THE EFFECT OF THE TURNING STRATEGY ON MANEUVERING ABILITY USING THE TREADPORT

by

Abhijeet Vijayakar

A thesis submitted to the faculty of The University of Utah in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computer Science

The University of Utah

December 2000
ABSTRACT

One-dimensional (1D) treadmills are locomotion interfaces in which the user walks on a linear treadmill belt while travelling through the virtual world. They are good at simulating straight-line locomotion through the virtual world, but since the treadmill belt can only move forward or back special strategies have to be devised to allow the user to change his direction of motion. This thesis addresses the problem of turning on the second-generation Treadport, a 1D treadmill-style interface designed and built by Sarcos Research Corp. Two turning strategies are presented for the Treadport. The first is a restricted version of the turning strategy used on the first generation of the device and uses rate control for turning. The second, new strategy utilizes the angle of twist of the user’s torso and head to determine when the user wants to turn, and how much of a turn is desired. Proportional control is used for turning, with the novel addition of a “trigger” condition to initiate turning. It was expected that the new strategy would allow users to turn better in the virtual world than the old one.

An experiment was carried out to compare the relative performance of the two strategies as applied to a maneuvering task. Users were asked to walk from one end to another of a virtual path while avoiding as many of the obstacles along the path as possible. The results clearly showed that the new strategy employing head twist allowed users to maneuver more precisely through the virtual world than the old strategy. In a more general sense, this also indicates that a suitably designed turning strategy using proportional control can outperform one using rate control on 1D treadmills.
CONTENTS

ABSTRACT ........................................................................................................ ii

LIST OF FIGURES ......................................................................................... v

LIST OF TABLES ............................................................................................ vi

CHAPTERS

1. INTRODUCTION ....................................................................................... 1

1.1 Locomotion Interfaces ........................................................................ 1

1.2 Treadmill-Style Locomotion Interfaces .............................................. 2

1.2.1 The Treadport ................................................................................. 3

1.3 Locomotion Rendering Issues ............................................................ 4

1.4 The Problem of Turning with 1D Treadmills ...................................... 5

1.5 Discussion of Control Issues in Turning on 1D Treadmills ............... 6

1.6 This Research ...................................................................................... 7

1.6.1 Scope ............................................................................................ 8

1.7 Road Map .......................................................................................... 9

2. BACKGROUND .......................................................................................... 10

2.1 Previous Locomotion Interfaces ......................................................... 10

2.2 Rate Control vs. Proportional Control .............................................. 13

2.3 Turning on Treadmill-Based Locomotion Interfaces ....................... 13

2.3.1 Turning on the First-Generation Treadport ................................... 15

3. STRATEGIES TO BE COMPARED ......................................................... 17

3.1 Strategy I: Turning Using Sidestep ....................................................... 17

3.2 Strategy II: Turning Using Head Twist .............................................. 18

3.2.1 Intent to Turn and Extent of Turn ................................................ 18

3.2.2 The Head Twist Strategy ............................................................... 19

3.2.3 Consequences of the Exponential Decay in the Rate of Bearing Change . .... 21

3.2.4 Implementation Details ................................................................. 21

3.2.4.1 Basic Algorithm .................................................................. 23

3.2.4.2 Termination Conditions for the Turn ..................................... 24

3.2.4.3 Angle Turned in One Clock Cycle ......................................... 25

3.2.4.4 A More Precise Definition of Torso Twist ............................. 26

3.2.5 Why Head Twist Works Better Than Torso Twist in Indicating Extent of Turn ........ 27
3.2.6 Limitations and Compromises .......................... 28
3.3 Conclusion .................................................. 29

4. AN EXPERIMENT TO COMPARE TURNING STRATEGIES 30
   4.1 Experiment Objective and Task .......................... 30
   4.2 Pre-Experiment Tuning .................................. 30
   4.3 Experiment Details ...................................... 31
      4.3.1 Independent Measures .............................. 31
      4.3.2 Dependent Measures ............................... 31
      4.3.3 Design .............................................. 31
      4.3.4 Procedure .......................................... 32
      4.3.5 Hardware ............................................ 33
      4.3.6 Virtual Environments ............................... 33
      4.3.7 Collision Detection ................................. 36
         4.3.7.1 Width of the Rectangular Parallellepiped .... 37
      4.3.8 Subjects ............................................ 42
   4.4 Results .................................................. 42
      4.4.1 Number of Collisions .............................. 42
      4.4.2 Learning Effects .................................. 44
      4.4.3 Effect of Difficulty Level ......................... 45
      4.4.4 Traversal Time and Traversed Distance ........... 45
   4.5 User Preference .......................................... 46
   4.6 Summary of Results ..................................... 47

5. CONCLUSIONS AND FUTURE WORK ............................... 48

APPENDIX: INSTRUCTIONS TO PARTICIPANTS .................. 52
REFERENCES ................................................................ 54
LIST OF FIGURES

1.1 The second-generation Treadport at the University of Utah . . . . . . 4
2.1 Control actions required for turning at various user speeds . . . . 15
3.1 Strategy I: The amount of sidestep determines the rate of turning . . . 18
3.2 Strategy II: The torso needs to be twisted enough to express intent to
   turn, and the angle of head twist must be constant for a sufficient
   period of time, in order to initiate turning. . . . . . . . . . . . . . . 20
3.3 As the exponential decay constant $k$ decreases, the average rate of
   turning decreases and the time to complete the turn increases . . 22
3.4 Torso twist is measured relative to a line parallel with the side edges of
   the Treadport and perpendicular to the front screen . . . . . . . 26
4.1 Overhead view of a sample path for the experiment . . . . . . . . . 34
4.2 User’s perspective of a sample path for the experiment . . . . . . . 35
4.3 The side walls along the path are divided into several different segments
   for collision detection . . . . . . . . . . . . . . . . . . . . . . . . . . 38
4.4 The view pyramid for the front screen is formed by connecting the user’s
   eye point with the four edges of the screen . . . . . . . . . . . . . 39
4.5 An object that lies to one side of the eye point will not be projected
   onto the front screen even if the user grazes it with his shoulder . . . 41
4.6 Average number of obstacles that users collided with on paths of dif-
   ferent difficulty levels. The labels on the $x$-axis have the following
   meanings: 1 represents data for all paths combined, while 2, 3 and
   4 represent data for the “easy”, “moderate” and “difficult” paths
   respectively . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 43
LIST OF TABLES

4.1  Variation of inter-obstacle and inter-group separation with path difficulty 35
4.2  Average number of collisions with the sidestep and head twist strategies 42
4.3  Results of paired-sample $t$-tests on the difference between the number of collisions using the sidestep and head twist strategies . . . . . . . . 44
CHAPTER 1

INTRODUCTION

1.1 Locomotion Interfaces

A *locomotion interface* is the name given to any mechanism that allows the user to travel through a virtual environment (VE) by self-propulsion, either on foot or by operating a human-powered vehicle such as a bicycle. Hollerbach [10] considers repetitive limb motion (such as occurs in walking or bicycling) as the distinguishing feature of locomotion interfaces. A graphics display is used to indicate the user’s position in the VE, generally by displaying an image of the world as seen from the user’s current position and orientation; head-mounted displays (HMDs) and CAVE-style systems [7] are commonly used display devices. The user’s movement in the real world is tracked by optical, mechanical or other means and converted into a motion through the virtual world.

Locomotion interfaces are energy-extractive devices: that is, the user has to expend a significant amount of energy to move through the virtual world. Other interfaces that allow travel through virtual worlds may not require such energy expenditure. For example, Peterson et al. [20] developed a so-called “sufficient-motion interface”, the Virtual Motion Controller, at the University of Washington. The device consists of a concave disc on which the user stands, and works as a first-order controller, converting displacements from the center of the disc into corresponding velocities in the VE. Thus, to keep moving in a certain direction at a certain speed, the user simply needs to step off by the correct amount in that direction, and no further action on his part is required. The amount of energy expended would not increase with traversed distance. A similar case exists with
driving or flight simulator systems, or systems where a joystick is used for motion. Such interfaces allow virtual-world traversal, but the amount of energy expended by the user is minimal. In addition, repetitive limb motion is not required; thus, according to the definition, a bicycle simulator can be a locomotion interface, but not a driving simulator. The expenditure of energy is expected to increase the user’s involvement in the virtual world, and simulate aspects of real-world locomotion not otherwise possible – for example, the deterioration in decision-making ability due to fatigue.

Locomotion interfaces have a wide variety of applications, for example:

- **Virtual design.** A designer can walk through 3-D models of buildings before they are built.

- **Training.** Fire fighters or soldiers can be trained in virtual environments designed to simulate field conditions more precisely than any mock-up. Having to expend energy to perform tasks would add an extra level of realism to the system.

- **Education.** Students can maneuver through foreign countries, historical sites or the lunar surface.

- **Exercise and recreation.** Hikers may enjoy a walk through the woods in the comfort of their homes.

- **Psychophysical research.** Sensorimotor integration and navigation issues, such as the relation between optical flow and locomotion [22], can be studied through manipulation of the virtual environment.

### 1.2 Treadmill-Style Locomotion Interfaces

In a treadmill-style locomotion interface, users walk on a treadmill belt while locomoting through the virtual world. Such devices may be broadly classified into
two types: \textit{two-dimensional} (2D) and \textit{one-dimensional} (1D) treadmills. (Examples of 2D and 1D treadmills are discussed in the next chapter.) In a 2D treadmill, the treadmill belt can move in an arbitrary direction in the ground plane, thus simulating an infinite horizontal surface. 1D treadmills consist of a belt that can only move backward or forward; they are especially strong at simulating straight-line locomotion, but special techniques have to be used if the user is to be allowed to change his direction of motion in the virtual world. This is the problem of \textit{turning}, discussed in greater detail later in this chapter. First we consider a specific 1D treadmill-style interface with particular relevance to this research.

\subsection{The Treadport}

The Treadport is a 1D treadmill-style locomotion interface designed and built by Sarcos Research Corporation. The user walks on a 1D treadmill belt, and his position is tracked by means of a tether attached to his back. The first generation Treadport, consisting of a commercial treadmill and an active mechanical tether, was used in the Dismounted Infantry Training Program of the U.S. Army [8], as well as for conducting research into locomotion interfaces at the University of Utah [6, 27]. An improved version of this device, the second generation Treadport, is currently being used in our labs. The new Treadport consists of a bigger belt (6 feet $\times$ 10 feet as compared to 4 feet $\times$ 8 feet for the old one), a more responsive and lower friction mechanical tether, and more accurate position sensors than the old one [11]. A CAVE-style, back-projected display, consisting of a front screen and two side screens (a total field of view of approximately 180 degrees), is used for rendering the graphics (see Figure 1.1). The strap in the figure that goes from the user's back up to the ceiling is a restraint mechanism. It works like a seat belt, locking when it is jerked down with a sufficiently high acceleration, and is used to prevent the user from falling on the belt in case he loses his balance.
1.3 Locomotion Rendering Issues

Locomotion rendering may be defined as the presentation of mechanical stimuli to simulate normal locomotion. Aspects of locomotion that might be rendered are discussed in [10]. 1D and 2D treadmills are considered below in the light of each of these issues.

1. **Forward motion and turning.** Motion along a straight line is the particular strength of 1D treadmills. While 2D treadmills allow the user to walk in arbitrary directions, there are issues, such as user instability due to the centering action of the belt, that have not been resolved yet. 1D treadmills, on the other hand, suffer from relatively unnatural turning because of the one-dimensional belt motion.

2. **Virtual slopes and uneven terrain.** If the treadmill belt can both pitch and roll, as in Noma’s ATLAS system, walking at any angle to a slope can be simulated. On the current Treadport, although the tilt mechanism is not operative, appropriate pulling and pushing forces are applied with the help of the active tether to simulate slope. A study has found these to be quite
convincing [27]. Uneven terrain (e.g., stairs) is difficult to represent on both types of treadmills. The Ground Surface Simulator of Noma et al. [18] employs a linear treadmill, with a flexible belt that can be deformed from underneath by vertical stages, and thus is particularly good at representing uneven terrain.

3. **Collisions with obstacles.** The user may be prevented from walking through virtual obstacles simply by stopping the belt, but this will not prevent him moving forward in the real world even if no virtual locomotion results. Applying a reaction force with an active tether such as on the Treadport is likely to be more effective. 2D treadmills will likely be less effective than 1D treadmills at enforcing such unilateral constraints because of the relative difficulty of placing an active tether on the user.

4. **Allowable body postures.** All locomotion interfaces do not allow the full set of body postures that a user may adopt during real-world locomotion. Although the relatively large walking area on the Treadport allows actions, such as crawling, that may not be possible on smaller devices, the tether does limit the user’s choice of body postures to some extent (for example, turning around completely is not possible). This must be weighed against the proven advantages of using such a tether. In addition, devices which use other methods of tracking the user have to operate within the limitations of those devices (line of sight issues with optical trackers, for example), and this will limit allowable body postures too.

1.4 **The Problem of Turning with 1D Treadmills**

While 1D treadmills have their advantages, especially for motion along a straight line, turning is a distinct problem. Since the treadmill belt can move only forward or back, special strategies have to be devised to allow motion in a direction other than that of the belt. The most basic artifact of turning on a 1D treadmill is
that of re-indexing. Any action that the user performs to effect turning must be compensated for with a similar “reverse” action before the user can turn in the opposite direction. For example, if the user steps off to the left side of the belt to indicate that he wants to turn left, he must come back to center before he can indicate a right turn by stepping to the right; if a leftward body twist indicates a desire to turn left, the user must first come back to a front-facing orientation before he can indicate a right turn by a corresponding twist of the body. Re-indexing is a fundamental problem with turning on 1D treadmills and cannot be completely eliminated; however, strategies that mitigate its effects are likely to be preferable to those that do not. This research explores possible turning strategies for the Sarcos Treadport.

1.5 Discussion of Control Issues in Turning on 1D Treadmills

Templeman et al. [26] divide any control technique for virtual locomotion into two parts: the control action made by the user (e.g., stepping off to the side of the belt or walking in a straight line), and the controlled effect produced by the system (e.g., turning the user in the virtual world, or translating him forward or back). These terms are used in the discussion below.

Since 1D treadmills make it difficult to turn in a completely natural fashion, other control actions must be used to signal that the user wishes to turn. Actions such as using a joystick or pointing with the hand in the desired direction of motion are not considered since they seem too obviously artificial. The user can signal desire to turn, then, by means of at least the following motor control actions:

1. Twisting the head
2. Twisting the torso
3. Twisting the feet
4. Stepping to the side of the belt (also called sidestepping)

The first three actions are obviously not independent of each other: a twist of the torso is most often accompanied by a twist of the head (but not necessarily vice versa); twisting the feet is very difficult to do without twisting the torso as well; and so on. Of these, twisting the feet to any significant angle on the belt is difficult because of the uni-directional belt motion; the other actions can be used to signal turning, possibly in conjunction with each other.

The conversion of one or more of these control actions into the corresponding controlled effect, turning, can be done using either of the following control modes:

1. Rate control

2. Proportional control

*Rate control* is a control strategy in which the “extent” to which the control action is performed determines the *rate* at which the controlled effect takes place (steering wheels and joysticks work this way). For example, if rate control was used with a head twist strategy for turning, the amount by which the user twisted his head would determine the rate of his turn in the virtual world.

*Proportional control*, in contrast, results in a controlled effect that is *proportional* to the extent of the control action. With proportional control in the above example, the angle of turn in the virtual world would be proportional to the amount of head twist by the user.

Any combination of the control actions listed above can be used in conjunction with either rate control or proportional control to effect turning.

1.6 *This Research*

This thesis addresses the problem of turning on the second-generation Treadport. Two turning strategies are presented for the Treadport. The first is a restricted
version of the turning strategy used on the first generation of the device (described in greater detail in the next chapter): sidestepping is the control action, and rate control is the control mode. The second strategy is a new one that uses the angle of torso twist and the angle of head twist of the user to determine when turning is desired and how much to turn. Proportional control is used as the control mode, but with the novel inclusion of a “trigger” condition to initiate turning. The research hypothesis is that this new strategy will allow users to turn “better” on the Treadport. To measure quantitatively the benefit of the new strategy over the old one, an experiment was conducted with subjects having to perform an obstacle-avoidance task while navigating a VE on the Treadport. At a more general level, the purpose of the experiment was to compare the relative benefits of rate and proportional control when applied to turning on 1D treadmills. The experiment and its results are discussed in a later chapter.

1.6.1 Scope

The general problem of turning on 1D treadmills is obviously a vast one. Many design possibilities exist with regard to the control actions that can be used to indicate turning and the properties of the turning effect that occurs as a result of such actions (the rate at which turning occurs, the extent of the turn, etc). While it is not feasible to exhaustively explore this space within the scope of this thesis, it is possible to obtain an idea of what types of turning strategies are better than others by sampling suitably from this space. The strategies presented in this thesis represent examples of the two possible control modes (rate control and proportional control), and the intent is to compare the class of turning strategies that use proportional control to those that use rate control by means of the experiment described. However, no claim is made that these turning algorithms are the best possible in their categories. Rather, the attempt was to keep both strategies at
roughly the same level of complexity so as to allow a fairer comparison between
classes of strategies. The experiment described was conducted at normal walking
speeds and in relatively simple virtual environments. It is possible that a completely
different turning strategy may perform better for turning when the user is at rest,
or for very high user speeds; such cases will not be covered by this research.

Finally, turning in the real world can occur with various body parts in many
different orientations; only turning that involves a certain subset of such body part
orientations will be covered. Restrictions on the types of turning considered are
discussed once the turning strategies to be compared are described in Chapter 3.

1.7 Road Map

This thesis addresses the problem of turning on the Treadport. The next chapter
discusses the background relevant to this work. Chapter 3 describes the turning
strategies that were compared by means of an experiment. The experiment itself
and its results are discussed in Chapter 4. Finally, Chapter 5 summarizes the
conclusions of this research and outlines questions that need to be explored further.
CHAPTER 2

BACKGROUND

2.1 Previous Locomotion Interfaces

Several different types of locomotion interfaces have been built. This section describes the most common categories these devices fall into, with examples of each.

1. **Pedaling device systems** may employ a stair-stepper device (such as in the OSIRIS system of Lorenzo et al. [17]), or a bicycle. Brogan et al. [3] have implemented a system consisting of a bicycle mounted on a tilting platform, with a tilt range of ±12 degrees, to simulate going up and down hills.

2. **Walking-in-place** is a common metaphor for traversing a virtual environment on foot. Slater, Usoh and Steed [25] developed a walking-in-place system that employed a neural network to recognize when the user took a virtual step by considering the action of head bobbing, although error rates of 10% were reported. More recently, Templeman, Denbrook and Sibert [26] describe their walking-in-place system called “Gaiter”, that allows a wide variety of natural actions in addition to locomotion through the VE by means of in-place steps. Excess leg motion – defined as any motion of the legs that does not participate in physical displacement of the body – is used to determine the amount of virtual displacement that each step produces, with force sensors on the feet used for segmenting successive steps. Iwata has experimented with devices that involve sliding the user’s feet over the ground in some fashion to move through the virtual world. An earlier implementation employed roller skates
on the user’s feet [13], while a later version used shoes with low-friction film on the soles [14].

3. Mounting each foot on separate three degree-of-freedom (DOF) foot platforms is another, very general solution that allows representation of uneven and soft terrain. In Iwata’s GaitMaster [15], the user’s feet are mounted on two 3-DOF platforms. The platforms themselves are mounted on a turntable to accommodate turning. However, safety is a major concern with such devices.

4. Treadmill-style interfaces are yet another technique by which the user can traverse large virtual spaces while remaining confined to a small real-world space. As explained before, in such devices, users walk on a treadmill belt while locomoting through the virtual world. While early treadmill-style interfaces like those in Brooks’ Architectural Walkthrough project at UNC [4] employed a non-motorized treadmill, more recent implementations of such devices usually employ an active treadmill belt and control algorithms that vary the belt velocity to keep the user centered on the belt surface. Below we consider examples of 2D and 1D treadmills.

2D treadmills. The most natural way of implementing a treadmill-based locomotion interface is by having a treadmill belt that can move in an arbitrary direction in the ground plane. The user can then walk in any direction, and the belt can move accordingly to keep him centered. Such two-dimensional devices have been implemented, generally by using two sets of treadmill belts moving perpendicular to each other: for example, the Omni-Directional Treadmill of Darken, Cockayne and Carmein [8] and the Torus Treadmill of Iwata and Yoshida [15]. Issues such as loss of user balance due to the centering action of the belt [8], or the limited surface area available for walking [15], have not been resolved fully, however.
1D treadmills. One-dimensional treadmills offer distinct advantages over their 2D counterparts for locomotion along a straight line in the virtual world. The user walks on a treadmill belt that can only move forward or back; the user’s position may be tracked using optical, magnetic or ultrasonic trackers, in the same manner as with other locomotion interfaces. In addition, mechanical tracking may be achieved by having a tether attached to the user’s back as in the Sarcos Treadport [11]. The tether can also be used to haptically express virtual world terrain conditions such as the presence of slope [27] or to prevent the user from walking through virtual objects, by pushing or pulling appropriately on him. A less obvious use of the tether is to provide an artificial inertial force [6], similar to the inertial force that a person experiences when he accelerates on the ground in the real world (but which is absent when accelerating on a moving treadmill belt). This artificial force greatly increases the user’s stability on the belt, and a study has shown that users overwhelmingly prefer such a tether force to no force [6]. It is not clear how such a tether force may be applied on a two-dimensional treadmill, especially if the device is coupled to a CAVE-style graphics display; for this reason alone, 1D treadmills are likely to be superior to 2D ones for straight-line locomotion.

The ATLAS system of Noma et al. [18] represents an interesting modification of a traditional 1D treadmill. The user walks on a linear treadmill belt mounted on a spherical joint; thus, when the user turns, the belt swivels so that he is still walking along its direction of motion. This provides most of the advantages of straight-line walking in 1D treadmills and the relative ease of turning found in 2D treadmills, at least in theory. In practice, system lags in rotating the treadmill make turning less natural.
2.2 Rate Control vs. Proportional Control

Some form of rate or proportional control is used in practically all control algorithms. In teleoperation [24], rate control is used when the workspace of the slave manipulator is much larger than that of the master, while proportional control is used when the workspaces are of the same (or comparable) size. Rate control allows quick movement of the slave arm across its workspace but is difficult to control with any precision. Proportional control allows more precise control over the motion of the slave arm but imposes restrictions on the workspace within which the slave arm can be used. A hybrid rate-proportional control approach has been described by Salcudean et al. [23], with a coarse-motion transport robot operated in rate control mode and a fine-motion manipulator wrist, identical to the master, operated in proportional control mode.

Rate control is considered to be a key feature of passive motion interfaces such as joysticks [10]. With a joystick in rate control mode (the normal use), for example, the user only needs to keep the joystick deflected from its zero position to keep moving. If proportional control were used, a certain deflection of the joystick would only move the user through a corresponding distance in the virtual world once. The user would need to bring the joystick back to center and repeat the motion in order to move once again. Since this repeated cycling of the control action corresponds to the definition of a locomotion interface, it is considered that rate control is the key feature of passive motion interfaces while proportional control is the key feature of locomotion interfaces.

One of the strategies described in this thesis is an example of rate control, while the other is an example of proportional control.

2.3 Turning on Treadmill-Based Locomotion Interfaces

Two-dimensional treadmills allow for a natural method of turning; however, there are issues with control that have not been fully resolved. In the Omni-
Directional Treadmill (ODT) [8], users can walk in any direction in a natural manner; however, they are prone to lose balance if they turn while accelerating from rest, and especially if they decelerate to a stop and turn simultaneously, due to a misalignment between the direction of forward motion and the centering motion of the belt. Iwata [12] has proposed a washback technique to overcome this, but the technique has not been implemented, and it remains to be seen how well it performs in practice. Iwata and Yoshida [15] have implemented a Torus Treadmill, similar to the ODT, that allows arbitrary motion in two dimensions; turning is thus natural, but the device suffers from a small walking area (2 m × 1.8 m) and underpowered belt motors that limit walking speeds to 1.2 m/s. It is also not clear whether the problems associated with accelerating or decelerating during turning reported for the ODT by Darken et al. [8] have been adequately handled in the current implementation of the device: as stated before, the washback technique has not been implemented yet.

The ATLAS system of Noma et al. [18] provides a similarly natural mechanism for turning by swiveling the belt in the direction the user is turning, although doubts have been expressed about the effects of lags in rotation on the user’s turning motion [10].

One-dimensional treadmills have not been as widely reported in the literature. Both the Architectural Walkthrough project at UNC [4] and the traversed distance-estimation experiments of Witmer and Kline [28] used a non-motorized treadmill as one of the locomotion devices. In Brooks’ Walkthrough project, the direction of motion was controlled by moving a pair of bicycle handlebars. In Witmer and Kline’s experiments, users could not control their direction of motion, but were instructed to always walk forward at one of two different speeds; thus, there was no question of turning. In the Treadport used in the U.S. Army Dismounted Infantry Training Program, turning was accomplished by twisting the waist [8]. All of these
turning strategies involve some level of artifice, requiring the user to grow used to them before he can use them naturally.

2.3.1 Turning on the First-Generation Treadport

The first-generation Treadport at the University of Utah used a rate-control strategy for turning [11], and this will be treated in some detail below. Two different control actions (possibly in combination) were used to effect turning depending on the speed of the user. Figure 2.1 indicates how different control actions are required for turning at different user speeds.

- At very low user speeds (forward or backward), the rate of turning in the virtual world is proportional to the amount of torso twist of the user.

- At high speeds, the rate of turning is proportional to the amount of sideways displacement on the belt (a quantity called sidestep). This is done to avoid requiring the user to twist his body at high speeds, which can be awkward and possibly dangerous.

- At intermediate speeds, a mixture of both torso twist and sidestep is used to turn. The relative contributions of the two vary linearly with user speed, with the contribution due to twist decreasing, and that due to sidestep increasing, as user speed increases.

The user speeds, $v_1$ and $v_2$, at which the control changes behavior (the so-called “break points”) are shown in Figure 2.1. At a speed $v$ that lies within the

```
<table>
<thead>
<tr>
<th>User Speed</th>
<th>Low</th>
<th>Intermediate</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control Actions</td>
<td>Torso twist only</td>
<td>Torso Twist + Sidestep</td>
<td>Sidestep only</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>$v_1$</td>
<td>$v_2$</td>
</tr>
</tbody>
</table>
```

Figure 2.1. Control actions required for turning at various user speeds
intermediate-speed region ($|v_1| < |v| < |v_2|$), the rate of turning $R$ may be expressed in the following manner as a function of the angle of torso twist $t$ and the amount of sidestep $s$:

$$R = \frac{|v_2 - v| \cdot K_t \cdot t + |v - v_1| \cdot K_s \cdot s}{|v_2 - v_1|}$$

where $K_t$ is the proportionality constant for torso twist and $K_s$ is the proportionality constant for sidestep.

In practice, the speed $v_2$ corresponding to the upper break-point in Figure 2.1 is selected such that at normal walking speeds the user is in the high-speed region of the control; that is, where only sidestep is used to turn.
CHAPTER 3

STRATEGIES TO BE COMPARED

This chapter discusses two possible turning strategies for the Treadport, both essentially obtained by “mixing and matching” from the alternatives presented in Section 1.5. The first strategy is a restricted version of the turning strategy on the first-generation Treadport discussed in Chapter 2 and uses rate control; the second is a new strategy that uses proportional control, using both the user’s torso twist and his head twist to determine when and how much to turn.

3.1 Strategy I: Turning Using Sidestep

The turning strategy using only sidestep functions as follows: the control action used to signal turning is the action of sidestepping (stepping off to the side of the belt), and the control technique used is rate control. The amount by which the user steps off to the side of the belt will determine his rate of turn in the virtual world (see figur 3.1). To keep turning, a user may walk forward on the belt while continuing to remain sidestepped; when he is done turning, he needs to come back to the center of the belt, thus re-indexing.

Thus, given that the user is sidestepped by an amount \( s \) and that \( K_s \) is the proportionality constant for sidestep, the rate of turning \( R \) in the virtual world may be expressed as:

\[
R = K_s \cdot s
\]

This strategy functions like the old turning strategy on the Treadport, in the high-speed region.
3.2 Strategy II: Turning Using Head Twist

3.2.1 Intent to Turn and Extent of Turn

To discuss this turning strategy it is helpful to consider *intent to turn* and *extent of turn* separately. Intent to turn indicates that the user wishes to turn, while extent of turn indicates how much the user wishes to turn. The turning strategy on the first-generation Treadport, as well as the sidestep strategy presented above, do not clearly distinguish between the two, since a control action that indicates intent to turn (e.g., twisting the torso or sidestepping) implicitly indicates the extent of turn desired as well (by the amount of torso twist or the amount of sidestep, in conjunction with the time for which it is maintained). However, there is no
fundamental reason to have the same control action determine both intent to turn and extent of turn; this is utilized in the strategy below.

3.2.2 The Head Twist Strategy

In this strategy, intent to turn is determined from the angle of torso twist. The user's torso needs to be twisted beyond a certain small "epsilon" angle on either side of the front facing orientation to establish intent to turn: this ensures that the small twists that occur naturally even when the user is walking straight forward are not construed as indicating desire to turn. The extent of turn desired can then be determined from at least the following two actions:

1. The angle of head twist.

2. The angle of torso twist.

That is, the angle of turn in the virtual world is determined to be the amount of head twist or the amount of torso twist: proportional control is used with the proportionality constant set to 1.

The angle of head or torso twist must be kept constant for a certain short period before the program "guesses" that this is the angle of turn desired and the turn is initiated. If this were not done the user would be put through multiple turns while he was twisting his head or torso to align them with the object he wished to turn to.

Both these strategies were implemented. Preliminary user feedback, however, indicated that the strategy using head twist was much preferred to the one employing torso twist. (Section 3.2.5 discusses possible reasons for this preference.) As a result, no further experimentation was carried out with the torso twist strategy. The rest of this section describes the head twist strategy only, and this is the one that is meant when Strategy II is referred to.
Thus, to turn in the virtual world, the user must perform the following control actions (see Figure 3.2):

1. Twist his torso beyond the “epsilon” angle in the direction of the desired turn.

2. Twist his head in the same direction as the torso, so that the angle of head twist is equal to the desired angle of turn.

3. Keep the angle of head twist constant for a small period of time.

These conditions together form the “trigger” that initiates turning. When these conditions are satisfied the user is turned in the virtual world by an angle equal to the angle of head twist. The user’s bearing – the angle made with north in the virtual world – changes at an exponentially decaying rate: that is, the rate

![Diagram of turns](image)

Figure 3.2. Strategy II: The torso needs to be twisted enough to express intent to turn, and the angle of head twist must be constant for a sufficient period of time, in order to initiate turning.
of turning is fast at the start of the turn, but reduces exponentially as the turn proceeds.

3.2.3 Consequences of the Exponential Decay in the Rate of Bearing Change

As explained above, the rate of change of the user’s bearing in the virtual world decays exponentially with time. Mathematically, the fraction of the total angle turned, a time $t$ after the turn starts, may be represented as:

$$F = 1 - e^{-kt}$$

where $k$ is an exponential decay constant. The graphs in Figure 3.3 show how the fraction of the turn completed in a given time decreases as $k$ decreases (that is, the turn takes longer to complete).

On the current Treadport, the graphics display is on three screens, one to the left, one to the right, and one in front of the user. The effect of a turn in the virtual world of $x$ degrees is that an object that makes an angle of $x$ degrees with the user’s front-facing orientation when the turn starts (and which thus lies either to the side of the front screen or on one of the side screens) has been moved to the center of the front screen by the end of the turn. Due to the exponential decay in the rate of turning, the object will appear to move quickly toward the center of the front screen at the start of the turn, but will slow down as the turn nears completion. Presumably, the user will be drawn to turn back to center as the object that he wants to turn to moves in that direction, thus effecting a (somewhat) natural re-indexing.

3.2.4 Implementation Details

The control software for the Treadport was written in ControlShell, a component-based development system for real-time applications [21]. Applications are written
Figure 3.3. As the exponential decay constant $k$ decreases, the average rate of turning decreases and the time to complete the turn increases.
in ControlShell by connecting components together as in a data-flow diagram, specifying what input data each component receives and what output data it produces. This produces a data dependency graph that is used to determine the order in which components will be “run” at each clock tick.

### 3.2.4.1 Basic Algorithm

The basic algorithm for the turning control, executed once per clock cycle, may be expressed in C-like pseudo-code as shown below. The “counter” referred to is a software counter that is decremented when counting down once per cycle.

```c
if (turning has not been initiated)
{
    if (user’s torso is twisted sufficiently to express intent
to turn)
    {
        if (angle of head twist has remained constant since
the last cycle)
        {
            if (the counter has been started and is not at zero yet)
                decrement the counter;
            else if (the counter is at zero)
                set flag to indicate that turning has been initiated
                and record the angle of turn desired;
            else    /* the counter has not been started */
                start counter;
        }
        else    /* angle of head twist has changed since
            last cycle */
```
reset counter;
}
else  /* user’s torso is not twisted sufficiently to express
intent to turn */
    do nothing;
}
else  /* turning has been initiated */
{
    if (the angle turned so far in the virtual world is close
        enough to the desired angle of turn)
        stop turning;
    else
        change the user’s bearing by an increment appropriate for
        one cycle (explained below)
}

Thus, a turn is initiated when the torso is twisted beyond the “epsilon” angle
and the angle of head twist has remained constant for the time it takes the counter
to count down to zero. Once the turn is initiated, it continues till the angle turned
is “close enough” to the angle of turn desired. There are other ways in which the
user can terminate the turn: these are discussed below.

3.2.4.2 Termination Conditions for the Turn

The basic algorithm outlined above indicates one termination condition for the
turn: when the angle turned in the virtual world is close enough to the desired
angle. The turn will also be aborted (that is, the angle turned will not be equal
to what the program guessed was the desired angle) under one of the following
conditions:
1. While re-indexing, the user turns his head in the direction opposite to the
direction of re-indexing: that is, instead of toward the center of the front
screen, away from it; the program then infers that it has made a mistake in
guessing what the desired angle is, since the user is obviously not following
the object that lies at that angle to the center of the front screen.

2. While re-indexing, the user’s head twists on the other side of the front-facing
orientation before the object that lies at the guessed desired angle of turn has
had time to rotate to the center of the front screen; the program, similar to
the case above, then infers that it over-estimated the desired angle, and that
the user was actually interested in turning to an object at a smaller angle than
that.

If either of these conditions are satisfied, turning immediately stops; the user
can initiate a turn again by satisfying the conditions listed in section 3.2.2.

3.2.4.3 Angle Turned in One Clock Cycle

As explained previously, the rate of turning in the virtual world decays exponen-
tially with time. If $M$ is the total angle of turn desired, and $k$ is the exponential
decay constant, the angle $x$ turned a time $t$ after the turn starts may be expressed
as:

$$x = M \cdot (1 - e^{-kt})$$

In a time $\Delta t$, then, the change in the angle turned, $\Delta x$, may be easily derived
as:

$$\Delta x = M \cdot k \cdot e^{-kt} \cdot \Delta t$$

The user’s bearing (the angle made with North in the virtual world) is changed
by an amount $\Delta x$ each clock cycle, where $\Delta t$ is the clock period.
3.2.4.4 A More Precise Definition of Torso Twist

The torso twist used in the turning strategy above is measured relative to an imaginary line passing along the center of the Treadport and perpendicular to the front screen (see Figure 3.4).

The Treadport has a potentiometer that measures the angle of twist at the front end of the tether. However, in the situation shown in Figure 3.4, this potentiometer will read a twist of zero since the user is facing along the direction of the tether;

![Diagram of Torso Twist](Image)

Figure 3.4. Torso twist is measured relative to a line parallel with the side edges of the Treadport and perpendicular to the front screen.
this is clearly not what is desired since the user probably wants to turn to an object on one of the side screens. The actual twist $A$ made relative to a line parallel to the side edges of the Treadport can be calculated using the following simple formula:

$$A = T + \tan^{-1}(d_y / d_x)$$

where $T$ is the twist recorded by the potentiometer at the end of the tether, and $d_x$ and $d_y$ are the displacements from the back end of the Treadport as shown in Figure 3.4.

3.2.5 Why Head Twist Works Better Than Torso Twist in Indicating Extent of Turn

The head twist and torso twist strategies differ only in the way the extent of turn is computed. In the torso twist strategy, the desired angle of turn is set to be the angle of torso twist. This can be considered more natural than the strategy that uses head twist to determine extent of turn, since in real life people do have to turn their bodies in a particular direction before they walk in that direction. It is not easy to do this on the Treadport, however, since the feet are constrained to move along the direction of the belt (or at a small angle to it) even if the torso is twisted: turning by large angles will be difficult with this strategy. (The fact that twisting the body by large angles when walking at normal speeds on the belt is awkward is the reason that sidestep was introduced into the old turning control in the first place.)

The strategy that uses head twist to determine desired angle of turn can be considered less natural than the torso twist strategy, since the angle of turn in the virtual world is determined from the angle of head (rather than torso) twist. It must be noted, however, that the angle of head twist is taken into account only when the torso is also twisted sufficiently to express intent to turn (although not
necessarily to the same extent as the head). This means, for example, that it is still possible for the user to look around him while continuing to walk in a straight line in the virtual world. He will not experience any turn until he twists his torso in the direction he wishes to turn in.

Further, twisting the head to indicate angle of turn has the advantage that it is much easier to do for moderate to large angles than twisting the torso. When effecting a turn on the Treadport, it is observed that the user’s head generally swivels quickly in the direction he wishes to turn to, while the torso twists more slowly. The angle of torso twist lags behind the angle of head twist especially when the user is not stationary, since it is difficult to quickly turn one’s body when walking in a straight line on the belt. For large angles of turn, it may not even be possible for the user to twist his torso by that angle when walking along the belt. The result is that the angle of torso twist does not in general indicate, either quickly enough or accurately enough, the desired angle of turn. Using the head angle to guess the desired angle of turn means that we can make this determination, and thus initiate the turn, fairly quickly; this in turn means that the awkward torso twist has to be maintained for less time.

Subjective user feedback bears out these advantages of the head twist strategy. All subjects who used both strategies found it much more difficult to make large turns with the torso twist strategy, and the twist of the torso required was considered awkward and difficult to perform. Consequently, the torso twist strategy was dropped in favor of the head twist strategy.

3.2.6 Limitations and Compromises

Re-indexing is still necessary with Strategy II. An advantage of using head twist to determine angle of turn is that it makes it more likely that the user will be “looking at” the object he wishes to walk toward when he makes a turn: thus,
when this object starts moving to the center of the front screen, the user’s gaze will presumably be drawn back to center along with it. However, it must be noted that while re-indexing, the user is required to do a “reverse twist” of his torso and head; the vestibular and kinesthetic feedback that results from such actions will indicate turning in the opposite direction from that desired. The sensation of circular vection as a result of the rotating visual field during re-indexing should override this feedback to some extent.

Only certain types of turning can be performed with this strategy. In the real world it is perfectly possible to turn while the head and torso are facing straight ahead, or while they are twisted in opposite directions. Such turns are not possible with this strategy: the torso has to be twisted enough to express intent to turn, in the direction of desired turn, and the head has to be twisted in the same direction by the amount of turn desired. In practice, these limitations are not particularly restrictive.

Finally, there is always the possibility that certain actions may be misinterpreted by the software, and a turn may be made where none was desired. One of the ways this could happen is if the user tries to look back over his shoulder, so that he twists his body around. The software will take the user through a nearly 180 degree turn in this case, although the user did not intend to turn. This is not impossible with the Treadport, but the fact that there are only three screens with a total field of view of approximately 180 degrees makes it unlikely that the user will turn back by this amount, since all the graphics displayed on the screens can be viewed with a much smaller turn of the head.

3.3 Conclusion

This chapter has described one turning strategy involving rate control and one involving proportional control for the Treadport. The next chapter discusses an experiment to compare maneuvering ability using each of these turning strategies.
CHAPTER 4

AN EXPERIMENT TO COMPARE

TURNING STRATEGIES

4.1 Experiment Objective and Task

The objective of the experiment was to compare which of the turning strategies presented in the previous chapter allowed the user to better maneuver in the virtual world, as measured by an obstacle avoidance task.

Subjects were asked to walk along a straight line path, with certain obstacles along the way. Their task was to avoid as many of these obstacles as possible. The virtual paths used for the experiment are described in a later section.

4.2 Pre-Experiment Tuning

Before the experiment was conducted the parameters for the control strategies were tuned. The sidestep strategy (Strategy I) required a single parameter to be tuned: the proportionality constant for sidestep \( (K_{\text{side step}}) \). Strategy II required the following parameters to be tuned:

1. The “epsilon” angle beyond which a twist of the torso indicates intent to turn \((\epsilon_{\text{twist}})\).
2. The time for which the angle of head or torso twist needs to remain constant in order to indicate desired angle of turn \((t_{\text{constant}})\).
3. The exponential decay constant \( K_{\text{twist}} \) (discussed previously as \( k \)) that determines the average rate of turning.
Two expert users were put on the Treadport, and the values of these parameters were changed till a particular set of values felt “right” for both users. The parameter values obtained from this procedure were as follows:

- $K_{\text{side step}} = 1.5$
- $\epsilon_{\text{twist}} = 5$ degrees
- $t_{\text{constant}} \approx 1/15$ second
- $K_{\text{twist}} = 1.5$

During the experiments the control parameters were set at these values for all subjects. No subject reported feeling uncomfortable with the preset parameters.

### 4.3 Experiment Details

#### 4.3.1 Independent Measures

The independent measures for the experiment were the turning strategy (I or II) and the “difficulty level” of the virtual path which the subjects were required to traverse (three levels). The measure of difficulty level is explained later.

#### 4.3.2 Dependent Measures

The primary dependent measure for the experiment was the number of obstacles that subjects collided with when traversing the virtual path. Of secondary importance were the time taken for traversal of each path, and the average distance traversed per path.

#### 4.3.3 Design

The experiment employed a within-subjects $2 \times 3$ factorial design on both independent factors: that is, all subjects were required to traverse paths of all...
three difficulty levels, using both turning strategies. The order of strategies was shuffled between subjects so that half the subjects were presented Strategy I first and half were presented Strategy II first. All paths were presented in the same order, in ascending order of difficulty; this was with the intention that subjects would encounter paths of increasing difficulty with time, commensurate with their increasing proficiency with the turning strategy.

4.3.4 Procedure

Subjects were first asked to read a written description of the experiment; a copy of this document may be found in the Appendix. They were then allowed to practice walking on the Treadport for about five minutes. This was necessary since most subjects were naive users. Following this, they were allowed to practice on a test virtual path to get used to the turning strategy. They were not required to perform the task during this time. Following about two to three minutes of this training, they were given practice with performing the experimental task. A virtual path similar to the ones used in the experiment was presented, and subjects were required to travel from one end of the path to the other with as few collisions as possible with the obstacles along the path. No force feedback to simulate reaction forces was implemented; thus, subjects could “pass through” any of the obstacles. Subjects were allowed to traverse the virtual path three times, allowing them to refine their strategy to avoid obstacles as they became more proficient at the task. This was then followed by the actual experiment. Three paths were presented in ascending order of difficulty, and subjects were allowed to walk along each path twice, trying to avoid as many obstacles as possible each time. The number of obstacles they collided with was recorded for each path. Subjects were asked specifically to maintain a comfortable walking pace throughout the experiment, and in particular not to slow down or stop if they collided with a number of obstacles in
succession. During the experiment, the experimenter verbally directed subjects to maintain their walking speed if they appeared not to be following these instructions.

The same steps were then repeated for the other turning strategy, except that this time the subjects were not asked to practice walking on the Treadport. Specifically, two to three minutes of practice with the turning strategy, plus three trials on performing the task using that strategy, were followed by six trials on the same three paths as for the first strategy. The average time for the entire experiment was approximately 40 minutes.

### 4.3.5 Hardware

The graphics were rendered on an SGI Onyx2 running IRIX 6.5 with 4 R12000 CPUs, 2 InfiniteReality Engines and 512 MB of RAM, and displayed on three CAVE-style, back-projected screens with a total field of view of approximately 180 degrees. The controller for the Treadport ran on a Motorola 68000 processor board with 4 MB of RAM and using VxWorks v.5.2 as the operating system. The user’s head position was tracked for Strategy II using the Intersense IS-600 head tracker. The Treadport has been described previously.

### 4.3.6 Virtual Environments

The graphics program for the experiment was written in World ToolKit [9]. The virtual environments used were of a single basic type, with minor differences to vary the difficulty level. An overhead view of a sample path used in the experiment is shown in Figure 4.1.

The lines in the figure represent walls (both parallel and perpendicular to the direction of user motion along the path) that the user was supposed to avoid.

The same path from the user’s perspective would look as shown in Figure 4.2.

As can be seen from Figure 4.1, the obstacles along the path (excluding the side walls) were grouped together in groups of three, with a relatively large distance
between groups of obstacles. This was to allow the user to navigate each group of obstacles practically independent of each other. The difficulty level of the paths was then adjusted by varying the separation between the obstacles within a group. The inter-obstacle separation within a group, and the separation between groups of obstacles, varied with path difficulty as shown in Table 4.1.

All paths used in the experiment contained 3 groups of 3 obstacles each. Each path measured 60 m in length and 3 m in width. All obstacles (including the side walls) were 3 m high. The walls perpendicular to the user’s direction of motion were 1.9 m wide; thus, there was a gap of about 1.1 m between such a wall and the side wall (see Figure 4.1). As explained later, however, the user was represented
Table 4.1. Variation of inter-obstacle and inter-group separation with path difficulty

<table>
<thead>
<tr>
<th>Difficulty Level</th>
<th>Inter-obstacle separation</th>
<th>Inter-group separation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>4.5 m</td>
<td>6 m</td>
</tr>
<tr>
<td>Moderate</td>
<td>4 m</td>
<td>7 m</td>
</tr>
<tr>
<td>High</td>
<td>3.5 m</td>
<td>8 m</td>
</tr>
</tbody>
</table>

in the program by a rectangular parallelepiped 0.1 m in width; thus, the effective room for the user to pass through was 1 m.

The side walls along each path were colored red. A texture was applied to walls perpendicular to the direction of user motion. Different textures were applied to walls flush with the left and the right edges of the path, as well as to walls occurring in paths of different difficulty. The ground resembled a typical road, with a double yellow line running along the center of the path (see Figure 4.2).

![Figure 4.2. User’s perspective of a sample path for the experiment](image-url)
4.3.7 Collision Detection

Collision detection was implemented in order to count the number of obstacles the user collided with, although no force feedback was given in the event of a collision. The user was represented as a rectangular parallelepiped of height 2 m, width 0.1 m and thickness 0.05 m, and this was checked for collision with all walls along the path (each represented as a rectangle with zero thickness) during each iteration of the graphics loop. This meant that collision detection could never be performed at greater than the graphics update rate (about 30 frames a second), so in theory if a user travelled fast enough, he could completely “pass through” a wall between two successive checks for collision. In preliminary simulations, it was found that the component of the user velocity normal to the wall would have to be around 2.6 m/s in order for this to happen. In the actual experiment, none of the subjects even approached this critical speed, and the problem never arose in practice.

Once it was determined that the user had collided with a wall, that wall was not checked for collision in future. This was to prevent the reverse situation of that described above: that is, when the user was moving so slowly that two successive collision checks would both detect him in collision with the same wall.

The side walls along the path presented another problem, since it was not clear where to consider that one segment of the wall ended and the next began. This was important because the user could collide only once with each segment of the wall before that segment was removed from consideration for future collision checks. If each segment of the wall was made too large, then two distinct collisions with the side wall that happened to lie within the same segment would not both be recorded; on the other hand, if the segments were made too small, the user could be grazing

\[\text{1The orientation of the parallelepiped was always perpendicular to the user's facing at any given time: that is, oriented in the same way as the user's body.}\]
along the wall and this would be recorded as several distinct collisions. A reasonable size for each segment was taken as the length of a single obstacle group, plus the inter-group separation. Thus, the side walls were divided into segments as shown in Figure 4.3.

Wall segments of this length seemed to work well in practice, mostly avoiding the problems mentioned above.

4.3.7.1 Width of the Rectangular Parallelopiped

As stated above, the dimensions of the rectangular parallelopiped representing the user were 2 m (height) \( \times \) 0.1 m (width) \( \times \) 0.05 m (thickness). It appears at first glance that the width of the parallelopiped is too narrow. Although it is true that a typical person’s shoulders are certainly wider than 0.1 m, this value was chosen after a number of trials with different widths. An analysis of what the graphics program does when it projects an image onto the screen reveals why.

As shown in Figure 4.4, the planes forming the viewing pyramid for the front screen are obtained by connecting the user’s eye point with the four edges of the front screen. If the user is in the center of the treadmill belt, this will be a symmetric pyramid. The same procedure applies, however, even if the user is displaced off center. In that case the pyramid may be an oblique pyramid, and elongated or shortened compared to the one in the figure; however, it will still be formed by connecting the user’s eye point to the four edges of the front screen.

It must also be noted that when the user is walking on the Treadport, there is a compelling sense of having walked through an object when the user’s eye point passes through that object. When this happens, it can be seen from Figure 4.4 that the object will be projected to fill the whole of the front screen. On the other hand, if the object misses the eye point by even a small amount it will lie outside the viewing pyramid for the front screen and will be projected onto one of the side
Figure 4.3. The side walls along the path are divided into several different segments for collision detection
Figure 4.4. The view pyramid for the front screen is formed by connecting the user's eye point with the four edges of the screen.
screens. It then appears to the user that he has successfully avoided the object. See Figure 4.5.

Consider what would happen if the user was represented by a rectangular parallelopiped with a width comparable to the width of a person’s shoulders. Suppose there is an spherical object as shown in Figure 4.5, and that the user is walking toward it at a constant speed so that he is always in the center of the treadmill belt. Successive positions of the object are shown as it moves towards the user. It can be seen that the object is originally within the viewing pyramid for the front screen but that it passes outside this pyramid as the user gets closer to it. When collision occurs (at the point labeled $P$ in the figure), the user’s shoulders pass through the object, yet because it is outside the viewing pyramid it is not projected onto the front screen. The result of this is that the user feels that he has avoided the object (since he did not have a sense of “passing through” it), but the software records a collision, since his shoulders have brushed the object. Thus, although representing the user by a human-sized parallelopiped is desirable, it leads to a disconnect between what the user feels he has collided with (any object that he has “walked through”, that is any object through which his eye point passes), and what the software considers him to have collided with (any object that the human-sized parallelopiped has passed through). In order to make the results of the collision check by the software more consistent with the user’s subjective experience, different widths for the rectangular parallelopiped were tried out. It was found that 0.1 m was the largest width at which there was not a noticeable disparity between when the user thought he had collided with an object and when this was recorded by the software. This was therefore taken as the width for the rectangular parallelopiped in the program.
Figure 4.5. An object that lies to one side of the eye point will not be projected onto the front screen even if the user grazes it with his shoulder.
4.3.8 Subjects

Twelve subjects (seven males and five females) were used to conduct the experiment. Subjects were students or faculty at the University of Utah. Two of the test subjects were expert users; the other subjects were either first-time users, or relatively inexperienced users, of the Treadport.

4.4 Results

4.4.1 Number of Collisions

Figure 4.6 shows the average number of obstacles that users collided with on paths of different difficulty levels considered separately, as well as for all paths combined. Since each subject was presented three different paths, each at one level of difficulty, and allowed to traverse each path twice, there was a total of six data points per subject. The graph for all paths combined is thus drawn from a total of 72 data points, while the graphs for paths of each difficulty level are drawn from 24 data points each.

The average number of obstacles that users collided with using the two strategies is summarized in Table 4.2.

It seems obvious that the head twist strategy performs much better than the sidestep strategy, both when all the data is taken together and when the data for each difficulty level is considered separately. As the difficulty level of the path increases the performance of the sidestep strategy worsens appreciably; the

Table 4.2. Average number of collisions with the sidestep and head twist strategies

<table>
<thead>
<tr>
<th>Difficulty level</th>
<th>Sidestep strategy</th>
<th>Head Twist strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>All paths combined</td>
<td>3.44</td>
<td>0.88</td>
</tr>
<tr>
<td>Low</td>
<td>1.96</td>
<td>0.88</td>
</tr>
<tr>
<td>Moderate</td>
<td>3.83</td>
<td>0.63</td>
</tr>
<tr>
<td>High</td>
<td>4.54</td>
<td>1.13</td>
</tr>
</tbody>
</table>
Figure 4.6. Average number of obstacles that users collided with on paths of different difficulty levels. The labels on the x-axis have the following meanings: 1 represents data for all paths combined, while 2, 3 and 4 represent data for the “easy”, “moderate” and “difficult” paths respectively.
head twist strategy performs worst on the “difficult” path and (counter-intuitively) better on the “moderate” path than the “easy” path, but the difference between its performance on the moderate path and the difficult path is not as pronounced as for the sidestep strategy.

The two strategies were compared by performing paired-sample \( t \)-tests on the data for all paths combined and for paths of each difficulty level. The results are summarized in Table 4.3.

The difference between the two strategies is highly significant, and as the path difficulty increases the \( t \)-value, and thus the significance level of the results, increases as well.

### 4.4.2 Learning Effects

The experimental data was analyzed to determine whether the order in which the strategies was presented to users had a significant effect on the results: in other words, was there a significant degradation or improvement in performance between the strategy that was presented earlier and the one that was presented later?

The average number of obstacles collided with using the strategy that was presented earlier (irrespective of which one it was) was 2.46, while this number was 1.86 for the strategy presented later. A paired-sample \( t \)-test showed that this difference was below significance \((t = 1.55, p = 0.1267)\). We may conclude that any learning that occurred did not affect the results to a significant degree.

Table 4.3. Results of paired-sample \( t \)-tests on the difference between the number of collisions using the sidestep and head twist strategies

<table>
<thead>
<tr>
<th>Difficulty level</th>
<th>( t )-value</th>
<th>( p )</th>
</tr>
</thead>
<tbody>
<tr>
<td>All paths combined</td>
<td>10.37</td>
<td>&lt; 0.0001</td>
</tr>
<tr>
<td>Low</td>
<td>3.00</td>
<td>0.0063</td>
</tr>
<tr>
<td>Moderate</td>
<td>7.78</td>
<td>&lt; 0.0001</td>
</tr>
<tr>
<td>High</td>
<td>9.75</td>
<td>&lt; 0.0001</td>
</tr>
</tbody>
</table>
4.4.3 Effect of Difficulty Level

To determine whether the difficulty level affected performance, paired-sample t-tests were carried out on intra-strategy data, considering the difficulty level as the independent variable.

For the sidestep strategy, the difference between the easy and moderate paths was highly significant ($t = -4.63$, $p < 0.0001$), as was also the difference between the easy and difficult paths ($t = -6.97$, $p < 0.0001$). The difference between the moderate and difficult paths, however, was not significant ($t = -1.43$, $p = 0.1654$).

For the head twist strategy, neither the difference between the easy and moderate paths ($t = 1.03$, $p = 0.3136$) nor between the easy and difficult paths ($t = -0.76$, $p = 0.4578$) was significant. However, the difference between the moderate and difficult paths was significant ($t = -2.30$, $p = 0.0306$).

It was concluded that the difficulty level of the paths did impact performance, but less so for the head twist strategy than the sidestep strategy. When the paths were relatively easy users seemed to be able to “get by” with even the sidestep strategy, but as the difficulty level went up their performance degraded rapidly. There was more of a “performance buffer” with the head twist strategy: increasing level of path difficulty did not degrade the performance quite as much.

4.4.4 Traversal Time and Traversed Distance

Due to some missing data points, analysis of users’ traversal time could only be performed with 64 (rather than 72) samples. The average time a user took to traverse a path using the sidestep strategy was 56.38 seconds; with the head twist strategy this was 51.54 seconds. Somewhat unexpectedly, this proved to be a highly significant difference ($t = 7.78$, $p < 0.0001$): users took significantly less time to traverse a given path using the head twist strategy than the sidestep strategy. Since users were asked to maintain a constant pace throughout the experiment
(and this would likely be the same for both strategies), the significant difference observed could be because subjects in general took a longer route with the sidestep strategy than the head twist strategy. If subjects were unable to control their motion precisely with the sidestep strategy (as indicated by the significant difference in the number of collisions), they would be more likely to veer off the optimum path through the virtual world, and this would increase their traversal time as well.

This hypothesis was borne out by an analysis of the total distance traversed by subjects with the two strategies. With the sidestep strategy, subjects on average travelled a distance of 60.88 m per path; with the head twist strategy this distance was 58.33 m.² (Note that many subjects stopped immediately after clearing the last obstacle, several meters short of the end of the path; this probably accounts for the otherwise surprising fact that these lengths are approximately the same as the straight-line distance from one end of the path to the other.) The difference in the traversed distances was highly significant ($t = -6.54, p < 0.0001$): that is, subjects did indeed travel a significantly shorter distance with the head twist strategy than with sidestep strategy, at least partially explaining why there was a significant difference in traversal times.

The difference in traversal times also meant that the better performance observed with the head twist strategy was not simply a matter of subjects taking more time to negotiate the path, and thus doing so more carefully.

### 4.5 User Preference

All subjects except one preferred the head twist strategy to the sidestep strategy; the one dissenter felt that both head twist and sidestep strategies were equally good (although her performance in the experiment, as measured by the number

---

²The user position was recorded approximately three times a second, and the traversed distance was calculated by considering the user path to be made up of straight-line segments between these points.
of collisions, was worse with the sidestep strategy than the head twist strategy). Most subjects felt that having the head angle determine the angle of turn allowed them to simply “walk towards” an object – especially important to make the sharp turns required in the experiment. This was a definite advantage over the sidestep strategy, using which many subjects tended to turn too far when trying to avoid an obstacle, colliding with the side wall as a result. One subject lost her bearings completely while navigating one of the virtual paths using the sidestep strategy: she was unable to estimate when to re-index in order to stop turning, and kept going around in circles. Most people, on the other hand, adapted quickly and easily to the head twist strategy.

4.6 Summary of Results

The head twist strategy seemed to significantly outperform the sidestep strategy. Less collisions were recorded with the head twist strategy at all levels of path difficulty, and the degradation in performance as the level of path difficulty increased was much less with the head twist strategy than with the sidestep strategy. Users also took significantly less time to traverse a given path with the head twist strategy than with the sidestep strategy, and the path followed was significantly shorter. Learning effects were not significant. Finally, all users but one preferred the head twist strategy to the sidestep strategy.
CHAPTER 5

CONCLUSIONS AND FUTURE WORK

It seems clear from the experimental results presented in the previous chapter that the head twist strategy performs better than the sidestep strategy in allowing users to maneuver through the virtual world. In general, it appears that it is easier to maneuver using proportional control than rate control. User feedback strongly indicated that the ability to “look where you were going” was an important win for the head twist strategy. With the sidestep strategy users had to essentially perform some kind of mental integration in order to determine how much of a sidestep, if maintained for the time they had going at their current pace, would be sufficient to allow them to pass through the available gap between two obstacles.\(^1\) With the head twist strategy this reduced to simply looking at, and walking towards, the gap as best they could. This was more natural, and in the limited time they had to avoid the obstacles, a crucial difference. Not only did users hit less obstacles using proportional control, they also walked a significantly shorter distance and took significantly less time.

This research represents a first step toward answering the question of the “best” turning control for 1D treadmills. Although the result presented here is only applicable with reference to the narrow class of turning controls presented, the user speeds at which the experiments were conducted, and the specific characteristics

\(^1\)Users were not allowed to sidestep and then wait there till the object they wanted to walk toward rotated to the center of the screen. The experimenter instructed them to maintain their normal walking pace whenever they slowed down.
of the Treadport itself, it provides a foundation for more such work under different conditions. Below I list a number of other questions that remain to be answered. Some of these are specific to turning on 1D treadmills; others apply to locomotion interfaces in general.

1. What other turning strategies might work well for 1D treadmills across a broad range of applications?

The turning strategies presented here represent a very small class of possible strategies: those using proportional or rate control implemented in a certain way to achieve turning. It is important to note that even within the class of strategies that use proportional (or rate) control there are different ways of implementing the control; for example, in preliminary testing other strategies were tried that used proportional control, but rotated the graphics back with the user only when he re-indexed of his own accord. There are undoubtedly many other formulations, and it has not been proved that the exponential decay strategy presented here works better than any other.

A single performance metric – maneuvering ability – was used in the experiment, and significant differences were found between the head twist and sidestep strategies. Using other performance metrics may not yield such clear differences: a preliminary study conducted on the Treadport found no significant difference between the head twist and sidestep strategies when the performance measure was how well users could maintain a sense of their orientation in the virtual world.² The data sample was admittedly small (16 data points), but it is reason to perform tests using as many orthogonal measures

²It is likely that the performance metric for the preliminary experiment was less than ideal, since just walking on the Treadport provides a powerful enough sense of immersion in the virtual environment that the effects of the turning strategy on orientation estimates are likely to be minimal, and this may have diluted the results to below significance.
of performance as possible before claiming that one strategy is “better” than another.

2. **Are different turning controls best for different user speeds?**

In the experiment reported in this thesis, users were constrained to keep moving at a normal walking pace. It has not been proved that the result obtained generalizes to other user speeds.

3. **What other measures of “goodness” of a turning strategy are relevant?**

Much work in the literature has focused on the effect of specific properties of the interface and the virtual environment on the extent of path integration (see for example, [1, 5, 16, 19, 20]; [1] contains a number of references to similar work). The accuracy of path reproduction [15], as well as performance on tasks that measure route knowledge or maneuvering ability [20], have also been measured. This research was primarily concerned with maneuvering ability. There will undoubtedly be other criteria to judge whether a turning strategy is “good” or “bad”; see [2] for a discussion of possible performance metrics. It is entirely possible that subjective metrics (e.g., user comfort) will prove to be more important than quantitative measures of performance, and this deserves investigation.

4. **How do the strategies described in this thesis compare to real-world turning?**

No attempt was made in this thesis to use real-world performance as a benchmark. For the experiment that was carried out, it is almost impossible to see how such performance could have been less than perfect – that is, if people were asked to navigate a corridor with obstacles (as used in the experiment) 3.5 to 4.5 m apart in the real world, it would be difficult not to avoid them all. However, it may not be clear how good real-world turning is with respect to
these strategies when other performance metrics are used. Before we test how well competing strategies allow us to perform certain tasks, it would be helpful if we had a measure of real-world performance at those tasks as a guide.

Very little work has been carried out previously on comparing alternative controls for a specific component of locomotion on a particular device. Most work in the area has had an “all-versus-nothing” approach, in that comparisons are made between one highly impoverished way of traversing a VE (for example, a joystick) versus a much richer method (for example, using a sufficient-motion controller like the VMC [20] or an active device such as the Torus Treadmill [15]). Under these circumstances, it is not surprising that most such research finds a strong tilt in favor of devices that require whole-body (or near whole-body) participation by the user in order to move through the VE. However, devices that allow real walking through VE s are no longer a curiosity, but fairly commonplace in major research labs. It is not appropriate any longer to ask if such devices provide advantages over devices that require less body interaction to navigate the VE, since this has consistently been found to be the case. The more important questions now concern how best to overcome the limitations of such locomotion interfaces, with the ultimate goal of making locomotion on such devices indistinguishable from walking in the real world. The answers will probably be qualified by the specific device or class of devices (different techniques are likely to work best for 2D and 1D treadmills, for example) as well as by what the definition of “best” is. Nevertheless, it is important to compare alternatives for different locomotion controls treating the existence of such devices as a given, rather than to ask if such devices are useful at all. This research has attempted to take a step, however small, in that direction.
APPENDIX

INSTRUCTIONS TO PARTICIPANTS

In the following experiment you will be asked to navigate a 3-D virtual world. The device you will be using to perform the experiment is called the Treadport. You will have a chance to practice on the Treadport before you perform the experiment.

The objective of the experiment is to compare two different strategies on the Treadport. The strategies being compared will be explained to you by the experimenter once you get on the Treadport.

We will be measuring how well you can maneuver through the virtual world using each of the two strategies. The virtual world will be a straight path, with obstacles in the form of walls obstructing your progress. Your task will be to walk from one end of the path to the other while avoiding as many of the walls as possible. This includes the side walls along the path.

We will require you to perform the same task using each strategy. You will be allowed to practice on the Treadport with a strategy before we perform the experiment using that strategy. This practice will consist of two stages:

1. First you will be asked to walk along a virtual path, to simply get used to the strategy. There will be no obstacles to avoid.

2. The second phase of training will require you to navigate three virtual paths, avoiding obstacles along the way.

Following the training you will be asked to walk along six virtual paths, avoiding as many obstacles as possible on each path. These steps will then be repeated for the other strategy. The total time for the experiment should be less than one hour.
Important:

We ask that you maintain a comfortable walking pace throughout the experiment. In particular, *do not* slow down or stop even if you run into a number of walls in succession. Doing so will invalidate the results since the object of the experiment is to measure your maneuverability at normal walking speeds.

This concludes the overview of the experiment. Feel free to ask the experimenter any questions you may have. Thank you for your participation in this experiment, and good luck!
REFERENCES


