation. This numerical integration process dominated the computation time. The final cost function thus incorporated torque/torque change cost and hand positioning error.

Numerical optimisation itself was performed using the Simplex algorithm [Hartley (1985)]. Gradient descent was first attempted but was found to oscillate around possible local optima, and it was difficult to select a suitable

learning parameter adaptively. The Simplex algorithm maintains a simplex of $n+1$ solutions for n parameter optimisation and requires only function evaluations (and not gradients). An initial solution is used to construct the $n+1$ solutions by duplicating the solution vector but for each copy setting at least one parameter value to an extreme value. The $n+1$ solutions thus form a simplex in n dimensional with the initial solution somewhere inside it. The Simplex algorithm is rarely the most efficient method of solving numerical optimisation, but is initialised with a broad sampling of the search space, has fairly consistent convergence properties and may be restarted to check the validity of a solution.

Computational experiments into the timing and convergence properties of this optimisation process may be found in Section 5.3.3. Since reasonable results were obtained with the Simplex method, an implementation of more complicated numerical methods such as BFGS was not attempted. It was considered unlikely that such methods would yield more than a 2-10 fold decrease in computation time, and therefore offered only a benefit of convenience.

Algorithmic optimisations

The most significant algorithmic optimisation was to reduce the dimensionality of the search space. The kinematic specification of the humanoid used throughout this work has 30 DOFs. As will be discussed in Section 5.1.1 the head was constrained to orientate automatically to gaze at the hand position. The legs were also static throughout motion. The remaining DOFs were thus reduced to the arms and back. These 17 DOFs are still unfortunately too numerous to facilitate convergence in a reasonable time. The DOFs were therefore partitioned into two groups.

Figure 4.2 shows the left and right DOF sets. The right set includes the back, right arm and right wrist. The right set facilitates the full body movement, arm and wrist orientation necessary to reach a given target. The left set is restricted to the left arm which is free to move in an unconstrained

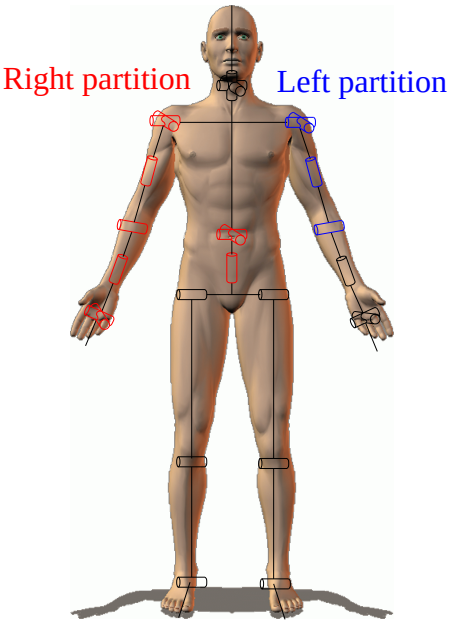

Figure 4.2: Left and right DOF set partitions for dynamic optimisation

manner. For dynamically optimal motions the left arm is typically used to balance and store inertia. The Simplex optimisation procedure was first applied to the right DOF set, while the left arm remained locked in the default orientation at the side of the body. In the second phase the optimal right DOF functions remained fixed while the left DOFs were optimised, reducing the total torque in the left DOFs by allowing the arm to swing and bend, compensating the main body and right arm motion.

This optimisation was sufficient to yield convergence in the search for dynamically optimal motions with the Simplex method. The gradient descent method first attempted however, necessitated the calculation of partial derivatives of the cost function. Since the time required to calculate the cost was dominated by numerical integration, it was desirable to optimise this algorithmic process. This was possible since the partials were calculated by perturbing a single DOF in a given solution, *i.e.*, shifting the location of a single knot. The solution cost without the perturbed DOF was then subtracted from the cost with the perturbed knot to approximate the local gradient. It

was therefore sufficient in most cases to calculate the integral only over the region affected by the perturbation, *i.e.*, between the two adjacent knots.

4.3.3 Minimum torque posture methods

The minimum torque posture models (MTP & MTPVT) were designed to generate reasonably high fidelity human motions with minimum computational cost. In order to replicate the qualities of human motion it combines the characteristics of several human motion models: minimum torque, minimum torque change, minimum jerk and, optionally, virtual trajectory control.

It was hoped that minimum torque would approximate minimum effort. Minimum torque change should likewise approximate the natural dynamics of the body in motion and yield minimum effort motions. Minimum angular jerk interpolation should approximate minimum jerk hand trajectories, and the angle interpolation also reduces the burden of repeated inverse kinematics solutions. The use of PD-servos when the resulting motion is used as a virtual trajectory should precipitate the double peaked velocity profiles and velocity hooks caused by overshooting and inaccuracy of hand placement observed in human motions.

The primary optimisation exploited by the method is to calculate torque and torque change costs at only the knots in the original motion's specification. This avoids the computational burden of repeated numerical integrations experienced by the minimum torque and torque change models presented above.

Inverse kinematics with torque & torque change minimisation

The fundamental component in the implementation of the MTP and MTPVT models was dynamically optimal inverse kinematics. This was based on the improved extended Jacobian method of IK described in Section 2.2.2.

The improved extended Jacobian IK algorithm is iterative. The inputs at each iteration include the current joint angles, a default posture, and a Cartesian offset to the current hand position. The default posture is used to

overcome the kinematic redundancy by constraining postures to deviate as little as possible. The default posture is therefore typically a relaxed neutral pose with all the limbs in a natural orientation.

The default posture was used to incorporate dynamic optimisation into the IK algorithm. Default postures encouraging dynamically optimal motions were calculated between IK iterations.

A cost function (described below) was used to evaluate the dynamic cost of the current IK solution posture. By numerically estimating the partial derivatives of the cost with respect to each joint angle the gradient was assembled. This gradient vector was scaled and subtracted from the current IK solution posture. The resulting posture was similar to the current solution but more dynamically optimal. This improved posture was supplied as the default posture for IK iteration. Thus, while the hand position was solved by IK iterations, the kinematic redundancy was resolved by compelling the posture to be dynamically optimal.

The cost function incorporated sum squared torque and sum squared torque change between the postures for each knot. These terms were calculated by first using forward kinematics to calculate the link positions given a set of joint angles, and then performing inverse dynamics to calculate the necessary torques. Since torque change calculation required the results for each adjacent posture, all of the postures were optimised simultaneously. A single IK iteration was thus performed on each posture in turn, rather than solving each posture independently.

Having calculated a dynamically optimal set of postures corresponding to each knot in the motion specification, the postures were interpolated with minimum angular jerk. The resulting motion was either output (MTP) or used as a virtual trajectory guiding a PD-servo spring-damper system under dynamic simulation (MTPVT).

Algorithmic optimisations

It was discovered by experimentation that the IK iterations had a tendency to converge without meeting the correct hand target. This occurred frequently near the boundary of the reachable workspace. In order to overcome this problem an integral offset to the target hand position was maintained. This offset was added to the Cartesian target supplied to the IK algorithm and was initialised to the zero vector. Since convergence is usually close to completion within about a hundred iterations, the hand placement error vector after every hundred iterations was added to the integral offset. This compelled the IK iterations more strongly towards the target position, and precipitated the correct attainment of all targets within the workspace.

It was found that the dynamic optimisation itself also had a tendency to affect the correct placement of the hand in the final IK solution. In order to penalise postures that damaged that correct placement of the hand an error term equal to the squared distance between the target hand position and the actual hand position was incorporated into the cost function, and thus indirectly into the default posture. With this enhancement the algorithm converged satisfactorily in most cases.

With these optimisations it was found that in order to satisfy the dynamically optimal default posture, *and* the hand placement requirement the joint angle limits were often violated. Joint angle violations were therefore also penalised in the cost function. Normally the joint angle limits could be applied to the current posture, interleaved with each IK iteration but it was found that this constrained the iteration sequence of postures to remain within the space of valid postures too strictly so that convergence was often unable to progress.

The final cost function thus combined: sum squared torque cost, sum squared torque change cost, hand placement error cost, and joint angle violation cost.

Lastly, it was found that the calculation of the default posture was the most significant computational burden. Furthermore, the default posture

tended to vary very slowly with respect to IK iterations. The ratio of IK iterations to default posture calculations was therefore parameterised and a large value was found to be optimal.

4.4 Motion display tool

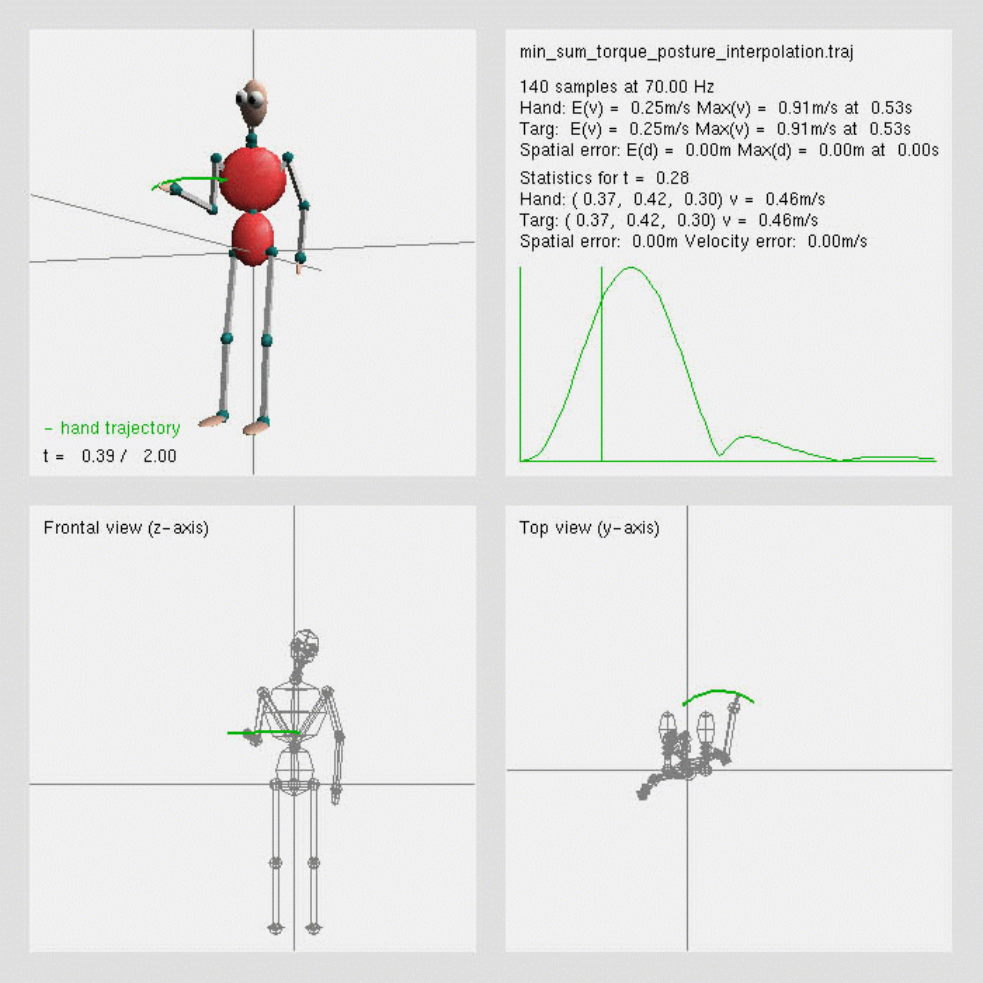

Figure 4.3: Interactive display tool: `mdtool`

The motion display tool, `mdtool`, was used to view animated motions interactively and visualise differential properties such as hand velocity vec-

tors *etc.* It was most useful for quickly verifying the correctness of motions visually, examining velocity profiles, and comparing target trajectories with actual trajectories.

Invoking `mdtool` with no arguments (or badly formatted arguments) yields the following output:

```
mdtool: Motion Display Tool -- Josh Hale 2001
mdtool <motion log file> [mdtool options] [gl options]
  -r        Reverse input byte order (toggle)
  -f<int>   Set framerate
```

The `mdtool` application is intended for interactive use and thus has only two parameters that are specified upon execution: the byte order for stored floating point data and the target frame rate for the animated display. The frame-rate is automatically limited to facilitate reasonable interaction, and window updates are minimised but nevertheless it is useful to be able to specify it in order to regulate processor usage during multi-tasking.

Interaction during runtime is accomplished with the keyboard and mouse. Pressing `h`, or any key without a defined function yields the following guide in the command shell:

```
Keyboard controls:-
a    --  Toggle display of animated figure
t    --  Toggle display of trajectory and target trajectory
s    --  Toggle zoomed out view
w    --  Cycle line width for trajectory display
y    --  Toggle display of velocity vectors
u    --  Toggle display of trajectory and target trajectory cursors
r    --  Reset parameters to default values
m    --  Cycle rendering modes
i    --  Cycle colour sets
f    --  Display framerate and system clock
p    --  Print the statistical data to the command prompt
o    --  Examine statistical properties at current time index
0    --  Hide statistical properties
1-5  --  Select a default set of display flags
```

```
<spc>--  Refresh window
q    --  Quit program

|<       <<     <slow    <||    ||>    slow>   >>     >|
 z        X        x      C      c       v      V      b

Additionally, ',' and '.' may be used to step through sample frames
Spc  --  Refresh display
```

The application window shown in Figure 4.3 was written using OpenGL. The rendering code for the humanoid figure itself was borrowed from the robot simulator and consists of a few hundred lines instantiating OpenGL shape primitives according to specified joint angles.

Of the four sub-windows, the top left is an interactive display window, the bottom two are fixed orthogonally aligned views, and the top right is the statistics window used to display graphs and textual information.

Dragging the mouse in the display window rotates the viewing angle. The adjacent keys from `z` to `b` act like the cuing controls of a video editing system facilitating, in either direction, jumping to the end of the motion, double speed playback, normal speed playback, $\frac{1}{10}^{th}$ speed playback, frame by frame advance and pausing.

The statistics window displays the following information in descending order. The name of the `traj` file, the number of samples and sampling rate are printed first. Integrated statistics including the average hand velocity, the maximum hand velocity and its instant of occurrence are then displayed. Similar statistics for the target hand trajectory and displacement between actual and target hand trajectories are also displayed. The remaining textual information is instantaneous and so the time instant is printed, along with the hand position, target hand position and instantaneous spatial error and velocity error.

The algorithms for generating the statistical data are interleaved with rendering code so that the application does not freeze during initial calculation of statistics but the results are not available until calculations are complete.

The percentage of samples integrated is instead displayed. Also, the integral statistics in the example are based on the position and velocity of the actual and target hand trajectories. These statistics are selected in the source code itself, but provision was made to easily interchange alternative statistics at compile time such as sum squared torque *etc.*

Beneath the textual display a graph utility was included. In the example it displays the hand velocity on a normalised scale. The instant under examination in the three humanoid windows is indicated with a vertical line on the graph -the time cursor. In the example it may be seen that the hand is accelerating from the initial posture, but has not yet reached peak velocity. The velocity graph also reveals a secondary velocity peak following the main point-point motion.

Key controls in any window facilitate the following remaining features. The humanoid figure may be hidden to speed up rendering and increase visibility of the other components. Likewise, the hand trajectory and target hand trajectory may be hidden or displayed. Velocity vectors for the hand trajectory and target trajectory may be hidden or displayed. The view may be zoomed, the colours may be cycled through a set of three palettes with predominantly white, black or light green colours intended for printed display, projected display and monitor display respectively. Since rendering was accomplished using OpenGL, it was straightforward to switch the rendering accuracy from wire frame to smooth shading to optimise rendering time or display quality respectively. The width of lines used for trajectory and vector display may be controlled to suit the display resolution. A set of preset states may be used to quickly select rendering options for investigating the trajectory, the visual appearance of motion, *etc.* Figure 4.4 shows the humanoid figure, and an example hand trajectory in yellow along with the target trajectory in black. The wire frame and shaded rendering modes are also compared.

Finally, the source code may be found on the CD accompanying this thesis. This includes the files `display_tools.c`, `display_tools.h` and `mdtool.c`

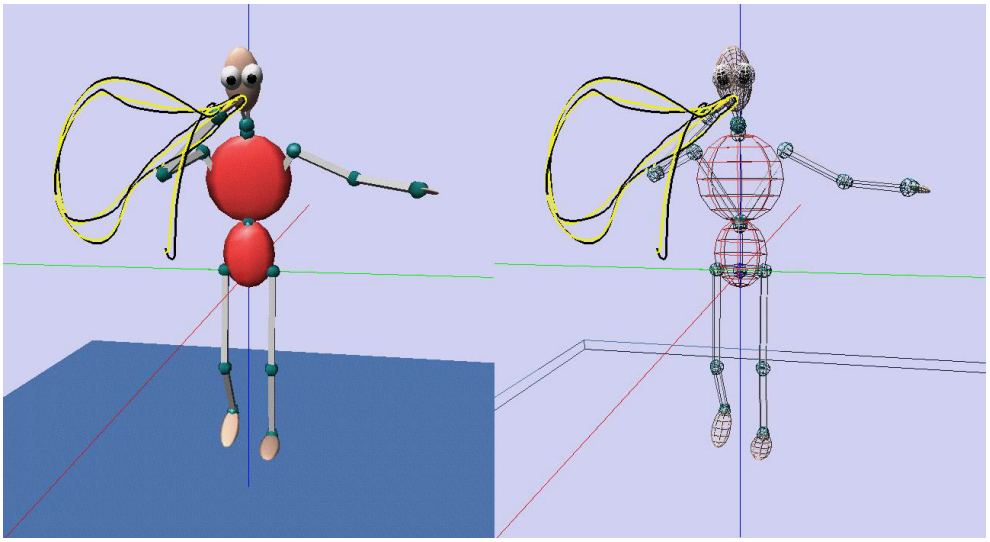

Figure 4.4: `mdtool` rendering modes

which total two and a half thousand lines. The motion logging library is also used (`log_tools.c` and `log_tools.h`).

4.5 Motion capture

Samples of human motion were necessary to validate the effectiveness of the motion generation algorithms and facilitate psychovisual testing. A series of full body movements were recorded, and converted into the same `traj` format as the artificially generated motions. The motion capture hardware, software processing, and motion post-processing are discussed in this section.

4.5.1 Motion capture system

The Optotrak system was used to capture full body human motions. Optotrak is an optical capture system. As such it has very accurate position resolution, but suffers from marker visibility and occlusion problems.

A single Optotrak camera array consists of three high speed cameras that monitor a set of infrared light emitting diodes (LEDs). The LEDs are

strobed at different frequencies so the Optotrak control unit is capable of identifying each LED individually. Each camera has an approximately 50° view from about $1-2$ meters to about $6-7$ meters in front of the lenses. The Cartesian position of a marker can be resolved from a single camera array. By using three camera arrays however, a greater accuracy and more importantly, greater marker visibility was achieved. Since it is important to place markers on every limb of the body it is particularly difficult to obtain good sensor visibility with a single camera array.

Figure 4.5 shows the motion capture environment and marker placement used.

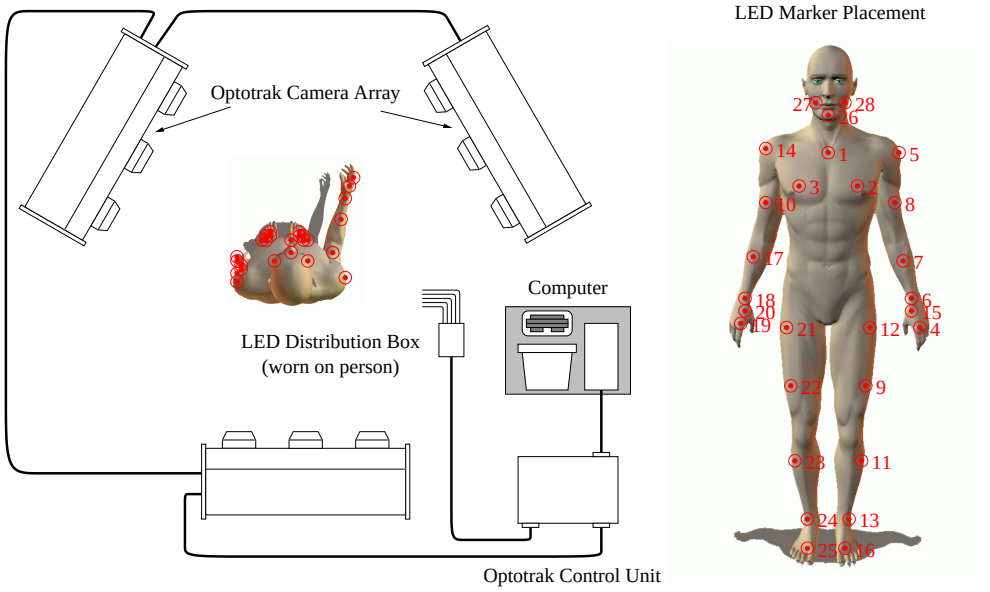

Figure 4.5: Motion capture environment & marker placement

4.5.2 Software processing

In order to capture human motion and store it in the `traj` file format several phases of software processing were necessary. Each phase following the first corresponds to an off-line transformation accomplished with a different piece of software.

1. **Raw motion capture** was recorded using three linked Optotrak camera arrays. Data was recorded at 70 Hertz with 28 markers distributed over the body.

2. **Cartesian marker positions** were derived using Optotrak software. The raw data stored in camera coordinates was converted into Cartesian positions according to a camera calibration file.

   The camera calibration file may be generated by motion capturing a special rigid body with 25 markers fixed onto it. Since the marker positions are known to the software, it can derive the camera transformations for each array.

3. **ASCII marker data** was produced using a short Matlab routine to parse the Optotrak data files and output easily interpreted sequences of ASCII floating point values. Since some markers were occluded, or could not be resolved accurately, their values were flagged by setting them to $-10,000$.

4. **Complete ASCII marker data** that contained no missing marker data was generated by linearly interpolating between the frames in which a marker vanished and reappeared. In the case that a marker was missing at the start or end of a sequence, the first or last value was duplicated constantly.

   This phase was accomplished with the `correctData.c` utility, which may be found on the CD accompanying this thesis.

5. **Joint angle data** was generated using a special processing technique developed by Marcia Riley, a researcher at ATR [Riley et al. (2000)].

   This technique required the kinematic specification of the humanoid (which was assumed to be the robot DB) and a sample of motion data with a known posture. A motion sequence was recorded in a default position looking forwards with the arms hanging by the side of the

body. The marker positions on the body were derived automatically from this data.

For a given set of marker data and a candidate set of joint angles, the approximation error may be determined by calculating the marker positions according to the candidate joint angles and then summing the squared distances between each *calculated* marker position and the corresponding *actual* marker position.

For each frame of motion the joint angles were determined by iteratively minimising this error term using Gaussian optimisation. The coherence between subsequent frames was exploited by initialising the candidate joint angles to the solution obtained in the previous frame.

6. **Traj files** were then generated by converting the ASCII format joint angle data output by the optimisation technique, and storing them in the efficient binary `traj` format. This was accomplished with the utility `convertData.c`, also stored on the CD accompanying this thesis.

7. **Filtering** was performed on some motion samples where bad marker data or poor convergence of the Gaussian optimisation lead to consistent but noisy data. This post-processing is discussed further in the next section.

8. **Concise motions** were cropped from the 15-35 second samples recorded initially. There are many phases of data transformation, and it is possible that occlusion, noisy data or non-convergence of the Gaussian optimisation may render portions of motion unusable. It was therefore necessary to perform each motion about five times and crop the samples at this stage.

4.5.3 Post-processing

A number of degrading factors affected the human motion capture data. This section summarises the sources of error and discusses the legitimacy of

post-processing techniques.

Markers may become occluded from all three camera arrays in certain postures, or may move outside the cameras' view space. The captured marker data in fact presented few occasions when markers where missing. It was very rare for markers on the right arm or trunk of the body to be missing (the most significant components of motion). Rather, the left hand and left foot markers often drifted in and out of the viewable region. Missing markers were typically absent for less than 500 milliseconds. It was therefore considered reasonable to linearly interpolate the most temporally adjacent marker data over absent periods. An individual marker has little affect on the optimisation algorithm used to determine joint angles, and since this algorithm also exploits the coherence between subsequent postures it tends to prevent sudden unrealistic changes in joint angles. Furthermore, the leg joint angles hardly varied at all throughout the motions, which were performed in a static standing posture. Nevertheless, motions with more than one or two concurrently missing markers, or markers missing for more than 500 milliseconds were not used.

The sensor resolution of the Optotrak system is very high, and it is in general capable of resolving marker positions to within a single millimetre. However, attaching markers to the human body is awkward and was accomplished using double sided adhesive pads with micropore tape to secure the wires leading to each LED marker. Each marker is therefore subject to motion relative to the body. There is little motion relative to the local area of skin, but bending of the joints, and stretching of the skin and muscles means that markers drift relative to the kinematic topology of the skeleton.

The joint angle data output by the Gaussian optimisation algorithm was affected by this noise. In general, irrespective of noise, the kinematic solutions were consistent. Any uncertainty in markers' local positions was overcome by the consistency of their placement relative to each other. The kinematic data output corresponded with the original motions in every case.

However, the localised uncertainty precipitated very low amplitude oscil-

lations in the joint angles. The oscillations were typically only 1 or 2 degrees, but occurred randomly over subsequent frames, *i.e.*, at 70 Hertz. These oscillations were not visible in most cases, but the wrist for example, could be seen to tremor slightly throughout some motions.

In order to generate consistent motions to display in psychovisual tests it was necessary to remove this high frequency component of motion. The joint angles demonstrating oscillations were therefore smoothed by averaging over seven frames. The results in Section 5.4 show that the high frequency oscillations were present only in a few of the captured motions, and were found to occur in a frequency band normally containing no energy. It was therefore concluded that the oscillations were an artefact of data conversion rather than a trait of the original motion.

Chapter 5

Computational, theoretical & psycho-visual testing

Experimental results

This chapter presents empirical results associated with the biomimetic motion production models formalised in Chapter 3 and expounded algorithmically in Chapter 4.

Psychovisual testing was performed to assess the naturalness and similarity of motion models. The testing process is described, from experimental design to results, in Section 5.1.

Results include the perceived naturalness and similarity of the motion production models, a feature based analysis of the resultant motions and models, the computational characteristics of the motion generation algorithms, and a frequency based analysis of the motion capture data.

Following this perceptual evaluation, the synthetically generated motions are analysed in Section 5.2 according to features commonly found in human motion. The presence or absence of these features is made clear using velocity profile and hand trajectory graphs and the results are summarised. Some motion models exhibiting particular unique characteristics are then discussed.

Various computational statistics including motion generation time, iteration counts, and costwise improvement were gathered. These statistics are analysed and presented in Section 5.3.

Finally, the motion capture samples are analysed in Section 5.4. The energy present in joint angle trajectories at different frequency bands is displayed. 'Clean' and noisy joint angles are compared. The results explain and justify the use of smoothing to correct artefacts appearing in some motion capture samples. The artefacts were introduced during the conversion of raw motion capture to posture data.

This chapter is thus primarily concerned with the collection and presentation of empirical data. While some explanations are offered along with the results, the analysis and significant conclusions are presented in the following chapter.

5.1 Psychovisual testing

This section presents the experimental design and results of psychovisual testing. These experiments involved displaying motion clips to subjects and recording numerical responses.

The aims of the psychovisual testing are presented fully. Corresponding design aspects including the choice of motions to display are presented including the actual motion specifications. Practicalities of testing are described and the user interface is displayed.

The results reveal the perceived naturalness of each model, and the perceived similarity of the models. Graphs comparing the perceived naturalness of computer animated and robotically performed motions are displayed. The perceived similarities of the models are displayed as a two dimensional graph (of reduced dimensionality) for the computer animated and robotically performed motions.

The results are discussed further, and conclusions are drawn in Section 6.1, in the following chapter.

5.1.1 Experimental design

The primary aims of psychovisual testing were to evaluate the human-like naturalness of each motion model and the comparative similarities of the models. This was accomplished by displaying motion clips to experimental subjects and collating their responses.

Motion clips generated by synthetic motion models and genuine human motion capture data were included. In order to assess the subjects' responses to virtual humans *and* a robotic humanoid, the motions were displayed with both media, in separate experiments. Including the motion capture, fifteen motion models were represented.

Motion specifications

In order to obtain meaningful results it was desirable to obtain different responses for each motion model. The motion specifications selected for the experiments were therefore designed to emphasise the characteristics of each motion model.

It was not possible to offer subjects a fee for participation. The total experimental time was therefore a factor in obtaining voluntary subjects. Since each specification yielded one clip for each of the fifteen models, the time for each clip, and the number of specifications was thus limited. Since the number of motion specifications that could be incorporated was limited, it was necessary to design the specifications efficiently.

A theoretical comparison of the motion models was made by recourse to the features of human motion presented in Section 2.5. The following features were found to divide the synthetic models: hooks, curvature, velocity profile, and joint configurations.

The corrective hooks found in fast movements were expected in models incorporating forward dynamic simulation (MJVT, EPH, MTPVT). Curved hand paths were expected with intrinsically planned kinematic models (MAV, MAA, MAJ, MAS) and, it was thought probable that they would also appear in the dynamically optimal models (MTC, MT, MTP, MTPVT).

Discontinuous acceleration profiles were expected for first order derivative models (MV, MAV), and discontinuous jerk profiles were expected for second order derivative models (MA, MAA). Bell shaped velocity profiles were expected for models incorporating higher order derivatives (MJ, MAJ, MS, MAS) and for dynamically optimal models (MT, MTC, MTP). Bell shaped profiles with skewing for fast motions and via-point motions were expected for the the models incorporating forward dynamics (MJVT, MTPVT).

The types of kinematic configuration could be grouped as follows: fully optimised postures[1] at every instant for extrinsic kinematic models (MV, MA, MJ, MS), interpolations between fully optimised postures (MAV, MAA, MAJ, MAS), approximations to such interpolations affected by forward dynamics (EPH, MJVT), and unpredictable configurations resulting from various optimisation functions (MT, MTC, MTP, MTPVT).

These features represent several axes of of variation between the models. In designing a set of motion specifications it is therefore desirable to incorporate specifications that emphasise and de-emphasise the presence of each feature so that a broad range of comparisons are represented. Hooks are emphasised by fast motions with a fixed end point, and minimised by slow motions or motions without a terminal target. The effect of path curvature is emphasised for otherwise straight point-point hand paths, and de-emphasised for inherently curved multi-target trajectories. The velocity profile is likely to be emphasised by slow straight movements, and de-emphasised by fast or complex movements. At the least, these cases represent conditions with significantly different velocity profiles. The differences between joint configurations are emphasised by trajectories that pass through awkward to reach regions of the workspace that are solved in clearly alternative ways kinematically.

Besides the major considerations above, a few remaining factors affected the choice of motion specifications. It was desirable to incorporate motions

[1]The improved extended Jacobian IK algorithm was iterated to stable convergence for each point on the hand trajectory.

perpendicular to the viewing axis as well as motions roughly parallel to the viewing axis. Motions perpendicular to the viewing axis ensure full visibility of the arm in motion while conversely, motions parallel to the viewing axis challenge the perception of the arm and hand. Incorporating both cases reduces the likelihood that perceptual results only pertain to specific cases of motion perception. Finally, from the perspective of writing a doctoral thesis it was also desirable to demonstrate the features of the motion production environment in full, *i.e.* using point-point as well as cyclic splines *etc.*

No.	**Type**	**Duration (s)**	**Knots**
1	Point-point	4.5 (4.23)	$\begin{pmatrix} 0.28 & 0.4 & 0.55 \\ 0.25 & 0.36 & 0.09 \end{pmatrix}$
2	Point-point	1.5 (0.35)	$\begin{pmatrix} 0.52 & 0.3 & 0.34 \\ 0.08 & 0.33 & 0.34 \end{pmatrix}$
3	Point-point	1.5 (0.3)	$\begin{pmatrix} 0.23 & 0.35 & 0.61 \\ 0.21 & 0.35 & 0.22 \end{pmatrix}$
4	Via-point	5 (4.39)	$\begin{pmatrix} 0.37 & 0.38 & 0.29 \\ 0.12 & 0.42 & 0.42 \\ -0.05 & 0.36 & 0.18 \end{pmatrix}$
5	Via-point	1.5 (0.55)	$\begin{pmatrix} 0.37 & 0.37 & 0.29 \\ 0.34 & 0.39 & 0.54 \\ 0.01 & 0.34 & 0.40 \end{pmatrix}$
6	Via-point	1.5 (0.65)	$\begin{pmatrix} 0.1 & 0.3 & 0.29 \\ 0.1 & 0.38 & 0.53 \\ 0.04 & 0.52 & 0.31 \end{pmatrix}$
7	Cyclic	9.6 (4.8)	$\begin{pmatrix} 0.16 & 0.27 & 0.26 \\ 0.185 & 0.47 & 0.53 \\ 0.2 & 0.62 & 0.26 \\ 0.18 & 0.47 & 0.08 \end{pmatrix}$

Table 5.1: Motion specifications for psychovisual testing

The final choice consisted of the seven specifications summarised in Table 5.1. The duration column shows the total duration of the video clips/animations and in brackets the duration specified during generation. Motions 1-6 were padded with a period of stillness at each end of the movement so that the onset and termination of motion could be determined visually. Conversely, motion 7 was repeated twice to emphasise its cyclic nature. Figure 5.1 shows the postures corresponding to each knot and illustrates the direction of motion in each case.

Motions 1,4 and 7 are slow, the others are fast. Motions 1-3 have straight (point-point) paths. Motions 4-7 have curved (via-point or cyclic) paths. Motion 6 is predominantly along the viewing axis towards the eye. While motions 1-6 have a fixed endpoint motion 7 is cyclic, and loops twice in the 9.6 second duration. Motions 1 and 3 are downwards, and thus intuitively assisted by gravity.

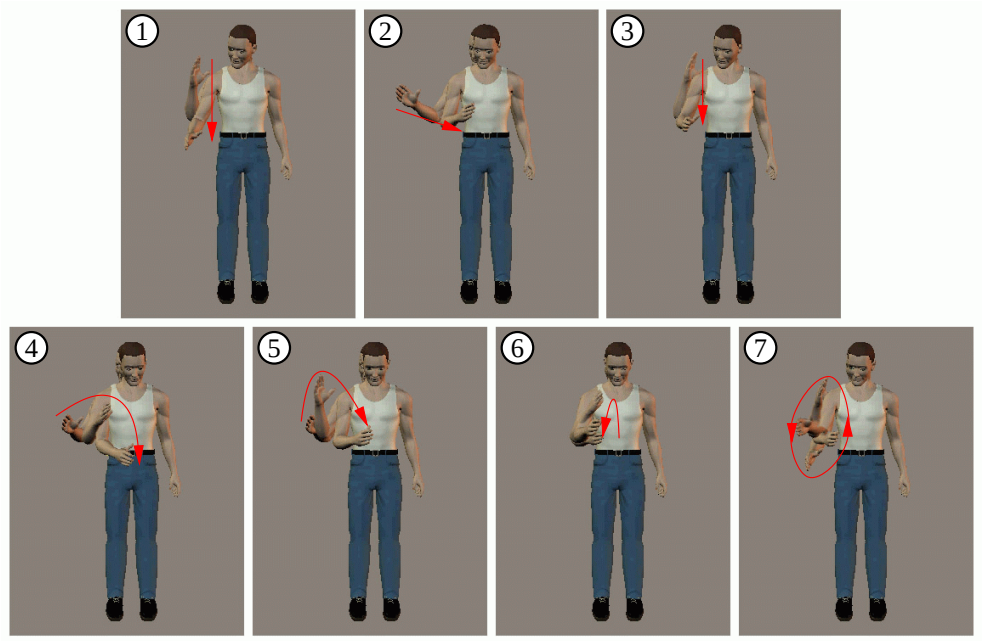

Figure 5.1: Motions used for psychovisual testing

Conventions

A number of design decisions and some arbitrary but consistent conventions were adopted as follows.

The head was compelled to look at the hand throughout the motion. This was decided because without explicit control the head was either totally still for the whole motion or still relative to the torso. In both cases an unnatural impression was created by zombie like movement. By choosing the hand as a gaze point the impression that the humanoid was concentrating on the movement was instead conveyed.

The appearance of the virtual human was chosen to reveal the motion of the body as well as possible. Initially the human appeared naked to facilitate maximal visibility of motion. However, this also seemed somewhat unusual and a skin tight vest was added along with a pair of jeans and shoes. The clothed human appears less distracting. The appearance of the humanoid robot could not be easily modified and was recorded as is, using digital video.

The durations of all the corresponding motions (generated with the same specification but a different model) were cropped to the unbracketed durations in Table 5.1. For kinematically generated motions, all the movement occurs within the bracketed duration and the remaining frames are still. Dynamic motions were fully simulated and overshoots or oscillation were possible in the remaining time. This window around the significant component of motion was necessary to display such instabilities and, as mentioned above, to allow the subject to perceive the onset and termination of motion as distinct from the start and end of a clip. Some motions thus came to rest at exactly the specified bracketed durations, others were subject to placement inaccuracies and required a short extra period to stabilise and some did not even stabilise completely (the EPH model was found to oscillate indefinitely in some cases). It was thought that the duration might affect subjects' perception or corresponding motions, and in order to remove this factor the cropping was applied uniformly, *i.e.* the EPH model was not granted extra time to achieve a stable posture.

Only the cyclic motion (motion 7) began and ended in a moving state. The hand cycle was performed twice to emphasise the continuous nature of the motion and de-emphasise the position at the start of the clip.

Method

Four experiments were performed. These involved two experimental designs performed twice each using the robot or computer graphic humanoid. The experiments were designed to assess the naturalness of each motion model, and the similarity of the models. A total of 105 motion clips were assembled, representing each of the 7 motion specifications with each of the 14 synthetic motion production models or the genuine human motion capture.

In the naturalness experiment ten subjects were shown each of the 105 motions individually. Subjects were permitted to view each clip as many times as they liked before responding. Subjects were instructed to respond with an integer value from 1 to 10 reflecting the human-like naturalness of the motion. A larger value indicated a greater naturalness. The motions were shown in a randomised order, generated afresh for each subject. The number of times each subject viewed each clip, the response times and the numerical responses were recorded. In fact only the numerical responses were analysed. The total time required to show all the clips without a break is 6 minutes 28 seconds. Subjects viewed some clips multiple times, and required time to consider and respond to each of the 105 clips so that the experiment required 15-25 minutes. The user interface may be seen in Figure 5.2.

In the similarity experiment ten subjects were shown pairs of motions (not simultaneously). Motions generated using the same specification, but different models were shown as pairs. Every possible pair of models was accounted for, but due to the large number of possible combinations only two motion specifications were included (4 & 5). It is a coincidence that the number of possible pairs, $\frac{15\times14}{2} = 105$, is equal to the total number of clips recorded. Subjects were allowed to watch each clip as many times as they liked and were instructed to respond with an integer between 1-10 reflecting

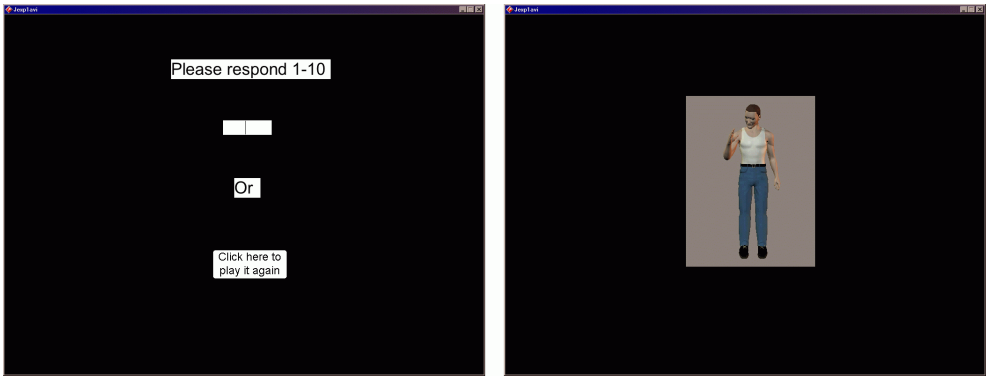

Figure 5.2: User interface of the psychovisual testing utility

the similarity of the two clips. A larger value indicated a greater similarity. The total time required to view all the clips is 22 minutes 45 seconds. The total experiment time was around 35-40 minutes.

The experiments were performed on a PC using `AVI` files. The keyboard was used to enter a numerical response, or the mouse was used to press a button to view an animation again. Some subjects were tested in Japan and some in Glasgow. The total experiment time was about 1 hour and 15 minutes.

Practical issues

Animation clips were prepared in digital format. Following generation with `mctool` the `traj` motion files were used to create `BVH` motion files, and to control the humanoid robot.

`BVH` files were imported into Poser and `AVI` files were produced. It was necessary to keep the clip data within a reasonable storage bound so a frame size of 276×364 was chosen, with a frame rate of $30Hz$. The total storage required by all the `AVI` clips was $253MB$.

The robot motion clips were created by recording robot performances using digital video. The robot experiment room in ATR is well equipped with continuously variable lighting, so that arranging the robot for filming

was straightforward. Having recorded the digital video each motion clip was then cropped to the exact length given in Table 5.1 and the onset of motion was synchronised uniformly.

In order to perform the experiments a tool was written using Director. This software facilitates the production of stand-alone applications featuring multimedia types and human-computer interactions. Dr Harold Hill, working at ATR kindly wrote scripts for Director. These scripts fulfilled the experimental specifications, displaying animations and recording subject responses in log files *etc.* and was compiled for a Windows PC environment.

All the experimental clips, Director software and Director scripts are on the CD accompanying this thesis.

5.1.2 Results

Ten subjects were tested in each of the experiments. Table 5.2 shows the perceived naturalness for each of the fourteen synthetic models, displayed using a computer generated humanoid. Table 5.3 shows the perceived naturalness values when the motions were displayed using a humanoid robot. All ten subject's responses were combined by averaging the responses to each trial, categorised by motion production model. The variance across subjects is quite high, as may be seen in the tables but an analysis of variance (ANOVA) rejects the null hypothesis that the means are stastically equivalent with confidence values over 99.5% for both tables. The ANOVA statistics are $F(14, 1035) = 7.16, p = 2.2 \times 10^{-14}$ and $F(14, 1035) = 2.27, p = 0.0047$ for the CG and robot results respectively. A more detailed statistical analysis and interpretation is presented in Section 6.1.

The perceived similarity experiments were performed with the same experimental paradigm. The subject's responses were normalised to have equal mean values before combining the results by averaging. This removed the absolute differences between subjects' rankings of all 105 motions taken as a group, but facilitated a relative comparison of different models' similarities. The resultant similarity values were additively inverted to convert the

similarity values (higher numbers correspond to more similar pairs) into dissimilarity values (higher values correspond to more dissimilar pairs).

By interpreting the dissimilarity values as distances between points representing the motion models the model similarities could be visualised diagrammatically. In general $n-1$ dimensions are required to assemble n points at arbitrary distances from each other. The two dimensional visualisations in Figures 5.3 & 5.4 are thus *dimensional reductions*. Since it is not possible to place the points so that the distances correspond exactly the sum squared placement errors are minimised. The diagrams were generated using gradient descent with randomised starting configurations and no relations between the points were initially specified besides the distances derived from the experimental results. The radius of the circles surrounding each point correspond to the average placement error between the point and each other point in the diagram. The human motion has no circle because it's distance from every other motion is exact. This was achieved by weighting the distance error for the human motion significantly higher than for the other motions.

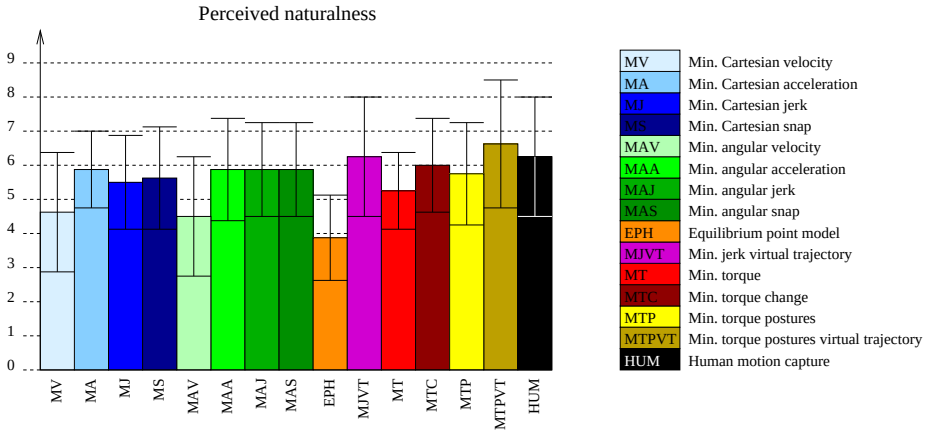

Table 5.2: Perceived naturalness of rendered motions

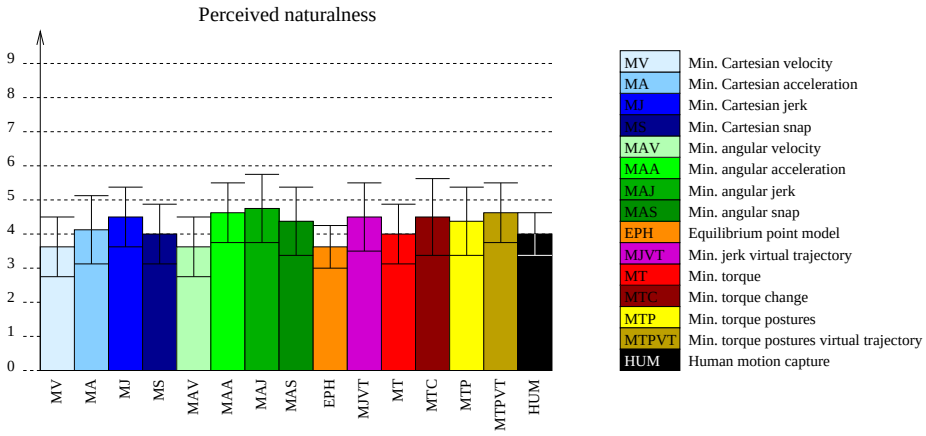

Table 5.3: Perceived naturalness of robot motions

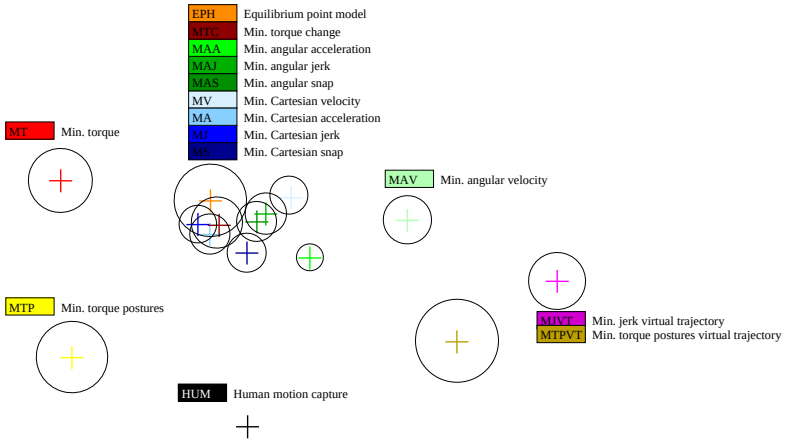

Figure 5.3: Relative similarities of models based on rendered motions

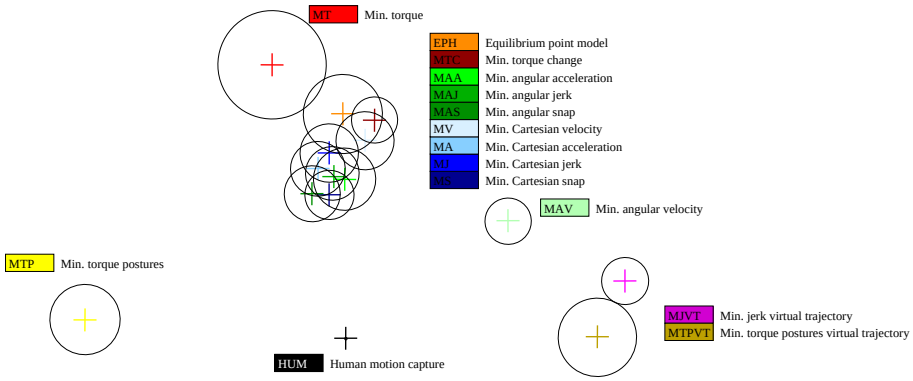

Figure 5.4: Relative similarities of models based on robot motions

5.2 Motion properties

This section presents empirical results regarding the synthetically generated motions, and motion capture.

Synthetic motions are analysed with respect to features present in human motion as discussed in Section 2.5. These features include the nature of the hand trajectory -its curvature, velocity profile and the presence or absence of corrective hooks. Hand trajectories and velocity profiles are presented and the nature of these features is summarised for each synthetic model.

Two models demonstrating unique characteristics are also discussed: the equilibrium point λ model (EPH), and the minimum torque model (MT).

Conclusions based on the observed features are drawn in Section 6.1.

5.2.1 Velocity profiles

Point-point reaching movements produced by humans tend to exhibit bell shaped hand velocity profiles as was discussed in Section 2.5.2 & 2.5.3. When the movements are very fast the profiles can be skewed, and for highly curved motions there can be multiple peaks in the velocity profile. The synthetic motion models exhibit a variety of velocity profiles.

The Cartesian interpolations yield velocity profiles that are linear, quadratic and bell shaped for MV, MA or MJ & MS respectively. This may be seen in Figure 5.5 which shows the Cartesian interpolations' velocity profiles for motion 1, a 4.5 second point-point movement. The corresponding human motion velocity profile appears in Figure 5.8, and displays a more approximate bell shape. The same shape may also be seen in Figure 5.9 which shows the velocity profile for motion 2, a fast point-point movement. Figure 5.10 shows the profiles for motion 5 (a fast via point movement). In this case the profile is double peaked for all but the MV motion, which has a piecewise linear form.

The velocity profiles for the angular interpolations have very similar characteristics, although the correspondence between joint angles and hand po-

sition is non-linear. The result is that the velocity profiles are skewed as may be seen most clearly in Figure 5.6 where the minimum velocity angular interpolation (MAV) has a linear, but non-constant profile.

The minimum torque change (MTC) model yields approximately bell shaped profiles as can be seen in Figures 5.7, 5.9 & 5.10. Minimum torque (MT) motions however, can have peculiar trajectories when the duration of motion is long. This phenomenon is discussed further below, and can be seen in Figure 5.7. When the duration is short however, this does not occur and the profile is bell shaped with the usual single or multiple peaked form (Figures 5.9 & 5.10).

The equilibrium point model (EPH) yields unstable motion because no velocity damping term was included in the exponential springs generating joint torques. The result is an unstable profile that can be seen in Figure 5.7.

Finally the velocity profiles for the optimal postures models (MTP & MTPVT), and the minimum jerk virtual trajectory model (MJVT) are shown in Figure 5.8 along side the human motion itself, and also in Figures 5.9 & 5.10. These profiles are all bell shaped and slightly skewed in the case of the two virtual trajectory models (MTPVT & MJVT).

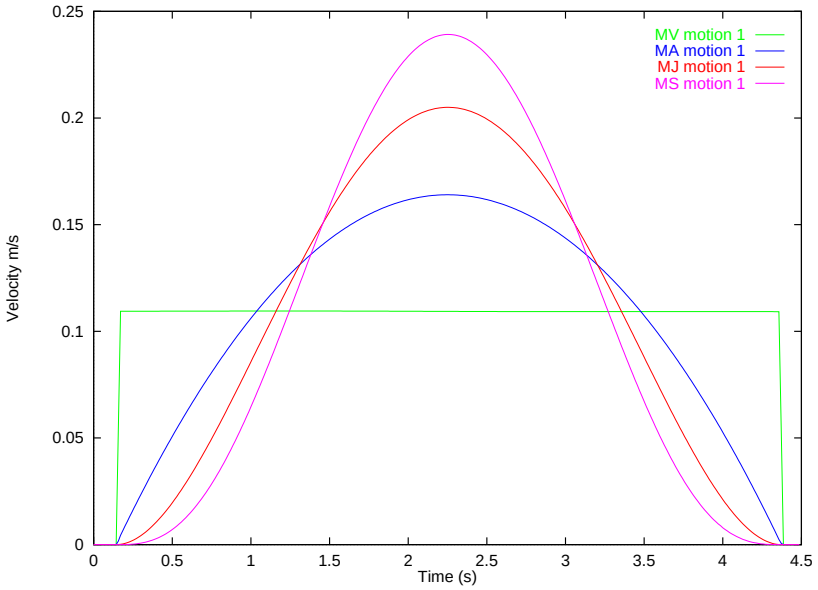

Figure 5.5: Velocity profiles for Cartesian interpolations, motion 1

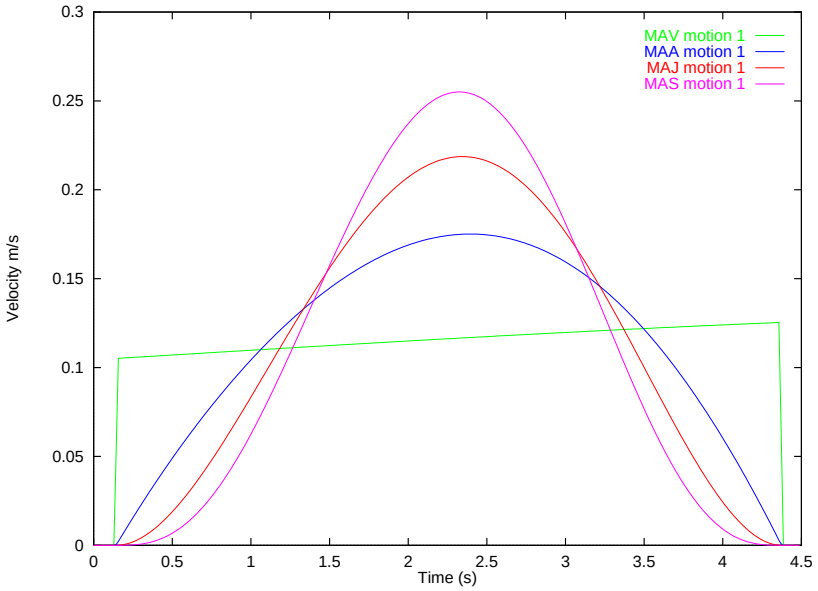

Figure 5.6: Velocity profiles for angular interpolations, motion 1

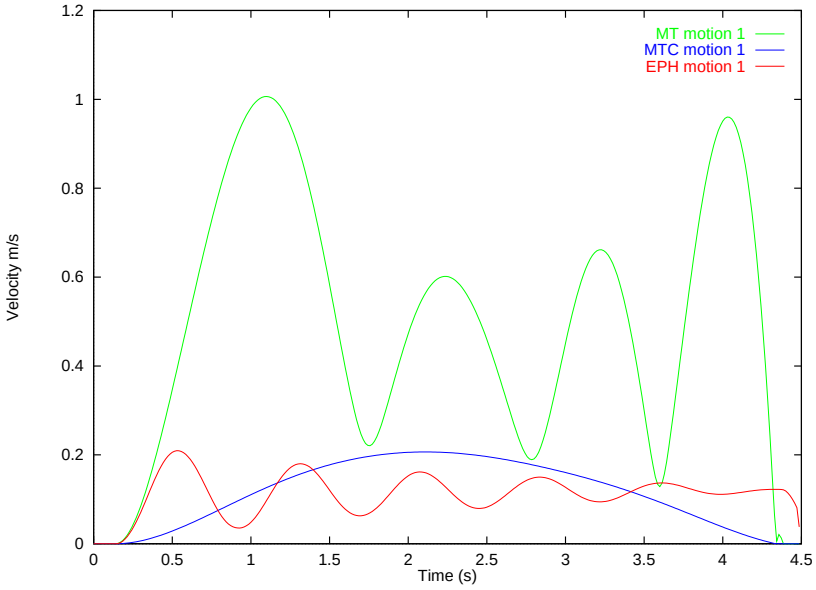

Figure 5.7: Velocity profiles for MT, MTC and EPH, motion 1

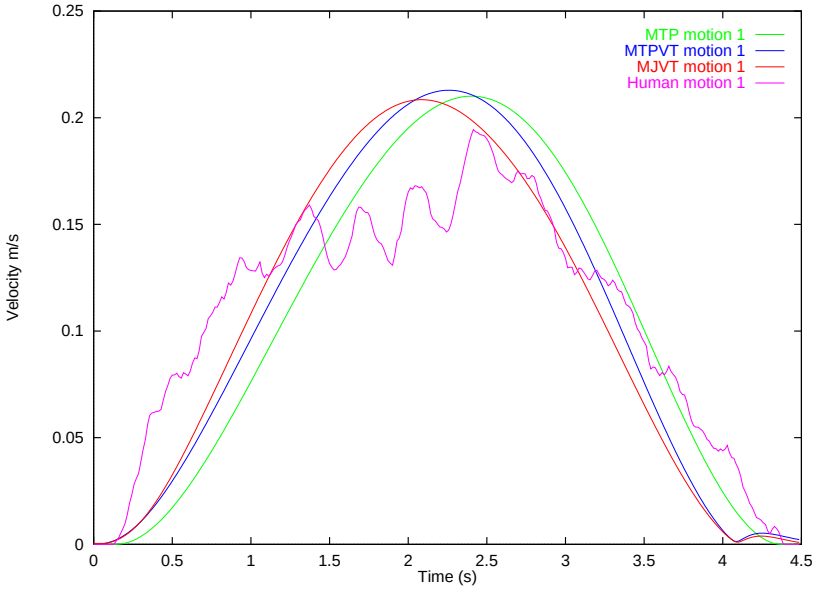

Figure 5.8: Velocity profiles for MTP, MTPVT and human movement, motion 1

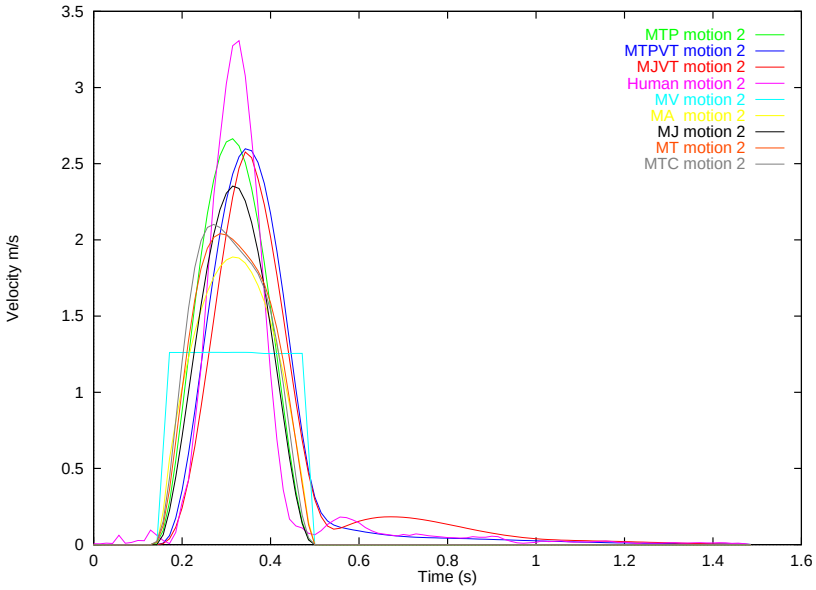

Figure 5.9: Velocity profiles for motion 2

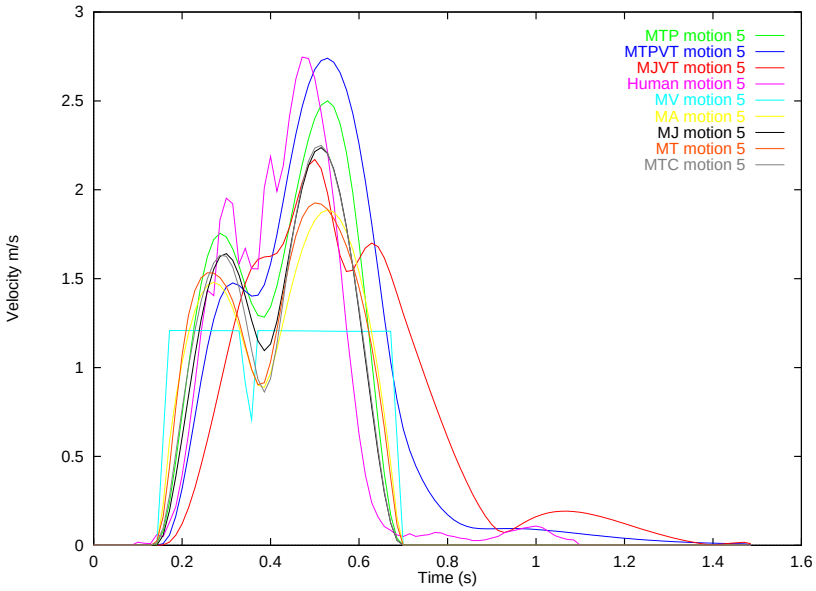

Figure 5.10: Velocity profiles for motion 5

5.2.2 Curvature

Point-point movements of the hand produced by humans tend to exhibit a slight curvature depending on the region of the reach-space, as was discussed in Section 2.5.1.

The Cartesian interpolations yield straight paths for point-point motions. This can be seen in Figure 5.11. The angular interpolations (MAV, MAA, MAJ & MAS) yield curved hand trajectories, as do the optimal posture methods (MTP & MTPVT). MTC yields a curved trajectory although the shape is more obviously skewed because the hand trajectory is determined by dynamic factors not considered by the other models. The human hand trajectory is also gently curved, although the curvature is not entirely uniform and is less pronounced than that of the synthetic models. Figure 5.12 shows the curved nature of the EPH motion. The hand trajectory for MT is peculiar because the duration of the motion is long (see Section 5.2.6).

Curved trajectories are observed for every model in Figure 5.13 which shows the hand trajectories for motion 5 which is a via-point motion that is inherently curved. Figure 5.14 shows that the usual curvature is produced by the MT model for motion 5 because the duration of motion is short. The speed of the motion causes the EPH model to become unstable, and the trajectory is not well controlled. When the EPH model does exhibit a stable trajectory its curvature is however workspace dependent. This is because the model interpolates postures, albeit indirectly in force space.

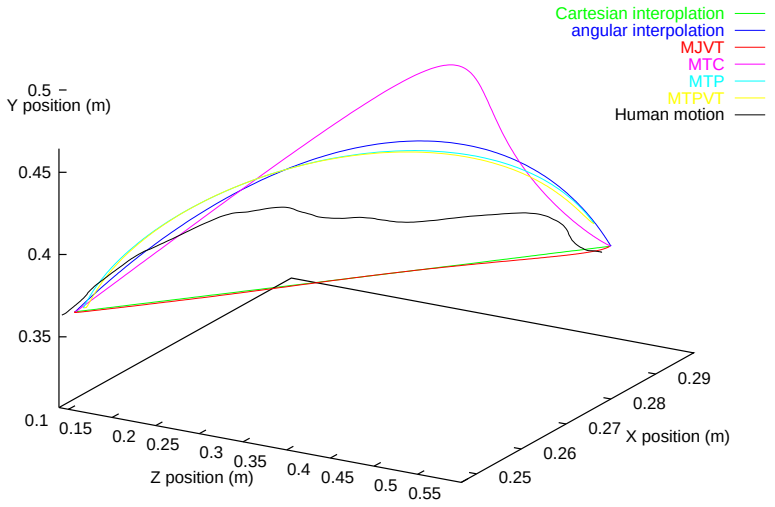

Figure 5.11: Hand trajectories for motion 1 (a)

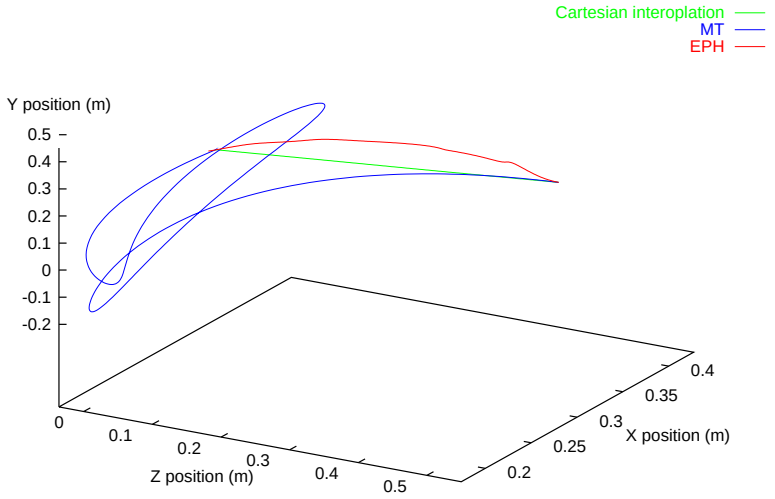

Figure 5.12: Hand trajectories for motion 1 (b)

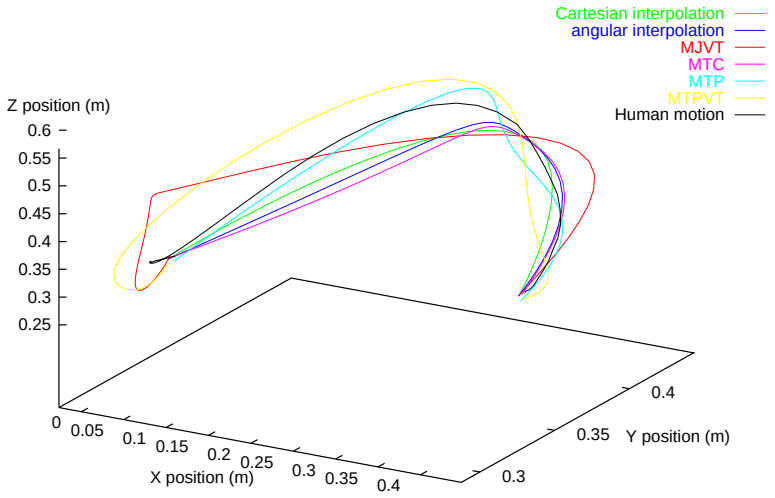

Figure 5.13: Hand trajectories for motion 5 (a)

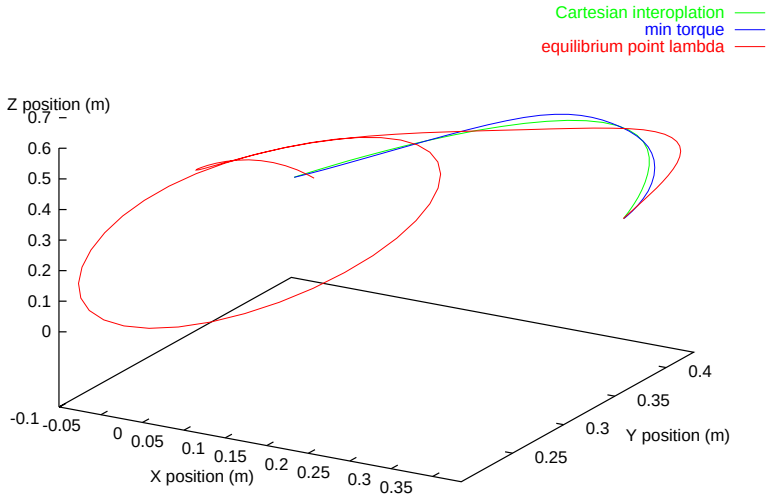

Figure 5.14: Hand trajectories for motion 5 (b)

5.2.3 Corrective hooks

Fast motions to a target point produced by humans tend to exhibit corrective hooks at the end of the hand trajectory as was discussed in Section 2.5.10.

Corrective hooks were replicated by the motion models incorporating forward dynamic simulation, *i.e.*, MJVT, MTP & MTPVT. The EPH model incorporated forward dynamics but was too unstable to demonstrate the hooks. The velocity profiles presented above reveal the corrective motions. After the main component of motion, a final secondary velocity bump may be seen. This is due to the hand correcting an overshoot past the target point.

There is only a slight correction appearing in motion 1 because the motion was slow enough to incur hardly any overshoot, as may be seen in Figure 5.8 and in Figure 5.11. For the motions 2 and 5 which are faster, the overshoot is more significant as may be seen in Figures 5.9 & 5.10. The corrective motion for the MTPVT does not appear as a velocity peak, but is apparent when the MTPVT trajectory profile is compared with the MTP velocity profile. The hand trajectories also reveal the overshoot present in the virtual trajectory motions and the human motion (Figure 5.13).

5.2.4 Summary of motion features

Table 5.4 below summarises the appearance of the motion features discussed above for each motion model.

Cartesian interpolations have polynomial curvature that is workspace independent and do not exhibit hooks. The velocity profile may be constant, parabolic or bell shaped depending on the order of the interpolation.

The angular interpolations have similar slightly skewed velocity profiles, no hooks, but exhibit workspace dependent curvature.

The virtual trajectory models (MJVT & MTPVT) have asymmetric bell shaped velocity profiles because of their spring damper systems. They are also the only models to exhibit proper hooks, which is due to the forward

dynamic simulation. The EPH model has unstable trajectories that are too unsteady to exhibit clear hooks.

Method	**Hand velocity profile**	**Curvature**	**Hooks**
MV	Constant	Piecewise linear	No
MA	Parabolic	Cubic	No
MJ	Bell shape	Quintic	No
MS	Bell shape	Heptic	No
MAV	Approx. constant	Workspace dependent	No
MAA	Approx. parabolic	Workspace dependent	No
MAJ	Approx. bell shape	Workspace dependent	No
MAS	Approx. bell shape	Workspace dependent	No
MJVT	Asymmetric bell shape	Approx. quintic	Yes
EPH	Unstable bell shape	Workspace dependent	Unstable
MT	Bell shape/unpredictable	Workspace dependent	No
MTC	Approx. bell shape	Workspace dependent	No
MTP	Approx. bell shape	Workspace dependent	No
MTPVT	Approx. asymmetric bell	Workspace dependent	Yes

Table 5.4: Summary of motion features w.r.t. motion model

5.2.5 Instability of equilibrium point λ motion

The motions generated with the EPH model were unstable. This model generated forces exponentially according to joint angles, but unlike the MJVT and MTPVT models that incorporate springs and *dampers*, EPH does not incorporate a velocity damping term. This causes each limb to experience torques accelerating the limb towards the target posture irrespective of its angular velocity. The result is that the limbs do not slow down as they approach their target angles and may oscillate back and forth, on either side of the target angle. Figure 5.7 shows the effect on the hand velocity for a point-point motion. The trajectories for slow motions may be stable, as in Figure 5.12 but can also be drastically unstable as in Figure 5.14.

5.2.6 Unpredictability of minimum torque motion

It has been seen in Figures 5.7 & 5.12 that the MT motions can be quite peculiar. This phenomenon involves the hand trajectory behaving in an unpredictable manner yielding complex velocity profiles. The peculiar behaviour is observed in slow motions of a fixed duration. Figure 5.15 shows equally spaced frames in the minimum torque performance of motion 1, which lasts 4.5 seconds. The stable MAJ motion is shown underneath for comparison. The motion specification is point-point as can be seen from the MAJ frames.

The optimisation process revealed that rather than move in a roughly straight line from start point to end point, it is more efficient in terms of torque to release the limb, allowing it to swing back and forth, eventually coming to rest at the target point.

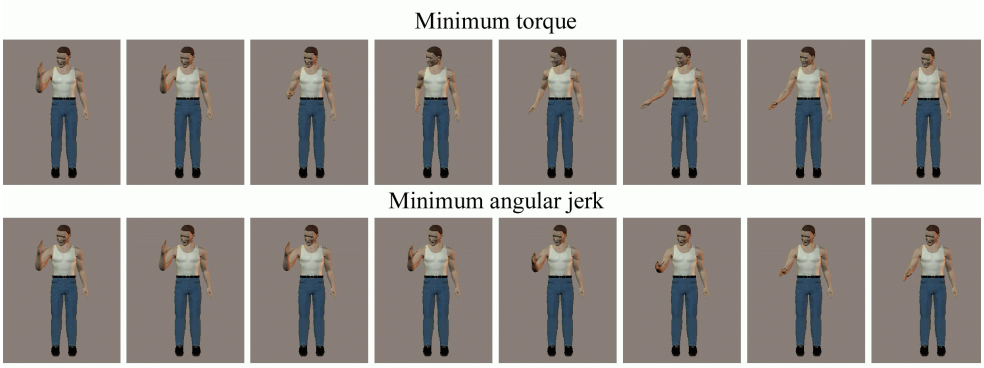

Figure 5.15: Peculiar nature of long minimum torque motions

5.3 Evaluation of motion software

This section presents empirical analyses of the significant computational processes used in motion generation.

Timing statistics comparing each of the synthetic motion models are presented. The absolute times do not constitute objective empirical results

because there is a wide range in speed between processors, and it is notoriously difficult to obtain consistent speed measurements across architectures. The absolute timing statistics do however reveal which models can be used interactively and more significantly, the relative times indicate the relative computational efficiency of each model.

The N^{th} derivative minimisation process is profiled through timing and iteration statistics. The examination is detailed and tests the process for a range of derivative minimisations (2^{nd} to 7^{th}) for cyclic and non-cyclic splines. The dynamic optimisation process applied to torque and torque change optimisation is analysed in terms of iteration and cost statistics revealing the relative reduction in optimal cost. Detailed convergence statistics for each motion are also displayed as graphs.

These results are analysed and conclusions are drawn in Section 6.1.

5.3.1 Motion generation

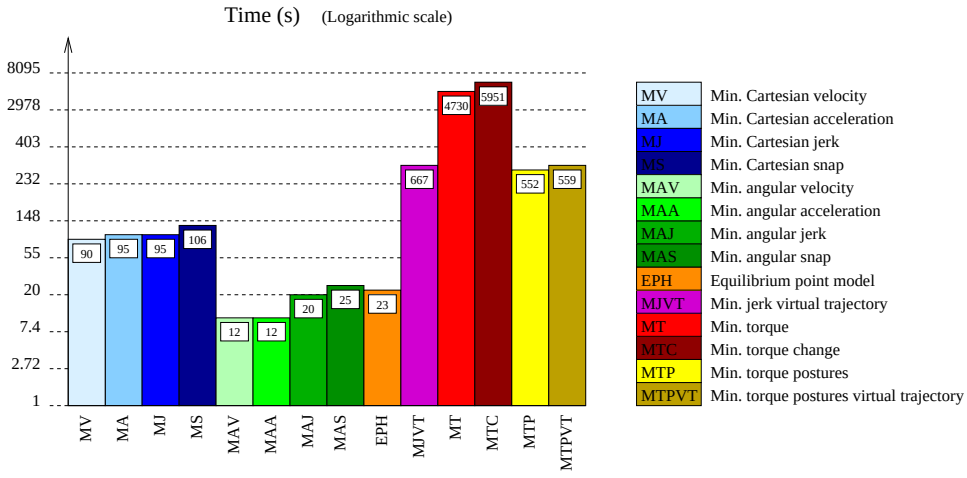

Table 5.5: Motion generation times for each synthetic model

Table 5.5 compares the time efficiency of each motion generation method. The combined time required to generate all of the seven test motions described in Section 5.1.1 is displayed. This includes four point-point motions,

three via-point motions and one cyclic motion. The time scale is logarithmic because there is a large range of resultant times.

5.3.2 N^{th} derivative minimisation

The N^{th} derivative minimisation method described in Sections 3.3 & 4.3.1 is profiled in this section. An analytical solution for arbitrary derivative minimisation was presented, but in order to optimise the knots themselves a gradient descent method was implemented. The solution space was well behaved and convergence was achieved quickly.

Table 5.6 displays the number of iterations required to solve the seven test motions with the minimum Cartesian and angular derivative models. Motions 1-3 were point-point, and thus the start and end passage times were fixed. A minimum of 10 iterations yielding no cost improvement were required before convergence was accepted and so the point-point motions (which were handled in the same way as splines with free passage times) all required 11 iterations. Likewise, the last 10 iterations of every motion represent the convergence test.

Model	**Motion**						
	1	2	3	4	5	6	7
MV	11	11	11	70	22	29	80
MA	11	11	11	56	17	19	68
MJ	11	11	11	37	31	28	94
MS	11	11	11	53	25	19	235
MAV	11	11	11	71	47	45	146
MAA	11	11	11	69	27	47	82
MAJ	11	11	11	63	53	23	181
MAS	11	11	11	53	26	23	110

Table 5.6: Iteration statistics for N^{th} derivative optimisation

Experiments testing the reliability of the algorithm are shown in Table 5.7. Normally, the initial knot passage times were set to equally divide the total duration specified but in order to test the reliability of convergence the knot passage times were initialised randomly (still in increasing order) and the convergence statistics measured over 30 trials. Table 5.7 shows the mean and variance of the iteration count and total time required to iterate the spline passage times to convergence.

Motion	**Iterations**		**Time (s)**	
	Mean	Var.	Mean	Var.
MJ 4	83.63	13.97	0.78	0.13
MJ 5	40.70	15.50	0.38	0.14
MJ 6	51.97	27.02	0.49	0.25
MJ 7	151.53	30.94	6.45	1.31
MS 4	94.90	36.20	2.43	0.92
MS 5	55.06	29.49	1.44	0.73
MS 6	50.40	28.33	1.30	0.72
MS 7	158.71	112.34	18.96	13.35

Table 5.7: Optimisation statistics for N^{th} derivative passage times

The extensibility of the optimisation algorithm was tested and the results are shown in Figures 5.16-5.19. The iteration counts and time required to achieve convergence are shown. The order of derivative minimisation was varied from 2 (acceleration) to 7. The number of knots in the trajectory was varied from 2 to 14. The knots were initialised randomly in Cartesian space, and the optimisation was performed for both cyclic and non-cyclic trajectories.

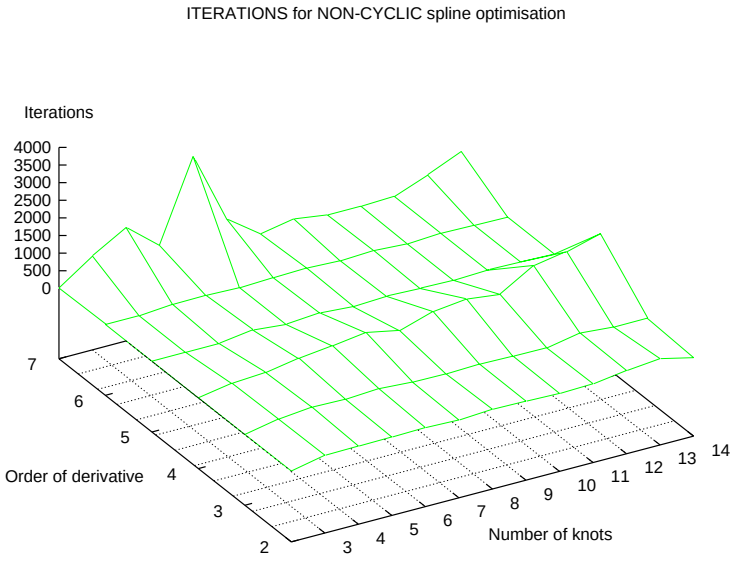

Figure 5.16: Iteration counts for multiple N^{th} derivative optimisations

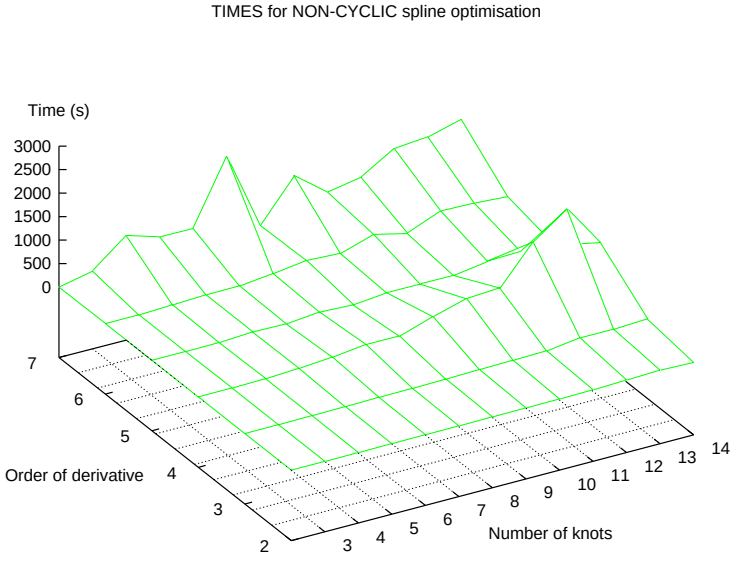

Figure 5.17: Timings for multiple N^{th} derivative optimisations

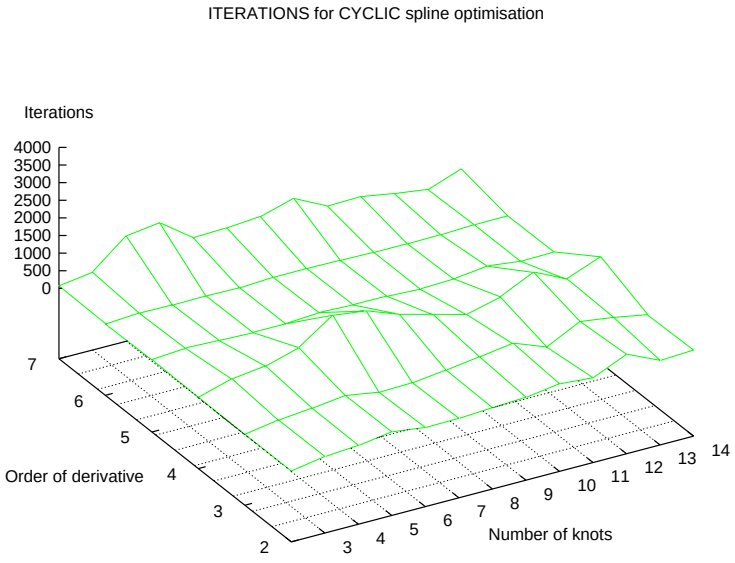

Figure 5.18: Iteration counts for multiple N^{th} derivative optimisations

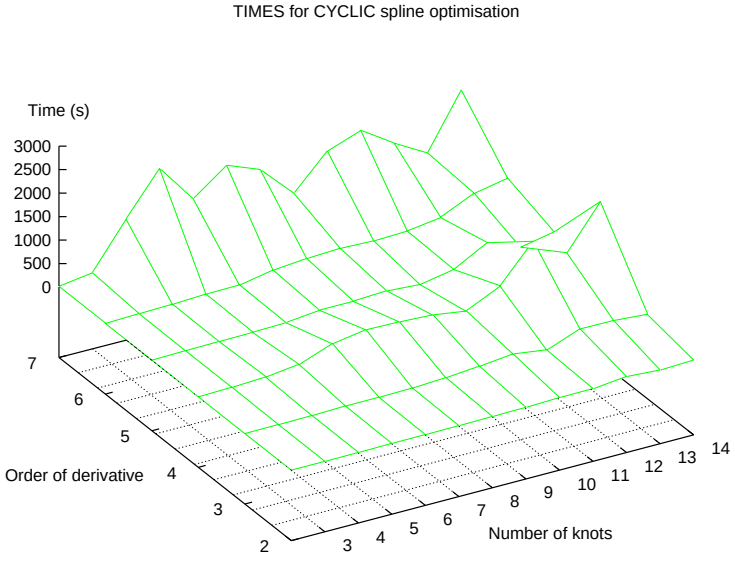

Figure 5.19: Timings for multiple N^{th} derivative optimisations

5.3.3 Dynamic optimisation

The Simplex algorithm was applied to the models requiring dynamic optimisation. These were the minimum torque (MT), minimum torque change (MTC), minimum torque postures (MTP) and minimum torque postures virtual trajectory (MTPVT) models. The corresponding algorithmic methods were described in Sections 4.3.2 & 4.3.3.

These were the slowest models to compute, as is apparent from Table 5.5 on page 147. The actual iteration counts required by each of the seven test motions for torque and torque change optimisation are shown in Table 5.8. Due to the range of results the scale is logarithmic. The maximum iteration count was 50,000 but due to the nested structure of the Simplex algorithm implementation the termination test was not applied after a uniform number iterations so that some solutions have completed between 50,001 and 50,078 iterations.

Table 5.9 compares the torque and torque change cost at initialisation and following optimisation. Recall that motions were initialised with minimum angular jerk spline trajectories so that the initial costs are expected to be modest approximations to minimum torque and minimum torque change. The torque costs for the minimum torque change motions and vice versa are also shown.

The Simplex algorithm maintains $n + 1$ solution vectors for an optimisation in n dimensions. The convergence criterion was measured in terms of the variation between these solutions. As optimisation proceeds these solutions should convergence upon a single point. When the variation among solutions reduced to 0.01 the algorithm was considered complete. However, one aspect of the Simplex algorithm is that it is difficult to formulate an exact convergence criterion. The solution may oscillate around a minimum for an indefinite number of iterations and the additional limit of 50,000 iterations was also necessary.

Figures 5.20-5.25 show the mean and variance during torque and torque change minimisation for each of the seven motions. Figure 5.23 reveals a

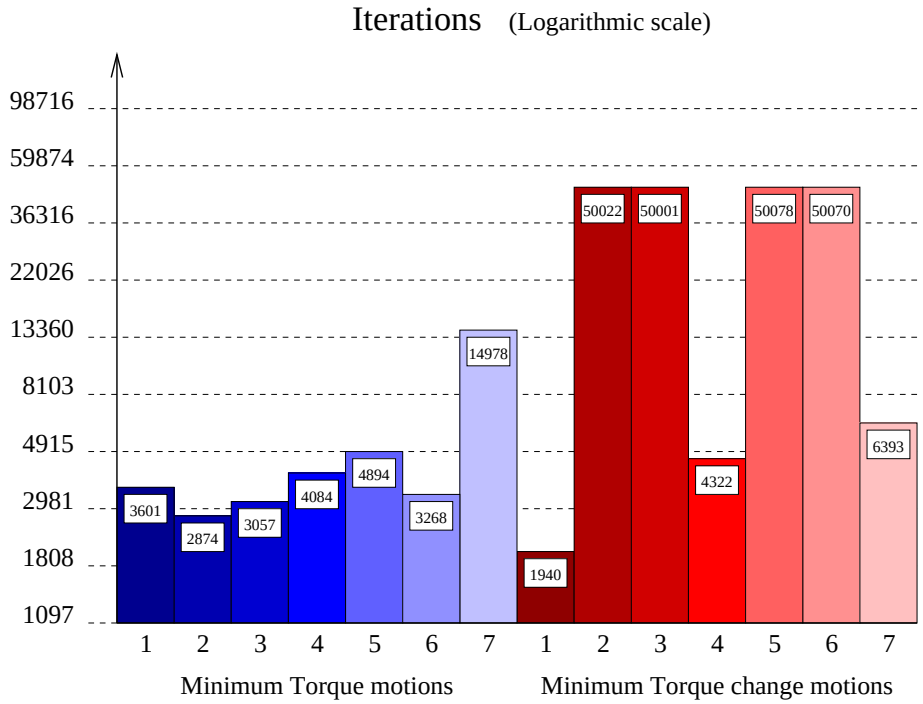

Table 5.8: Dynamic optimisation iteration counts

slightly unstable and gradual reduction in the variance of motion 7. Comparing this with Figure 5.22 reveals a corresponding very slight reduction in cost. Similar graphs with extended axes are not shown for the torque change optimisations because neither the mean nor the variance significantly reduced in the period up to 50,000 iterations. Minimum torque change motions 2,3,5 & 6 completed with variances of 0.09, 1.03, 0.84 & 0.18 respectively. All other iterations completed when the variance reached 0.01.

Motion	**Torque cost**			**Torque change cost**		
	Initial	Final	%	Initial	Final	%
MT 1	435.95	161.32	37%	0.62	1908.99	307902%
MT 2	91.80	70.80	77%	2564.02	4036.66	157%
MT 3	258.89	194.88	75%	10164.87	30714.79	302%
MT 4	419.20	153.19	37%	3.83	1863.06	48644%
MT 5	315.08	253.17	80%	6946.90	12163.39	175%
MT 6	243.63	202.88	78%	3181.78	4747.33	149%
MT 7	1339.68	616.48	46%	47.32	575.03	1215%
MTC 1	453.95	475.36	105%	0.62	0.69	111%
MTC 2	91.80	97.79	107%	2564.02	2343.62	91%
MTC 3	258.89	248.33	96%	10164.87	9186.12	90%
MTC 4	419.20	414.68	99%	3.83	2.98	78%
MTC 5	315.08	315.31	100%	6946.90	5764.85	83%
MTC 6	243.63	262.17	108%	3181.78	2732.95	86%
MTC 7	1339.68	1385.05	103%	47.32	35.43	75%

Table 5.9: Torque and torque change cost results for dynamic optimisation

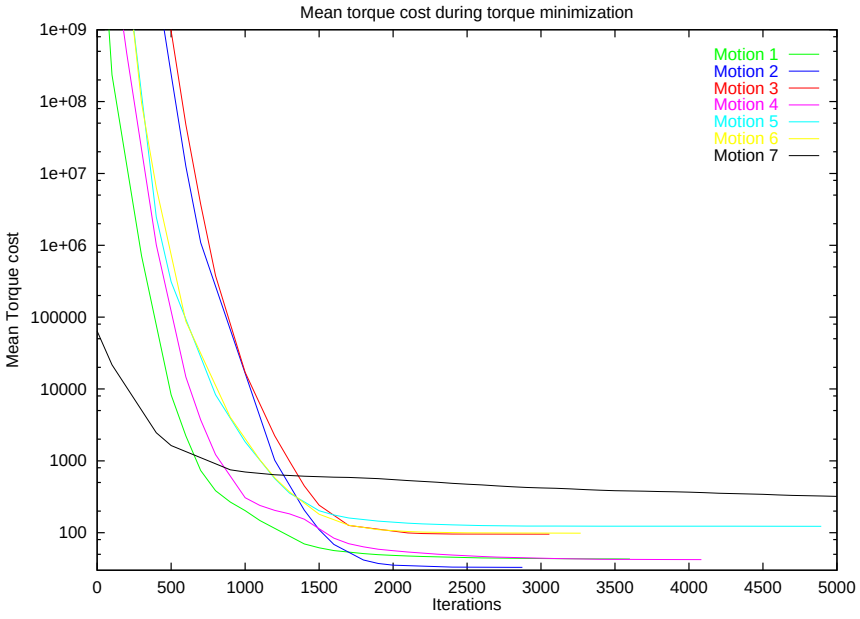

Figure 5.20: Mean cost for min. torque optimisations

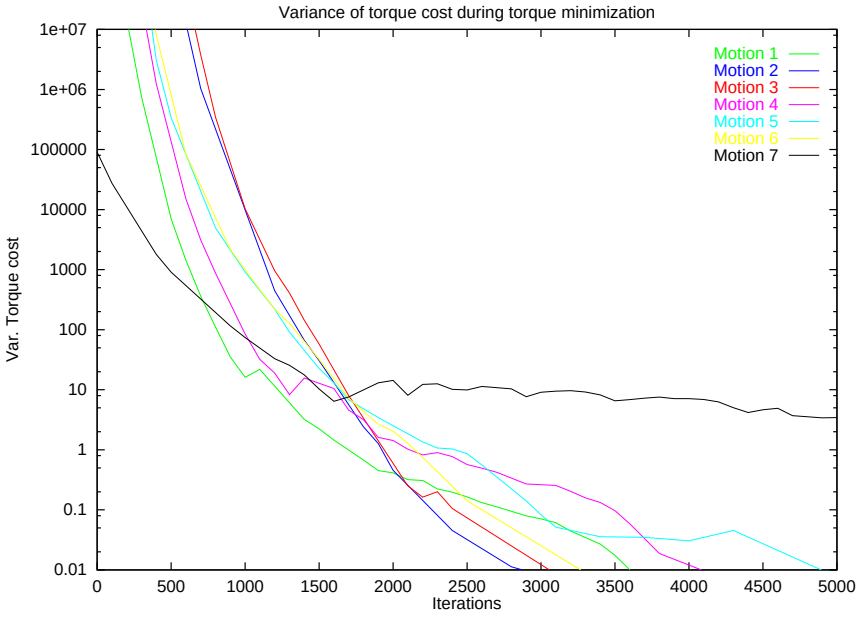

Figure 5.21: Variance of cost for min. torque optimisations

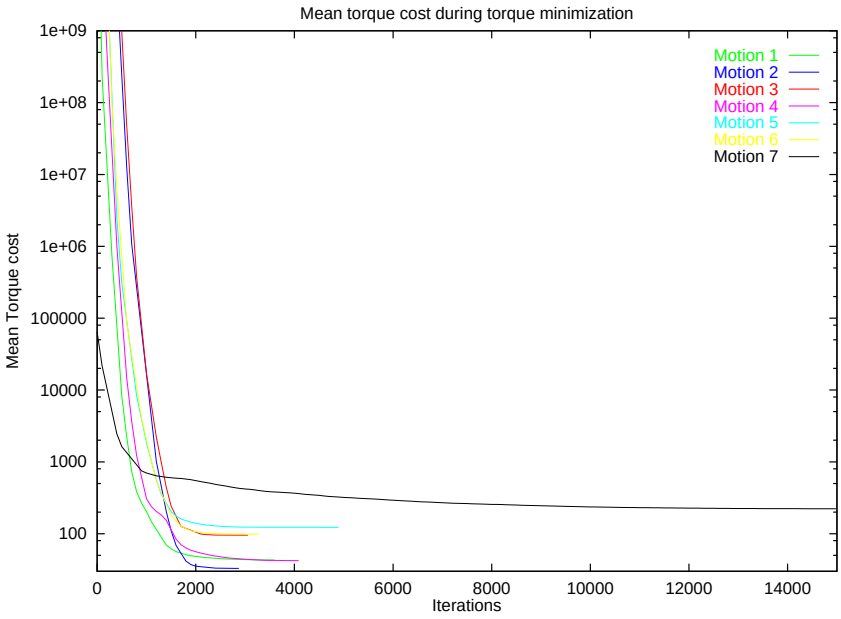

Figure 5.22: Mean cost for min. torque optimisations (extended axis)

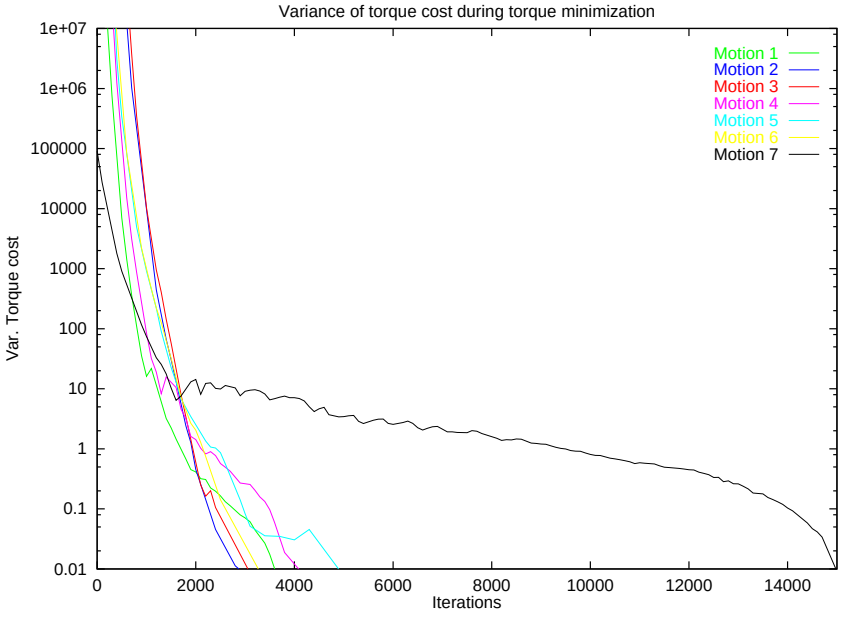

Figure 5.23: Variance of cost for min. torque optimisations (extended axis)

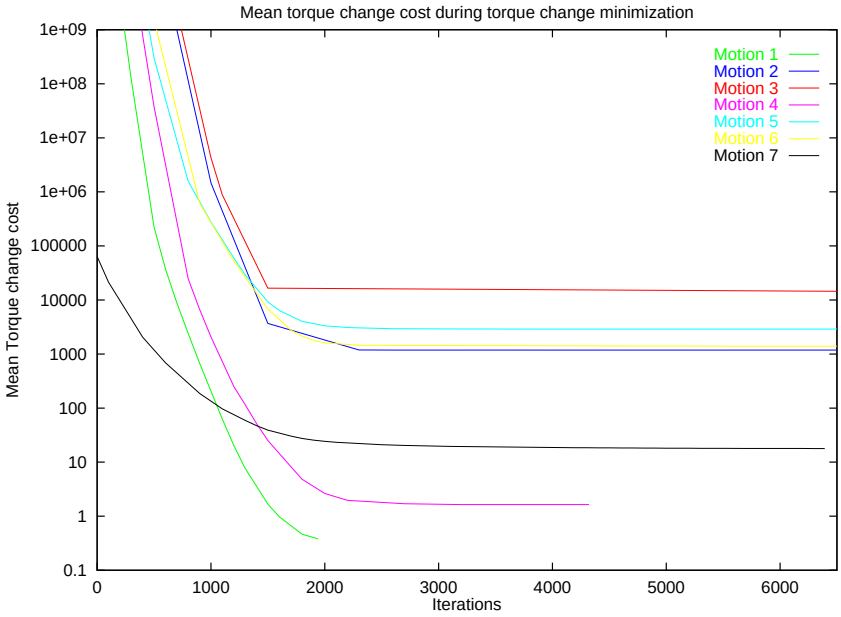

Figure 5.24: Mean cost for min. torque change optimisations

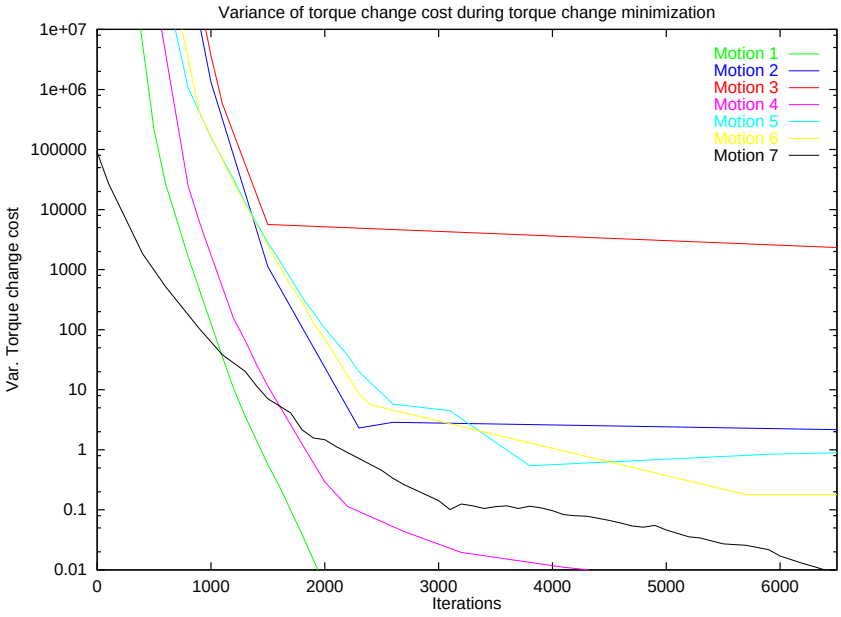

Figure 5.25: Variance of cost for min. torque change optimisations

5.4 Motion capture

This section presents frequency based analyses of the test motions recorded using motion capture. Motion capture was described in Section 4.5. It was found that a component of the motion capture conversion process was liable to yield high frequency oscillation artefacts. The recorded motion was therefore filtered by smoothing over a small window as described in Section 4.5.3. The results below show that the oscillations are indeed an artefact, and that the smoothing applied to remove them preserved the characteristics of the original motion.

Digital high and low pass filters were used to assess the energy in each joint angle trajectory. Table 5.11 shows the results of this analysis applied to motion 1 in full. The results for all seven motions are not displayed since the full data is cumbersome and not especially informative.

Chebyshev filters were used for the analysis because of their sharp cut-off characteristics at the pass frequency. Butterworth filters could have been used instead in order to ensure a maximally flat gain in the pass band but the aim was to categorise the energy in the signal according to it's frequency rather than accurately preserve a specific band of the spectrum. Fourteen coefficient filters were selected, since they were deemed to be quite sufficient for the required accuracy ensuring that the pass band ripple (variation in the signal gain at the pass frequency) was insignificant. High and low pass filters at $5Hz$, $10Hz$ and $20Hz$ integrated over the whole trajectory were used to categorise the energy in the trajectory. The ratios of energy in the high and low pass halves of frequency space are also displayed. The joint names in the table are abbreviated with 'L.' for left, 'R.' for right, 'S' for shoulder, 'H' for humerus, 'E' for elbow, 'W' for wrist, 'T' for torso, 'H' for head, and the angle of rotation is appended using the abbreviations, 'FE' for flex/extend, 'AA' for adduct/abduct, 'R' for rotate, 'B' for bend, 'T' for twist, and 'N' for nod.

Almost all of the energy occurs below the $5Hz$ pass band, and the ratio of energy above $10Hz$ to that below $10Hz$ is at most 0.001 for motion 1. Motion

1 did not have any oscillation artefacts due to the conversion process, and may be regarded as 'clean'.

Motions 2,3,4 and 5 all have some joints with spurious oscillations. The ratio of high to low pass energy in the $10Hz$ pass band was found to be a good discriminator for the joints suffering the oscillation artefact.

Table 5.10 shows the mean and variance of the energy ratios for each motion. These statistics are presented for the unsmoothed data, for the 'safe' joints that do not exceed a cutoff of a 2.5% energy ratio at $10Hz$ and for the final data in which the artefacts have been smoothed. The large drop in this ratio obtained by excluding the unsafe joints reflects the fact that the unsafe joints are separated by a large margin from the safe joints and the cutoff threshold is not an arbitrary point on a continuum of values. The unsafe joints are collected in Table 5.12 where each joint name is postfixed with the index of the motion it is from.

Besides the separation between the ratios of safe and unsafe joints, another fact supports the assumption that the high frequency components are artefacts. The joints demonstrating the artefact vary among the motions. Motions 1, 6 and 7 do not have any artefacts (and the ratio at $10Hz$ is at most 0.002). But motions 2,3,4 and 5 have the artefact in joints 13, 5 & 9, 2 & 2, and 1,6,7,12 & 13 respectively. In general therefore each joint has a demonstrably low ratio in at least 5 out of 7 cases. If the high frequency oscillation were related to a characteristic of human motion it seems unlikely that it would reveal itself only intermittently and in different joints of the body.

Smoothing the unsafe joints by averaging adjacent samples reduces the energy ratio at each frequency. The energy itself above $5Hz$ was also reduced to 8% of its original value, the energy above $10Hz$ to 10% of its original value and the energy above $20Hz$ to 11% of its original value. The energy *below* $5Hz$ however, was only reduced to 81% of its original value indicating that the low frequency motion has been mostly retained.

Motion & Joint set	**5*Hz* ratio**		**10*Hz* ratio**		**20*Hz* ratio**		**Num.**
	Mean	Var.	Mean	Var.	Mean	Var.	**Joints**
Motion 1 Unsmoothed set	0.002	0.002	0.001	0.001	0.000	0.000	20
Motion 1 Safe set	"	"	"	"	"	"	"
Motion 1 Corrected set	"	"	"	"	"	"	"
Motion 2 Unsmoothed set	0.028	0.024	0.008	0.008	0.003	0.003	20
Motion 2 Safe set	0.018	0.007	0.005	0.002	0.002	0.001	19
Motion 2 Corrected set	0.016	0.009	0.005	0.003	0.002	0.002	20
Motion 3 Unsmoothed set	0.034	0.038	0.007	0.008	0.001	0.001	20
Motion 3 Safe set	0.016	0.013	0.003	0.003	0.000	0.001	18
Motion 3 Corrected set	0.029	0.033	0.005	0.006	0.001	0.001	20
Motion 4 Unsmoothed set	0.009	0.011	0.007	0.008	0.003	0.004	20
Motion 4 Safe set	0.004	0.004	0.003	0.003	0.001	0.001	18
Motion 4 Corrected set	0.006	0.006	0.004	0.005	0.002	0.002	20
Motion 5 Unsmoothed set	0.054	0.060	0.014	0.013	0.005	0.004	20
Motion 5 Safe set	0.015	0.011	0.006	0.005	0.002	0.002	15
Motion 5 Corrected set	0.019	0.014	0.006	0.005	0.002	0.002	20
Motion 6 Unsmoothed set	0.019	0.017	0.002	0.002	0.000	0.000	20
Motion 6 Safe set	"	"	"	"	"	"	"
Motion 6 Corrected set	"	"	"	"	"	"	"
Motion 7 Unsmoothed set	0.004	0.003	0.002	0.002	0.001	0.001	20
Motion 7 Safe set	"	"	"	"	"	"	"
Motion 7 Corrected set	"	"	"	"	"	"	"

Table 5.10: Energy statistics for all motion captures

Joint	**5*Hz***			**10*Hz***			**20*Hz***		
	LP	HP	Ratio	LP	HP	Ratio	LP	HP	Ratio
L.SFE	15.127	0.003	0.000	15.339	0.001	0.000	15.497	0.000	0.000
L.SAA	7.395	0.001	0.000	7.483	0.000	0.000	7.554	0.000	0.000
L.HR	18.797	0.017	0.001	19.019	0.008	0.000	19.196	0.002	0.000
L.EB	11.367	0.004	0.000	11.529	0.002	0.000	11.649	0.000	0.000
L.WR	2.699	0.014	0.005	2.759	0.007	0.002	2.798	0.002	0.001
L.WFE	0.480	0.007	0.016	0.483	0.003	0.006	0.487	0.001	0.001
L.WAA	2.724	0.003	0.001	2.767	0.001	0.000	2.798	0.000	0.000
R.SFE	10.806	0.011	0.001	10.823	0.003	0.000	10.886	0.001	0.000
R.SAA	30.883	0.008	0.000	31.041	0.003	0.000	31.242	0.001	0.000
R.HR	4.117	0.014	0.004	4.160	0.005	0.001	4.203	0.002	0.000
R.EB	147.909	0.010	0.000	149.420	0.003	0.000	150.703	0.001	0.000
R.WR	62.882	0.074	0.001	63.663	0.037	0.001	64.273	0.009	0.000
R.WFE	7.937	0.016	0.002	8.181	0.008	0.001	8.329	0.003	0.000
R.WAA	18.391	0.028	0.002	18.656	0.012	0.001	18.847	0.003	0.000
TFE	7.315	0.002	0.000	7.427	0.001	0.000	7.506	0.000	0.000
TAA	0.524	0.001	0.003	0.525	0.001	0.001	0.528	0.000	0.000
TR	3.974	0.008	0.002	4.027	0.003	0.001	4.067	0.001	0.000
HN	33.281	0.006	0.000	33.771	0.002	0.000	34.133	0.001	0.000
HT	25.600	0.007	0.000	26.014	0.003	0.000	26.304	0.001	0.000
HR	48.844	0.014	0.000	49.540	0.005	0.000	50.052	0.002	0.000

Table 5.11: Energy response for unfiltered motion capture 1

Joint	**5*Hz***			**10*Hz***			**20*Hz***		
	LP	HP	Ratio	LP	HP	Ratio	LP	HP	Ratio
R.WFE2	1.945	0.467	0.240	2.046	0.141	0.069	2.177	0.049	0.023
L.WR3	1.186	0.118	0.099	1.245	0.042	0.034	1.287	0.007	0.006
R.HR3	1.066	0.310	0.291	1.092	0.050	0.046	1.116	0.009	0.008
L.HR4	1.425	0.075	0.052	1.438	0.052	0.036	1.449	0.021	0.014
R.WR4	18.758	1.201	0.064	19.062	0.996	0.052	19.275	0.536	0.028
L.SFE5	0.587	0.089	0.151	0.644	0.019	0.030	0.642	0.007	0.010
L.WFE5	1.193	0.114	0.095	1.246	0.035	0.028	1.274	0.014	0.011
L.WAA5	0.369	0.073	0.197	0.382	0.010	0.026	0.379	0.005	0.013
R.WR5	32.425	5.572	0.172	35.416	1.380	0.039	36.693	0.385	0.010
R.WFE5	2.913	0.714	0.245	2.857	0.189	0.066	2.816	0.045	0.016

Table 5.12: Energy response for joints with high frequency artefact

Chapter 6

Experimental analyses, criticism & review

Conclusion

This thesis has described the design and implementation of a synthetic motion production environment. The motivation and theoretical foundations have been established, software has been written and tested, and synthetic motions have been subjected to human evaluation.

This chapter concludes the thesis. Conclusions are presented. Criticisms are discussed. The achievements of the work are summarised, and further research directions are outlined.

The analysis of results presented in Chapter 5 is divided into conclusions relevant to five topics: the synthetic models, the significance of motion features, recommendations for model selection, the N^{th} derivative method, and the dynamic optimisation method.

The next section presents criticisms -most of which are discussed in terms of their relevance to the experimental method, and conclusions. Finally, the chapter concludes with sections summarising the work accomplished and outlining further research directions.

6.1 Analysis

The results of psychovisual testing revealed both the perceived naturalness and similarities of motions produced with each of the fourteen synthetic models and the recorded human motion. The results were summarised in Tables 5.2 & 5.3 on page 134 and Figures 5.3 & 5.4 on page 135. The implications of these results are discussed below with respect to each motion production model, features of motion, and model selection in Sections 6.1.1-6.1.3.

The results displayed in Tables 5.2 & 5.3 show the mean values, which are averages of ten subject's responses. In order to assess the statistical comparability of the results, the variance among the subject's responses was analysed. A two-way ANOVA of the responses to motions displayed by the robot with 15 levels for the model and 7 levels for the motion reveals significant main effects for both the motion ($F(6, 945) = 105.6$, $p < 0.0005$) and the model ($F(14, 945) = 3.79$, $p < 0.001$), and interaction between motion and model ($F(84, 945) = 1.74$, $p < 0.0005$). The same analysis applied to the motions displayed using the CG figure reveal significant main effects for both the motion ($F(6, 945) = 3.66$, $p < 0.002$) and the model ($F(14, 945) = 7.31$, $p < 0.0005$) but no evidence was found for interaction between motion and model. Further statistical comparisons may be found in Table 6.1 which shows the results of one-tailed two element contrasts between the mean values for each model averaged over the 10 subjects and 7 motions. The two element contrasts yield $t(1035)$ statistics and p-values that have been converted into certainty values equal to $100 \times (1-p)$. Where the contrast revealed a significant statistical difference (greater than 95% certainty) between two model's means a $>$ or $<$ symbol appears beside the numerical value according to whether the row's model is better or worse than the column's model.

The results of computational testing revealed the efficiency and stability of the N^{th} derivative minimisation algorithm, and the convergence characteristics of the dynamic optimisation process. Conclusions based on these results are discussed in Sections 6.1.4 & 6.1.5 respectively.

CG motion models

	MV	MAV	MA	MAA	MJ	MAJ	MS	MAS	MT	MTC	MTP	MTPVT	EPH	MJVTC	HUM
MV	-	60.3%	< 99.9%	< 99.9%	< 98.7%	< 99.9%	< 99.2%	< 99.9%	91.6%	< 99.9%	< 99.9%	< 99.9%	> 97.3%	< 99.9%	< 99.9%
MAV	60.3%	-	< 99.9%	< 99.9%	< 99.4%	< 99.9%	< 99.6%	< 99.9%	94.9%	< 99.9%	< 99.9%	< 99.9%	> 95.3%	< 99.9%	< 99.9%
MA	> 99.9%	> 99.9%	-	5.8%	8.3%	5.8%	84.3%	54.4%	> 98.0%	54.4%	64.5%	94.5%	> 99.9%	71.2%	77.2%
MAA	> 99.9%	> 99.9%	5.8%	-	83.3%	0.0%	78.3%	54.4%	> 96.6%	63.1%	5.9%	< 96.6%	> 99.9%	78.3%	83.3%
MJ	> 98.7%	> 99.4%	8.3%	83.3%	-	83.3%	57.4%	86.0%	0.4%	90.4%	79.4%	< 99.7%	> 99.9%	< 96.0%	< 97.3%
MAJ	> 99.9%	> 99.9%	5.8%	0.0%	83.3%	-	78.3%	54.4%	> 96.6%	63.1%	5.9%	< 96.6%	> 99.9%	78.3%	83.3%
MS	> 99.2%	> 99.6%	84.3%	78.3%	57.4%	78.3%	-	8.4%	5.1%	86.8%	73.7%	< 99.5%	> 99.9%	94.1%	< 96.0%
MAS	> 99.9%	> 99.9%	54.4%	54.4%	86.0%	54.4%	8.4%	-	> 97.3%	5.8%	60.3%	< 95.7%	> 99.9%	74.9%	0.4%
MT	91.6%	94.9%	< 98.0%	< 96.6%	0.4%	< 96.6%	5.1%	< 97.3%	-	< 98.5%	< 95.3%	< 99.9%	> 99.9%	< 99.5%	< 99.7%
MTC	> 99.9%	> 99.9%	54.4%	63.1%	90.4%	63.1%	86.8%	5.8%	> 98.5%	-	68.6%	93.2%	> 99.9%	67.2%	73.7%
MTP	> 99.9%	> 99.9%	64.5%	5.9%	79.4%	5.9%	73.7%	60.3%	> 95.3%	68.6%	-	< 97.6%	> 99.9%	82.4%	86.8%
MTPVT	> 99.9%	> 99.9%	94.5%	> 96.6%	> 99.7%	> 96.6%	> 99.5%	> 95.7%	> 99.9%	93.2%	> 97.6%	-	> 99.9%	5.1%	0.4%
EPH	< 97.3%	< 95.3%	< 99.9%	< 99.9%	< 99.9%	< 99.9%	< 99.9%	< 99.9%	< 99.9%	< 99.9%	< 99.9%	< 99.9%	-	< 99.9%	< 99.9%
MJVTC	> 99.9%	> 99.9%	71.2%	78.3%	> 96.0%	78.3%	94.1%	74.9%	> 99.5%	67.2%	82.4%	5.1%	> 99.9%	-	57.4%
HUM	> 99.9%	> 99.9%	77.2%	83.3%	> 97.3%	83.3%	> 96.0%	0.4%	> 99.7%	73.7%	86.8%	0.4%	> 99.9%	57.4%	-

Robot motion models

	MV	MAV	MA	MAA	MJ	MAJ	MS	MAS	MT	MTC	MTP	MTPVT	EPH	MJVTC	HUM
MV	-	57.7%	91.9%	< 99.8%	< 99.5%	< 99.9%	87.0%	< 98.5%	90.0%	< 99.5%	< 98.5%	< 99.7%	54.6%	< 99.5%	84.4%
MAV	57.7%	-	8.6%	< 99.6%	< 99.2%	< 99.8%	82.5%	< 97.6%	86.2%	< 99.2%	< 97.6%	< 99.5%	53.1%	< 99.1%	79.3%
MA	91.9%	8.6%	-	92.5%	8.6%	< 95.3%	60.7%	78.2%	54.6%	8.6%	78.2%	91.3%	90.0%	87.9%	65.1%
MAA	> 99.8%	> 99.6%	92.5%	-	59.2%	59.2%	> 95.7%	74.6%	94.0%	59.2%	74.6%	53.1%	> 99.7%	60.7%	> 96.6%
MJ	> 99.5%	> 99.2%	8.6%	59.2%	-	68.0%	93.0%	66.6%	90.7%	0.0%	66.6%	56.2%	> 99.4%	5.6%	94.5%
MAJ	> 99.9%	> 99.8%	> 95.3%	59.2%	68.0%	-	> 97.4%	8.5%	> 96.3%	68.0%	8.5%	62.2%	> 99.8%	69.4%	> 98.0%
MS	87.0%	82.5%	60.7%	< 95.7%	93.0%	< 97.4%	-	5.3%	56.2%	93.0%	5.3%	94.9%	84.4%	92.5%	54.6%
MAS	> 98.5%	> 97.6%	78.2%	74.6%	66.6%	8.5%	5.3%	-	8.5%	66.6%	0.0%	72.0%	> 98.0%	65.1%	87.9%
MT	90.0%	86.2%	54.6%	94.0%	90.7%	< 96.3%	56.2%	8.5%	-	90.7%	8.5%	93.0%	87.9%	90.0%	60.7%
MTC	> 99.5%	> 99.2%	8.6%	59.2%	0.0%	68.0%	93.0%	66.6%	90.7%	-	66.6%	56.2%	> 99.4%	5.6%	94.5%
MTP	> 98.5%	> 97.6%	78.2%	74.6%	66.6%	8.5%	5.3%	0.0%	8.5%	66.6%	-	72.0%	> 98.0%	65.1%	87.9%
MTPVT	> 99.7%	> 99.5%	91.3%	53.1%	56.2%	62.2%	94.9%	72.0%	93.0%	56.2%	72.0%	-	> 99.6%	57.7%	> 96.0%
EPH	54.6%	53.1%	90.0%	< 99.7%	< 99.4%	< 99.8%	84.4%	< 98.0%	87.9%	< 99.4%	< 98.0%	< 99.6%	-	< 99.3%	8.5%
MJVTC	> 99.5%	> 99.1%	87.9%	60.7%	5.6%	69.4%	92.5%	65.1%	90.0%	5.6%	65.1%	57.7%	> 99.3%	-	94.0%
HUM	84.4%	79.3%	65.1%	< 96.6%	94.5%	< 98.0%	54.6%	87.9%	60.7%	94.5%	87.9%	< 96.0%	8.5%	94.0%	-

Table 6.1: Two-way contrast certainties for CG and robot motion models

6.1.1 Synthetic models

The perceptual results revealed that recorded human motion looks very natural when performed by a computer generated humanoid. When performed by a humanoid robot however, the human motion is not perceived as very natural relative to the other models. This may indicate that the performance by the robot is worse than that of the CG humanoid. The mechanical nature of the robot introduces physical control constraints and the robot is fundamentally unable to move as smoothly and gracefully as a human being. It may also be true that humans have different expectations regarding how a robot should move. For example, people may perceive motion performed by robots as natural when it conforms to whatever expectations they have learned from seeing robots moving in an industrial setting, or in science fiction films.

The similarity of human motion to each synthetic model reveals that no model emerges as being the most similar to human motion despite the wide range of results regarding the naturalness of motion. This, combined with the presence of a number of synthetic models that are perceived with a similar or greater naturalness than human motion reveals that similarity to human motion is not equivalent to perceptual naturalness. At least some of the criteria that humans use to assess the naturalness of motion are better embodied by synthetic models with robust mathematical descriptions. This revelation supports the case for synthetic motion generation over motion capture for animation.

Consistently poor models in terms of naturalness are the minimum Cartesian and angular velocity (MV & MAV) interpolations and the equilibrium point (EPH) motion. Velocity minimisation yields piecewise constant velocity with infinite spikes in the acceleration profile as discussed in Section 2.7.1. These acceleration spikes are visually apparent, and are probably responsible for the unnatural appearance of the MV and MAV motion. While the MV model is similar to the cluster of Cartesian and angular derivative minimisation models the MAV model is more isolated. This suggests that the acceleration spikes are more apparent through joint angle interpolation

-a plausible hypothesis since the acceleration spikes are explicitly embodied by every joint in motion simultaneously and sharp changes in the direction of hand motion sharply contrast with its otherwise gently curved nature.

It is interesting to observe that both MV and MAV motion are perceived as more natural relative to the other models on the robot than with the CG humanoid. This may be due to the implicit smoothing of the acceleration imposed by the physical constraints of accelerating the robot's limbs, so that the MV and MAV trajectories act like (very stiff) virtual trajectories on the robot. This is consistent with the conclusion discussed below, that the virtual trajectory process improves perceived naturalness.

The EPH model is probably the worst of all in terms of naturalness. This is almost certainly due to the unstable nature of limb placement due to the lack of a velocity damping term in the joint torque generators. Joints are thus accelerated towards their target angle with a torque exponentially proportional to the angular offset without consideration for the angular velocity of the limb. The result is an almost undamped oscillation about the target that is extremely apparent for quick motions. The failure of this model shows that forward dynamic simulation alone does not yield natural looking motion, and that the force generation function is critical. It is unlikely in fact that the EPH model offers any unique and beneficial characteristics for animation because it approximates a joint angle interpolation by interpolation in force space. In defence of the general equilibrium point theory however, it should be mentioned that these results pertain to a specific implementation and the conclusions do not extend to other equilibrium point models that may yield better motions. A description of the various equilibrium models may be found in D.3.9.

This may be demonstrated under the assumption that the torques required to maintain the postures throughout a point-point motion may be linearly interpolated between the torques required at the initial and final posture τ_0 and τ_1 respectively. Recalling from Section 3.2.7, the torque generated at a joint is based on the angle of the joint θ, and the λ parameter

for the joint (which is linearly interpolated between the λ parameters at the endpoints of the motion, λ_0 and λ_1). The λ parameters for the endpoints are chosen so that the torques generated at the joint when it is at the target angles are equal to the torques necessary to maintain the joint in a stable orientation (τ_0 and τ_1). Assuming without loss of generality that the duration of motion is $[0, 1]$ represented by time parameter t:

$$\lambda_i = \frac{1}{\beta_1}\ln\left(\frac{\tau_i}{\beta_0}+1\right) \tag{6.1}$$

$$\lambda(t) = \lambda_0 + t(\lambda_1-\lambda_0), \quad \tau(t) = \beta_0(e^{\beta_1(\theta(t)-\lambda(t))}-1) \tag{6.2}$$

By assuming the torque needed to balance the postures varies linearly between the endpoint postures, the torque may be substituted into the EPH force generation function to derive the target angle (the reverse of the usual usage).

$$\tau(t) = \tau_0 + t(\tau_1-\tau_0) \quad t\in[0,1] \qquad \text{by assumption} \tag{6.3}$$

$$\theta(t) = (1-t)\lambda_0 + t\lambda_1 + \frac{1}{\beta_1}\ln\left(\frac{(1-t)\tau_0+t\tau_1}{\beta_0}+1\right) \tag{6.4}$$

$$= (1-t)\theta(0) + t\theta(1) + \frac{1}{\beta_1}\left(\ln\frac{(1-t)\tau_0+t\tau_1+\beta_0}{(\tau_0+1)^{1-t}(\tau_1+\beta_0)^t}\right) \tag{6.5}$$

$$\text{since} \quad \theta(t) = \lambda(t) + \frac{1}{\beta_1}\ln\left(\frac{\tau(t)}{\beta_0}+1\right) \tag{6.6}$$

Equation 6.5 shows that the angle varies from a linear interpolation according to a complicated expression depending on β_0, β_1 and the torques τ_0 and τ_1. It turns out for reasonable choices of $\beta_1 \in [\frac{1}{20}, 2]$ and β_0 close to the stable gain of the joint divided by β_1 this complicated expression takes a maximum of 0.005 so that the angle is diverted by only 0.3 degrees from a linear interpolation. The EPH model could therefore be closely approximated by a linear joint angle interpolation virtual trajectory model. The curvature arising when the motion is slow enough for the EPH model to remain stable is thus explained. This curvature is evident in Figure 5.12 on page 142. Recall however, that the EPH model implemented in this work was a choice

from several different approaches to equilibrium point control. It is therefore not valid to dismiss the equilibrium point hypothesis based on these findings since other models do incorporate velocity damping and may yield different force generation functions.

The remaining Cartesian derivative minimisations (MA, MJ & MS) all yielded fairly good perceived naturalness but were never the best models. MS does not yield a significant improvement over MJ indicating that at most 3^{rd} order minimisation is necessary. The MA model also yields good results indicating that humans are most sensitive to the arch-shaped component of the velocity profile embodied by MA, MJ & MS models by comparison to the smoothed onset and termination of the velocity profile only embodied by the MJ & MS models (these features may be seen in Figure 5.5 on page 138). This is supported by the clustering of the three models in terms of similarity.

The remaining angular derivative minimisations (MAA, MAJ & MAS) were all consistently well perceived in terms of naturalness. They are likewise clustered together in terms of similarity. The results appear better than those of the Cartesian derivative minimisations indicating that angular interpolation is more natural than Cartesian interpolation. However, there is no statistical evidence to support this within the 95% threshold. The approximately equal naturalness among MAA, MAJ & MAS supports the conclusion that it is the arched component of the velocity profile that is most significant in terms of naturalness rather than the smoothed onset and termination of the velocity. Since the models are clustered with the Cartesian interpolations in terms of similarity it may be concluded that they are similar but possibly slightly better alternatives. The embodiment of gentle curvature into hand motions and smooth velocity profiles *at each joint* rather than just at the hand thus contribute to the perceived naturalness of motion.

The minimum jerk virtual trajectory model (MJVT) yields natural looking motion. Subjects ranked the model highly and the results are almost equal to the naturalness of human motion for the CG humanoid and exceed that of the human motion performed by the robot although there is no

evidence for a significant difference between the models' perceived naturalnesses. The motion is significantly more natural than that of MJ for the CG humanoid, and also significantly better than three of the minimum derivative interpolations (MV, MAV, MJ) indicating that the virtual trajectory process has a significant and beneficial effect on the naturalness. The result is not so strong for motions performed on the robot, but as mentioned above the physical nature of the robot compels the interpolated trajectories to act as virtual trajectories so that on the robot there is a relative improvement in these interpolations compared to the MJVT. The fact that MJVT is isolated in terms of similarity, is approached only by MTPVT, and is very different from MJ suggests that the virtual trajectory process introduces an identifiable and beneficial characteristic to motions and also clearly differentiates the resultant motion from the virtual trajectory itself. Virtual trajectories introduce skewed velocity profiles and corrective motions. The latter is probably more apparent and significant.

The minimum torque model (MT) yielded poor-medium results in terms of naturalness. By comparison to a number of better models (MA, MAA, MAJ, MAS, MTC, MTP, MTPVT, MJVTC, HUM) it is clear that the computational expense of this model is not justified. MT is also isolated from the other models. While it is possible that MT captures some specific characteristic of human motion not replicated by any of the other models, it is more likely that the model is simply isolated by its poor replication of human motion. The intuitive hypothesis that physical or metabolic effort minimisation leads to natural motions is therefore false. Alternative optimal characteristics of motion are therefore more likely hypotheses *e.g.* stability, simplicity, or consistency of muscular effort. The unnatural appearance of MT motion may be due to the bizarre trajectories that it sometimes produces. Besides being visually distracting, the unpredictability of MT motions makes the model especially awkward for an animator who may have to repeatedly add new knot points to artificially constrain a motion to a more natural path.

The minimum torque change model (MTC) yields good naturalness and

better than minimum torque. This supports the hypothesis that motions are planned using minimum torque change (see Section 2.7.9) rather than minimum torque in so far as humans themselves recognise that minimum torque change is visually more natural. The quality of the MTC motions demonstrates that minimal torque change might be a criterion assessed by the human brain when perceiving motion. However, it is also possible that minimal torque change approximates some other more easily assessed criterion such as smoothness in joint angle space. This is supported by the perceived similarity of joint angle and Cartesian interpolations to MTC, and by the fact that torque estimation requires an assessment of the dynamics of a figure in motion whereas smoothness is a purely kinematic property.

Since the computational requirements of MTC and MT are similar, it is safe to conclude that MTC should be used for animation in preference to MT. However, MTPVT, MTP, MJVT or even MAJ may be chosen in preference to MTC based on computational considerations and the similar or better perceived naturalness attributable to these models. The clustering of MTC among the derivative minimisations further supports the use of models such as MAJ in preference to MTC since they are also perceptually similar.

The minimum torque postures model (MTP) demonstrates a reasonable perceived naturalness by comparison with other models. It is isolated however, and in particular it is not similar to MT or MTC, nor are the results as good as MTC. It therefore does not capture the same qualities of motions embodied by these models as was intended, but nevertheless yields good motion.

Finally, the minimum torque postures virtual trajectory model (MTPVT) yields very good perceived naturalness. It is consistently better than MTP and thus supports the use of the virtual trajectory process, strengthening similar conclusions drawn from a comparison of MJ and MJVT. MTPVT is likewise very different from MTP indicating that the virtual trajectory process makes a significant and visually apparent change to the motion. The fact that it is most closely associated with the MJVT also suggests that the

results of the virtual trajectory process are not only identifiable, but are the most significant features in classification embodied by the MTPVT model.

6.1.2 Motion features

By considering the perceptual results comparatively it is possible to draw conclusions regarding the perceptual significance of features embodied by different models. Three easily identifiable features of the hand trajectory have been discussed: corrective hooks, curvature, and the shape of the velocity profile.

The velocity profile emerges as the most significant factor in generating a natural looking motion. This is supported by the large difference in perceived naturalness between the MV & MAV models versus the MA, MAA, MJ & MAJ, *etc.* models. The former two have exactly or approximately piecewise linear hand velocity profiles. The latter set however, has parabolic or bell shaped velocity profiles. The arched nature of the velocity profile is also much more significant than the smoothed onset and termination embodied only by MJ, MAJ and higher derivatives.

Corrective hooks were only demonstrated by the two models incorporating stable forward dynamics: MJVT and MTPVT. Both of these models were perceived as more natural overall than their counterparts without the virtual trajectory process: MJ & MTP. As discussed above, the virtual trajectory process introduces significant classifiable characteristics to motions improving the perceived naturalness. While these include both corrective hooks and a skewed velocity profile it is likely that the hooks and slightly inaccurate hand placement associated with fast motions is the more significant contributor to naturalness. This is supported by the fact that humans appear to pay little attention to slight adjustments in the velocity profile such as the difference between truly bell shaped, and merely arch shaped velocity profiles. The fact that trajectory hooks were replicated using purely mechanical spring principles also suggests that the hooks observed in human motions could be due to the elastic properties of the musculo-skeletal system, *i.e.* the hooks

emerged during at the end of a motion when the virtual trajectory point was stationary so that no further motion planning was taking place.

Lastly, the presence of curvature in hand trajectories may improve the perceived naturalness, but is less significant than the presence of hooks. This is supported by the statistically insignificant improvements demonstrated by the angular interpolations (MAA, MAJ & MAS) over the Cartesian interpolations (MA, MJ & MS). The interpolation of joint angles yields workspace dependent curvature. Since the curvature observed in human hand movements is also workspace dependent it is possible that humans have expectations regarding the curvature of hand movements based on their own experiences of motion production. However, it is not certain that the workspace dependent curvatures of joint angle interpolation and human motion are equivalent, and any correlations between the two may in fact be due to the physical nature of motion in articulated figures as was discussed in Section 3.2.4.

6.1.3 Model selection

The criteria by which it is necessary to select a motion production model vary according to the application. The application may be for CG virtual humanoids, or a humanoid robot. There may be a real-time requirement so that motion must be planned on the fly, or there may be time to prepare the motion off-line.

Based on the perceptual and computational results a number of model recommendations can be made. Firstly, it can be safely concluded that neither torque minimisation, nor torque change minimisation need be used. This is a relief since the computational efficiency is very poor, and the models require significant computational effort to implement.

For a CG humanoid a good choice with respect to simplicity of implementation and computational effort is the MAJ or even the MAA model. These models can be implemented in an extremely minimal fashion based purely on a given target point and the current joint angles and velocities, as was demonstrated in Section 3.3.2. Solutions could be evaluated quickly enough

to facilitate visualisation techniques displaying whole ranges of motions dependent on different motion specifications simultaneously.

If forward dynamic simulation is available this could be extended to a minimum angular jerk virtual trajectory. Unfortunately forward dynamics requires considerable implementation effort, but can usually be calculated at or close to real-time although it is unlikely that a range of solutions could be displayed simultaneously. Probably the best model choice assuming that forward dynamics is available is the MTPVT model, which requires some further effort since an optimisation method must also be implemented but can be solved quickly for very natural results.

In the case of a humanoid robot forward dynamics are implicit in the physics of the robot. A minimum angular jerk virtual trajectory motion could be planned in constant time using the on-line method mentioned above. This target motion path can act as a virtual trajectory by programming the joint torques according to the spring-damper model, thus yielding the velocity smoothing and hooks revealed to improve the naturalness of motion. If off-line planning is possible the MTP model would yield a motion plan that could be brought about in the same way, yielding the high quality MTPVT motion.

6.1.4 N^{th} derivative minimisation method

The results in Section 5.3.2 prove that the N^{th} derivative optimisation algorithm was valid. The 1^{st} to 4^{th} derivative minimisations applied to Cartesian and joint angle trajectories were correct and efficiently calculated. The key component of the minimum N^{th} derivative motion models -the general analytical N^{th} derivative solution presented in Section 4.3.1, is thus applicable from a computational perspective.

Table 5.6 on page 148 shows that convergence was achieved within a few hundred iterations. This was ratified by repeated trials shown in Table 5.7 on page 149.

A further analysis of the implementation demonstrated the extensibility

of the method. Figures 5.16-5.19 on pages 150-151 show that the method converges for splines with up to 14 knots, and up to 7^{th} derivative minimisation. The knot count is correlated with an increase in the number of iterations required to find the optimal knot passage times and the time required to calculate the solution. This is to be expected since the main work of the algorithm is to invert a compound matrix of size $(k+1)N/2$ for N^{th} derivative minimisation over k knots. The size of the compound matrix thus increases by $N/2$ for each extra knot.

Increasing the degree of the derivative minimised can have a more substantial impact depending on the number of knots -a size increase of $(k+1)/2$. The iteration counts and total convergence time increase dramatically as the degree reaches 7. It is thought that this is due to numerical precision issues affecting the behaviour of the gradient descent technique. It may alternatively be due to some unrecognised properties of the solution space, perhaps explaining the peaks in iteration and time costs observed in 4^{th} order minimisations. For example, the disproportionate variance revealed by snap minimisation of motion 7 revealed in Table 5.7 suggests that 4^{th} minimisation of cyclic splines seems to be particularly sensitive to the initial knot passage times. It is however a general conclusion of this thesis that 3^{rd} order minimisation is sufficient for the purpose of motion production -an observation that reduces the significance of the iteration costs at higher derivatives.

The dimensionality of the splines is relatively insignificant. The compound timing matrix is the same for each dimension and need be inverted only once. Since the inversion process dominates all other factors the dimensionality of the spline does not substantially affect the total solution time. Table 5.7 shows that approximately 10% more iterations and $3\frac{1}{2}$ times as much time is required for 30 dimension angular splines compared to 3 dimension Cartesian splines.

The use of cyclic or non-cyclic splines is significant. This is to be expected since non-cyclic splines offer fixed boundaries at either end so that optimal passage times can effectively propagate towards the centre of the spline dur-

ing optimisation. In the cyclic case all the passage times except the first are free to move and the optimal passage times must propagate around the cycle until they are all stable. Table 5.7 reveals the substantial increase in cost for cyclic splines (motion 7 was cyclic).

In general it is safe to conclude that the method is sufficient for motion generation and may be extended to handle higher derivatives and knot counts. For 7^{th} order minimisation and above it would be advisable to use double precision floating point accuracy (single precision was used). For extreme values of these parameters it might also be advisable to partition the spline into adjacent non-cyclic segments since these could be optimised separately. Indeed, a hierarchical process of optimisation could also be implemented according to which the passage times of each segment were determined prior to refinement of their internal knot passage times. The key component of the algorithm is the analytical method expounded in Section 5.3.2, and while gradient descent has been shown to be sufficient, a wide variety of optimisation processes could be applied to specific instances of the method with particular requirements or parameter ranges.

6.1.5 Dynamic optimisation

The results in Section 5.3.3 demonstrate that Simplex method applied to torque and torque change optimisation yielded reasonable results. Convergence to a parameterised minimum variance was achieved in 10 of 14 cases, and to an approximately constant variance in the remaining 4 cases.

Table 5.8 on page 153 shows that all of the minimum torque motions converged to 0.01 variance (a satisfactory criterion for iteration to terminate). The first six motions converged in around 4000 iterations, while motion 7 with is long duration required closer to 15,000 iterations. Figures 5.20 and 5.21 on page 155 show the corresponding reduction in variance and mean cost[1]. The mean cost function has a clear 'knee' for each optimisation. While the knee of

[1]Recall that the Simplex algorithm maintains $n + 1$ solution vectors for optimisation in n dimensional space, and that these vectors converge on a minima as a group.

motion 7 is quickly reached, the variance is fairly constant for around 13,000 iterations before it abruptly reduces. This may be seen in the extended versions of the two graphs found on page 156.

Three minimum torque change motions converged to 0.01 variance but the remaining four continued to the maximum number of iterations. This does not however indicate that the optimisation did not converge. It is notoriously difficult to formulate consistent convergence criteria for the Simplex algorithm. Figures 5.24 & 5.25 on page 157 show that the knee has been achieved for each of the four motions that did not reduce to 0.01 variance. The extended versions of these graphs were not shown because the mean and variance continued with the approximate values shown until the termination after 50,000 iterations. It is thus reasonable to assume that despite not reaching the arbitrarily chosen 0.01 variance, the optimisation converged to a minima. The persistent variance reflects the rough nature of the solution space.

The results do however suggest that torque change is harder to minimise than torque. The higher final variance for motions MTC 2, 3, 5 & 6 indicates that the solution space is rougher than that of the torque motions. MTC motions 1 & 7 do converge in fewer iterations than their MC counterparts (Table 5.8) but in these cases the initial solution was closer to the minimum torque change solution as is shown in Table 5.9 on page 154. Minimum torque change is more difficult to optimise since postures within a given time window are constrained to have similar accelerations by the torque change criterion. Adjustments to a given posture can thus have a dramatic effect on the total torque change cost, making the solution space very stiff.

This table also reveals that MTC motion 1 does not in fact reduce in cost due to the optimisation process. The Simplex method converged on a solution near to the initial motion but was unable to improve the cost. This is not surprising since in this case the initial cost was highly optimal. Motion 1 is a slow point-point hand trajectory descending vertically.

The initial optimality of MTC motion 1 is symptomatic of the fact that

minimum jerk splines (which were used in the initialisation of the Simplex algorithm) are closer to minimum torque change motions than minimum torque motions. This similarity is supported by three observations. Firstly, it can be seen that the initial torque change costs are much closer to their optimised costs than the minimum torque motions, despite both optimisations beginning with the same initialisation vectors.

Secondly, torque minimisation dramatically increases the torque change cost. Minimum torque change motions are thus definitely not the same as minimum torque motions according to the numerical results. Conversely, the torque change minimisation has a relatively insignificant affect on the torque cost of the motions. These two results suggest that the minimum jerk motions more closely resembled minimum torque change than minimum torque motions because torque change minimisation does not appear to alter the motions significantly from their initial forms.

Thirdly, torque is approximately proportional to angular acceleration in the presence of a relatively constant moment of inertia, *i.e.*, when the mass configuration doesn't change suddenly and cause a large effect on the torque at a particular joint (this is very much the case in motion 1). Likewise, torque change approximates acceleration change, *i.e.,* jerk.

One conclusion of this observation is that torque minimisation might converge more quickly if minimum angular *acceleration* splines were used to initialise the optimisation rather than the minimum angular jerk splines that were used.

The dynamic minimisations were very slow to calculate by comparison to other motion models. The results generated demonstrated that dynamic minimisation is at least possible, although it cannot be proven that the results are globally minimal. The persistent variance observed in Simplex solutions reveals the rough nature of the solution space since even in small clusters, sets of solutions do not enclose a region of smoothly decreasing cost. More complex global minimisation strategies are applicable and could provide stronger reassurances of absolute optimality but require a significant increase in soft-

ware coding or specialised hardware. Implementing analytical derivatives of the sum squared torque cost metric or the torque change metric would be awkward but would facilitate faster optimisation strategies taking advantage of the cost function's derivative. The BFGS algorithm for example, would yield faster results than the Simplex algorithm which does not depend on the derivative.

However, a general conclusion of this thesis is that dynamic optimisation of torque or torque change is not necessary for motion production and these problematic issues are irrelevant. Minima were found with the Simplex method, although these could not be guaranteed to be the global minima. Despite the difficulties of optimisation over an intractably large and badly behaved solution space reasonable results were obtained, psychovisual experiments facilitated, and significant conclusions were drawn.

6.2 Criticisms

The following three sections discuss criticisms of the research described in this thesis. They deal with the experimental design and method, conclusions based on the experimental results and the optimisation algorithms used to solve the more time consuming synthetic models.

6.2.1 Experimental method

A number of criticisms may be levelled at the experimental method. The motions designed to test the various models and the method by which they were displayed to subjects were not ideal.

Firstly, the head motion was specified to gaze at the hand throughout motion as was discussed in Section 5.1.1. This led to the impression that the humanoid was concentrating on the task and the location of the hand. However, other research has revealed that in fast reaching motions humans tend to saccade[2] the gaze point to the target at the onset of motion. Particularly in

[2] 'Saccade' is the term for a rapid eye movement redirecting the line of sight.

the fast motions it would therefore be more natural specify the motion of the head so as to gaze at the endpoint at the onset of motion. This psychovisual cue might have influenced subjects to interpret the motions differently.

This issue is symptomatic of a larger and more serious complaint. The *context* of motion is not well defined. Subjects were not told the goals of the humanoids performing the motions. It is not clear from the abstract hand motions what the *purpose* of the motion is. This might have a significant effect on subject's estimates of naturalness but also invades the area of motion specification. A motion to place the hand so as to lift a cup would look strange if the goal were instead to catch a ball and vice versa. In order to judge the naturalness of a motion effectively therefore, a subject may need to know the context of the motion. However, the production of motions according to such varied contexts requires an effort in both motion specification as well as automatic motion synthesis. The motions displayed to subjects were thus deliberately as vague as possible regarding the context of the motion in order to obtain reasonable general results regarding the motions models. On the other hand, experiments using the synthetic models presented in this thesis might reveal different trends when the motion was more adequately situated and the goals of the motion were explicit.

The uniformity of specifications for each motion model including the human motion allowed the models to be compared with each other. Unfortunately it is not possible to specify motions in the same way for a human being. The timing of the motions could not be rigidly specified without making the motion pathologically specific and artificial. It was therefore necessary to record the human motions first and derive corresponding specifications to feed into the synthetic models. This means that the motion specification process was post hoc based on the resulting human motion rather than the internal goals that were impossible to obtain. It could be argued that the information derived from the human motion and used for synthetic motion specification invalidates the automatic nature of synthetic motion generation. However, this post hoc process was only necessary to facilitate consistency

across all the models including the human motion. It is also more likely that an animator would like to synthesise motion similar to human motion according to its resultant characteristics rather than according to the most human-like interpretation of motion goals. In the latter case, the resultant motion might not satisfy the animators requirements.

Two final criticisms of the motion specifications are that there were only seven motions and that they were designed to contrast a set of motion features chosen prior to the analysis of the results. A larger set of motion samples might reveal different results according to other motion features that may not yet be known. This relates to the context problem since it suggests that according to the type and goals of motion required, humans might expect different visual characteristics. This problem can only be solved by further experimentation and illustrates that the range of possible motion goals and contexts is too large to be solved by a single set of psychovisual experiments.

Besides the design of the motions used for psychovisual testing, the display methods are also imperfect. For example, the decision to use videos of the robot in motion rather than place subjects in front of the actual robot was taken for practical reasons. However, it is possible that subjects might perceive the motion differently when exposed to the actual robot. Subjects might have a better perception and attribute different qualities to the motion models. It was confirmed through verbal communication with Sony robot researchers that subjects do experience different emotional responses to robot behaviour when they are situated with the robot in question as opposed to watching it by video. Also, subjects might respond differently depending on their impression of the robot. This might be affected by the appearance of the robot and its surroundings so that a robot that is obviously dressed to look like a human by concealing its inner mechanics with synthetic skin might affect humans' expectations of how it should move. The robot used for these experiments was partially covered in semi-transparent plastic sections so that its mechanical nature is apparent. Besides assuming the kinematic configuration of a humanoid there is no effort made to simulate the appear-

ance of a human. However, taking specific steps to control the appearance of the robot merely selects a particular prototype of robot appearance and does not solve the general problem of the robot's appearance and setting. This was considered to be a problem too specific to tackle along with the development and testing of the motion production models themselves.

Finally, the same criticism can be levelled at the CG humanoid. The appearance may likewise affect humans' expectations of how the figure should move and how natural the motions appear. An effort was made to render the CG humanoid in as realistic a way as way feasible but it is not necessarily valid to generalise the results to all CG humanoids. For example, Hodgins *et al.* [Hodgins et al. (1998)] found that different qualities of CG humanoids including wire-frames and polygonal models were perceived differently by subjects. Different responses might have been elicited in the psychovisual testing depending on the quality and appearance of the humanoid figure. This issue limits the generality of the psychovisual results with respect to CG humanoids but there is no easy solution besides repeating the experiments using a range of humanoids and either validating the synthetic motion models for different rendering conditions or revealing a set of motion models ideally suited to each condition.

6.2.2 Experimental conclusions

Factors reducing the general validity of the experimental conclusions are discussed below.

The chiefest concern is the relatively small number of subjects and consequently high variance in the results. The naturalness perception results had inter-subject variances of 10.5% and 7.7% for the CG and robot humanoids respectively. The similarity perception experiments had variances of 9.0% and 9.3%. These variances *do* indicate that it was reasonable to draw conclusions from the results. However, due to the small number of subjects it was necessary to normalise each subjects' responses to zero mean and unit variance before comparing the results, as described in Section 5.1.2. This

prevents an absolute comparison between the computer graphic humanoid and the robot motions. The conclusion in Section 6.1.1 that the performance of genuine human motion by the robot may be worse than by the computer graphic humanoid is therefore based purely on the poor quality of the human motion *relative* to all the other motion models.

In order to make robust conclusions it would be desirable to repeat the experiments taking into account the factors discussed in the previous section but most significantly, increasing the number of subjects. Limiting factors included the time constraints, the breadth of research covered in this thesis, and the fact that financial incentives for subjects were not available. It was thought that 10 subjects per experiment was an ambitious but attainable figure. The experiments were completed in about 30 individual sessions of around an hour each when voluntary subjects completed one or two experiments depending on their tolerance of the repetitive task.

Of secondary concern, the conclusions regarding the synthetic models may be criticised as follows. The minimum torque (MT) and torque change (MTC) models were based on minimum jerk optimisations. In producing the MT and MTC motions, the space of motions defined by Cartesian knots interpolated using a minimum jerk spline was searched for an optimal motion. This casts a shadow over any conclusions comparing the relative similarity of those models or the uniqueness of MT and MTC motions. The use of minimum jerk splines was a compromise necessitated by the extremely awkward problem of attaining MT and MTC optimality. The use of minimum jerk splines is supported by research that has already shown the models to be similar but such assumptions clearly reduce the strength of any further conclusions in agreement with those results. However, while the minimum jerk motions produced by the MJ model were specified with only 2,3 or 4 knots (given in the input motion specification) the density of knots was much higher for the MT and MTC search spaces, *i.e.*, by increasing the knot density dramatically, that resultant space of motions is much larger. While MTC motions were similar to MJ motions both perceptually and according

to perceived naturalness, the MT motions were dissimilar to MJ motions by both accounts. This proves that the search space facilitated significant and perceivable deviation from minimum jerk motion and lends strength to the conclusion that it is not merely an experimental artefact that MTC and MJ motion are similar.

Regarding the similarity of MJ and MTC motion, some existing research may throw further light on their apparent similarity. Nakano et al. (1999) concluded that minimum torque change is not significantly different from minimum jerk unless the joint viscosity is incorporated into the dynamic model. They also concluded that the differences are slight and it requires the performance of many motions in order to recognise the differences. The model used in this research lacked any model of the elastic or viscous qualities of the joints and was frictionless. Optimisation including joint viscosity is extremely difficult and it is doubtful whether any meaningful minimum torque change results could be obtained with a full humanoid linkage incorporating joint elasticity and viscosity. The lack of joint viscosities is nevertheless a possible criticism of the MT and MTC models but it is reasonable to assume that the possible benefit is minimal besides being extremely impractical, if not impossible to implement.

Regarding the equilibrium point λ model (EPH), it was explicitly assumed that force and torque are equivalent. The equilibrium model (Section 2.7.13) is stated in terms of the forces generated by the muscles supporting a joint. The equations of the model yield the muscle forces. In order to implement the model it was applied not to muscle force but to joint torque. The λ control parameters were calculated with this in mind but the space of muscle force and joint torque may yield differences in the resultant motion. It was decided to operate in torque space since the control of the humanoid is significantly more straightforward. Torques apply directly to joints but muscles yield forces dependent not only on the contractile component of the muscle but also on the angle of the joint. This is significant, and has even been explicitly modelled since the affect is not trivial (see Section 2.6.1). The results for the

EPH model may thus have been different if the EPH forces had been applied to muscles where the resultant torques were calculated with attention to the joint angles. However, this necessitates a muscle model where the relative strengths and connection kinematics of the muscles are explicitly modelled. Building such a model is a highly complex and time consuming business. The choice to use joint torques was therefore taken because it was infeasible to build a muscle model. Besides the time required, the validation of the connection kinematics and muscle properties also poses an awkward research task. Moreover, the results indicated that the EPH model yielded very bad motion because of instability. Since this was caused by the absence of a velocity damping term in the torque generation function the problem would be equally apparent in muscle modelled EPH motion.

Finally, it was concluded that forward dynamic simulation is a significantly beneficial component in the generation of perceptually realistic human motion. This was due to the replication of trajectory hooks in fast motions. As a proviso, it should be mentioned that these features are highly sensitive to the spring-damper parameters. The spring-damper parameters used in this research were obtained from the humanoid robot specification. A suitable stable set of parameters had thus been developed already. It may require some experimental iterations to select such suitable parameters. In planning a human motion replication system from scratch however, the implementation of the forward dynamics functions themselves is a considerably more time consuming task than selecting suitable spring-damper parameters.

6.2.3 Dynamic optimisation

In order to optimise the total torque cost of a motion with respect to joint angles by a gradient based method it is necessary to evaluate the partial derivatives of the total torque cost with respect to each joint angle. When an analytical formulation of these partials for a given posture is available, the partials for the whole motion may be calculated by summing the partials calculated at a number of sampled postures. This is not however guaranteed

to be much faster than a numerical evaluation that involves numerically integrating torque costs calculated using inverse dynamics since a similar sampling density may be required in both cases.

In this work the Simplex method was used. This avoids the use of derivatives but a derivative based method might yield an improvement in efficiency if the derivatives could be calculated quickly. However, it was estimated that the time improvement yielded by an analytical derivative formulation would not justify its implementation. The complexity of the equations involved is acute. Forward and inverse kinematic and dynamic formulations were generated automatically using Mathematica and patched into the robot control code by hand[3]. These equations required an order of approximately one thousand lines each. Automatically generating the derivatives was thought likely to produce an even larger set of equations. Furthermore, specifying the equations in Mathematica and subsequently patching them into the motion production environment was thought to be an extremely time consuming task. It was thought unlikely that implementing analytical derivatives would yield a significant improvement in the time complexity

Some authors have quoted impressive results in this area. Cohen (1992) was able to solve dynamic minimisations for articulated figures. The figures had only two or three articulations however, and individual iterations required up to 20 seconds in the examples presented. Rose et al. (1996) solved torque minimisation for a 44 DOF figure performing a 0.4 second motion in 77 seconds on a $100MHz$ Pentium processor. However, their results were obtained for motion transitions where the initial and final posture were specified and the DOFs were constrained to vary monotonically between two postures. Their use of analytically formulated partial derivatives for the torque cost in a given dynamic posture allowed a relatively fast evaluation of the cost function, although it was still required to numerically integrate these derivatives to obtain the cost of the whole motion. They used the BFGS optimisation

[3]The kinematic and dynamic formulations are part of the robot control and simulation environment prepared by Stefan Schaal.

algorithm which is widely regarded as among the most efficient for problems of this nature. The results were however limited to short durations and transitions between two specified motions.

In Section 4.3.2 a two stage joint angle optimisation strategy was described. This was necessary since the dimensionality of the optimisation space had a quadratic effect on the convergence time. Rather than optimise all the joints simultaneously, the first phase optimised all the active joints (the back, right arm and wrist), and the second phase subsequently optimised the remaining passive joints that did not contribute directly to the motion. This invalidates the absolute optimality of the search since complex interactions between the active and passive joints could reduce the total torque or torque change. However, it was found that very little motion was observed in the passive joints except to counter-balance the active joints' motion. Likewise the cost contribution from the passive joints was very low, suggesting that they had a very slight relationship with the overall dynamic cost. This assumption is supported by the observation that if the passive limb made any complex motions it would require additional torque or torque change to make the motion but the inertial contribution of the limb to the rest of the body would be limited by the torque required to move the limb itself, *i.e.* the limb makes the best contribution to the total dynamic cost by acting in a relaxed manner simply counter-balancing the motion of the active joints. Ideally this would be demonstrated by including all the joints in a single optimisation pass, but computational efficiency requirements meant that partitioning was necessary.

6.3 Achievements

The work accomplished and described in this thesis includes significant theoretical, empirical and material developments. The research was initially motivated by the need to generate realistic human motion for computer animation. Motion capture is a time consuming and awkward process compa-

rable with conventional filming, involving a similar lengthy post-processing phase. Little effort has been made to scientifically validate alternative motion generation strategies while the growing requirements of humanoid animation and the expanding field of humanoid robotics require efficient, well founded synthetic motion generation algorithms.

This thesis has investigated synthetic motion generation with a robust foundation in human motion production research. As such, the research demonstrates a significant theoretical advance since thus far little animation research has paid attention to the wealth of human motion production research, both empirical and theoretical.

Research into human motion was therefore included in this thesis' literature survey. A large number of *features* typically embodied by human motion were identified and discussed along with their empirical bases. The properties of the musculo-skeletal system define the physical dynamics of the body and are thus highly significant to any accurate discussion of human motion. Muscle characteristics and muscle modelling research was therefore discussed. Having established this technical background, a large number of human motion production theories were surveyed. These theories formed the basis of fourteen synthetic motion production models.

Since the research is relevant to both computer animation and robots the area of synthetic humans was considered generally. Existing dynamics based humanoids were surveyed, and the many computational issues involved in developing such a software system were discussed. Since dynamic simulation is fundamental to such systems the area was also surveyed and significant areas were summarised and discussed. Humanoid robotics was also discussed and a brief survey of existing humanoid robots was presented. The control processes of such systems were discussed and the humanoid robot used in this research was described.

Having established a firm basis, twelve synthetic motion production models were planned by direct recourse to a selection of existing motion production theories. A further two models were developed by considering compu-

tational efficiency as well as motion production theory. Effort was made to select a broad range of models from the human motion production domain in order to ensure a broad range of possible conclusions. The synthetic methods were also described in a rigid and formal way. The formal mathematical structures thus developed are a significant contribution to the theoretical aspect of motion production since they not only define the models rigidly but facilitate robust mathematical reasoning. For example, the direct relationship between a large set of minimised derivative trajectory models was made apparent.

The synthetic models were implemented in software but it was first necessary to analyse their computational requirements. Having done so, and with reference to the necessary computational structures, a motion production environment was planned. The motion production environment facilitated motion generation, analysis and production in both computer animated and robotic systems. The organisation of this environment and the demonstration of its effectiveness are thus significant research contributions. The software itself included over $16\frac{1}{2}$ thousand lines of code and is also a significant contribution to robotics and animation. The largest components were the `mctool`, `mdtool` and `mltool` applications providing motion generation, visual analysis & profiling, and editing functions respectively. Other software components included general motion logging functions, numerical computation and robot control code. While most of these functions are composed according to existing algorithmic techniques, the research into dynamics based motion optimisation, and N^{th} derivative minimisation are significant contributions to computer science.

Psychovisual testing was performed using synthetically produced motions, as well as motion recorded from a human using the Optotrak motion capture system. Some specific research was carried out to justifiably improve the quality of the motion capture data prior to testing. Four psychovisual tests were performed in a total of 40 experiment runs. This required about 30 subjects sessions of around an hour each since some subjects were willing

to complete more than one experiment at a sitting. In addition to the psychovisual tests a broad range of computational statistics were also gathered investigating the efficiency and accuracy of the algorithmic methods. The gathered empirical data were analysed using traditional methods although in some cases analysis software was created such as the dimensional reduction analysis applied to perceived similarity of motion models. The results were discussed in detail and conclusions of relevance to robotics, human motion production and computer animation were drawn. These have also prompted a number of directions for future research, which are discussed in the next section.

In summary, this research has contributed to computer animation, robotics, human motion production and psychophysics. Motion production models have been identified and rigidly formalised. A broad and powerful software environment has been blue printed and implemented. New algorithmic techniques have been developed. Empirical psychovisual data has been gathered. Significant conclusions have been drawn.

On the philosophical side, this research has proven by demonstration that robust specifications of humanoid motion can be analysed with respect to their naturalness, or by recourse to salient motion features. The decoupling of motion generation and performance allows the same techniques to be applied across platforms and in different areas such as computer animation and robotics. It has been shown that it is unnecessary to delve into the depth of complex muscle modelling techniques, nor implement demanding dynamic optimisations in order to generate synthetic motion that is perceived as well as genuine human motion. It has also been revealed that the similarity of synthetic motion to human motion may not be equated with apparent naturalness. It has thus been shown that human motion perception and planning strategies are not equivalent, and further, that simple algorithmic techniques are capable of generating motion that can appear at least as natural as human motion and that these techniques may in some cases be implemented on-line.

6.4 Further research

There are a broad range of possible extensions to the research accomplished thus far. A number of directions are discussed below, beginning with some specific techniques for improving the methods already implemented. The discussion then expands to include related areas in which new developments are possible. Techniques for improving the existing methods focus on numerical optimisation and psychovisual testing.

Improving the optimisation methods would facilitate more robust and convincing results for the MT and MTC models and allow the investigation of other possibly more complex dynamic optimisations. The simplest method of facilitating more iterations is to use a faster computer. This is not unreasonable since super computers are now available in many locations and might reduce computation times by several orders of magnitude. Optimisations requiring such high end computation power would not be suitable for computer animators working with the current generation of microprocessors but experimental results could nevertheless be obtained. Indeed, it might be proven that such complex methods were not in fact necessary.

An alternative to increasing the computation power is to improve the efficiency of the numerical techniques. Analytical methods might be developed to speed up the evaluation of cost function integrals. While such analytical methods would be specific to the exact cost metric in use, it might be possible to use or develop a symbolic mathematical integration system suitable for a range of cost metrics. Reducing the cost evaluation time benefits a broad range of numerical optimisation methods. The Simplex method for example would improve in speed proportionally, and it might become feasible to perform multiple restarts in different areas of the search space thus strengthening the claim to global optimality.

A more sophisticated technique is multi-resolution optimisation. Multi-resolution optimisation aims to solve a problem by dealing with a hierarchy of sub-problems with a range of detail from coarse to fine. For example, in order to optimise the space of spline trajectories defined by say 32 knots, an

initial solution might be found for the space of splines defined by 4 knots. The optimal 4 knot trajectory might then be used to initialise a search over the space of splines defined by 8 knots, and then again to 16 and eventually 32 knots. This method encourages global optimality since the initial pass finds a generally optimal solution without regard to fine details which may be taken care of by more accurate layers.

An alternative to a hierarchical search that also exploits a change of search space is spline re-parameterisation. When optimising the knots of a spline according to a cost function integrated over the whole trajectory the spline could be re-parameterised to place knots directly in the regions of highest cost. This allows changes in knot positions to have a maximal effect on the overall cost while maintaining a small number of knots. Some fixed knots may be necessary to maintain the spline trajectory following re-parameterisation, but since they are not included in the optimisation process they have little effect on the total computation time.

The use of a carefully chosen orthogonal basis could help to encourage convergence. In straightforward gradient descent each control parameter is optimised individually, *e.g.* a set of knots in joint angle space defining a trajectory are optimised by perturbing each joint angle at each knot individually, calculating the change in cost while all the other joint angles are held constant, and assembling a partial derivative for the cost function. However, this can result in slow optimisation since joint angles must often change in a coordinated manner so that for example, the shoulder and elbow may rotate in opposite directions maintaining the same hand position but reducing the torque needed to hold the posture. A slight change in either angle without the corresponding change in the other tends to increase the cost and so the change occurs in very slight alternating steps in each joint angle. By partitioning the joint angles carefully, such mutual relationships can be encapsulated in the control parameters, *i.e.*, instead of using two parameters to control the two angles separately one parameter is connected to both joints yielding the appropriate mutual change in angles, and the second parameter

controls only one of the two joint angles so that the full range of possible configurations is still possible. The coordinated change of joint angles may therefore be obtained using adjustments biased towards single control parameters and therefore requires less oscillation over alternative iterations, and consequently fewer iterations. The key factor is that the steepest gradients of the solution space are aligned with the parameter space axes and are therefore more rapidly identified by numerically calculated partial derivatives.

Finally, abandoning local optimisation altogether, global optimisation strategies may be applicable. Global strategies such as genetic algorithms and simulated annealing could be used to find more convincingly robust global solutions and are ideally suited to high power parallel computers.

With regard to psychovisual testing, the experiments could be improved and expanded in a number of ways. In Section 6.2.1 it was mentioned that humans tend to fixate on the target point of a hand motion at the onset of motion. This could easily be incorporated into the motion specifications to improve the potential realism and strength of experimental conclusions. It was also mentioned that the realism of the computer generated humanoid might have a significant affect on the perceived naturalness and so the motions already developed could be used to render animations with various CG humanoids. For example, performing the experiments with high detail texture mapped humanoids, low detail polygonal models, lines, and even point lights attached to limb segments, the importance of the rendering method on the perceived naturalness could be assessed. Additionally, a real human actor could be recorded performing the motions. Given sufficient training an actor might be able to mimic the synthetic motion models and yield further results quantifying the relative naturalness of the motions. By simultaneously motion capturing or using video analysis the motion could be analysed after the performance and motions that accord with the synthetic models could be selected. Such an actor might be trained using video or first person perspective virtual reality equipment allowing them to monitor their own

and the target motions simultaneously.

A variety of viewpoints including the first person perspective could also be rendered for use with subjects assessing the motions since the experiments were all performed with the humanoid in the same orientation relative to the viewer. The humanoid used for rendering also had a single consistent set of kinematic parameters (limb lengths *etc.*). Varying these parameters and the appearance of the model could yield results regarding both humans' ability to assess limb kinematics, and the relative importance of such features on the perceived quality of motions. With respect to humanoid robotics, subjects could be situated with the robot in order to compare the perceptual results of video and actual robot motion experiences.

Alternative synthetic motion models may also be developed and assessed. There are a range of equilibrium point models that may be compared and further dynamic optimisations may be designed, not to mention the broad range of human motion production hypotheses remaining, some of which were discussed in Section 2.7 and Appendix D.3.9. Existing models may be augmented to include the 'noise' about some idealised trajectory that is characteristic of specific performances of motion by huamns. Bayesian analysis might be applied to motion capture to determine the nature of the 'noise'. A wider set of motion specifications may be necessary to adequately compare such models. Such specifications should be designed to emphasise the cases for which the models are expected to perform well and the cases where the models yield particularly unique motions.

Beyond these improvements to the existing techniques and psychovisual experiments the work may be developed into a number of new areas. In the realm of motion generation for example, the problems of context, emotion and style may be tackled. These areas may be approached by recording appropriate motions and attempting to recognise and propose discriminating mathematical characteristics. Synthetic models may thus be constructed and ranked using psychovisual testing. 'Context' refers to the issue of motion generation according to the environment and requirements of the motion such

as optimal speed, accuracy, the effect of fatigue, *etc.* The context of catching for example may prioritise speed, accuracy or even stability depending on the mass of the object to be caught. Motions may be performed in a way that conveys an emotional expression so that a happy motion for example, can be light and bouncy whereas an angry motion might be tense and jerky. Some work has already opened these areas which could be further explored with a similar strategy to the investigation made in this thesis. General motion styles may also be investigated in a similar way, perhaps yielding formal mathematical models approximating properties such as grace and stability. These could be used to model dancing, martial arts, and athletic motions, *etc.*

Mathematical models of athletic motions, and contributing concepts such as stability, rigidity, *etc.* may be applied to athletic analysis. The work may thus be developed towards the analysis and optimisation of physical motions that can be applied to sports and ergonomics for the benefit of athletes and manufacturers respectively.

Synthetic models may also be used to reverse engineer recorded motions. Such a process could be of benefit to motion recognition by providing metrics and feature extractions by which to classify motions.

Finally, the work so far has considered the visual perception of motion. However, humans often communicate and interact socially using physical contact. This adds another significant aspect to the experience of motion -the sense of touch. Motions performed using different synthetic models may appear visually similar when performed by a humanoid robot but be dramatically different when a human subject is also able to detect the force characteristics of the limb by means of physical contact. This suggests a new range of experiments where humans physically interact with a humanoid robot, and may reveal entirely new results for the synthetic models investigated only visually so far. Cooperative physical interactions between humans and robots have only recently become safe to investigate academically and represent a new step in the evolution of humanoid robots. The problems and

possibilities opened up by physical interaction with robots include models of motion planning, force generation, as well as physical communication, body language, social cues, *etc.* Other work by the author has already investigated one mutually cooperative physical interaction with a humanoid robot, and provides a starting point for investigating the naturalness of synthetic motion according the physical experience of the motion [Hale and Pollick (2000a,b)].

Appendix A

Definitions

All the definitions used in the text to specify motion models formally are gathered in this Appendix. The definitions are numbered in the same manner as in the text, but are gathered into meaningful categories: type definitions, correspondence constraints, and motion models. The numbering system is therefore non-sequential in this Appendix.

A.1 Type definitions

Definition 3 (Knot specification)

$$(T, \{p_0, \ldots, p_n\}) \in \mathbb{K}_d \quad \Leftrightarrow \quad T \in \mathbb{R}^+, \quad n, d \in \mathbb{Z}^+$$
$$\forall i \in [0, n] \quad p_i \in \mathbb{R}^d$$

$\mathbb{K}_d$ *represents the set of valid knot specifications in* d *dimensions.* T *represents the duration of the trajectory intended to pass through the set of knots* $\{p_0, \ldots, p_n\}$ *in sequence.*

Definition 4 (Spline specification)

$$(T, s) \in \mathbb{S}_d \quad \Leftrightarrow \quad T \in \mathbb{R}, \quad d \in \mathbb{Z}^+, \quad s : \mathbb{R} \mapsto \mathbb{R}^d$$
$$\forall t \in [0, T] \quad \forall \epsilon \in \mathbb{R}^+ \quad \exists \delta > 0 \quad \text{s.t.}$$
$$\forall t' \in [0, T] \quad \text{s.t.} \quad |t - t'| < \delta, \quad |s(t) - s(t')| < \epsilon$$

$\mathbb{S}_d$ *represents the set of continuous functions defined over the interval* $[0, T]$ *that map into* d *dimensional space.*

Definition 5 (Motion specification)

$$(T, f, \{p_0, \ldots, p_n\}) \in \mathbb{M}_d \quad \Leftrightarrow \quad T, f \in \mathbb{R}^+, \quad d \in \mathbb{Z}^+, \quad n = \left\lfloor \frac{T}{f} \right\rfloor$$
$$\forall i \in [0, n] \quad p_i \in \mathbb{R}^d$$

$\mathbb{M}_d$ *represents the space of all valid motions in* d *dimensions.* T *and* f *represent the duration and sampling frequency respectively, and* $\{p_0, \ldots, p_n\}$ *represent the set of ordered samples.* d *should typically be equal to the number of DOFs, so that each sample is a posture.*

A.2 Correspondence constraints

Definition 6 (Spline-knot correspondence)

$$SKC(s, k) \quad \Leftrightarrow$$

$$\exists d, n \in \mathbb{Z}^+ \quad \text{s.t.} \quad s = (T, s') \in \mathbb{S}_d, \quad k = (T, \{p_0, \ldots, p_n\}) \in \mathbb{K}_d$$
$$\exists (t_0, \ldots, t_n) \in \mathbb{R}^{n+1} \quad \text{s.t.} \quad \forall i, j \in [0, n] \quad \text{s.t.} \quad i < j,$$
$$t_i < t_j \quad \wedge \quad s'(t_i) = p_i \quad \wedge \quad t_0 = 0 \quad \wedge \quad t_n = T$$

The spline s *correctly interpolates the knot set* k *in duration* T*.*

Definition 7 (Motion sampling correspondence)

$$MSC(s, m) \quad \Leftrightarrow$$

$$\exists d, n \in \mathbb{Z}^+ \quad \text{s.t.} \quad s = (T, s') \in \mathbb{S}_d, \quad m = (T, f, \{p_0, \ldots, p_n\}) \in \mathbb{M}_d$$
$$\forall i \in [0, n], \quad p_i = s'\left(\frac{i}{f}\right)$$

The motion m *is a valid sampling of the spline* s *at frequency* f*.*

Definition 8 (Optimal spline-knot correspondence)

$$OSKC(s,k,C) \quad \Leftrightarrow$$

$$\exists d \quad \text{s.t.} \quad s \in \mathbb{S}_d, \quad C : \mathbb{S}_d \mapsto \mathbb{R}$$
$$SKC(s,k) \quad \wedge \quad \forall s' \in \mathbb{S}_d \quad \text{s.t.} \quad SKC(s',k), \quad C(s) \leq C(s')$$

The d dimensional spline s correctly interpolates the knot set k in duration T, and has minimal cost according to the function C over the space of all such splines that correctly interpolate k.

Definition 9 (Kinematic spline correspondence)

$$KSC(s_3, s_{NDOFS}) \quad \Leftrightarrow$$

$$s_3 = (T, s_3') \in \mathbb{S}_3, \quad s_{NDOFS} = (T, s_{NDOFS}') \in \mathbb{S}_{NDOFS}$$
$$\forall t \in [0,T], \quad FK(s_{NDOFS}'(t), s_3'(t))$$

The postures specified in s_{NDOFS} yield the hand positions specified in s_3 at every point on the spline.

Definition 14 (Kinematic knot correspondence)

$$KKC(k_3, k_{NDOFS}) \quad \Leftrightarrow$$

$$k_3 = (T, \{p_0, ..., p_n\}) \in \mathbb{K}_3, \quad k_{NDOFS} = (T, \{p_0', ..., p_n'\}) \in \mathbb{K}_{NDOFS},$$
$$\forall i \in [0,n], \quad FK(p_i', p_i)$$

The postures specified in k_{NDOFS} yield the hand positions specified in k_3 at every knot.

Definition 20 (Minimum torque postures spline)

$$MTPS(k,s,d) \quad \Leftrightarrow$$

$$d \in \mathbb{R}, \quad k = (T, \{p_0, ..., p_n\}) \in \mathbb{K}_3, \quad s = (T, s') \in \mathbb{S}_{NDOFS}$$
$$\exists K_{NDOFS} \subset \mathbb{K}_{NDOFS}, \quad \exists k_{MTP} \in K_{NDOFS}, \quad \text{s.t.}$$

$$k_{NDOFS} = (T, \{p'_0, ..., p'_n\}) \in K_{NDOFS} \quad \Leftrightarrow \quad \forall i \in [0, n], \quad FK(p'_i, p_i)$$
$$\forall k' \in K_{NDOFS}, \quad C(k_{MTP}) \leq C(k') \quad \wedge$$
$$OSKC\left(s, k_{NDOFS}, \int_o^T \left(\frac{d^3 s'_3}{dt^3}\right)^2 dt\right)$$

$$\text{where} \quad C : K_{NDOFS} \mapsto \mathbb{R}^+, \quad C\left((T, \{q_0, ..., q_{(nd)}\}\right) =$$
$$\beta_0 \sum_{i=0}^{nd} \sum_{j=0}^{NDOFS-1} \tau_{ij}^2 \quad + \quad \beta_1 \sum_{i=1}^{nd} \sum_{j=0}^{NDOFS-1} (\tau_{ij} - \tau_{(i-1)j})^2, \quad \text{where}$$
$$\tau \in \mathbb{R}^{(nd+1)\times NDOFS} \quad \wedge \quad \forall i \in [0, nd], \quad FD(q_i, \overline{0}, \tau_i, \overline{0})$$

d is the number of knots that will be inserted into the optimal posture set for every knot in the specification k. k_{NDOFS} is the optimal posture set, and s is a minimum angular jerk spline through the optimal postures. C is the cost function for optimisation and incorporates both torque and torque change cost according to the scaling factors β_0 and β_1 respectively.

A.3 Motion models

Definition 10 (Minimum Cartesian velocity motion)

$$MV(k, m, f) \quad \Leftrightarrow$$

$$k = (T, \overline{p}) \in \mathbb{K}_3, \quad m = (T, f, \overline{p'}) \in \mathbb{M}_{NDOFS}$$
$$\exists s_3 = (T, s'_3) \in \mathbb{S}_3, \quad s_{NDOFS} \in \mathbb{S}_{NDOFS} \quad \text{s.t.}$$

$$OSKC\left(s_3, k, \int_o^T \left(\frac{ds'_3}{dt}\right)^2 dt\right) \quad \wedge$$
$$KSC(s_3, s_{NDOFS}) \quad \wedge \quad MSC(s_{NDOFS}, m)$$

Definition 11 (Minimum Cartesian acceleration motion)

$$MA(k,m,f) \quad \Leftrightarrow$$

$$k = (T, \overline{p}) \in \mathbb{K}_3, \quad m = (T, f, \overline{p'}) \in \mathbb{M}_{NDOFS}$$
$$\exists s_3 = (T, s_3') \in \mathbb{S}_3, \quad s_{NDOFS} \in \mathbb{S}_{NDOFS} \quad \text{s.t.}$$

$$OSKC\left(s_3, k, \int_o^T \left(\frac{d^2 s_3'}{dt^2}\right)^2 dt\right) \quad \wedge$$
$$KSC(s_3, s_{NDOFS}) \quad \wedge \quad MSC(s_{NDOFS}, m)$$

Definition 12 (Minimum Cartesian jerk motion)

$$MJ(k,m,f) \quad \Leftrightarrow$$

$$k = (T, \overline{p}) \in \mathbb{K}_3, \quad m = (T, f, \overline{p'}) \in \mathbb{M}_{NDOFS}$$
$$\exists s_3 = (T, s_3') \in \mathbb{S}_3, \quad s_{NDOFS} \in \mathbb{S}_{NDOFS} \quad \text{s.t.}$$

$$OSKC\left(s_3, k, \int_o^T \left(\frac{d^3 s_3'}{dt^3}\right)^2 dt\right) \quad \wedge$$
$$KSC(s_3, s_{NDOFS}) \quad \wedge \quad MSC(s_{NDOFS}, m)$$

Definition 13 (Minimum Cartesian snap motion)

$$MS(k,m,f) \quad \Leftrightarrow$$

$$k = (T, \overline{p}) \in \mathbb{K}_3, \quad m = (T, f, \overline{p'}) \in \mathbb{M}_{NDOFS}$$
$$\exists s_3 = (T, s_3') \in \mathbb{S}_3, \quad \exists s_{NDOFS} \in \mathbb{S}_{NDOFS} \quad \text{s.t.}$$

$$OSKC\left(s_3, k, \int_o^T \left(\frac{d^4 s_3'}{dt^4}\right)^2 dt\right) \quad \wedge$$
$$KSC(s_3, s_{NDOFS}) \quad \wedge \quad MSC(s_{NDOFS}, m)$$

Definition 15 (Minimum angular velocity motion)

$$MAV(k,m,f) \quad \Leftrightarrow$$

$$k=(T,\overline{p}) \in \mathbb{K}_3, \quad m=(T,f,\overline{p'}) \in \mathbb{M}_{NDOFS},$$
$$\exists k_{NDOFS} \in \mathbb{K}_3, \quad \exists s_{NDOFS}=(T,s'_{NDOFS}) \in \mathbb{S}_{NDOFS} \quad \text{s.t.}$$

$$KKC(k,k_{NDOFS}) \quad \wedge$$
$$OSKC\left(s_{NDOFS},k_{NDOFS},\int_o^T \left(\frac{ds'_{NDOFS}}{dt}\right)^2 dt\right) \quad \wedge$$
$$MSC(s_{NDOFS},m)$$

Definition 23 (Minimum angular acceleration motion)

$$MAA(k,m,f) \quad \Leftrightarrow$$

$$k=(T,\overline{p}) \in \mathbb{K}_3, \quad m=(T,f,\overline{p'}) \in \mathbb{M}_{NDOFS},$$
$$\exists k_{NDOFS} \in \mathbb{K}_3, \quad \exists s_{NDOFS}=(T,s'_{NDOFS}) \in \mathbb{S}_{NDOFS} \quad \text{s.t.}$$

$$KKC(k,k_{NDOFS}) \quad \wedge$$
$$OSKC\left(s_{NDOFS},k_{NDOFS},\int_o^T \left(\frac{d(s'_{NDOFS})^2}{dt^2}\right)^2 dt\right) \quad \wedge$$
$$MSC(s_{NDOFS},m)$$

Definition 24 (Minimum angular jerk motion)

$$MAJ(k,m,f) \quad \Leftrightarrow$$

$$k = (T, \overline{p}) \in \mathbb{K}_3, \quad m = (T, f, \overline{p'}) \in \mathbb{M}_{NDOFS},$$
$$\exists k_{NDOFS} \in \mathbb{K}_3, \quad \exists s_{NDOFS} = (T, s'_{NDOFS}) \in \mathbb{S}_{NDOFS} \quad \text{s.t.}$$

$$KKC(k, k_{NDOFS}) \quad \wedge$$
$$OSKC\left(s_{NDOFS}, k_{NDOFS}, \int_o^T \left(\frac{d(s'_{NDOFS})^3}{dt^3}\right)^2 dt\right) \quad \wedge$$
$$MSC(s_{NDOFS}, m)$$

Definition 25 (Minimum angular snap motion)

$$MAS(k, m, f) \quad \Leftrightarrow$$

$$k = (T, \overline{p}) \in \mathbb{K}_3, \quad m = (T, f, \overline{p'}) \in \mathbb{M}_{NDOFS},$$
$$\exists k_{NDOFS} \in \mathbb{K}_3, \quad \exists s_{NDOFS} = (T, s'_{NDOFS}) \in \mathbb{S}_{NDOFS} \quad \text{s.t.}$$

$$KKC(k, k_{NDOFS}) \quad \wedge$$
$$OSKC\left(s_{NDOFS}, k_{NDOFS}, \int_o^T \left(\frac{d(s'_{NDOFS})^4}{dt^4}\right)^2 dt\right) \quad \wedge$$
$$MSC(s_{NDOFS}, m)$$

Definition 16 (Minimum jerk virtual trajectory motion)

$$MJVT(k, m, f) \quad \Leftrightarrow$$

$$k = (T, \overline{p}) \in \mathbb{K}_3, \quad m = (T, f, \overline{p'}) \in \mathbb{M}_{NDOFS}$$
$$\exists s_3 = (T, s'_3) \in \mathbb{S}_3, \quad \exists s_{VT} = (T, s'_{VT}) \in \mathbb{S}_{NDOFS},$$
$$\exists s_{NDOFS} = (T, s'_{NDOFS}) \in \mathbb{S}_{NDOFS}, \quad \exists \tau : \mathbb{R} \mapsto \mathbb{R}^{NDOFS}, \quad \text{s.t.}$$

$$OSKC\left(s_3, k, \int_o^T \left(\frac{d^3 s'_3}{dt^3}\right)^2 dt\right) \quad \wedge \quad KSC(s_3, s_{NDOFS}) \quad \wedge$$

$$\forall t \in [0,T], \quad \forall i \in [0, NDOFS-1],$$
$$\tau_i(t) = \beta_0(s'_{VTi}(t) - s'_{NDOFSi}(t)) - \beta_1 s'_{NDOFSi}(t) \quad \wedge$$
$$FD\left(s'_{NDOFS}(t), \frac{ds'_{NDOFS}}{dt}(t), \tau(t), \frac{d(s'_{NDOFS})^2}{dt^2}(t)\right) \quad \wedge$$
$$MSC(s_{NDOFS}, m)$$

The virtual trajectory s_{VT} is specified in the same way as the minimum Cartesian jerk trajectory. The dynamic constraint formulated with FD asserts that S_{NDOFS} is a dynamically consistent trajectory where the internal torques τ are defined by spring-damper systems between the actual angles and the angles in the virtual trajectory. The parameters of the spring-dampers are β_0 and β_1.

Definition 17 (Equilibrium point λ motion)

$$EPH(k, m, f, \overline{t}) \quad \Leftrightarrow$$

$$\overline{t} = (t_0, \ldots, t_n) \in \mathbb{R}^{n+1}, \quad k = (T, \overline{p}) \in \mathbb{K}_{NDOFS},$$
$$k_{NDOFS} = (T, k'_{NDOFS}) \in \mathbb{K}_{NDOFS},$$
$$s_{NDOFS} = (T, s'_{NDOFS}) \in \mathbb{S}_{NDOFS}$$
$$\exists \lambda = (\lambda_0, \ldots, \lambda_n) \in \mathbb{R}^{(n+1)\times NDOFS},$$
$$\exists \tau, \tau' = (\tau'_0, \ldots, \tau'_n) \in \mathbb{R}^{(n+1)\times NDOFS},$$
$$\exists \Lambda : \mathbb{R} \mapsto \mathbb{R}^{NDOFS} \quad \text{s.t.}$$

$$KKC(k, k_{NDOFS}) \quad \wedge \quad \forall i \in [0,n], \quad FD(k'_{NDOFSi}, \overline{0}, \tau'_i, \overline{0}),$$
$$\forall j \in [0, NDOFS-1], \quad \lambda_{ij} = k'_{NDOFSij} - \frac{1}{\beta_0} ln\left(\frac{\tau'_{ij}}{\beta_1} + 1\right),$$
$$\forall t \in [0,T], \quad \exists i \quad \text{s.t.} \quad t_i \le t \le t_{i+1} \quad \wedge$$
$$\Lambda(t) = \lambda_i \left(\frac{t_{i+1} - t}{t_{i+1} - t_i}\right) + \lambda_{i+1}\left(\frac{t - t_i}{t_{i+1} - t_i}\right) \quad \wedge$$
$$\forall t \in [0,T], \quad FD\left(s'_{NDOFS}(t), \frac{ds'_{NDOFS}}{dt}(t), \tau(t), \frac{d(s'_{NDOFS})^2}{dt^2}(t)\right)$$
$$\wedge \quad \tau(t) = \beta_0 e^{\beta_1(s'(t) - \Lambda(t))} \quad \wedge \quad MSC(s_{NDOFS}, m)$$

The postures corresponding to each Cartesian knot are represented by k_{NDOFS}. τ' represents the torques necessary to maintain those postures with zero velocity and acceleration. The set of λ parameters that precipitate these torques in a static configuration are represented by λ itself, and Λ gives their linear interpolation over the duration of the whole motion. Finally s_{NDOFS} is a motion that is dynamically consistent when all the internal torques are generated according to the appropriate λ torque generation equation.

Definition 18 (Minimum torque change motion)

$$MTC(k,m,f) \quad \Leftrightarrow$$

$$k = (T, \overline{p}) \in \mathbb{K}_3, \quad m = (T, f, \overline{p'}) \in \mathbb{M}_{NDOFS},$$
$$\exists k_{NDOFS} \in \mathbb{K}_3, \quad \exists \tau = (\tau_0, \ldots, \tau_{NDOFS-1}) : \mathbb{R} \mapsto \mathbb{R}^{NDOFS},$$
$$\exists s_{NDOFS} = (T, s'_{NDOFS}) \in \mathbb{S}_{NDOFS} \quad \text{s.t.}$$

$$KKC(k, k_{NDOFS}) \quad \wedge \quad MSC(s_{NDOFS}, m) \quad \wedge$$
$$OSKC\left(s_{NDOFS}, k_{NDOFS}, \int_o^T \left[\sum_{i=0}^{NDOFS-1} \left(\frac{d\tau_i}{dt}\right)^2\right] dt \quad \text{where}\right.$$
$$\left.\forall t \in [0,T], \quad FD\left(s'_{NDOFS}(t), \frac{ds'_{NDOFS}}{dt}(t), \tau(t), \frac{d(s'_{NDOFS})^2}{dt^2}(t)\right)\right)$$

Definition 19 (Minimum torque motion)

$$MT(k,m,f) \quad \Leftrightarrow$$

$$k = (T, \overline{p}) \in \mathbb{K}_3, \quad m = (T, f, \overline{p'}) \in \mathbb{M}_{NDOFS},$$
$$\exists k_{NDOFS} \in \mathbb{K}_3, \quad \exists \tau = (\tau_0, \ldots, \tau_{NDOFS-1}) : \mathbb{R} \mapsto \mathbb{R}^{NDOFS},$$
$$\exists s_{NDOFS} = (T, s'_{NDOFS}) \in \mathbb{S}_{NDOFS} \quad \text{s.t.}$$

$$KKC(k, k_{NDOFS}) \quad \wedge \quad MSC(s_{NDOFS}, m) \quad \wedge$$

$$OSKC\left(s_{NDOFS}, k_{NDOFS}, \int_o^T \left[\sum_{i=0}^{NDOFS-1} \tau_i^2\right] dt \quad \text{where}\right.$$
$$\left.\forall t \in [0,T], \quad FD\left(s'_{NDOFS}(t), \frac{ds'_{NDOFS}}{dt}(t), \tau(t), \frac{d(s'_{NDOFS})^2}{dt^2}(t)\right)\right)$$

Definition 21 (Minimum torque postures motion)

$$MTP(k,m,f,d) \quad \Leftrightarrow$$

$$\exists s \in \mathbb{S}_{NDOFS} \quad \text{s.t.} \qquad MTSP(k,s,d) \quad \wedge \quad MSC(s,m,f)$$

Definition 22 (Minimum torque postures virtual trajectory motion)

$$MTPVT(k,m,f,d) \quad \Leftrightarrow$$

$$\exists s_{VT} = (T, s'_{VT}), s_{NDOFS} = (T, s'_{NDOFS}) \in \mathbb{S}_{NDOFS},$$
$$\exists \tau : \mathbb{R} \mapsto \mathbb{R}^{NDOFS}, \quad \text{s.t.}$$

$$MTSP(k, s_{VT}, d) \quad \wedge \quad \forall t \in [0,T], \quad \forall i \in [0, NDOFS-1]$$
$$\tau_i(t) = \beta_0(s'_{VTi}(t) - s'_{NDOFSi}(t)) - \beta_1 s'_{NDOFSi}(t) \quad \wedge$$
$$FD\left(s'_{NDOFS}(t), \frac{ds'_{NDOFS}}{dt}(t), \tau(t), \frac{d(s'_{NDOFS})^2}{dt^2}(t)\right)\Big) \quad \wedge$$
$$MSC(s,m,f)$$

s_{VT} is the virtual trajectory spline generated according to the minimum torque postures model. The constraint formulated using FD and τ states that S_{NDOFS} must be a dynamically consistent motion generated with internal torques calculated using the virtual trajectory as a target for a spring-damper system in each joint.

A.4 Cyclic splines

The formulation of motion definitions for cyclic motions analogous to those above was not presented in the main text. It is a theoretically straightforward extension but requires a re-statement of eleven of the definitions above. Rather than virtually duplicate each of the definitions however, the extension to cyclic motions is described below.

Firstly, cyclic definitions for the MJVT, EPH and MTPVT models may be disregarded. These models require forward dynamic simulation and it is both undesirable and computationally challenging to constrain motion to cyclic trajectories in dynamic space. The MJVT and MTPVT models may however, be guided by cyclic virtual trajectories.

For the remaining models the existing definitions may be converted to handle cyclic motions under the assumption that the knot sequence for a cyclic trajectory will duplicate the initial knot at the end of the knot sequence, *e.g.*, $k = (T, \{k_0, \ldots, k_n\})$. Recall that splines were defined as a type, and quantification over the domain of splines was used as a condition in the definition of each motion model. By constraining the elements of the spline type itself to be cyclic all resulting trajectories will be cyclic. It is necessary to explicitly incorporate the order of continuity at the cycle point. The following cyclic spline type is thus defined.

Definition 26 (Cyclic spline specification)

$$
\begin{aligned}
&(T, s) \in \mathbb{S}_{d,c} \quad \Leftrightarrow \\
&\quad T \in \mathbb{R}, \quad d \in \mathbb{Z}^+, \quad c \in \mathbb{N}, \quad s : \mathbb{R} \mapsto \mathbb{R}^d \\
&\quad \forall k \in [0, c], \quad \forall t \in [0, T] \quad \forall \epsilon \in \mathbb{R}^+ \quad \exists \delta > 0 \quad \text{s.t.} \\
&\quad \forall t' \in [0, T] \quad \text{s.t.} \quad |(t \mid T) - (t' \mid T)| < \delta, \\
&\quad \left| \frac{ds^k}{dt}(t \mid T) - \frac{ds^k}{dt}(t' \mid T) \right| < \epsilon
\end{aligned}
$$

$\mathbb{S}_d, k$ *represents the set of functions cyclically continuous over* $[0, T]$ *up to the* c^{th} *order. Note that* $(t \mid T)$ *is the value of* $c_0 T + t$ *that falls in the range* $[0, T]$ *for some* $c_0 \in \mathbb{Z}$.

The definitions for cyclic minimum velocity match those for the non-cyclic case except that s_3 and s_{NDOFS} are replaced by $s_{3,0} \in \mathbb{S}_{3,0}$ and $s_{NDOFS,0} \in \mathbb{S}_{NDOFS,0}$ respectively -likewise for acceleration, jerk and snap where the order of continuity required is 1, 2 and 3 respectively. The splines drawn from S_{NDOFS} for minimum torque, torque change, and minimum torque postures require 2^{nd} order continuity since they are interpolated by minimum jerk. In the latter case the optimisation constraint is formulated over the domain of knots, and so the final knot in k_{NDOFS} must also be constrained to equal the first knot.

Appendix B

BVH motion format

The following text contains the header of the BVH file format used to import motions into Poser. It is taken from a 4.5 second motion sampled at 30 Hertz. The motion data has been cut, but the link dimensions and topology are contained in the header.

```
HIERARCHY
ROOT Hip
{
  OFFSET 0.00  0.00  0.00
  CHANNELS 6 Xposition Yposition Zposition Xrotation Zrotation Yrotation
  JOINT Abdomen
  {
    OFFSET 0.000000 2.159 0.000000
    CHANNELS 3 Xrotation Zrotation Yrotation
    JOINT Chest
    {
      OFFSET 0.000000 0.152 0.000000
      CHANNELS 3 Xrotation Zrotation Yrotation
      JOINT Neck
      {
        OFFSET 0.000000 3.175 0.000000
        CHANNELS 3 Xrotation Zrotation Yrotation
        JOINT Head
        {
```

```
    OFFSET 0.000000 1.143 0.000000
    CHANNELS 3 Xrotation Zrotation Yrotation
    End Site
    {
      OFFSET 0.000000 2.1 0.000000
    }
  }
}
JOINT Left Collar
{
  OFFSET 1.21 1.87 0.000000
  CHANNELS 3 Xrotation Zrotation Yrotation
  JOINT Left Shoulder
  {
    OFFSET 0.85 0.131 0.000000
    CHANNELS 3 Xrotation Zrotation Yrotation
    JOINT Left Forearm
    {
      OFFSET 0.000000 -2.921 0.000000
      CHANNELS 3 Xrotation Zrotation Yrotation
      JOINT Left Hand
      {
        OFFSET 0.000000 -3.143 0.000000
        CHANNELS 3 Xrotation Zrotation Yrotation
        End Site
        {
          OFFSET 0.000000 -0.506 0.000000
        }
      }
    }
  }
}
JOINT Right Collar
{
  OFFSET -1.21 1.87 0.000000
  CHANNELS 3 Xrotation Zrotation Yrotation
  JOINT Right Shoulder
  {
```

```
        OFFSET -0.85 0.131 0.000000
        CHANNELS 3 Xrotation Zrotation Yrotation
        JOINT Right Forearm
        {
          OFFSET 0.000000 -2.921 0.000000
          CHANNELS 3 Xrotation Zrotation Yrotation
          JOINT Right Hand
          {
            OFFSET 0.000000 -3.143 0.000000
            CHANNELS 3 Xrotation Zrotation Yrotation
            End Site
            {
              OFFSET 0.000000 -0.506 0.000000
            }
          }
        }
      }
    }
  }
}
JOINT Left Thigh
{
  OFFSET 2.286 0.000000 0.000000
  CHANNELS 3 Xrotation Zrotation Yrotation
  JOINT Left Shin
  {
    OFFSET 0.000000 -4.826 0.000000
    CHANNELS 3 Xrotation Zrotation Yrotation
    JOINT Left Foot
    {
      OFFSET 0.000000 -3.937 0.000000
      CHANNELS 3 Xrotation Zrotation Yrotation
      End Site
      {
        OFFSET 0.000000 0.000000 2.2
      }
    }
  }
```

```
   }
   JOINT Right Thigh
   {
     OFFSET -2.286 0.000000 0.000000
     CHANNELS 3 Xrotation Zrotation Yrotation
     JOINT Right Shin
     {
       OFFSET 0.000000 -4.826 0.000000
       CHANNELS 3 Xrotation Zrotation Yrotation
       JOINT Right Foot
       {
         OFFSET 0.000000 -3.937 0.000000
         CHANNELS 3 Xrotation Zrotation Yrotation
         End Site
         {
           OFFSET OFFSET 0.000000 0.000000 2.2
         }
       }
     }
   }
 }
 MOTION
 Frames: 135
 Frame Time: 0.033333
0.0 12.5 0.0 0.0 0.0 0.0 0.0 0.0 -0.13789 5.71645 -0.00922 ...
...
...
```

Appendix C

The ATR research environment

The research described in this thesis was conducted at Glasgow University and an international research company, ATR. The practical research, programming, motion capture, motion rendering and robotic performance, *etc.*, was mostly conducted at ATR.

ATR is a partially government funded research institute in Japan. It was founded in 1985 and the first papers were published in 1986. It currently employs 277 researchers and annually publishes around 500-700 papers nationally and 350-450 internationally. About 120 patents are also registered every year.

There are six departments: spoken language translation, adaptive communications research, human information science, media information science, the technology liaison centre, and the brain imaging centre. This research was conducted as part of the CyberHuman project in the human information science department, headed by Professor Mitsuo Kawato. Other research currently active in this group is briefly discussed in Section 2.3.1. Further information can be found on the website at www.atr.co.jp.

Three visits were made to ATR, in 1999, 2000 and 2001-2002. These visits lasted for three months, three months and five months respectively. The visits were financially supported by the EPSRC and, in 2000, by the British Council under the REES summer research scheme.

There were certain practical issues concerning performing doctoral research at ATR. Since time was limited financially, and by visa specifications, *etc.*, it was necessary to formulate a robust research plan prior to arrival in Japan and work to this schedule. While some researchers working under permanent contract at ATR were available for discussion via email, it was in general necessary to complete all the research with notes and digital records during each visit. Resources such as the humanoid robot and motion capture suite for example, were only available on site in Japan.

It was a required policy that all visiting researchers make a formal presentation of their findings at some point during a stay at ATR and present an internal report at the conclusion of each visit. Besides these, three papers were presented at international conferences [Hale and Pollick (2000a,b, 2002)]. Funding was received for a trip to the Humanoids 2000 conference in Pittsburgh that occurred during a visit to ATR.

ATR is an international research institute with researchers from all over the world. The language of research was primarily English and presentations were made in English. During a public opening of ATR called "Open House" however, poster presentations were made in Japanese. While English speaking non-natives were always present at ATR, technical support staff often spoke only Japanese and it was necessary to communicate in Japanese. Likewise, it necessary to gain a familiarity with corporate customs and develop a cultural awareness throughout the time spent at ATR.

Appendix D

Extended discussions

D.1 Dynamic simulation

This section describes various significant computational issues of forward dynamic simulation. The following topics are included: forward dynamics formulation, numerical integration, joint configuration, collision detection, collision resolution, contact detection and resolution, joint constraints, joint limits, closed loops and friction.

D.1.1 Forward dynamics formulations

The mathematical tools of FD were introduced to computer animation by Armstrong and Green (1985), and Wilhelms and Barsky (1985). Armstrong and Green (1985) subsequently presented a formulation of the equations tailored to suit articulated figures and based on the work of Goldstein, yielding the so called 'Armstrong recursive formulation'. This recursive evaluation strategy applies to a tree form articulated structure and requires time linearly proportional to the number of DOFs. The Armstrong method has become a definitive reference for dynamics based animation research.

All the dynamics formulations are variations of Newton's second law $F = ma$. Going beyond the relatively simple dynamics of separate rigid bodies [Barzel and Barr (1988)], the primary contribution of research in ar-

ticulated dynamics has been the development of recursive schemes or matrix methods facilitating fast evaluation [Girard (1991)]. Recursive schemes usually require time linear in the number of joints but are not well behaved numerically and are harder to manipulate mathematically [Green (1991)]. Matrix schemes can facilitate simultaneous solution of motion and force specification effectively combining FD and ID. Among the recursive strategies are recursive D'Alembert method [Park et al. (1992)], recursive Newton-Euler method [Ko and Badler (1996), van de Panne and Fiume (1993)] and its formulation using Featherstone's succinct spatial algebra [Lathrop (1986)]. Matrix formulations include the Gibbs-Appell formulation [Wilhelms and Barsky (1985), Wilhelms (1987)] which partitions force, torque, mass and velocity into a neat equation but is $O(n^4)$, the Euler matrix formulation which is faster due to the sparse nature of the matrix [Wilhelms (1991)], the Lagrange[1] matrix method [Witkin and Kass (1988), Grzeszczuk and Terzopoulos (1995), Baraff (1996)] based on the Lagrangian dynamics formulation solved using Langrangian multipliers and the D'Alembert matrix formulation based on Gaussian elimination [Isaacs and Cohen (1987)].

An overview of dynamics formulations for humanoid articulated figures may be found in [Wilhelms (1991)].

D.1.2 Numerical integration

The easiest numerical integration method is the intuitive technique of multiplying the current velocity/acceleration by a time step dt, adding the result to the current position/velocity and moving the time index forward by dt. This is in fact the 'simple Euler method' and can be derived from a Taylor series expansion of the differential equations [Witkin and Baraff (1995)]. It is applicable because while the dynamics equations are a second order problem (acceleration is the second derivative of position) it is separable and can be reformulated as two parallel first order problems in terms of velocity and

[1]The Lagrangian dynamics formulation is equivalent to Newton's $F = ma$ formulation but the equations are expressed in terms of momentum -the product of mass and velocity.

acceleration respectively [Witkin (1995b)].

The simple Euler method has been used for forward dynamics [Kunii and Sun (1990), Kass (1995a)] but because the equations are stiff[2] and the solution space is highly non-linear, more advanced schemes such as the implicit Euler method have been used [Kass (1995b)]. In fact many complex schemes have been applied such as the 4th order Runge-Kutta method [Wilhelms and Forsey (1988)], predictor-corrector Runge-Kutta method with adaptive step-sizing [Isaacs and Cohen (1987, 1988)], 4th order Runge-Kutta starter method with an implicit 4th order Adams-Moulton method [Kearney et al. (1993)], adaptive Sarafyan embedded form [Wilhelms et al. (1988)], adaptive Adams predictor method [Barzel and Barr (1988)], partitioning of variables into stiff and non-stiff sets [Armstrong and Green (1985)], and over-sampling [Wilhelms (1987)].

A comprehensive discussion of numerical integration schemes for FD may be found in [Green (1991)].

D.1.3 Joint configuration

The joint angles of an articulated figure must be represented. The simplest approach is to use Euler angles which specify the rotation in the x, y and z axes. Unfortunately this leads to singularities and indeterminacies when joints become axis aligned [Wilhelms et al. (1988)]. Alternatives include the Denevit-Hartenberg screw formulation (originating in robotics literature) that represents joint configurations as a couple composed of joint twist and axial translation [Maciejewski and Klein (1985), Vasilonikolidakis and Clapworthy (1991), Cohen (1992)], or quaternions that represent orientation in 3D using four values [Lake and Green (1991), Baraff (1995), Lamouret and Gascuel (1996), Burbanks (1996)]. The fourth value of the quaternion is effectively a scaling parameter that should be unity, but is involved in quaternion composition operations. This feature facilitates simple normalisation of a

[2]Stiff systems require small learning parameters because of localised areas where gradients are very large.

quaternion so that numerical drift can be prevented. By comparison, correcting the 3×3 orthogonal matrices representing an Euler angle rotation requires re-orthogonalisation -a calculation with no uniquely defined solution.

D.1.4 Collision detection

For polyhedral objects naïve collision detection involves testing every face of every polyhedron for intersection. These intersection tests may be carried out efficiently if all the faces are triangles [Aftomsis (1996)] or using Plücker coordinates [Yamaguchi and Niizeki (1997)] but it is also necessary to reduce the number of tests for the sake of efficiency. Typically, a hierarchy of tests of increasing durations is used to discard pairs of polyhedra that cannot intersect [Baraff (1995)], *e.g.*, planar tests separating all the vertices of two polyhedra, axis aligned bounding boxes, and octrees for closest point determination [Vemuri et al. (1998)]. A particularly efficient method is based on Voronoi regions (3D space partitioned according to the closest *feature* of a polyhedron -vertex, edge, or face) and utilises the expected coherence of polyhedra location over subsequent time frames [Mirtich (1998)]. Various other strategies have also been proposed for specific conditions such as the analytical event based method for large numbers of polyhedra [Dworkin and Zeltzer (1993)].

D.1.5 Collision resolution

It is possible to calculate exactly the resulting velocity following an instantaneous collision between two particles [Witkin (1995b)], or two polyhedra [Dworkin and Zeltzer (1993), Baraff (1995)] whether the collision is elastic or inelastic. However, in general elastic collisions do not occur instantaneously and involve some deformation of the colliding bodies. While analytical solutions are faster to solve, spring-damper systems have been used to model non-instantaneous collisions [Wilhelms et al. (1988)] but yield stiff numerical systems due to the high spring coefficients necessary.

Collision resolution is discussed further by Mirtich (1998).

D.1.6 Contact logging & contact resolution

When two bodies come into non-instantaneous contact, force is transmitted between the bodies. For polyhedral bodies it is possible to log contacts in terms of vertex/face contacts and edge/edge contacts. For convex polyhedra these 'support points' may be identified and logged relatively straightforwardly [Dworkin and Zeltzer (1993)]. In order to calculate the appropriate forces and ensure that polyhedra do not inter-penetrate it is necessary to solve a quadratic program [Baraff (1995)]. This problem has been solved using iterated solutions of linear programs [Baraff (1994)], Lagrangian multipliers [Vasilonikolidakis and Clapworthy (1991), Baraff (1996)], spring-damper systems [Wilhelms (1987)], and Baumgarte stabilisation [Hodgins and Pollard (1997)].

The resolution of contact forces for rigid bodies is discussed by Mirtich (1998).

D.1.7 Joint constraints

To ensure that the links of an articulated figure remain connected at the joints it is necessary to constrain the links. These constraints cannot be enforced kinematically and must be satisfied by forces to preserve the laws of dynamics. The necessary forces have been calculated by Newton-Raphson iteration [van Overveld (1994)], single value decomposition of linear systems [Barzel and Barr (1988)], Lagrangian multipliers solved by non-iterative matrix sparse matrices [Baraff (1996)] (which also facilitate non-holonomic, *i.e.*, velocity constraints), solving second order differential systems [Kearney et al. (1993)] (which allow for runtime adjustment of constraints if symbolic algebraic evaluation is incorporated), using Jacobian matrices to calculate constraint forces and damped feedback terms to smother inaccuracies [Witkin (1995a)], and by evaluating the Jacobian in a disjoint fashion to minimise

the size of equations generated symbolically [Witkin et al. (1995)]. Generalised optimisation by means of gradient descent has also been used to solve constraints for hanging rods [van Overveld and van Loon (1992)].

D.1.8 Joint limits

The range of an articulation in a humanoid figure may be less than its geometrical range, *e.g.*, the elbow joint does not bend backwards. Joint limits may be enforced by temporarily inserting springs generating torques when joints approach their limits, by modelling limits as collisions, or by incorporating the range-limited joints into constraint resolution [Wilhelms et al. (1988)]. Lagrangian multipliers have been used to impose joint limits [Baraff (1996)] but rigid limits are undesirable since they don't mimic the natural springy motion of human limbs [Wilhelms (1987)]. Elastic joint limits [McKenna and Zeltzer (1996)], spring-damper systems [Wilhelms (1991)] and damping of angular velocity [Wilhelms et al. (1988)] have all been implemented.

D.1.9 Closed loops

The closed loop force transmission problem is a particularly awkward and time consuming issue in forward dynamics. It occurs when a chain of supportive contacts becomes cyclic, such as a few blocks forming an arch supported by a plane, or a humanoid clapping or even just standing. Full solutions are typically time consuming, and include a formulation using Featherstone's spatial algebra [Lathrop (1986)] and a Jacobian method based on Lagrangian multipliers [Yamane and Nakamura (2000)]. Some authors have modelled closed loops using constraints solved by Lagrangian multipliers [Baraff (1996), Isaacs and Cohen (1988)], and some partial solutions exist. An approximation of the contact force based on the mass distribution over the feet while standing for example, has been used to resolve the closed loop between the floor and legs of a humanoid [Ko and Badler (1996)], and an open-loop method has been used to resolve closed loops by reducing the

number of degrees of freedom in a closed loop by hand [Park et al. (1992)].

D.1.10 Friction

Sliding bodies usually come to rest as a result of friction forces between the bodies. The basics of friction with respect to particle systems are discussed by Witkin (1995b) and the more complex issues of static friction (resisting acceleration from zero velocity) and dynamic friction (acting against a non-zero velocity) are discussed by Baraff (1994). Friction has been modelled by applying a force proportional to the area of contact between polyhedral bodies [Dworkin and Zeltzer (1993)], and for water submerged bodies, by scaling the dot product of link velocity with surface normal multiplied by surface area [van de Panne and Fiume (1993)].

D.2 Characteristics of muscles

The following sections discuss the neural pathways by which human muscles are controlled, the process by which muscle fibres contract in order to generate a force and factors affecting their efficiency.

D.2.1 Muscle stimulation & spinal reflexes

The brain stimulates muscle fibres by means of α motor neurones that fire impulses down the spinal cord and from there into the muscle tissue itself. Individual muscle fibres are incapable of regulating their force response, being either extended or contracted. By increasing the number of α motor neurones firing and their firing frequency a larger number of muscle fibres may be stimulated and contracted so that the force generation of the muscle may be increased.

Below the brain-muscle control path are the spinal reflexes. There are three reflexes: the myotatic, the inverse myotatic, and the flexion reflex. These are also known as the stretch reflex, the tension reflex and the with-

drawal reflex respectively. Spinal reflexes involve a pathway from sensory neurones to motor neurones via connections in the spinal cord itself. Since they do not require the propagation of a signal to the brain and back they are faster than the higher level control processes.

The withdrawal reflex activates joint flexors in response to pain or noxious stimuli. The signal loop begins with a sensory neurone that stimulates an interneurone in the spinal cord. The interneurone stimulates a motor neurone in the spinal cord, and the motor neurone stimulates the flexor muscles. Since two nerve junctions are involved this reflex is *bisynaptic* (synapses are signal junctions between neurones). The interneurone facilitates some processing in the spine so that the withdrawal reflex is capable of adaptation and thus for example, some lateral movement accompanies the withdrawal motion of the hand.

The stretch reflex and the tension reflex are fundamental to muscle control. The stretch reflex causes a muscle to contract when its length or rate of extension increases beyond a given threshold and the tension reflex causes a muscle to relax when the tension exceeds a given threshold.

The stretch reflex is a monosynaptic loop directly from the sensory neurone to the motor neurone. Spindle fibres that respond to changes in muscle length are embedded in the body of a muscle as can be seen in Figure D.1. The spindle fibres' afferents reach into the spinal cord (an *afferent* is a nerve ending carrying information from the periphery of the nervous system in the direction of the central nervous system). These afferents directly stimulate the so-called α motor neurones that activate the muscle. The threshold of the spindle fibres is controlled by γ neurones, the efferents of which contact the spindle fibre (an *efferent* is a nerve ending carrying information from the central nervous system towards the periphery). The brain connects to the γ and α neurones in the spine and the spindle fibre's signals are also reported to the brain. The spinal reflex arc and its monosynaptic nature yield a much faster feedback control process than can be achieved by control from the brain. For example, the knee jerk stretch reflex takes about $50ms$ whereas

hand-eye coordination tasks typically require about a 150-250ms reaction delay.

The tension reflex is a bisynaptic loop. Tension sensors called the 'Golgi apparatus' between muscle fibres and tendons respond to excessive forces. Signals from the Golgi apparatus travel via an interneurone to the α neurones and inhibit them. Since it is bisynaptic this feedback loop is slower than the stretch reflex but is capable of suppressing it. The tension reflex prevents excessive forces being generated and also prevents muscles from tearing themselves or ripping the tendons from the bones. Signals from the Golgi apparatus also travel to the brain.

Muscle action is well understood from the level of motion generating chemical reactions between protein molecules within individual muscle fibres [Nordin and Frankel (1989)] to the biomechanics of the skeletal and muscular system [Pitman and Peterson (1989), Wirhed (1991)].

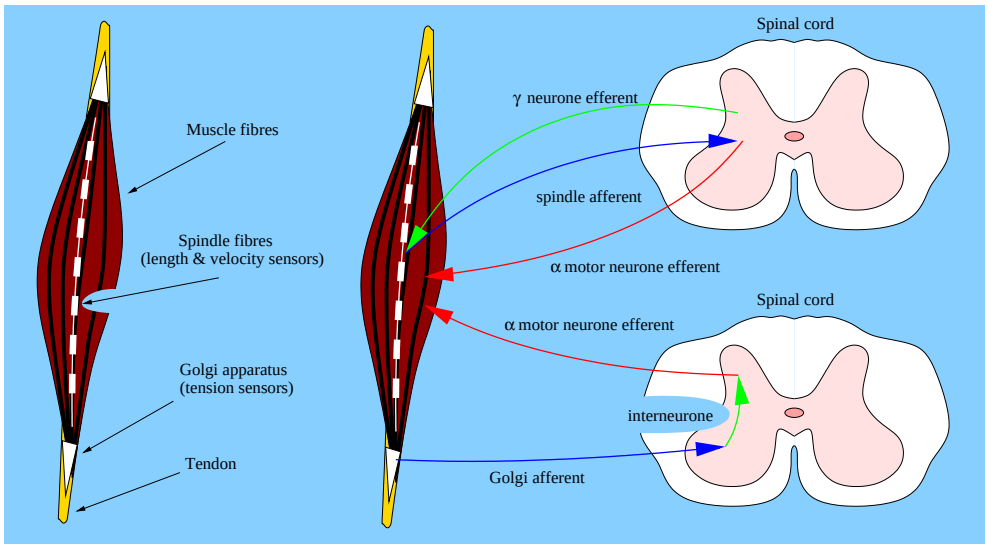

Figure D.1: Spinal reflex system

D.2.2 Force characteristics of muscles

Muscles generate between 30 and 100 Newtons per square centimetre of tissue and are capable of shortening to about half their rest length. Individual fibres contract for about $7.5ms$ (eyeball muscles) up to $100ms$ (leg muscles). Force generation is complex and the torque generated at a joint by a muscle is dependent upon many factors including: the number of muscle fibres contracting, the length of the muscle, the angle of the joint, the velocity of muscle shortening, the temperature of the muscle, and fatigue. As deforming bodies, muscles also have complicated shape characteristics including an elastic spring response to stretch due to the tendons and a viscous damping response to velocity due to blood vessels and spongy material embedded in the muscle [Pitman and Peterson (1989)].

As more muscle fibres contract the force generated increases linearly. Muscle fibres contain *actin* and *myosin* protein molecules that slide along each other in a chemomechanical reaction catalysed by adenosene triphosphate (ATP). These protein molecules are very long but when a muscle is lengthened the number of myosin molecules within reach of actin molecules is reduced so that the force generation reduces. Likewise, when a muscle is shortened actin molecules begin to overlap, once again reducing the efficiency of the muscle. The force generation with respect to length is thus approximately parabolic with its peak around the mean muscle length.

The angle of a joint and the angle of muscle attachment (pennation angle) affect the torque generation at the joint. If the muscle is perpendicular to the bones at a joint the torque generation is optimal. However, the muscles must remain close to the bones and are often almost parallel to the bones they pull, in which case the torque generation is reduced. The velocity of muscle contraction has an inverse affect on force generation. A muscle can generate maximal force when it is stationary. The faster a muscle contracts the fewer the number of muscle fibres that can keep up with the rate of contraction and thus generate force. As muscle temperature increases force generation also increases by as much as 5% per $^{\circ}C$. This increase is also

dependent on muscle velocity and the greatest improvements are observed at high contractile velocities. Finally, when a muscle becomes fatigued there is less ATP available to catalyse fibre contraction and the force generation decreases.

The force response of a muscle is highly dependent upon the stretch reflex. It has been shown that areflexive muscles having no stretch reflex have a linear force/stiffness relationship [Shadmehr and Arbib (1992)]. Normal reflexive muscles however, have an exponential increase in stiffness with muscle force. The stretch reflex thus yields a greater stiffness for lesser base contraction and facilitates more stable joint control in a metabolically efficient manner.

D.3 Human motion production

This section presents some further information related to human motion production that was omitted from the main text for the sake of brevity. This includes a technical description of the 'planar manipulandum', a discussion of some empirical results, some variations on the models already presented and a summary of evidence supporting and assailing a number of models.

D.3.1 The planar manipulandum

The planar manipulandum is a device capable of accurately measuring a human's hand position over a planar surface and supplying forces. The hand is supported by a mechanical framework that is made effectively frictionless through pneumatic cushioning. The hand itself may also be hidden from view and typically a marker indicating hand position is projected from above onto a surface covering the arm. The device may be used to test humans' adaptation to visual and physical perturbations during movement tasks since the position and forces generated by the hand may be measured and physical or visual perturbations may be made. Figure D.2 shows a manipulandum in planar and frontal views.

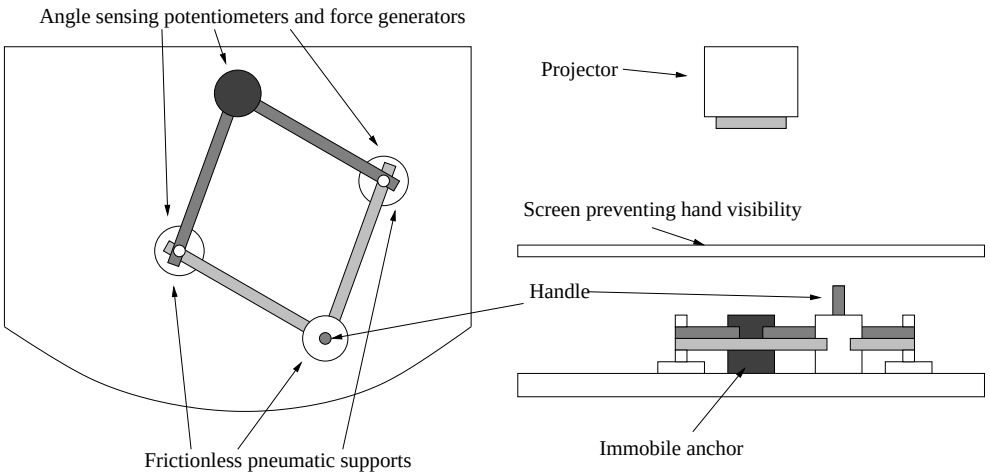

Figure D.2: Planar manipulandum apparatus

D.3.2 Smoothness

It has been observed empirically that the trajectory of the hand in motion tends to be smooth, as was mentioned in Section 2.5.4. The cause of this smoothness is the subject of an unresolved debate. Jordan et al. (1994) argue that smoothness is due to implicit smoothing properties of the dynamic systems underlying movements -the musculo-skeletal system and the physical environment. Wann et al. (1988) propose that humans deliberately move in a way that feels smooth and Kawato (1992, 1996) argues that smoothing occurs at the motor planning level. Either way it is a reasonable assumption that smooth motions reduce the wear and tear on the musculo-skeletal system and are thus desirable.

D.3.3 $\frac{1}{3}$ power law

Empirical analyses have confirmed the validity of the $\frac{1}{3}$ power law [Abend et al. (1982), Flash and Hogan (1985)]. However, although Todorov and Jordan (1998) commented that while the law captures very well the maxima and minima of the actual speed profiles, it is otherwise not a satisfactory model of the details they found in their experimental data. They stated that

the predominant error is in the predicted values at the speed extrema. This is unfortunately implicit in the law since it cannot apply to inflection points or straight segments since zero curvature implies infinite speed and likewise at low curvatures the predictions are dubious. Also, the decrease of velocity to zero at the endpoints of a trajectory cannot be modelled.

The modified $\frac{1}{3}$ power law was proposed [Viviani and Schneider (1991)] to address these complaints and incorporates two free parameters, one of which depends on the average velocity. It is therefore less descriptive but nevertheless can cope with zero velocity and curvature. The authors also noted that the exponent of the law in fact varies slightly with age from about 1/4.35 at age six up to 1/3 at age twelve.

Various theories have been advanced attempting to explain the origins of the $\frac{1}{3}$ power law. Pollick and Sapiro (1997) demonstrated that the law emerges if trajectories obey constant affine velocity suggesting that planar hand trajectories occur at constant velocity in the viewing plane. It has also been suggested that the law is an emergent property of the visco-elastic properties of muscles [Gribble and Ostry (1996)]. Viviani and Flash (1995) claimed that the law is equivalent to the generation of two dimensional motion by coupling a pair of orthogonal harmonic motions.

Perhaps the most robust attack against the $\frac{1}{3}$ power law came from Wann et al. (1988) who demonstrated that curved motion produced with a triangular velocity profile yields a power law with a similar exponent, although this tends rather to discredit interpretations about the way the brain plans movement rather than the actual consistency of the law with real movements. They did however state that "Given the high proportion of significant departures from a one-third law under varying trajectory conditions, it must be concluded on the basis of these data that the law does not adequately describe any general principle of trajectory execution and is a suspect basis for any extended analysis technique." They suggested that the law may be a limit case approximation or may result from the adoption of inappropriate analysis techniques but conceded that the law may be a good approxima-

tion for movements at high tempos whereas motions performed more slowly yield significant deviations. This conclusion is supported by Tomassen and Teulings (1985) who stated that while the law may hold for simple drawing motions it is not robust for complex patterns or normal handwriting.

Finally, although not conclusive evidence it is interesting to note that not only are the kinematic parameters related to the arm's trajectory reflected in the discharge patterns of motor cortical cells, but that the discharge components relating to curvature and velocity are inversely related [Schwartz (1992)].

D.3.4 Minimum jerk

The extrinsic nature of the minimum jerk hypothesis means that trajectories are planned without any consideration of joint angles. In particular, point-point trajectories are always straight. This feature is a common criticism due to the workspace dependent curvature of human point-point motions although some work has demonstrated that to an extent hand trajectories are invariant under translation, rotation, and scaling of time or amplitude [Schmidt (1980, 1985)]. Empirically, minimum jerk has been found to be consistent with fast movements made without accuracy constraints [Kelso et al. (1979), Morasso (1981), Atkeson and Hollerbach (1985)]. Furthermore it has been shown that the jerk cost of movements decreases with practise [Schneider and Zernicke (1989)] suggesting that it is a characteristic reflecting an internal goal of human motor learning.

It has been commented that "the third derivative of position (jerk) seems to be the lowest possible order of the performance index that accounts for the major features of human movements" [Flash and Hogan (1985)]. And indeed it was correspondingly asserted that lower order performance indices such as minimum acceleration and minimum torque predict large discontinuities in acceleration and have been rejected by experimental data [Kawato (1996)]. Schneider and Zernicke (1989) however, observed that since human sensory apparatus reports muscle length and rate of length change it is unclear how

the brain could calculate derivatives of a higher order than acceleration. A detailed experimental criticism [Kawato (1992)] of the model concluded that it is accurate for point-point movements in front of the body, but for trajectories from the side of the body to the front of the body, trajectories resisting a spring and large up-down motions the model does not correspond well with empirical data, neither does it solve the redundant IK and ID problems.

D.3.5 Minimum jerk virtual trajectory control

In support of MJVT it has been stated that "straight virtual trajectories adequately account for the known behaviours of even single-joint movements" [McIntyre and Bizzi (1993)]. According to Flash (1987) the model yields good agreement between measured and predicted hand positions for chosen values of stiffness gain coefficients and fine details of trajectory curvature are also replicated: trajectory hooks, wiggling (oscillations perpendicular to the direction of motion), deviations from straight paths dependent on workspace region, overshooting when stiffness and viscosity do no provide sufficient damping, and velocity profiles skewed towards the onset of motion for faster movements. Flash also criticised the model however, noting that for point-point motions in less than 0.5 seconds trajectories diverge significantly from observed motions. The linear model of stiffness and viscosity may not correspond to actual muscle dynamics and it is difficult to estimate the elastic and viscous coefficients in the PD-servo. For example, the dynamics of muscle excitation and contraction are ignored, time histories of agonist/antagonist contractions are ignored and stiffness is assumed to be invariant with position. Muscle viscosity also depends on velocity non-linearly [Gielen and Houk (1984)]. Kawato (1992) also criticised the model, noting that it does not support the efficient refinement of movement during practise.

While these criticisms are legitimate, many go beyond what can be expected of a simple formulation of motion planning. MJVT is a concise model that accurately replicates many aspects of human motion and partitions the

dynamic aspects of control entirely into the on-line motion production phase. Indeed, the minimum jerk trajectory itself can be generated on-line so that no advance planning beyond the specification of a target point is required. To establish a closer correspondence with the time varying and non-linear properties of the musculo-skeletal system the model could be straightforwardly extended, albeit at the expense of its appealing simplicity.

D.3.6 Cartesian derivative minimisation

The Cartesian derivative based methods such as acceleration, jerk and snap minimisation have received criticism. In particular Cartesian interpolation does not tackle the highly redundant problems of inverse dynamics and inverse kinematics, *i.e.*, they do not solve the problem of finding limb orientations and muscle forces meeting their Cartesian targets. While this does not threaten their validity, more comprehensive models incorporating results such as the kinematic upper arm-forearm phase law are desirable.

Schneider and Zernicke's (1989) observations about the body's sensory apparatus also stand against jerk and snap minimisation:

> Primary sources of afference in the human body provide information about relative position (joint angle receptors) or stretch and rate of stretch (muscle spindles). The crucial question is how a system that primarily senses joint angle, stretch and rate of stretch could know about third derivatives.

D.3.7 Minimum torque change

The minimum torque change model was tested and found to be in agreement with empirical evidence in a number of conditions. Unconstrained horizontal point-point motions in front of the body yielded roughly straight paths with bell shaped velocity profiles coincident with minimum jerk. Point-point motions from the side of the body yielded convexly curved paths. Via-point movements yielded a single peaked velocity profile if the via-point was near

to a straight line path and double peaked profiles when the path was highly curved. The paths were also workspace dependent by comparison to minimum jerk which is workspace independent. Movements performed with an external spring force acting on the arm affected the minimum torque path yielding a non-bell shaped velocity profile that better accords with human motion unders the same conditions.

D.3.8 Minimum motor command change

This model was presented along with a biologically plausible mechanism of calculation in the form of a cascade neural network model [Kawato (1992)]. The neural network itself is trained by back-propagation which is a biologically improbable computation but the network may be unravelled into a serial process of balancing neural signals which is more plausible. The model yields Fitt's law relating movement distance, duration and accuracy [Fitts (1954)]:

$$\text{Movement time} = \beta_0 + \beta_1 \log\left(\frac{\text{Distance}}{\text{Target width}}\right) \tag{D.1}$$

It has been shown that the model yields results consistent with various data including large up-down motions, small fore-aft motions [Atkeson and Hollerbach (1985)] and via-point motions [Uno et al. (1989)] where the hand travels from start to target point and passes through the via-point.

D.3.9 Equilibrium point models

There has been much discussion of the equilibrium point model in human motion production literature. Three different models have emerged. Besides Feldman's so-called λ model, described in Section 2.7.13, Latash has developed what is also referred to as a λ model [Latash and Gottlieb (1991)]. It is similar to Feldman's but the movement of the equilibrium point is more complex because the central controller takes the dynamics of the system into account. While Feldman's model moves the equilibrium point in a monotonic

way from initial to final position, Latash suggests that to deal with problems like inertial loads that must be given sizable acceleration and deceleration forces, the equilibrium point moves in a more N-shaped path, to stretch the spring more and yield larger forces. This is significant in human motor control theory because it assumes that the central nervous system plans for dynamic loads. Bizzi proposed the so-called α model [Bizzi et al. (1976)]. Bizzi's model does not address the problem of moving different dynamic loads, but places more emphasis on the elastic properties of muscles.

All of these models are based on the experimental observation that muscles are controlled by spindle fibre apparatus setting the rest length of exponential spring-like force generators [Feldman (1966b)]. Supporting research has claimed that straight virtual trajectories (of the equilibrium point) adequately account for the known behaviours of elbow movements [McIntyre and Bizzi (1993)]. However, it has also been shown that the joint stiffness of real movements is less than that assumed by the equilibrium point model [Gomi et al. (1992), Gomi and Kawato (1996)].

Bibliography

Abend, W., E. Bizzi, and P. Morasso (1982). Human arm trajectory formation. *Brain 105*, 331–348.

Aftomsis, M. (1996, September). Intersection of Generally Positioned Polygons in R^3. http://~aftomsis/cart3d/degeneracy/bool-intersection.html.

Amaya, K., A. Bruderlin, and T. Calvert (1996, May). Emotion from Motion. In *Proceedings of Graphics Interface 96*, Toronto, Canada, pp. 222–229.

Armstrong, W. and M. Green (1985, May). The Dynamics of Articulated Rigid Bodies for Purposes of Animation. In *Proceedings of Graphics Interface 85*, Montreal, Quebec, Canada, pp. 407–415.

Atkeson, C. and J. Hollerbach (1985). Kinematic features of unrestrained arm movements. *Journal of Neuroscience 5*, 2318–2330.

Atkeson, C. G., J. G. Hale, F. Pollick, M. Riley, S. Kotosaka, S. Schaal, T. Shibata, G. Tevatia, A. Ude, S. Vijayakumar, and M. Kawato (2000, July-August). Using Humanoid Robots to Study Human Behavior. *IEEE Intelligent Systems Magazine, special issue on Humanoids 15*(4).

Aydin, Y. and M. Nakajima (1999, May). Realistic Articulated Character Positioning and Balance Control in Interactive Environments. In *Computer Animation 99*, Geneva, Switzerland, pp. 160–168. IEEE Computer Society Press.

Badler, N. I. (1982, November). Human Body Models and Animation. *IEEE Computer Graphics and Applications 2*, 6–7.

Badler, N. I. and D. M. Chi (1999, May). Virtual Human Animation Based on Movement Observation and Cognitive Behavior Models. In *Computer Animation 99*, Geneva, Switzerland, pp. 128–137. IEEE Computer Society Press.

Badler, N. I., C. Phillips, and B. Webber (1993). *Simulating Humans*. Oxford University Press.

Baerlocher, R. and R. Boulic (1998, October). Task-Priority Formulations for the Kinematic Control of Highly Redundant Articulated Structures. In *IEEE IROS 98, International Conference on Intelligent Robots and Systems, Victoria, Canada*, Victoria, Canada, pp. 323–329.

Baraff, D. (1994, July). Fast Contact Force Computation for Nonpenetrating Rigid Bodies. In *SIGGRAPH 94*, ACM Computer Graphics Proceedings, Annual Conference Series, Orlando, Florida, pp. 23–34. Addison-Wesley.

Baraff, D. (1995, August). Rigid Body Dynamics. In *An Introduction to Physically Based Modelling*, ACM Computer Graphics, SIGGRAPH 95 Course Notes 34, Los Angeles, California, pp. G1–G68.

Baraff, D. (1996, August). Linear-Time Dynamics Using Lagrange Multipliers. In *SIGGRAPH 96*, ACM Computer Graphics Proceedings, Annual Conference Series, New Orleans, LA, pp. 63–70. Addison-Wesley.

Barzel, R. and A. H. Barr (1988). A Modelling System Based On Dynamic Constraints. In *SIGGRAPH 88*, ACM Computer Graphics Proceedings, Annual Conference Series, pp. 179–188.

Bentivegna, D. and C. Atkeson (2000, September). Using Primitives in Learning from Observation. In *Humanoids 2000*, Massachusetts Institute of Technology, Boston, MA, USA.

Bhat, D. and J. K. Kearney (1996, October). On Animating Whip-type Motions. *The Journal of Visualization and Computer Animation 7*(4), 229–249.

Binet, A. and J. Couriet (1893). Sur la vitesse des mouvements graphiques. *Travaux du Laboratoire de psychologie physiologique, La Revue de France et de l'étranger 35*, 664–671. In French.

Bingham, G. (1993). Scaling judgements of lifted weight: Lifter size and the role of the standard. *Ecological Psychology 5*(1), 31–64.

Bizzi, E., A. Polit, and P. Morasso (1976). Mechanisms underlying the achievement of final head position. *Journal of Neurophysiology 39*, 434–444.

Boone, G. N. and J. K. Hodgins (1997). Slipping and Tripping Reflexes for Bipedal Robots. *Autonomous Robots, Special Issue on Biped Locomotion of Autonomous Robots 4*(3).

Boulic, R., R. Mas-Sanso, and D. Thalmann (1997, July-September). Complex Character Positioning Based on a Compatible Flow Model of Multiple Supports. *The Journal of Visualization and Computer Animation 3*(3), 245–261.

Breen, D. E. (1997, October-December). Cost Minimization for Animated Geometric Models in Computer Graphics. *The Journal of Visualization and Computer Animation 8*(4), 201–220.

Brogan, D. C., R. A. Metoyer, and J. K. Hodgins (1998). Dynamically Simulated Characters in Virtual Environments. *IEEE Computer Graphics and Applications 18*(5), 58–69.

Bruderlin, A. and L. Williams (1995, August). Motion Signal Processing. In R. Cook (Ed.), *SIGGRAPH 95*, ACM Computer Graphics Proceedings, Annual Conference Series, Los Angeles, California, pp. 97–104. Addison-Wesley.

Burbanks, A. (1996, November). Quaternions in C++. http://info.lboro.ac.uk/departments/ma/gallery/quat/intro.html.

Cerezo, E., A. Pina, and F. J. Seron (1999). Motion and Behaviour Modelling: State of Art and New Trends. *The Visual Computer 15*(3), 124–146.

Cohen, M. F. (1992, July). Interactive Spacetime Control for Animation. In *SIGGRAPH 92*, ACM Computer Graphics Proceedings, Annual Conference Series, Chicago, IL, pp. 293–302. Addison-Wesley.

Cutting, J. and L. Kozlowski (1977). Recognizing Friends by Their Walk: Gait Perception Without Familiarity Cues. *Bulletin Psychonomic Society 9*(5), 353–356.

Dempster, W. and G. Gaughran (1967). Properties of Body Segments Based on Size and Weight. *American Journal of Anatomy 120*, 34–54.

Densley, D. and P. Willis (1997, June). Emotional Posturing: A method towards achieving emotional figure animation. In *Computer Animation 97*, Geneva, Switzerland, pp. 8–14. IEEE Computer Society Press.

Derwort, A. (1938). Untersuchungen über den Zeitablauf figurierter Bewegungen beim Menchen (Investigations on the time course of tracing movements). *Pflügers Archiv für die Gesamte Physiologie des Menschen und der Tiere 240*, 661–675.

Dooley, M. (1982, November). Anthropometric Modeling Programs - A survey. *IEEE Computer Graphics and Applications 2*, 17–25.

Dornay, M., Y. Uno, M. Kawato, and R. Suzuki (1996). Minimum muscle tension change trajectories predicted by using a 17-muscle model of the monkey's arm. *Journal of Motor Behavior 28*, 83–100.

Dworkin, P. and D. Zeltzer (1993, September 4-5). A New Model for Efficient Dynamic Simulation. In *4th Eurographics Workshop on Animation and Simulation*, pp. 135–147.

El-Nasr, M. S., T. Ioerger, and J. Yen (1999, May). Emotionally Expressive Agents. In *Computer Animation 99*, Geneva, Switzerland, pp. 48–57. IEEE Computer Society Press.

Feldman, A. (1966a). Functional Tuning of the Nervous System With Control of Movement or Maintenance of a Steady Posture -II. Controllable Parameters of The Muscles. *Biophysics 11*, 565–578.

Feldman, A. (1966b). Functional Tuning of the Nervous System With Control of Movement or Maintenance of a Steady Posture -III. Mechanographic analysis of the execution by man of the simplest motor tasks. *Biophysics 11*, 766–775.

Feldman, A. (1986). Once more on the equilibrium point hypothesis (lambda-model) for motor control. *Journal of Motor Behavior 25*, 140–152.

Fitts, P. (1954). The information capacity of the human motion system in controlling the amplitude of movement. *Journal of Experimental Psychology 47*, 381–391.

Flash, T. (1983). *Organizing principles underlying the formation of hand trajectories. Ph.D. thesis.* Harvard/MIT Division of Health Sciences and Technology.

Flash, T. (1987). The control of hand equilibrium trajectories in multi-joint arm movements. *Biological Cybernetics 57*, 257–274.

Flash, T. and N. Hogan (1985). The co-ordination of arm movements: An experimentally confirmed mathematical model. *Journal of Neuroscience 5*, 1688–1703.

Fley, J. D., A. van Dam, S. K. Feiner, J. F. Hughes, and R. L. Phillips (1995). *Introduction to Computer Graphics*. Addison-Wesley. ISBN 0-201-60921-5.

Foley, J. (1980). Binocular distance perception. *Psychological Review 87*(5), 411–434.

Gielen, C. and J. Houk (1984). Nonlinear viscosity of human wrist. *Journal of Neurophysiology 52*, 553–569.

Girard, M. (1991). Constrained Optimization of Articulated Animal Movement in Computer Animation. In N. Badler, B. Barsky, and D. Zeltzer (Eds.), *Making them Move*, Chapter 10, pp. 209–233. Morgan Kaufmann Publishers. ISBN 1-55860-106-6.

Gleicher, M. (1997, April). Motion Editing with Spacetime Constraints. In *Proceedings of 1997 Symposium on Interactive 3D Graphics*, Providence, RI.

Gomi, H. and M. Kawato (1996). Equilibrium-point control hypothesis examined by measured arm stiffness during multijoint movement. *Science 272*, 117–120.

Gomi, H., Y. Koike, and M. Kawato (1992). Human hand stiffness during discrete point-to-point multi-joint movement. In *Proceedings of IEEE Engineering in Medicine and Biology Society*, Los Alamos, CA, pp. 1628–1629.

Green, M. (1991). Using Dynamics in Computer Animation: Control and Solution Issues. In N. Badler, B. Barsky, and D. Zeltzer (Eds.), *Making them Move*, Chapter 14, pp. 281–315. Morgan Kaufmann Publishers. ISBN 1-55860-106-6.

Gribble, P. and D. Ostry (1996). Origins of the power-law relation between movement velocity and curvature - modelling the effects of muscle mechanics and limb dynamics. *Journal of Neurophysiology 76*, 2853–2860.

Grzeszczuk, R. and D. Terzopoulos (1995, August). Automated Learning of Muscle-Actuated Locomotion Through Control Abstraction. In R. Cook (Ed.), *SIGGRAPH 95*, ACM Computer Graphics Proceedings, Annual Conference Series, Los Angeles, California, pp. 63–70. Addison-Wesley.

Haggard, P. and J. Richardson (1996). Spatial patterns in the control of human arm movement. *Journal of Experimental Psychology: Human Perception and Performance 22*, 42–46.

Hale, J. G. and F. E. Pollick (2000a, 7-8 September). Playing 'Sticky Hands' With a Humanoid Robot. In *Proceedings of the First IEEE-RAS International Conference on Humanoid Robots (Humanoids 2000)*, Massachusetts Institute of Technology, Boston, MA, USA.

Hale, J. G. and F. E. Pollick (2000b, April-May). 'Sticky Hands' Interaction with an Anthropomorphic Robot. In *WIRE 2000, Workshop on Interactive Robotics and Entertainment*, Carnegie Mellon University, Pittsburgh, PA, pp. 117–124.

Hale, J. G. and F. E. Pollick (2002, August 10-11). Biomimetic motion synthesis for the upper limb based on human motor production. In *Workshop for motor control in humans and robots*, Edinburgh University.

Harris, C. and D. Wolpert (1998). Signal-dependent noise determines motor planning. *Nature 394*, 780–784.

Hartley, R. (1985). *Linear and Nonlinear Programming: An Introduction to Linear Methods in Mathematical Programming.* Ellis Horwood Limited. ISBN 0-85312-644-5.

Hasan, Z. (1986). Optimized Movement Trajectories and Joint Stiffness. *Biological Cybernetics 53*, 373–382.

Hatze, H. (1978). A general myocybernetic control model of skeletal muscle. *Biological Cybernetics 28*, 143–157.

Hill, A. (1938). The heat of shortening and the dynamic constants of muscle. In *Proceedings of the Royal Society of London - Series B: Biological Sciences*, pp. 136–195.

Ho, K. H. (1996, February). A Fuzzy Adaptive Model of Emotion and Personality for an Autonomous Robot. In *AROB 96*, Proceedings of the International Symposium on Artificial Life and Robotics, Beppu, Japan, pp. 150–154.

Hodgins, J. K., J. F. O'Brien, and J. Tumblin (1998, October-December). Perception of Human Motion With Different Geometric Models. *IEEE Transactions on Visualization and Computer Graphics 4*(4).

Hodgins, J. K. and N. S. Pollard (1997). Adapting Simulated Behaviors For New Characters. In *SIGGRAPH 97*, ACM Computer Graphics Proceedings, Annual Conference Series, Los Angeles, California, pp. 153–162. Addison-Wesley.

Hodgins, J. K., W. Wooten, D. Brogan, and J. O'Brien (1995, August). Animating Human Athletics. In R. Cook (Ed.), *SIGGRAPH 95*, ACM Computer Graphics Proceedings, Annual Conference Series, Los Angeles, California, pp. 71–78. Addison-Wesley.

Hoff, B. and A. Arbib (1993). Models of Trajectory Formation and Temporal Interaction of Reach and Grasp. *Journal of Motor Behavior 25*(3), 175–192.

Hogan, N. (1984). An organizing principle for a class of voluntary movements. *Journal of Neuroscience 4*, 2745–2754.

Hollerbach, J. (1990). Planning of arm movements. In D. Osherson, S. Kosslyn, and J. Hollerbach (Eds.), *Visual cognition and action*, pp. 183–211. MIT Press.

Hollerbach, J. and C. Atkeson (1986). Characterization of joint-interpolated arm movements. In H. Heuer and C. Fromm (Eds.), *Generation and modulation of action patterns*, pp. 41–54. Springer-Verlag.

Isaacs, P. M. and M. F. Cohen (1987). Controlling Dynamic Simulation with Kinematic Constraints, Behaviour Functions and Inverse Dynamics. In

SIGGRAPH 87, ACM Computer Graphics Proceedings, Annual Conference Series, pp. 215–224.

Isaacs, P. M. and M. F. Cohen (1988). Mixed Methods for Complex Kinematic Constraints in Dynamic Figure Animation. *The Visual Computer 4*, 296–305.

Jordan, M., T. Flash, and Y. Arnon (1994). A Model of the Learning of Arm Trajectories from Spatial Deviations. *Journal of Cognitive Neuroscience 6*(4), 359–376.

Kass, M. (1995a, August). Energy Functions and Stiffness. In *An Introduction to Physically Based Modelling*, ACM Computer Graphics, SIGGRAPH 95 Course Notes 34, Los Angeles, California, pp. D1–D5.

Kass, M. (1995b, August). An Introduction to Continuum Dynamics for Computer Graphics. In *An Introduction to Physically Based Modelling*, ACM Computer Graphics, SIGGRAPH 95 Course Notes 34, Los Angeles, California, pp. E1–E12.

Kavounoudias, A. and J.-P. Roll (1987). The plantar sole is a 'dynamometric map' for human balance control. *NeuroReport 9*, 3247–3252.

Kawato, M. (1992). Optimization and Learning in Neural Networks for Formation and Control of Coordinated Movement. In D. Meyer and S. Kornblum (Eds.), *Attention and performance XIV*, pp. 821–849. MIT Press.

Kawato, M. (1996). Trajectory formation in arm movements: Minimization principles and procedures. *Advances in motor learning and control*, 225–259.

Kearney, J., S. Hansen, and J. Cremer (1993, April-June). Programming Mechanical Simulations. *The Journal of Visualization and Computer Animation 4*(2), 113–129.

Kelso, J., D. Southard, and D. Goodman (1979). On the nature of human interlimb coordination. *Science 203*, 1029–1031.

Ko, H. and N. I. Badler (1996, March). Animating Human Locomotion with Inverse Dynamics. *IEEE Computer Graphics and Applications 16*(2), 50–59.

Koga, Y., K. Kondo, J. Kuffner, and J.-C. Latombe (1994, July). Planning Motions with Intentions. In *SIGGRAPH 94*, ACM Computer Graphics Proceedings, Annual Conference Series, Orlando, Florida, pp. 395–408. Addison-Wesley.

Kojic, M., Mijailovic, and N. Zdravkovic (1998). Modeling of Muscle Behaviour by the Finite Element Method Using Hill's Three-Element Model. *International Journal for Numerical Methods in Engineering 43*, 941–953.

Kōmura, T. and T. L. Kunii (1997). A Muscle-Based Feedforward Controller of the Human Body. *Computer Graphics forum 16*(3), 165–176.

Kōmura, T., Y. Shinagawa, and T. L. Kunii (1999). Calculation and Visualization of the Dynamic Ability of the Human Body. *The Journal of Visualization and Computer Animation 10*, 57–78.

Korein, J. U. and N. I. Badler (1982). Techniques for Generating the Goal-Directed Motion of Articulated Figures. *IEEE Computer Graphics and Applications 2*(9), 71–81.

Kunii, T. L. and L. Sun (1990). Dynamic Analysis-Based Human Animation. In T. Chua and T. Kunii (Eds.), *CG International '90*. Springer-Verlag.

Lacquaniti, F., C. Terzuolo, and P. Viviani (1983). The law relating kinematic and figural aspects of drawing movements. *ACTA Psychol. 54*, 115–130.

Lake, R. and M. Green (1991, 23-26 September). Dynamic Motion Control of an Articulated Figure Using Quaternion Curves. In J. Staudhammer

and Q. Peng (Eds.), *Proceedings of CAD / Graphics 91*, Hangzhou, China. International Academic Publishers.

Lamouret, A. and M.-P. Gascuel (1996). Scripting Interactive Physically-Based Motions with Relative Paths and Synchronization. *Computer Graphics forum 15*(1), 25–34.

Laszlo, J., M. van de Panne, and E. Fiume (1996, August). Limit Cycle Control and its Application to the Animation of Balancing and Walking. In *SIGGRAPH 96*, ACM Computer Graphics Proceedings, Annual Conference Series, New Orleans, LA, pp. 155–162. Addison-Wesley.

Latash, M. and G. Gottlieb (1991). An equilibrium-point model for fast single-joint movement. I. Emergence of strategy-dependent EMG patterns. *Journal of Motor Behavior 23*, 163–178.

Lathrop, R. H. (1986, 7-10 April). Constrained (Closed-Loop) Robot Simulation by Constraint Propagation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, San Francisco, CA.

Lauk, M., C. C. Chow, A. E. Pavlik, and J. J. Collins (1998, Jan). Human Balance Out of Equilibrium: Nonequilibrium Statistical Mechanics in Posture Control. *Statistical Review Letters 80*(2).

Li, Z., S. J. Gortler, and M. F. Cohen (1994, July). Hierarchical Spacetime Control. In *SIGGRAPH 94*, ACM Computer Graphics Proceedings, Annual Conference Series, Orlando, Florida, pp. 35–42. Addison-Wesley.

Lien, S. and J. Kajiya (1984). A Symbolic Method for Calculating the Integral Properties of Arbitrary Nonconvex Polyhedra. *IEEE Computer Graphics and Applications 4*(10), 35–41.

Maciejewski, A. A. and C. Klein (1985). SAM-Animation Software for Simulating Articulated Motion. *Computers and Graphics 9*(4), 383–391.

McIntyre, J. and E. Bizzi (1993). Servo hypotheses for the biological control of movement. *Journal of Motor Behaviour 25*, 193–202.

McKenna, M. and D. Zeltzer (1996). Dynamic Simulation of a Complex Human Figure Model with Low Level Behavior Control. *Presence Teleoperators and Virtual Environments 5*(4), 431–456.

Mirtich, B. (1998, May). Rigid Body Contact: Collision Detection to Force Computation. In *Proceedings of the Workshop on Contact Analysis and Simulation, IEEE International Conference on Robotics and Automation.* Also available as MERL Technical Report 98-01.

Morasso, P. (1981). Spatial control of arm movements. *Experimental Brain Research 42*, 233–237.

Morasso, P. and F. Mussa-Ivaldi (1982, March). Trajectory Formation and Handwriting: A Computational Model. *Biological Cybernetics 16*(2), 50–59.

Multon, F. and B. Arnaldi (1999, May). Human Motion Coordination: Example of a Juggler. In *Computer Animation 99*, Geneva, Switzerland, pp. 148–159. IEEE Computer Society Press.

Nakano, E., H. Imamizu, R. Osu, Y. Uno, H. Gomi, T. Yoshioka, and M. Kawato (1999). Quantitative examinations of internal representations for arm trajectory planning: Minimum commanded torque change model. *Journal of Neurophysiology 81*, 2140–2155.

Nelson, W. (1983). Physical principles for economies of skilled movements. *Biological Cybernetics 46*, 135–147.

Ngo, J. T. and J. Marks (1993, August). Spacetime Constraints Revisited. In *SIGGRAPH 93*, ACM Computer Graphics Proceedings, Annual Conference Series, Anaheim, California, pp. 343–350. Addison-Wesley.

Nordin, M. and V. H. Frankel (1989). *Basic Biomechanics of the Musculoskeletal System* (Second ed.). Philadelphia, London: Lea & Febiger. ISBN 0-8121-1227-X.

Osu, R., Y. Uno, Y. Koike, and M. Kawato (1997). Possible Explanations for Trajectory Curvature in Multijoint Arm Movements. *Journal of Experimental Psychology, Human Perception and Performance 23*(3), 890–913.

Park, J., D. Fussell, M. Pandy, and J. C. Browne (1992, April). Realistic Animation Using Musculotendon Skeletal Dynamics. Technical Report TR 92-26, Department of Computer Science, University of Texas at Austin.

Pearson, K. (1991). Sensory Elements in Pattern Generating Networks. In N. Badler, B. Barsky, and D. Zeltzer (Eds.), *Making them Move*, Chapter 5, pp. 111–129. Morgan Kaufmann Publishers. ISBN 1-55860-106-6.

Pitman, M. and L. Peterson (1989). Biomechanics of Skeletal Muscle. In M. Nordin and V. Frankel (Eds.), *Basic Biomechanics of the Musculoskeletal System*, Chapter 5. Lea & Febiger. ISBN 0-8121-1227-X.

Pollick, F. E. and G. Sapiro (1997). Constant Affine Velocity Predicts the 1/3 Power Law of Planar Motion Perception and Generation. *Vision Research 37*(3), 347–353.

Press, W. H. (Ed.) (1992). *Numerical Recipes in C: The art of scientific computing*. Press Syndicate of the University of Cambridge, Cambridge. (available online at www.nr.com).

Ramsay, J. and B. Silverman (1997). *Functional Data Analysis*. New York: Springer. ISBN 0-387-94956-9.

Riley, M. and C. G. Atkeson (2002). Robot Catching: Towards Engaging Human-Humanoid Interaction. *Autonomous Robots 12*, 119–128.

Riley, M., A. Ude, and C. G. Atkeson (2000). Methods for Motion Generation and Interaction with a Humanoid Robot: Case Studies of Dancing and

Catching. In *Proceedings of the AAAI and CMU Workshop on Interactive Robotics and Entertainment 2000*, Pittsburgh, Pennsylvania.

Roberts, T. D. (1995). *Understanding Balance The Mechanics of Posture and Locomotion* (First ed.). 2-6 Boundary Row, London SE1 8HN, UK: Chapman & Hall. ISBN 0-412-60160-5.

Rose, C., B. Guenter, B. Bodenheimer, and M. Cohen (1996, August). Efficient Generation of Motion Transitions using Spacetime Constraints. In *SIGGRAPH 96*, ACM Computer Graphics Proceedings, Annual Conference Series, New Orleans, LA, pp. 147–154. Addison-Wesley.

Schmidt, R. (1980). On the theoretical status of time in motor program representations. In G. Stelmach and J. Requin (Eds.), *Tutorials in Motor Behavior*, pp. 145–165. Amsterdam: Elsevier-North Holland, Publishing Co.

Schmidt, R. (1985). Identifying units of motor behavior. *Behavioral and Brain Sciences 8*, 163–164.

Schneider, K. and R. Zernicke (1989). Jerk-cost modulations during the practice of rapid arm movements. *Biological Cybernetics 60*, 221–230.

Schwartz, A. (1992). Motor cortical activity during drawing movements: Single-unit activity during sinusoidal tracing. *Journal of Neurophysiology 68*, 528–541.

Sejnowski, T. (1998, August). Making smooth moves. *Nature 394*, 725–726.

Shadmehr, R. and M. Arbib (1992). A Mathematical Analysis of the Force-Stiffness Characteristics of Muscles in Control of a Single Joint System. *Biological Cybernetics 66*, 463–477.

Shinagawa, Y., J.-i. Nakajima, T. L. Kunii, and K. Hara (1997, June). Capturing and Analyzing Stability of Human Body Motions Using Video Cam-

eras. In *Computer Animation 97*, Geneva, Switzerland, pp. 48–57. IEEE Computer Society Press.

Soechting, J. and F. Lacquaniti (1981). Invariant characteristics of pointing movements in man. *Journal of Neuroscience 1*, 710–720.

Soechting, J., F. Lacquaniti, and C. Terzuolo (1986). Coordinates of arm movements in three-dimensional space. *Neuroscience 17*, 295–311.

Soechting, J. and C. Terzuolo (1986). An algorithm for the generation of curvilinear wrist motion in an arbitrary plane in three-dimensional space. *Neuroscience 19*(4), 1393–1405.

Soechting, J. and C. Terzuolo (1987a). Organization of Arm Movements in Three-Dimensional Space. Wrist Motion is Piecewise Planar. *Neuroscience 23*(1), 53–61.

Soechting, J. and C. Terzuolo (1987b). Organization of Arm Movements. Motion is Segmented. *Neuroscience 23*(1), 39–51.

Stokes, V. P., H. Lanshammar, and A. Thorstensson (1999, January). Dominant Pattern Extraction From 3-D Kinematic Data. *IEEE Transactions on Biomedical Engineering 46*(1), 100–106.

Tevatia, G. and S. Schaal (2000). Inverse Kinematics for Humanoid Robots. In *IEEE International Conference on Robotics and Automation (ICRA 2000)*.

Todorov, E. and M. I. Jordan (1998). Smoothness Maximization Along a Predefined Path Accurately Predicts the Speed Profiles of Complex Arm Movements. The American Physiological Society, by reprint: E.Todorov, MIT, E25-526, 45 Carleton St., Cambridge, MA 02139.

Tomassen, A. and H. Teulings (1985). Time, size and shape in handwriting: exploring spatio-temporal relationships at different levels. In J. Michon and J. Jackson (Eds.), *Time, mind & behaviour*. Springer-Verlag.

Turvey, M., E. Saltzman, and R. Schmidt (1991). Dynamics and Task-Specific Coordinations. In N. Badler, B. Barsky, and D. Zeltzer (Eds.), *Making them Move*, Chapter 7, pp. 157–171. Morgan Kaufmann Publishers. ISBN 1-55860-106-6.

Ude, A., C. Man, M. Riley, and C. Atkeson (2000, September). Automatic Generation of Kinematic Models for the Conversion of Human Motion Capture Data into Humanoid Robot Motion. In *IEEE-RAS International Conference on Humanoid Robots*, Cambridge, Massachusetts.

Uno, Y., M. Kawato, and R. Suzuki (1989). Formation and control of optimal trajectory in human multi-joint arm movement - minimum torque change model. *Biological Cybernetics 61*, 89–101.

Uno, Y., R. Suzuki, and M. Kawato (1989). Minimum muscle-tension change model which reproduces human arm movement. *Proceedings of the 4th Symposium on Biological and Physical Engineering*, 229–302. (Japanese).

Unuma, M., K. Anjyo, and R. Takeuchi (1995, August). Fourier Principles for Emotion-based Human Figure Animation. In *SIGGRAPH 95*, ACM Computer Graphics Proceedings, Annual Conference Series, Los Angeles, California, pp. 91–96. Addison-Wesley.

van de Panne, M. and E. Fiume (1993, August). Sensor-Actuator Networks. In *SIGGRAPH 93*, ACM Computer Graphics Proceedings, Annual Conference Series, Anaheim, California, pp. 335–343. Addison-Wesley.

van Overveld, C. (1993, October-December). Building Blocks for Goal-direction Motion. *The Journal of Visualization and Computer Animation 4*(4), 223–250.

van Overveld, C. (1994, January-March). A Simple Approximation to Rigid Body Dynamics for Computer Animation. *The Journal of Visualization and Computer Animation 15*(1), 17–36.

van Overveld, C. and E. van Loon (1992, January-March). Hanging Cloths and Dangling Rods: a Unified Approach to Constraints in Computer Animation. *The Journal of Visualization and Computer Animation 3*(1), 45–59.

Vasilonikolidakis, N. and G. J. Clapworthy (1991). Inverse Lagrangian Dynamics for Animating Articulated Models. *The Journal of Visualization and Computer Animation 2*, 105–113.

Vemuri, B., Y. Cao, and L. Chen (1998). Fast Collision Detection Algorithms with Applications to Particle Flow. *Computer Graphics forum 7*(2), 131–134.

Viviani, P. and T. Flash (1995). Minimum-Jerk, Two-Thirds Power Law, and Isochrony: Converging Approaches to Movement Planning. *Journal of Experimental Psychology: Human Perception and Performance 21*(1), 32–53.

Viviani, P. and R. Schneider (1991). A developmental study of the relation between geometry and kinematics in drawing movements. *Journal of Experimental Psychology: Human Perception and Performance 17*, 198–218.

Viviani, P. and C. Terzuolo (1982). Trajectory determines movement dynamics. *Neuroscience 7*, 431–437.

Wann, J., I. Nimmo-Smith, and A. Wing (1988). Relation Between Velocity and Curvature in Movement: Equivalence and Divergence Between a Power Law and a Minimum-Jerk Model. *Journal of Experimental Psychology: Human Perception and Performance 14*(4), 622–637.

Watt, A. and M. Watt (1995). *Advanced Animation and Rendering Techniques Theory and Practice*. Addison-Wesley. ISBN 0-201-54412-1.

Wilhelms, J. (1987, June). Using Dynamic Analysis for Realistic Animation of Articulated Bodies. *IEEE Computer Graphics and Applications 7*(6), 12–27.

Wilhelms, J. (1991). Dynamic Experiences. In N. Badler, B. Barsky, and D. Zeltzer (Eds.), *Making them Move*, Chapter 13, pp. 265–281. Morgan Kaufmann Publishers. ISBN 1-55860-106-6.

Wilhelms, J. and B. Barsky (1985, May). Using Dynamic Analysis to Animate Articulated Bodies such as Humans or Robots. In *Proceedings of Graphics Interface 85*, Montreal, Quebec, Canada, pp. 97–104.

Wilhelms, J. and D. R. Forsey (1988, June). Techniques for Interactive Manipulation of Articulated Bodies Using Dynamic Analysis. In *Graphical Interface*, Calgary, Alberta, Canada, pp. 8–15. Canadian Information Processing Society.

Wilhelms, J., M. Moore, and R. Skinner (1988, December). Dynamic Simulation: Interaction and Control. *The Visual Computer 4* (1), 283–295.

Wing, A. and E. Miller (1984). Peak velocity timing invariance. *Psychological Research 46*, 121–127.

Winters, J. and L. Stark (1987). Muscle Models: What is Gained and What is Lost by Varying Model Complexity. *Biological Cybernetics 55*, 403–420.

Wirhed, R. (1991). *Athletic Ability & the Anatomy of Motion*. Wolfe Medical Publications Ltd. ISBN 0-7234-1540-4.

Witkin, A. (1995a, August). Constrained Dynamics. In *An Introduction to Physically Based Modelling*, ACM Computer Graphics, SIGGRAPH 95 Course Notes 34, Los Angeles, California, pp. F1–F12.

Witkin, A. (1995b, August). Particle System Basics. In *An Introduction to Physically Based Modelling*, ACM Computer Graphics, SIGGRAPH 95 Course Notes 34, Los Angeles, California, pp. C1–C12.

Witkin, A. and D. Baraff (1995, August). Differential Equation Basics. In *An Introduction to Physically Based Modelling*, ACM Computer Graphics, SIGGRAPH 95 Course Notes 34, Los Angeles, California, pp. B1–B8.

Witkin, A., M. Gleicher, and W. Welch (1995). Interactive Dynamics. In *SIGGRAPH90-3D*, Proceedings of the Symposium on Interactive 3D Graphics, Snowbird, UT, USA, pp. 11–22.

Witkin, A. and M. Kass (1988). Spacetime Constraints. In *SIGGRAPH 88*, ACM Computer Graphics Proceedings, Annual Conference Series, pp. 159–168.

Wolpert, D. (1997). Computational approaches to motor control. *Trends in Cognitive Sciences 11*(6), 209–216.

Wolpert, D., Z. Ghahramani, and M. Jordan (1994). Perceptual distortion contributes to the curvature of human reaching movements. *Experimental Brain Research 98*, 153–156.

Wooten, W. L. and J. K. Hodgins (1997, August). Transitions Between Dynamically Simulated Motions: Leaping, Tumbling, Landing and Balancing. In *Visual Proceedings of ACM SIGGRAPH 97*, Los Angeles, CA.

Yamaguchi, F. and M. Niizeki (1997). Some basic geometric test conditions in terms of Plücker coordinates and Plücker coefficients. *The Visual Computer 13*(1), 29–41.

Yamane, K. and Y. Nakamura (2000, April-May). Dynamics Filter: Towards Real-Time and Interactive Motion Generator for Human Figures. In *WIRE 2000, Workshop on Interactive Robotics and Entertainment*, Carnegie Mellon University, Pittsburgh, PA, pp. 27–34.

Yang, L., F. Albregtsen, and T. Taxt (1997, March). Fast Computation of Three-Dimensional Geometric Moments Using a Discrete Divergence Theorem. *Graphical Models and Image Processing 59*(2), 97–108.

Zeltzer, D. (1982). Motor Control Techniques for Figure Animation. *IEEE Computer Graphics and Applications 2*(9), 53–59.

Zhao, J. and N. Badler (1994, October). Inverse Kinematics Positioning Using Nonlinear Programming for Highly Articulated Figures. *ACM Transactions on Graphics 13*(4), 313–336.

Index