Minimal Motion Capture with Inverse Kinematics for Articulated Human Figure Animation

Submitted in fulfilment of the requirements for
the Degree of
MASTER OF SCIENCE
of Rhodes University

by

LUIS CASANUEVA

February 1999

Abstract

Animating an articulated figure usually requires expensive hardware in terms of motion capture equipment, processing power and rendering power. This implies a high cost system and thus eliminates the use of personal computers to drive avatars in virtual environments.

We propose a system to animate an articulated human upper body in real-time, using minimal motion capture trackers to provide position and orientation for the limbs. The system has to drive an avatar in a virtual environment on a low-end computer. The cost of the motion capture equipment must be relatively low (hence the use of minimal trackers).

We discuss the various types of motion capture equipment and decide to use electromagnetic trackers which are adequate for our requirements while being reasonably priced. We also discuss the use of inverse kinematics to solve for the articulated chains making up the topology of the articulated figure.

Furthermore, we offer a method to describe articulated chains as well as a process to specify the reach of up to four link chains with various levels of redundancy for use in articulated figures. We then provide various types of constraints to reduce the redundancy of non-defined articulated chains, specifically for chains found in an articulated human upper body. Such methods include a way to solve for the redundancy in the orientation of the neck link, as well as three different methods to solve the redundancy of the articulated human arm. The first method involves eliminating a degree of freedom from the chain, thus reducing its redundancy. The second method calculates the elevation angle of the elbow position from the elevation angle of the hand. The third method determines the actual position of the elbow from an average of previous positions of the elbow according to the position and orientation of the hand. The previous positions of the elbow are captured during the calibration process.

The redundancy of the neck is easily solved due to the small amount of redundancy in the chain. When solving the arm, the first method which should give a perfect result in theory, gives a poor result in practice due to the limitations of both the motion capture equipment and the design. The second method provides an adequate result for the position of the redundant elbow in most cases although fails in some cases. Still it benefits from a simple approach as well as very little need for calibration. The third method provides the most accurate method of the three for the position of the redundant elbow although it also fails in some cases. This method however requires a long calibration session for each user. The last two methods allow for the calibration data to be used in latter session, thus reducing considerably the calibration required.

In combination with a virtual reality system, these processes allow for the real-time animation of an articulated figure to drive avatars in virtual environments or for low quality animation on a low-end computer.

Acknowledgements

This work would not have been possible without the support of the staff from the Department of Computer Science at Rhodes University.

Thanks should also go to my family for supporting me all along.

I acknowledge the guidance received from my supervisors Professor Shaun Bangay and George Wells.

On top of being one of my supervisors, Professor Shaun Bangay was also a 'catalyst' when helping me with the design and implementation of the system, as well as providing numerous ideas and constructive feedback. My deepest gratitude and admiration go to him.

Professor Bangay also leads the virtual reality special interest group (VRSIG) at Rhodes University. As such he also qualifies for my gratitude as a friend, as do all the members of the VRSIG, for their contributions to the system.

Special thanks should go to my group of proofreaders, including Jody Balarin, Shaun Bangay, Austin Poulton, and Steri Panagou.

Last, but not least, to Daemon, Lurka, Samba, Yoda and the other guys, for the endless hours of Quake and Starcraft that made the time fly along so quickly.

Contents

ABSTRACT II

ACKNOWLEDGEMENTS III

CHAPTER 1. INTRODUCTION 1

1.1 ARTICULATED FIGURE ANIMATION 1
1.2 MOTION CAPTURE 1
1.3 AVATARS AND VIRTUAL ENVIRONMENTS 2
1.4 INVERSE KINEMATICS, REDUNDANCY AND CONSTRAINTS 3
1.5 ORGANISATION 4

CHAPTER 2. APPROACHES TO FIGURE ANIMATION 5

2.1 TYPES OF 3D ANIMATION 5
2.1.1 Key Frame systems 5
2.1.2 Parametric systems 7
2.1.3 Scripting systems 7
2.2 ARTICULATED FIGURE ANIMATION 8
2.2.1 Articulated figures 9
2.2.2 Robotics terminology 10
2.2.3 Problems with articulated figure animation 13
2.3 MOTION CAPTURE 14
2.3.1 Types of motion capture equipment. 14
2.3.2 Most Commonly Used Motion Capture Equipment 27
2.3.3 Conformance to our project requirements 28
2.3.4 Summary 29
2.4 INVERSE KINEMATICS 30
2.4.1 Key frame and interpolation 30
2.4.2 Dynamics 32
2.4.3 Kinematics 34
2.5 GENERAL METHODOLOGIES 37
2.5.1 Without motion capture equipment 38
2.5.2 With motion capture equipment 38
2.5.3 Jack 39
2.6 SUMMARY 40

CHAPTER 3. ISSUES IN MINIMAL MOTION CAPTURE 41

3.1 EQUIPMENT 41
3.1.2 Virtual reality (VR) system 43
3.2 PROTOTYPE 44
3.2.1 Human articulated skeleton 44
3.2.2 Redundancy in the articulated skeleton 47
3.2.3 Constraints for the initial prototype 48
3.3 ISSUES RAISED FOR THE PROTOTYPE 49
3.3.1 Articulated figure animation 50
3.3.2 Motion Capture 51
3.3.3 Inverse kinematics 51
3.4 SUMMARY 52

CHAPTER 4. DESIGN 54

4.1 ARTICULATED FIGURE ANIMATION 54

4.2 MOTION CAPTURE 55
4.2.1 Trackers suitability *55*
4.2.2 Trackers placement *56*
4.3 INVERSE KINEMATICS 57
4.3.1 Specifying the inverse kinematics chain *57*
4.3.2 Reducing the redundancy for various articulated chains *60*
4.3.3 The sphere intersection and the arm *74*
4.3.4 Various methods of solving the articulated chain *76*
4.4 SUMMARY 80

CHAPTER 5. IMPLEMENTATION 81

5.1 ARTICULATED FIGURE ANIMATION 81
5.1.1 RhoVeR and CoRgi *81*
5.1.2 Need for interpolation *82*
5.2 MOTION CAPTURE 82
5.2.1 Attaching the trackers to the body *82*
5.2.2 Difficulty in achieving the proper position for the trackers *83*
5.3 INVERSE KINEMATICS 84
5.3.1 Eliminating a DOF *84*
5.3.2 Hand elevation *88*
5.3.3 State space *89*
5.4 SUMMARY 90

CHAPTER 6. RESULTS 91

6.1 ARTICULATED FIGURE ANIMATION 91
6.2 MOTION CAPTURE 91
6.3 INVERSE KINEMATICS 92
6.3.1 Eliminating a DOF *92*
6.3.2 Hand elevation *95*
6.3.3 State space *102*
6.3.4 A comparison of the three inverse kinematics methods used *107*
6.4 SUMMARY 112

CHAPTER 7. CONCLUSION 113

7.1 SUMMARY 113
7.2 ACCOMPLISHMENT 115
7.3 FUTURE WORK 116

BIBLIOGRAPHY 118

APPENDIX A

APPENDIX B

APPENDIX C

APPENDIX D

List of Figures

Figure 2.1 Spline interpolation between key frame position data over time.....6
Figure 2.2 A robotic manipulator with three degrees of freedom 9
Figure 2.3 DH notation 12
Figure 2.4 a) actual wrist topology b) DH wrist topology.....13
Figure 2.5 Hand held laser scanner.....16
Figure 2.6 Image based motion capture system for face and body motion capture...18
Figure 2.7 Shadowing of the trackers.....19
Figure 2.8 Left: the information available to the computer. Right: the information available to the user.....20
Figure 2.9 High definition infrared camera.....21
Figure 2.10 An exo-skeletal mechanical motion capture equipment.....23
Figure 2.11 A sequence of key and interpolated frames.....32
Figure 2.12 A two link articulated manipulator in two dimensions.....34
Figure 3.1 InsideTrak board, transmitter and receiver.....42
Figure 3.2 Anatomical representation of the human upper body 45
Figure 3.3 Representation of the human upper body articulated figure as commonly used by articulated figure animators.....45
Figure 3.4 Representation of the human upper body as seen in figure 3.3 but without the clavicle joint.....47
Figure 3.5 a) The rotation of the neck can not be defined by the head and chest links alone. b) Using simple constraints, one can find a value for α based on the value for β 48
Figure 3.6 The articulated figure animation in action 49
Figure 4.1 Twist joint and an equivalent joint in a human skeleton 58
Figure 4.2 Flexion joint and an equivalent joint in a human skeleton 59
Figure 4.3 Spherical joint and an equivalent joint in a human skeleton 59
Figure 4.4 The one link chain 61
Figure 4.5 The two links chain 61

Figure 4.6 The three links chain 64
Figure 4.7 Articulated chain comprising four non-zero length links 67
Figure 4.8 Intersection of the two spheres in a circle 75
Figure 4.9 Intersection of the two spheres 75
Figure 4.10 DOF eliminated from the wrist joint 77
Figure 4.11 Plane intersecting the elbow circle 78
Figure 5.1 Commercially available equipment to achieve a secure and comfortable fit of the motion capture trackers to the body 83
Figure 6.1 Two different types of intersections. Notice the difference in the length of the offsets 94
Figure 6.2 Elevation of the hand (in red) and of the elbow (in blue) over time 96
Figure 6.3 Calculated elbow elevation (in blue) vs. real elbow elevation (in red) ... 99
Figure 6.4 The user's avatar using the "hand elevation" method 101
Figure 6.5 Real position of the elbow in the x-axis (in red) vs. calculated position of the elbow in the x-axis (in blue) from time 3581 to time 3683 103

List of Tables

Table 2.1 Comparison of the various motion capture equipment types discussed.....29

Table 4.1 Summary of the various types of articulated chains discussed and the possible position of the undetermined joints.....74

Chapter 1. Introduction

God created man in his own image [The Bible, Genesis 1:26]. As such, we usually like to create our representation in three dimensional (3D) virtual worlds according to our own image, or at least based on an articulated figure of some kind. We propose a system to animate an articulated figure in real-time, using real-time data from minimal motion capture equipment, with the intent of using it to drive avatars in 3D virtual environments. Emphasis is placed on a simple yet effective solution that is computationally cheap. Various methods are investigated to solve the inverse kinematics problem in redundant articulated systems as well as to reduce the redundancy of such systems.

1.1 Articulated figure animation

The field of computer graphics has evolved tremendously in the last twenty years. While creating images of astonishing realism is a relatively common and easy task (although computationally expensive), other fields of computer graphics have not had such success. One of such fields is articulated figure animation [Badler, 1987]. Although some companies (like Pixar [Pixar Studios], or Mainframe [Mainframe Entertainment]) manage to do some impressive articulated figure animation, the process is still painfully slow, requires specialised software and hardware, and the results are still far from being realistically accurate (specially with human articulated figures).

1.2 Motion Capture

To solve some of these drawbacks, animation companies turn to motion capture. This provides them with accurate information defining the motion required to animate the articulated figure. As such, the animator does not need to specify the movements of the

articulated figure anymore. However, motion capture equipment is usually extremely expensive and thus limited to large companies that can afford such prices.

1.3 Avatars and virtual environments

Another use for articulated figure animation is in 3D virtual environments. In such interactive environments the user likes to represent himself by using an avatar. In the Hindu mythology, an avatar is the incarnation of a God on Earth. Likewise, 3D avatars are the user's 'incarnation' in cyberspace. The term avatar is used to refer to "any graphical representation of a human in a multi-user computer generated environment" [Vilhjálmsson, 1996]. If there is a need to interact in such 3D environments, the avatar chosen usually resembles the topology of an articulated human figure. This allows the users to interact with, and understand the other users' interaction, in an intuitive manner.

These virtual environments are usually run on low-end computers by common people who like to interact with other people through the Internet. As such, these people can not afford the use of costly motion capture equipment.

Another area where articulated figures are used is in the computer games industry. 'Shoot them up' games like Quake [Id Software, 1996] or Unreal [Unreal, 1998] require the user to use both keyboard and mouse. The mouse is usually used to move the head of the user's avatar (and hence move what the user sees and how the user aims), while the keyboard is used to drive the avatar around. A head tracker coupled to a head mounted display would eliminate the need for the mouse, since the user can now move his head around and see the corresponding part of the 3D world in the head mounted display. In the future, with the advent of cheap low level motion capture equipment, the user will be able to use tracking equipment to track not just the position of his head, but also his hands. In this way, the user will have more control over his avatar.

In the last two years, extraordinary advances have been made in the production of very low cost 3D accelerated video hardware, and companies like Intel, 3Dfx, nVidia, and S3 are already preparing a third generation of home-user targeted 3D accelerated video hardware. Similarly, processor manufacturers like Intel and AMD are providing 3D accelerated instructions in their new processors Katmai [Intel Katmai] and K6-2 with 3Dnow respectively [AMD 3DNow].

All this drive in the 3D field will increase the number of virtual environments and virtual reality systems. These systems will need to be populated by avatars. These avatars will need to be driven by some sort of input devices. It is hoped by the author that this drive will reduce the prices of low-end motion capture equipment to an acceptable level for the home computer consumer.

1.4 Inverse kinematics, redundancy and constraints

Articulated figure animation, animation of an avatar in virtual environments and animation of an avatar in computer games all require an articulated figure to be animated, preferably in real-time, and preferably using low cost equipment. To animate articulated figures, two main methods can be used: dynamics and kinematics. We investigate the use of kinematics, and more precisely, inverse kinematics to animate an articulated figure. We also limit the number of motion capture devices used to only four electromagnetic trackers so as to keep the price of the system as low as possible. As such, redundancy in the articulated figure tends to be rather large. We discuss various methods for reducing such redundancy by applying constraints to the system. Finally, we implement a system to animate an articulated human upper body in real-time using only four motion capture trackers.

1.5 Organisation

- In the next chapter, we will provide an overview of the various fields covered in this project. These include the various types of articulated figure animation, articulated figure notations, the different types of motion capture equipment available, kinematics and dynamics, and related terms.

- In chapter three we describe an early prototype implementation to identify real-time issues for the motion capture equipment and issues of redundancy in the inverse kinematics system.

- Chapter four describes our proposed design for reducing redundancy in various kinematics systems, and specifically for the human upper body.

- In chapter five we discuss our implementation of the designs from chapter four.

- In chapter six we discuss the results of this implementation.

- Finally in chapter seven we discuss the contributions and conclusions of our work as well as some of the pitfalls uncovered during this project and ways to address them.

Chapter 2. Approaches to figure animation

In the seventies, much effort was put into programming computers to produce realistic images [Badler, 1987]. Since then, computers have been used to create realistic animations. These animations range from scientific visualisation of simulations to full motion films for the entertainment and advertising industry [Watt and Watt, 1992]. Computers were first used to replace the traditional two dimensional (2D) animation methods, but the computers also allowed for a fast and cheap way to create 3D animations. Currently, computer graphics imagery (CGI) is used in most animation studios, be it for the creation of traditional cartoon animation (like Disney Studios) or in 3D special effects for the latest Hollywood blockbuster motion picture. However, animation in 3D of articulated figures has been avoided by most of the CGI studios due to the difficulty of achieving an animation that is realistic.

2.1 Types of 3D animation

There are various approaches for doing 3D animation, all of which are applicable to the animation of characters. As such, we need to decide on a suitable animation system for our requirements. The most common ones are the key frame systems, the parametric systems, and the scripting systems.

2.1.1 Key Frame systems

The most popular approach to 3D animation is the three dimensional key frame system. It is based on the traditional key frame approach used in 2D animation. In the traditional 2D animation key frame approach, the chief animator specifies certain animation key frames (hence the name), and then his subordinates create the in-between frames. In the 3D key

frame animation system, position and orientation of bodies are specified by the animator for certain key frames only. The system then evaluates and interpolates the in-between frames [Watt, 1989].

As an example, consider figure 2.1. In this figure, the position for a certain number of key frames are given at certain points in time, and the position for the in-between key frames are then calculated using splines.

Key frame number	*Time (s)*	*Position*
1	0.0	7
2	1.0	8
3	2.0	6
4	4.0	2
5	8.0	0

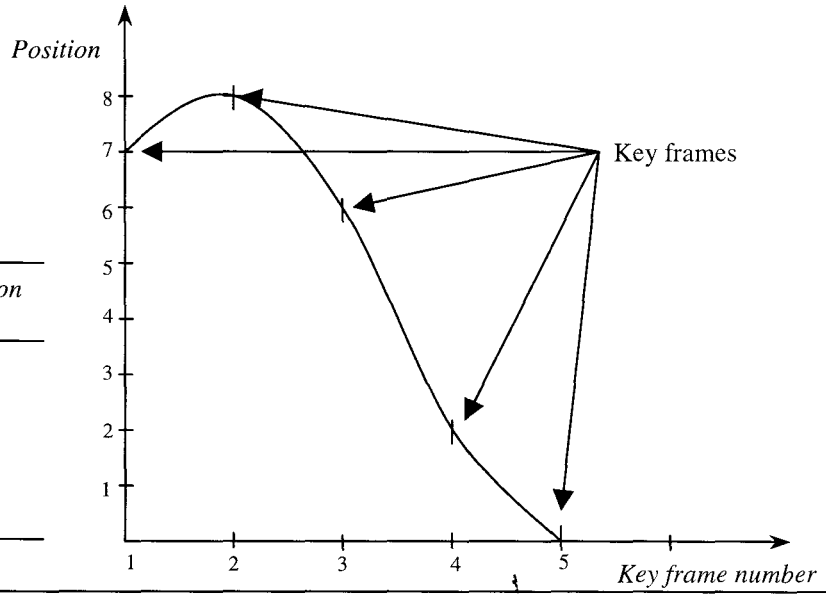


Figure 2.1 Spline interpolation between key frame position data over time.

As a result, the spline can then be sliced in an arbitrary number of in-between points as required for each key frame.

This method requires a precise specification of the position and orientation of the animated bodies, and thus only allows for a low level of abstraction [Watt, 1989].

2.1.2 Parametric systems

Parametric systems are derived from the 3D key frame system, but in this case parameters are specified to control the behavior of the object, instead of just controlling its position, in the in-between frames. Parametric systems ease the creation of more complex tasks.

An example of a parametric system is *em* [Hanrahan and Sturman, 1985]. Hanrahan uses parameters that are modified by input data from an external input device (like a joystick or mouse). For example:

$Spin \mathrel{+}= 10 * joyz$

Here the *spin* parameter is modified by the input from a joystick. These parameters can be altered in real time thus interactively changing the animation. These parameters do not have to be altered specifically using external control. They can also be altered by algorithms previously defined by the animator. In this way, the animator has a greater degree of control on the animation. This method however still lacks a high level of abstraction.

2.1.3 Scripting systems

Scripting systems use a standard high level programming language or a specially designed script to create the animation. They allow for greater level of abstraction, but have some drawbacks in that the animation can only be seen after the program or script has been written [Watt, 1989] and thus can not be used for real-time animation. Scripting systems also require the animator to be a programmer.

Examples of such scripting systems in articulated figure animation include the early Labanotation [Calvert and Chapman, 1982], which specified the position and orientation

of every joint in a script. Zeltzer [Zeltzer, 1982] used high level controls, allowing the user to specify a high level of abstraction commands (such as "walk to the door") to drive the animation. Another example of scripting systems is found in most 'shoot them up' computer games such as Quake and Quake II [Id Software]. In these games each actor has a set of scripted animations (like 'shoot', 'run', 'salute') which are performed according to the input from the user or the environment.

2.2 Articulated Figure Animation

When animating an articulated figure, one would like to have a figure that is as realistic as possible. One would like muscles and skin that deform with the various poses of the figure, clothes that will adapt realistically to the various movements of the figure, hair that flows realistically as the air passes through it, and sharp and accurate facial features that change with the various expressions of the figure. All that on top of a realistic movement of the body. Unfortunately, these attributes are amongst the hardest problems to solve in 3D animation and are, as such, research topics of their own, and well beyond the scope of our project.

We will concentrate on animating an articulated skeleton of the figure. The muscles, skin, hair and clothes can then be layered on top of the animated skeleton.

Although much of the research covered in this thesis can be applied to any type of articulated figure, we use a human figure for our implementation and explanations. As such, some material is specifically aimed at human articulated figures.

2.2.1 Articulated figures

An articulated figure is composed of rigid links and joints. These various links and joints mimic the skeleton of the figure we want to animate. The links represent the various bones, while the joints connect the links together. An articulated figure is made up of chains of joints and links.

In this respect, an articulated figure is very much like a robotic manipulator. As such, much of the robotics literature is applicable in the field of articulated figure animation. Just like in the robotics field, the joints in an articulated figure can be of two types, revolute and prismatic, as shown in figure 2.2. A revolute joint is a joint that revolves around a point. A prismatic joint is a joint that slides around an axis (like a chameleon's tongue or E.T.'s neck). As such, the length of a revolute joint is zero whereas the length of a prismatic joint is variable.

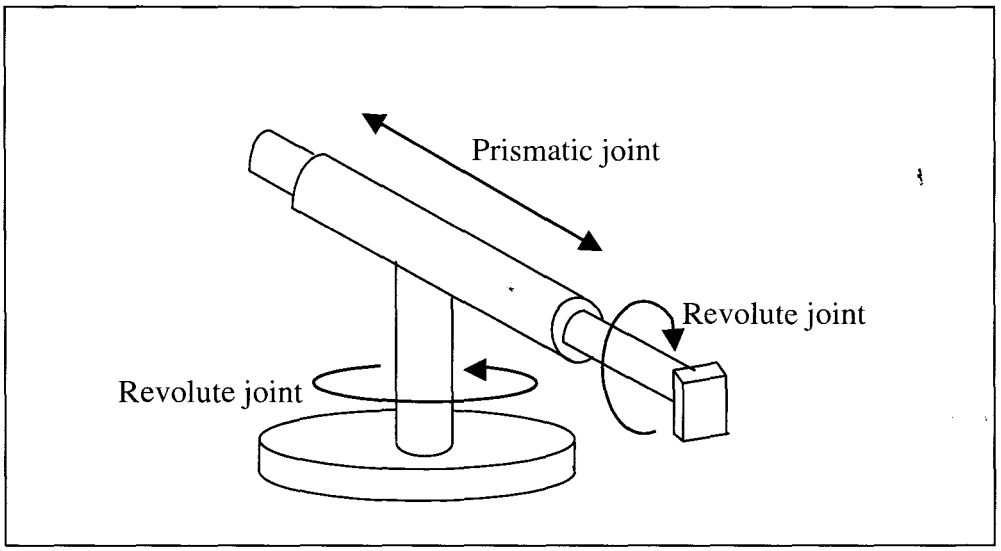


Figure 2.2 A robotic manipulator with three degrees of freedom.

However, prismatic joints are not common in articulated figure animation, and the human articulated figure does not contain prismatic joints. Prismatic joints are more common in robotics, inherited from telescopic machinery equipment (like in figure 2.2).

2.2.2 Robotics terminology

The field of robotics has been around for longer than the field of articulated figure animation. Since these two fields are very similar, most of the terms and methods in robotics have been adopted by the articulated figure animation research community.

In robotics, as well as in articulated figure animation, the way in which a figure moves can be specified by two methods: kinematics and dynamics.

- Kinematics is the study of the motion of the object, regardless of the forces acting upon it. It is concerned with the position, velocity and acceleration of the object at a certain moment in time. Kinematics is further subdivided into forward and inverse kinematics. These two terms will be defined in this chapter.

- Dynamics is concerned with the forces acting on the object. Attributes of concern are the mass of the object, its inertia, the acceleration of the object (both linear and angular) and the forces acting on it.

These two methods are used, combined or not, to specify the movement of an articulated figure.

As mention earlier, an articulated figure is made up of a collection of chains interconnected with each other. Each chain is made up of the rigid links and joints defined earlier. Each independent chain can be considered as a robotic manipulator. In an articulated figure closed chains are very rare. As such, each chain has a beginning point and an end point. A chain starts at the beginning point and moves outwards. The starting

point of a chain is bounded, usually to another chain, and called the root of the chain. The end point of the chain is free and is called the end effector. The set of possible positions that the end effector can be in is called the reach of the articulated chain.

When a chain is in a certain position, we say that it has a certain pose. The more links and joints a chain has, the more different positions it can be in. This is because as the number of links and joints increase, the degrees of freedom (DOF) increase too. Watt [Watt and Watt, 1992] defines the number of DOF in an articulated chain as "the number of independent position variables necessary to specify the state of the articulated chain". The more DOF an articulated chain has, the harder it is to specify the position of the links in the chain due to the increase in the number of different positions available.

Now that we have defined what an articulated chain (or manipulator) is, we need to define a notation to represent it. The most common notation is the Denavitt-Hartenberg (DH) notation [Denavitt and Hartenberg, 1955].

The DH notation specifies the position and orientation of each link with respect to its neighbours by defining a reference coordinate frame to each link. This coordinate frame is specified by four variables. These variables are the length of the link, the distance between links, the twist between links, and the angle between links.

They are defined as follows (refer to figure 2.3) [Watt and Watt, 1992]:

- a_i (the length of the link) is the distance from z_i to z_{i+1}, measured along x_i.
- α_i (the twist of the link) is the angle between z_i and z_{i+1} measured about x_i.
- d_i (the distance between links) is the distance between the z_{i-1} and x_i axes measured along z_i.
- Θ_i (the angle between links) is the angle between x_{i-1} and x_i measured about z_i.

The z_i-axis of coordinate frame i lies along the axis of the joint. The x_i is normal to the axis and points towards *joint* $_{i+1}$. The origin of the i^{th} coordinate frame is where the common normal to z_i and z_{i+1} intersects z_i.

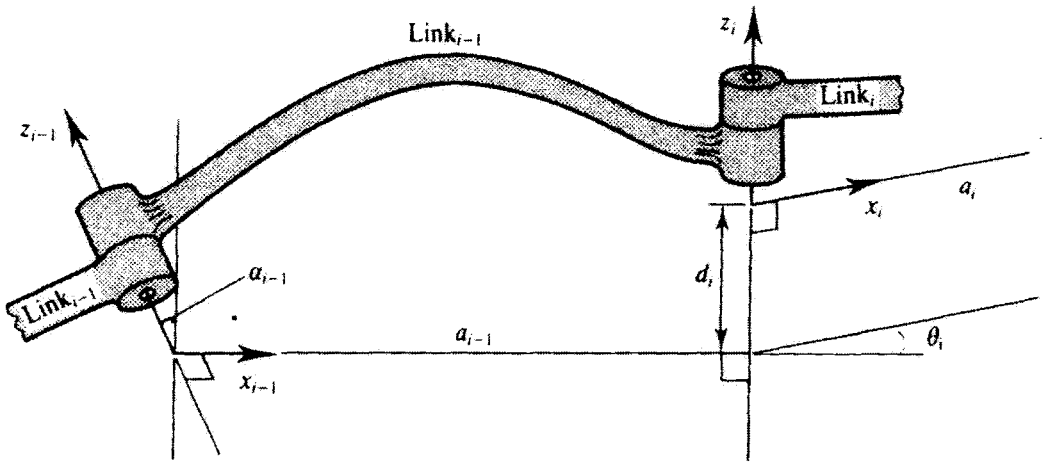


Figure 2.3 DH notation

A linear transformation matrix can then transform between the coordinate frames of adjacent links. These matrices can be combined to form a compound matrix defining the articulated chain.

Using the DH notation, each adjacent pair of links has only one DOF between them. If one wants to represent a ball type joint, one has to combine more than one link together, and assign a link length of zero to the intermediate links. As such, a link such as the human wrist with three DOF will consist of four links and three joints as seen in figure 2.4. The first and fourth link will represent the lower arm and hand, while the second and third links will have a length of zero.

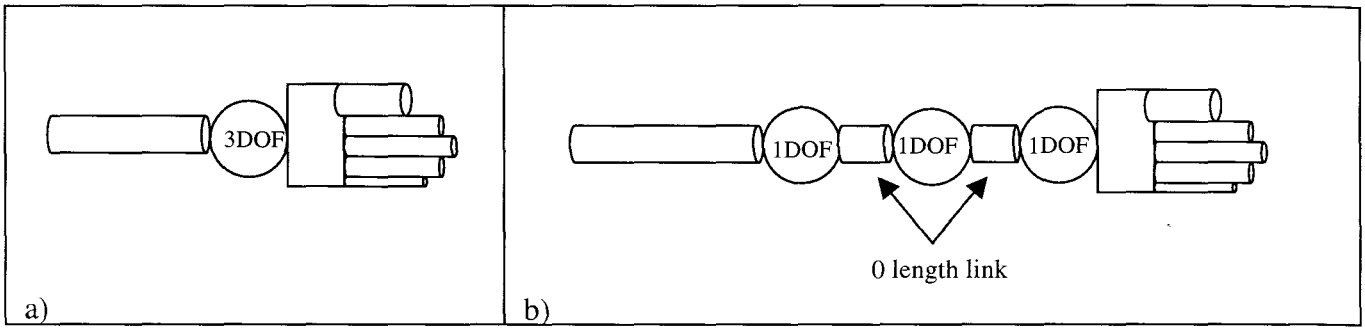


Figure 2.4. a) actual wrist topology. b) DH wrist topology

However, the DH notation can not be used to specify hierarchical tree structures. To specify the human articulated skeleton one must break it into different sub-chains that are only logically connected to each other. Also the DH notation is usually found to be somehow unintuitive [Watt and Watt, 1992] when working with articulated figures as explained in section 4.3.1.

Other alternatives include the axis position joint representation [Sims and Zeltzer, 1998], or a simple yet quite intuitive method used by Korein [Korein, 1985]. The axis position joint representation is able to specify hierarchical structures with multiple branching, although it now requires seven parameters, as opposed to four. It is also more intuitive than the DH notation [Watt and Watt, 1992]. Korein uses revolute and spherical joints [Burton, 1979], [Suh and Radcliffe, 1978], [Hunt, 1978] to specify three types of motion: a flexion motion, a twist motion, and a spherical motion. These motion types are explained in chapter 4.

2.2.3 Problems with articulated figure animation

Now that we have a notation to represent our articulated figure, the animator can specify the variables for each link, starting from the root link and going towards the end effector.

The problem with this is that if the articulated chain is composed of a large number of links and joints, the animator is required to specify a lot of variables and all of these variables will affect dramatically the position of the end effector. This problem is further explained in section 2.4.3.1.

Another problem is that to create an animation it is up to the animator to animate the figure. This means that the accuracy of the movement of the articulated figure depends on how good the animator is. To get around a 'bad' animator, one can use motion capture data to generate a correct and precise motion for the articulated figure. There are various types of motion capture equipment commercially available. The next section describes such equipment with emphasis on the requirements for our system.

2.3 Motion capture

2.3.1 Types of motion capture equipment.

There are various types of commercial and experimental motion capture equipment available. In this section, we will review some of this equipment, with reference to its advantages and limitations for articulated figure animation and specifically for this project. For this project, we require motion capture equipment that fulfils certain requirements. These requirements are:

1. The motion capture system needs to have a high update rate and low latency so that real-time motion capture is possible.

2. The equipment needs to have a relative low cost since the system is aimed at personal computer users.

3. The equipment does not need to have a large number of receivers. One of the aims of this project is to solve for the position and orientation of an articulated figure animation with input data from a limited number of data trackers.

4. The data from the trackers needs to be accurate to a certain degree, but extreme accuracy (such as less than one centimetre of position error) is not a strong prerequisite. As such, interference in electromagnetic equipment from foreign electromagnetic fields or ferromagnetic objects must be reduced. Also, measurement drifting (drifting in the position data as a result of small errors in previous data) must be limited if not avoided.

5. The equipment must not encumber the user to the point of being uncomfortable.

6. The equipment must lack the need for user specific calibration or such calibration must be quick and simple to perform.

Some motion capture systems can only provide a three DOF position value. In this case, using three or more of these three DOF devices can provide six DOF position and orientation in 3D space by arranging the three trackers in an orthogonal (L-shaped) position.

2.3.1.1 Optical systems

There are various ways of achieving optical tracking, each with a different method and equipment.

2.3.1.1.1 Position-Sensing Detector Systems

In this method, the object to be captured is tagged with photoelectric devices. These devices are sensitive to the amount of light received, and generate a current inversely proportional to the distance between the light source and the detecting device. The

position of the detecting devices is calculated using triangulation by comparing the various intensities at the various detectors.

2.3.1.1.2 Structured Light Systems

This method involves the use of a laser to illuminate the scene across a known pattern. The scene is then captured by a camera as the laser scans it with the object in it. A 3D coordinate system is created by the intersection of the camera plane and the laser light reflected by the surface of the object to be tracked [NCR, 1995].

2.3.1.1.3 Laser Radar

In this method, a laser plane is used to scan the surface of an object repetitively. The light reflected from the object is detected by a camera at an angle from the projected laser plane. The camera detects a curve that matches the contour of the figure scanned [Polhemus].

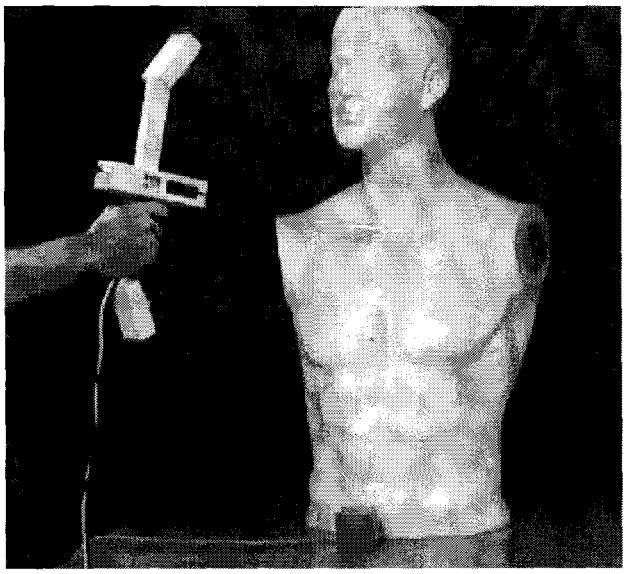


Figure 2.5 Hand held laser scanner

2.3.1.1.4 Laser Interferometry

This method uses a steered laser beam to track a retro-reflector mounted on the object tracked. The angle (in 2D) and the time of flight of the laser light will determine the 3D position of the object. Several lasers coupled with several retro-reflectors can be used to track multiple positions.

2.3.1.1.5 Advantages and disadvantages

Due to the use of lasers as the source of light, optical systems tend to be extremely accurate. However, they can not track a large number of objects simultaneously. Some of these systems also suffer for shadowing problems since the user can obstruct the path of the light to a specific reflector. Shadowing (or occlusion) problems occur when one of the trackers is obscured from the receiving camera by the user thus causing the loss of data from the tracker.

These disadvantages mean that optical systems are insufficient for real-time tracking of human body [Frey *et. al.* 1996].

2.3.1.2 Image-based systems

In these systems, the object is tagged with targets that are visible to the tracking system. Using multiple cameras, the position of the targets can be determined in 3D by triangulation. In theory, only two cameras are necessary but usually more cameras are used to increase the redundancy of the system and to avoid shadowing.
Typically infrared light emitting diodes are used as the targets. These diodes are the only objects visible to system (as seen in figure 2.6), so the system does not get readings from false trackers. These markers can be very small and are thus suitable for articulated figure animation as well as for face recognition (as seen in figure 2.6).

Figure 2.6. Image based motion capture system for face and body motion capture.

Usually the trackers are placed on the body and the cameras in the ground (but the opposite is possible, although less practical).

2.3.1.2.1 Advantages and disadvantages

Image based systems benefit from having a large area of operation in which the subject can move. Also, since there are no cables involved, the user can move unencumbered. Since the price of passive markers is very low, using passive markers will not increase the price of the system while allowing for a large number of trackers to be used. Thanks to high speed cameras, high sampling rates can be achieved.

Image based systems suffer from shadowing or occlusion problems (as seen in figure 2.7). Although this can somehow be alleviated by using more cameras, it can not be totally eliminated.

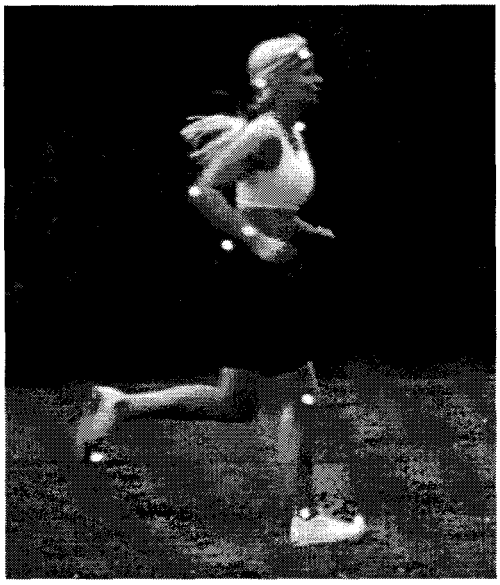


Figure 2.7. Shadowing of the trackers. One can clearly see that some of the left hand side trackers are not visible to this camera and indeed one can not tell if the right arm is obscuring a tracker on the right hand side of the body.

The system may be susceptible to changes in ambient illumination. Also reflections can cause false marker readings.

Determining the correspondence of targets in each camera view can be a problem and is at best computationally expensive, on top of the computational complexity of processing images from multiple sources. As seen in figure 2.8, the left hand side shows the data available to the computer while the right hand side show the data available to the user. Usually the easiest process is for the user to relate the point in the left hand side section to their corresponding positions in the articulated figure animation using the right hand side section as a guide.

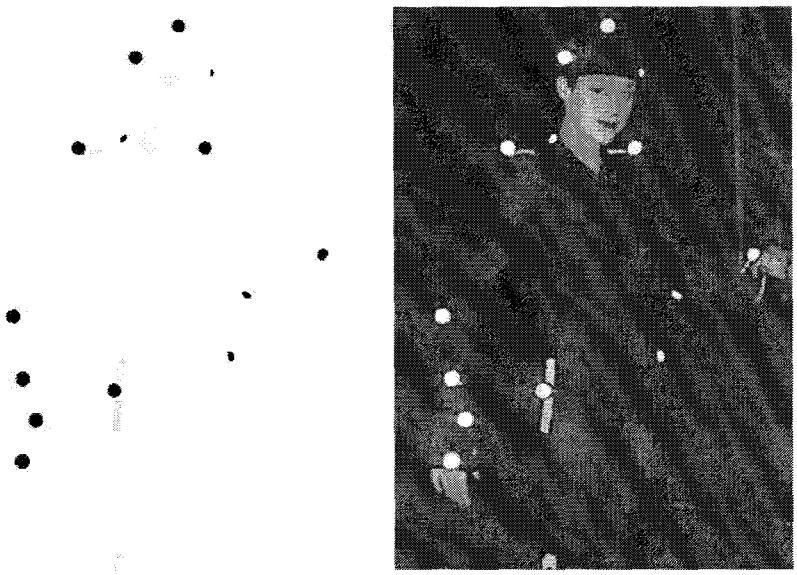


Figure 2.8. Left: the information available to the computer. Right: the information available to the user.

Image based systems are therefore not suited for real-time motion capture. Also, usually only 3D position can be found. Some of the equipment used in image based motion capture can be rather expensive (like high definition infrared cameras as seen in figure 2.9) making it one of the most expensive body tracking solutions available.
Even though image based motion capture is not suited for real-time body tracking, it could be used for gesture recognition with a greater degree of success.

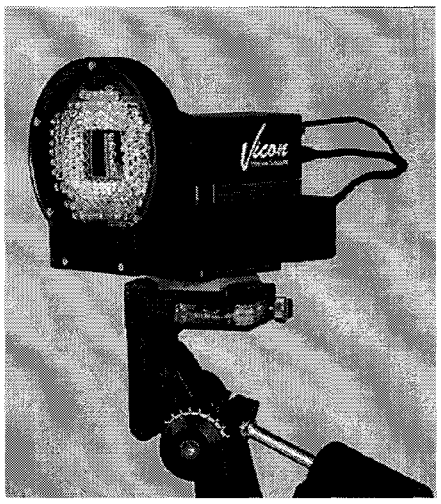


Figure 2.9 High definition infrared camera.

2.3.1.3 Electromagnetic systems

Electromagnetic tracking systems are the most common motion capture devices used for human body tracking, along with image based tracking systems. The main reason is that they are fairly inexpensive when compared with other motion capture systems.

They use artificially generated electromagnetic fields which induce voltages in the receivers which are attached to the body. Three orthogonal electromagnetic fields are generated by the transmitter. These induce a current in three orthogonal detector coils which is proportional to the spatial position and orientation of the transmitter [Frey *et. al.* 1996].

2.3.1.3.1 Advantages and disadvantages

As stated earlier, electromagnetic tracking systems are fairly inexpensive when compared with other motion capture systems. They allow the tracking of numerous objects within an acceptable accuracy and range. They benefit for the lack of shadowing effects, as

opposed to image based systems. They also benefit from a low latency and a high update rate, making them suitable for real-time motion capture.

However, electromagnetic systems are sensitive to interference from background magnetic fields and from metallic objects. They also have limited range, with a decrease in accuracy as the distance between the transmitter and the receiver increases.
Also, most of the equipment available requires wires connecting the receivers and the transmitters together.
Electromagnetic systems uses either AC or DC magnetic fields, with the DC magnetic fields being less sensitive to surrounding metallic objects [NCR, 1995].

2.3.1.4 Mechanical systems

Mechanical tracking devices are subdivided into body based and ground based systems.

2.3.1.4.1 Body based mechanical devices

Body based (or exo-skeletal) systems are those mounted on, or carried on the body of the subject. They are used to determine relative positions of various parts of the subject's body, or to measure the position of an object relative to a fixed point on the body of the user. Exo-skeletal systems are usually used to measure the angles in the joints between links, or to determine the position of an end-effector with respect to a fixed point in the body of the figure. An example of a body based mechanical device is illustrated in figure 2.5.

Figure 2.10 An exo-skeletal mechanical motion capture equipment

2.3.1.4.2 Ground based mechanical devices

Ground based systems are mounted on a fixed surface and not carried by the user. They are used to determine the position of an object with respect to that fixed surface. They usually provide 6 DOF information about an object manipulated by the user, relative to some fixed position different from the user's body.

Unlike body based systems, ground based systems are limited to operations in the area delimited by the maximum reach of the device, whereas body based systems move along with the user.

2.3.1.4.3 Advantages and disadvantages

Mechanical based systems are usually quite obtrusive and encumbering. This is not much of a problem when only few of these systems are used, but when fully mapping an articulated figure, the weight and encumbrance of the system is prohibitive. Problems also arise if the system is to be used for multiple users since the system would then require extensive recalibration to adapt it to the new user. Additionally, any displacement between the device and the part of the body that the device is attached to needs to be minimised since such displacement will not be recorded although the motion does occurs. Some errors can also occur due to the misalignment between the joints of the device and the joints of the body part it is supposed to measure, since the centres of rotation are different.

On the other hand, mechanical systems are usually quite accurate and have a great degree of precision. They usually benefit from high update rates and low latency [NCR, 1995], making them very attractive for real-time systems. They also lack the drawbacks of other systems like measurement drifting, electromagnetic interference or shadowing problems.

Some mechanical systems can also be used as force-feedback devices, having not only the function of data capturing devices, but also providing some haptic response to the user.

2.3.1.5 Acoustic systems

Acoustic tracking systems use high frequency sound waves to determine the position of objects. Two methods are used.

2.3.1.5.1 Time of flight method

This uses the speed of sound to calculate the time difference between the transmission of the high frequency sound wave and its reception. This time can be used to determine the distance between the transmitter and receiver. With at least three receivers, one can calculate, by triangulation, the position in 3D space of the object tracked.

2.3.1.5.2 Phase difference

In the phase difference (or phase coherence) method, the system senses the signal phase difference between the signal sent by the transmitter and the signal that the receiver obtains.

This method is far more precise than the time of flight method. However, it suffers from various problems: the calculated position of the object tracked will be erroneous if the object moves farther than one-half of the signal wavelength during the period of one

update [Frey *et. al.* 1996]. Also, because the position is determined incrementally from the previous position, any slight error will be carried onto the next position as drift errors.

2.3.1.5.3 Advantages and disadvantages

The acoustic tracking methods suffer from interference from reflections of the acoustic signal as well as from ambient noise. They have a small range of action, although it is larger than that of electromagnetic systems. The main problem however is the need of a line of sight at all times, so the system suffers from shadowing problems.

2.3.1.6 Inertial systems

Although somewhat experimental, inertial systems are very promising.
They work by calculating the acceleration of the system and then getting the position by using the double integral of that acceleration. This technique has been around for some time, and has been tested in aeroplane, ship and submarine navigation. Up to now, the systems was too bulky and heavy to be used on a person, and the errors where too large for the system to be precise enough. In the last years, several companies have started to produce micro-machined accelerometers which could be used for our purpose [Frey *et. al.* 1996].

2.3.1.6.1 Advantages and disadvantages

The inertial tracking system has the potential to be the best method of body tracking in the near future [Frey *et. al.* 1996]. It benefits from the lack of signal interference, metallic object interference or shadowing. It also benefits from the lack of wires connecting the receivers to the transmitters like in most electromagnetic tracking equipment. It is independent of range although it suffers from drift problems due to accumulate measurement errors over time. It is however experimental and rather expensive.

2.3.1.7 Spread-Spectrum

Spread-spectrum systems have been in use for some time in the form of Global Positioning Systems (GPS), although they haven't been used to track small objects like the human body.

GPS uses a group of satellites orbiting around the earth which emit signals intersected by a navigational receiver. Using the time of flight method, one can determine the exact distance between satellite and receiver. Triangulation is used to calculate the spatial position of the receiver.

This system could be adapted to a smaller scale and used to track objects such as a human body by placing the transmitters in the corners of a room and using small receivers.

This system would benefit from long range tracking, as well as the lack of cables between transmitter and receiver.
However, there exists the potential for high frequency electromagnetic radiation exposure to the user, and the price tag is rather high.
This is nevertheless an experimental system that should improve over time.

2.3.1.8 Digital Poseable Mannequins

This is a poseable mannequin which has position encoders for each joints in the mannequin. As the mannequin is moved, the values at the joints are recorded and passed onto a controlling computer. The mannequins are usually sub-scale, and are not limited to the human body but full size mannequins are available.

Although the digital poseable mannequin benefits from low latency and a high update rate, the fact that the animator has to adjust every limb individually means that this is not a real-time device.

The digital poseable mannequin is used in a stop-motion animation fashion, and is thus familiar to such users. The low price also makes this system quite attractive for non real-time articulated figure animation.

2.3.2 Most Commonly Used Motion Capture Equipment

The most commonly used motion capture equipment depends on the needs of the user, but is broadly divided into two sections: optical and electromagnetic equipment.

2.3.2.1 Optic equipment

Optic equipment is mostly used by animation or entertainment industries to capture the motion of humans. These industries have no need for real-time data, since the captured data is later going to be processed for the creation of animations. As such, Optic equipment gives them the benefit of being able to track a large amount of points in the surface of the object. This eliminates the need for six DOF data since a large amount of data points will provide sufficient information for the articulated figure to be recreated. Since there is no real-time requirement, the captured images can be processed individually by a human person who will relate each marker in the image, to their corresponding position in the articulated figure.

2.3.2.2 Electromagnetic equipment

When real-time data is required, electromagnetic motion capture equipment is the system most commonly used. It provides real-time data with six DOF and a high update rate. This system is used in virtual environments, and when real-time animation is required. It benefits from a low cost and an acceptable level of accuracy.

2.3.3 Conformance to our project requirements

The various types of motion capture equipment described above possess some of the requirements for our project as stated in 2.3.1 to a certain degree. We display those requirements in relation to each type of motion capture equipment in table 2.1. Each equipment type is rated with a number of stars from ***** to * according to how well they fit our requirements in relation to the other types of equipment (***** means very well while * means not suitable at all). This table is based on [Frey *et. al.* 1996] and [Frey *et. al.*supp, 1996].

The requirements from section 2.3.1 are repeated here:

1. The motion capture system needs to have a high update rate and low latency so that real-time motion capture is possible.

2. It needs to have a relative low cost since the system is aimed at personal computer users.

3. It does not need to have a large number of receivers. One of the aims of this project is to solve for the position and orientation of an articulated figure animation with input data from a limited number of data trackers.

4. The data from the trackers needs to be accurate to a certain degree, but extreme accuracy (such as less than one centimetre of position error) is not a strong prerequisite. As such, interference in electromagnetic equipment from foreign electromagnetic fields or ferromagnetic objects must be reduced. Also, measurement drifting (drifting in the position data as a result of small errors in previous data) must be limited if not avoided.

5. The equipment must not encumber the user to the point of being uncomfortable.

6. The equipment must lack the need for user specific calibration or such calibration must be quick and simple to perform.

Requirement (see above)	1	2	3	4	5	6
Mechanical	*****	****	***	****	*	****
Electromagnetic	****	*****	****	***	****	****
Acoustic	**	***	**	*	**	**
Optical	*	**	*	*	**	***
Image based	*	**	*	*	*****	***
Inertial	****	*	****	****	****	****
Spread spectrum	****	*	***	****	***	***
Digital poseable mannequin	*****	****	**	*****	**	****

Table 2.1 Comparison of the various motion capture equipment types discussed.

2.3.4 Summary

Most of the motion capture equipment reviewed falls in one of two categories: they can either provide real-time data, but can not provide enough sensors; or if they can provide enough sensors, real-time animation can not be achieved.

The electromagnetic motion capture system provides real-time data with a low latency and high update rates. Also it is a relatively cheap motion capture system when compared to other motion capture systems with similar characteristics.
The electromagnetic motion capture system appears to be quite suitable for our purpose.

2.4 Inverse Kinematics

As mentioned in section 2.2.2, a more in depth explanation of kinematics, dynamics and related terms is given in this section.

To animate an articulated figure composed of various chains of links and joints, one varies over time the local rotations (and translations if prismatic joints are used) at each joint. One also needs to apply a global translation to the root joint of each articulated chain.

To specify and control these rotations and translations, two fundamental approaches are available: kinematics and dynamics.

2.4.1 Key frame and interpolation

Using kinematics to specify the motion of an articulated figure involves setting the position and orientation of each link in the figure at specific intervals. Using dynamics to specify the same motion involves calculating the position and orientation at which each link in the figure will be according to the forces applied to the links. This calculation is done at certain intervals since the calculation of the position of the actual position of the articulated figure requires the previous position.

Either using kinematics or dynamics, one finds oneself with data about the articulated figure at certain time intervals.

This fits nicely with the key frame approach in 3D computer animation in that both approaches involve the availability of position (and orientation) data at certain intervals in time.

The intervals at which the position and orientation of the joints are specified can be set according to a certain time frame, or according to the bursts of new data for the joints. One can set a specific time length for the interval. Alternatively, the interval can be specified by the arrival of new position and orientation data.

If the intervals are not short enough for the animation to be realistic, one might need to interpolate the position and orientation for the in-between frames so as to generate a fluent movement of the articulated figure.

For a virtual environment, at least ten frames per second are necessary to provide the sense of motion although a higher frame rate would be welcome. For professional animation, a frame rate of thirty frames per second is necessary to achieve the desired quality.

Key frame interpolation methods have been available in computer animation for quite some time and one can refer to [Gomez, 1985], [Hanrahan and Sturman, 1985], [Shoemake, 1985], [Stern, 1983], [Sturman, 1987] for further reference.

The use of interpolation implies a delay of at least one frame. Even if the interpolation process in instantaneous, one still needs to have the data for the end (second) frame in a pair of consecutive frames, before any interpolation can be calculated. This implies that there always is a one frame delay as shown in figure 2.11, unless predictive filters are used. If predictive filters are used, one avoids the one frame delay but increases the computational cost of the process since now the expected position (calculated by the predictive filters needs to be computed), as well as decreasing the accuracy of the system, since the position of the not yet known key position is estimated.

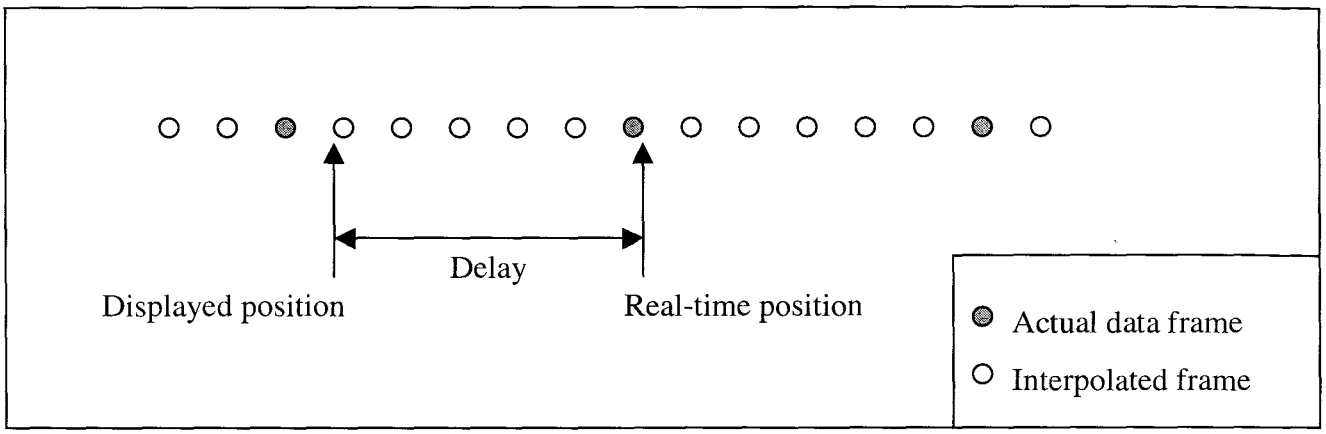


Figure 2.11 A sequence of key and interpolated frames

If the rate at which the data for the animation is supplied is fast enough, there might not be a need to interpolate between frames.

2.4.2 Dynamics

Dynamics uses the mass of the articulated figure, and the forces and torque acting on that figure to determine the values for each link and joint in the figure. Since a real figure moves according to the forces applied on it, dynamics tends to give a more realistic result than kinematics [Armstrong *et. al.*, 1985], provided that a proper method is used, and that the data used is accurate.

One of the drawbacks of dynamics is that since the mass of each link is usually required, the system needs extensive calibration for different users. Another drawback is the difficulty in measuring forces applied to the articulated figure using available motion capture equipment. There are two ways of measuring forces in dynamics, depending on what type of motion capture equipment is used.

2.4.2.1 Measuring forces directly

Using mechanical motion capture equipment, one can measure forces and torque applied to the figure. On top of the drawbacks of mechanical motion capture, one finds that some forces or torque can not be measured accurately, if at all.

Also one needs a lot of information due to the large amount of forces needed to accurately animate the articulated body. This means a lot of motion capture equipment, increasing the problems with mechanical motion capture equipment (notably weight and encumbrance).

2.4.2.2 Measuring acceleration

The other way to measure force is to use Newton's second law of motion which states that:

Force = Mass * Acceleration

If we know the mass of the articulated links, and we can get the acceleration, then we can calculate the force. If the motion capture system provides acceleration data (like the inertial motion capture equipment) then the process is straightforward. If the motion capture system does not provides acceleration data, but it provides position data, one can still calculate the acceleration in the system although one needs three position data points to get an acceleration value.

Although dynamics methods for articulated figure animation are often more realistic than kinematics methods, the need for the estimation of many forces and the fact that dynamics methods are computationally expensive [Sudhanshu *et. al.*, 1988] usually makes them a second choice over kinematics methods.

2.4.3 Kinematics

As stated in section 2.2.2, kinematics in further subdivided into forward and inverse kinematics.

2.4.3.1 Forward kinematics

Given the values of the angles for all the joints in an articulated chain, forward kinematics is the process that will return the position of the end effector of that chain. Consider figure 2.12 for an example.

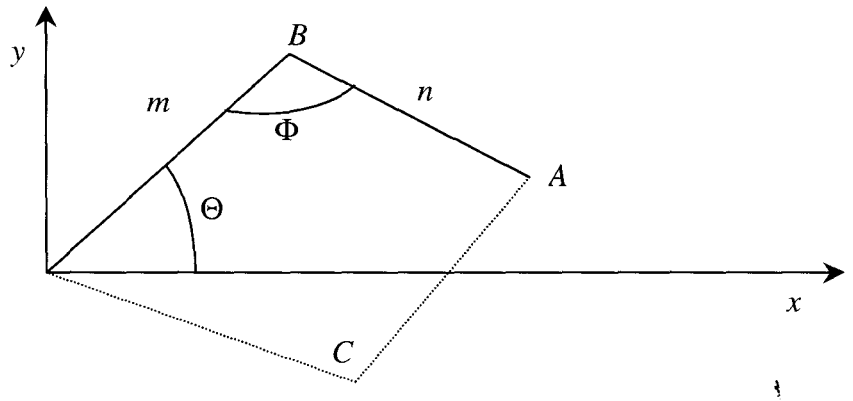


Figure 2.12. A two link articulated manipulator in two dimensions.

Given Θ and Φ, and with the lengths of the links m and n constant, one can easily calculate the x and y values for the point A, end effector of the articulated chain.

The position of A is given by the formula:

$$A = [m\cos\Theta + n\cos(\Theta + \Phi), m\sin\Theta + n\sin(\Theta + \Phi)]$$

Equation 1

This is the forward kinematics equation for the articulated chain.

When an animator wants to animate an articulated structure, he must specify the value of all the angles at all the joints. The position and orientation of the end effector is given by the accumulation of the transformations at each joint of the articulated chain. As the articulated chain increases in number of links and joints, the DOF of the chain increase dramatically, resulting in an increase in the complexity of the specification of the joint angles.

Assume that the animator wishes to animate a human figure and he wishes to get the hand (end effector of the arm articulated chain) in a certain position and orientation. To do that, he must specify all the angles at all the joints in the arm chain, starting from the chest of the figure and going outwards. A small change in one of the angles closer to the root of the chain can cause a massive change in the position of the end effector. The more DOF, the larger the change, hence the increase in complexity.

2.4.3.2 Inverse kinematics

Matters would be much simpler if the animator just had to specify the position and orientation of the end effector, and then the system would provide the angles at all the joints.

This is exactly what inverse kinematics does. Given the position and orientation of an end effector in an articulated chain, inverse kinematics will return the values for the joint-angles that will satisfy the system. Inverse kinematics is the inverse of the forward kinematics mapping. However, since forward kinematics has a many-to-one mapping, then inverse kinematics has a one-to-many mapping. This means that an input will not always give you one single correct solution. In fact, the solution ranges from no solution at all, one solution (if really lucky), multiple solutions, or an infinite number of solutions.

Consider the example in figure 2.11 again. Given the point at position A, there are two solutions that will solve the inverse kinematics system. These two solutions are illustrated by the two types of line styles in figure 2.11, giving two different positions (B and C) for the elbow.

These two solutions are given by the following set of equations:

$$\Phi = \cos^{-1}\left[\left(x^2 + y^2 - m^2 - n^2\right)/(2mn)\right]$$

$$\Theta = \cos^{-1}\left[(m + \cos\Phi)x - (n\sin\Phi)y\right]/\left(m^2 + 2mn\cos\Phi + n^2\right)$$

and

$$\Phi = 360^{o} - \cos^{-1}\left[\left(x^2 + y^2 - m^2 - n^2\right)/(2mn)\right]$$

$$\Theta = \cos^{-1}\left[(m + \cos\Phi)x + (n\sin\Phi)y\right]/\left(m^2 + 2mn\cos\Phi + n^2\right)$$

When a robotic system, or an articulated chain, has more than one solution, we say that the system is underspecified and that redundancy exists in the system.

2.4.3.3 Redundancy

A kinematics system is said to be redundant when more than one solution can be found that will satisfy the system. The system in figure 2.11 is redundant since two solutions for the joint angles can be found for the position of the end effector at A (the solutions with the elbow positions at B and C).

If all the joint angles in the system are directly specified, the system is said to be fully specified and only one unique solution exists (assuming that the length of the links is known and constant). If more information about the system is known, then we say that the system is overspecified, and some of the inverse kinematics algorithms can be simplified, increasing the efficiency of the system.

When a system is not fully specified, one can add constraints to the system to reduce the number of solutions.

2.4.3.4 Constraints

A constraints defines something about the system that must be true. It usually takes the form of added equations that must be solved for along with the inverse kinematics set of equations. For example, a ball rolling down an inclined plane would actually go through the plane if gravity was the only force acting on it. However, the ball must stay on top of the plane. The ball is thus constrained to stay on one side of the plane. Using this constraint, the system can be modelled accurately. The first system to use constraints on an animation was Sutherland's famous Sketchpad [Sutherland, 1963].
Inverse kinematics and constraints will be further explained in chapter three, where we illustrate issues involved in an implementation of the human upper body.

2.4.3.4 Further inverse kinematics

Inverse kinematics is a vast field of study and a description of every notion in inverse kinematics is beyond the scope of this project. The inverse kinematics notions discussed in this thesis are sufficient for the reader to understand the project. Thus no further knowledge of inverse kinematics methodologies will be discussed in this thesis. For further information about inverse kinematics, one can refer to [Watt and Watt, 1992] and [Wellman, 1993].

2.5 General methodologies

In commercially available packages, one tends to find two types of methods of animating articulated figures using inverse kinematics: with and without motion capture equipment.

2.5.1 Without motion capture equipment

In systems that do not use motion capture equipment, the animator specifies the position of the end effector, and the inverse kinematics engine chooses one of the available positions (if any) as the answer. The animator can then decide if that position is suitable of if another position would suit his animation better. In this way redundancy is eliminated by letting the animator choose the preferred position. Redundancy is also reduced by the application of constraints to the system. As an example, 3D studio MAX allows the user to set limits on the rotation axis of each joint, set the amount of friction on each joint as a spring tension and as a damping value. Other commercially available systems use similar methods to solve the inverse kinematics problem. Real-time animation using this system is impossible due to the fact that the motion data must be entered by the animator manually.

2.5.2 With motion capture equipment

In systems that use motion capture equipment, one tends to find that a large amount of trackers are used to reduce redundancy and hence to provide a system that is fully specified, if not over specified. If the type of trackers used have suitably high rates of update and low latency, one can achieve a real-time animation system.

Because of the use of a large number of trackers, these systems tend to be rather expensive.

Very few systems are available that make use of a low number of trackers while still achieving an adequate result in real-time articulated figure animation. In this cases, because of the redundancy in the system, one tends to find that the links in the articulated figure tend to stick in awkward positions [Badler *et. al.*, 1995].

2.5.3 Jack

One of such systems has been developed at the University of Pennsylvania's Center for Human Modelling and Simulation. The system is called *Jack* and was developed to investigate the modelling and animation of human movement (http://www.cis.upenn.edu/~hms/jack.html). *Jack* is now commercially available.

Jack provides an interactive environment for controlling articulated figures and includes many useful tools for such purpose such as realistic behavioural controls, anthropometric scaling, task animation and evaluation systems, view analysis, automatic reach and grasp, and collision detection and avoidance. The system can use forward and inverse kinematics and dynamics when calculating reach and grasp and poses.

Jack was used by Badler, Hollick and Granieri [Badler *et. al.*, 1995] to investigate issues in real-time control of virtual humans using minimal sensors. The difference between their system and our system is that their inverse kinematics system was not designed to be run on low-end hardware. As such, the inverse kinematics is solved by using powerful and expensive hardware. Also, since *Jack* would solve the inverse kinematics, constraints and articulated figure topology issues, they concentrated more in other issues such as calibration for various users, where to position the sensors in the articulated figure, and how to increase the update rates and reduce latency in the electromagnetic tracking equipment. They also use a collision avoidance feature of *Jack* to increase the constraints on the system. However, this feature is computationally expensive.

They concluded that the system, using the collision avoidance feature was a generally accurate, if not exact representation of the actual posture of the user. They found the system to be extremely useful in interactive environments although it was not accurate enough for high-fidelity motion recording and animation.

2.6 Summary

In this chapter, we describe various types of 3D animation, as well as background information on articulated figure animation. We also describe the various types of motion capture equipment available, while pointing out the advantages and deficiencies of each type with respect to the requirements of the project. We then describe the issues related to the use of kinematics and dynamics in articulated figure manipulation. Finally, we give some background information about the methodologies used to achieve similar goals as ours.

Chapter 3. Issues in Minimal Motion Capture

After the theoretical background in the previous chapter, some practical implementation is required as proof of concept and to explore and explain various issues of the project. As such, an early prototype is implemented. This prototype allows us to test the motion capture equipment and to determine whether or not the choice is adequate for our purpose. It also raises issues with the use of minimal motion capture and the presence of redundancy in the articulated chains. Additionally, the prototype allows for an early implementation of a very simple inverse kinematics system. The prototype consists of the upper body of a human articulated figure as representation of an avatar. A limited number of trackers (4) are used to determine the position and orientation of the articulated system.

3.1 Equipment

3.1.1 Motion capture equipment

For the prototype, we decided to use electromagnetic trackers. The reason for this decision is that these trackers seem to provide us with an adequate source of motion capture data, while being one of the cheapest motion capture equipment available.

We chose to use two Polhemus InsideTrak units with four receivers. The InsideTrak is a low cost version of the Polhemus Fastrak. The InsideTrak (see figure 3.1) plugs into a PC via an ISA slot, hence minimising transport delays. Each board can have up to two receivers and one transmitter. A single InsideTrak board can thus measure the position and orientation in 3D space of two separate objects.

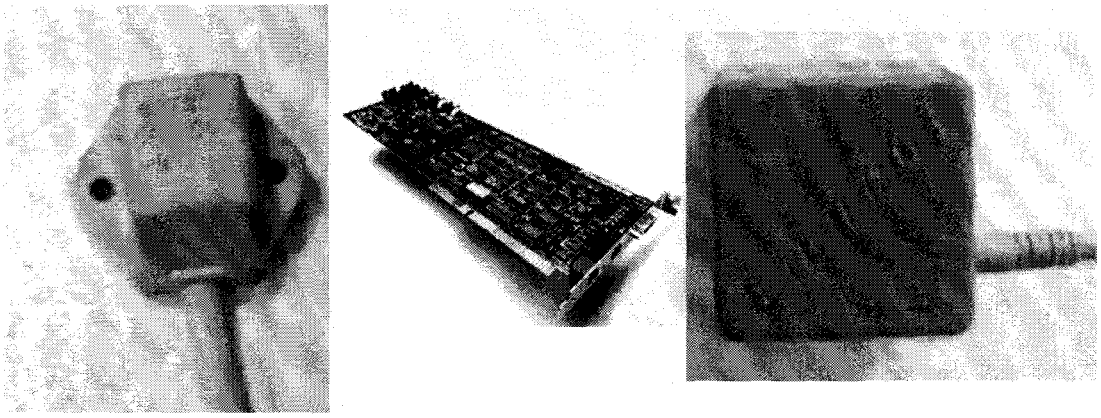


Figure 3.1 InsideTrak board, transmitter and receiver.

The system provides accurate readings of a resolution of 0.0003 inches per inch of transmitter and receiver separation and 0.03 degrees of orientation when the receivers are located within thirty inches from the transmitter [Polhemus]. The trackers have a maximum range of up to five feet, although in the limit of those ranges, the performance decreases. The system has a latency of twelve milliseconds and an update rate of thirty updates per second with two receivers.

The system benefits from a low latency and high update rates, making it quite suitable for real-time animation. The system provides both 3D position and orientation, so fewer trackers are needed to provide six DOF data. It is relatively cheap compared with other types of motion capture equipment, and is the cheapest amongst the electromagnetic trackers commercially available.

However, the Polhemus InsideTrak has a very limited range. This range could be increased with the addition of specialised equipment like the Polhemus Long Ranger, but this would increase the cost of the system. Also, since we only intend to model the human upper body, the limitation in the range does not cause a great problem.

To capture a fully specified articulated figure would require a large number of trackers each with a cable per transmitter and receiver. This would cause some problems with

encumbering the user, but since we plan to use only four receivers, we find that four wires do not cause a big encumbering nuisance, so this limitation is also eliminated.

The last problem with the InsideTrak system is interference from metallic objects and other electromagnetic sources. We have found that the system allows for quite a large amount of interference with minimal loss in the accuracy of the position. The accuracy of the orientation tends to suffer slightly more than that of the position, but the system is still quite robust. Since our research is not involved with producing a system applicable for robotic machinery but producing an animation system for interaction, precision is not of a very high order, provided that the movements of the figure are accurate enough to represent the movements of the real person using the system.

In conclusion, we found the Polhemus InsideTrak tracking system to be adequate for our purpose.

3.1.2 Virtual reality (VR) system

The prototype is implemented using the RhoVeR (Rhodes University Virtual Reality) system. This system is a parallel and distributed VR system created at the Computer Science Department of Rhodes University. It can be though of as a VR operating system in that it manages VR devices like the motion capture equipment or the 3D rendering of the scene. This VR system is selected for various reasons:

- First of all, the motion capture equipment is already integrated in the system, so there is no need to implement a driver to control it.

- The system also provides an integrated 3D rendering device.

- Since it is a distributed system, the whole process can be subdivided into smaller processes and distributed across a network of computers. As an example, one could

capture the motion data in a computer, process this data via another computer, solve for the inverse kinematics system on a third computer, and finally render the animation on a fourth computer, all connected through a network.

- Lastly, RhoVeR is used because of previous knowledge of the system.

For a more in depth description of RhoVeR, please refer to [Bangay *et al*, 1996], [Bangay, 1996],[Gain, 1996], and [Casanueva, 1996].

3.2 Prototype

3.2.1 Human articulated skeleton

Since we are limiting ourselves to only four trackers, there are some compromises that have to be made. The first is that we will limit ourselves to animating only the upper body of a human articulated figure. This is not such a bad thing since the lower body is much more suited for scripting animation than the upper body. As such, the legs can be animated using some kind of script according to whether the user wants to walk, run, dance, stand on one leg, and so forth. A lot of work has been done in animating the human lower body using scripts and one can refer to [references, [Badler and Ko, 1996], [Magnenat and Thalmann, 1985], [Perlin and Goldberg, 1996] for further references.

The upper body of a human articulated figure can be represented as shown in figures 3.2 and 3.3.

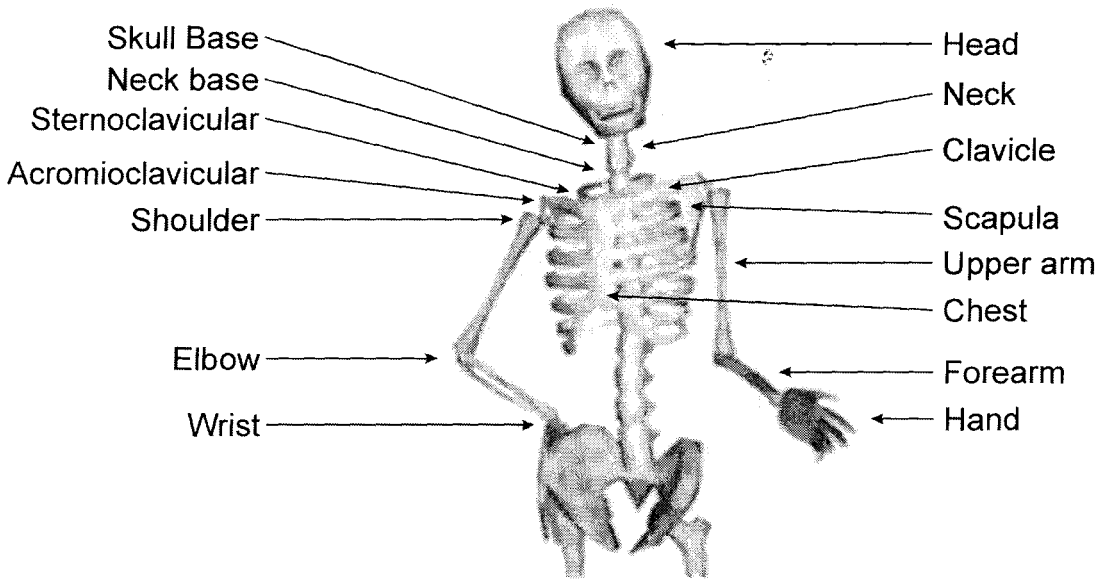


Figure 3.2. Anatomical representation of the human upper body. The links are denoted on the right hand side of the figure while the joints are denoted of the left hand side.

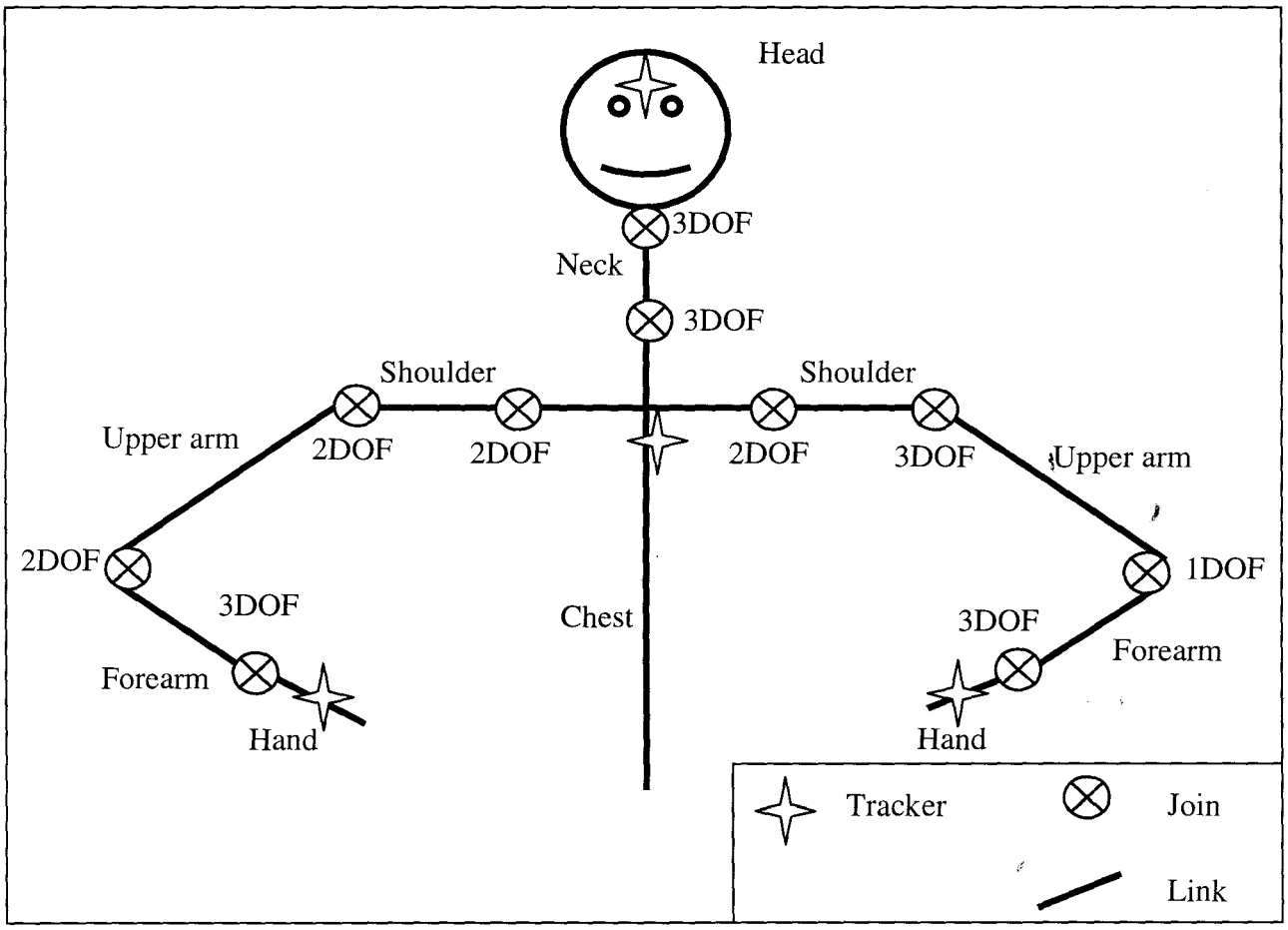


Figure 3.3: Representation of the human upper body articulated figure as commonly used by articulated figure animators.

As one can see, the upper body is composed of three articulated chains:

- The chest, neck, head chain, made up of three links and six DOF.
- The chest, left clavicle, left upper arm, left forearm, left hand, composed of five links and nine DOF.
- The right chest, right clavicle, right upper arm, right forearm, right hand, composed of five links and nine DOF.

As stated earlier, we plan to use only four six DOF trackers. The trackers are to be positioned at the chest, the head and one at each hand, as shown in figure 3.3.

This means that we have to determine the position and orientation of each nine DOF arms from the position and orientation of the hand link. Because of the complexity introduced by the nine DOF, this task is highly complicated. Another compromise has to be made. We thus decide to eliminate the clavicle joint as seen in figure 3.4. This clavicle joint only has a restricted two DOF movement, so the resulting articulated figure is functionally almost identical to the full articulated figure.

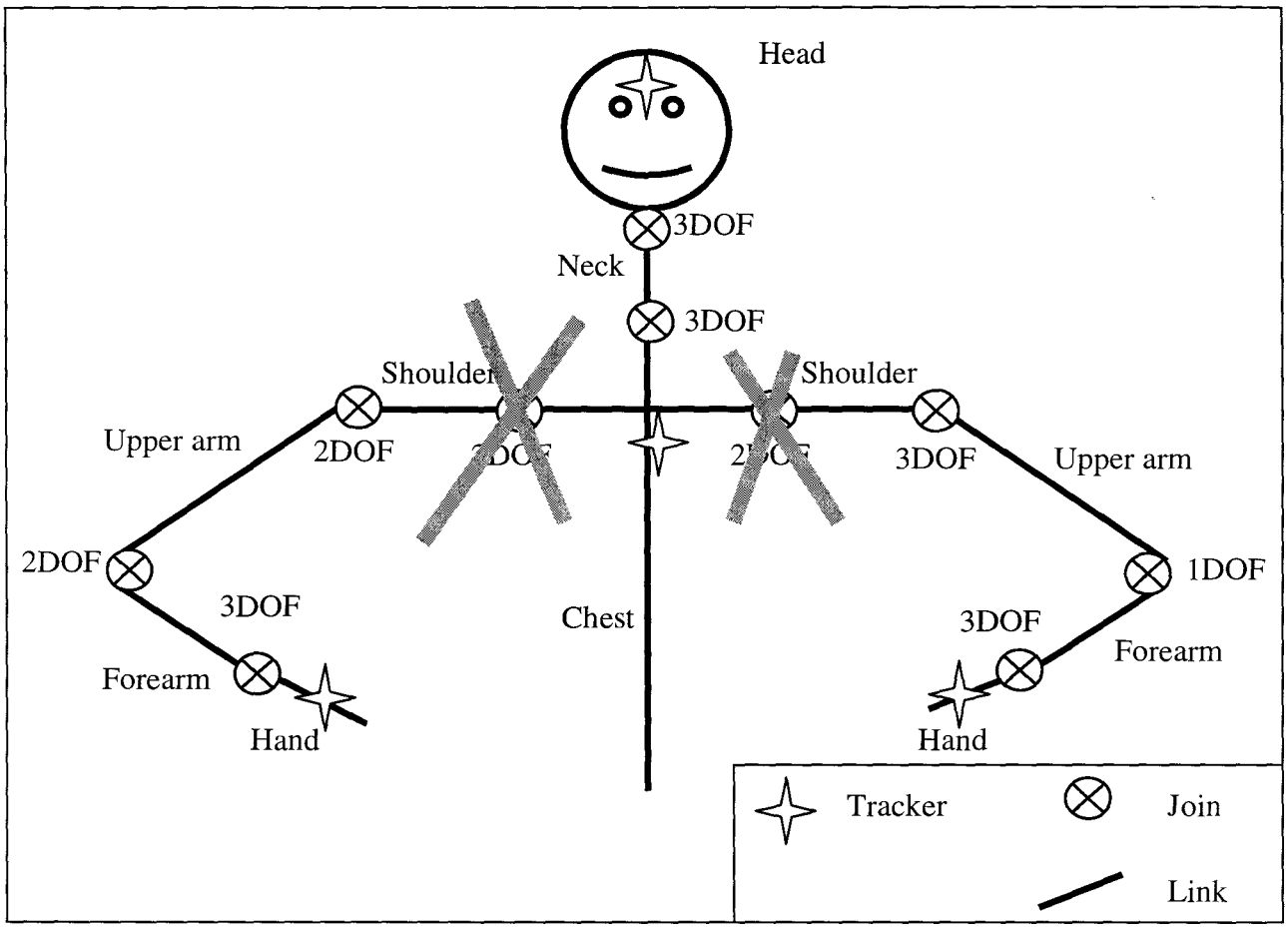


Figure 3.4 Representation of the human upper body as seen in figure 3.3 but without the clavicle joint

3.2.2 Redundancy in the articulated skeleton

As a result of the elimination of the clavicle joint, our articulated figure is now composed of three chains, one with six DOF and two with seven DOF. However, a single six DOF tracker is unable to provide enough data about the position and orientation of all the links in the articulated figure. Therefore all the chains have a certain degree of redundancy. To reduce such redundancy, various constraints can be applied.

3.2.3 Constraints for the initial prototype

3.2.3.1 The chest-neck-head chain

The chest-neck-head chain is composed of three links and two three DOF joints. Each joint has one DOF on each of the following axis: the front-back axis, the left-right axis and the up-down axis(or roll, elevation and azimuth in Euler angles).
Using a six DOF tracker one the head of the articulated figure, and another one on the chest, one can not determine the rotation along the vertical (up-down) neck axis for the two joints, as shown in figure 3.5.a).

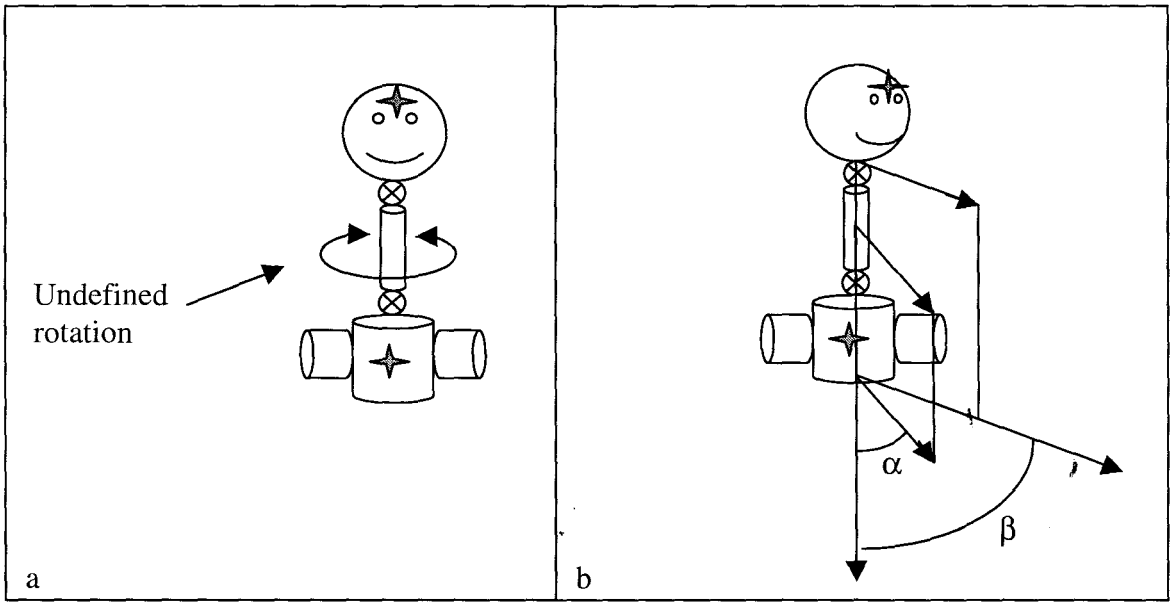


Figure 3.5. a) The rotation of the neck can not be defined by the head and chest links alone. b) Using simple constraints, one can find a value for α based on the value for β.

To specify such rotation, one uses a constraint. A simple and effective constraint is to limit the undefined angle α to be a weighted factor of the angle β between the chest and head. A usual weight for that angle is for it to be half the chest-head angle, as shown on figure 3.5.b).

Using such constraint, we manage to get a fully specified articulated chain.

3.2.3.2 The chest - upper arm - forearm - hand chain

The chest - upper arm – forearm - hand chain contains four links and nine degrees of freedom. These DOF can be specified in a three-one-three or a two-two-three DOF configuration as shown for the left and right arms in figure 3.3. Either way, one can not find a unique position for the elbow joint, and orientation for the upper arm and the forearm, making the human articulated arm a redundant chain.

Since the initial prototype is designed to test the concept as well as the equipment, we decided to apply a very restrictive constraint. The position of the elbow joints is constrained to be in the same plane as the body and the hands. This means that although the orientation of the links is not restricted in any way, the position of the links was restricted in one dimension. This allows us to use a simple set of inverse kinematics equations for a two dimensional system, similar to equations 2 and 3.

3.3 Issues raised for the prototype

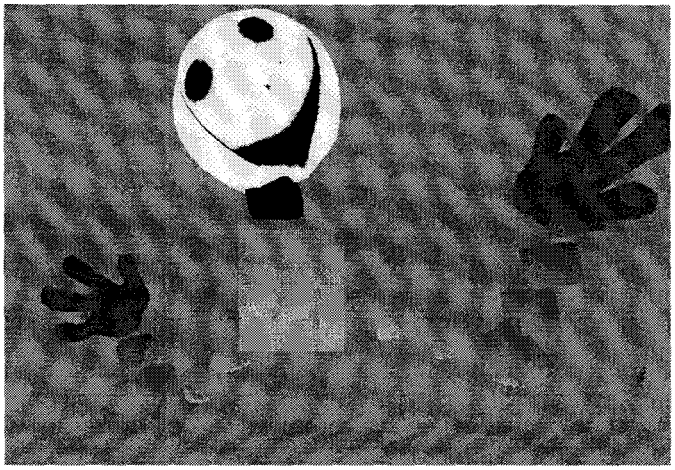


Figure 3.6. The articulated figure animation in action.

3.3.1 Articulated figure animation

For the prototype, a simple avatar representation is used, as seen in figure 3.6.

The dimensions of the user are ignored, and a standard set of dimensions are used. This avoids any user specific calibration of the system. Since the user is very likely to use an avatar representation different from his own (in fact he could use a representation which is not even based on a human figure), this is an acceptable compromise. It is found that due to the cartoonish nature of the avatar, the difference between the user's dimensions and the dimensions of the avatar do not cause any noticeable discrepancy in the description of the movements. The user can immediately identify himself with the avatar thanks to the fast response of the animation. Spectators can also easily identify the relation between the user and the avatar.

Kinematics is used to animate the articulated figure for various reasons. First of all, the nature of the captured data lends itself to kinematics more than dynamics. The motion capture system provides position and orientation but not forces, which are required in dynamics. One could use the position to calculate the acceleration of the limbs and then, with previous knowledge of the mass of each limb, calculate the force that causes the movement. However, the force calculated is the resultant force, sum of all forces in the system and there is no way of calculating the individual forces. These forces might be needed to resolve redundancy issues. Also, the fact that the data given by the trackers is not very precise means that inaccuracies in the acceleration calculations will produce errors. This is specially noticeable with the orientation data used to calculate torque forces. Also the mass of each limb needs to be accurately measured, requiring extensive and precise calibration Lastly, kinematics systems tend to have a lower complexity and so tend to be computationally cheaper than dynamics systems [Calvert and Chapman, 1982].

3.3.2 Motion Capture

The motion capture equipment used has proven to be more than adequate for the task. It has provided fast data rates and low latencies, allowing for real-time animation. However the animation has a frame per second ratio of only around ten frames per second. This is mainly due to the virtual reality system and the rendering engine used. This causes a bottleneck and thus limits the frames per second ratio of the animation well beyond the capabilities of the motion capture equipment. The latency is low and is quite adequate for our purpose. The latency tends to increase slightly as the system runs. This is also due to the VR system used.

The precision of the trackers is good. There has been some "shaking" of the joints due to inaccuracies in the motion capture system or due to electromagnetic interference, but this can be eliminated by inserting a filter that will filter out minimal changes in the position or orientation of the data.

Because of the cartoonish nature of the avatar used, inaccuracies in the position and orientation are hard to spot and are easily forgiven by the user or spectator.

3.3.3 Inverse kinematics

Although the inverse kinematics used are basically of a two dimensional nature as explained in 3.2.3.2, the avatar looks as if it is animated using proper 3D kinematics. This is due to the fact that the links are able to rotate in 3D space and that the two dimensional inverse kinematics is only used to calculate the position of the elbows. Also the "clownish" appearance of the avatar somehow hides the nature of the 2D kinematics. No one yet has noticed the fact that the system uses 2D kinematics, except for the people to whom this was pointed out.

Due to the simplicity of the inverse kinematics system, solving for the inverse kinematics is a computationally cheap process. This makes the process very suitable for real-time animation.

This inverse kinematics method can be used to represent an avatar in a virtual environment. However, the avatar will not be able to fully interact with the world (specifically when the interaction requires the hands of the avatar to be in front of the body, which is not possible because the hands are bound to the 2D plane and are thus at the same 'depth' as the body).

For animation, this inverse kinematics method is unacceptable since it can not represent the full motion of the human upper body.

3.4 Summary

The first prototype proves that the system is feasible. The electromagnetic trackers are accurate and have a high update rate and a minimal latency making them suitable for real-time motion capture and animation. The first prototype also allows for a description and investigation of issues related to solving the inverse kinematics problem such as the presence of redundancy in the system and the use of constraints. The use of only four trackers for the human articulated upper body results in redundant articulated chains which need constraints to be fully specified. A better inverse kinematics system has to be used along with added constraints in order to fully specify the system, or at least in order to reduce the redundancy of the system. The VR system used proved to be a limiting factor is the update rate of the motion capture equipment. The actual update rate is adequate for usage in a virtual environment and for low quality animation, but proves inadequate for high quality animation. A way of increasing the update rate would provide an acceptable update rate for high quality animation. Nevertheless, the whole process is fast enough to be used in real time.

Although the first prototype proves that the system is feasible, it also highlights deficiency areas. Such areas include the below specification performance of the update rate of the motion capture equipment due to the VR system used. Other areas are the inverse kinematics system used. A proper 3D system is required to increase the accuracy of the poses of the avatar with respect to the poses of the user. In the following sections we address such deficiency areas. This also allows us to investigate three different inverse kinematics systems.

Chapter 4. Design

As a result of the implementation of the first prototype, we can consider the trackers to be quite adequate in providing real-time motion capture with a satisfactory degree of precision. Still, they could benefit from an improved update rate. However the inverse kinematics equations used are not complete enough to reflect the true position of the various components of the articulated figure. This could be solved by adding more trackers to the user, but this would increase the cost of the system. Using various constraints, one can try to reduce the redundancy of the system to provide an acceptable mapping from the real user to the virtual articulated figure.

In the second prototype, we design and implement various methods of dealing with the redundancy of articulated chains, and in particular the articulated human upper body as discussed in chapter 3 and shown in figure 3.4.

4.1 Articulated figure animation

Since the articulated figure can be used as an avatar, as well as for articulated figure animation, the position of the hands of the user needs to be mapped accurately if the user is expected to interact with his environment.

This fact precludes the use of a scripting animation system since each move that the user does would have to be mapped into a pre-recorded move. A large number of pre-recorded moves and an expensive algorithm would have to be used to determine the exact move. The number of moves can be extremely large (consider mapping each different move that a human arm can do). To achieve an acceptable latency, one would have to limit the number of moves stored, thus limiting the accuracy of the animation. A scripting system could be used for parts of the body that do not require such accurate mapping. To map the lower body of an articulated figure, one could use tracker information to determine the

type of walking (or standing) that the figure is doing, and map it to a specific scripted animation. This mapping could not be just limited to the speed at which the person walks, but also to certain mannerisms of the walking or pose of the person.

The only acceptable choice for our system is to use key frame animation. Key frame animation is concerned with the change of position of the various parts of the articulated figure. This fits nicely with our goal since the position of the tracked parts of the articulated figure can be faithfully animated, while the use of inverse kinematics will determine the position of the non tracked parts of the articulated figure.

Animating the figure is rather simple because one only needs to worry about calculating the change in position and orientation for all the links in the articulated figure. Since the update rates for the position and orientation of the tracked links is quite high (at about thirty frames per second according to the specifications of the equipment), one does not need to worry about interpolating between frames. This also reduces the workload, which in turn reduces the latency of the system.

4.2 Motion capture

4.2.1 Trackers suitability

In the first prototype (discussed in chapter 3) the motion capture trackers provided an update rate of more than ten frames per second. This is well below the specifications of the product and thus leads us to suspect that this lower than specified performance is due to the VR system used (as explained in section 5.1.1). The latency is quite low although it could also benefit from a more suitable VR system. The trackers suffer from a decrease in accuracy as the range between transmitter and receiver increases, and also as a result of ferromagnetic or electromagnetic interference. However, if the user limits his general

position to the range of the trackers, the accuracy is acceptable to animate the articulated figure.

Still, the motion capture system is quite suitable to animate an avatar in real time at a rate of around ten frames per second.

4.2.2 Trackers placement

As stated earlier, the articulated figure is composed of various articulated chains. A tracker should be placed at the main root of the articulated figure. This tracker is used to translate the whole figure around the virtual world. Other trackers should be attached to the root of a chain that requires its position to be known exactly. Trackers are also best placed at the end effectors of the articulated chains. However, different criteria will determine if this is so. To investigate these criteria, we will consider two example chains: the human articulated arm and the human articulated leg. Both chains are composed of four links and three joints. However, the human tends to use his hands to interact much more often than his legs. As such, it is very important to know the accurate position of the hands.

If we put the tracker in the forearm, we can track the position of the elbow accurately but the orientation of the hand is then completely unknown to us. There are no viable constraints that can determine the precise orientation of the hand, and thus the position of the hand is also compromised (we know where the root of the hand is, because we know the position of the forearm-hand joint, but we can not know the position of the end of the hand). This makes for an unintuitive avatar and a flawed articulated figure in articulated figure animation. It is preferable to put the tracker at the hand, and infer the position of the elbow using inverse kinematics and constraints (even if the position of the elbow in not totally accurate) than to put the tracker at the position of the forearm and not know the position and orientation of the hand.

To model the leg one finds a redundancy problem similar to the arm. If we put the tracker on the foot, the position of the knee is not known exactly and will have to be inferred in a similar way as for the elbow in the arm articulated chain. If the tracker is placed on the lower leg, the orientation of the foot is not known. In the case of the leg however, we can use the fact that the foot usually rests on the ground to constraint the orientation of the foot. Also, since humans do not tend to interact much with their environment using their feet (except when walking or kicking) the orientation of the foot is not critical when working with an avatar. This setup can be more problematic when dealing with an animation because of the need in an animation for a high level of accuracy.

All trackers should be attached firmly to their specific link so as to avoid errors in the orientation of the readings. This is especially important for the tracker attached to the root link, since any error in the reading of this tracker will affect the rest of the figure as opposed to only affecting the specific articulated chain.

4.3 Inverse kinematics

4.3.1 Specifying the inverse kinematics chain

Although the DH notation can accurately specify non-branched kinematics chains, it is somewhat non-intuitive. This is especially true when trying to visualise the movement of a chain in 3D space for articulated figures. One of the reasons for this non-intuitiveness is that there are far too many different variables that must be tracked at the same time. To simplify this, we decide to use our own specification, based on Korein's work [Korein, 1985]. This approach ignores the variables that are not affected by the movement of the articulated figure (and are thus constant) and makes use of some simplifications in the structure of the joints.

We assume that the links are straight and of constant length. We decide to limit ourselves to revolute joints. Prismatic joints are extremely rare in articulated figure animation so this assumption is not really restricting us.

In doing so, we have eliminated two parameters in the DH notation and transformed them into a single constant. Indeed, a_i (the length of the link) and d_i (the distance between links) are combined to form the length of our link. Since prismatic joints are not considered, the length of our link is kept constant.

We define each one DOF revolute joint to be of one of two types:

- Twist: when the axis with respect to which the joint is articulated and the direction of the joint coincide (this would be the equivalent of roll in Euler angles) as seen in figure 4.1. This is equivalent to α_i (the twist of the link) in the DH notation.

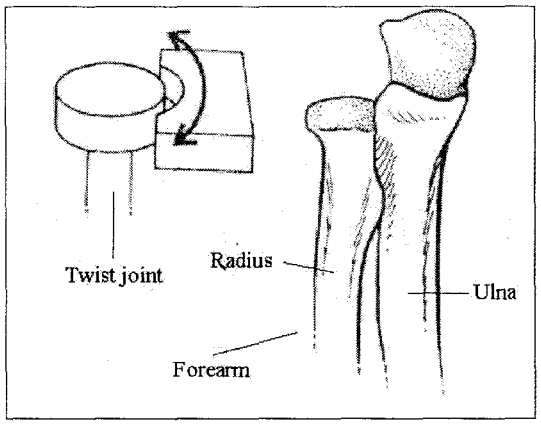


Figure 4.1 Twist joint and an equivalent joint in a human skeleton

- Flexion: when the axis with respect to which the joint is articulated and the direction of the joint do not coincide (this would be the equivalent of azimuth or elevation in

Euler angles), as seen in figure 4.2. This is equivalent to a combination of Θ_i (the angle between links) and d_i (the distance between links) in the DH notation.

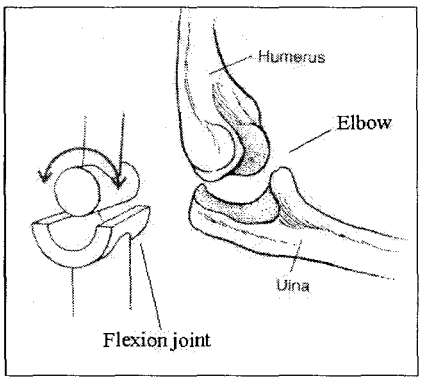


Figure 4.2 Flexion joint and an equivalent joint in a human skeleton

As we can see, our specification is not that different to the DH notation. However, we find that it makes 'visualising' the position, orientation and reach of the articulated chain far more intuitive and easier.

We also define a spherical joint as a joint that has two flexion joints and one twist joint, as seen in figure 4.3.

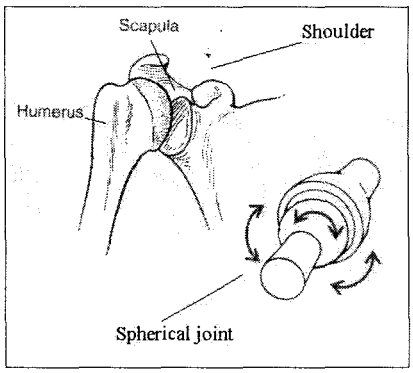


Figure 4.3 Spherical joint and an equivalent joint in the human skeleton

One can clearly see that a point at the end of a link attached to a spherical joint will always lie in the sphere centred at the joint, and with radius equal to the length of the link (hence the name for the joint).

Furthermore a joint with more than one twist DOF and no flexion DOF can be replaced by a joint with a single twist DOF since the reach of such joints is identical.

Similarly, a joint with more than two flexion DOF and no twist DOF can be replaced by a joint with two flexion DOF since the reach of such joints is identical.

4.3.2 Reducing the redundancy for various articulated chains

Using our specification for articulated chains, we analyse various types of articulated chains. All the chains analysed comprise a root joint (joint A in subsequent figures) and a root link (link P in subsequent figures). The position, orientation and length of the root link are known. This is so because a tracker would be attached to the root link. The amount of DOF in the root joint A is irrelevant due to the fact that we already know the position, orientation and length of the root link.

The amount of DOF in each joint is different for each case but is known. The lengths of all the links is also known. Also two joints can not be in the same place (they can not coincide).

4.3.2.1 The one link articulated chain

The link is composed of a root joint and a root link as seen in figure 4.4.

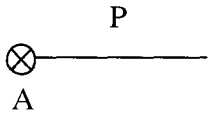


Figure 4.4 The one link chain.

As stated before, since the position, orientation and length of the root link are known, then there is no real challenge in determining the position of the end effector, since the end effector in this case is the root link. Therefore the reach of the chain is determined by the position of the end tip of link P. Thus the reach is a single point.

4.3.2.2 The two links articulated chain

This chain is composed of a root joint and a root link, and a second link attached to the end of the root joint as seen in figure 4.5.

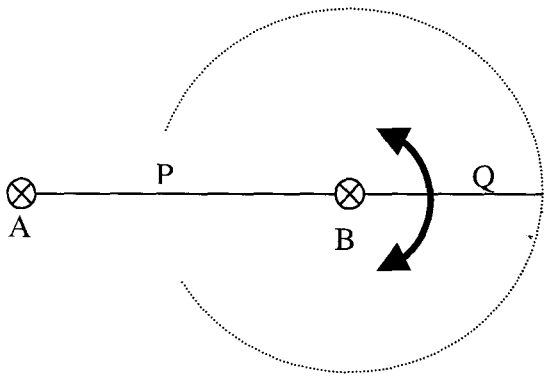


Figure 4.5 The two links chain.

This chain can have a number of DOF at joint B.

4.3.2.2.1 One DOF at joint B

With the joint at B having only one DOF, two options arise:

- If the joint at B is a flexion joint, the circle lies in the plane of the page (as shown in figure 4.5). In this case, the roll of the link Q is constant and no further constraints are required to specify it.

- If the joint at B is a twist joint, then the circle lies in the plane whose normal is the link P, and that contains the last point of link Q. In this case, the roll of link Q is undefined.

4.3.2.2.2 Two DOF at joint B

This chain has two DOF in the B joint. In this case, three different cases arise:

- If those two DOF at joint B are both flexion DOF, then the reach of the link Q will have the form of a sphere. The centre of the sphere is joint B and the radius is the length of link Q. In this case, the roll of link Q is constant.

- If those two DOF at joint B are one a flexion DOF and one a twist DOF, then the reach of the link Q will also have the form of a sphere. The centre of the sphere is joint B and the radius is the length of link Q. In this case however, the roll of link Q is undefined.

- If the two DOF are both twist DOF, then the reach of the link Q will be the same as that of a single twist DOF as seen in section 4.3.2.2.1. That is the reach will have the form of a circle. The circle will lie in the plane whose normal is the link P, and that contains the last point of link Q. In this case, the roll of link Q is undefined.

4.3.2.2.3 Three DOF at joint B

This chain has three DOF in the B joint. In this case, four cases arise:

- If these three DOF are all flexion DOF, then joint B can be replaced with a two DOF flexion only joint without affecting the position of the last point of link Q. We thus have a reach in the form of a sphere as explained in section 4.3.2.2.2. The roll of link Q is thus constant.

- If these three DOF are all twist DOF, then joint B can be replaced by a joint with one twist DOF without affecting the position of the last point of link Q. We thus have a reach in the form of a sphere as explained in section 4.3.2.2.1. In this case, the roll of link Q is undefined.

- If joint B consists of two twist DOF and one flexion DOF, then joint B can be replaced by a joint consisting of a flexion DOF and a twist DOF without affecting the position of the last point of link Q. We thus have a reach in the form of a sphere as explained in section 4.3.2.2.2. In this case, the roll of link Q is also undefined.

- Finally if joint B consists of two flexion DOF and one twist DOF, a spherical joint as described in section 4.3.1 can replace it. The reach of this joint is in the form of a sphere. The roll of link Q is undefined.

4.3.2.3 The three links articulated chain

This chain is composed of a root joint and a root link, a second link attached to the end of the root joint and a third link attached to the second link as seen in figure 4.6.

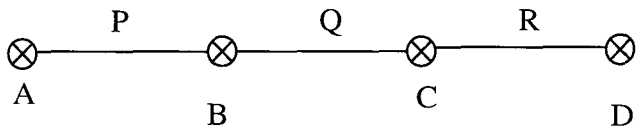


Figure 4.6 The three links chain

At this point, the complexity of the chain means that the chain is highly redundant. Most of the time the end point of link R will be found anywhere inside the sphere with B as centre and the added length of links Q and R as radius. Unfortunately, this does not provide us with any useful knowledge since this space is far too large.

When applying these techniques to articulated figure animation, one must remember that the length of articulated figure chains is usually kept quite short to reduce a large amount of redundancy. As such, the longest chain in our articulated human upper body consists of a four link chain with seven DOF. One must also remember that trackers are usually placed at the end effector of the articulated chain, or at least very near the end effector of the chain.

This means that we know the position orientation and length of both the root link and the end effector link.

An astute reader would observe that this makes sections 4.3.2.1 and 4.3.2.2 irrelevant since in the first section the information of the link is already known (since it is the root link) and in the second case the information of the previously unknown link is now known (since it is the end effector link). However, this is not entirely right. Sections 4.3.2.1 and 4.3.2.2 provide the reach for short specific chains. These chains can be used to construct the now longer chains by taking two short chains and joining them at the two end effectors. Thus one of the old root links stays as the new root link while the other root link now becomes the end effector.

Therefore the new three link chain is composed of a two link chain as described in section 4.3.2.2 and a one link chain as described in section 4.3.2.1. The two link chain has link P as link root (in figure 4.6) and link Q as end effector, while the one link chain has link R as root link. The two chains are joined at joint C.

Since the chain is continuous, links Q and R must be connected at joint C. This means that, since we know the position, orientation and length of links P and R, the position of joints B and C can easily be calculated. This means that we now know the position and part of the orientation of link Q. The only missing factor in the orientation is the roll of link Q.

Still, let us consider the intersection of the two link chain and the one link chain making up the three link chain for various joint types at joint B.

4.3.2.3.1 One DOF at joint B

If the chain has one DOF at B we can get two different cases:

- If the joint is a flexion joint, the intersection is that of a circle (reach of chain P-Q) and a point (reach of chain R), giving us a point as expected. Also the roll of link Q is constant irrespective of the type of joint at C.

- If the joint is a twist joint, the intersection is that of a circle (reach of chain P-Q) and a point (reach of chain R), giving us a point as expected. In this case, the roll of link Q is determined by the type of joint at C: if C has a flexion DOF, then the roll is constant. However if C has a twist DOF, then the roll is undefined.

4.3.2.3.2 Two DOF at joint B

If the chain has two DOF at B we can get three different cases:

- If the joint at B has two twist DOF, then is can be substituted for a single twist joint just as in section 4.3.2.2.2. This then becomes the second case in section 4.3.2.3.1. The intersection is that of a point and a circle, giving us a point. The type of joint at C determines the roll of link Q: if C has a flexion DOF, then the roll is constant. However if C has a twist DOF, then the roll is undefined.

- If the joint at B has two flexion DOF, then the intersection is that of a sphere and a point, giving us a point. In this case, the roll of link Q is constant irrespective of the type of joint at C.

- If the joint at B has a twist DOF and a flexion DOF, then the intersection is that of a sphere and a point, giving us a point. In this case, the roll of link Q is determined by the type of joint at C: if C has a flexion DOF, then the roll is constant. However if C has a twist DOF, then the roll is undefined.

4.3.2.3.3 Three DOF at joint B

If the chain has three DOF at B we can get four different cases:

- If all three DOF are flexion DOF, then joint B can be replaced with a two DOF flexion joint just like in section 4.3.2.2.3. This then becomes the second case in section 4.3.2.3.2. The roll of link Q is constant.

- If all three DOF are twist DOF, then joint B can be replaced with a one DOF twist joint just like in section 4.3.2.2.3. This then becomes the second case in section 4.3.2.3.1. The type of joint at C determines the roll of link Q: if C has a flexion DOF, then the roll is constant. However if C has a twist DOF, then the roll is undefined.

- If two DOF are twist DOF and one DOF is a flexion DOF, then joint B can be replaced with a joint composed of one DOF twist and a one DOF flexion This then becomes the third case in section 4.3.2.3.2. In this case, the roll of link Q is

determined by the type of joint at C: if C has a flexion DOF, then the roll is constant. However if C has a twist DOF, then the roll in undefined.

- If two DOF are flexion DOF and one DOF is a twist DOF, then joint B can be replaced with a spherical joint. The intersection is that of a point and a sphere. The roll of link Q is determined by the type of joint at C: if C has a flexion DOF, then the roll is constant. However if C has a twist DOF, then the roll in undefined.

4.3.2.4 The four link chain

As with the three link chain, the four link chain is composed of two two link chains (as shown in figure 4.7) joined at their respective end effectors.

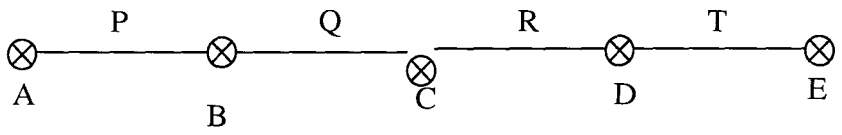


Figure 4.7 Articulated chain comprising four non-zero length links

The two two link chains are chain P-Q and chain T-R which are joined at joint C.

The number of DOF of joint C does not affect the position of joint C. This position is determined by the links Q and R and thus by the joints B and D since the length of the links is constant. The number of DOF at C will however determine the possible orientation of the links Q and R. More specifically, it will determine the amount of roll that these links can achieve.

4.3.2.4.1 One DOF at joint B and one DOF at joint D

This case is equivalent to two chains as described in section 4.3.2.2.1 connected front to front at the tip of the end effectors. Whether the joints at B and D are of a twist or flexion joint, the reach for such sub-chains is always a circle as seen in section 4.3.2.2.1. The intersection of these two links is thus the intersection of two circles. This means at most two points. Theoretically, the intersection could be a circle if the joints B and D coincided and the length of links Q and R where the same. However, this can not happen since joints B and D can not coincide. For link Q to have an undefined roll, both the joints at B and C must have twist DOF. Otherwise the roll of link Q is constant. Similarly link R requires both joints C and D to have twist DOF for its roll to be undefined.

4.3.2.4.2 Two DOF at joint B and one DOF at joint D

In this case the chain is made up of a two link sub-chain containing a two DOF joint in the non-root joint (as seen in section 4.3.2.2.2) and another two link sub-chain containing a one DOF joint in the non-root joint (as seen in section 4.3.2.2.1). If joint D is a two DOF joint while joint B is a one DOF joint, this method can be applied by reversing the order of the chain.

Three possible cases can happen:

- If joint B is made up of two twist DOF, then it can be replaced by a one twist DOF joint as seen in section 4.3.2.2.2. The reach will then be in the form of a circle. The intersection of the two sub chains will be at most two points as seen in section 4.3.2.4.1. For link Q to have an undefined roll, joint C must also have at twist DOF. Otherwise the roll of link Q is constant. Similarly link R requires both joints C and D to have twist DOF for its roll to be undefined.

- If joint B is made up of two flexion DOF, then the reach of the sub-chain is in the form of a sphere as seen in section 4.3.2.2.2. The intersection of the sphere and the

circle will usually give a two point intersection. In some special cases, the intersection of these two can give a circle. This happens when the circle (reach of the one sub-chain) lies in the sphere (reach of the other sub-chain). Since both DOF at joint B are flexion DOF, link Q is always constant. For link R to be undefined, both joints C and D must have twist DOF.

- If joint B is made up of one flexion and one twist DOF, then the reach of the sub-chain is in the form of a sphere as seen in section 4.3.2.2.2. The intersection of the sphere and the circle will usually give a two point intersection. In some special cases, the intersection of these two can give a circle. This happens when the circle (reach of the one sub-chain) lies in the sphere (reach of the other sub-chain). If the joint at C has a twist DOF, then the roll of link Q is undefined. For link R to be undefined, both joints C and D must have twist DOF.

4.3.2.4.3 Two DOF at both joints B and D

In this case the chain is made up of two two link sub-chains containing a two DOF joint in the non-root joint (as seen in section 4.3.2.2.2).

Four possible cases can happen:

- If both joints B and D have two twist joints, then they can both be replaced by single twist joints. The intersection of the two chains is thus that of two circle, giving at most two points. If joint C has a twist DOF, then both links Q and R are undefined.

- If both joints B and D have two flexion joints, then the intersection is that of two spheres, giving and intersection in the shape of a circle. Theoretically the intersection could be a sphere if the joints B and D coincided and the length of links Q and R where the same. However, this can not happen since joints B and D can not coincide. No matter what type of joint C is, it is always constant.

- If one of the joints (lets say joint B) has a flexion and a twist DOF while the other one (lets say joint D) has two twist joints, then joint D can be replaced by a single twist joint. In this case, the intersection is that of a sphere and a circle. This intersection will usually give a two point intersection. In some special cases, the intersection of these two can give a circle. This happens when the circle (reach of the one sub-chain) lies in the sphere (reach of the other sub-chain). If the joint at C has a twist DOF, then the roll of link Q is undefined. For link R to be undefined, both joints C and D must have twist DOF.

- If one of the joints (lets say joint B) has a flexion and a twist DOF while the other one (lets say joint D) has two flexion joints, then the intersection is that of two spheres. This will give a circle (it can not give a sphere as discussed in the second point of this section). The roll of link R is always constant while that of link Q required that joint C have a twist DOF for it to be undefined.

4.3.2.4.4 Three DOF at joint B and one DOF and joint D

In this case the chain is made up of a two link sub-chain containing a three DOF joint in the non-root joint (as seen in section 4.3.2.2.3) and a two link sub-chain containing a one DOF joint in the non-root joint (as seen in section 4.3.2.2.1).

Four different cases arise:

- If joint B has three twist DOF, then it can be replaced by a one twist joint. In this case we have the same topology as in section 4.3.2.4.1 and thus the same results.

- If joint B has three flexion DOF, then it can be replaced by a two flexion joint. In this case we have the same topology as the second case in section 4.3.2.4.2 and thus the same results.

- If joint B has two twist DOF and one flexion DOF, then it can be replaced by a one twist DOF and one flexion DOF. In this case we have the same topology as the third case in section 4.3.2.4.2 and thus the same results.

- If joint B has two flexion DOF and one twist DOF, then it can be replaced by a spherical joint. The intersection is that of a sphere and a circle. This is usually two points although it can sometimes be a circle. The roll of link Q is undefined if joint C has a twist DOF. The roll of link R is undefined if both joint C and joint D have a twist DOF.

4.3.2.4.5 Three DOF at joint B and two DOF and joint D

In this case the chain is made up of a two link sub-chain containing a three DOF joint in the non-root joint (as seen in section 4.3.2.2.3) and a two link sub-chain containing a two DOF joint in the non-root joint (as seen in section 4.3.2.2.2).

Four major cases arise: most of them can be reduced to previously covered cases.

- If joint B has three twist DOF, then it can be replaced by a one twist joint. In this case we have the same topology as in section 4.3.2.4.2 and thus the same results.

- If joint B has three flexion DOF, then it can be replaced by a two flexion joint. In this case we have the same topology as in section 4.3.2.4.3 and thus the same results.

- If joint B has two twist DOF and one flexion DOF, then it can be replaced by a one twist DOF and one flexion DOF. In this case we have the same topology as in section 4.3.2.4.3 and thus the same results.

- If joint B has two flexion DOF and one twist DOF, then it can be replaced by a spherical joint. If joint D has two twist DOF, then it can be replaced by a twist DOF.

The intersection is that of a sphere and a circle. This is usually two points but it can be a full circle. The roll of links Q and R is undefined if joint C has a twist DOF.

- If joint B has two flexion DOF and one twist DOF, then it can be replaced by a spherical joint. If joint D has two flexion DOF or one flexion DOF and one twist DOF, then the intersection is that of a two spheres, giving an intersection in the shape of a circle. Link Q is undefined if joint C has a twist DOF. For joint R to be undefined, both joints C and D must have a twist DOF.

4.3.2.4.6 Three DOF at both joint B and joint D

In this case the chain is made up of two two link sub-chains containing a three DOF joint in the non-root joint (as seen in section 4.3.2.2.3).

Four major cases arise: most of them can be reduced to previously covered cases.

- If joint B or joint D have three twist DOF, then they can be replaced by a one twist joint. In this case we either have the same topology as in section 4.3.2.4.1 or as in section 4.3.2.4.4, and thus the same results.

- If joint B or joint D have three flexion DOF, then they can be replaced by a two flexion joint. In this case we have the same topology as in section 4.3.2.4.3 or as in section 4.3.2.4.5, and thus the same results.

- If either joint B or joint D (or both) have two twist DOF and one flexion DOF, then the link can be replaced by a one twist DOF and one flexion DOF. In this case the resulting topology is similar to a topology found in sections 4.3.2.4.3 or 4.3.2.4.5 and thus the same results.

- If both joint B and joint D have two flexion DOF and one twist DOF, then they can both be replaced by spherical joints. The intersection of these two is a circle. The roll of links Q and R are undefined if joint C has a twist DOF.

- If joint B has two flexion DOF and one twist DOF, then it can be replaced by spherical joint. Depending on the type of joint at D, we get a topology similar to the ones in sections 4.3.2.4.4 or 4.3.2.4.5.

4.3.2.5 The five link chain

Due to the increase in the number of links, this chain is highly redundant and the possible reach for the undetermined joints is far too large to be useful.

4.3.2.6 Summary

We provide a summary of sections 4.3.2.1, 4.3.2.2, 4.3.2.3 and 4.3.2.4 in the form of a table. The articulated chain must be reduced to a simplified chain if possible before using the table.

The roll of the non-root links is undefined if the joints of the link have both a twist DOF.

Chain topology	Reach for the undetermined joint
----	A point
----1----	A circle
----2----	A sphere
----3----	A sphere
----1----x----	A point
----2----x----	A point
----3----x----	A point
----1----x----1----	Two points
----2----x----1----	A circle (in special cases) or two points
----2----x----2----	Two points or a circle according to the type of joints
----3----x----1----	A circle (in special cases) or two points
----3----x----2----	A circle (in special cases) or two points
----3----x----3----	A circle

Table 4.1 Summary of the various types of articulated chains discussed and the possible position of the undetermined joints.

4.3.3 The sphere intersection and the arm

The articulated arm as seen in figure 3.4 is a four non-zero length links with a joint configuration of 3-1-3 DOF. More specifically, the 3-1-3 configuration is a spherical-flexion-spherical joint configuration. This means that the roll for both the upper arm and fore arm are defined.

This also means that the position of the elbow is restricted to a circle in 3D space. This circle is the intersection of two spheres as seen in figure 4.8 and 4.9. One of the spheres has as centre the shoulder joint and as radius the upper arm link length. The other sphere has its centre at the wrist and has as radius the forearm link length.

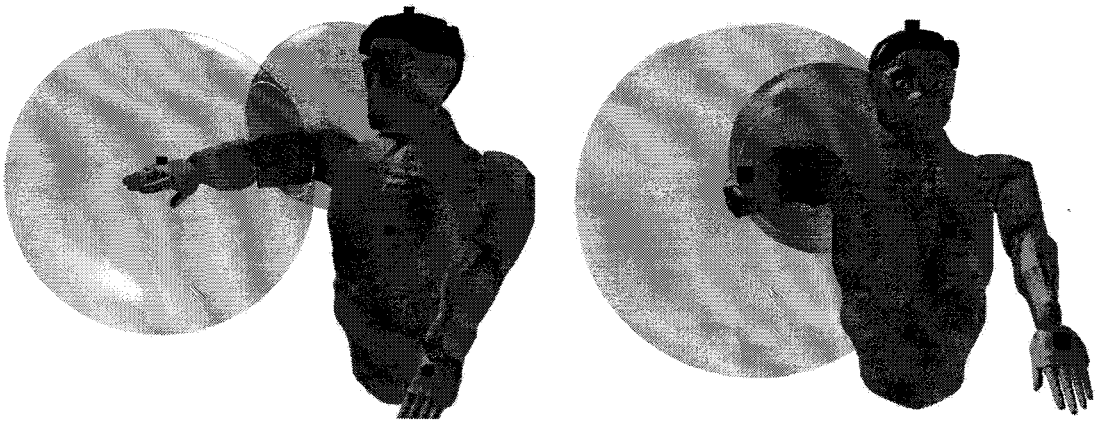


Figure 4.8 Intersection of the two spheres in a circle. The spheres each have as centre the shoulder and the wrist and as radius respectively the upper arm and the forearm. The elbow always lies in the intersecting circle (in white). The blue cubes represent the positions of the trackers.

The values for the centre and radius of the circle can be calculated as seen in figure 4.9 and in equations x, y ,z.

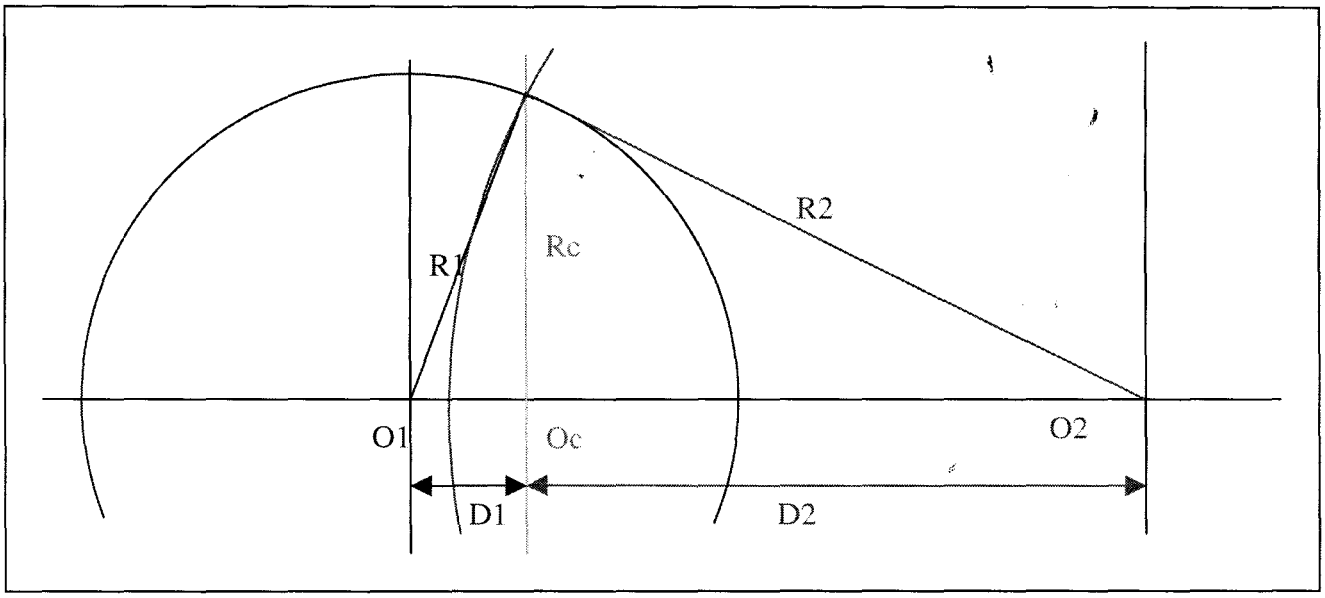


Figure 4.9 Intersection of the two spheres

The two spheres have centres at O1 and O2 and have as radii R1 and R2 respectively. They intersect in a circle that has its centre at Oc and a radius Rc. The distance between the centre of the circle and the centres of the two spheres are D1 and D2 respectively.

We can determine the centre and radius of the circle using simple geometry:

The centre (Oc) of the circle is given by:
$Oc = (O1 * R2 + O2 * R1) / (R1 + R2)$

The radius (Rc) of the circle is given by:
$Rc^2 = R2^2 - D2^2 = R1^2 - D1^2$

Although the position of the elbow is not known yet, we know that it has to lie somewhere in the intersecting circle. This idea is fundamental for the rest of this project and should be bared in mind at all times. As such, we denote this circle the 'elbow-circle' for further reference. However, since the position of the elbow is still unknown, further constraints are required to reduce the level of redundancy of the articulated chain and determine the position of the elbow precisely.

One of such constraints limits the position of the elbow to a certain arc in the 'elbow-circle'. Indeed, one can not rotate the elbow around the whole circle. The elbow in the human arm is limited to a certain arc which is smaller than 180°. We denote this arc as the 'elbow-arc' for further reference.

4.3.4 Various methods of solving the articulated chain

4.3.4.1 Eliminating a DOF

The easiest way to reduce the level of redundancy in a chain is to reduce the number of DOF. This method has already been used to eliminate the clavicle and in doing so, eliminate two DOF. By eliminating another DOF from the chain, the resulting chain is much easier to solve for. Six DOF chains are commonly used to represent the human arm [Hodgins, 98] when the arm is not required to perform all the movements of a real human arm.

When trying to eliminate a DOF from a chain, one needs to consider two main aspects:

- It is best to eliminate the DOF with the smaller movement range so as to keep a large number of possible moves.

- Choose the DOF to be eliminated according to the method above unless the DOF plays an important part in animating the articulated chain. In this case eliminate the next lowest movement range DOF.

In this case the DOF eliminated is the movement in the wrist along the axis perpendicular to the plane specified by the palm of the hand (as seen in Figure 4.10). This DOF has one of the smallest movement range in the arm chain and since it is next to the end effector, it's elimination should not affect the pose of the arm drastically.

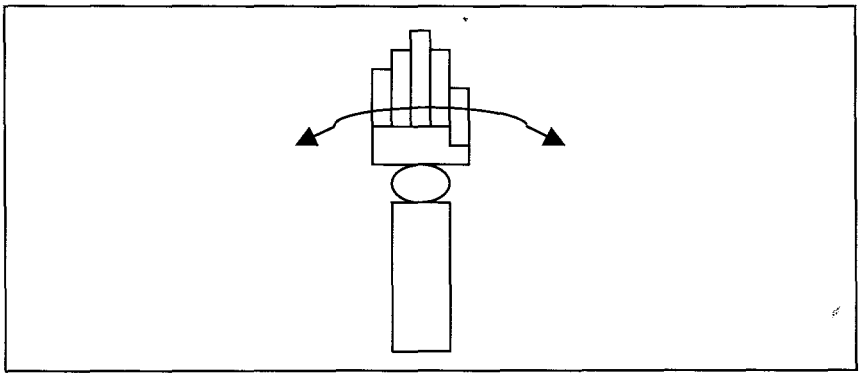


Figure 4.10 DOF eliminated from the wrist joint

It is hoped that because of these two features (the small movement range and being close to end effector), the resulting articulated chain will not differ much in its function.

When we eliminate the DOF mentioned earlier, we can find a plane that intersects the circle in at least one point. This plane is the plane that belongs to the hand and has the thumb of the hand as normal, when the thumb is in the open position (see figure 4.11).

Figure 4.11 Plane intersecting the elbow circle.

This method will produce at most two points of intersection. One of them will be the true position of the elbow. Using some simple constraints, it is possible to determine the correct position for the elbow between the maximum two points. These constraints are discussed in section 5.3.1.1.

4.3.4.2 Relationship theory

During the implementation of the first prototype, we noticed that there seemed to be a relationship between the elevation angle of the hand, and the position of the elbow in the 'elbow-circle'. It seems that the higher the hand is pointing, the lower in the circle the elbow is, and vice versa. However, there do seems to be some exceptions. The

relationship also seems to be independent of the azimuth or roll values of the hand. If this theory holds true, it could be used to increase the constraints on the articulated chain to provide a fully specified chain. Even if the relationship between the hand and the position of the elbow is not always correct, it is possible to use such a relationship to animate the articulated figure realistically (although not totally accurately).

If the relationship is determined to be a mathematical function of some sort, it could easily be incorporated in the inverse kinematics calculations without dramatically affecting the performance of such calculations.

4.3.4.3 The state-space theory

Since there seems to be a relationship between the elevation angle of the hand and the position of the elbow, there might be a similar relationship between the position and orientation of the hand with respect to the chest and the position of the elbow with respect to the chest. Although a mathematical function relating these values would be ideal, such a function can be extremely hard to determine. However, if such a function exists, then for a subset of input values close to a specific input point, the resulting output value should be close to the output of the specific input point. As such, we can generate a table that will hold output values for subsets of input values taken at regular intervals. The table holds the position of the elbow with respect to the position and orientation of the hand. The table is filled up with values for the position and elevation of the hand and position of the elbow taken using motion capture equipment. If there exists a one-to-one mapping between these two sets of data, or if the elbow lies in a relatively small area for each different position and orientation data of the hand, then we can assume that a relationship between these two exists. As such the table can then be used to determine the position of the elbow.

The creation of the table can be problematic because entries for data about the position and orientation of the hand and the position of the elbow for every possible combination of the articulated chain will have to be captured.

4.4 Summary

To reduce the redundancy of an articulated chain, constraints are applied to the chain. In this chapter we have provided a geometric constraint to reduce the position of the middle joint in a four link chain in 3D, matching the human articulated arm, to a limited subset in space of either two points or a circle. Various theories are then proposed to further reduce the redundancy of such chain. These theories are implemented and the results analysed in subsequent chapters.

Chapter 5. Implementation

5.1 Articulated figure animation

5.1.1 RhoVeR and CoRgi

The RhoVeR VR system is made up of a number of processing modules consisting of some data associated with a process. These processes run concurrently. This means that each object in the virtual world has its own process. As such, even a relatively small virtual world soon becomes saturated with small processes. This tends to reduce the performance of the system, especially when the system is run on one single machine. As a result of such reduction in performance, the motion capture system performs below the standard specification. This is reflected in a larger than normal latency and above all a smaller than normal update rate.

As such, the system was changed to a different VR system. The VR system chosen is CoRgi (Child of RhoVeR (gi)). This VR system is derived from the lessons learned from the implementation of RhoVeR. The concurrent processes have been eliminated, making the system far more efficient.

The system can still run on various separate machines. As such, we have the motion capture equipment system running in one machine, and supplying motion capture data via a network to another machine where the inverse kinematics system is running.

Using the CoRgi system, it is hoped that the motion capture equipment will deliver the standard specification update rates and latency.

5.1.2 Need for interpolation

Interpolation frames are usually needed when the animator does not want to specify all the frames in an animation. He simply just specifies a certain number of key frames and the animation software specifies the in-between frames. In our animation, the frames are specified by the data from the motion capture equipment. Therefore, if this data is supplied at a sufficiently fast rate, it will specify all the frames needed in the animation. This leaves no need for any interpolation. Using the RhoVeR system, we achieve around ten frames of motion capture data per second. This is adequate for animating an avatar in a virtual environment, but is inadequate for film animation (which requires thirty frames per second). By moving into the CoRgi VR system, we hope to achieve the data frame rate needed for achieving film animation rates.

5.2 Motion capture

5.2.1 Attaching the trackers to the body

The motion capture trackers should be attached as firmly as possible to reduce errors in the position and most importantly in the orientation. It is indeed quite easy to slightly offset the orientation of the trackers if they are not attached firmly to the body. It is also important to keep the user as comfortable as possible. One can use commercially available equipment to secure the trackers to the body as seen in figure 5.1.

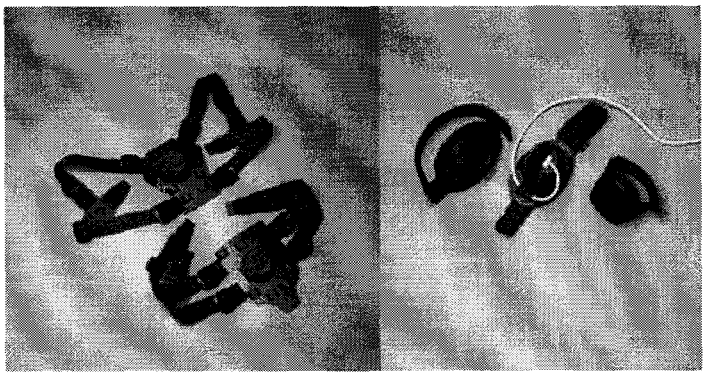


Figure 5.1 Commercially available equipment to achieve a secure and comfortable fit of the motion capture trackers to the body.

Alternatively, one can build similar equipment out of non ferromagnetic materials such as velcro, lycra, and wood for the boards. The trackers can then be stuck to the wooden boards. The wooden boards are held on the body with straps made of lycra. The straps are secured to each other with velcro. The wooden boards can be slightly padded to as to increase the degree of the comfort of the user.

5.2.2 Difficulty in achieving the proper position for the trackers

Since the trackers are attached to the surface of the body, while the links and joints of a real person are inside the body, there exists a certain offset between the captured position of the links and the actual position of the bones that the links represent. The positions of the captured links must therefore be adjusted. Using the orientation of the trackers, one can easily translate the position of the trackers by a certain offset so as to correspond with the actual position of the skeleton.

5.3 Inverse kinematics

5.3.1 Eliminating a DOF

As explained in 4.3.4.1, a plane defined by the position and orientation of the hand will intersect the 'elbow-circle' at most two points.

5.3.1.1 Implementation

These two points are found by taking the equations of the plane and the 'elbow-circle' and intersecting them. To simplify these equations, the following transformations are first applied. The position of the shoulder is translated to the origin of the coordinate frame. The whole system is then rotated so that the position of the wrist lies in one of the axis (the z axis, for example). The equation of the circle in 3D space is then simplified to the equation of a circle in a 2D plane as follows.

$$X^2 + Y^2 = Rc^2$$

equation 5.1

And

$$Z = Oc = w$$

equation 5.2

(where w is the value along the z-axis for the centre of the circle)

The plane defined by the position and orientation of the hand is also translated and rotated by the same amounts as for the circle. The equation of the plane is given by:

$$aX + bY + cZ = d$$

equation 5.3

The intersection of this plane and the 'elbow-circle' is found by solving for the equations 5.1, 5.2 and 5.3 as follows:

$$X^2 + Y^2 = R^2$$

equation 5.4

$$Z = w$$

equation 5.5

$$aX + bY + cZ = d$$

equation 5.6

From equations 5.5 and 5.6:

$$aX + bY + cw = d$$

$$aX = d - cw - bY$$

$$a^2X^2 = (d - cw - bY)^2$$

equation 5.7

From equation 5.4:

$$X^2 = R^2 - Y^2$$

$$a^2X^2 = a^2(R^2 - Y^2)$$

equation 5.8

Combining equations 5.7 and 5.8:

$$(d - cw - bY)^2 = a^2(R^2 - Y^2)$$

$$(d - cw)^2 - 2(d - cw)(bY) + b^2Y^2 = a^2(R^2 - Y^2)$$

$$d^2 - 2cdw + c^2w^2 - 2(d - cw)(bY) + b^2Y^2 = a^2R^2 - a^2Y^2$$

$$(a^2+b^2)Y^2-2(bd-bcw)Y+d^2-2cdw+c^2w^2-a^2(R^2-w^2)=0$$

This is a quadratic equation therefore:

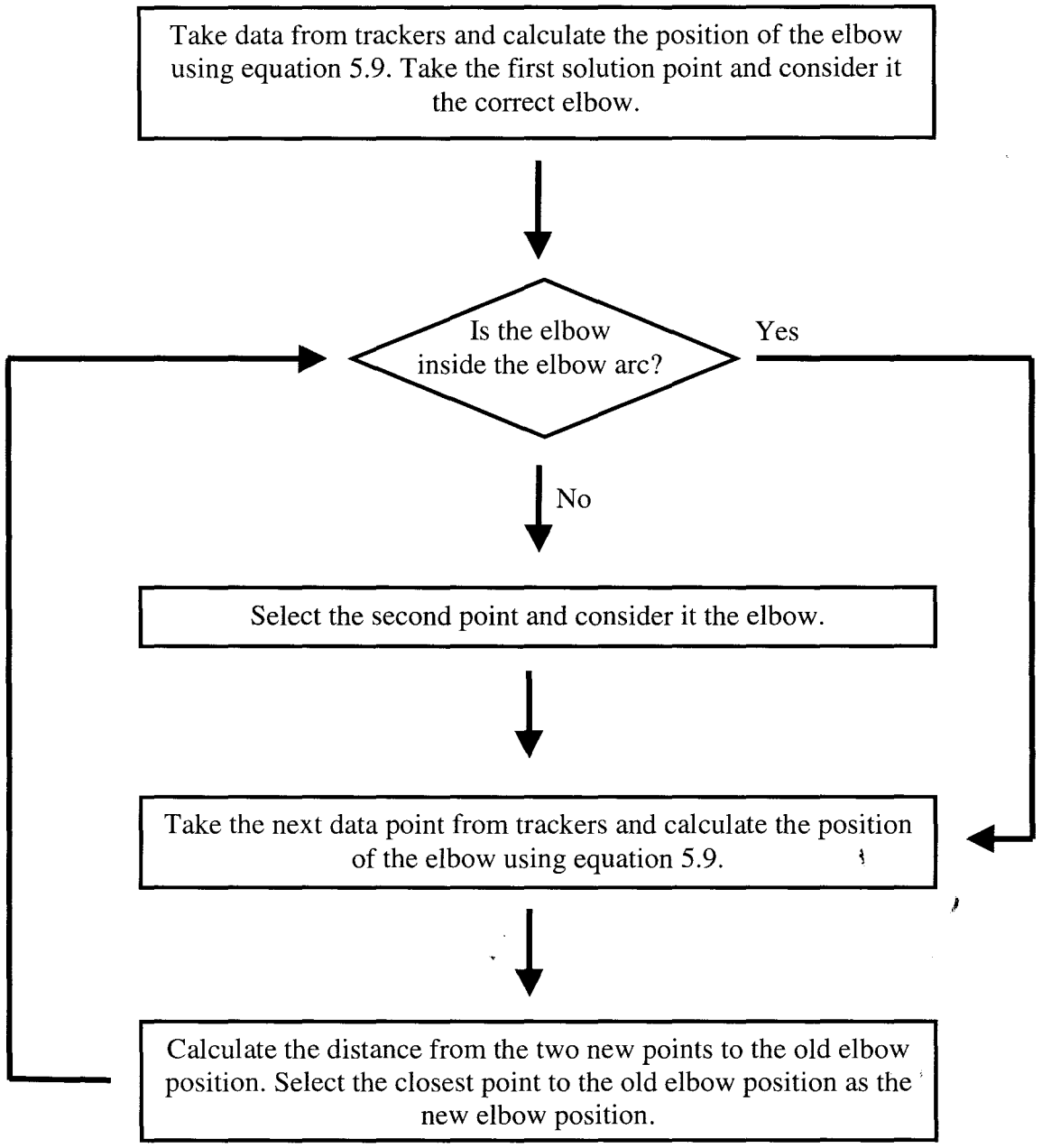
$$Y=\left[2(db-bcw)\pm\sqrt{4(bd-bcw)^2-4(a^2+b^2)(d^2-2cdw+c^2w^2-a^2R^2)}\right]/[2(a^2+b^2)]$$

And substituting for the two Y values in:

$$X=\sqrt{R^2-Y^2}$$

we get two values for X.

Out of these two points, only one is the correct position of the elbow. The easier way to determine the correct point is to apply some simple constraints to the position of the elbow. We know that the elbow can only move in an arc that is at most 180 degrees. We thus constrain the elbow point to be in this 'elbow-arc'. If the point considered as the elbow gets out of the 'elbow-arc', then we know that we have the wrong point and that the second point is the correct one. For the next data frame, the point closer to the previous correct point will become the new correct elbow. This point is checked to make sure that it is inside the 'elbow-arc'. The algorithm is explained below.



Algorithm 5.1 Solving for the two possible elbow positions

At the beginning of the animation the user just needs to twist his wrist a few times. This will force one of the elbow points out of the elbow-arc thus revealing the correct elbow point.

5.3.2 Hand elevation

As stated in 4.3.4.2, the elevation of the hand can possibly be used as a constraint to determine the position of the elbow in the 'elbow-circle'.

The hypothesis is that the elevation of the elbow (E) in the 'elbow-circle' with respect to the horizontal plane is a factor of the elevation of the hand (H) also with respect to the horizontal plane.

$E = k.H + \text{constant}$

To determine the relationship, if there is any, more than of 8000 data points were taken over a period of around nine minutes, for the position and orientation of the hand, the shoulder and the elbow while performing various common movements. The data for the true elbow position and for the hand elevation are then used to calculate the respective elevations H and E. These two elevations are then compared to determine if a relationship between the two exists. If so, the hand elevation can be used to determine the position of the elbow in the 'elbow-circle' and thus in 3D space.

Although we do not expect the relationship to be exact every time, we hope that it will be a good approximation to a valid position for the elbow, if not the correct one. After an analysis is done on the motion capture data points, the system will be implemented to see if the articulated figure looks realistic, even if the elbow might not be in the actual position.

5.3.3 State space

In 4.3.4.2 we decided to investigate if a relationship exists between the elevation of the hand and that of the elbow. Similarly, in 4.3.4.3 we decide to investigate if there is a relationship between the position and orientation of the hand (with respect to the shoulder) and the position of the elbow.

Therefore, we create a six-dimensional grid, using the position and orientation of the hand as reference. We then store the corresponding average position of the elbow in the grid. The grid is divided into 256 units for the maximum range of values for the position of the hand, and into 64 units for the orientation of the hand. This gives us a 256x256x256x64x64x64 grid.

Ideally, one would prefer a function that would give us the position of the elbow, given the position and orientation of the hand. However, determining such function is a highly complex task. Since this is just a test to determine if these constraints are viable or not, we decide to use experimental data to create a table that will mimic such function.

When implementing the system, creating a six dimensional array of [256x256x256x64x64x64] size is not viable due to the astronomical amount of memory required by the structure. Also, since we only have around 8000 data points, most of the array would be empty. As such, we have decided to use two octrees: one to separate the position, and a second one to separate the orientation. This dramatically reduces the amount of space used to store the data structure.

The drawback is that to find the corresponding average position of the elbow for a certain hand position and orientation, we have to go down two octrees of depths of eight and six respectively.

5.4 Summary

In this chapter, we propose the use of a different VR system to the one used for the implementation of the first prototype as discussed in chapter three, with the aim of improving the performance of the system. We also discuss issues about the usage of the motion capture equipment. Finally, we discuss issues dealing with the implementation of the three proposed inverse kinematics methods: the elimination of a DOF, the hand elevation relationship and the state space method.

Chapter 6. Results

6.1 Articulated figure animation

Using the new CoRgi VR system has increased the overall frame rate of the animation to around sixty frames per second.

The avatar uses a model with a reduced number of polygons, thus allowing for the rendering of the system in low-end machines without 3D graphics support. For an improved avatar model, the use of some kind of 3D graphics hardware is recommended.

The avatar used is composed of separate body parts that are used to represent the corresponding links in the articulated figure.

6.2 Motion capture

The new CoRgi VR system improves the update rate of the motion capture considerably, under optimum conditions.

When the trackers are close to the maximum tracking range, the system looses accuracy and the update rate decreases dramatically. This also happens when the system is affected by electromagnetic and ferromagnetic interference.

Still, to animate the human upper body, the motion capture system is adequate.

However, the motion capture system fails in providing a high level of accuracy as required for the inverse kinematics method discussed in section 6.3.1.

6.3 Inverse kinematics

6.3.1 Eliminating a DOF

In section 5.3.1, we described the implementation of the six DOF articulated human arm. In this section we look at the results from the implementation.

6.3.1.1 Problems encountered

6.3.1.1.1 Correct and incorrect elbow swap

During the animation, it is possible for the two points to swap, thus giving the incorrect elbow position. This would happen if the distance between the previous correct elbow position and the actual incorrect elbow position (distance B) is smaller than the distance between the previous correct elbow position and the actual correct elbow position (distance A). This would happen as the plane intersecting the 'elbow-circle' approaches the tangent to the circle. When the plane coincides perfectly with the tangent, the two points are identical. However this rarely happens. What is more likely to happen is that as the plane approaches the tangent, a fast twist of the wrist will cause the distance B to be smaller to the distance A. This needs to happen in between two data frames. Since the data frames come at thirty frames per second, the fast twist has to be quite fast. As such this problem is infrequent. When it happens, the correct position is corrected the moment the wrong position steps outside the elbow-arc as explained in section 5.3.1.1. This however means that sometimes (although quite infrequently) the elbow is in the wrong position and tends to make the articulated figure appear awkward.

6.3.1.1.2 Practical problems

Although this method would be satisfactory in theory, in practice one finds that it lacks satisfactory results. This is due to various factors.

- Error in the positioning of the tracker at the hand. The tracker needs to be perfectly aligned with the length of the hand-forearm section. Failure to do so will result in the intersection between the plane and the circle being offset by a small amount. This causes one of two errors discussed later. Although the trackers are attached to the hands quite firmly as discussed in section 5.2, the movement of the user will eventually lead to the misalignment of these trackers, resulting in the errors discussed later.

- Error in holding the hand in a steady position. Since we have chosen to ignore a flexion DOF of the wrist, the user must not under any circumstances move the wrist along that DOF. Failure to do so will result in the two types of errors discussed later. In practice holding the wrist that steady is almost impossible to achieve thus the errors do occur.

- Errors in the calibration of the length of the upper arm, forearm, and chest segments. If a link in the virtual articulated figure is shorter or longer than its counterpart in the real figure, the two types of error will result again. This is due to the user trying to keep the tracker aligned, but because of the discrepancies in the length of the link, the elbow-circle will be at the wrong place, and thus the elbow-circle will be intersected by the hand plane at a certain offset from where it should intersect it.

In all three cases above, the errors resulted in the plane intersecting the 'elbow-circle' at a certain offset from the correct intersecting position.

These offsets will cause two types of errors: Either the plane will intersect the 'elbow-circle' at the wrong position, or it will not intersect the 'elbow-circle' at all.

If the plane is close to the tangent of the 'elbow-circle', even a small offset might move the plane sideways causing it to miss the 'elbow-circle' totally, thus resulting in no intersection.

Similarly, if the offset is towards the 'elbow-circle', the plane will intersect the 'elbow-circle' at the wrong position. However, in this case there will be a solution to the system, although an incorrect one. Also, the fact that the offset might be relatively small does not mean that the distance between the correct and incorrect position is also small. This is illustrated in figure 6.1.

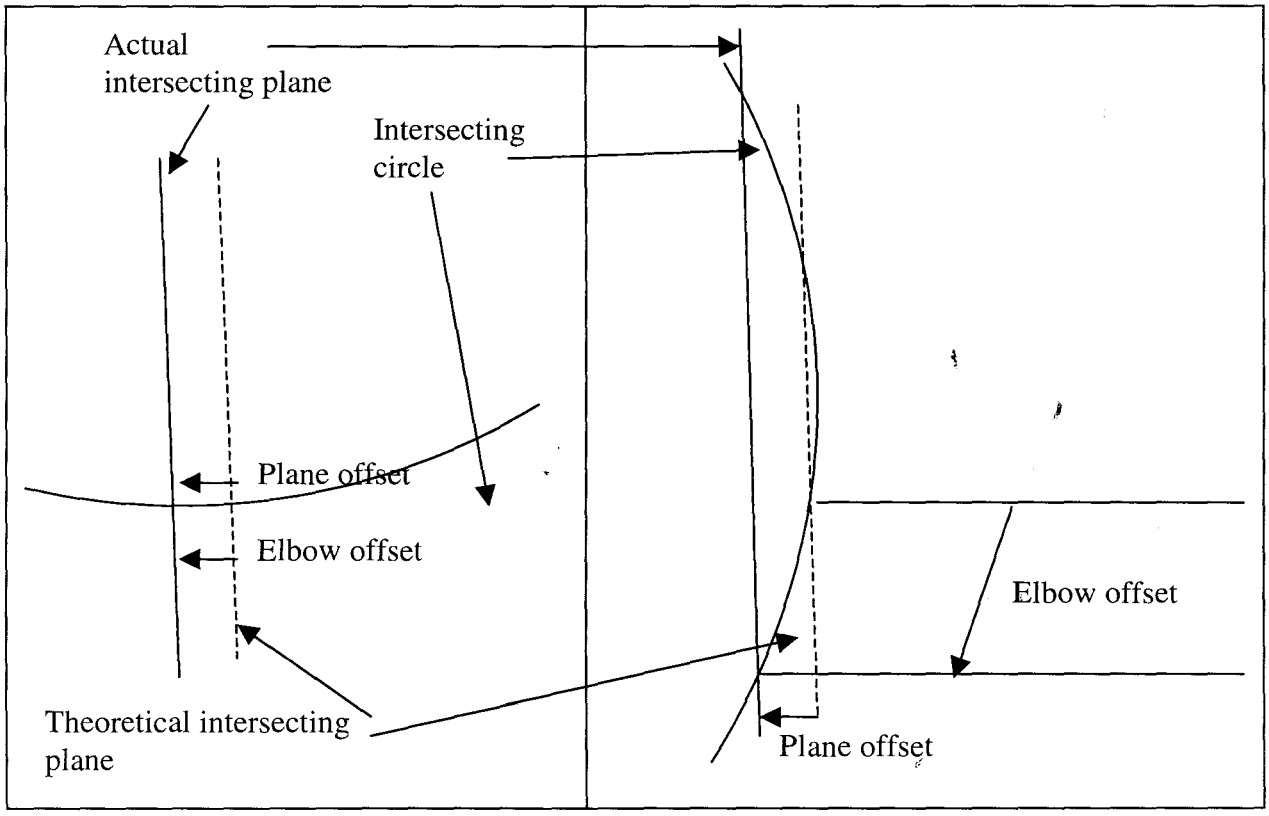


Figure 6.1 Two different types of intersections. Notice the difference in the length of the offsets.

As seen, the drift induced by the error is dependent on the way that the plane intersects the 'elbow-circle'. If the plane intersects the 'elbow-circle' perpendicular to the tangent at the point of intersection, then the offset in the erroneous elbow position is the same as the offset in the plane. However, if the plane intersects the circle in an arc that is parallel to the tangent, then the offset in the erroneous elbow position is larger than the offset in the plane, thus causing an increased drift in the elbow position.

6.3.1.2 Summary

Due to the problems explained in sections 6.3.1.1.1 and 6.3.1.1.2, the system is not suited for our purpose. The animation of the articulated figure tends to produce non-realistic elbow movements.

The problems with this system are mainly due to the need for a very precise motion capture system, as well as a thorough calibration of each individual user.

6.3.2 Hand elevation

In section 5.3.2, we explained that there might be a relationship between the elevation of the hand (H) and the elevation of the elbow (E).

The assumption is that

$E = k.H + \text{constant}$

6.3.2.1 Graph representation and analysis

After capturing a large number of data points of the position and orientation of the hand and the elbow, the elevation of the hand and that of the elbow is plotted against time as seen in figure 6.2 and in appendix B. Appendix B contains the same graph as in figure 6.2 but separated into various graphs so as to increase the time resolution and make it easier to recognise the features of the two curves.

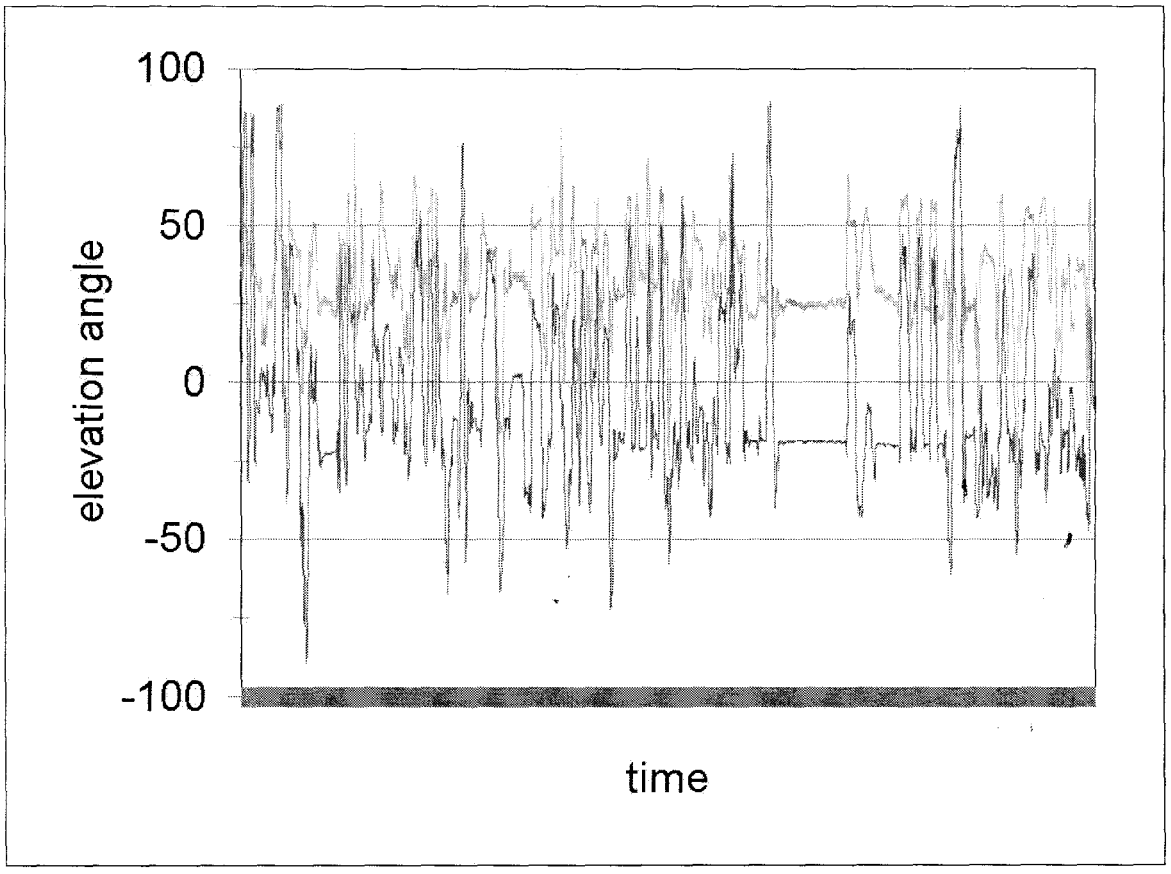


Figure 6.2 Elevation of the hand (in red) and of the elbow (in blue) over time.

From these graphs, one can appreciate that almost all of the features found in the hand elevation data are also found in the elbow elevation data. In the next section we analyse the graphs in appendix B while noting relevant features of the data plots.

6.3.2.2 Graph analysis

In chart B.2, from time 0 to time 132, one can clearly see that although the amplitude of the two lines is different, the pattern of the two lines does correspond. From time 132 to time 287, no correspondence can be found. This section will give an improper elbow elevation from the hand elevation. From time 287 to time 675, the two lines seem to correspond although again, the difference in amplitude is not constant. From time 675 to time 720, no correspondence can be found, although this section is rather small. Again from time 947 to time 977, the two curves differ considerably.

In chart B.3 the two lines correspond quite accurately except for the following small time intervals: from 1308 to 1411, 1696 to 1720, 1750 to 1789 and 1909 to 1969.

Similarly in chart B.4 the two lines correspond quite accurately except for the following small time intervals: from 2075 to 2095, 2333 to 2412 and 1705 to 1723.

Similarly in chart B.5 the two lines correspond quite accurately except for the following small time intervals: from 3049 to 3069, 3316 to 3337, 3450 to 3470 and 3533 to 3562.

Similarly in chart B.6 the two lines correspond quite accurately except for the following small time intervals: from 4367 to 4410 and 4781 to 4833 only.

In chart B.7 the two lines correspond quite accurately all over.

In chart B.8 the two lines correspond quite accurately except for the following time interval: from 6831 to 6960 only.

Similarly in chart B.9 the two lines correspond quite accurately except for the following small time intervals: from 7135 to 7247, 7350 to 7417, 7647 to 7753 and 7952 to 8017.

From these graphs, we can conclude that there does seem to exist a relationship between the hand elevation and the elbow elevation, although this relationship is not constant throughout the whole graph. The relationship does not hold true for certain time intervals in the graph, and when it does, the difference in amplitude between the hand elevation and elbow elevation is not constant. However, these time intervals are relatively short. One must them decide if it is worth using a constraint that will work for most of the time but that may give, for small periods of time, positions of the elbow that are quite different from the actual position of the true elbow.

6.3.2.3 Regression analysis

We thus decided to do a regression analysis. The results are as follows:

x coefficient: 1.125435
y intersect: -44.1961
R squared: 0.357302

This means that to a certain (although quite low) degree,

Elbow elevation = 1.125435 * hand elevation – 44.1961 Equation 6.1

Figure 6.3 shows this equation (in blue) plotted against the real elbow elevation, as well as the real elbow elevation line (in red).

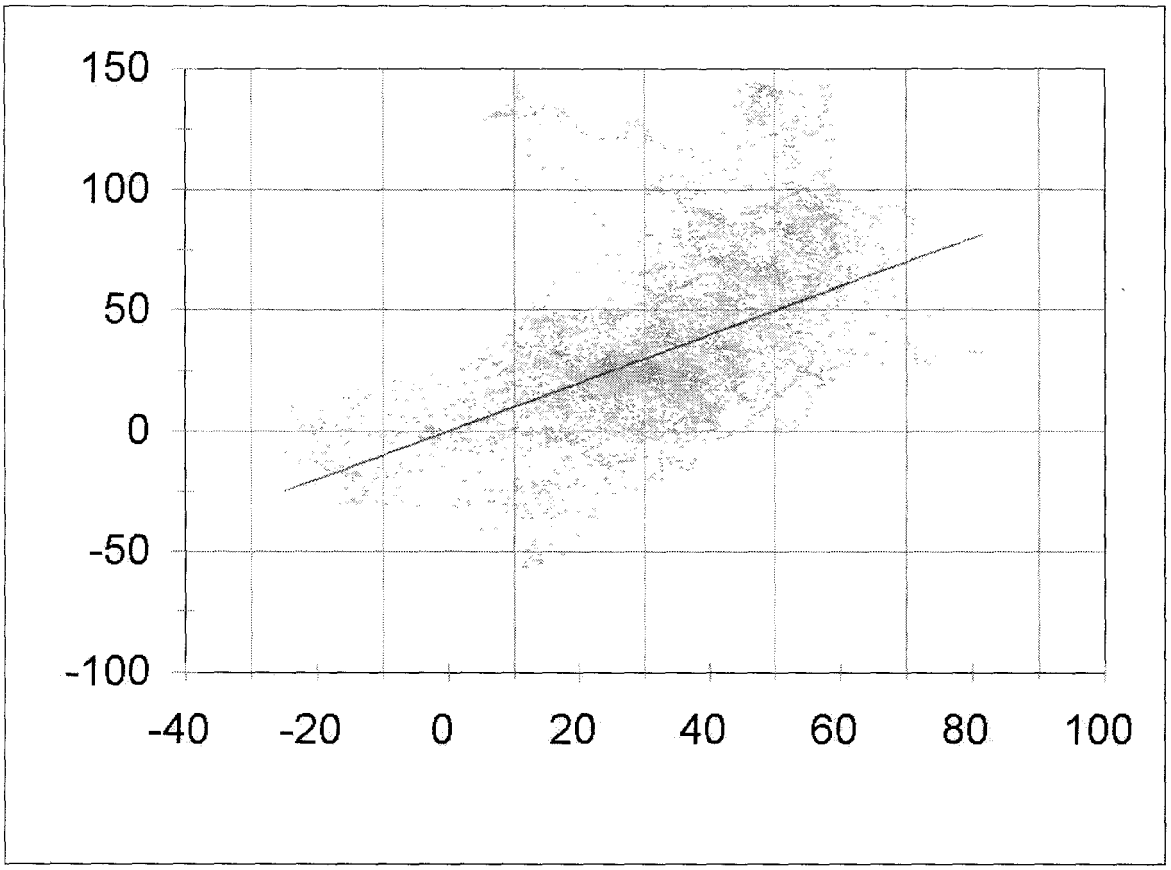


Figure 6.3 Calculated elbow elevation (in blue) vs. real elbow elevation (in red).

From figure 6.3, one can appreciate that the calculated elbow elevation data seems to follow the direction of the real elbow elevation data line, although the difference in amplitude is rather large.

However, these results are based on the whole set of data, and we have seen in section 6.3.2.2 that some segments of data do not follow the proposed relationship. As such, we decide to do a regression analysis on the data except for the data highlighted in section 6.3.2.2 as being highly inconsistent. The results are as follows:

x coefficient: 1.364432

y intersect: -52.4902

R squared: 0.517896

As can be seen from the value of R squared, the fit is much better than that of the previous regression. Still, there is a large chance for error. The new equation looks like this:

Elbow elevation = 1.364432 * hand elevation – 52.4902 Equation 6.2

Using these values means that most of the time, the movement of the figure will be representative of the movement of the real person, but it also means that when an error occurs, it will be larger than with the previous set of regression values.

6.3.2.4 Amplitude change

In section 6.3.2.2 and 6.3.2.3, we found that although the elevation of the hand does seem to have a relationship with the elevation of the elbow for most of the time, the change in amplitude does not seem to be constant. We thus decided to use other data from the hand such as hand roll and hand azimuth to see if we can improve the fit. The hypothesis is that the elbow elevation (E) is related to the hand elevation (H) as follows:

E = ("hand attributes" * k).H + constant

Where "hand attributes" is either the hand roll, or hand elevation, or a combination of both. This would decrease the difference in amplitude between the hand elevation and the elbow elevation.

Unfortunately, no such relationship could be found. This leaves us with the results from section 6.3.2.3. Although these results are not as promising as the ones from section 6.3.2.2, we decided to implement the system to see how it would perform in practice.

6.3.2.5 Implementation results

When the system is implemented, using equation 6.2, the movement of the elbow is accurate for most of the poses that the user does. However, it still fails in some of the poses. When the system fails, the position of the elbow is totally incorrect, with very little resemblance to the actual pose. Luckily, the failure rate is quite low, and only happens with non-common poses.

The resemblance in movement between the human figure and the animated character is definitely acceptable for use as an avatar in a virtual environment (as can be seen in figure 6.4 and appendix D). The system can also be used in proper 3D animation, although the animator will have to put up with the overhead of manually correcting the figure when the errors occur.

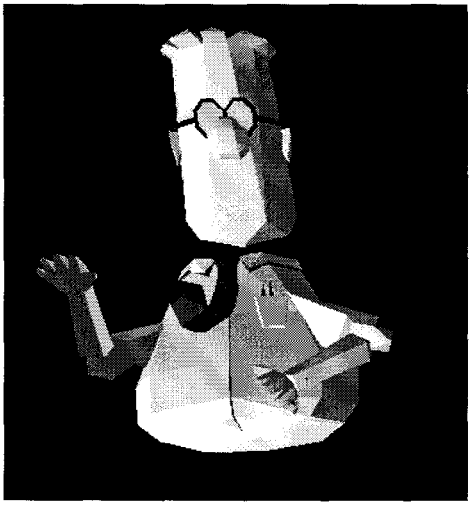


Figure 6.4 The user's avatar using the "hand elevation" method

6.3.3 State space

In 5.3.3 we decided to divide the position and orientation of the hand in a 256x256x256x64x64x64 six dimensional grid. The size of the grid (256x256x256x64x64x64) was chosen to be fine enough to be representative of the actual position and orientation of the data while still being coarse enough to allow enough data points in the grid spaces so that the mean is representative of the data set. We then used motion capture data to get the average elbow position for each different space state. We used the same set of data as used in the hand elevation method.

To analyse the results from the implementation, consider the curve of figure 6.5.

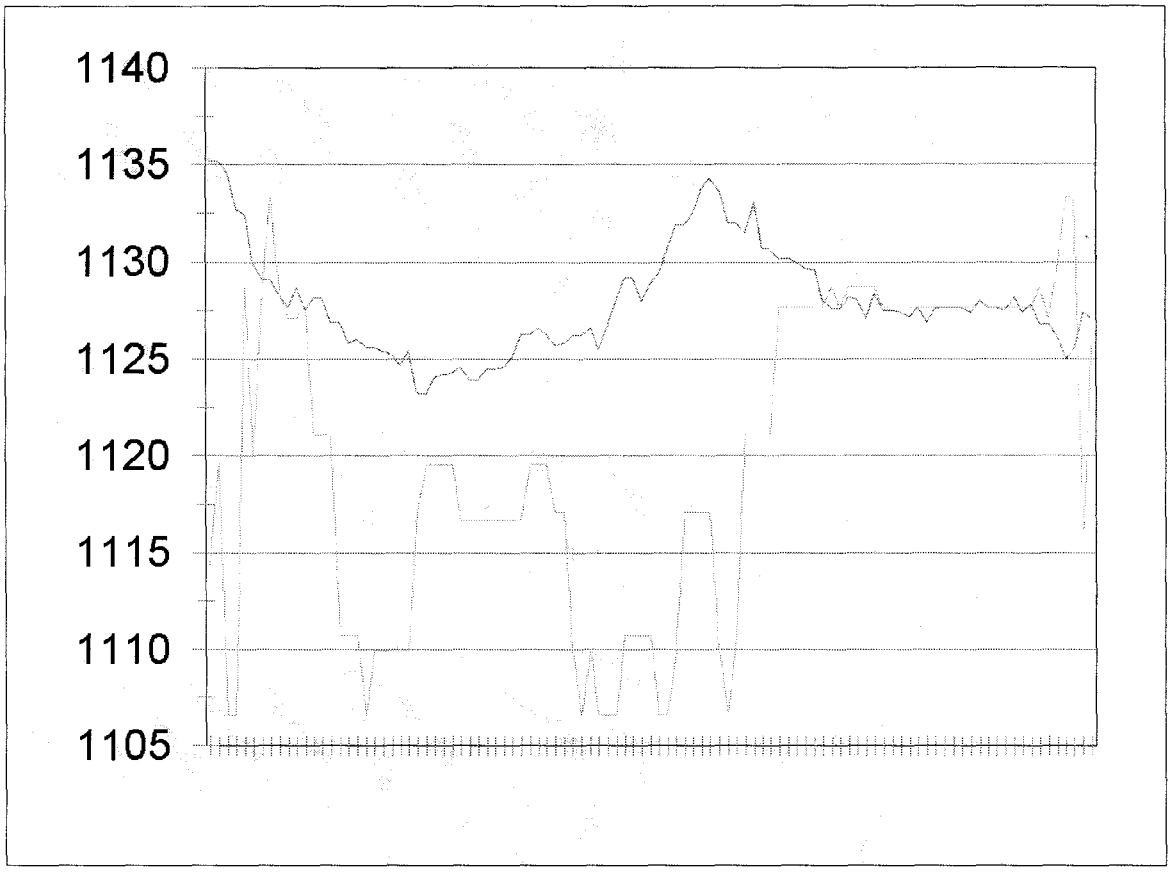


Figure 6.5 Real position of the elbow in the x-axis (in red) vs. calculated position of the elbow in the x-axis (in blue) from time 3581 to time 3683. The y-axis represent the position of the elbow with an amount of 20 units roughly equivalent to two centimetres in real life.

This curve represents a section of the overall curve displayed in appendix C. Most specifically, it represents the time interval from 3581 to 3683. The red line represents the actual elbow position in the x-axis while the blue line represents the corresponding estimated elbow position in the x-axis for the same time interval.

This portion of the curve was selected because the calculated average elbow position in this curve is made up of at least twelve elbow positions for each entry. This is not the case in most of the curve. As stated earlier, the data is separated in a

256x256x256x64x64x64 grid. This means that there is an astronomical number of entries for the grid. As such, a large amount of data is needed to generate a grid with at least a decent amount of entries for the most common movements of the chain.

The amount of entries for each elbow position is displayed in yellow in the graphs of appendix C. One can clearly see that most of the time the yellow curve is on one. When the yellow curve is on a very low number, the calculated elbow position is not representative of the movement of the real elbow (unless the movement of the arm corresponds exactly with the recorded movement during calibration).

The problems described in sections 6.3.3.1, 6.3.3.2, 6.3.3.3 and 6.3.3.4 are applicable when the calculated elbow position corresponds to a space grid where the average comes from a decent number of hits (at least twelve hits). This is usually the case for commonly used moves and poses.

One can clearly appreciate various points:

6.3.3.1 Non matching positions for the real and calculated elbow

First of all, one can clearly see that the calculated elbow position does not match the actual elbow position. This is due to the fact that the calculated elbow position is an average of all the elbow positions corresponding to the actual position and orientation of the hand. In actual fact, the difference in not that striking. The graph of figure 6.5 represents a zoomed up part of the curve. A difference in position of twenty units in the y-axis in the graph is roughly equivalent to three centimetres in real life, so an error in the elbow position of three centimetres is not that significant. In fact, the average difference between the real position and the calculated one is quite small. It is 2.4825 units. Of course, this is for the whole of the curve, and is not representative of the difference in elbow positions that a user would find when using the system. However, the difference for the section of the curve with an average calculated from more than twelve data points

is actually 4.9445 units (the section corresponding to points 5300 to 5700 was not considered because this area represents the hand at rest with minor hand movement. This can be seen in graph B7 of appendix B). This corresponds in real life approximately to one and a half centimetres. Of course the system will only be this accurate when there is an entry corresponding to the hand position and orientation which has enough data points to make a good average.

6.3.3.2 High differences between adjacent grid spaces

The differences between adjacent grid spaces tend to be rather large. As can be seen from the graph, most of those differences are around two centimetres in real life. This is also due to the average of the elbow positions, because the various paths of the hand leading to a certain position and orientation of the hand are not always the same, and might be extremely different for each other, thus giving very different elbow position average for adjacent elbow grid spaces.

This gives the elbow a very strange and unrealistic shaking movement, as the arm moves. Unfortunately, extensive training will not improve this since it is caused by the different paths that the elbow can possibly take.

6.3.3.3 Discreteness of the values in adjacent grid spaces

The values for the calculated elbow are too discrete. As can be seen from the figure, the curve tends to be steady in one value and then suddenly jumps onto another value. This gives the curve somewhat jagged edges. This accentuates the perception of the problem discussed in section 6.3.3.2.

Increasing the resolution of the space cells can reduce this jagged appearance. However, this would reduce the number of hits per space box, thus reducing the validity of the average elbow position for the cells.

Another solution would be to smoothen out the jagged curve using some kind of interpolation methodology. This would solve the problem of the jagged appearance of the curve, thus reducing the perception from users of the shaking elbow problem (as seen in section 6.3.3.1). However, it would not reduce the shaking of the elbow. Instead, the elbow will now have a smoother, and less noticeable, shaking.

6.3.3.4 Problems caused by the difference in position between the real and calculated elbow position

As seen in section 6.3.3.1, the calculated position of the elbow will sometimes be different from the actual elbow position. Although the difference between these two is usually quite small, and thus visually the elbow looks in the correct position, this poses a problem with the articulated figure itself. Since the calculated elbow position is now different from the real elbow position, the lengths of the new upper arm and forearm links are now different to the real lengths.

To solve this problem, one can map the calculated elbow position to a corresponding elbow position in the elbow circle. Once the position of the elbow is inside the elbow circle, the lengths of the links will be accurate.

Alternatively, one could use some kind of stretching mechanism to stretch the graphical representation of the limbs to represent the new lengths. Since the difference in error position is usually quite small, the difference in link lengths is also quite small, so the stretching will not be noticeable. However, if the error is large, then the stretching will become more noticeable.

One could also use an articulated figure that has each limb separated. As such, a change in the length of the links will result in a figure that has the links slightly closer or farther

apart than it should. Again, with a small change in link length, the error will not be noticeable. For our implementation, we decided to use this method.

6.3.3.5 Overall results

The problems described in sections 6.3.3.1, 6.3.3.2, 6.3.3.3 and 6.3.3.4 are applicable when the calculated elbow position corresponds to a space grid where the average comes from a decent number of hits (at least twelve hits). This is usually the case for commonly used moves and poses. However, if the movement is not used frequently, then the corresponding space grid might have a poor average for this position. This will produce an exaggeration of the problems discussed earlier, with very poor performance for the likeliness of the animated figure.

The problem is even greater when there is no corresponding elbow position for a hand position and orientation. One solution to this problem is to use the closest elbow position available. This usually produces large real elbow – calculated elbow differences, causing the same errors as those discussed in the previous paragraph.

The only solution for the types of errors illustrated in the previous two paragraphs is to get a much larger motion capture sample with a much larger option of movement ranges. However, this is quite tedious and time consuming.

6.3.4 A comparison of the three inverse kinematics methods used

This section compares the three inverse kinematics methods used with respect to various criteria. These criteria are accuracy, robustness, ease of use, required processing power, and reflection of the user's mannerisms in the avatar.

6.3.4.1 Accuracy

The "eliminating a DOF" method has the best accuracy possible, if the problems discussed is section 6.3.1.1.2 can be avoided, since the calculated position of the elbow is mathematically correct. However, avoiding these problems is almost impossible with the current system. As such, the first method results in the worst accuracy for the movement of the elbow.

The "hand elevation" method has a decent accuracy for the elbow position. The calculated elbow position always lies in the elbow circle and is as such always correct (although not always accurate). The accuracy of the position of the elbow tends to be quite good for most of the movements of the arm. However, when the elevation of the hand approaches the vertical axis and the hand is in certain positions, the position of the elbow is totally different from its intended position. Fortunately, this does not happen frequently. An average of the difference in elbow elevation between the real elbow elevation and calculated elbow elevation gives an angle of 14.97 degrees. Although this seems to be quite a large angle, it was calculated over the whole curve, including the poses that give totally incorrect elevations between the real and the calculated elbow elevation. This means that the average for the more accurate poses should be lower. Still the difference is much higher in the statistical analysis than it appears when the implemented system is used in a virtual environment.

The "state space" method has the best accuracy when the position and orientation of the hand have a good matching average elbow position as discussed in section 6.3.3.1. Although the position is rarely perfectly accurate, it is usually a very close approximation to the correct one. This accuracy tends to decrease dramatically when the system hits an elbow with a bad average (an average made of very few points). It still is considerably more accurate than the other two methods.

6.3.4.2 Robustness

Due to the problems discussed in section 6.3.1.1.2, the "eliminating a DOF" method sometimes tends to give no solution for certain inputs.

Due to the fact that the inverse kinematics method to find the elbow in the "hand elevation" method is continuous, an input to the system will always produce an output. Moreover, the elbow position will always lie in the elbow circle, thus being a possible elbow position.

Because of the large amount of data required to fill a 256x256x256x64x64x64x grid, as well as the large amount of possible poses for the articulated arm, some of the grids will not have an elbow position. As such the "state space" method will lack outputs for some inputs.

6.3.4.3 Ease of use

As seen in section 6.3.1.1.2, the "eliminating a DOF" method requires very precise calibration. Although the calibration does not require a long amount of time, the precision of the calibration of the lengths of the links of the articulated figure, as well as the precision of the attachment and calibration of the trackers must be extremely precise to achieve correct results. Also the system needs to be calibrated every time that the system is used.

The "hand elevation" method requires the least amount of calibration. Once the values for k and the constant in E = k.H + constant (see section 5.3.2) have been determined, they can be used with most users. The only calibration required then is the length of the forearm and upper arm links. However, if the user is much smaller or larger than the user used for the calibration, the results will deteriorate. Still, to find the values for k and the constant just requires some simple calibration with minimal precision, although for a long

time. For the prototype the calibration session lasted around 15 minutes (including nine minutes of actual motion capture). Also, once the values for k, the constant and the lengths of the links are determined for a user, they can be used repeatedly, thus eliminating the need for calibration every time that the system is run.

The "state space" method requires a similar type of calibration as for the "hand elevation" method. However, it requires a much longer motion capture time to provide a large space of possible solutions, as well as providing good quality solutions. Also, once the octrees are populated with the data for a user, they can be used again and again, thus eliminating the need for calibration every time that the system is run.

6.3.4.4 Processing requirement

This section compares the amount of processor power required by each inverse kinematics method. All three kinematics methods have a constant order of complexity.

The method with the smallest processor power requirement is the hand elevation. This just requires determining the elbow circle, determining the elbow elevation angle from the hand elevation, and the corresponding position in the elbow circle for the elbow elevation angle.

The method with second smallest processor power requirement is the "eliminating a DOF" method. This method requires determining the elbow circle, the equation for the intersecting plane, and the intersection between the elbow circle and the intersecting plane.

Lastly, the method that requires the most processor power is the "state space" method. This requires the setup of the octrees and calculation of the average for each data point, in the initialisation of the system. When running, the system requires to look through the two octrees for the elbow values. This means going down an octree of depth eight and

then an octree of depth six. As such at every new hand data point, the system must go down the octree to a depth of fourteen. This is done quite fast. Results show that a Pentium 133Mhz CPU can do more than three thousand searches per second down the two octrees. This means that any slowing down of the system is not due to the complexity of the inverse kinematics system.

6.3.4.5 User's mannerisms

This section describes if the method used is capable of accurately reproducing the user's mannerisms when the user moves. When we say user's mannerisms, we mean any specific attribute in the position of the elbow that is specific to the user.

Since the "eliminating a DOF" method is a true reflection of the user's pose, the user's mannerisms will be correctly reflected when using this method, as long as the method works correctly.

The hand elevation method will only be able to reproduce the user's mannerisms to a very small degree. This will be reflected in the values of k and the constant, which seem to be specific to the user. This means that if values for k and the constant used are those of a different user, then the articulated figure will have the mannerisms of the other user, and not those of the actual user.

Similarly, the "state space" method will only keep the mannerisms of the user if the data used to decide on the position of the elbow belongs to the actual user. However, this method reflects the mannerisms of the user in a much better way than the previous one, since the degree of accuracy is much better.

6.4 Summary

In this chapter, we discuss the results of the implementation of the system.

The motion capture equipment proves to be adequate for our purpose, although the range limitation as well as its tendency to suffer from electromagnetic and ferromagnetic interference can cause some inconvenience and could be of certain concern.

We also discuss the advantages and disadvantages of the three inverse kinematics methods proposed.

The results indicate that the "eliminating a DOF" method is not robust enough for either the use as an avatar in a virtual environment or for articulated figure animation. The calibration required is also too complex for a normal user and it needs to be performed every time that the user wants to use the system.

The "hand elevation" system provides a robust inverse kinematics system which gives relatively precise elbow positions while requiring minimal calibration and processor usage. The system only requires to be calibrated once per user. Although sometimes the elbow tends to be in a highly incorrect position, for most of the arm poses the elbow is accurately represented. As such, this system can be used for representation of avatars in virtual environments, and can be used as a simple articulated figure animation system.

The "state space" system is not as robust as the "hand elevation" system, but tends to give closer approximations of the elbow position. Its processing cost is higher than that of the other two systems, but only requires a once off calibration per user. Just like the "hand elevation" system, this system can be nicely used for representation of avatars in virtual environments. However, due to the "shaking" of the elbow, the system is not adequate for articulated figure animation of high quality.

Chapter 7. Conclusion

7.1 Summary

In this thesis, we discuss the issues related to the design and implementation of an animation of an articulated figure. The system is designed to run in real time, using motion capture data from minimal motion capture equipment. The system is designed to drive an avatar in a virtual environment on a low-end computer without specialised hardware except for the motion capture equipment. The system could also be used for any application requiring the animation of an articulated figure such as articulated figure animation, the animation of characters in computer games or for use in a simple virtual video conferencing system, as long as the user is prepared to accept the cosmetic shortcomings of the system.

To achieve such purpose, we investigate the various types of articulated figure animation techniques, as well as the available types of motion capture equipment. From this we conclude that a key frame system is the most beneficial system to use. Similarly, we conclude that the use of electromagnetic motion capture equipment provides us with the necessary motion capture data while satisfying other conditions such as a high update rate, a low latency, a lack of required post processing and a relatively low cost.

We also describe some background information into the field of robotics and that of articulated figure animation. We examine the benefits and hindrances of the use of kinematics and dynamics as systems to specify the animation. We decide to use inverse kinematics to specify the movement and pose of the articulated figure animation. We implement a first and simple prototype to investigate and illustrate issues dealing with the redundancy in animating an articulated figure. Although our work can be applied to other articulated figure topology, we use an articulated figure representative of the human

upper body. This first implementation allows us to test the performance of the motion capture equipment under the VR system used. It also allows us a first exploration of the use of constraints to reduce the redundancy of the system.

As such, we propose the use of a new VR system to improve the performance of the system and of the motion capture equipment used. We also decide on a system to specify the articulated figure.

We develop a system used to reduce the redundancy of an articulated chain composed of four links in which the length of the links is known as well as the number of DOF for each joint. The position of the first and last link must also be known. This system can be used to reduce the redundancy of any articulated chain and is not specific to the human upper body.

We then use this system to reduce the redundancy of the various articulated chains that compose the articulate human upper body.

We subsequently investigate the use of various types of constraints to further reduce the redundancy of the system to the point of the system being fully specified. These constraints are used to solve the undetermined orientation of the neck (section 3.2.3.1), the reduction of the elbow from a circle in 3D space to two points (the "eliminating a DOF" method), the selection of the correct elbow position (section 5.3.1.1), the determination of the elbow position using the elevation of the hand (the "hand elevation" method), and the determination of the elbow position using the position and orientation of the hand with respect to the body (the "state space" method).

7.2 Accomplishment

The system manages to animate a human-like articulated figure in real-time. The VR system used is fast enough to provide the motion capture data with a good update rate and low latency. We use electromagnetic trackers due to their good update rate and low latency, relative low price and availability of six DOF data. Using only four motion capture trackers, we manage to get enough data to fully specify the articulated human upper body with the use of inverse kinematics and constraints. The inverse kinematics system and constraints are simple enough to only require a low amount of processing power.

Using the constraints referred to at the end of the previous section, we manage to animate an articulated human upper body to a reasonable degree. We offer a method to reduce the possible position of the elbow in an articulated human arm to a circle in 3D space. We also propose three different inverse kinematics methods to determine the position of the elbow in the articulated human arm as well as a method to determine the orientation of the neck.

The first method to determine the position of the elbow ("eliminating a DOF") should give a perfect result theoretically. In practice however, it requires extremely accurate calibration of the system and imposes quite limiting constraints on how the user is allowed to hold his hand. As such it is not adequate even for use as an avatar in a virtual environment.

The second method ("hand elevation") determines the position of the elbow from the elevation of the hand. This method proves adequate for most of the user's poses but fails for some of them. This means that the system can be used as an avatar in a virtual environment where the position of the elbow is not critical, but is unsuitable for articulated figure animation unless the user is prepared to put up with the limitations of this method.

The third method ("state space") determines the position of the elbow from an average of previous elbow positions according to the position and orientation of the hand. This method proves adequate when an elbow position is available but fails otherwise. Even when an elbow position is available, the system suffers from slight shaking of the elbow. Still, the system can be used to represent an avatar in a virtual environment. This system is also more suitable for low quality articulated figure animation.

Although two of the inverse kinematics methods proposed can be used for low quality articulated figure animation, they are not suitable for high quality articulated figure animation. This is due to the restrictions in the number of motion capture trackers used. Thus the redundancy of the system can not be decreased considerably without loss of quality in the determination of the position and orientation of the links.

7.3 Future work

As a result of our work, an avatar with relatively accurate poses can be animated in real time from motion capture data. However, there are plenty of areas that can be improved:

Investigation of other alternatives to the methods presented to reduce the redundancy of articulated kinematic chains can provide with better solutions to solving the inverse kinematics problem. Consequently, the accuracy of the poses of the articulated figure will increase, thus increasing its potential for high quality animation.

Using a scripted system to drive the lower part of the body could be a useful addition to the system since this would mean the availability of a full body as opposed to an upper body only.

One could also investigate the performance increase by distributing various parts of the system across the network. As an example, consider if it is better to calculated the inverse kinematics system locally and then send the information of the pose of the avatar through a network like the Internet, or is it better to send the raw position of the trackers through the network and them computing the inverse kinematics calculations at the destination.

The system could also be used in a VR conferencing system, although it would probably require some kind of facial animation or facial expression to be useful. The system could then be compared to common video conferencing systems in term of usefulness as a conferencing system, in terms of bandwidth requirements (especially for multiple users) or in terms of the possibilities of having a conference in a (modifiable) virtual world with (modifiable) avatars.

As stated earlier, the system can be used to animate avatars in virtual worlds or computer games. Although not entirely related to the research in kinematic chains, the effect of the usage in virtual worlds of an avatar that has an increased interaction capability are worth investigating.

Bibliography

[AMD 3DNow] AMD 3DNow http://www.amd.com/advances/columns_features/issue31/pg02_side_bar2.html

[Armstrong et. al., 1985] Armstrong, W., Green, M., and Lake, R.. ***The dynamics of articulated rigid bodies for purpose of animation***, The Visual Computer, 1(4), p 231-240, 1985.

[Badler, 1987] Norman Badler, 1987, ***Articulated Figure Animation***, IEEE Computer Graphics, Guest Editor's Introduction, June 1987.

[Badler, 1997] Norman Badler, 1997, ***Virtual Humans for Animation, Ergonomics, and Simulation***, Center for Human Modeling and Simulation, Department of Computer and Information Science, University of Pennsylvania, http://www.cis.upenn.edu/~badler/vhpaper/vhlong/vhlong.html

[Badler and Ko, 1996] Hyeongseok Ko and Norman I. Badler, ***Animating Human Locomotion with Inverse Dynamics***, IEEE Computer Graphics, March 1996.

[Badler and Phillips, 1991] Gary B. Phillips and Norman I. Badler, ***Interactive Behaviour for Bipedal Articulated Figures***, SIGGRAPH'91, Computer Graphics, Volume 25, Number 4, July 1991

[Badler *et. al.*, 1995] Norman I. Badler, Michael J. Hollick, John P. Granieri, ***Real-Time Control of a Virtual Human Using Minimal Sensors*** Presence 2(1), pp. 82-86. 1995

[Bangay, 1996] Shaun Bangay, ***Modelling Parallel and Distributed Virtual Reality Systems for Performance Analysis and Comparison***,

Ph.D. Thesis, Department of Computer Science, Rhodes University, November 1996.

[Bangay et. al., 1996] Shaun Bangay, James Gain, Greg Watkins, Kevan Watkins, ***RhoVeR: Building the Second Generation of Parallel/Distributed Virtual Reality Systems***, First Eurographics Workshop on Parallel Graphics & Visualisation, Bristol(UK), 26-27 September 1996.

[Burton, 1979] Burton, ***Kinematics and Dynamics of Planar Machinery***, Prentice Hall, 1979.

[Calvert and Chapman, 1982] T.W. Calvert and J. Chapman, ***Aspects of the Kinematic Simulation of Human Movement***, IEEE Computer Graphics, November 1982.

[Casanueva, 1996] Luis Casanueva, ***Creating Surface Models in Virtual Reality***, Honours Thesis, Computer Science Department, Rhodes University, November 1996.

[Casanueva *et al*, 1997] Luis Casanueva, Shaun Bangay and George. Wells, ***A Distributed Virtual Reality Interface for use in Articulated Figure Animation***, Teletrafic97, 1997.

[Casanueva and Bangay, 1998] Luis Casanueva, Shaun Bangay ***Minimal Motion Capture for Articulated Figure Animation***, Santac98, 1998.

[Fetter, 1982] William A. Fetter, ***A Progression of Human Figures Simulated by Computer Graphics***, IEEE Computer Graphics, November 1982.

[Frey et. al., 1996] W. Frey, M. Zyda, R. McGhee, B. Cockayne, ***Off-the-Shelf, Real-Time, Human Body Motion Capture for Synthetic***

Environments, Naval Postgraduate School, Technical Report NPSCS-96-003, June, 1996.

[Frey et. al. supp, 1996] W. Frey, M. Zyda, R. McGhee, B. Cockayne, ***Off-the-Shelf, Real-Time, Human Body Motion Capture for Synthetic Environments***, additional "Supplement", Naval Postgraduate School, Technical Report NPSCS-96-003, June, 1996.

[Gain, 1996] James Gain, Virtual Sculpting: ***An Investigation of Directly Manipulated Free-Form Deformation in a Virtual Environment***, M.Sc. Thesis, Department of Computer Science, February 1996.

[Gomez, 1985] J. Gomez, ***Twixt: A Three Dimensional Animation System***. Computers and Graphics, 9 (3) : 291 – 298, March 1985.

[Hanrahan and Sturman, 1985] P. Hanrahan and D. Sturman, ***Interactive Animation of Parametric Models***. The Visual Computer, 1 : 260 – 266, 1985.

[Herbison-Evans, 1982] Don Herbison-Evans, ***Real-Time Animation of Human Figure Drawings with Hidden Lines Omitted***, IEEE Computer Graphics, November 1982.

[Hunt, 1978] Hunt, K. H., ***Kinematic Geometry of Mechanisms***, Clarendon Press, Oxford, New York, 1978.

[Id Software, 1996] http://www.idsoftware.com

[Intel Katmai] Intel Katmai, http://developer.intel.com/drg/news/katmai.htm

[Korein, 1985] Korein, J., 1985, A Geometric Investigation of Reach. MIT Press, Cambridge, MA.

[Magnenat & Thalmann, 1985] Magnenat-Thalmann N. and Thalmann D., ***Principles of Computer Animation***, Tokyo: Springer, 1985.

[Mainframe Entertainment] Mainframe Entertainment Inc. http://www.mainframe.bc.ca

[NRC,1995] National Research Council (NRC), Committee on Virtual Reality Research and Development, Virtual Reality: Scientific and Technological Challenges, 1995, National Academy Press, Washington, DC

[OVRT, 1998] OVRT Resources for the Humanoid Animation Working Group, http://www.nist.gov/itl/div894/ovrt/projects/vrml/h-anim/jointInfo.html

[Parent, 1998] Richard Parent, ***Algorithms and Models to fully specify motion***, http://www.cis.ohio-state.edu/~parent/book/Modl.html, Ohio State University, 1998.

[Parke, 1982] Frederick I. Parke, ***Parameterized Models for Facial Animation***, IEEE Computer Graphics, November 1982

[Perlin and Goldberg, 1996] K. Perlin and A. Goldberg, ***"Improv: A System for Scripting Interactive Actors in Virtual Worlds," in Computer Graphics*** (SIGGRAPH '96 PROCEEDINGS), pp. 205--216, ACM, 1996.

[Pixar Studios] Pixar Animation Studios, Richmond, CA 94804, http://www.pixar.com

[Polhemus] Polhemus inc., http://www.polhemus.com.

[Shoemake, 1985] K. Shoemake, ***Animating Rotation with Quaternion Curves***, Computer Graphics, 19 (3) : 245 – 354, July 1985.

[Sims and Zeltzer, 1988] Sims, K. and Zeltzer, D., ***A Figure Editor and Gait Controller for Task Level Animation***, SIGGRAPH Course Notes, 4, p 164-181,1988.

[Stern, 1083] G. Stern, Bbop – ***A Program for Three Dimensional Animation***, Nicograph Proceedings, pages 403 – 404, 1983.

[Sturman, 1987] D. Sturman, ***Interactive Keyframe Animation of 2D articulated models***. SIGGRAPH Course Notes: Computer Animation: 3D Motion Specification and Control, pages 17 – 26, 1987.

[Sudhanshu et. al., 1988] Sudhanshu K. Semwal, Ron Hightower, and Sharon Stansfield ***Mapping Algorithms for Real-Time Control of an Avatar Using Eight Sensors***, Presence, Vol. 7, No. 1, p 1-21, February 1988.

[Suh and Radcliffe, 1978] Suh, C. H. and Radcliffe, C. W., ***Kinematics and Mechanisms Design***, John Wiley and Sons, New York, 1978.

[Unreal, 1998] Unreal, http://www.gtinteractive.com/unreal/

[Vilhjálmsson, 1996] Hannes Högni Vilhjálmsson, 1996, ***Avatar interaction***, http://lcs.www.media.mit.edu/people/hannes/project/index.html

[Watt, 1989] Alan Watt - ***Fundamentals of Three Dimensional Computer Graphics***, Addison-Wesley Publishing Company, 1989.

[Watt and Watt, 1992] Alan Watt, Mark Watt - ***Advanced Animation and Rendering Techniques. Theory and Practice***, Addison-Wesley Publishing Company, 1992.

[Welman, 1993] Chris Welman, ***Inverse Kinematics and Geometric Constraints for Articulated Figure Manipulation***, M.Sc. Thesis, School of Computing Science, Simon Fraser University, 1993.

[Willmert, 1982] K. D. Willmert, ***Visualizing Human Body Motion Simulations***, IEEE Computer Graphics, November 1982.

Appendix A

Pictures from the implementation of the first prototype.

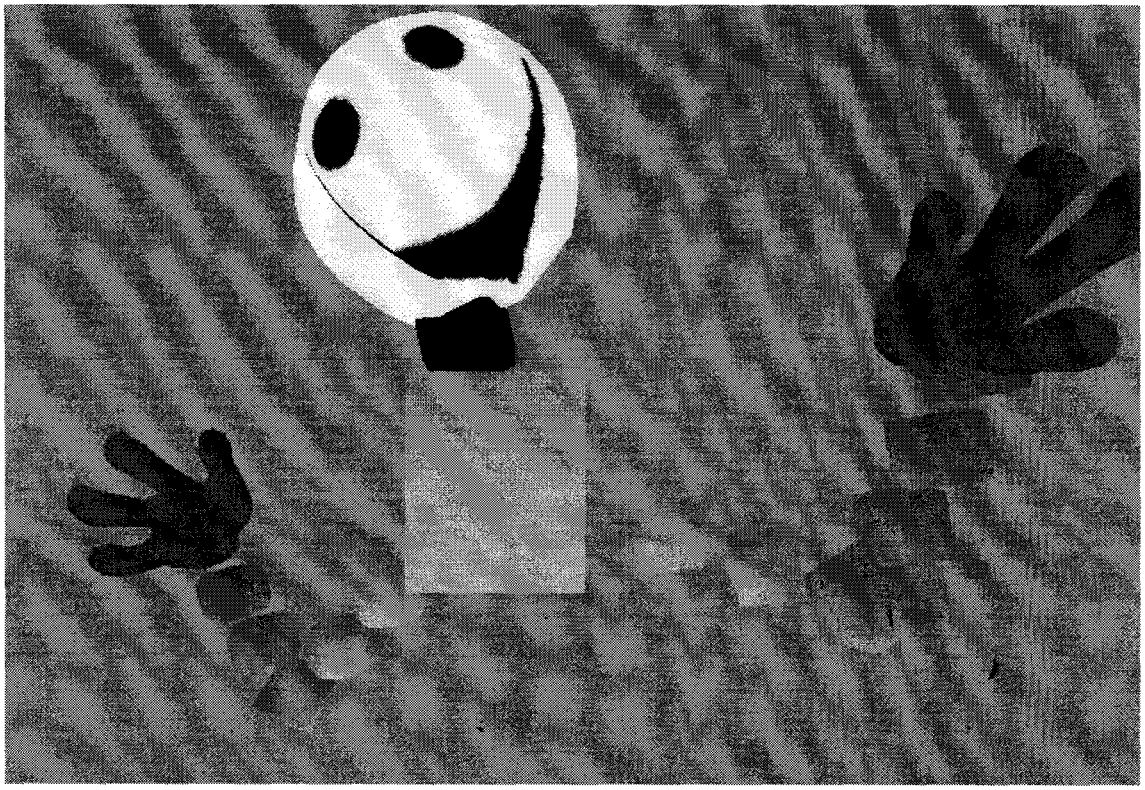


Figure A.1.- A picture of the avatar from the first prototype.

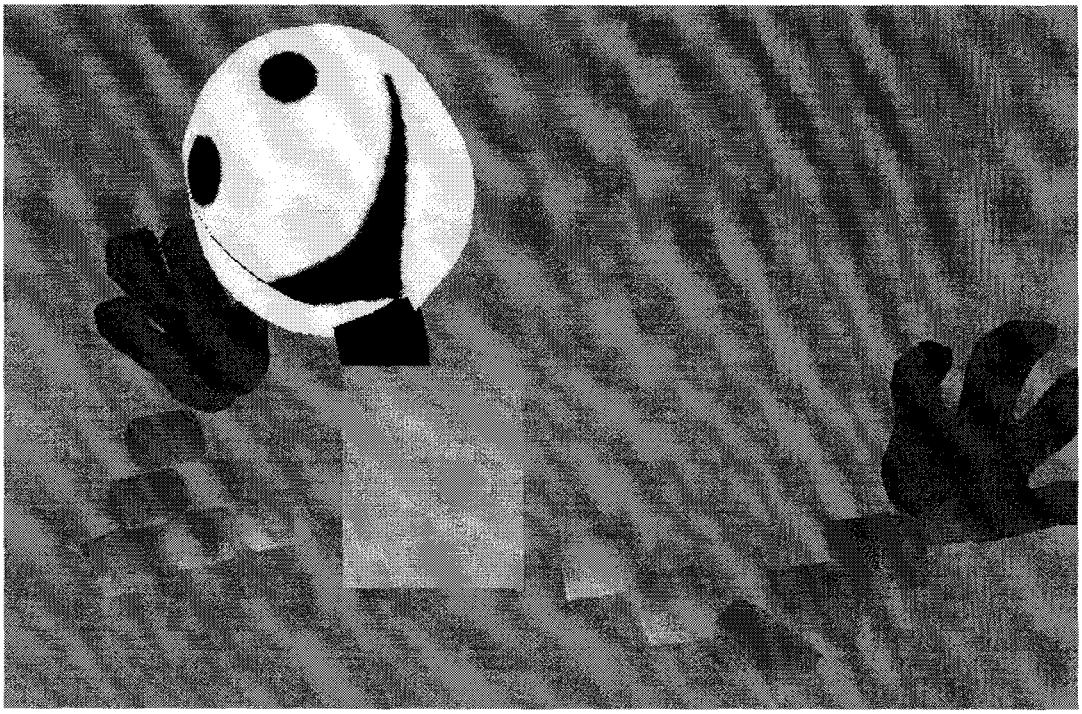


Figure A.2.- Another picture of the avatar.

Figure A.3.- The author's pose corresponding to the picture in figure A.3.

Appendix B

Graphs for the hand elevation relationship method.

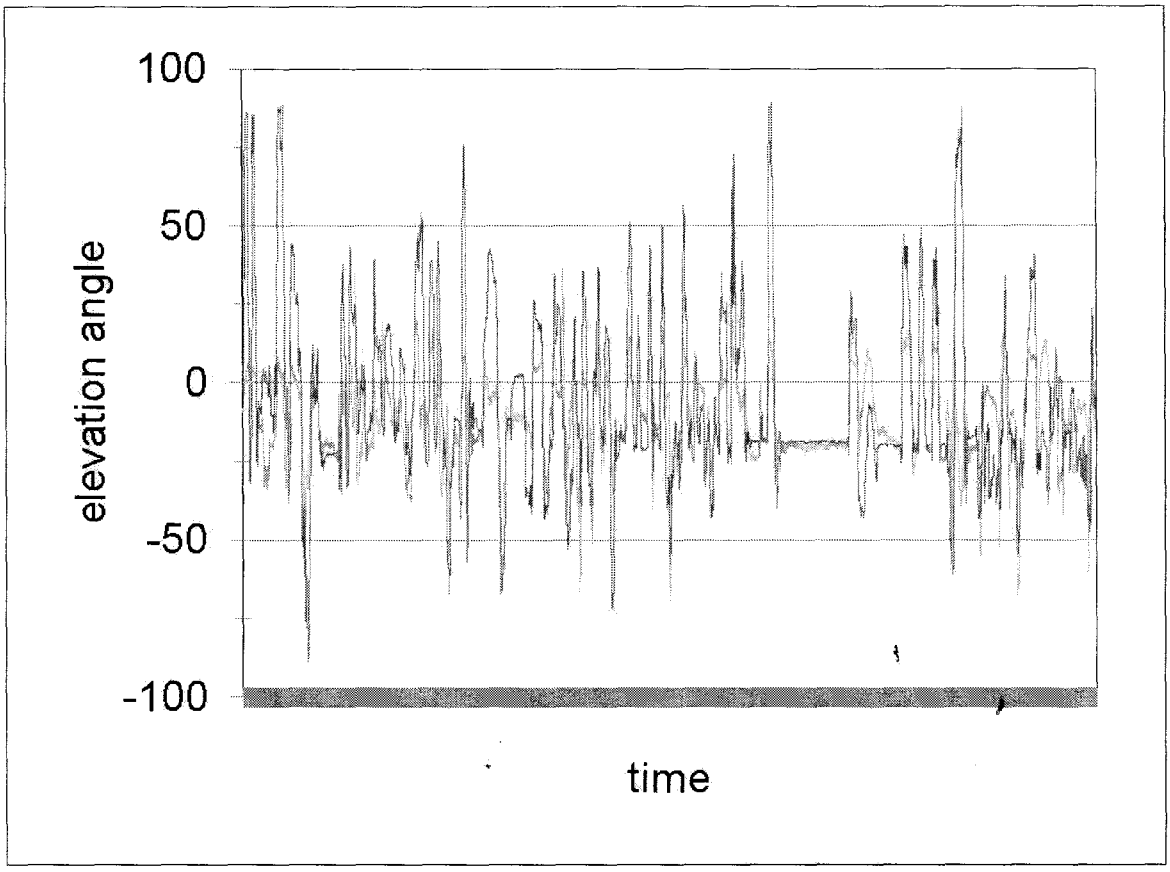


Chart B.1.- Elevation of the hand and (elbow – 45 degrees) vs. time. This is the same graph as the one in section 6.3.2.1. except that an offset of 45 degrees has been subtracted from the elbow elevation.

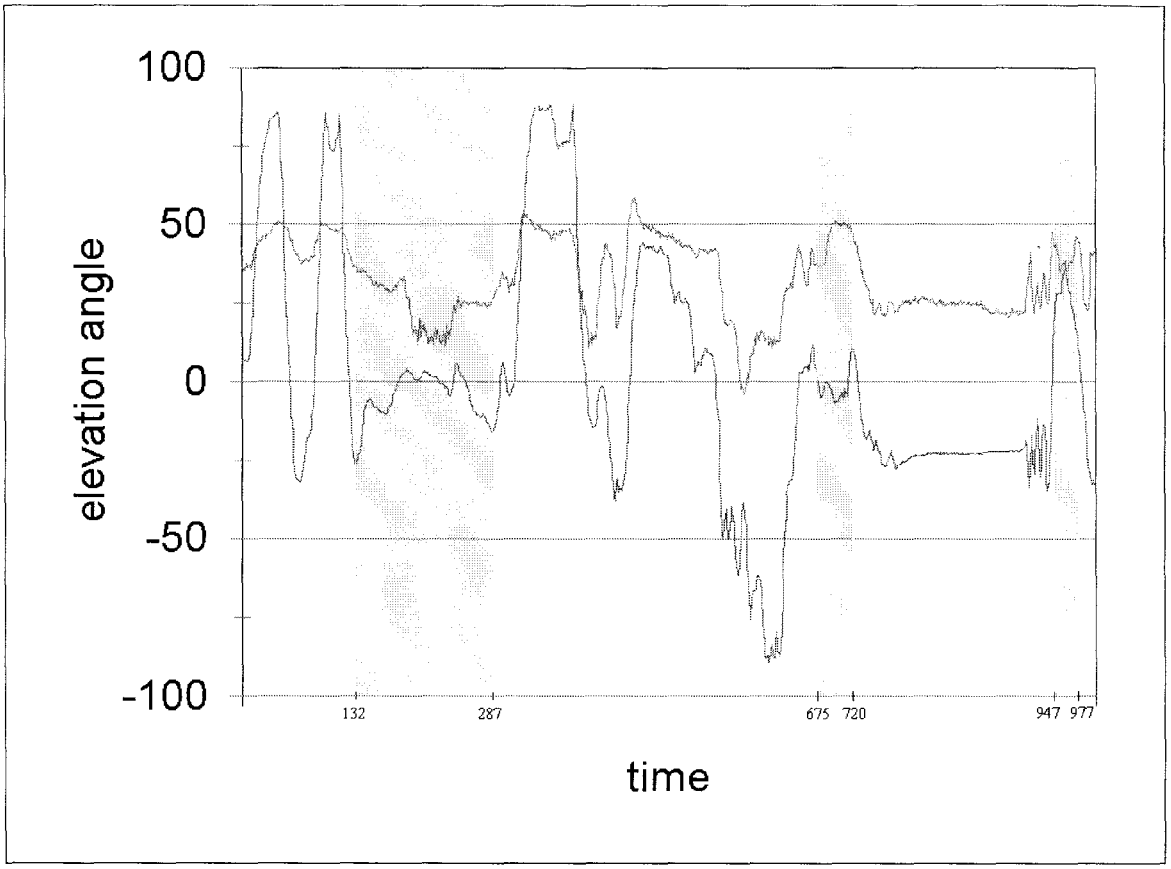


Chart B.2.- Elevation of the hand and elbow vs. time for time=1 to time=1000. The shaded areas represent the areas noted in section 6.3.2.2.

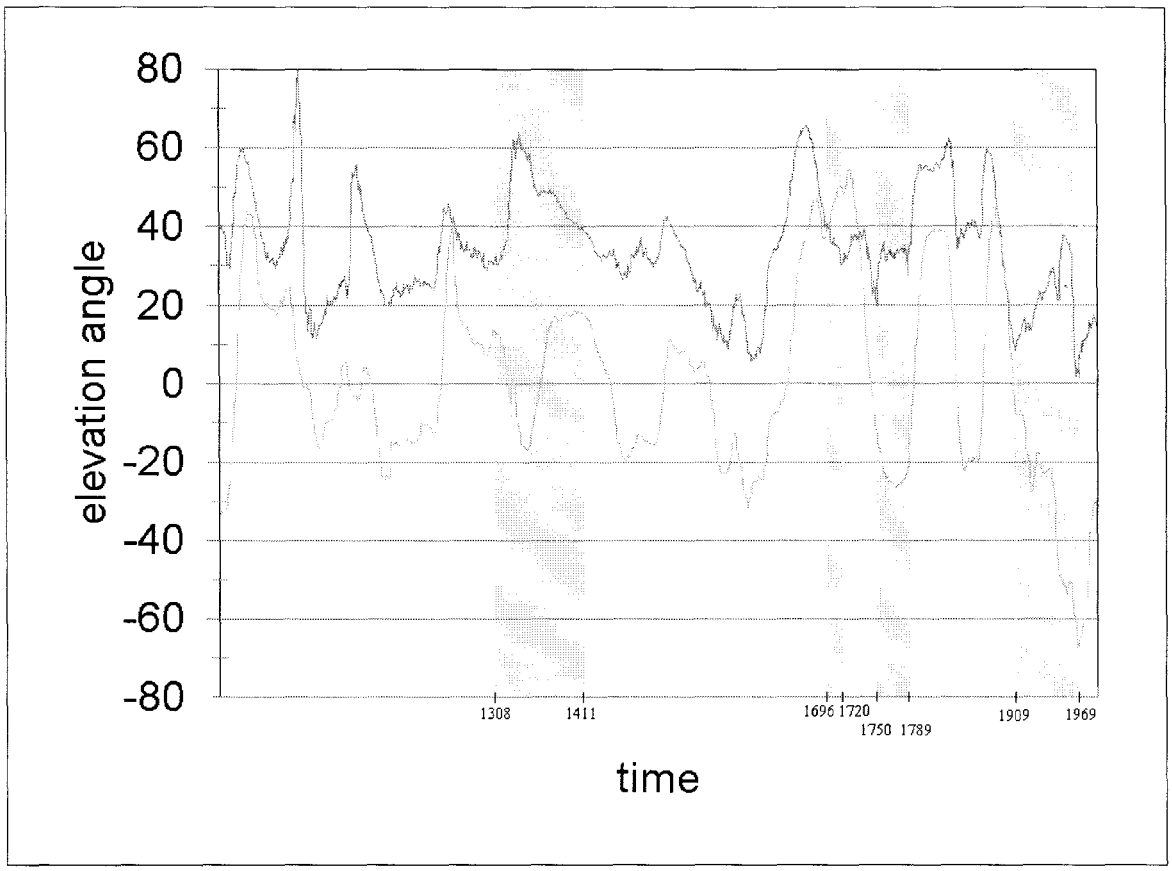


Chart B.3.- Elevation of the hand and elbow vs. time for time=1001 to time=2000.

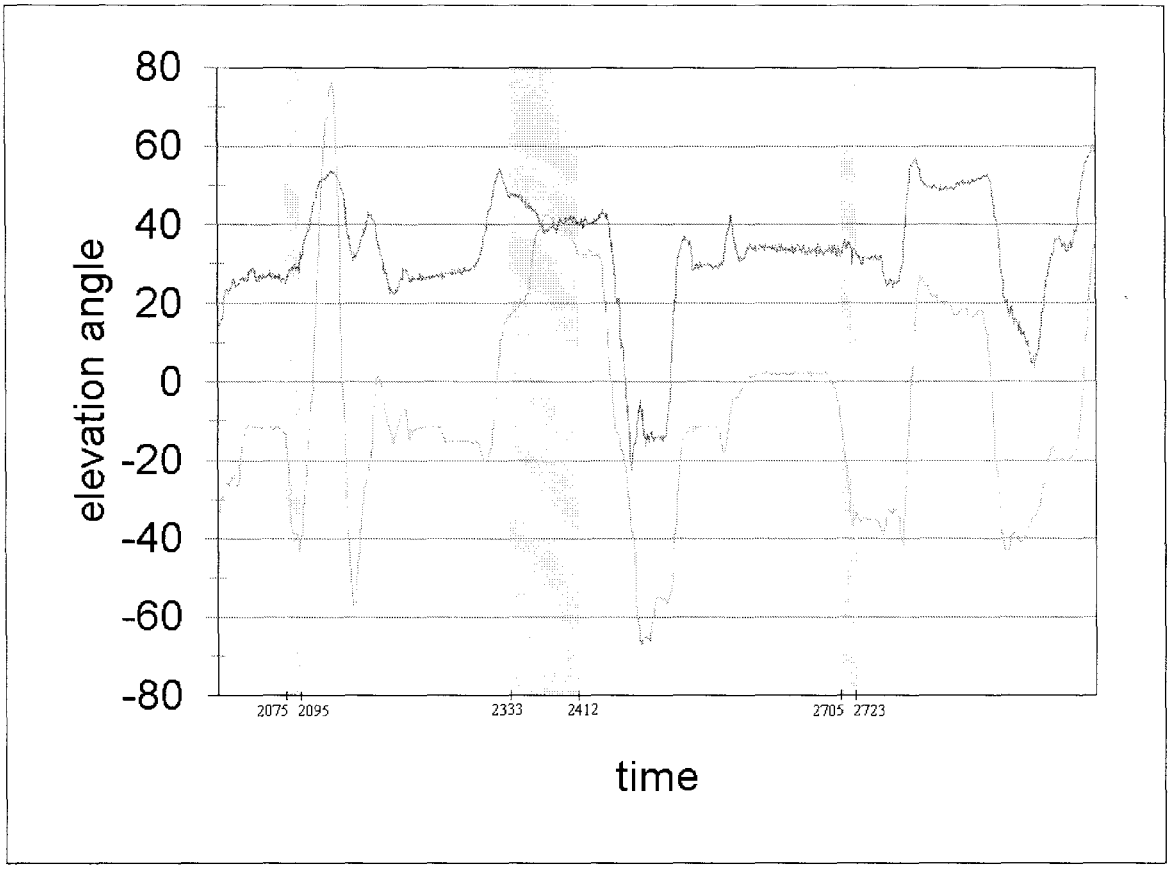


Chart B.4.- Elevation of the hand and elbow vs. time for time=2001 to time=3000.

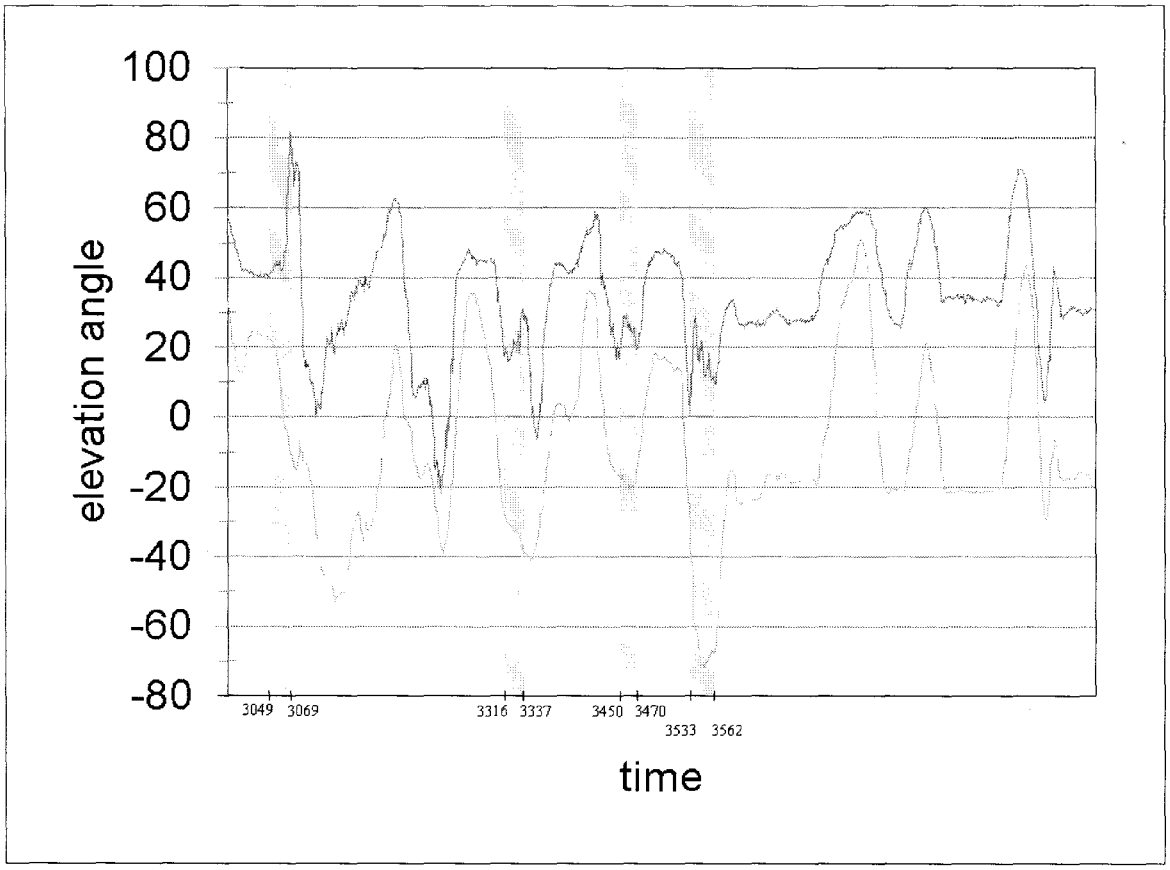


Chart B.5.- Elevation of the hand and elbow vs. time for time=3001 to time=4000.

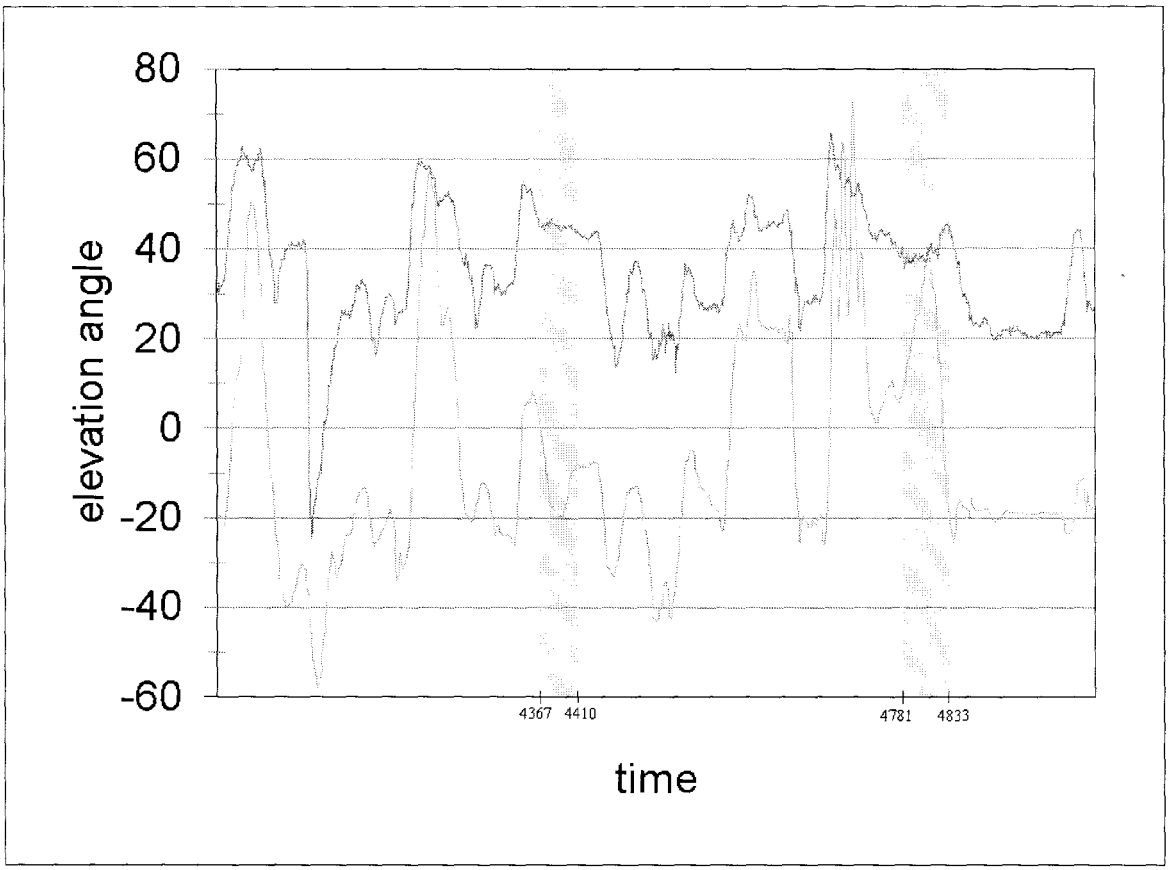


Chart B.6.- Elevation of the hand and elbow vs. time for time=4001 to time=5000.

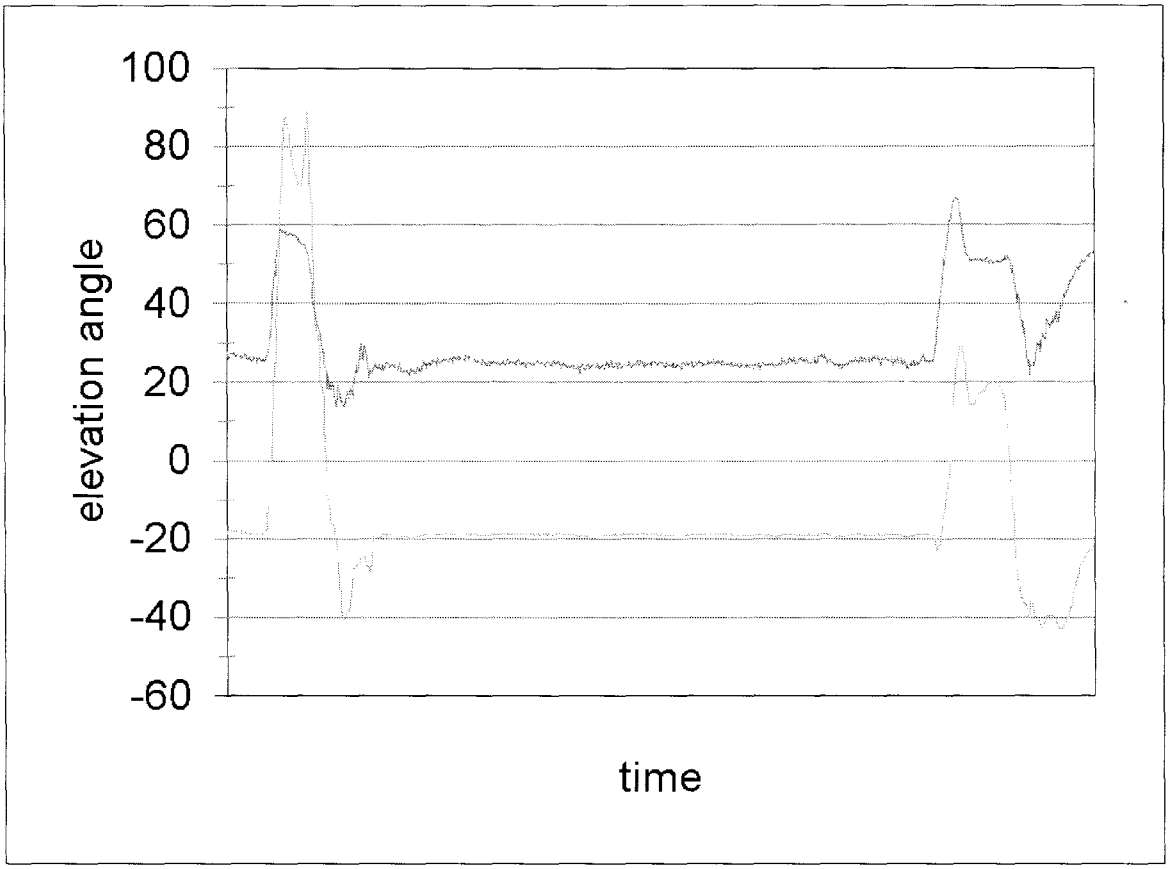


Chart B.7.- Elevation of the hand and elbow vs. time for time=5001 to time=6000.

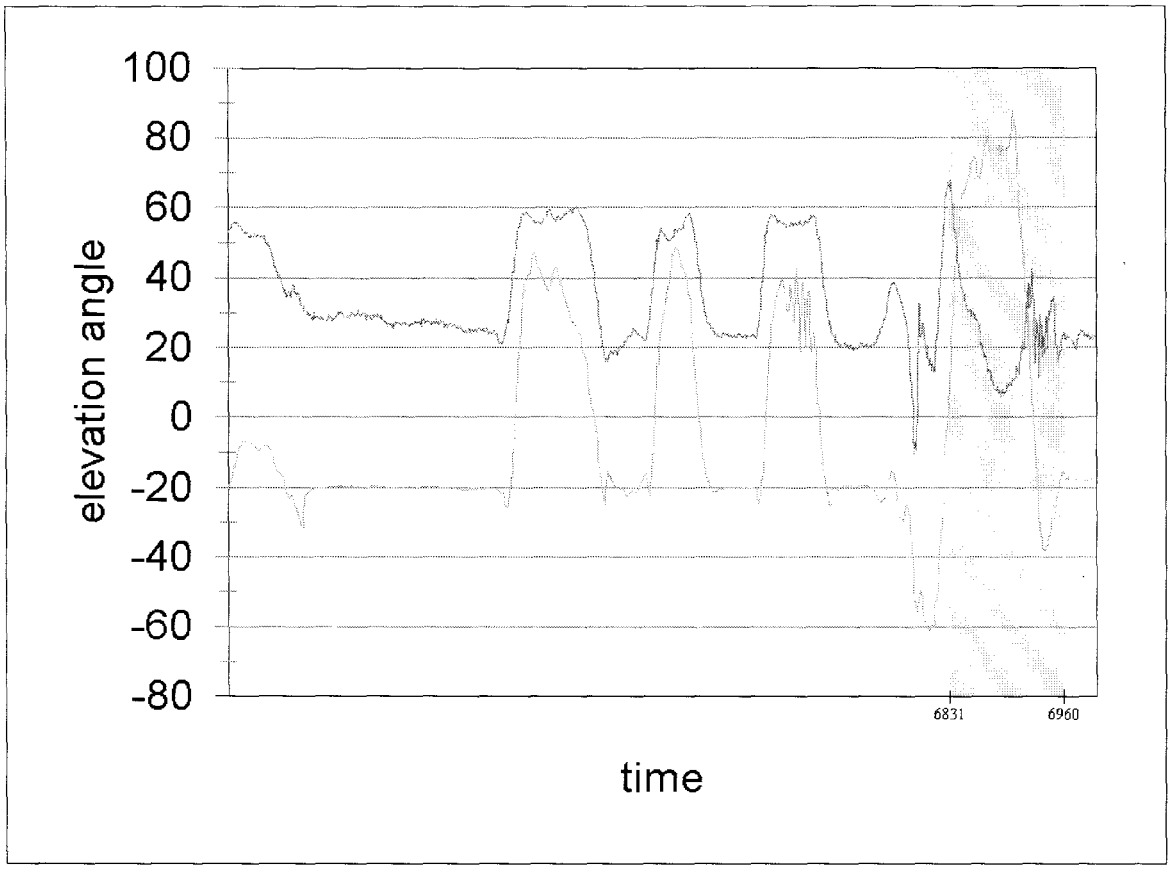


Chart B.8.- Elevation of the hand and elbow vs. time for time=6001 to time=7000.

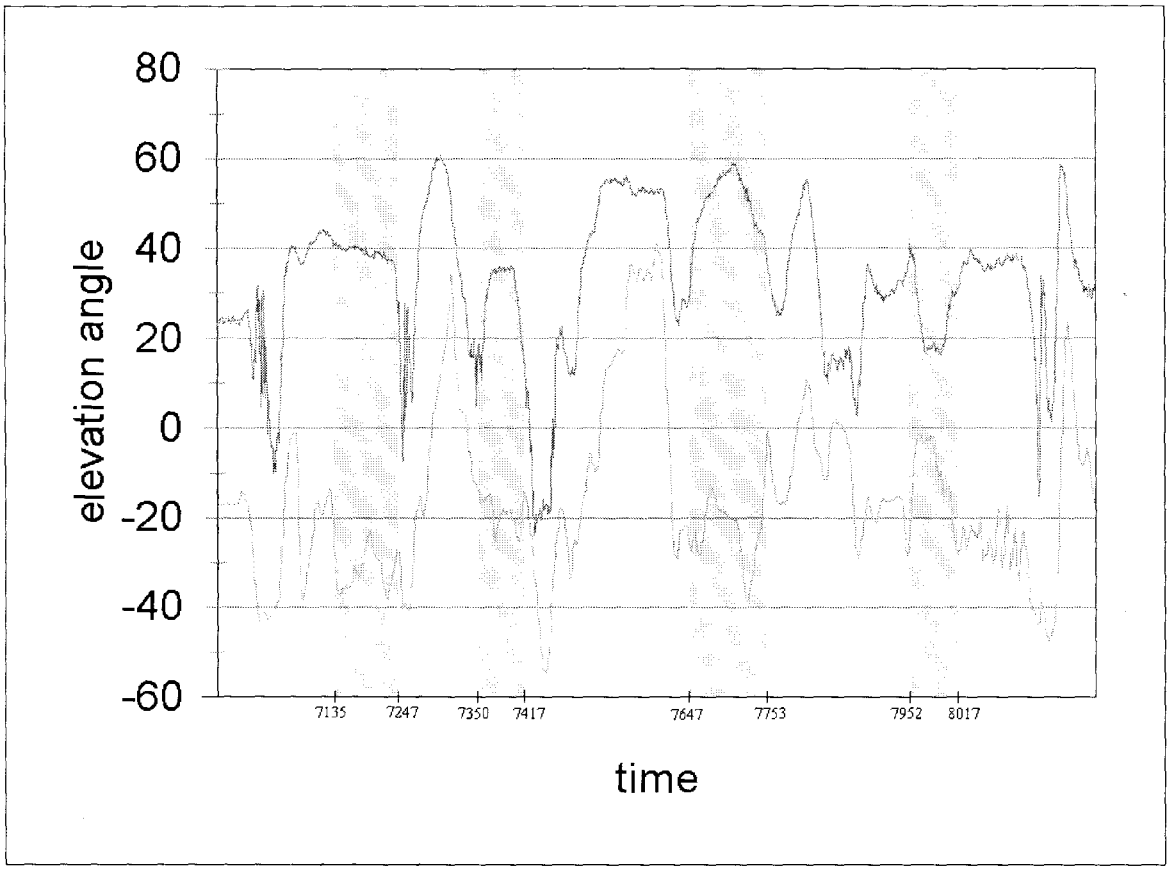


Chart B.9.- Elevation of the hand and elbow vs. time for time=7001 to time=8200.

Appendix C

This appendix presents graphs of the results from the "state space" inverse kinematics system. The graphs represent the position of the calculated elbow (in red) and the position of the real elbow (in blue). As one can see, the two lines match almost perfectly. The yellow line indicates the amount of data used to calculate the average of the elbow position.

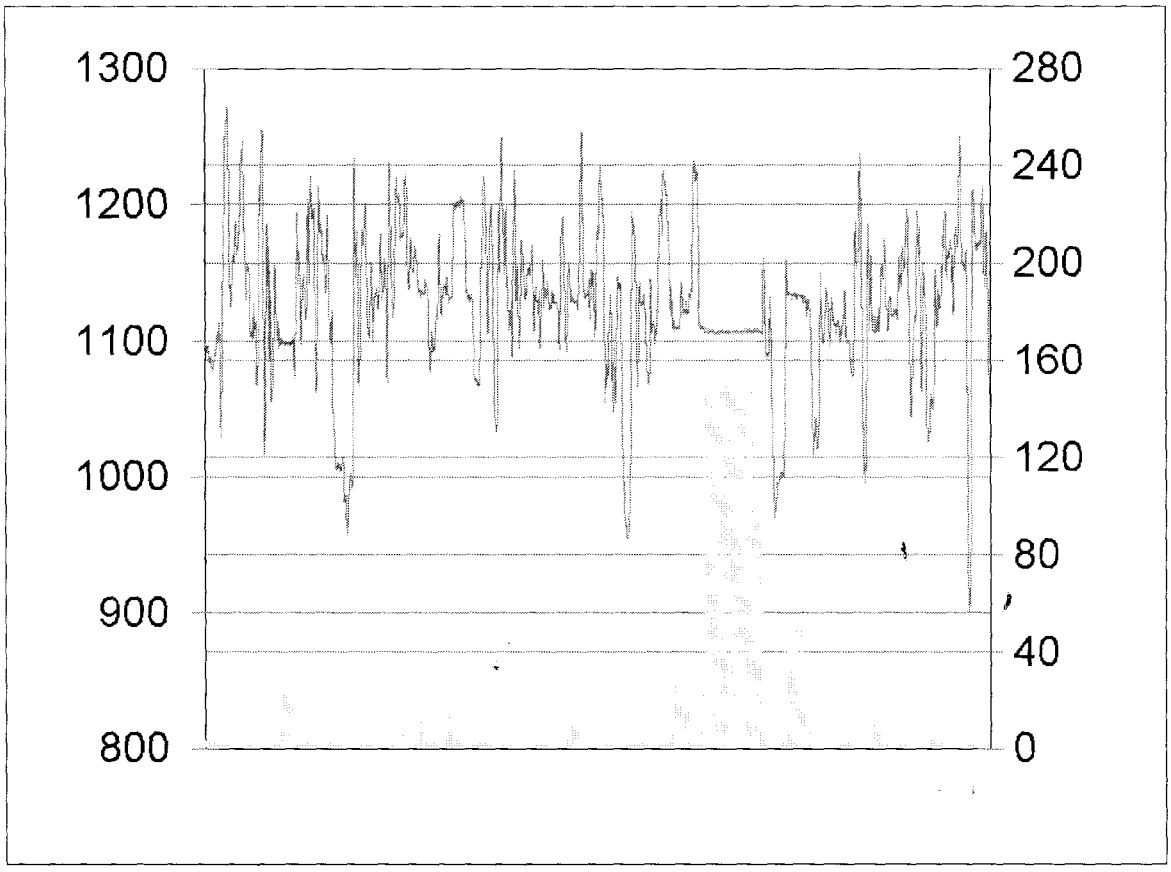


Figure C.1.- Position of the calculated (in red) and real elbow position (in blue) along the x-axis.

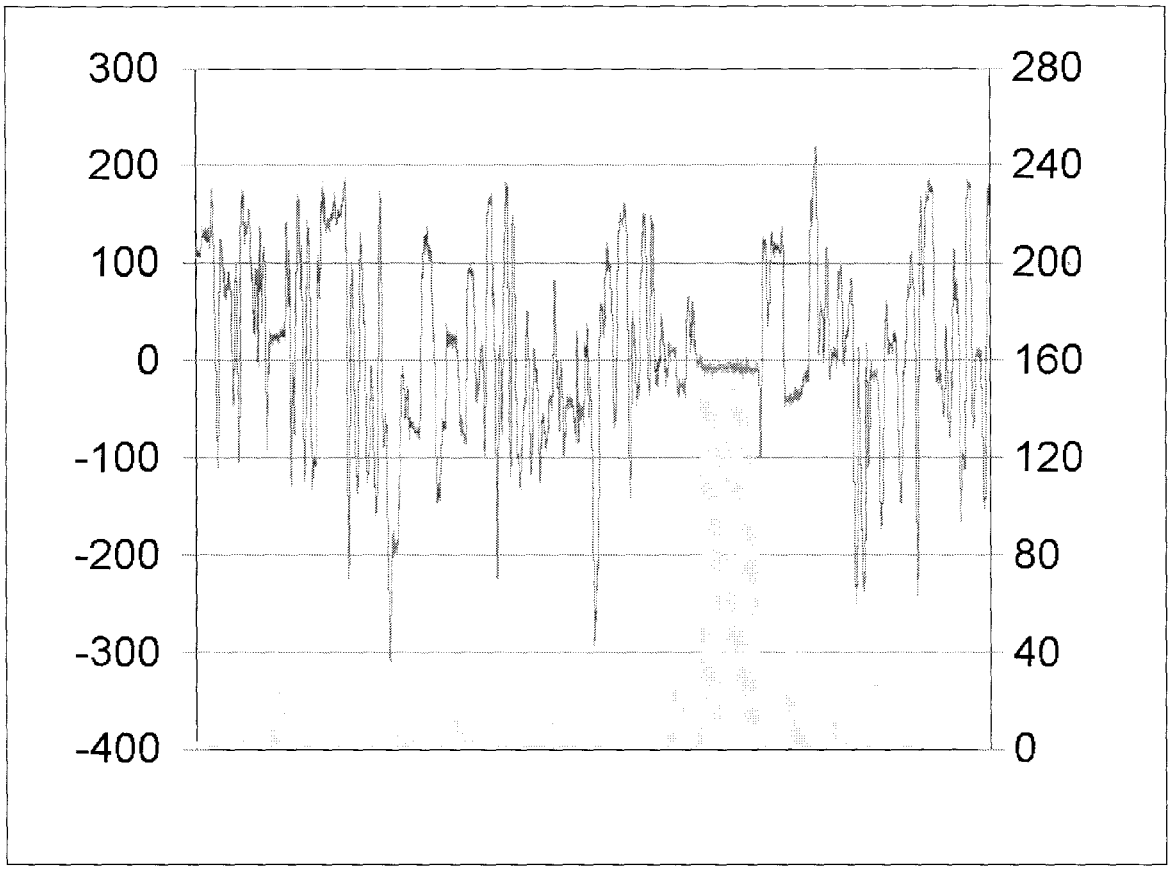


Figure C.2.- Position of the calculated (in red) and real elbow position (in blue) along the y-axis.

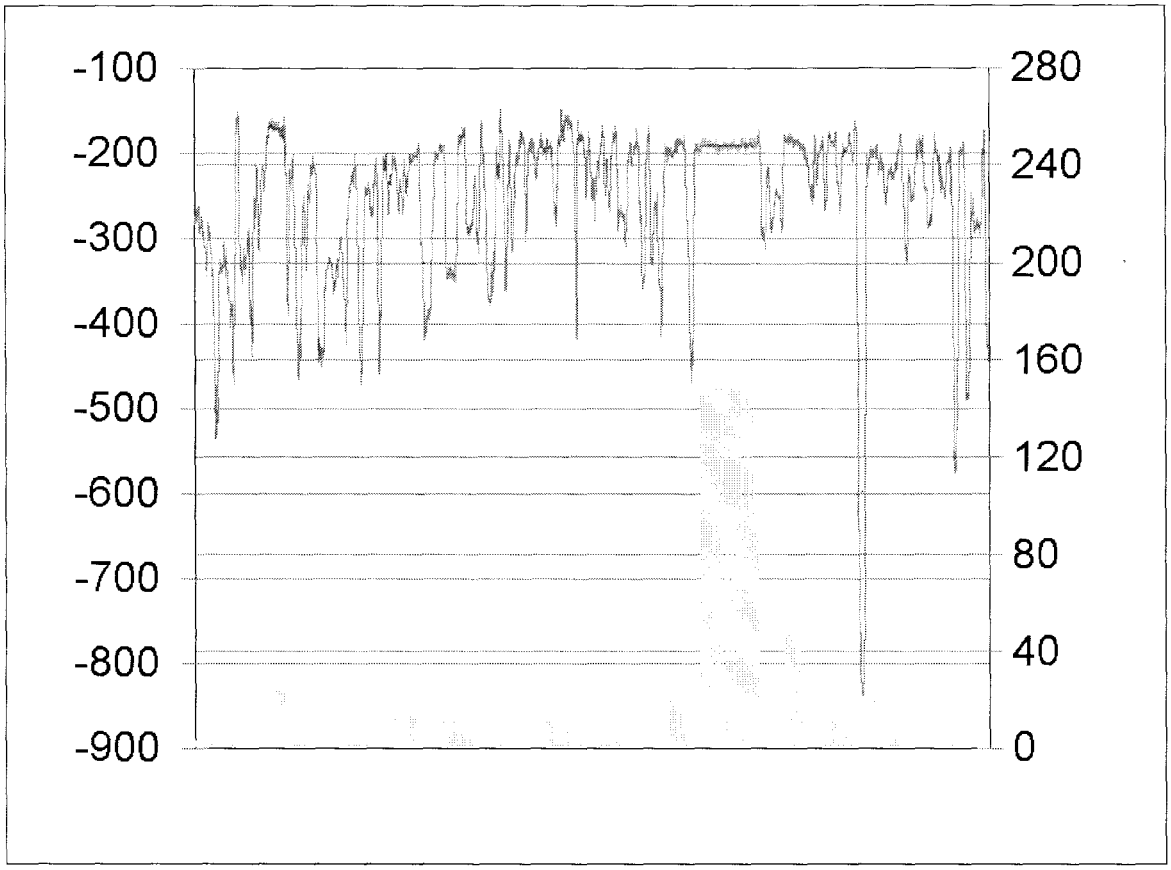


Figure C.3.- Position of the calculated (in red) and real elbow position (in blue) along the z-axis.

Appendix D

Pictures from the implementation of the second prototype. The system was used without any calibration at the beginning of the session. Instead, calibration data from a different user was used.

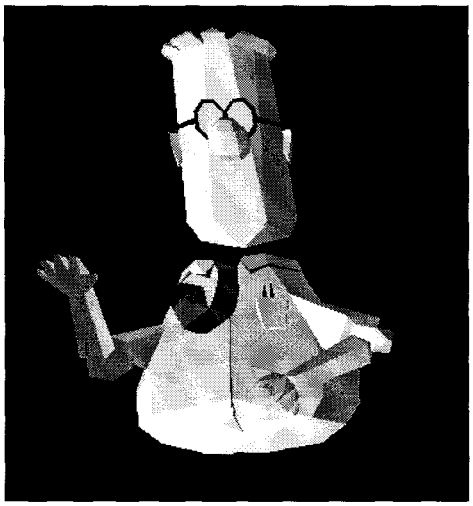


Figure D.1.- The avatar from the second prototype greeting the user.

Figure D.2.- The "supervisor" avatar after the author delivers his weekly progress report.

Figure D.3.- So much work that the user just lost his head.