

**REAL-WALKING MODELS IMPROVE  
WALKING-IN-PLACE SYSTEMS**

Jeremy D. Wendt

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Computer Science.

Chapel Hill  
2010

Approved by:

Frederick P. Brooks, Jr.

Mary Whitton

David Adalsteinsson

Gary Bishop

Troy Blackburn

Gordon Pitz

© 2010  
Jeremy D. Wendt  
ALL RIGHTS RESERVED

# ABSTRACT

JEREMY D. WENDT: Real-Walking Models Improve Walking-In-Place Systems  
(Under the direction of Frederick P. Brooks, Jr.)

Many Virtual Environment (VE) systems require a walking interface to travel through the virtual world. Real Walking – the preferred interface – is only feasible if the virtual world is smaller than the real-world tracked space. When the to-be-explored virtual world is larger than the real-world tracked space, Walking-In-Place (WIP) systems are frequently used: The user’s in-place stepping gestures are measured and identified by the WIP system to move virtual viewpoint through the virtual world. When the system-generated forward motions do not match the user’s intended motions, the user becomes frustrated and the VE experience degrades.

This dissertation presents two real-walking-based models that enable WIP systems to generate walking-like speeds. Direction from in-place stepping gestures is not covered. The first model (GUD WIP) calculates in-place step frequency – even when only a portion of a step has been completed. The second (The Forward Walking Model) measures a user’s step-frequency-to-walk-speed function in real-time from head-track-data alone. The two models combined enable per-user-calibrated real-walking-like speeds from in-place stepping gestures.

This dissertation also presents validation studies for each model and a usability study that demonstrate that WIP systems that employ these models are better than a previous speed-focused WIP system.



To the joy of my life, my wife Heather. When I doubted I could do this, I leaned on her confidence in me.



# Acknowledgements

Each of the following people deserve a lot of credit for helping me complete this work. I appreciate each of them considerably more than the following brief sentences can express.

- Fred Brooks marked every page of this dissertation – guiding the thinking that went into the research before and after it occurred. This work is drastically improved due to his years of help and guidance.
- Mary Whitton helped me see that this topic was worth pursuing – both the first time, and several times since. She also gave much helpful feedback on my writing.
- Robert Burton of Brigham Young University encouraged me to go to graduate school since my freshman year and has helped me frequently since I left BYU.
- David Adalsteinsson pushed for more rigor when I analyzed head-tracker logs. Chapter 3 would not exist if not for our many discussions.
- Troy Blackburn taught me a lot of the walking biomechanics described throughout this dissertation.
- Gary Bishop led me through my Forward Walk Model Fourier analysis and helped throughout the research.
- Gordon Pitz helped me design and analyze the two user studies herein. As I told him months ago, he absolutely saved my life once.
- Ming Lin provided me two years of Physically-Based Modeling experience and mathematical tools that helped me a lot with this VE research.
- Greg Welch helped me code and analyze my Kalman filters.
- Bill Baxter showed me the ropes during my first two years of graduate school.
- Eric Burns taught me what makes a dissertation, and showed me what steps to take.
- Jason Jerald discussed perception and statistics when preparing for our oral exams.
- Rich Holloway – friend and boss – helped me see the big picture during our many walks.
- Tabitha Peck let me take over the lab although she also needed to graduate.
- Luv Kohli helped me get shaders into my final user study.
- The EVE Group is always good at finding mistakes in a presentation. The final

presentation is always better for it.

- John Thomas, David Harrison, and Herman Towles helped me several times as I built my WIP system's infrastructure.
- Missy Wood has been a constant friend.
- Tammy Pike made Sitterson Hall's entrance a wonderful place for my first six years in graduate school – her absence leaves a big hole.
- My extended family – especially parents – have been a continual emotional help.
- Without my Savior and comfort from prayer, none of this would have been possible.

This work was also supported by various funding agencies:

- The National Science Foundation, Graduate Research Fellowship Program and Research Fund
- The North Carolina Space Grant program
- NVIDIA's intern program
- Emergent Game Technology's intern program
- Intel Corporation
- The Office of Naval Research's VIRTE program
- U. S. Army Research Office
- NIH National Institute of Biomedical Imaging and Bioengineering
- NIH National Center for Research Resources
- SAIC
- National Science Foundation Grant No. CNS-0751187

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of these funding agencies.

The UNC GUD WIP demo system used an excellent model of UNC's Ackland Art Museum's *Plum, Pine, and Bamboo – Seasonal and Spiritual Paths in Japanese Art* exhibit. I would like to thank 3rdTech for the DeltaSphere<sup>TM</sup>-3000-scanned model. It was produced using 3rdTech's SceneVision<sup>TM</sup>-3D and InnovMetric's PolyWorks<sup>®</sup> software. The model is copyright-protected (2004) by the Ackland Art Museum and 3rdTech, Inc.



# TABLE OF CONTENTS

<b>LIST OF TABLES</b>	<b>xv</b>
<b>LIST OF FIGURES</b>	<b>xvii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Definitions . . . . .	2
1.2 Walking Interface Taxonomy . . . . .	2
1.2.1 Real Walking . . . . .	3
1.2.2 Redirecting Interfaces . . . . .	4
1.2.3 Treadmill Interfaces . . . . .	4
1.2.4 Stationary User Walking Interfaces . . . . .	5
1.3 Comparing Walking Interfaces . . . . .	7
1.4 Thesis Statement . . . . .	7
<b>2 GUD WIP: A Model for Deriving Real-Walking-like Speeds from Walking-In-Place Inputs</b>	<b>11</b>
2.1 Previous WIP Systems . . . . .	11
2.2 Background: Biomechanics of Real Walking . . . . .	15
2.3 The GUD WIP Model . . . . .	18
2.3.1 GUD WIP During Steady State . . . . .	18
2.3.2 How GUD WIP Fulfills LLCM-WIP's Speed-Control Goals . . . . .	19
2.4 UNC GUD WIP Implementation and Performance . . . . .	23
2.4.1 Gait Event Identification . . . . .	24
2.4.2 Gait Cycle Implementation . . . . .	25
2.4.3 Miscellaneous Implementation Details . . . . .	26

2.4.4	Equipment . . . . .	26
2.5	Validation: Consistency . . . . .	27
2.5.1	Method . . . . .	28
2.5.2	Results . . . . .	30
2.5.2.1	Within-Step Consistency . . . . .	31
2.5.3	Discussion . . . . .	33
<b>3</b>	<b>The Forward Walk Model: A Model to Extract Real Walking Parameters from Head Track Data Alone</b>	<b>37</b>
3.1	Introduction . . . . .	37
3.1.1	Model Overview . . . . .	37
3.1.2	Other Applications for Walking Parameters . . . . .	38
3.2	The Forward Walking Model . . . . .	40
3.3	Approximation Methods . . . . .	42
3.3.1	Full-Model-Fitting Method . . . . .	43
3.3.1.1	Implementing the Full-Model-Fitting Method . . . . .	43
3.3.1.2	Full-Model-Fitting Method example . . . . .	48
3.3.1.3	Full-Model-Fitting Method discussion . . . . .	48
3.3.2	Expedited Fitting Method . . . . .	51
3.3.2.1	Implementing the Expedited Method . . . . .	51
3.3.2.2	Expedited Method example . . . . .	52
3.3.2.3	Expedited Method discussion . . . . .	53
3.3.3	Recognizing Walking . . . . .	54
3.4	Comparison With Existing Systems . . . . .	55
3.5	Fourier Techniques . . . . .	55
3.6	Validation: Approximation Methods Provide Accurate Results . . . . .	56
3.6.1	Recap of Method of Real Walking Portion of User Study (Section 2.5)	56
3.6.2	Results . . . . .	57
3.6.3	Data Validity . . . . .	58
3.6.4	Discussion . . . . .	58

<b>4 Usability Study</b>	<b>63</b>
4.1 Method . . . . .	64
4.2 Results . . . . .	71
4.2.1 Per-Trial Measures . . . . .	72
4.2.2 Simulator Sickness . . . . .	78
4.2.3 Presence . . . . .	79
4.2.4 Self-Reported Usability . . . . .	80
4.3 Discussion . . . . .	83
4.3.1 Walk Speed, Accuracy, and Stopping . . . . .	83
4.3.2 Hallways . . . . .	84
4.3.3 Training . . . . .	84
4.3.4 Simulator Sickness . . . . .	84
4.3.5 Direction from shins . . . . .	85
4.3.6 Short Steps . . . . .	85
4.3.7 Average Speed . . . . .	86
4.3.8 Curved Paths . . . . .	87
4.3.9 In-Place-Stepping-to-Speed Conversion . . . . .	88
4.3.10 Forward Walk Model and User Variations . . . . .	88
4.3.11 Step-Frequency-to-Walk-Speed Function Curvature . . . . .	89
<b>5 Conclusion</b>	<b>91</b>
5.1 Future Work . . . . .	93
5.1.1 GUD WIP . . . . .	93
5.1.2 Forward Walk Model . . . . .	95
5.1.3 User Study . . . . .	97
5.1.4 Joystick . . . . .	98
<b>Appendices</b>	<b>99</b>
<b>A GUD WIP Implementation</b>	<b>101</b>
A.1 Overview . . . . .	101

A.2	Filtering Shin Data ( <code>FilteredFoot</code> ) . . . . .	102
A.3	Per-Foot Gait State ( <code>GaitParameterTracker</code> ) . . . . .	104
A.4	Gait Cycle Tracker ( <code>GaitCycleTracker</code> ) . . . . .	105
A.5	LLCM-WIP ( <code>LlcmWip</code> ) . . . . .	106
A.6	Walk Direction ( <code>FilteredTrackerAverager</code> ) . . . . .	107
A.7	Creating Final Velocity ( <code>SpeedCombiner</code> ) . . . . .	108
<b>B</b>	<b>Forward Walking Model Analysis: Levenberg-Marquardt Fitting to Perfect Sinusoids</b>	<b>111</b>
B.1	Introduction . . . . .	111
B.2	Analysis . . . . .	111
B.2.1	Varying Window Size . . . . .	113
B.2.2	Varying Sampling Resolution . . . . .	114
B.2.3	Varying Model Parameters . . . . .	114
B.2.4	Varying Initial Estimates . . . . .	116
<b>C</b>	<b>Forward Walking Model Analysis: Fourier Fitting to Perfect Sinusoids</b>	<b>119</b>
C.1	Introduction . . . . .	119
C.2	Discrete Fourier Principles . . . . .	119
C.2.1	Basis frequencies . . . . .	120
C.2.2	Possible Solutions . . . . .	121
C.3	Increase the Window Size . . . . .	121
C.3.1	Increase the Sample Duration . . . . .	121
C.3.2	Repeat the Input Sample . . . . .	122
C.3.3	Repeat and Mirror the Input Sample . . . . .	122
C.4	Vary the Basis Frequencies . . . . .	123
C.4.1	Method Description . . . . .	123
C.4.2	Results . . . . .	124
C.4.3	Discussion . . . . .	127
C.5	Vary the Basis Frequencies and the Window Size . . . . .	127

C.5.1	Method Description . . . . .	127
C.5.2	Results . . . . .	127
C.5.3	Discussion . . . . .	129
C.6	Conclusion . . . . .	130
<b>D</b>	<b>Usability Study Questionnaire</b>	<b>131</b>
	<b>BIBLIOGRAPHY</b>	<b>139</b>



# LIST OF TABLES

2.1	<b>Quality of fit results:</b> The quality of fit ( $r^2$ ) for the quadratic functions to the average speeds for each subject in each interface (e.g., Figure 2.15). Real walking's and GUD WIP's fits are generally quite good, but LLCM-WIP's fits are rarely as good. . . . .	31
3.1	<b>Range constraints for various walking parameters:</b> $c_z$ is constrained to be within the middle two quartiles of the current data window's z-value range. All non-listed parameters have no constraints. These minimum and maximum possible values were chosen based on pilot data and from what may be found in the literature (summarized in (Inman et al., 1981)). . . . .	45
3.2	<b>Estimated parameters' accuracy:</b> Methods' step frequency, walk speed, and walk direction errors compared with "ground truth" averages at the 50 <sup>th</sup> , 95 <sup>th</sup> , and 100 <sup>th</sup> percentiles. Results are split between perfect walking-recognized trials (100%), and imperfect (<100%). To illustrate, the top row shows that of the 107 perfect walking-recognized trials for the Full-Model-Fitting Method, 53 had error less than 0.6%, 99 had error less than 2%, and the worst had 3.1% error. . . . .	59
4.1	<b>Subjects – Descriptive statistics:</b> The statistics for the subjects in my user study as gathered in the questionnaire. . . . .	70
4.2	<b>Complaint frequency – balance and jerkiness:</b> Number of subjects who complained during free response periods about balance problems and jerky visuals split by LLCM-WIP and combined GUD WIP systems. Statistical significance of each result shown on the bottom line. . . . .	79
4.3	<b>Full questions to abbreviations:</b> Figure 4.11 uses these shortened versions of the longer questions listed here. . . . .	80

4.4 **Complaint frequency – short steps, step-to-speed function, direction:** Number of subjects who complained during free response periods about inability to take short steps, a poor step-to-speed function, and direction problems – split by LLCM-WIP and Dean GUD WIP, and Personalized GUD WIP systems. Each result’s Chi-square test’s statistical significance shown on the bottom line. . . . . 82



# LIST OF FIGURES

1.1	<b>A VE walking interface taxonomy:</b> The boxes provide names for walking interfaces classes. The diamonds present design choices that divide two or more walking interface classes. . . . .	3
1.2	<b>Stationary User walking interface feedback loop:</b> The continual feedback loop experienced by Stationary User walking interface users. . . . .	6
2.1	<b>Previous WIP systems' speeds:</b> Resulting motion (thicker solid line, right y-axis) from (a) the Virtual Treadmill, (b) Redirected WIP, (c) Loco4 WIP, and (d) Yan, 2004's WIP. The WIP systems's speeds are driven by the same in-place steps (dashed line, left y-axis). . . . .	13
2.2	<b>Speeds from LLCM-WIP, GUD WIP, and real walking:</b> Resulting motion (thicker solid line, right y-axis) from (a) LLCM-WIP (Feasel et al., 2008), (b) GUD WIP, and (c) real walking. The two WIP systems are driven by the same in-place steps (dashed line, left y-axis) as Figure 2.1. The real-walking motion is caused by steps taken at approximately the same frequencies as the in-place steps for the WIP systems. Note LLCM-WIP's resulting speed varies considerably. GUD WIP's resulting speed is far more steady – similar to real walking. . . . .	14
2.3	<b>Walking gait cycle:</b> The gait cycle defined for rhythmic walking (Inman et al., 1981). Gait-cycle periods are shown in the inner cycle. The dominant leg's stance and swing phases are shown on the outer arcs. Gait-cycle events are named and illustrated outside the outer arcs (most important feature circled). Gait-cycle period times are generally not equal. . . . .	16
2.4	<b>Step frequency vs. gait-component gait-cycle percent time:</b> Stance, swing, and double-support percent-times vary linearly with step frequency. Data points from (Murray et al., 1964; Dean, 1965; Molen and Rozendal, 1966; Murray et al., 1966; Finley and Cody, 1970; Murray et al., 1970; Molen et al., 1972; Zarrugh et al., 1974; Öberg et al., 1993). . . . .	17

2.5	<b>Step frequency vs. walk speed:</b> As step frequency increases, walk speed increases. Equation 2.2 from (Dean, 1965) (assumed height 1.67m) is fitted to data points from (Murray et al., 1964; Dean, 1965; Molen and Rozendal, 1966; Murray et al., 1966; Finley and Cody, 1970; Murray et al., 1970; Molen et al., 1972; Zarrugh et al., 1974; Öberg et al., 1993). . . . .	17
2.6	<b>In-place stepping gait cycle:</b> Periods, events, phases, and illustrations correspond with those in Figure 2.3. Period times are not generally equal. .	19
2.7	<b>GUD WIP state machine:</b> The current state is maintained until a state-exit criterion (shown on transitions) is fulfilled. . . . .	20
2.8	<b>Starting latency in GUD WIP:</b> Output speed is divided into three sections: <b>Start-up 1:</b> When the first state has begun (Foot Off detected), use the maximum LLCM-WIP-predicted value. <b>Start-up 2:</b> When one or two states have completed, estimate step frequency from Equation 2.7. <b>Steady state:</b> When three or more states have completed, the step frequency is the inverse of the past three states' duration. . . . .	22
2.9	<b>Stopping latency in GUD WIP:</b> Output speed during stopping with three important events marked: First, upon entering a double support state, the new state's duration is estimated (Equation 2.8). Second, when that estimate is exceeded, the output speed decreases to match the step frequency required by the state's actual duration (Equation 2.7). Third, once the step frequency drops below a minimum threshold, all output is stopped. . . . .	23
2.10	<b>A single foot's in-place-step state transition diagram:</b> $P_{\text{vert}}$ is heel height. $V_{\text{vert}}$ is the heel's vertical velocity. The gait-cycle event corresponding to each arc is named next to the arc's exit criterion. The current per-foot gait state (Grounded, Ascending, or Descending) is maintained unless an exit criterion is fulfilled. The dashed transition handles possible input errors and is not a part of steady-state operation. . . . .	24

2.11	<b>The UNC GUD WIP implementation’s state machine:</b> The current state is maintained unless an exit criterion is fulfilled. The dashed transitions are for software robustness on input errors. . . . .	25
2.12	<b>A user in the UNC GUD WIP system:</b> The user wears beacons for 6-DOF trackers on his shins (inset). Beacon-tracking cameras surround the subject on the floor. The heel’s position is approximated by rigid-body transform from the shin’s pose. . . . .	26
2.13	<b>UNC GUD WIP block diagram:</b> The UNC GUD WIP demo system uses four computers: Two specialty computers for the tracker systems, one for GUD WIP, and one for rendering. . . . .	27
2.14	<b>Identifying five steps:</b> Five steps identified for (a) real walking (from the minimum head position during bobbing), and (b) Walking-In-Place (from foot position maxima). The same five-step period is used for both GUD WIP and LLCM-WIP computations. . . . .	29
2.15	<b>Results for Subject 103:</b> Each trial provides a single data point. With four trials at each of four step frequencies, I calculate a quadratic function fit for each of three interfaces. Table 2.1 provides all subjects quality-of-fits.	30
2.16	<b>Between-subject variation by interface:</b> The function fits for (a) real walking, (b) LLCM-WIP, and (c) GUD WIP for all seven subjects from my study. Note the variation of the fits for real walking and LLCM-WIP: Some are nearly linear, some curve upward, some downward. There was no significant within-subject correlation of these curvature changes. . . . .	32
2.17	<b>Within-step consistency by interface:</b> The normalized standard deviation measures within-step consistency: If the normalized standard deviation is low, then the speed was consistent within steps. This shows the mean normalized standard deviation and its standard deviations by virtual walking interface. . . . .	33

3.1	<b>Head-bobbing-caused error hinders accurate walk speed and direction estimation:</b> The instantaneous (a) horizontal walk speed (solid, lighter line) and (b) over-head-view of walk direction (thicker sold lines) compared to those obtained by my method (dashed lines). . . . .	38
3.2	<b>User-centered frame:</b> The body-centric frame (r, f, u) is related to the world frame (x, y, z) by rotating around z by $\theta$ . . . . .	41
3.3	<b>Full-Model-Fit Results:</b> User walked a loop. Thicker, lighter line: head-track data. Thinner, darker line: Full-Model-Fit Method. Background color: walking recognized when white. (a) Overhead view with magnification inset. (b) Horizontal walk speed. (c) Horizontal walk direction. (d) Vertical position. (e) Step frequency. (f) Two-norm error. . . . .	49
3.4	<b>Rightward-bob/angular-velocity error:</b> This simulated, overhead view of a single rightward-bob of walking data (solid line) demonstrates what errors arise when fitting the bobbing-free path to too little data. The dashed line shows the users' true bobbing-free path. However, with only one bob's data, an angular-velocity-including path (dotted line) mistakes the bob for the path. . . . .	50
3.5	<b>Acceleration/forward-bob error:</b> The resulting forward bob amplitude for a user quickly walking a straight line at a constant pace. Note that LMA's resulting forward-bob amplitude increases dramatically during the brief starting and stopping periods. . . . .	50
3.6	<b>Expedited Method results compared Full-Model-Fit Method:</b> User walked a loop. Thicker, lighter line: head-track data. Thinner, darker line: Full-Model-Fit Method. Dashed-line: Expedited Method. Background color: walking recognized when white. (a) Overhead view with magnification inset. (b) Walk speed. (c) Walk Direction. (d) Step Frequency. Note that although the Expedited bobbing-free position is somewhat less accurate than the Full-Model's, the bobbing-free walk directions are nearly identical. . . . .	53

3.7	<b>Walk-identified deciles:</b> Each trial recognized some percentage of the samples on the manually-tagged window (Figure 2.14(a)) as walking. Each method performed perfectly – recognizing all samples as walking – on 107 of the 123 trials (87%). To illustrate, the 20% decile shows that two Full-Model-Fit Method trials recognized between 20-29% of the windows’ samples as walking. The Expedited Method trended toward better scores on the 13% of the trials that imperfectly identified walking. . . . .	57
4.1	<b>Study hallways:</b> The four hallways users experienced – Straight, Obtuse, Acute, and Curved. The hallways were textured to look like mine shafts. All hallways were the same length (50 meters). . . . .	66
4.2	<b>Between-trial resting room:</b> Each trial began and ended in this resting room. This image shows a hallway wall (far right) entering the resting room, the stopping target (on the ground), the scoreboard (center wall), and the door (darkened patch on the left wall). . . . .	66
4.3	<b>The stopping target:</b> Subjects had to be within the white circle to end a trial. Stopping distance was measured as the horizontal distance from the center of the black dot to subject’s eyes’ location. . . . .	67
4.4	<b>Exclusion criterion for direction calibration error:</b> The average total time and collision time for each subject in Straight Hallway trials. X’s indicate subjects determined as outliers whose data was replaced by data from a new subject experiencing the same WIP system after the direction calibration error was corrected. . . . .	70
4.5	<b>Main effects by WIP system:</b> The not-significant means and standard errors among WIP systems as measured by Total Time, Collision Time, Stopping Error, and Normalized Total Time. . . . .	73

4.6 **Training effects:** The means and standard errors for Total Time, Collision Time, Stopping Error, and Normalized Total Time by set with statistically-significant fits for all WIP interfaces. Total Time, Collision Time, and Normalized Total Time show statistically significant training effects. Stopping Error shows a trend. . . . . 75

4.7 **Effect of hallway type:** The means and standard errors for Total Time, Collision Time, Stopping Error, and Normalized Total Time by hallway for all WIP interfaces. Total Time, Collision Time, and Normalized Total Time have a main effect by hallway and are shown with their statistically significant fits. Stopping Error has none. . . . . 77

4.8 **Total Time interaction – training and hallway:** Interaction between training effect (X axis) and hallway effect (different lines) for Total Time. Mean and standard error shown for each datapoint. The statistically significant interaction is almost certainly explained by the cross-over interaction for the Straight Hallway with the Curved and Obtuse Hallways between the first and second sets. . . . . 78

4.9 **Collision Time interaction – training and hallway:** Interaction between set number (X axis) and hallway (different lines) for Collision Time. Mean and standard error shown for each datapoint. The statistically significant interaction probably refers to the difference in when different hallways have derivative changes. . . . . 78

4.10 **Simulator sickness by WIP system:** The means (thick horizontal line) and 95% confidence intervals for the mean (boxes) of LLCM-WIP and combined GUD WIP systems. This difference is statistically significant: LLCM-WIP can cause worse nausea than GUD WIP. Kevin Arthur reports that five previous studies had mean nausea scores between 7 and 27 (Arthur, 2000). 79

4.11	<b>Questionnaire responses:</b> Box plots of all subjects' WIP judgements questionnaire responses (Section II). LLCM-WIP scores are the white boxes, Dean GUD WIP are the medium gray boxes, and Personalized GUD WIP are the darker boxes. Statistical significance is shown below each grouping – significant results are highlighted. . . . .	81
4.12	<b>Instantaneous speed comparison:</b> Given the same inputs over a period of eight seconds, this subject (if experiencing LLCM-WIP) would experience his speed change from around 0.5m/s to 3m/s multiple times per second. A GUD WIP subject would experience a relatively consistent output speed. .	86
4.13	<b>Curved hallway results:</b> The not-statistically-significant differences for subjects on the Curved Hallways by system (means and standard errors shown).	87
5.1	<b>The walking phases:</b> Development, rhythmic, and decay phases shown against a smoothed plot of walking speed for a straight-line path. This dissertation focuses mainly on improving Walking-In-Place systems for rhythmic speeds. Little is known about the characteristics of development and decay.	92
A.1	<b>UNC GUD WIP system overview:</b> This diagram illustrates the data flow through UNC GUD WIP. Each box represents a class that processes its input data and outputs data for later classes. . . . .	101
A.2	<b>The FilteredFoot class:</b> The VRPN-input shin position is translated to the foot position, and each component is filtered. The VRPN-input quaternion rotation is transformed to Euler angles, filtered, and transformed back to quaternions. . . . .	103
A.3	<b>The GaitParameterTracker class:</b> The state transition diagram for a single foot as it takes in-place steps as implemented by the <code>GaitParameterTracker</code> class. The current per-foot gait state (Grounded, Ascending, or Descending) is maintained unless an exit criterion (on transitions) is fulfilled. The dashed transition handles possible input errors: It is not a part of steady-state operation. . . . .	104

A.4	<b>The GaitCycleTracker class:</b> The fundamental GUD WIP state machine. The current state is maintained unless an exit criterion is fulfilled. The dashed transitions are for software robustness on input errors. . . . .	105
A.5	<b>The LlcMwip class:</b> Implements Feasel’s LLCM-WIP algorithm (Feasel et al., 2008). . . . .	107
A.6	<b>The FilteredTrackerAverager class:</b> This class averages the position and orientation of the user. UNC GUD WIP only uses the average horizontal rotation to estimate walk direction. . . . .	108
A.7	<b>The SpeedCombinerClass class:</b> This class receives speed and direction inputs and determines the final GUD WIP velocity. . . . .	109
B.1	<b>lmfit results:</b> Given input data (points), <code>lmfit</code> approximates model parameters for the best fit (sinusoidal curve). Input data and resulting model fit have amplitude=0.3 m, frequency=2 Hz, phase= $\pi$ radians, and offset=2 m.	112
B.2	<b>Varying window size:</b> For all window sizes, convergence was excellent: Even before it converges at approximately 1.0 seconds, the errors are very small. Window size varies (0.25...3.5 sec). Resolution: 256 samples per window. Parameters: amplitude = 0.03 m, frequency = 1.0 Hz, phase = 1.0 radians, and offset = 1.8 m. Initial estimates: 10% error added to all parameters. . . . .	113
B.3	<b>Varying sampling resolution:</b> A log-log plot. For all resolutions, convergence was excellent. The fit-line’s slope (-0.507) indicates approximate square-root convergence with increased resolution. Window size: 0.5 sec. Resolution varies (16, 32, ... $2^{18}$ ). Parameters: amplitude = 0.03 m, frequency = 1.5 Hz, phase = 1.0 radians, offset = 1.8 m. Initial estimates: 30% error added to all parameters. . . . .	114



B.4	<b>Varying parameter values:</b> Varying (a) amplitude, (b) frequency, (c) phase, and (d) offset. All results are better than required accuracy ( $10^{-6}$ would likely be sufficient). Window size: 0.5 sec. Resolution: 256 samples. Parameters (when not the varying parameter): amplitude = 0.03 m, frequency = 1.5 Hz, phase = 1.0 Hz, offset = 1.8 m. Initial estimates: 30% error added to all parameters. . . . .	115
B.5	<b>Varying initial estimates:</b> Varying (a) all parameters, (b) frequency only, and (c) all parameters except frequency. Two-norm errors at zero converged to the global minimum. All others converged to an incorrect local minimum. Window size 0.5 s. Resolution: 256 samples. Parameters: amplitude = 0.03 m, frequency = 1.5 Hz, phase = 1.0 radians, offset = 1.8 m. Initial estimates: When not varying, 0% added. . . . .	117
B.6	<b>Failed fit:</b> When the frequency was set to $\sim 1.5$ times the actual input frequency, <code>lmfit</code> went to a local minimum fit with frequency approximately twice that of the input data. Note that the phase that resulted places the fit's maximum and minimum approximately in line with the input data's. . . . .	118
C.1	<b>Repeating the input signal:</b> This causes discontinuities (at arrows) in the input function that show up as high-frequency components, and obscures the original function's phase. . . . .	122
C.2	<b>Repeating and mirroring the input signal:</b> Although this avoids function discontinuities, it still causes first-derivative discontinuities (at arrows) in the input data that show up as high-frequency components, and obscures the original function's phase. . . . .	123
C.3	<b>Vary basis frequency – magnitude and phase:</b> An absolute value sinc function and linearly varying phase. This example shows the result of testing 512 equally-spaced frequencies between 0.5 and 5.5 Hz with Equation C.7. The magnitude peak is highest near 3 Hz, with approximately a correct magnitude, and phase. . . . .	124

C.4	<b>Maximum magnitude frequency with varying window size:</b> The error's shape repeats but decreases in size with increased window size. It converges to the correct value. . . . .	126
C.5	<b>Phase affects maximum magnitude frequency:</b> The maximum-magnitude frequency for a 2.0 Hz sine with varying phase (x-axis) and a fixed window size (1.0 cycle). . . . .	126
C.6	<b>Varying basis frequency and window size method – magnitude and phase:</b> Input is a fixed sine curve (frequency = 1.75 Hz, amplitude = 1.0, phase = 1.22 radians, window size = 3.0 sec). The maximum magnitude is highest near 1.75 Hz, with approximately a correct amplitude and phase. . .	128
C.7	<b>Input resolution changes method's accuracy:</b> The maximum-magnitude frequency for a 2.0 Hz signal with increasing spatial resolution. . . . .	129
C.8	<b>Phase affects maximum-magnitude frequency:</b> Input is a single sine (frequency = 2.0 Hz, varying phase (0. . . $\pi$ ) radians, window size = 3.0 sec). . . . .	129

## CHAPTER 1

# Introduction

Virtual Environment (VE) walking interfaces are a fundamental – and open – problem. Real Walking – the preferred solution – is often infeasible due to tracked space constraints: Virtual worlds are often larger than tracked space. Walking-In-Place (WIP) is a walking interface that eliminates space constraints: The user’s in-place steps move his virtual viewpoint through the virtual world. Unfortunately, studies have demonstrated Real Walking induces higher presence and produces very different motions than previous WIP systems (Usoh et al., 1999; Whitton et al., 2005).

This dissertation presents two real-walking models that improve WIP systems’ generated speeds. The first model describes a biomechanics-based way to generate virtual walk speeds from in-place stepping gestures. The second model enables extracting personal step-frequency-to-walk-speed functions from head-track data alone. This dissertation validates the following thesis: *Walking-In-Place systems are measurably improved by using real-walking-based models.*

**Organization:** This chapter defines important VE terms, locates WIP systems among VE walking interfaces, and outlines the remainder of the dissertation. The remaining chapters describe the real-walking models (Chapter 2: GUD WIP, and Chapter 3: Forward Walk Model), and a usability study of the model-based WIP systems (Chapter 4). The final chapter (Chapter 5) concludes this work and presents avenues for future research. Four appendices provide further details about the UNC GUD WIP implementation, a convergence analysis of Levenberg-Marquardt on real-sinusoidal data, a description of alternate approaches to Forward Walk Model fitting, and the questionnaire used in the final user study.

## 1.1 Definitions

A person’s **environment** is his surroundings, as perceived by his senses (Gibson, 1986). A **virtual environment** (VE) is an illusion of an environment produced by synthetic sensory stimuli, perhaps perceived in conjunction with real stimuli for other senses. A **VE system** is a system for generating a coordinated set of synthetic stimuli for some senses.

VE systems are usually built to enable some task – training, visualization, exploration, or therapy. To complete the VE-system task, **interfaces** – techniques enabling computer-human interactions – must be provided for all task-required operations.

Among the most common interfaces is viewpoint motion – VE locomotion interfaces being among the most common. Exercise Science defines **locomotion** as “any rhythmic activity that produces translatory motion of the [center of mass (COM)]. This could include walking, running, swimming, flying, crawling, etc.” (Blackburn, 2008). I modify this for VEs as any input activity (including non-rhythmic) that produces rhythmic virtual viewpoint motion – where rhythmic motion can include some accelerations. The produced motion includes both speed and direction. This dissertation focuses solely on output speeds from walking interfaces.

In VE literature, **maneuvering** is non-rhythmic motion in a limited space: “Maneuvering tasks take place in a local area and involve small, precise movements. The most common use of maneuvering is to position the viewpoint more precisely within a limited local area to perform a specific task” (Bowman et al., 2005).

In both maneuvering and walking, walking interfaces control the viewpoint’s **pose** (position and orientation).

## 1.2 Walking Interface Taxonomy

Figure 1.1 shows a taxonomy of VE walking interfaces. This section briefly discusses each interface in this taxonomy. For a more in-depth discussion of VE walking interfaces, see Whitton and Razzaque’s VE locomotion chapter (Whitton and Razzaque, 2008a).

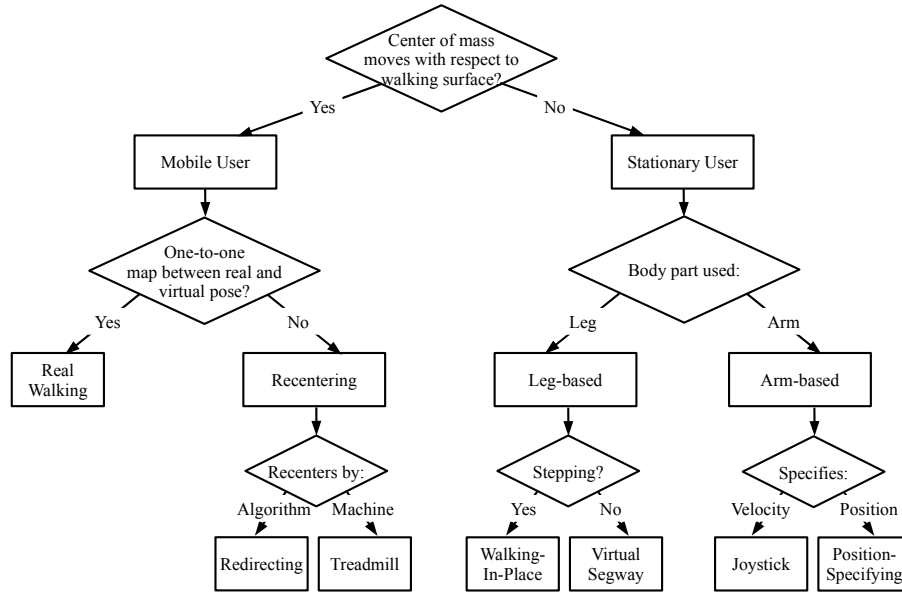


Figure 1.1: **A VE walking interface taxonomy**: The boxes provide names for walking interface classes. The diamonds present design choices that divide two or more walking interface classes.

### 1.2.1 Real Walking

Walking is an expert, second-nature skill by late childhood (Rose and Gamble, 2006). The VE Real-Walking interface provides the same kinesthetic, vestibular, acoustic, and tactile stimuli as real-world walking. In this dissertation, capitalized “Real Walking” always refers to this VE interface. It is the gold standard VE walking interface, used by many VE systems (Usuh et al., 1999; Meehan et al., 2002; Whitton et al., 2005; Ruddle and Lessels, 2009).

However, Real-Walking usage is limited by the following factors:

- Real Walking requires tracking the user through all walkable space. **Space constraints** usually limit Real-Walking-driven VEs to a single real-world room.
- High quality, large area tracking is **costly**.
- Since virtual visual feedback does not match the real-world space, user **safety** must be considered: Users can unwittingly walk into real-world walls, chairs, or other hazards.

The remaining walking interfaces trade Real-Walking quality for decreased space constraints. In all cases, the Real Walking interface’s simplicity is replaced by a more complicated-to-develop system.

### 1.2.2 Redirecting Interfaces

Redirecting Interfaces alter the mapping between real-world and virtual-world position to increase the reachable virtual space. Two proposed techniques enable users to really walk through a larger virtual space than the real-world tracked space:

- **Redirected Walking** turns the world at an imperceptible, but different rate as the user walks and turns his head (Razzaque et al., 2002; Nitzsche et al., 2004; Razzaque, 2005; Peck et al., 2008; Suma et al., 2009; Peck et al., 2010).
- **Scaled Translational Gain** scales the forward walk speed by a fixed amount (Williams et al., 2006; Interrante et al., 2007b).

Although each technique enlarges the reachable virtual space, in all cases, the user can reach a place where the real world doesn't permit forward walking although the virtual-world does: An orientation resetting mechanism must be employed (Williams et al., 2007; Peck et al., 2008).

**Summary:** Redirecting interfaces obtain larger reachable space at increased development cost – developing such systems is considerably harder than Real-Walking systems. They still require a large tracked space. User safety is still a concern.

### 1.2.3 Treadmill Interfaces

Treadmill Interfaces trade the freedom of walking in an open real-world space for increased reachable virtual space. They place a user on a moving surface which recenters the user continually while walking. These fall into one of four categories:

- **Passive, One-Dimensional Treadmills** recenter the user through his own motions and require a steering mechanism for two-dimensional motion (Brooks et al., 1992).
- **Active, One-Dimensional Treadmills** track the user's location and recenter the user automatically. These also require a steering mechanism (Feasel et al., 2010).
- **Passive, Two-Dimensional Treadmills** enable the user to walk in any direction, recentering the user through his motions (e.g., the Virtosphere (Virtosphere, 2008)).
- **Active, Two-Dimensional Treadmills** enable the user to walk in any direction and automatically recenter the user (e.g., the Omni-Directional Treadmill (Darken

et al., 1997), the Torus Treadmill (Iwata, 1999), and the CirculaFloor (Iwata et al., 2005)).

In each, the user may freely travel through virtual worlds without leaving a small real-world tracked space. However, each adds problems: Passive, One-Dimensional Treadmills require significant effort to push the belt under the user. Both one-dimensional systems require some non-natural turning metaphor. Both two-dimensional treadmill systems allow normal turning, but decrease safety when the user turns while walking: The walking surface's inertia continues to recenter in the previous direction, often making the user fall. **Summary:** As do Redirected walking interfaces, Treadmill interfaces recenter the user to get unlimited walking space. The treadmill options cost varying amounts – and require varying software development – but the head-tracking space is nicely decreased. User safety concerns have changed: Instead of worrying that the user will walk into real-world objects, the user could step off of the walking surface or fall on the walking surface: Many Treadmill systems employ a safety harness.

#### 1.2.4 Stationary User Walking Interfaces

All remaining walking interfaces – Walking-In-Place (WIP), Virtual Segway, Joystick, and Position Specifying – are very similar: To varying degrees, they trade similarity to Real Walking for increased reachable space. Of these, WIP is the most similar to Real Walking, but also the most complex.

All of these interfaces follow a similar pattern to create viewpoint motion (Figure 1.2). In all, the user perceives his virtual position, forms intent to move, and communicates that intent through some system-measured gesture. The system identifies the gesture, creates a motion vector, integrates that vector, and presents the user with his updated virtual location. The defining difference among these interfaces is the gesture that communicates the user's intent to the system:

- In **Walking-In-Place** systems, the user takes in-place steps (Slater and Usoh, 1994; Templeman et al., 1999; Yan et al., 2004; Feasel et al., 2008).

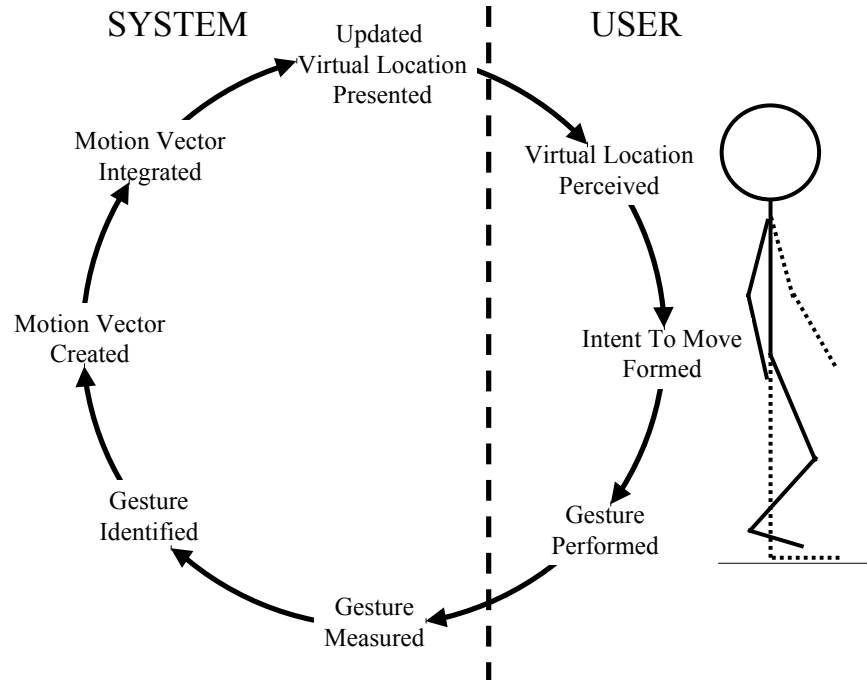


Figure 1.2: **Stationary User walking interface feedback loop:** The continual feedback loop experienced by Stationary User walking interface users.

- In **Virtual Segway** systems, the user leans or holds his legs in the direction he wishes to move (Bouguila et al., 2002).
- In **Joystick** systems, the user gestures with his hand – usually by pushing a directional to specify the direction and speed of motion (Usuh et al., 1999; Whitton et al., 2005; Templeman et al., 2006).
- In **Position Specifying** systems, the user moves a representation of his viewpoint within a map-like version of the environment (Brooks, 1987; Stoakley et al., 1995).

Section 2.1 provides more information about previous WIP systems’ output speed characteristics. For further information on various components of previous WIP systems, see Whitton and Razaque’s case study (Whitton and Razaque, 2008b).

Unlike in all other systems, the user is approximately stationary with respect to the real world. However, in all cases, much of walking’s richness is lost: The user uses less of his life-long walking experience in and derives fewer walking real perceptions from the walking interface.

**Summary:** Stationary User walking interfaces trade similarity to Real Walking for in-



creased reachable space. Excepting WIP interfaces, Stationary User interfaces' gestures have no resemblance to walking's gestures. These interfaces generally cost the least of all walking interfaces. Since they remain stationary, users are physically safest.

### 1.3 Comparing Walking Interfaces

Two walking interface studies have compared Real Walking and WIP interfaces.

Usoh replicated and improved upon a presence-measuring comparison study introduced by Slater (Usoh et al., 1999; Slater et al., 1995b). In Usoh's study, subjects employed one of a Real Walking, a WIP, or a Joystick-like (JS) interface to travel through a virtual world. After a brief training session, the subjects entered a room with a visible drop. The authors found that subjects' presence was highest with Real Walking, then WIP, then Joystick.

Whitton provided subjects three different walking interfaces: Real Walking, WIP, and JS (Whitton et al., 2005). The subjects executed straight-line paths to determined stopping locations with their virtual head positions logged. The authors found that the resulting paths' characteristics were very different from Real Walking for WIP and JS. Although they were unable to quantify how differing from Real Walking's characteristics affect a walking interface, the authors ended the paper warning system developers: "Carefully consider whether there might be unintended consequences of adopting Walking-In-Place or Joystick interfaces."

### 1.4 Thesis Statement

Usoh's and Whitton's comparison studies show that users respond worse to WIP systems than to Real Walking: WIP system users experience lower presence and significantly different paths. However, due to their significantly lower space requirements, WIP systems can often be used where Real Walking is infeasible. Therefore, improving WIP systems is an important goal.

Figure 1.2 provides an explanation for why WIP systems compare poorly to Real Walking. The user plans a virtual motion and steps accordingly. If the resulting virtual mo-

tion doesn't match the intended motion, a user-system miscommunication has occurred. User-system miscommunications distract users. In WIP systems, distraction is particularly troublesome for two reasons. First, walking's gestures and output movements are fundamentally related: If one takes a series of steps, one has learned to expect a resulting path with specific speed characteristics. Second, walking is not a primary goal of most VE-based applications: Training, data analysis, or therapy are common uses which require walking. When the walking interface distracts the user from the goal, the application loses effectiveness. To avoid distracting from the task at hand, the WIP system must interpret the user's input gestures and generate motion consistent with the user's expectations.

I expect that the closer any VE walking interface approaches walking in all characteristics, the better it will compare to Real Walking. Figure 1.2 indicates where these improvements can be made for Walking-In-Place systems. The in-place stepping gesture is already as close to walking's gesture as can be performed while remaining stationary. However, the way the system identifies the gesture and the way the system creates a motion vector can each be improved: If the WIP system employs real-walking-based models to identify steps and create the motion, the WIP motion would be more like real walking's motion. Even though motions include both speed and direction, this dissertation focuses on WIP-derived speeds only.

This dissertation demonstrates the following thesis: *Walking-In-Place systems are measurably improved by using real-walking-based models.*

First, two real-walking-based models are presented. Both focus solely on improving WIP system's speed characteristics. Chapter 2 presents a model (GUD WIP) for creating real-walking-based speeds from in-place-stepping gestures. With GUD WIP, I assert that step frequency is a sufficient metric for deriving intended walking speeds. Chapter 3 presents a model (the Forward Walking Model) for extracting step frequency, bobbing-free walk speed, and other parameters from head track data alone. The Forward Walk Model is used to measure and initialize WIP systems to a personalized step-frequency-to-walk-speed function. I assert that using this personalized function improves WIP systems. Each of these models is presented along with example implementations' details. GUD WIP is accompanied

by a validation that demonstrates that when really walking, step frequency and speed are consistently correlated, and that, on this metric, my GUD WIP implementation is more consistent than previous WIP systems. The Forward Walking Model is accompanied by a validation of the implementations' proper functioning. Although the dissertation may be fully understood without reading them, Appendices A, B, and C provide further details on the GUD WIP implementation, the approximation methods' characteristics for the Forward Walk Model, and an alternate approximation method attempted.

Chapter 4 presents a usability study which evaluates the thesis. In it, I compare LLCM-WIP – the most advanced WIP-speed system previous to the work herein – to two versions of my WIP system. The results of this study demonstrate that employing both models presented here improves WIP systems on the following metrics:

- users' rating output speeds as accurate
- users' rating curved paths as easy
- lower nausea levels
- user satisfaction with virtual speeds created from in place steps

Appendix D provides the questionnaire subjects answered during this user study.

Chapter 5 states various conclusions from this work, as well as introducing several avenues for future research.



## CHAPTER 2

# GUD WIP: A Model for Deriving Real-Walking-like Speeds from Walking-In-Place Inputs

This chapter presents a WIP model for deriving WIP speeds that match real walking better than earlier systems – leading to improved viewpoint movement and optical flow.

## 2.1 Previous WIP Systems

Many very good WIP systems have been proposed. Although each focused on a different system component, each had to sample in-place steps, and derive intended forward speeds.

Each of the following descriptions follows the same pattern:

- **System name:** author-given name, if any
- **System purpose:** author-given purpose for the system
- **Gesture recognition:** how in-place steps are recognized
- **Virtual motion characteristics:** magnitude, shape, and latency
- **Citations**

The motion characteristics for most of these systems are summarized in Figures 2.1 and 2.2.

**The Virtual Treadmill:** Slater, Usoh, and Steed’s *Virtual Treadmill* introduced Walking-In-Place systems. It recognizes in-place stepping motions via a neural net trained to in-place stepping head-motions. The user’s virtual viewpoint moves at a fixed “natural walking

speed” after between a half and a full step’s latency.<sup>1</sup> As shown in Figure 2.1(a), if the user steps frequently enough, forward motion is continuous across steps (Slater and Usoh, 1994; Slater et al., 1995a; Slater et al., 1995b; Slater, 2007; Steed, 2008).

**Gaiter:** Templeman, Denbrook, and Sibert’s *Gaiter* enables specifying non-forward stepping directions and both real and in-place steps. It recognizes in-place steps by tracking knee motions. It identifies an in-place step when the user’s knee reaches full extension and begins to return to its original position. The virtual viewpoint motion follows the forward motion of the knee as soon as the in-place step is recognized – approximately a half-step’s latency. *Gaiter*’s precise speed characteristics are unavailable, and so are not shown in Figure 2.1 or 2.2 (Templeman et al., 1999; Templeman, 2003; Templeman et al., 2006; Templeman, 2008).

**Redirected WIP:** Razzaque’s *Redirected WIP* keeps users facing forward in a three-walled cave, so that they do not see the open wall behind them however they turn in the VE. It recognizes in-place steps via a modified version of the Virtual Treadmill’s neural net trained on the user’s hips’ motions. The user’s virtual viewpoint follows a “sawtooth” motion pattern (Figure 2.1(b)). Other virtual motion characteristics are similar to the Virtual Treadmill (Razzaque et al., 2002; Razzaque, 2007).

**Loco4 WIP:** Feasel’s *Loco4 WIP* was created as a putatively better interface than JS and tested in an experiment comparing Real Walking, WIP and JS interfaces. It recognizes in-place steps through the foot’s impact – sampled via head-based accelerometer data. The virtual viewpoint moves a complete step as soon as a step is recognized: a delta function after a one-step latency (Figure 2.1(c)) (Whitton et al., 2005; Feasel, 2007).

**Yan, 2004:** Yan, Allison, and Rushton’s WIP system had lower starting latency than previous WIP systems. It recognizes in-place steps based on the knee’s vertical displacement – measured by knee-mounted position trackers. The virtual viewpoint moves at a speed determined by the knee’s maximum vertical speed early during an in-place step. This yields a latency of less than a quarter step. The speed is smoothed by Kalman filter and can result in between-step continuous motion – as shown in the latter part of Figure 2.1(d)

---

<sup>1</sup>Slater’s and Steed’s memories differ on this point.

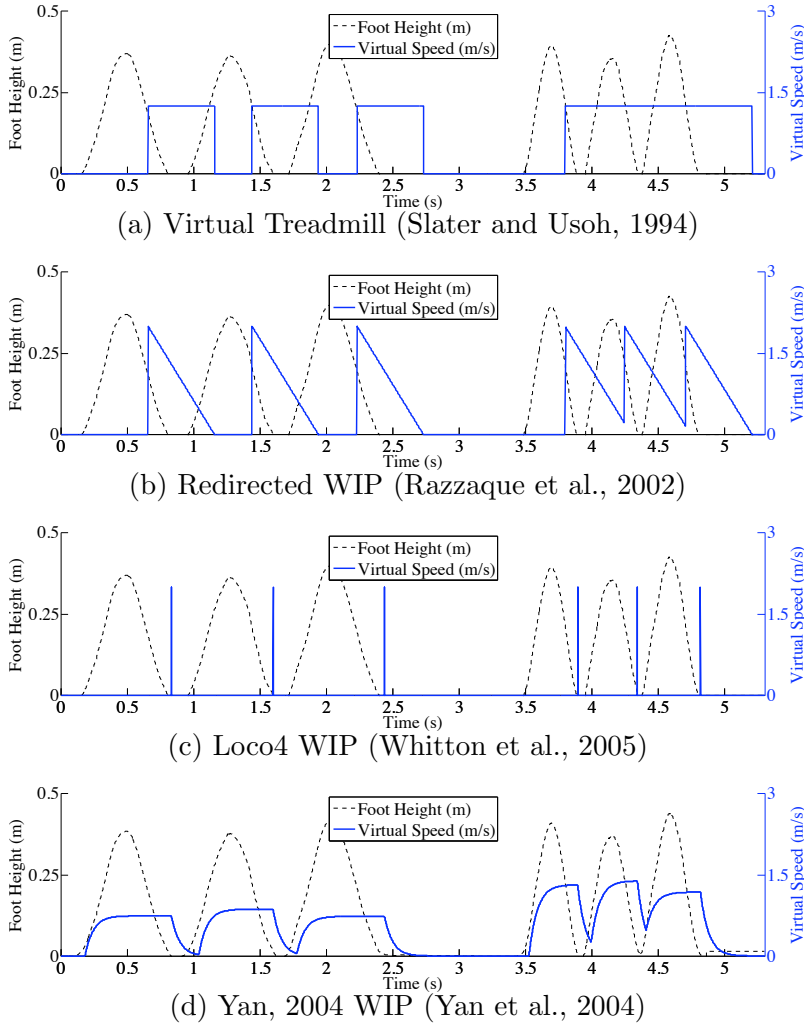


Figure 2.1: **Previous WIP systems’ speeds:** Resulting motion (thicker solid line, right y-axis) from (a) the Virtual Treadmill, (b) Redirected WIP, (c) Loco4 WIP, and (d) Yan, 2004’s WIP. The WIP systems’s speeds are driven by the same in-place steps (dashed line, left y-axis).

(Yan et al., 2004; Allison, 2008).

**LLCM-WIP:** Feasel’s *LLCM-WIP* improved on several WIP-speed characteristics. It is distinct from all others described as it does not explicitly recognize steps: It bases virtual motion on the absolute value of the vertical heel speed. The virtual viewpoint moves at this speed after being smoothed, offset, and scaled. This produces the undulating forward speed shown in Figure 2.2(a). The various operations yield approximately 140 ms latency<sup>2</sup>

<sup>2</sup>Figure 2.2 does not show this because my simulator’s filter was different from Feasel’s.

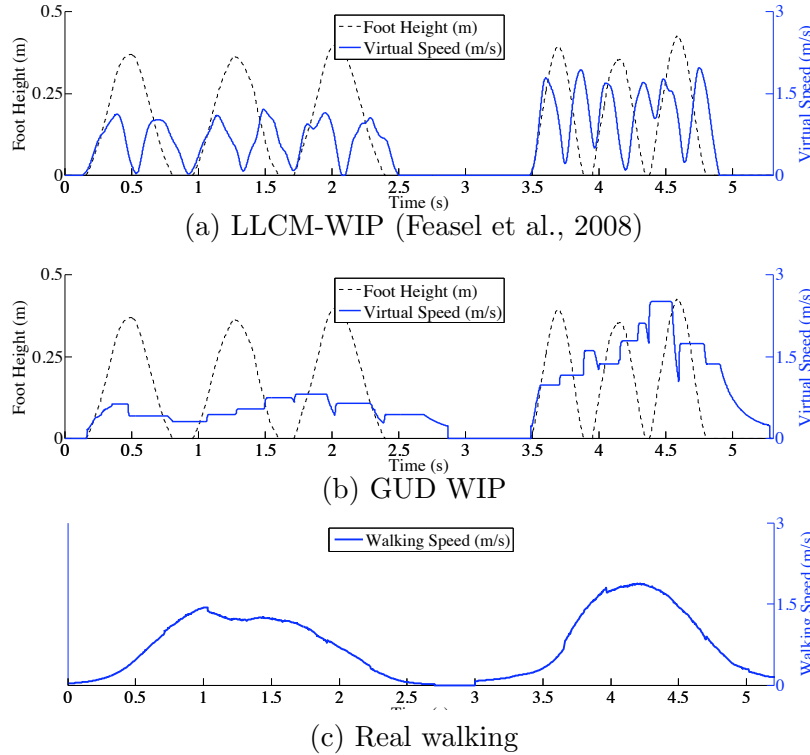


Figure 2.2: **Speeds from LLCM-WIP, GUD WIP, and real walking:** Resulting motion (thicker solid line, right y-axis) from (a) LLCM-WIP (Feasel et al., 2008), (b) GUD WIP, and (c) real walking. The two WIP systems are driven by the same in-place steps (dashed line, left y-axis) as Figure 2.1. The real-walking motion is caused by steps taken at approximately the same frequencies as the in-place steps for the WIP systems. Note LLCM-WIP’s resulting speed varies considerably. GUD WIP’s resulting speed is far more steady – similar to real walking.

– approximately a quarter step, similar to Yan’s system (Feasel et al., 2008).

As only LLCM-WIP explicitly focused on output speed, I compare my results (Figure 2.2(b)) only with LLCM-WIP, and real walking speeds (Figure 2.2(c))

Because my WIP model uses the biomechanics of human gait to “understand” stepping gestures, I call it *Gait-Understanding-Driven Walking-In-Place*, or *GUD WIP*.<sup>3</sup> The GUD WIP model has the following features:

- all LLCM-WIP-proposed speed-control goals satisfied:
  - smooth between-step locomotion
  - continuous within-step speed control
  - real-world maneuvering

---

<sup>3</sup>The pun is intentional: I firmly believe someone will develop a Better WIP.



- low starting latency
- low stopping latency
- a biomechanics-inspired state machine that measures step frequency at multiple points during in-place steps
- a biomechanics-inspired technique for estimating step frequency when only a fraction of a step has been completed
- a technique for identifying in-place step events
- an easily-modifiable technique for deriving gait-understanding-driven output speeds from step frequencies
- parameters for average users and a manual method for calibrating the system for any specific user

This chapter presents the GUD WIP model, describes portions of the UNC implementation, and presents a validation study that tests how consistently real walking, LLCM-WIP, and GUD WIP respond to different in-place step frequencies.

## 2.2 Background: Biomechanics of Real Walking

The *gait cycle* describes the pattern evident in non-pathological human gait (Inman et al., 1981). When altering walking speed, humans alter their gait cycle characteristics in well-known ways. By applying gait-cycle principles, virtual walking systems can avoid many user-system miscommunications.

The walking gait cycle (Figure 2.3) is conventionally defined as a repeating series of periods:

- Initial Double Support
- Non-dominant-leg Initial Swing
- Non-dominant-leg Mid Swing
- Non-dominant-leg Terminal Swing
- Second Double Support
- Dominant-leg Initial Swing
- Dominant-leg Mid Swing

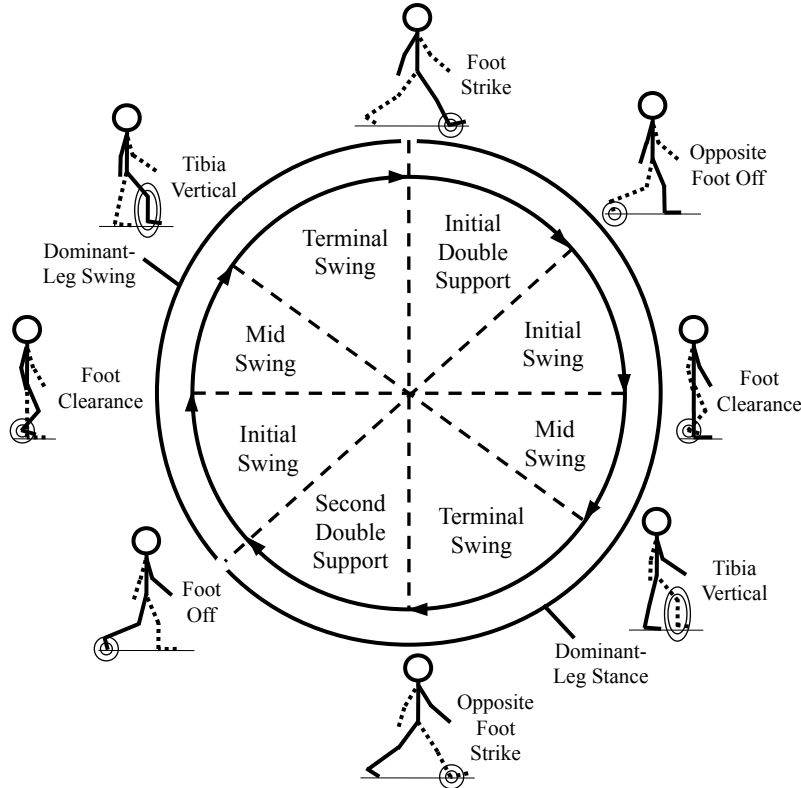


Figure 2.3: **Walking gait cycle:** The gait cycle defined for rhythmic walking (Inman et al., 1981). Gait-cycle periods are shown in the inner cycle. The dominant leg’s stance and swing phases are shown on the outer arcs. Gait-cycle events are named and illustrated outside the outer arcs (most important feature circled). Gait-cycle period times are generally not equal.

- Dominant-leg Terminal Swing

These periods are separated by gait cycle events. The periods combine into the *Stance* and *Swing* phases (Figure 2.3, outermost arcs).

The percentage of the gait cycle spent in these periods and phases varies linearly with step frequency (twice gait-cycle frequency, Figure 2.4). For instance, as step frequency increases, the stance phase’s gait-cycle percent-time decreases.

Step frequency ( $f$ ) and step length ( $l$ ) produce walk speed ( $|v|$ ):

$$|v| = f \cdot l \quad (2.1)$$

Biomechanics literature indicates that step frequency and step length are linearly cor-

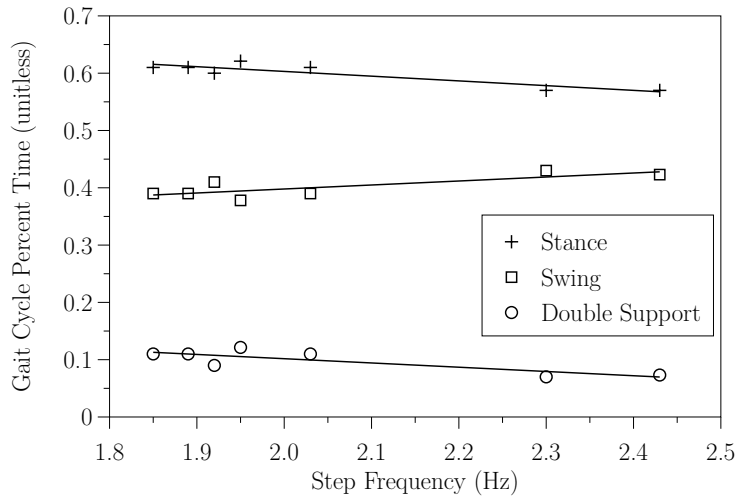


Figure 2.4: **Step frequency vs. gait-component gait-cycle percent time:** Stance, swing, and double-support percent-times vary linearly with step frequency. Data points from (Murray et al., 1964; Dean, 1965; Molen and Rozendal, 1966; Murray et al., 1966; Finley and Cody, 1970; Murray et al., 1970; Molen et al., 1972; Zarrugh et al., 1974; Öberg et al., 1993).

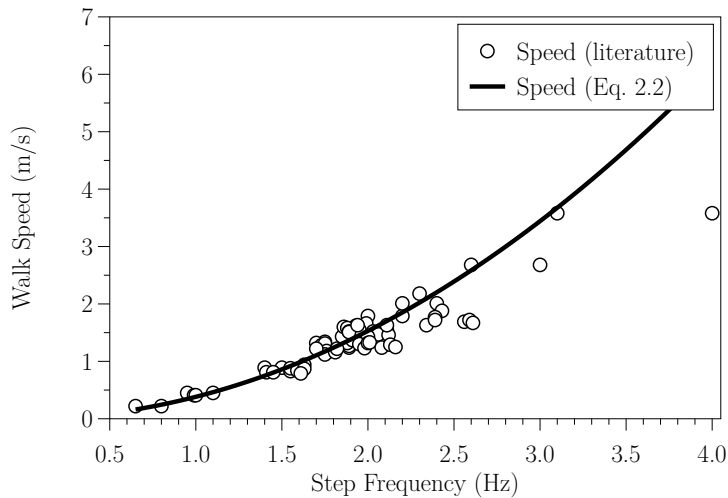


Figure 2.5: **Step frequency vs. walk speed:** As step frequency increases, walk speed increases. Equation 2.2 from (Dean, 1965) (assumed height 1.67m) is fitted to data points from (Murray et al., 1964; Dean, 1965; Molen and Rozendal, 1966; Murray et al., 1966; Finley and Cody, 1970; Murray et al., 1970; Molen et al., 1972; Zarrugh et al., 1974; Öberg et al., 1993).

related at normal walk speeds (Inman et al., 1981). Furthermore, step length positively correlates with height. By using these relationships, walk speed can be estimated from either step frequency or stride length. The relationship can be measured for any individual. From his and others' data, Dean proposed the following relationship between walk speed ( $|v|$ ), step frequency ( $f$ ), and subject height ( $h$ ) (Dean, 1965):

$$|v| = \left( \frac{f}{0.157} \times \frac{h}{1.72} \right)^2 \quad (2.2)$$

Data from several other biomechanics studies support this equation (Figure 2.5).

By applying these gait principles – gait cycle events, periods, and phases, their variation with step frequency, and step frequency's relationship to walk speed – and by using the user's height and step frequency, GUD WIP produces near-real-walking output speeds.

## 2.3 The GUD WIP Model

### 2.3.1 GUD WIP During Steady State

Modifying the walking gait cycle for Walking-In-Place is straightforward (Figure 2.6): The only significant difference is that the three walking swing periods (initial, mid, and terminal swing) are merged into two Walking-In-Place swing periods (initial and terminal). The events separating the three walking swing periods (foot clearance and tibia vertical) do not exist for Walking-In-Place; these are replaced by a single event: maximum foot height.

This Walking-In-Place gait cycle can be represented by a simple state machine (Figure 2.7): a state for each gait-cycle period, an edge for each gait-cycle event, a new no-gait state for extended periods of no input, and appropriate transitions for starting and stopping gait. At all times that inputs do not fulfill a gait-state exit criterion, this state diagram remains in its current state. Self-loops are omitted to simplify the figure.

The state-transition diagram indicates a direct way to measure rhythmic step frequency: After the user completes three successive gait states, the time required to complete those three states is a full step's time. This step frequency measurement persists during the following gait state. When any state finishes, the measurement is updated to include the

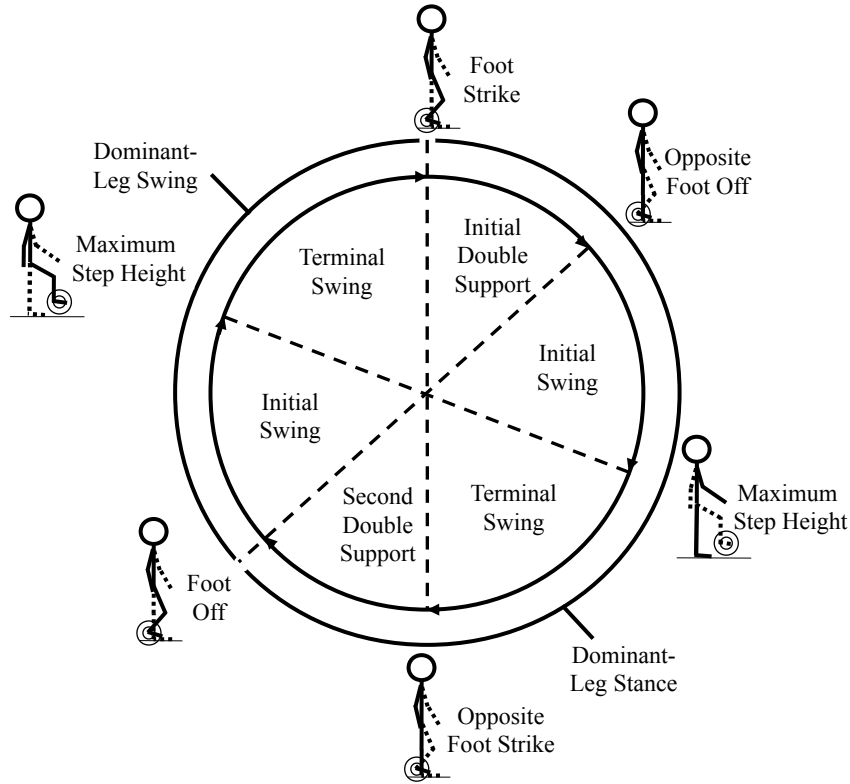


Figure 2.6: **In-place stepping gait cycle:** Periods, events, phases, and illustrations correspond with those in Figure 2.3. Period times are not generally equal.

time elapsed during only the three most recent states.

Using Dean’s equation (Equation 2.2), the measured step frequency and the user’s height yield intended speed.

### 2.3.2 How GUD WIP Fulfills LLCM-WIP’s Speed-Control Goals

GUD WIP achieves the virtual-walking speed-control goals presented with LLCM-WIP (Feasel et al., 2008):

- Smooth between-step walking speed
- Continuous within-step speed control
- Real-world turning and maneuvering
- Low starting latency
- Low stopping latency

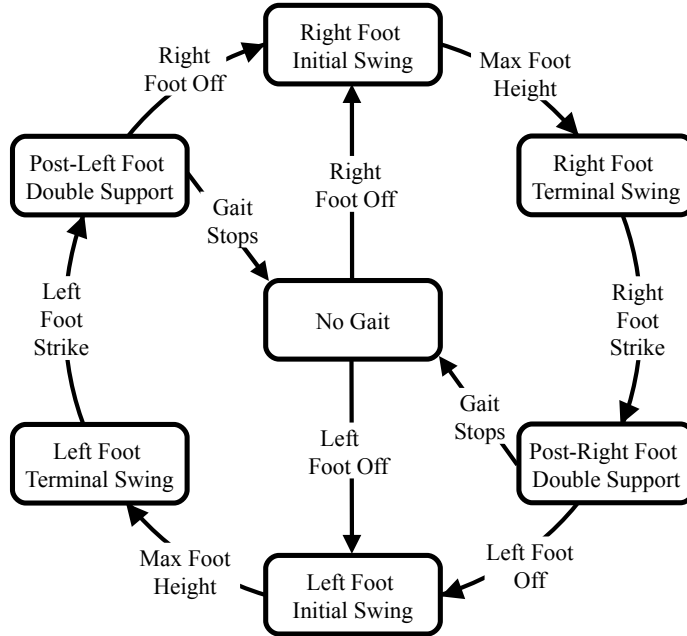


Figure 2.7: **GUD WIP state machine**: The current state is maintained until a state-exit criterion (shown on transitions) is fulfilled.

**Smooth between-step walking speed**: Since the measured step frequency persists through double support states, motion is smooth between steps – visually smoother than LLCM-WIP (compare Figures 2.2(a) and (b)). I present a metric for this smoothness in Subsection 2.5.2.1

**Continuous within-step speed control**: Since the measured step frequency updates at the completion of each state, within-step speed control is achieved: If the user slows down or speeds up his step motions within a step, the output speed updates accordingly, after the current state completes. This is not continuous within steps: It updates at three specific times within each step.

**Real-world turning and maneuvering**: To permit real-world motion without virtual motion, GUD WIP ignores horizontal foot motions and permits some vertical foot motion without identifying foot off. This allows users to rotate on the spot, or take small real steps so long as the user’s foot does not rise too high and too quickly. Additional information is provided in Subsection 2.4.1.

**Low starting latency**: As described so far, GUD WIP requires three completed states to measure step frequency – a full step’s latency. Through additional gait-cycle analysis, I

estimate step frequency to decrease starting latency.

Figure 2.4 shows that percent time ( $t_{\%}$ ) spent within each walking gait-cycle phase varies linearly with the step frequency ( $f$ ). Informal analysis of GUD WIP indicates that the same relationship holds for in-place stepping:

$$t_{\%} = m_{\text{state}}f + b_{\text{state}} \quad (2.3)$$

Figure 2.4 provides the slopes ( $m_{\text{state}}$ , seconds) and y-intercepts ( $b_{\text{state}}$ , unitless) for each state (is = initial swing, ts = terminal swing, and ds = double support). Assuming initial and terminal swing evenly split swing time, those values in step-frequency percent time are as follows:

$$m_{\text{is}} = 0.0806 \quad b_{\text{is}} = 0.2367 \quad (2.4)$$

$$m_{\text{ts}} = 0.0806 \quad b_{\text{ts}} = 0.2367 \quad (2.5)$$

$$m_{\text{ds}} = -0.1612 \quad b_{\text{ds}} = 0.5266 \quad (2.6)$$

By using the relationship between percent time and state time ( $t_{\%} = t_{\text{state}}/t_{\text{tot}}$ ), and the relationship between step frequency and total time ( $f = 1/t_{\text{tot}}$ ), the linear relationship between state percent time and step frequency (Equation 2.3) can be used to estimate step frequency from a single state's duration ( $t_{\text{state}}$ ) and hence decrease starting latency:

$$f = \frac{b_{\text{state}}}{t_{\text{state}} - m_{\text{state}}} \quad (2.7)$$

Thus, when only initial swing duration is known, the step frequency is estimated from that time alone (Equation 2.7). When both initial and terminal swing durations are known, GUD WIP estimates step frequency as the mean of the two estimates.

Only during the first initial swing state when stepping has begun, can no step frequency be estimated. To remove this starting latency, I use a variant of LLCM-WIP's predicted speed: LLCM-WIP transforms the vertical speed of the heel to yield the virtual walking speed; I use the maximum LLCM-WIP-transformed vertical heel speed in the first state.

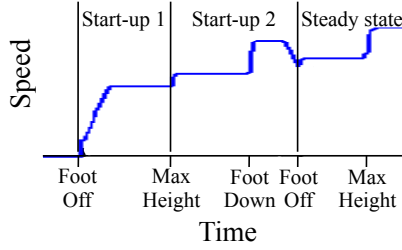


Figure 2.8: **Starting latency in GUD WIP**: Output speed is divided into three sections: **Start-up 1**: When the first state has begun (Foot Off detected), use the maximum LLCM-WIP-predicted value. **Start-up 2**: When one or two states have completed, estimate step frequency from Equation 2.7. **Steady state**: When three or more states have completed, the step frequency is the inverse of the past three states’ duration.

Thus, GUD WIP’s starting latency is the same as that of LLCM-WIP.

In summary, to create gait-informed virtual walking speeds at low starting latencies, GUD WIP uses the best gait-based information available at any time (Figure 2.8):

- **Start-up 1**: Within the first gait state, the maximum LLCM-WIP-generated speed is used.
- **Start-up 2**: When one or two states have completed, the completed states’ durations are averaged to estimate step frequency.
- **Steady state**: When continuous stepping has occurred for at least three states, the step frequency is calculated from the three most recent state durations.

**Low stopping latency**: GUD WIP’s stopping method also employs gait-understanding-driven state-duration prediction. In short, if a double-support state lasts longer than expected, “Gait Stops” is identified (see Figure 2.7).

Given step frequency ( $f$ ), the current state’s expected duration ( $t_{\text{state}}$ ) is estimated by inverting Equation 2.7:

$$t_{\text{state}} = m_{\text{state}} + \frac{b_{\text{state}}}{f} \quad (2.8)$$

Upon transitioning to a new state, GUD WIP estimates the new state’s duration. If the new state exceeds its estimated duration, the step frequency estimate decreases to account for this over-long state (Equation 2.7). Thus, the step frequency smoothly decreases until the next state begins.



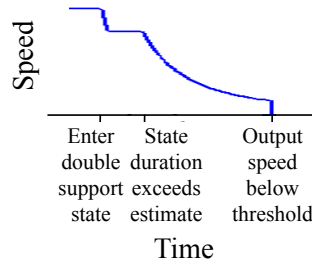


Figure 2.9: **Stopping latency in GUD WIP:** Output speed during stopping with three important events marked: First, upon entering a double support state, the new state’s duration is estimated (Equation 2.8). Second, when that estimate is exceeded, the output speed decreases to match the step frequency required by the state’s actual duration (Equation 2.7). Third, once the step frequency drops below a minimum threshold, all output is stopped.

The gait-stopping condition is a special over-long-state response for double-support states. Once step frequency in an over-long Double Support state decreases below a threshold, GUD WIP assumes the user intends to stop, and forward motion is halted (Figure 2.9). I threshold at 0.8 Hz, as rhythmic in-place stepping does not usually occur below it (Inman et al., 1981). Hence, virtual motion stops approximately 500ms after the last step ceases – approximately five times LLCM-WIP’s stopping latency.

There is an inherent trade-off between the goals of low stopping latency and between-step, continuous motion. If continuous motion between low-frequency steps is desired, stopping latency must exceed a Double Support state’s duration at low step frequencies. If lower stopping latency is desired, the step-frequency-cutoff threshold can be increased – but this removes smooth, low-step-frequency, between-step motion. Within GUD WIP, I chose continuous motion between steps over faster stopping latency. This tradeoff is alleviated by slowing virtual speed smoothly in over-long states. Visually GUD WIP’s smooth stop (Figure 2.2(b)) appears to better match real walking’s stopping characteristics (Figure 2.2(c)).

## 2.4 UNC GUD WIP Implementation and Performance

Any implementation of the model described in Section 2.3 implements GUD WIP. This section describes the UNC prototype’s key implementation details: identifying gait events

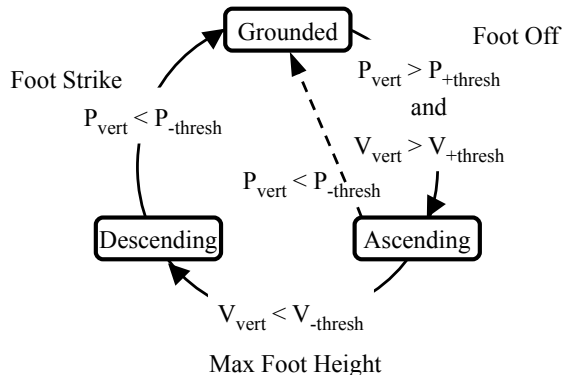


Figure 2.10: **A single foot’s in-place-step state transition diagram:**  $P_{\text{vert}}$  is heel height.  $V_{\text{vert}}$  is the heel’s vertical velocity. The gait-cycle event corresponding to each arc is named next to the arc’s exit criterion. The current per-foot gait state (Grounded, Ascending, or Descending) is maintained unless an exit criterion is fulfilled. The dashed transition handles possible input errors and is not a part of steady-state operation.

and implementing the gait cycle. Further details are found in Appendix A.

### 2.4.1 Gait Event Identification

Of the four in-place gait events (Foot Off, Max Foot Height, Foot Strike, and Gait Stops, see Figure 2.7), so far, only the implementation of Gait Stops has been described. This section describes my methods to detect the others.

Figure 2.10 illustrates gait-events identification for a single foot. I track each foot’s vertical position, filtering it with a PV-model Kalman filter (Kalman, 1960; Welch and Bishop, 2006). Thus I have both a smoothed position and a smoothed velocity for each foot. I use both in identifying gait events.

In this implementation, a foot is in one of three states: Grounded, Ascending, or Descending. Their steady-state transitions (Figure 2.10, on outside circle) define Foot Off, Max Foot Height, and Foot Strike. Foot Off occurs when the foot exceeds a vertical threshold ( $P_{+\text{thresh}}$ ) *and* an upward speed threshold ( $V_{+\text{thresh}}$ ). Requiring both conditions identifies stepping gestures without mistaking real-world maneuvering for intended steps. Max Foot Height occurs when the foot’s upward velocity decreases below a threshold ( $V_{-\text{thresh}}$ ). Foot Strike occurs when the foot’s vertical position descends below a threshold ( $P_{-\text{thresh}}$ ).

The following constants – when combined with my PV-model filter’s output – have

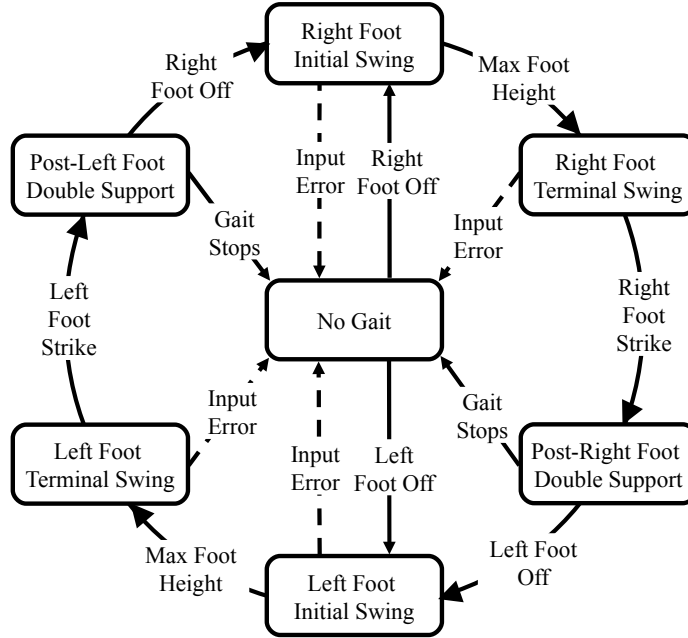


Figure 2.11: **The UNC GUD WIP implementation’s state machine:** The current state is maintained unless an exit criterion is fulfilled. The dashed transitions are for software robustness on input errors.

served to identify in-place-steps, and to permit undetected maneuvering steps:

$$P_{+\text{thresh}} = 0.02 \text{ m} \quad P_{-\text{thresh}} = 0.02 \text{ m} \quad (2.9)$$

$$V_{+\text{thresh}} = 0.5 \text{ m/s} \quad V_{-\text{thresh}} = 0.3 \text{ m/s} \quad (2.10)$$

Mathematically, an object descends when its vertical speed becomes negative. However, to compensate for the small latency added by the PV-model filter, I recognize descent slightly before the velocity actually becomes negative.

## 2.4.2 Gait Cycle Implementation

Implementing the gait cycle state machine shown in Figure 2.7 required increased robustness to input errors. The dashed transitions in Figure 2.11 handle the following error cases:

- **Tracker errors:** If shin tracker updates are lost, gait-state transitions may be missed. In these cases, the system returns to No Gait and awaits the next Foot-Off event.
- **Unexpected user input:** If a user performs an unrecognizable gesture (such as



Figure 2.12: **A user in the UNC GUD WIP system:** The user wears beacons for 6-DOF trackers on his shins (inset). Beacon-tracking cameras surround the subject on the floor. The heel’s position is approximated by rigid-body transform from the shin’s pose.

keeping his foot in the air) the system returns to the No Gait state and awaits the next Foot-Off event.

The surest catch for both error cases is an over-long-state recognition algorithm: If the predicted step frequency ever falls below the threshold, the system transitions to No Gait.

The step frequency measured by this state machine derives output speed via Dean’s formula for average users (Equation 2.2). If a user-specific step-frequency-to-walk-speed function is available, it can replace this generic equation.

### 2.4.3 Miscellaneous Implementation Details

When using LLCM-WIP estimation during start-up (Figure 2.8), I follow its proposed algorithm (Feasel et al., 2008).

The GUD WIP model yields only gait-based virtual walk speeds. Virtual walking also requires a walk direction; I employ the average of the shins’ direction.

### 2.4.4 Equipment

The UNC GUD WIP system uses an eight-camera PhaseSpace Impulse optical motion capture system (Figure 2.12). Seven LEDs on a rigid part-cylindrical plate are attached to each shin. Since three LEDs suffice for full tracking, this provides redundancy from forward

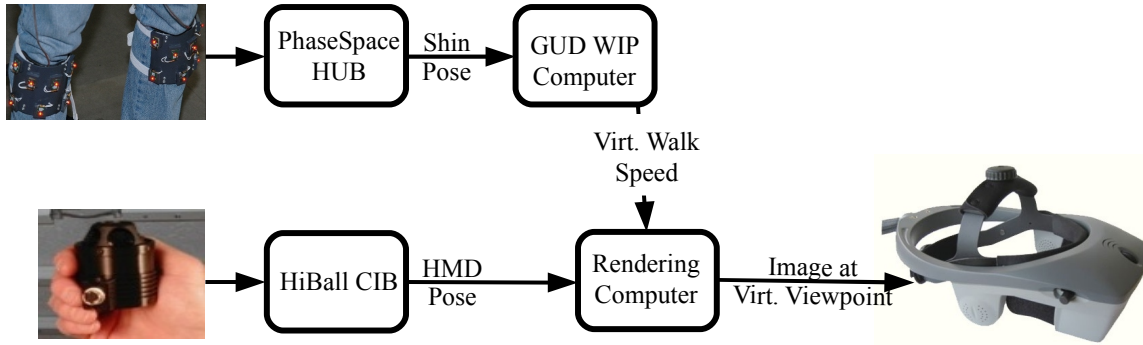


Figure 2.13: **UNC GUD WIP block diagram:** The UNC GUD WIP demo system uses four computers: Two specialty computers for the tracker systems, one for GUD WIP, and one for rendering.

views, while the curved surface enables tracking over a wide angle. The heel’s position is approximated from the shin’s pose via rigid-body transform. The cameras surrounding the user permit tracking from all angles.

This UNC GUD WIP implementation executes on a computer with Microsoft’s WindowsXP, an Intel Core2 2.4GHz CPU, NVIDIA GeForce 8600 GTS GPU, and 3 GB RAM. On this system, my GUD WIP implementation updates the virtual speed at approximately 1000Hz – GUD WIP’s calculations are not overly costly. Figure 2.13 provides a block diagram of all computers used by UNC GUD WIP.

With fellow graduate student Chris Oates, I created a demo VE system in which users travel through a model of a UNC Ackland Art Museum exhibit using the GUD WIP walking interface. This system employs a 3rdTech HiBall-3100<sup>TM</sup> tracker and an NVIS nVisor SX head-mounted display (HMD). The HMD imagery updates at 60 Hz, with vertical sync turned off to further decrease latency. Although not tested for this demo, a system using the same hardware in this lab had HiBall-measurement-to-HMD-image-update latency of approximately 50 ms.

## 2.5 Validation: Consistency

I assert that users experience specific consistent speed behaviors when really walking and that virtual walking systems should approximate those behaviors. Figure 2.2 visually demonstrates the speed characteristics of LLCM-WIP, GUD WIP, and real walking at

two different step frequencies. The improved smoothness visually evident when comparing LLCM-WIP's and GUD WIP's speeds demonstrates one important improvement provided by the GUD WIP model.

If, when really walking, speed changes consistently as a function of step frequency, this behavior would be of high importance for a WIP system. I assessed this property for three walking interfaces through a validation study:

- Hypothesis 1: In real walking, speed changes consistently with step frequency.
- Hypothesis 2: In LLCM-WIP, speed changes consistently with step frequency.
- Hypothesis 3: In GUD WIP, speed changes consistently with step frequency.

### 2.5.1 Method

**Design:** I performed a within-subjects, repeated-measures experiment. In each trial the user performed a walking interface (real walking or Walking-In-Place), stepping at one of four prescribed frequencies: 1.0, 1.5, 2.0, or 2.5 Hz. Each step frequency was repeated in four trials for each interface. Since LLCM-WIP and GUD WIP both interpret the same gesture – in place steps – only one set of data was collected for both. Each subject performed 32 trials, in all. A trial for each step frequency was performed before repeating a step frequency. Subjects performed all trials for one interface before proceeding to the next. Interface and step frequency order were assigned following Latin squares.

**Subjects:** Eight physically unimpaired subjects performed this user study (between 19–29 years old, approximately 1.75-1.9 meters, all male). All wore comfortable shoes, and pants which did not impair walking. Subjects were paid for their time.

**Procedure:** After the subject was told the study's purpose – to compare real walking and Walking-In-Place systems – he was set up for measurement in a stepping interface: Subjects wore a head-mounted 3rdTech HiBall-3100<sup>TM</sup> tracker when in real walking, the shin-mounted PhaseSpace beacons when in Walking-In-Place. In both interfaces, the subject's view of the real-world lab was unobstructed. Whether real walking or Walking-In-Place, the subject practiced stepping in time with a metronome set to a middle step frequency (1.67Hz). The subject then performed all trials with the metronome set to the appropriate

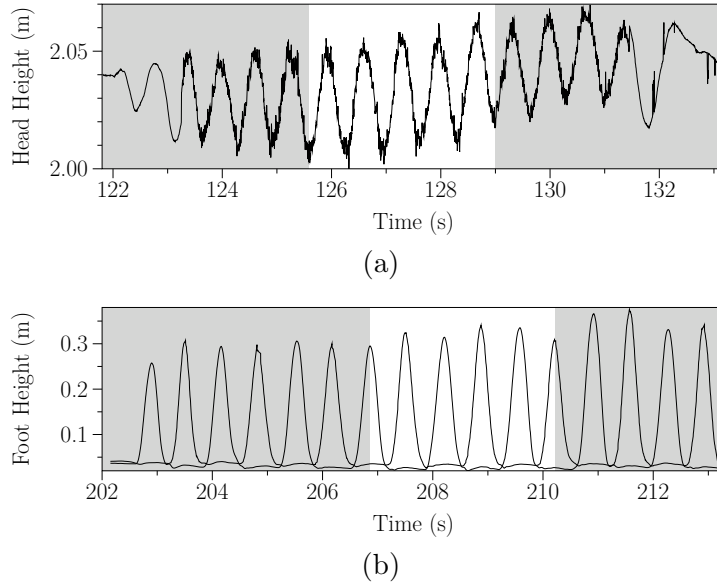


Figure 2.14: **Identifying five steps:** Five steps identified for (a) real walking (from the minimum head position during bobbing), and (b) Walking-In-Place (from foot position maxima). The same five-step period is used for both GUD WIP and LLCM-WIP computations.

step frequency. This procedure was repeated with the second interface.

For real walking, the subjects walked the longest straight-line distance possible in our head-tracked space (~9m). For Walking-In-Place, subjects took in-place steps until instructed to stop (15-30 steps).

**Measures:** During real-walking trials, I tracked the user’s head. During Walking-In-Place trials, I measured the shins’ positions. For real walking, I calculated speed through numeric differentiation of the subject’s head-tracked position. For Walking-In-Place, I calculated and recorded predicted virtual speed via both LLCM-WIP and GUD WIP as the subject stepped. I did not have access to these speeds until after the subject had completed all trials.

Both step frequency and speed were averaged over five rhythmic steps extracted from data from each walking interface. Five real-walking steps were identified as the time containing five complete vertical head-bobbing periods (Figure 2.14(a)). Five Walking-In-Place steps were identified as the time between five step’s maximum foot heights (Figure 2.14(b)) – with the same period for both GUD WIP and LLCM-WIP. For both interfaces, steps in the

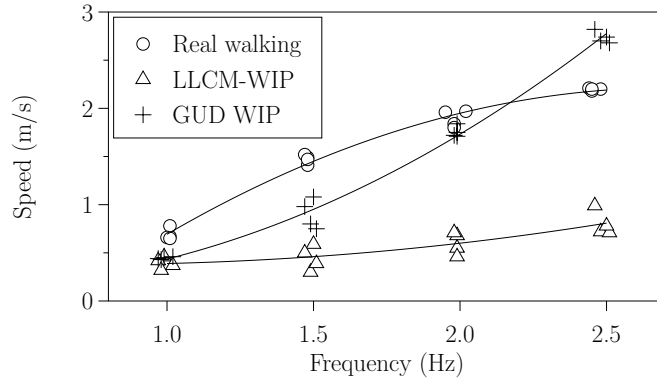


Figure 2.15: **Results for Subject 103:** Each trial provides a single data point. With four trials at each of four step frequencies, I calculate a quadratic function fit for each of three interfaces. Table 2.1 provides all subjects quality-of-fits.

middle of stepping were chosen to avoid start-up or stopping effects. Furthermore, I sought valid tracking and consistent stepping patterns.

While selecting the five steps, I did not look at the resulting speeds except on two or three trials when tracking errors required me to ensure that speeds had been available during some period.

Thus, for each trial, there were two measures: mean step frequency, and mean speed during the five-step period.

## 2.5.2 Results

Five real-walking condition trials were lost due to system errors. No subject lost more than one trial. One subject’s data were discarded due to his inability to increase and decrease his step frequency with the metronome, leaving me seven subjects.

**Step-frequency-to-walk-speed consistency:** I plotted each subject’s step frequencies and output speeds (each trial providing a single datapoint) for real walking, LLCM-WIP, and GUD WIP (Figure 2.15). I fitted both linear and quadratic functions to each condition’s datapoints. In all cases, the quadratic function had a higher  $r^2$  score indicating a better fit to the data, so I report and use only the quadratic fits. Real walking and GUD WIP both exhibit high-quality fits – LLCM-WIP exhibits less (Table 2.1).

**Curvature:** The characteristics of the real walking and LLCM-WIP fit functions varied



Walking Interface	n	Mean $r^2$	Std. Dev. $r^2$
Real walking	7	0.88	0.084
LLCM-WIP	7	0.67	0.178
GUD WIP	7	0.99	0.004

Table 2.1: **Quality of fit results:** The quality of fit ( $r^2$ ) for the quadratic functions to the average speeds for each subject in each interface (e.g., Figure 2.15). Real walking’s and GUD WIP’s fits are generally quite good, but LLCM-WIP’s fits are rarely as good.

considerably between subjects (Figure 2.16): Some subjects exhibit almost no curvature; others positive curvature; others negative. The curvatures varied by interface (Real walking  $d^2y/dx^2 = -1.02$ – $0.17$ ; LLCM-WIP  $d^2y/dx^2 = -0.70$ – $0.48$ ; GUD WIP  $d^2y/dx^2 = 0.70$ – $1.12$ ). A Pearson correlation did not find a significant within-subject correlation between real walking curvature and either of the virtual curvatures:

- Real-to-LLCM  $d^2y/dx^2$  correlation:  $r = -0.561$ ,  $p = 0.190$
- Real-to-GUD  $d^2y/dx^2$  correlation:  $r = -0.166$ ,  $p = 0.722$

### 2.5.2.1 Within-Step Consistency

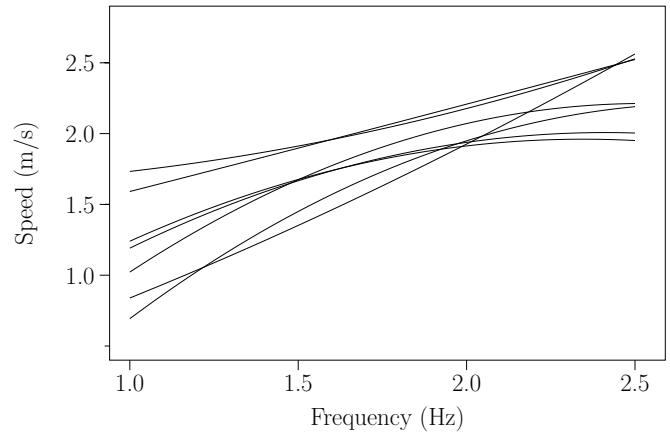
Previously, I have asserted that GUD WIP’s visual smoothness is better than LLCM-WIP’s. As a further test, I looked for *within-step consistency* in the virtual walking data: For LLCM-WIP and GUD WIP, did the output speed remain approximately the same within each manually-tagged window?

To test this, I calculated the mean and standard deviation for each manually tagged window’s resulting speeds. Because LLCM-WIP’s standard deviation varies with relative speed, I normalized the speed’s standard deviation ( $s$ ) by its mean ( $\bar{x}$ ):

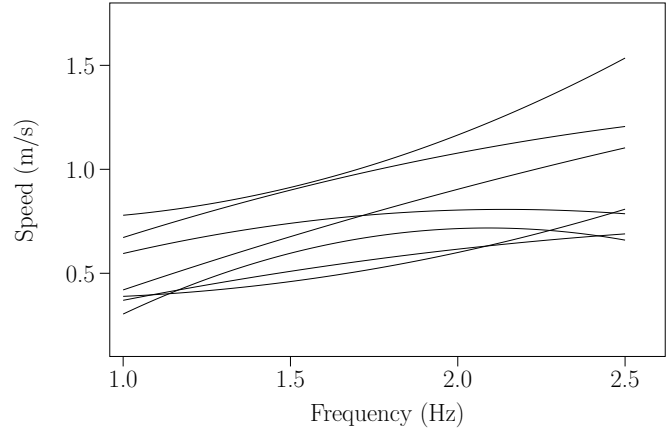
$$\hat{s} = \frac{s}{\bar{x}} \tag{2.11}$$

where  $\hat{s}$  is the normalized standard deviation.

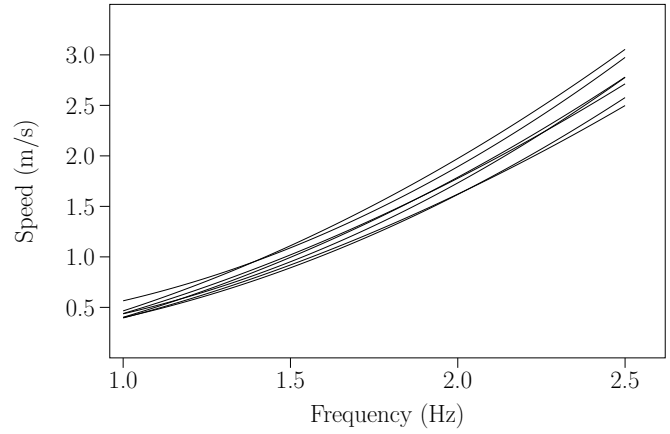
The mean normalized standard deviation by walking interface is shown in Figure 2.17. A repeated measures ANOVA showed that the difference between LLCM-WIP and GUD WIP is statistically significant. No change was found between the repeated trials (no training or fatigue for the standard deviations). Since sphericity failed for the repeated trials data, I



(a) Real walking



(b) LLCM-WIP



(c) GUD WIP

Figure 2.16: **Between-subject variation by interface:** The function fits for (a) real walking, (b) LLCM-WIP, and (c) GUD WIP for all seven subjects from my study. Note the variation of the fits for real walking and LLCM-WIP: Some are nearly linear, some curve upward, some downward. There was no significant within-subject correlation of these curvature changes.

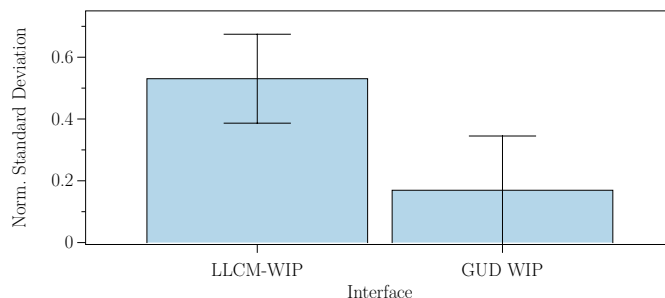


Figure 2.17: **Within-step consistency by interface:** The normalized standard deviation measures within-step consistency: If the normalized standard deviation is low, then the speed was consistent within steps. This shows the mean normalized standard deviation and its standard deviations by virtual walking interface.

use the Greenhouse-Geisser value.

- Interface difference:  $F(1, 12) = 75.491, p < 0.001, \omega = 0.92$
- Repeated trials:  $F(3.152, 37.822) = 0.645, p = 0.521$

Real walking was not included in this analysis because calculating its speed requires numeric differentiation. The numeric differentiation results in considerable noise in the output speeds – artificially increasing the standard deviations.

### 2.5.3 Discussion

**Step-frequency-to-walk-speed consistency:** In real walking, the subjects demonstrated within-subject consistent step-frequency-to-walking speed (Table 2.1, real walking data). This confirms my first hypothesis: Individuals consistently walk at the same speed when at the same step frequency. That is, they change step length and step frequency together, not separately. The second hypothesis is rejected: LLCM-WIP does not demonstrate high consistency. The third is accepted: GUD WIP is highly consistent.

I believe LLCM-WIP’s lower  $r^2$  fits fundamentally follow from LLCM-WIP’s algorithm. Since LLCM-WIP bases output speed on the vertical speed of the user’s heel, it is a function of both step frequency and step height. I found that even at consistent step frequencies, a single subject’s step heights varied (e.g., Figure 2.14(b)). Therefore, each step’s resulting vertical velocities differ.

**Speed Ranges:** The different within-subject speed ranges visible in the three conditions

(Figure 2.16) are unimportant. LLCM-WIP employs a user-specific multiplier to scale resulting output speeds. Herein, GUD WIP used Dean's mapping from step frequency to walking speed (Equation 2.2). However, a user-specific step-frequency-to-walk-speed function can be used.

Neither user-specific mapping was employed in this study: Finding the best user-specific function would have required all real-walking trials and analysis to occur before virtual trials.

Their respective user-specific mapping functions are an important difference between LLCM-WIP and GUD WIP: Had LLCM-WIP used the user-specific scalar, the resulting speeds for LLCM-WIP would have better matched real walking range. However, the LLCM-WIP output-speed curvatures would continue to show no significant correlation with real-walking's curvature. Had GUD WIP employed a user-specific step-frequency-to-walk-speed function, the resulting GUD WIP curve would precisely match real walking's because GUD WIP would base its resulting speed on that curve.

**Within-step consistency:** GUD WIP's visual within-step smoothness was confirmed by its statistically-significantly lower within-window output speed standard deviation. When the user takes consistent steps, GUD WIP's output speed is more consistently the same than LLCM-WIP's.

**Curvature:** I was surprised to see the between-subject variation in real-walking curvatures (some positive, some negative). I wondered if some within-subject stepping pattern, for both real walking and Walking-In-Place, could lead to matching curvatures for both real walking and LLCM-WIP. However, the non-significant, negative correlation indicates that this is likely a random difference with no meaningful within-subject cause.

GUD WIP's curvature is tautologically between-subject consistent because it based all subjects' output speed on Dean's quadratic equation (Equation 2.2). The slight variation is due to the subjects' varying heights.

LLCM-WIP's highly varying curvature is caused by different stepping behaviors for different subjects. LLCM-WIP's resulting speed depends on each step's height. I have seen that to reach higher WIP step frequencies, many users decrease their step height. This causes the negative curvatures. Zero curvature would indicate that the subject's steps

were approximately the same height at different step frequencies. Positive curvature would indicate larger step heights at increased step frequencies.

When performing the real-walking trials, many subjects shortened their step length to match the higher step frequencies. This is not the behavior described in biomechanics literature: The literature states that step length and step frequency increase together (Inman et al., 1981). Our tracked space’s maximum nine meter path may not have permitted the subjects to achieve a consistent walking speed at the higher step frequencies. Also, 2.5 steps per second is faster than some would choose to walk; they would choose to run instead. Together, these likely caused the real-walking function’s downward curvatures.

Employing a treadmill for future real-walking measurements would enable the subjects to achieve a comfortable, steady speed at any given step frequency.

**Improving real walking step-frequency-to-walk-speed functions:** I suggest that a better real-walking metric could be obtained by swapping the dependent and independent variables: Treadmill-walking subjects walk at fixed speeds, and step frequency is measured. However, this is only possible for real walking: Virtual walking speed can’t be fixed to find an in-place step frequency that matches. This technique could not be used for a virtual-versus-real walking study.

The technique used in this study required all subjects to walk at the same step frequencies. The technique described above requires all subjects to walk at the same speed. If variation in both step frequency and walk speed were permissible, I recommend allowing subjects to specify both: Tell subjects to walk “fast”, “medium”, or “slow” and allow them to determine what speeds and step frequencies they felt matched.

I believe either of these techniques would result in real walking step-frequency-to-walk-speed functions closer to those predicted by biomechanics. Nonetheless, these observations in no way invalidate this study’s results: Real walking evidences consistent step-frequency-to-walk-speed tendencies, and by this metric GUD WIP is more similar to real walking than LLCM-WIP.



## CHAPTER 3

# The Forward Walk Model: A Model to Extract Real Walking Parameters from Head Track Data Alone

### 3.1 Introduction

Chapter 2 introduces the GUD WIP model – demonstrating that WIP systems can effectively derive virtual speed from in-place step frequency. The validation user study presented in Section 2.5 demonstrated two important real walking characteristics:

1. A single person consistently walks at the same speed when walking the same step frequency (Table 2.1, real walking data), i.e., step length varies consistently with step frequency.
2. Different people’s step-frequency-to-walk-speed functions vary widely (Figure 2.16(a)).

The first indicates that step frequency is sufficient for WIP systems to create walking-like speeds. The second indicates between-user variation makes Dean’s average-user step-frequency-to-walk-speed function (Equation 2.2) a poor choice to fit all users. A user-specific step-frequency-to-walk-speed function would reduce system-user miscommunications. This chapter presents the Forward Walking Model which automates deriving a per-user step-frequency-to-walk-speed function from head-track data alone.

#### 3.1.1 Model Overview

To accurately present view-dependent virtual environments (VEs), VE systems must track the user’s head position. Due to cost and time to don full-body trackers, many VE systems

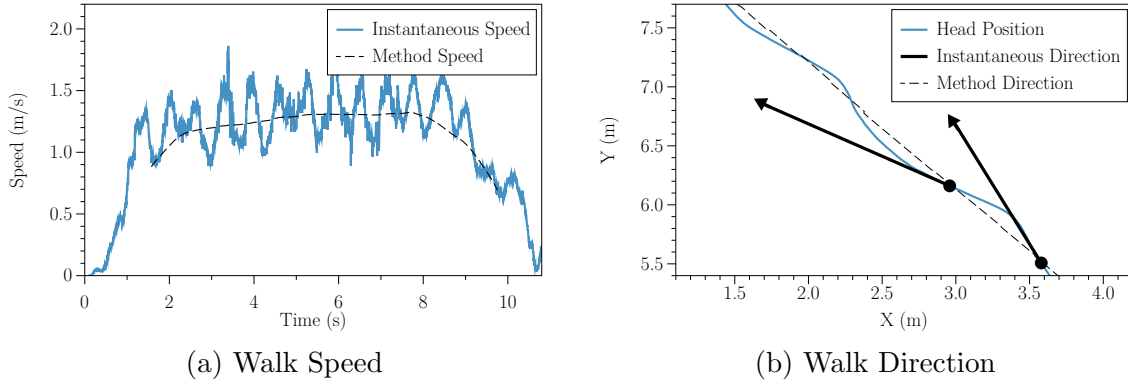


Figure 3.1: **Head-bobbing-caused error hinders accurate walk speed and direction estimation:** The instantaneous (a) horizontal walk speed (solid, lighter line) and (b) overhead-view of walk direction (thicker solid lines) compared to those obtained by my method (dashed lines).

track only the head. Previous research has demonstrated that users better perform tasks in VEs where the user really walks (Whitton et al., 2005; Ruddle and Lessels, 2009). Many VE applications – including self-avatar animation and Redirected Walking – could benefit from estimates of walking parameters such as walk speed, walk direction, step frequency, step length, and step timings. In this chapter, I present two methods that produce estimates for these values employing only head-track position data.

While walking, the motion of the head is governed by the biomechanics of human walking (Inman et al., 1981; Rose and Gamble, 2006). After basic translation, the most prominent characteristic of the head’s motion during walking is *head bobbing* which has three signed, orthogonal components:

- Right-left bobbing (hereafter *rightward bobbing*)
- Fore-aft bobbing (hereafter *forward bobbing*)
- Up-down bobbing (hereafter *upward bobbing*)

Bobbing complicates estimating a user’s instantaneous speed and direction (Figure 3.1). However, as will be shown, it can aid in estimating step frequency and step timings.

### 3.1.2 Other Applications for Walking Parameters

**Animating Self Avatars:** When comparing virtual walking interfaces, Usuh found that “presence significantly correlated with subjects’ degree of association with their virtual



bodies (avatars)” (Usoh et al., 1999). They suggested tracking the user’s limbs so the avatar could match his motion. The introduction of the Ninetndo Wii and similar game devices has added both user-body tracking and approximate self-avatar animations to gaming.

Walking-matching avatar motion has been demonstrated via full-body tracking systems, or by tracking a smaller number of key body parts. Chai and Hodgins track seven positions on the user’s body via simple video-based tracking, then employ a database including full-body-tracked motions to derive plausible walking self-avatar motions (Chai and Hodgins, 2005). Liu employs local linear models to simplify a database containing dozens of full-body-tracked animations to six key marker positions (Liu et al., 2006). From those six tracked positions, they derived full-body motions – including walking.

I am convinced that plausible user-matching walking animations can be created with only head tracking – using my methods’ estimates of the user’s position, walk direction, step length, and step timings.

**Redirected Walking:** Razzaque’s Redirected Walking and similar techniques enable users to walk through virtual worlds larger than the tracked space by disassociating the path traveled in the real world from the virtual-world path and imperceptibly redirecting users toward open space (Razzaque, 2005). To properly redirect the user, the algorithm needs good estimates of current position, walk direction, and speed – parameters that my methods provide.

Interrante’s Seven League Boots is a related technique that requires approximating the user’s bobbing-free walk direction to amplify the forward displacement (Interrante et al., 2007a). To estimate walk direction, they use a two-part approach using gaze direction as the bobbing-free direction when walking begins and a two-second-averaged instantaneous walk direction as the user continues to walk.

**User Study Analysis:** The methods presented herein could also improve post-hoc analysis of head-tracked walking logs resulting from comparative user studies. For instance, when comparing the characteristics of different walking interface’s paths, Whitton heavily smoothed Real Walking paths to eliminate head-bobbing’s effects (Whitton et al., 2005). This smoothing removes not only head bobbing, but also higher frequency components of

the underlying path: More smoothing leads to more smoothing of the bobbing-free path. The methods presented herein employ minimal smoothing to remove bobbing components.

**Chapter Overview:** This chapter presents the following:

- A mathematical model for forward-walking head movement,
- Two methods that approximate this model at real-time rates from head-track data, and
- Validation of these methods against real-walking data.

The model and approximation methods are the result of my research. They were refined through multiple meetings with David Adalsteinsson of my committee.

## 3.2 The Forward Walking Model

The Forward Walking Model presented here is based on the biomechanics of human walking (Inman et al., 1981; Rose and Gamble, 2006). A similar but simpler model was employed by Lecuyer (Lecuyer et al., 2006) to insert bobbing motions into keyboard or joystick-defined paths to simulate walking motions (see also (Hillaire et al., 2008; Terziman et al., 2009)).

Mathematically, forward-walking dynamic head positions ( $\mathbf{h}(t)$ ) are the combination of the three bobbing components ( $\mathbf{B}_{\mathbf{r}}(t)$ ,  $\mathbf{B}_{\mathbf{f}}(t)$ ,  $\mathbf{B}_{\mathbf{u}}(t)$ , for rightward, forward, and upward bobbing, respectively) and the remaining translation ( $\mathbf{T}(t)$ ):

$$\mathbf{h}(t) = \mathbf{B}_{\mathbf{r}}(t) + \mathbf{B}_{\mathbf{f}}(t) + \mathbf{B}_{\mathbf{u}}(t) + \mathbf{T}(t) \quad (3.1)$$

Each of the bobbing components can be estimated by a sinusoid:

$$\mathbf{B}_{\mathbf{v}}(t) = \mathbf{v}(t) \cdot a_{\mathbf{v}}(t) \cdot \sin(2\pi f_{\mathbf{v}}(t) \cdot (t - t_0) + \phi_{\mathbf{v}}(t)) \quad (3.2)$$

where  $\mathbf{v}$  is the bob's direction vector,  $a_{\mathbf{v}}$  is its amplitude,  $f_{\mathbf{v}}$  is its frequency,  $\phi_{\mathbf{v}}$  is its phase, and  $t_0$  is a centering term. Each varies with time.

The direction vectors for rightward ( $\mathbf{r}$ ), forward ( $\mathbf{f}$ ), and upward ( $\mathbf{u}$ ) are three orthonormal vectors defining a body-centric walking space that is related to the world-centric forward

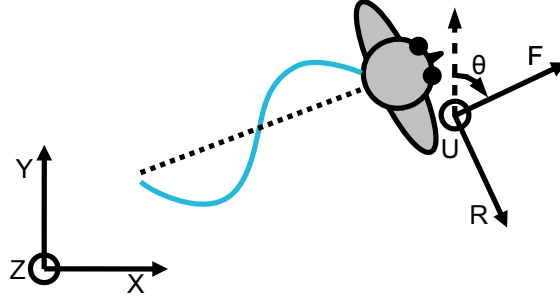


Figure 3.2: **User-centered frame:** The body-centric frame ( $r, f, u$ ) is related to the world frame ( $x, y, z$ ) by rotating around  $z$  by  $\theta$ .

direction in the horizontal plane by an angle ( $\theta$ ) (Figure 3.2):

$$\left[ \begin{array}{c|c|c} \mathbf{r}(t) & \mathbf{f}(t) & \mathbf{u}(t) \end{array} \right] = \left[ \begin{array}{c|c|c} \cos(\theta(t)) & -\sin(\theta(t)) & 0 \\ \sin(\theta(t)) & \cos(\theta(t)) & 0 \\ 0 & 0 & 1 \end{array} \right] \quad (3.3)$$

Equation 3.2 is constrained by biomechanics principles. Forward and upward bobs match the step frequency ( $f_s$ ); the rightward bob occurs at one-half the step frequency ( $f_s/2$ ).

If bobbing is removed from Equation 3.1, only the forward translation ( $\mathbf{T}$ ) remains. It is estimated as a bobbing-free head position ( $\mathbf{c}$ ), and a forward speed ( $s$ ):

$$\mathbf{T}(t) = \mathbf{c}(t) + \mathbf{f}(t) \cdot (s(t) \cdot (t - t_0)) \quad (3.4)$$

$$\mathbf{c}(t) = \left[ \begin{array}{ccc} c_x(t) & c_y(t) & c_z(t) \end{array} \right]^T \quad (3.5)$$

Finally, in order to better fit curved walking paths, the Forward Walk Model includes an angular velocity term ( $\omega(t)$ ) with the walk direction, replacing  $\theta(t)$  in Equation 3.3 with:

$$\theta(t) + \omega(t) \cdot dt \quad (3.6)$$

The full model of walking-induced head motion is found by expanding Equation 3.1 with

Equations 3.2–3.6:

$$\begin{aligned}
\mathbf{h}(t) = & \begin{bmatrix} \cos(\theta(t) + \omega(t) \cdot dt) \\ \sin(\theta(t) + \omega(t) \cdot dt) \\ 0 \end{bmatrix} (a_{\mathbf{r}}(t) \cdot \sin(\pi f_s(t) \cdot (t - t_0) + \phi_{\mathbf{r}}(t))) + \\
& \begin{bmatrix} -\sin(\theta(t) + \omega(t) \cdot dt) \\ \cos(\theta(t) + \omega(t) \cdot dt) \\ 0 \end{bmatrix} (a_{\mathbf{f}}(t) \cdot \sin(2\pi f_s(t) \cdot (t - t_0) + \phi_{\mathbf{f}}(t))) + \\
& \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} (a_{\mathbf{u}}(t) \cdot \sin(2\pi f_s(t) \cdot (t - t_0) + \phi_{\mathbf{u}}(t))) + \\
& \begin{bmatrix} c_{\mathbf{x}}(t) \\ c_{\mathbf{y}}(t) \\ c_{\mathbf{z}}(t) \end{bmatrix} + \begin{bmatrix} -\sin(\theta(t) + \omega(t) \cdot dt) \\ \cos(\theta(t) + \omega(t) \cdot dt) \\ 0 \end{bmatrix} (s(t) \cdot (t - t_0)) \quad (3.7)
\end{aligned}$$

Each of these parameters is necessary for different applications:

- User-specific step-frequency-to-walk-speed functions for WIP require step frequency ( $f_s$ ), and walk speed ( $s$ ).
- Real-time walking avatars require step frequency ( $f_s$ ), speed ( $s$ ), direction ( $\theta$ ), step length ( $l = s/f_s$ ), step timings ( $\phi_{\mathbf{r}}$ , foot-fall, foot-off, and swing events' timings), and bob amplitudes ( $a_{\mathbf{r}}$ ,  $a_{\mathbf{f}}$ ,  $a_{\mathbf{u}}$ ).
- Redirected Walking requires bobbing-free position ( $\mathbf{c}$ ), speed ( $s$ ), direction ( $\theta$ ), and angular velocity ( $\omega$ ).

### 3.3 Approximation Methods

I developed two methods for approximating the Forward Walking Model (Equation 3.7): A computationally-taxing Full-Model-Fitting Method, and an Expedited Method.

### 3.3.1 Full-Model-Fitting Method

This first method approximates all model parameters via the Levenberg-Marquardt Algorithm (LMA) – a numerical method for function minimization (Levenberg, 1944; Marquardt, 1963). Given a set of alterable parameters, LMA estimates the function’s gradient with respect to those parameters. It iterates to find a local minimum employing a hybrid Gauss-Newton/Gradient-Descent Algorithm.

I use the `lmfit` implementation of LMA (Wuttke, 2009) to fit head-track data ( $\mathbf{h}(t)$ ) to the Forward Walking Model (Equation 3.7) by minimizing the error between a 3 second window of head-track data samples and the model-predicted head position, given the following LMA-modifiable parameters:

- Step frequency ( $f_s$ )
- Bobbing sinusoid amplitudes ( $a_r, a_f, a_u$ )
- Bobbing sinusoid phases ( $\phi_r, \phi_f, \phi_u$ )
- Walk direction ( $\theta$ )
- Angular velocity ( $\omega$ )
- Walk speed ( $s$ )
- Bobbing-free head position ( $c_x, c_y, c_z$ )

#### 3.3.1.1 Implementing the Full-Model-Fitting Method

Each 3 second data window has  $\sim 150$  head-track data samples – each including time ( $t_i$ ) and position ( $h_x(t_i), h_y(t_i), h_z(t_i)$ ). Using the window’s center time ( $t_{0j} = (t_{n-1} - t_0)/2$ ), I seek a value for each LMA-modifiable parameter.

`lmfit` requires a method that accepts the LMA-modifiable parameters, and calculates the error ( $\epsilon(t_i)$ ) for each input datapoint. I use the two-norm ( $L_2$ ) error between the model-

predicted values (Equation 3.7) and the actual measurements as follows:

$$\epsilon(t_i) = \sqrt{\epsilon_{\mathbf{x}}(t_i)^2 + \epsilon_{\mathbf{y}}(t_i)^2 + \epsilon_{\mathbf{z}}(t_i)^2}, \text{ where} \quad (3.8)$$

$$\begin{aligned} \epsilon_{\mathbf{x}}(t_i) = h_{\mathbf{x}}(t_i) - \\ (c_{\mathbf{x}}(t_{0j}) + \cos(\tilde{\theta}(t_i)) \cdot \tilde{h}_{\mathbf{r}}(t_i) - \sin(\tilde{\theta}(t_i)) \cdot \tilde{h}_{\mathbf{f}}(t_i)) \end{aligned} \quad (3.9)$$

$$\begin{aligned} \epsilon_{\mathbf{y}}(t_i) = h_{\mathbf{y}}(t_i) - \\ (c_{\mathbf{y}}(t_{0j}) + \sin(\tilde{\theta}(t_i)) \cdot \tilde{h}_{\mathbf{r}}(t_i) + \cos(\tilde{\theta}(t_i)) \cdot \tilde{h}_{\mathbf{f}}(t_i)) \end{aligned} \quad (3.10)$$

$$\epsilon_{\mathbf{z}}(t_i) = h_{\mathbf{z}}(t_i) - (c_{\mathbf{z}}(t_{0j}) + \tilde{h}_{\mathbf{u}}(t_i)), \text{ and where} \quad (3.11)$$

$$\tilde{\theta}(t_i) = \theta(t_{0j}) + \omega(t_{0j}) \cdot (t_i - t_{0j}) \quad (3.12)$$

$$\tilde{h}_{\mathbf{r}}(t_i) = a_{\mathbf{r}}(t_{0j}) \cdot \sin(\pi f_s(t_{0j}) \cdot (t_i - t_{0j}) + \phi_{\mathbf{r}}(t_{0j})) \quad (3.13)$$

$$\begin{aligned} \tilde{h}_{\mathbf{f}}(t_i) = a_{\mathbf{f}}(t_{0j}) \cdot \sin(2\pi f_s(t_{0j}) \cdot (t_i - t_{0j}) + \phi_{\mathbf{f}}(t_{0j})) \\ + s(t_{0j}) \cdot (t_i - t_{0j}) \end{aligned} \quad (3.14)$$

$$\tilde{h}_{\mathbf{u}}(t_i) = a_{\mathbf{u}}(t_{0j}) \cdot \sin(2\pi f_s(t_{0j}) \cdot (t_i - t_{0j}) + \phi_{\mathbf{u}}(t_{0j})) \quad (3.15)$$

where  $\tilde{\theta}$ ,  $\tilde{h}_{\mathbf{r}}$ ,  $\tilde{h}_{\mathbf{f}}$ , and  $\tilde{h}_{\mathbf{u}}$  are the model-driven approximations for direction, and the rightward, forward, and upward components of the head's position.

Although these equations suffice for a naïve fit, I found the following factors important in improving the final result's correctness: parameters' initial values, constraining parameters, steady-state calculations, and bobbing-fit improvements. I address each in turn.

**Parameters' initial values:** LMA finds a local minimum without guarantee that it be a global minimum. Therefore, the initial values provided to `lmfit` must be closer to the global minimum than to other local minima.

Through experimentation, I found that initial values for speed, direction, and step frequency most influence global minimum discovery. I initialize speed and direction with

Parameter	Units	Min	Max
$a_{\mathbf{r}}$	m	0.005	0.10
$a_{\mathbf{f}}$	m	0.005	0.08
$a_{\mathbf{u}}$	m	0.005	0.08
$c_{\mathbf{z}}$	m	see caption	see caption
$f_s$	Hz	0.8	3.8
$s$	m/s	0	6
$\omega$	Radians/s	$-\pi$	$\pi$

Table 3.1: **Range constraints for various walking parameters:**  $c_{\mathbf{z}}$  is constrained to be within the middle two quartiles of the current data window’s z-value range. All non-listed parameters have no constraints. These minimum and maximum possible values were chosen based on pilot data and from what may be found in the literature (summarized in (Inman et al., 1981)).

the mean values of the sampled speeds and directions on the 3 second data window:

$$s_0 = \frac{1}{n-1} \cdot \sum_{i=1}^{n-1} \left( \sqrt{(h_{\mathbf{x}i} - h_{\mathbf{x}i-1})^2 + (h_{\mathbf{y}i} - h_{\mathbf{y}i-1})^2} \right) / (t_i - t_{i-1}) \quad (3.16)$$

$$\theta_0 = \frac{1}{n-1} \cdot \sum_{i=1}^{n-1} \text{atan2}((h_{\mathbf{y}i} - h_{\mathbf{y}i-1}), (h_{\mathbf{x}i} - h_{\mathbf{x}i-1})) - \frac{\pi}{2} \quad (3.17)$$

where  $s_0$  and  $\theta_0$  are the initial guesses for the speed and walk direction, and `atan2` is the C-language’s method for the arc-tangent that preserves the angle’s full range.

To estimate initial step frequency, the method counts vertical-bob crossings of the window-averaged vertical values. This coarse estimate is close enough for `lmfit` to yield the global minimum.

All remaining variables are initialized to the constraining range’s middle value.

**Constraining parameters:** LMA minimizes in the parameter’s gradient-vector direction and does not directly permit specifying permissible ranges for the parameters. However, the biomechanics of walking constrain several parameters’ possible values. For instance, rhythmic walking step frequencies stay within 0.8 to 3.8 Hz. Table 3.1 lists the constrained parameters’ permissible ranges.

To help `lmfit` keep variables within permissible ranges, I post-multiplied the error term ( $\epsilon$ , see Equation 3.8) with a multiplier ( $m$ ) – itself the sum of a multiplier ( $m_i$ ) for each

parameter ( $p_i$ ):

$$m = 1 + \sum_{i=0}^{\text{numParameters}} m_i \quad (3.18)$$

$$m_i = \begin{cases} 0 & \min \leq p_i \leq \max \\ (\min - p_i)/(\max - \min) & p_i < \min \\ (p_i - \max)/(\max - \min) & p_i > \max \end{cases} \quad (3.19)$$

This multiplier does not alter the parameters' gradients when they are within their acceptable ranges. For out-of-range parameters, the multiplier increases the gradient linearly with distance from the range – pushing them toward valid range.

**Steady-state calculations:** When the user walks at a steady pace, execution time can be improved by using inter-parameter relationships. I call this case “steady-state calculation”.

I have discussed Full-Model-Fit functionalizing for any head-track data window. However, a single new piece of head-track data advances the window only one data point: Most of the window remains a good fit for the previously-estimated parameters. Furthermore, mathematical relationships permit maintaining some parameters' previous values.

Several model parameters are related by derivatives:

$$\frac{d\theta}{dt} = \omega \quad (3.20)$$

$$\begin{bmatrix} \frac{dc_x}{dt} \\ \frac{dc_y}{dt} \end{bmatrix} = \begin{bmatrix} -\sin(\theta + \omega \cdot dt) \\ \cos(\theta + \omega \cdot dt) \end{bmatrix} \cdot s \quad (3.21)$$

$$\frac{d\phi}{dt} = 2\pi f \quad (3.22)$$

The backward Euler approximations for Equations 3.20–3.22 relate previous parameter values (at  $t_{0(j-1)}$ ) to current parameter values (at  $t_{0j}$ ) as follows:



$$\theta(t_{0j}) = \theta(t_{0(j-1)}) + \omega(t_{0j}) \cdot dt \quad (3.23)$$

$$\phi_{\mathbf{r}}(t_{0j}) = \phi_{\mathbf{r}}(t_{0(j-1)}) + \pi f_s(t_{0j}) \cdot dt \quad (3.24)$$

$$\phi_{\mathbf{f}}(t_{0j}) = \phi_{\mathbf{f}}(t_{0(j-1)}) + 2\pi f_s(t_{0j}) \cdot dt \quad (3.25)$$

$$\phi_{\mathbf{u}}(t_{0j}) = \phi_{\mathbf{u}}(t_{0(j-1)}) + 2\pi f_s(t_{0j}) \cdot dt \quad (3.26)$$

$$c_{\mathbf{x}}(t_{0j}) = c_{\mathbf{x}}(t_{0(j-1)}) - \sin(\theta(t_{0j}) + \omega(t_{0j}) \cdot dt) \cdot s(t_{0j}) \cdot dt \quad (3.27)$$

$$c_{\mathbf{y}}(t_{0j}) = c_{\mathbf{y}}(t_{0(j-1)}) + \cos(\theta(t_{0j}) + \omega(t_{0j}) \cdot dt) \cdot s(t_{0j}) \cdot dt \quad (3.28)$$

where  $dt = t_{0j} - t_{0(j-1)}$ .

Using Equations 3.23–3.28 to replace  $\omega(t_{0j})$ ,  $\phi_{\mathbf{f}}(t_{0j})$ ,  $\phi_{\mathbf{r}}(t_{0j})$ ,  $\phi_{\mathbf{u}}(t_{0j})$ ,  $c_{\mathbf{x}}(t_{0j})$ , and  $c_{\mathbf{y}}(t_{0j})$  in Equations 3.8–3.15 simplifies the error evaluation: The walk direction, three phases, and two horizontal-position components are no longer free parameters – the previous iteration’s results are used. After the simplified minimization completes, these six parameters are updated following Equations 3.23–3.28.

This steady-state enhancement speeds up `lmfit` execution by decreasing the number of free parameters, leading to fewer LMA iterations. However, the relationships stated in Equations 3.20–3.22 hold only so long as the angular velocity, step frequency, walk speed, and walk direction change only slightly. Larger changes indicate changes in walking pattern and require reverting to the non-steady-state, full-parameter solver. To avoid accumulating error, I reinitialize the steady-state four times per second by rerunning the non-steady-state solver.

**Bobbing-fit improvements:** Due to its relative simplicity, fitting the vertical component of the head’s position ( $h_{\mathbf{z}}$ , Equation 3.11) was swift and robust. At times, I found that the horizontal components – two bobs, direction, and speed – went to local minima that did not match the real-walking values. To address this I increased the weight of errors of the more robust vertical component, replacing Equation 3.8 with the following:

$$\epsilon(t_i) = m \cdot \sqrt{\epsilon_{\mathbf{x}}(t_i)^2 + \epsilon_{\mathbf{y}}(t_i)^2 + \lambda \cdot \epsilon_{\mathbf{z}}(t_i)^2} \quad (3.29)$$

Setting  $\lambda$  to 80 leads to good convergence.

### 3.3.1.2 Full-Model-Fitting Method example

**Implementation:** I implemented this method as a single-threaded C/C++ system. When run on a Macintosh laptop with 2.13 GHz Intel Core 2 Duo CPU, 2 GB RAM, and Mac OS X 10.5.8, as people walk, it ran at about 50 updates per second.

Figure 3.3 shows the results of running the Full-Model-Fitting Method against a sharply curved walking path. This case provides a demanding dataset that provides a good example of why the model needs to include angular velocity: As will be seen with the Expedited Method, a solely linear approximation does not fit as well. Figure 3.3(a): Angular velocity enables the method to stay within the original input data. Figure 3.3(b): The bobbing-induced speed changes are effectively ignored. Figure 3.3(c): The bobbing-induced direction change is removed. Figure 3.3(d): The bobbing-free vertical position cuts through the vertical bob. Figure 3.3(e): The step frequency is relatively stable at 1.6 Hz. Figure 3.3(f): the two-norm error is quite low at all times except during starting and stopping. The method works very well during rhythmic walking. Section 3.6 provides further validation of this method.

### 3.3.1.3 Full-Model-Fitting Method discussion

This example illustrates several inherent model-fitting difficulties.

**Non-orthogonal parameters:** Many conceptually different walk-model parameters can provide similar results as each other:

- *Direction and rightward bob:* Figure 3.4 illustrates this point. When provided a data window containing only one full cycle of rightward bob, an angular-velocity-including fit leads to considerable walk direction errors. (By employing only angular velocity, I am fitting the horizontal head components to a circular arc.) Approximately 1.25 rightward-bob cycles appears to be the minimum needed to eliminate this effect. Because of this requirement and the lowest acceptable step frequency (0.8 Hz), I use a 3 second window.

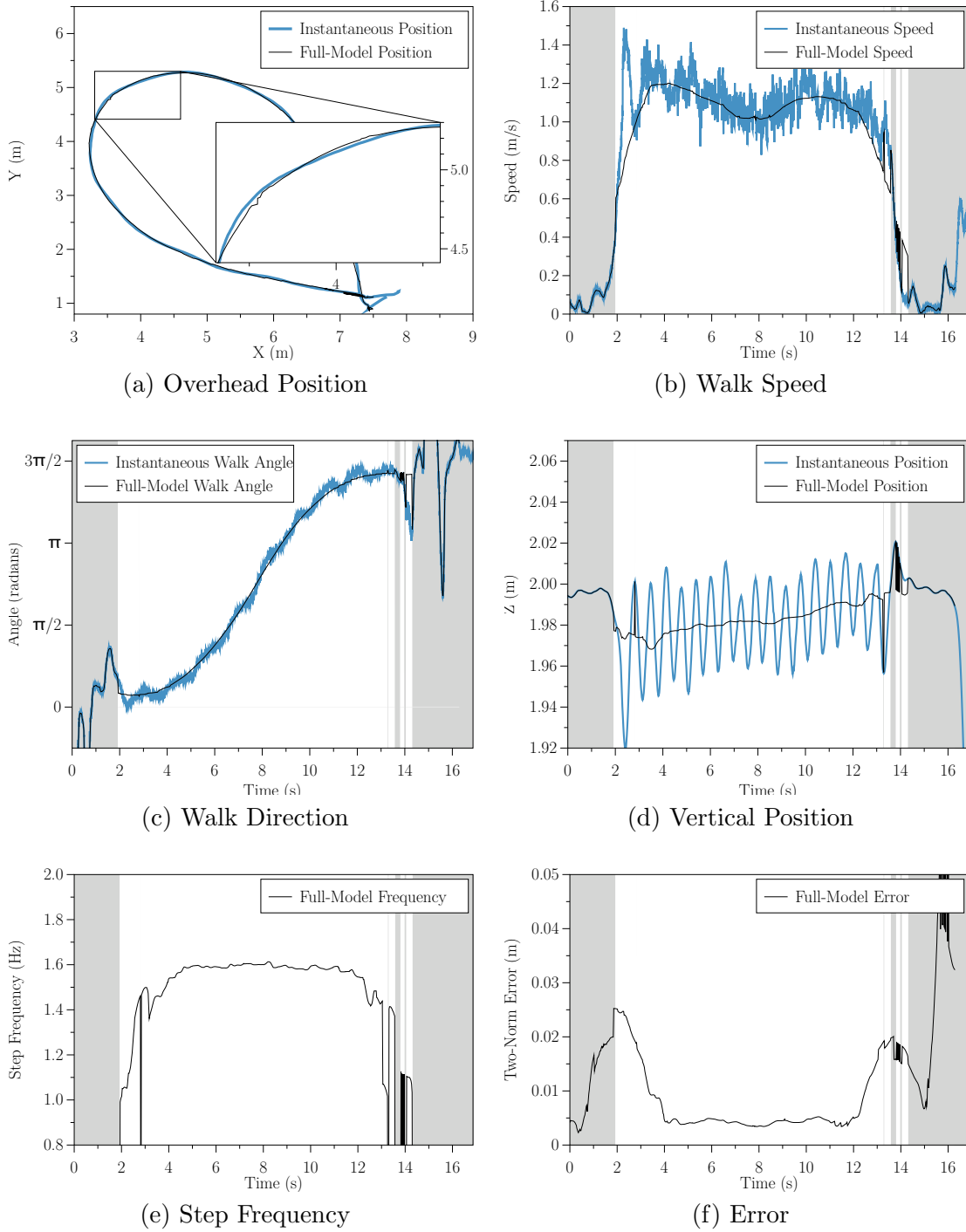


Figure 3.3: **Full-Model-Fit Results**: User walked a loop. Thicker, lighter line: head-track data. Thinner, darker line: Full-Model-Fit Method. Background color: walking recognized when white. (a) Overhead view with magnification inset. (b) Horizontal walk speed. (c) Horizontal walk direction. (d) Vertical position. (e) Step frequency. (f) Two-norm error.

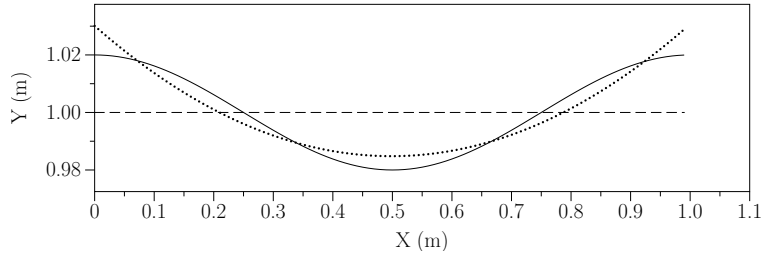


Figure 3.4: **Rightward-bob/angular-velocity error:** This simulated, overhead view of a single rightward-bob of walking data (solid line) demonstrates what errors arise when fitting the bobbing-free path to too little data. The dashed line shows the users’ true bobbing-free path. However, with only one bob’s data, an angular-velocity-including path (dotted line) mistakes the bob for the path.

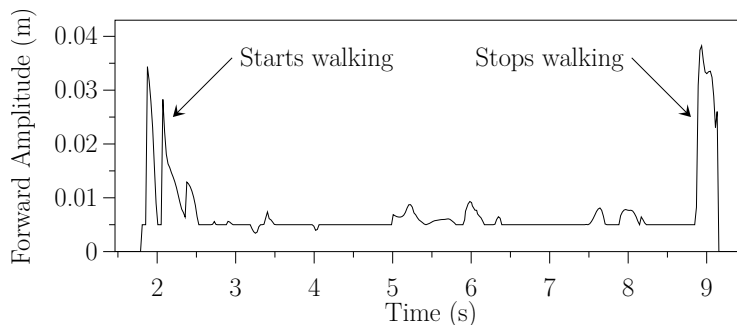


Figure 3.5: **Acceleration/forward-bob error:** The resulting forward bob amplitude for a user quickly walking a straight line at a constant pace. Note that LMA’s resulting forward-bob amplitude increases dramatically during the brief starting and stopping periods.

- *Acceleration and forward bob:* Changes in forward acceleration lead to forward-bob errors (Figure 3.5). LMA accounts for the acceleration-caused errors by increasing the forward-bob displacement.

**Accumulating error:** This error is best described by an example: During steady-state operation, the system updates the bobbing-free position by integrating a whole-window-fitting estimate of the angular velocity, walk direction, and walk speed (Equations 3.27 and 3.28). Even when these estimates are good fits for the whole window, they may not be good fits for the center time ( $t_{0j}$ ). This problem occurs most often when step frequency, walk speed, and walk direction change considerably. Thus, when step frequency, walk speed, or walk direction change at a rate higher than my experimentally determined thresholds (respectively, 1.5 Hz/s, 1.2 m/s<sup>2</sup>, 0.2 rads/s), I reset the LMA calculation to non-steady-state.

### 3.3.2 Expedited Fitting Method

Some applications do not need all of the parameters provided by the model. Others find the Full-Model-Fitting Method’s computational cost too high: To achieve 50 updates per second, it monopolizes a processor. Furthermore, since the model parameters are computed for the data window’s central time ( $t_{0j}$ ), the Full-Model-Fitting Method’s 3 second window leads to 1.5 seconds of latency. For many, this latency is too high. The Expedited Method trades fitting the complete model for computation at least 10 times faster than the Full-Model-Fitting Method, and approximately half the latency.

#### 3.3.2.1 Implementing the Expedited Method

The Expedited Method is a two-stage process: First, a partial-model LMA fit is performed. Second, other parameters are approximated.

**Stage 1 – LMA fit:** The Full-Model-Fitting Method heavily weights the vertical head-bob component of the model (Equation 3.29). Its 3 second window size is required to avoid confounding angular-velocity and rightward-bob model components (Figure 3.4). By fitting only the vertical components of the model (Equations 3.11, 3.15, and 3.26), the Expedited Method can fit to a smaller 1.5 second window. Many of the Full-Model-Fitting Method’s LMA-fitting enhancements still apply: Parameters must be properly initialized and constrained, and steady-state enhancements can be used.

Upon Stage 1 completion, there are good estimates for step frequency ( $f_s$ ), step phase ( $\phi_u$ ), the bobbing-free vertical position ( $c_z$ ), and the vertical bob’s amplitude ( $a_u$ ).

**Stage 2 – Approximate parameters:** Various of the remaining parameters can now be accurately approximated. I approximate speed ( $s$ ), direction ( $\theta$ ), and the horizontal position ( $c_x$  and  $c_y$ ).

The principal confounder in estimating each of these parameters is head bobbing. For such sinusoids, the following equation is true for any frequency, amplitude, or phase, and for any integer  $n$ :

$$0 = \int_i^{i+n \cdot \frac{1}{f}} a \cdot \sin(2\pi f \cdot x + \phi) dx \quad (3.30)$$

The integral of any sinusoid over an integral number of periods is zero.

Therefore, if  $\bar{X}_{t_{\text{init}}}^{t_{\text{fin}}}$  is the arithmetic mean from time  $t_{\text{init}}$  to  $t_{\text{fin}}$ , I estimate each of these bobbing-free parameters as follows:

$$\theta = \bar{X}_{(t_n - \frac{2}{f_s})}^{t_n} (\text{atan2}((h_{\mathbf{y}i} - h_{\mathbf{y}i-1}), (h_{\mathbf{x}i} - h_{\mathbf{x}i-1}))) - \frac{\pi}{2} \quad (3.31)$$

$$s = \bar{X}_{(t_n - \frac{2}{f_s})}^{t_n} (-\sin(\theta) \cdot (h_{\mathbf{x}i} - h_{\mathbf{x}i-1}) + \cos(\theta) \cdot (h_{\mathbf{y}i} - h_{\mathbf{y}i-1})) \quad (3.32)$$

$$c_{\mathbf{x}} = \bar{X}_{(t_n - \frac{2}{f_s})}^{t_n} (h_{\mathbf{x}}) \quad (3.33)$$

$$c_{\mathbf{y}} = \bar{X}_{(t_n - \frac{2}{f_s})}^{t_n} (h_{\mathbf{y}}) \quad (3.34)$$

where  $t_n$  is the time of the most recently received data sample. The mean over two step periods is used because the rightward bob occurs at one-half the step frequency.

**Two time samples:** Stages 1 and 2 use different size data windows to estimate their results: Stage 1 uses a fixed-width 1.5 second window; Stage 2 uses a variable-width 2 step-period window. The parameters are estimated at two different times. Thus there are two latencies: one is fixed (0.75s); the other varies (one step-duration). At step frequencies above 1.33Hz (a modest walking pace), the Stage 2 parameters' latency is lower than the Stage 1 parameters.

**Remaining parameters:** The model parameters not estimated by the Expedited Method are the angular velocity, and the horizontal bobbing's amplitudes and phases.

### 3.3.2.2 Expedited Method example

**Implementation:** When run on the same system configuration, and against the same input data as the Full-Model-Fitting Method, the Expedited Method runs ~10 times faster.

The Expedited Model's estimated parameters' results closely approximate those of the Full-Model-Fitting Method. Figure 3.6 shows some of these. Figure 3.6(a): During hard turns, the linear, angular-velocity-free approximation of the path is unable to stay within the curved path's bobs. Figure 3.6(b) and (c): Speed and direction are very similar to the Full-Model-Fitting Method's results, and the smaller window size better matches the acceleration and deceleration periods. Figure 3.6(d): Because of its smaller window size,

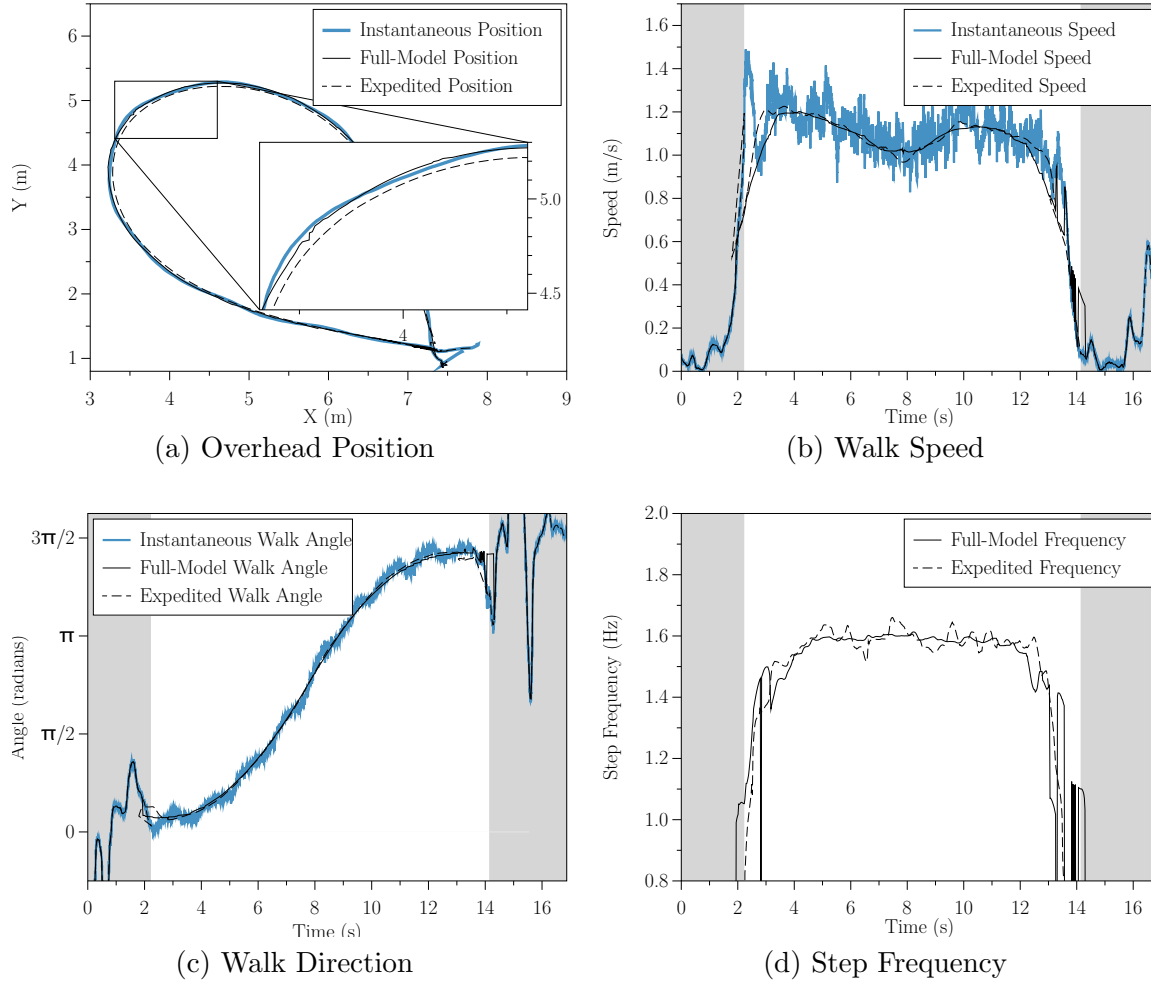


Figure 3.6: **Expedited Method results compared Full-Model-Fit Method:** User walked a loop. Thicker, lighter line: head-track data. Thinner, darker line: Full-Model-Fit Method. Dashed-line: Expedited Method. Background color: walking recognized when white. (a) Overhead view with magnification inset. (b) Walk speed. (c) Walk Direction. (d) Step Frequency. Note that although the Expedited bobbing-free position is somewhat less accurate than the Full-Model’s, the bobbing-free walk directions are nearly identical.

the Expedited Method’s step frequency varies more.

### 3.3.2.3 Expedited Method discussion

Since the second stage’s window size depends on the first-stage-estimated step frequency, two issues arise:

- The second stage parameters’ center time does not monotonically increase. When the step frequency decreases, the second-stage window size increases – moving the

window’s center time. If this window-size increase is large enough, the second-stage time can move backwards. This effect is visible in Figure 3.6(b) at approximately 2 seconds.

- When the first-stage model fit converges to an incorrect minimum, all second-stage parameters are inaccurate. This occurs most frequently when the user is not walking. Therefore, I added a Walking-Recognition Method, trusting second-stage results only when walking is recognized.

### 3.3.3 Recognizing Walking

When the user is walking, the vertical head bob matches a sinusoid very well. When the user is not walking, his vertical head position is largely unchanging. These observations provide the following Walking Recognition heuristic.

To recognize periods of walking, I perform two LMA fits to the vertical data: a linear fit, and a sinusoidal fit. Both the Full-Model-Fitting and Expedited Methods already perform the sinusoidal fit, and the linear fit (with only two parameters) is fast.

When the error for the vertical sinusoid fit (Equation 3.11) is lower than the linear fit’s error *and* all vertical parameters are within real-walking ranges (Table 3.1), the system recognizes the user as walking. When the user is not walking, the window-centered value of all instantaneously-approximable parameters ( $\mathbf{c}$ ,  $s$ , and  $\theta$ ) are output. When the user is walking, all method-approximated parameter values are output.

The results of this heuristic are shown with a white background in Figures 3.3, and 3.6. This method recognizes walking within the first half step and recognizes stops within the last half step. Some false negative blips are visible as spikes in the output in Figure 3.3. When walking is not recognized, both methods revert to the instantaneous value for position, speed, and direction; and to zero for step frequency, and step timing. The jagged-looking outputs seen near the edges of the white portions in the figures could be filtered by any application requiring real-walking data.



### 3.4 Comparison With Existing Systems

I know of no existing systems that provide high-frequency (50 Hz) updates for walking parameters. In Exercise Science, commercial systems exist that provide estimates of some of these parameters at 1-3 Hz (e.g., (Innovative Sports Training, 2010; GAITRite, 2010)). For instance, to estimate step length and step frequency, these systems employ trackers near the feet, and pressure sensors under each foot to identify foot-fall and foot-off location and events. Estimates are available only after two consecutive foot-fall events and update only with later foot falls.

Interrante estimate a bobbing-free walk direction through fixed-width averaging over a 2 second window – similar to Equation 3.31 (Interrante et al., 2007a). This leads to higher direction errors and latency than my methods. Their fixed window size – not an integral number of steps – generally includes a non-zero amount of all bobs (see Equation 3.30). The effect of this error declines as step frequency increases. With a two-second window, the latency in their estimates is greater than that of the Expedited Method for all step frequencies above 1 Hz.

### 3.5 Fourier Techniques

I attempted a Fourier-based approximation method before adopting LMA. However, the Fourier-based methods were considerably less accurate at matching step frequency than the LMA-based methods. The following is a brief description of my findings. Appendices B and C provide further details.

If one walks at a fixed step frequency ( $f$ ), his head’s vertical trace is approximately a sinusoid. Transformed to frequency domain, the trace becomes two spikes ( $+f$  and  $-f$ ). Windowing the signal in the spatial domain multiplies it by a box function – equivalent to convolving the frequency domain representation with a sinc function, with the sinc function’s maximum height at the original spike’s location.

Narrower box filters (with lower latency and less smoothing) lead to wider sinc functions – causing the  $-f$  sinc function’s magnitude to be non-negligible at the  $+f$  sinc function’s

center. Given perfect sine inputs, at 1.2 cycles per window, errors are on the order of 0.2Hz. Even at 3.0 cycles per window (a four-second window size for lower step frequencies), errors are on the order of 0.05Hz.

Given the same data, Levenberg-Marquardt minimization produces errors below  $10^{-6}$ Hz.

## 3.6 Validation: Approximation Methods Provide Accurate Results

Lacking a system that provides the same walking parameters at similar frequencies as my approximation methods, I had to validate the output of my methods against averaged results from manually-tagged windows: As part of the GUD WIP validation user study (Section 2.5), I gathered and manually tagged five-step regions in head-track logs of users really walking (Figure 2.14). I validated the approximation methods' outputs against average step frequency, walk direction, and walk speed from this dataset.

### 3.6.1 Recap of Method of Real Walking Portion of User Study (Section 2.5)

**Design:** Each subject walked an approximately 9m straight-line path four times at each of four prescribed step frequencies. The step frequencies were ordered using Latin Squares; each was executed before any repeated.

**Subjects:** There were eight paid subjects – all male, all with unimpaired walking, ages 19-29.

**Procedure:** Subjects wore a 3rdTech HiBall-3100<sup>TM</sup> head tracker. Subjects practiced walking to a metronome (1.67 Hz) then began their trials. The metronome's speed during trials was 1.0, 1.5, 2.0, or 2.5 Hz.

**Measures:** The head's vertical position was graphed and manually tagged to identify periods of consistent walking – periods containing five complete vertical head-bobbing cycles (Figure 2.14(a)). I estimated three “ground truth” average values over these five-step periods: average speed and direction were calculated following Equations 3.31 and 3.32; average

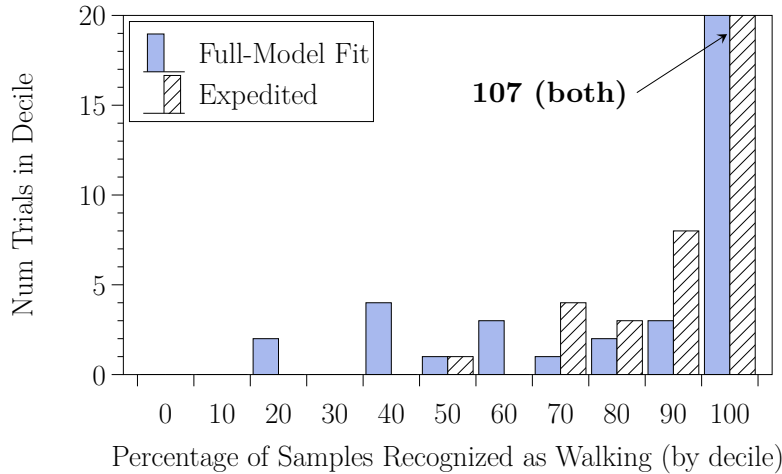


Figure 3.7: **Walk-identified deciles:** Each trial recognized some percentage of the samples on the manually-tagged window (Figure 2.14(a)) as walking. Each method performed perfectly – recognizing all samples as walking – on 107 of the 123 trials (87%). To illustrate, the 20% decile shows that two Full-Model-Fit Method trials recognized between 20-29% of the windows’ samples as walking. The Expedited Method trended toward better scores on the 13% of the trials that imperfectly identified walking.

step frequency was estimated as five steps divided by period duration.

For this validation, I also ran the Full-Model-Fitting and Expedited Methods against these data logs to estimate Forward Walking Model parameters.

### 3.6.2 Results

Five trials were lost due to system errors. No subject lost more than one trial.

**Accuracy of Recognition of Rhythmic Walking:** Before valid estimates can be obtained for step frequency, walk speed, and walk direction, the methods must converge to a walking-recognized state (Section 3.3.3). The number of method-recognized-walking samples was divided by the number of samples in the manually-identified window – yielding a per-trial percentage of correctly identified samples. A perfect approximation method would recognize 100% of the samples in the windows as walking. Figure 3.7 splits the trials by their percent-walk recognized deciles. Each of the estimation methods recognized walking perfectly in 107 of 123 total trials (87%). For the 13% of the trials that imperfectly recognized walking, the Expedited Method trended toward higher percentage-correct recognition scores.

**Estimated Parameters:** I averaged the methods’ estimates for step frequency, walk speed, and walk direction over the manually-identified windows. I compared these averaged, method-generated values ( $\text{val}_{\text{mg}}$ ) to the “ground truth” averaged values ( $\text{val}_{\text{gt}}$ ). For walk direction, I used the absolute error (in degrees). For step frequency and walk speed, I used the percent error ( $\text{err}_{\%}$ ):

$$\text{err}_{\%} = \frac{\text{abs}(\text{val}_{\text{gt}} - \text{val}_{\text{mg}})}{\text{val}_{\text{gt}}} \quad (3.35)$$

Table 3.2 reports the results. I divided the trials into two groups: perfect walking-recognized scores (100%), and imperfect scores (<100%). I report the error scores of the fiftieth, ninety-fifth, and hundredth percentiles. These three percentiles demonstrate the errors’ trends. When they recognized walking perfectly, each method gave highly accurate values almost uniformly. Even when failing to perfectly recognize walking, the methods yielded quite good results (less than 10% error) in about 95% of the trials.

### 3.6.3 Data Validity

The data analyzed herein have one major strength and one major weakness. Their weakness is that the data includes only straight-line paths. Although I tested these methods against a single curved path (Figures 3.3 and 3.6), errors may still exist in the approximation methods. However, since it includes angular velocity, I believe that the model itself is sound for curved paths.

The data’s strength is their wide range of step frequencies. Several subjects found both 1.0Hz (very slow) and 2.5Hz (very fast) rather difficult step frequencies. Because of these responses, I expect all users’ step frequencies to remain within this step-frequency range.

For the purposes of this dissertation – creating user-specific step-frequency-to-walk-speed functions – the data’s weakness is unimportant; its strength is very important.

### 3.6.4 Discussion

**Method-recognized walking:** I examined why 32 of 246 trials had imperfect walking recognition scores. The principal problem is too many bobbing cycles in the window. In 20

Method	Perc. Walk. Recog.	n	Metric	Units	50%	95%	100%
Full-Model	100%	107	Freq.	%Err	0.6	2.0	3.1
			Speed	%Err	1.5	4.4	8.2
			Dir.	deg	0.6	1.7	5.2
Expedited	100%	107	Freq.	%Err	0.5	1.7	3.1
			Speed	%Err	1.2	4.0	6.2
			Dir.	deg	0.5	1.1	2.3
Full-Model	<100%	16	Freq.	%Err	6.3	13.7	31.5
			Speed	%Err	2.6	7.0	83.9
			Dir.	deg	0.6	2.3	3.4
Expedited	<100%	16	Freq.	%Err	3.3	9.5	18.8
			Speed	%Err	1.8	4.1	5.3
			Dir.	deg	0.6	1.7	5.72

Table 3.2: **Estimated parameters’ accuracy:** Methods’ step frequency, walk speed, and walk direction errors compared with “ground truth” averages at the 50<sup>th</sup>, 95<sup>th</sup>, and 100<sup>th</sup> percentiles. Results are split between perfect walking-recognized trials (100%), and imperfect (<100%). To illustrate, the top row shows that of the 107 perfect walking-recognized trials for the Full-Model-Fitting Method, 53 had error less than 0.6%, 99 had error less than 2%, and the worst had 3.1% error.

of the imperfect-score trials, the subject walked at over 2 Hz. As stated in Subsection 3.3.1, the window size was chosen to include 1.2 head-bob sinusoidal cycles at the lowest allowed step frequency (0.8 Hz). At step frequencies of 2Hz or higher, the window contains at least 3 rightward-bob periods’ data. The more cycles of walking data, the more likely some portion of the Forward Walking Model (Section 3.2) is violated. The two most common model violations are within-window step frequency variation, and within-window bobbing-free vertical head position ( $c_z$ ) variation.

The error in response to over-large window size partially explains why the Expedited Method trended toward better recognition scores: The model assumptions hold better because it uses a smaller window size. I believe a fruitful later enhancement will be to automatically adapt the window size based on step-frequency estimates.

**Estimated parameters:** As Table 3.2 shows, the worst results are outliers in the Full-Model-Fit’s speed with imperfect walking-recognition scores. There are two such trials. In both cases, the method found a local minimum with a poor estimate of angular velocity. An angular-velocity-initializing value (similar to those for walk speed, walk direction, and step frequency) would fix this problem.

**Real-time considerations:** Both approximation methods execute at real-time or better rates. However, I suggest the following areas for real-time-processing consideration:

- *Head turns:* When the head turns, the head’s position moves somewhat – obscuring the bobbing components. To minimize such effects, I suggest fitting the model to the position of the base of the skull by rigid-body transform from the tracker location.
- *Latency:* The Full-Model-Fitting Method estimates parameters for a central value on a 3 second window (1.5s latency). The Expedited Method estimates Stage 1 parameters at the central value on a 1.5 second window (0.75s latency) and Stage 2 parameters at the central value on a 2 step-period window (~0.4-1.0s latency). In steady-state real walking, the rate of change for step frequency, step length, etc. is low. Therefore, although these parameters are most accurate for the window’s central time, systems can use them as the current value (zeroth-order approximation; as done by Interrante (Interrante et al., 2007a)). Higher-order approximations could lead to better latency-removed parameter estimates.
- *Trusting parameters:* Some parameters are more trustworthy than others. For instance, the estimates of amplitude and phase of the upward bob were usually both reliable and consistent. The forward bob was far less so (Figure 3.5). Any use of forward bob should be sensitive to the estimation errors the method reports.
- *Post-hoc filtering:* As is visible in Figure 3.3(b)–(e), there are times that the walk-recognized state toggles. I have seen this most during starting or stopping periods.

Similarly, when the system eliminates accumulated error after several steady-state calculations, some parameter's values show small discontinuities. I recommend using post-method filtering to remove both.

This approximation-method validation against real-walking data demonstrates that the approximation methods can be used to replace manually-tagging head-track logs to derive user-specific step-frequency-to-walk-speed functions. By using the Forward Walking Model, it is feasible to personalize GUD WIP instead of using Dean's average function (Equation 2.2).





## CHAPTER 4

# Usability Study

The two previous chapters introduce two new real-walking-based models and WIP systems based on those models. Each system is accompanied by a validation analysis that demonstrates that it functions properly: Like real walking, UNC GUD WIP creates consistent step-frequency-to-walking-speeds; the Forward Walk Model approximation methods generate walking parameters that match those that can be estimated from five-step averages of manually-tagged head-tracked data.

However, this dissertation's thesis is the following: *Walking-In-Place systems are measurably improved by using real-walking-based models.* This chapter presents a usability study to demonstrate that each of the two models measurably improves WIP systems. I built two variants of GUD WIP: The first employs the GUD WIP model using Dean's average-user step-frequency-to-walk-speed equation (Equation 2.2; hereafter, Dean GUD WIP). The second employs the GUD WIP model to measure in-place step frequency, and the Forward Walk Model to measure a per-user step-frequency-to-walk-speed function (hereafter, Personalized GUD WIP). To demonstrate that each of the models measurably improves WIP systems, I compare the model-based WIP systems to a previous speed-focused WIP system – LLCM-WIP (Feasel et al., 2008). Specifically, I test the following hypotheses:

1. Compared to LLCM-WIP, Dean GUD WIP and Personalized GUD WIP each measurably improve users' ability to travel in virtual environments by the following metrics: path completion speed, within-path walk accuracy (lack of collision), and stop accuracy.
2. Compared to LLCM-WIP, Dean GUD WIP and Personalized GUD WIP each measurably improve users' response to virtual-walking systems by at least one of the following metrics: self-reported usability, presence, and simulator sickness.

The study failed to produce statistically significant results for Hypothesis 1. However, several of Hypothesis 2's metrics demonstrate with statistical significance that these real-walking-based models improve WIP systems.

## 4.1 Method

**Design:** I performed a mixed-design experiment – between-subject condition (WIP system) with repeated-measures trials (four different hallways, each repeated four times). The trials were ordered by Latin Squares. Subjects were randomly assigned to one of three WIP systems:

- LLCM-WIP
- Dean GUD WIP
- Personalized GUD WIP

In the repeated trials, the subjects were instructed to quickly and accurately travel down a hallway to a specified endpoint.

**Subjects:** Twenty-nine physically unimpaired subjects performed the user study. Five were replacements due to procedure error (see below), resulting in twenty-four subjects (ages 18-57 years old (mean = 23); 18 male, 6 female). All wore comfortable shoes, and pants that did not impair walking. Subjects volunteered in response to an email or personal contact, and were paid for their time.

**Procedure:** Independent of WIP system, all subjects had the same real-world experience:

- Study introduction
- Walking measurement
- Virtual walking trials
- Questionnaires
- Exit interview

*Study introduction.* Subjects received information about the study's purpose, and an overview of what they would experience. Each subject read, and signed a consent form. Subjects were told about space constraints in VEs, and introduced to the idea of Walking-

In-Place interfaces to remove space constraints. The subject was not told to which WIP system he was assigned.

*Walking measurement.* Subjects walked a straight-line path (~7m) in our lab twelve times. Their head position was tracked with a 3rdTech HiBall-3000™ tracker attached to a head-band in such a way to permit full real-world vision. Subjects walked at a subject-defined slow, medium, or fast pace – each pace repeated four times, ordered randomly.

During these paths, the Expedited Method (Subsection 3.3.2) calculated bobbing-free walk speed and step frequency – saving one step-frequency/speed data point for each path (12 total points). These points were calculated as the average walk speed and step frequency within a middle, two-second portion of each path. If no two-second period with 100% Walk Identified was found, subjects repeated the trial. If two attempts did not obtain a good datapoint, that trial was skipped. The resulting datapoints were fitted with a quadratic function to provide a personalized step-frequency-to-walk-speed function.

The personalized step-frequency-to-walk-speed function was used differently by the WIP systems.

- LLCM-WIP’s algorithm defines a per-subject scalar ( $m_{\text{LLCM}}$ ). I used the function-derived 2 Hz-step-frequency walk speed ( $s_{2\text{Hz}}$ ) as follows:

$$m_{\text{LLCM}} = \frac{s_{2\text{Hz}}}{0.75} \quad (4.1)$$

The denominator is based on the average 2 Hz speed in Figure 2.16.

- Dean GUD WIP did not use the per-subject walk function.
- Personalized GUD WIP replaced Dean’s equation with this personalized function.

*Virtual walking trials.* At this point, subjects began virtual walking trials. After donning the head-mounted display, subjects saw a virtual world consisting of long, narrow hallways with varying number of turns, and curvature (Figure 4.1). These long narrow hallways test subjects’ abilities to travel accurately when employing a WIP system. The Straight Hallway measures virtual walking speed and accuracy (lack of collision) with little need to focus on walk direction. The angled hallways require local accuracy at two difficulty levels: The subject has to get to his desired position at the desired speed to quickly and accurately

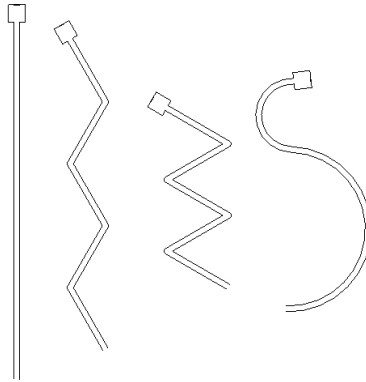


Figure 4.1: **Study hallways:** The four hallways users experienced – Straight, Obtuse, Acute, and Curved. The hallways were textured to look like mine shafts. All hallways were the same length (50 meters).

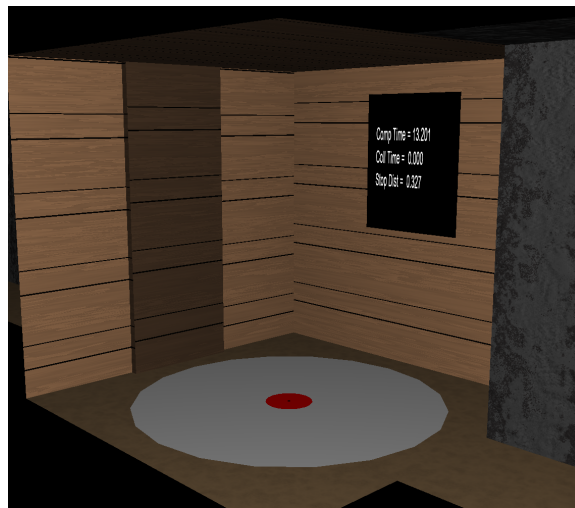


Figure 4.2: **Between-trial resting room:** Each trial began and ended in this resting room. This image shows a hallway wall (far right) entering the resting room, the stopping target (on the ground), the scoreboard (center wall), and the door (darkened patch on the left wall).

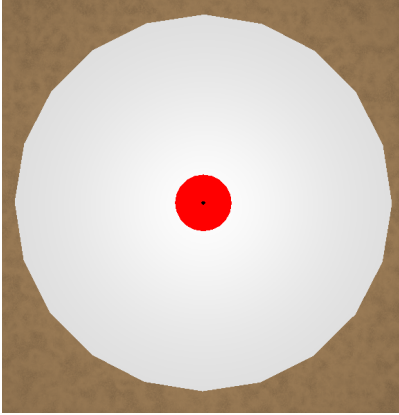


Figure 4.3: **The stopping target:** Subjects had to be within the white circle to end a trial. Stopping distance was measured as the horizontal distance from the center of the black dot to subject’s eyes’ location.

negotiate the turn without colliding with the walls. The Curved Hallway requires subjects to make continual small directional adjustments while walking.

The hallways were textured to appear like mine shafts. Each hallway ended and began in a small resting room (Figure 4.2). Trials began when the subject opened the resting-room door to enter the hallway. Subjects were instructed to walk each hallway as quickly and accurately (avoiding collisions) as possible. They were explicitly told that both speed and accuracy were important and they should focus on both. Upon ending a hallway in the second resting room, subjects were told to press a button to indicate that they were over the stopping position (Figure 4.3). Throughout the hallway, subjects received visual feedback – the screen flashed red – whenever their virtual positions placed their viewpoint within 0.21 m of a wall – this was defined as “colliding”. That is, collision occurred when the subject’s invisible avatar (a 0.21-meter-radius cylinder centered at the virtual viewpoint) collided with the walls.

At the end of each hallway, the subject’s total hallway time, total time in collision, and distance from the stopping position were reported on a wall-mounted board in the resting room (Figure 4.2).

Each group of four hallways was combined into a set. Subjects repeated each set four times – the within-set hallway ordering determined by Latin squares. Subjects were asked if they wanted to rest after each set. Subjects were required to remove the head-mounted

display and rest in the real world at least after every second set, approximately every 15 minutes. This real-world rest lasted approximately 5 minutes.

Subjects were not instructed how to effect different walking speeds within their assigned WIP system. The only instruction the experimenter provided was to take larger (higher) steps to ensure in-place steps were differentiable from real-world maneuvering.

*Questionnaires.* After completing all virtual trials, subjects completed a questionnaire, comprised of five sections. Three sections are in-house-created surveys to collect gender, race, and ethnicity data (required by our funding agency), subjects' Likert ratings of the virtual walking interface, and free responses (Appendix D, Sections II, IV, and V). The other two are standard VE questionnaires for virtual-world presence (Appendix D, Section III, modified from (Slater et al., 1998)) and simulator sickness (Appendix D, Section I, from (Kennedy et al., 1993)).

*Exit interview.* During the exit interview, I asked the subjects for any problems or strange issues they experienced, if they had any questions, and reminded them not to drive for 30 minutes after completing the study.

After the exit interview, interested subjects were told which WIP system they had been in and provided information on how the different systems created virtual walking speeds.

**Ecological validity:** I chose virtual mine shafts as the virtual world for this usability study because long, narrow hallways with curves and varying angle turns generally don't occur in normal buildings. Mines are built following much less rigid rules of taste: Miners follow the ore deposits, and these frequently lead to odd paths.

This virtual world choice leads to two ecological validity questions. First, are VE systems used for mining applications? I contacted Launa Mallett, Ph.D. of the National Institute for Occupational Safety and Health (NIOSH) who creates VE applications for training miners. She stated that mines such as those I'm using may exist for copper or other ores, and that NIOSH has interest in using VEs for training miners.

Second, do results derived from this study apply to non-mining VE systems? The series of long narrow hallways is intended to measure if WIP users are able to accurately and quickly travel their desired path. The mine-shaft hallway provides visual feedback for a

constrained path. However, WIP users in less constrained still want to travel a constrained path: In this situation, the constraint is self-imposed instead of environment-imposed. I believe these results apply to non-mining VE systems.

**Measures:** I measured the per-trial total hallway time (hereafter, Total Time), time in collision (Collision Time), and horizontal distance from the target at stopping (Stopping Error). To decrease between-subject variation, I normalized the non-Straight-Hallway Total Times by dividing them by the within-set Straight Hallway's Total Time. This Normalized Total Time measures how hallway complexity affects subjects' walk speed: A Normalized Total Time greater than one indicates the subject had to take more time in that hallway than they would have in the Straight Hallway. I calculated the Nausea (N), Occulomotor discomfort (O), Disorientation (D), Simulator Sickness (SS), and Presence scores from the questionnaire data. Free responses were coded for later analysis (method described with results).

**Exclusion criteria:** After running 16 subjects, I discovered a calibration error that increased the shin direction error associated with how the trackers were mounted on the shins. If the trackers were mounted approximately correctly, this error was negligible. However, if the trackers were mounted such that they did not point directly forward, the error could be quite large.

In discussion with Gordon Pitz and Mary Whitton, we determined that this error would exhibit itself by excessive collisions in the straight hallway: If the calibration error altered forward motion to one side, it would be harder to walk a long straight hallway without colliding. Therefore, I fixed the calibration error, and ran all remaining subjects without the error. After completing 24 subjects, I replaced those subjects from the first 16 subjects that had the worst average collision scores on the straight hallway (Figure 4.4) with new subjects in the same WIP system. Five subjects' data were replaced, resulting in 11 "with error" and 13 "without error" data. A MANOVA did not find a significant difference in Straight Hallway Collision Time or Total Time between the remaining "with error" and "without error" subjects.

- Total Time:  $F(1) = 0.343$ ,  $p = 0.564$

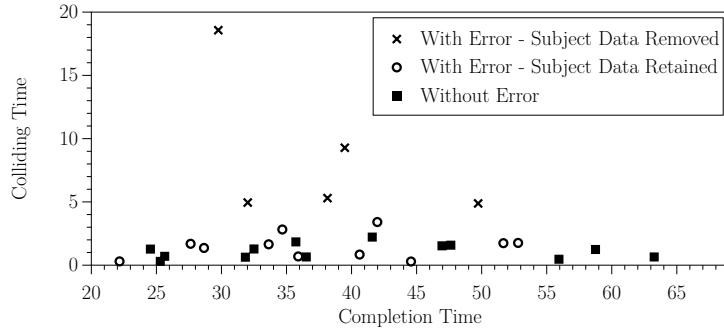


Figure 4.4: **Exclusion criterion for direction calibration error:** The average total time and collision time for each subject in Straight Hallway trials. X's indicate subjects determined as outliers whose data was replaced by data from a new subject experiencing the same WIP system after the direction calibration error was corrected.

Characteristic	Mean	Std. Dev.	
Age	23.0	8.5	
Characteristic	Male	Female	
Gender	18	6	
Characteristic	25 <sup>th</sup> Perc.	50 <sup>th</sup> Perc.	75 <sup>th</sup> Perc.
Video Game Time	5–10 hrs	10–20 hrs	20–40 hrs
Total VE Experience	0 times	0 times	2 times
Most Recent VE Experience	never	never	>1 year ago

Table 4.1: **Subjects – Descriptive statistics:** The statistics for the subjects in my user study as gathered in the questionnaire.

- Colliding Time:  $F(1) = 1.529$ ,  $p = 0.229$

**Between-interface subject characteristics:** Subjects were randomly assigned to one of the three WIP systems. No significant between-interface subject difference was found on any questionnaire-measured characteristics (Table 4.1, ANOVA for age, Kruskal-Wallis for others).

- Age:  $F(2, 21) = 0.096$ ,  $p = 0.909$
- Gender:  $H(2) = 1.278$ ,  $p = 0.528$
- Video Game Time:  $H(2) = 2.415$ ,  $p = 0.299$
- Total VE Experience:  $H(2) = 1.432$ ,  $p = 0.489$
- Most Recent VE Experience:  $H(2) = 1.746$ ,  $p = 0.418$



## 4.2 Results

Herein, results are considered significant at  $p < 0.05$ . Trends are reported for  $p < 0.10$ . In all discussion, statistical scores follow the relevant paragraph as a bulleted list. Post hoc tests, where necessary, are indented under the relevant tests. All non-within-subject-contrasts post hoc tests required three tests, requiring Bonferroni corrections for significance ( $p < 0.0167$ ) and trends ( $p < 0.0333$ ). (Within-subjects contrasts require no correction (still at the 0.05 and 0.1 levels).) Effect sizes ( $\omega$  or  $r$ ) are reported in the bulleted lists for all statistically significant main results. All effect sizes were calculated using Field and Hole's suggested methods (Field and Hole, 2003).

This study resulted in four different data sets for analysis:

- Per-trial measures – Total Time, Collision Time, Stopping Error, and Normalized Total Time (ANOVA)
- Post-experiment Likert-scale WIP judgements – Questionnaire, Section II (Kruskal-Wallis)
- Post-experiment standard questionnaires – SSQ, Section I (ANOVA), and Presence, Section III (Kruskal-Wallis)
- Post-experiment free responses – Questionnaire, Section V, and exit interview responses (Chi-square)

The first two were analyzed directly with the tests indicated above. The third set comes from standard questionnaires with explicit methods for calculating scores. These scores were then analyzed with the indicated tests. To analyze the fourth type, I and fellow graduate student Luv Kohli identified five themes of recurring complaints (Balance Problems, Jerky Motions, Walk Direction Issues, No Short Steps, and Stepping-To-Speed Problems). While blind to the randomly-assigned WIP system (condition), we coded the comments according to these categories. All codings were Kohli's with some guidance from me. Examples of statements from each of these categories follow:

- **Balance Problems:** “It was sometimes difficult to keep my balance while walking in place” (Subject 204, LLCM-WIP).

- **Jerky Motions:** “The motion seemed jerky. Every time I took a step I would be jerked forward. In the real world the visuals are smooth” (Subject 205, LLCM-WIP).
- **Walk Direction Issues:** “It took me a while to ‘steer with my knees’. The feeling that the most important component of my direction of travel is which way my knees were facing more than which way my feet swung” (Subject 210, LLCM-WIP).
- **No Short Steps:** “I was unable to take steps of varying size. There were times when I wanted to take small steps to maneuver around sharp corners or to position myself accurately above the target; this did not work” (Subject 215, Dean GUD WIP).
- **Stepping-To-Speed Problems:** “I felt like taking smaller steps more rapidly should have moved me faster than taking long, slow steps” (Subject 207, LLCM-WIP). Another example: “It kind of seemed like I moved at the same pace no matter how quickly I moved my legs. . .” (Subject 203, Dean GUD WIP).

#### 4.2.1 Per-Trial Measures

In all of the following, Total Time, Collision Time, and Stopping Error statistics were calculated by a single repeated-measures ANOVA. Normalized Total Time statistics were calculated by a separate repeated-measures ANOVA because the Straight Hallway could not be included: Since it is the normalizing value, the Straight Hallway Normalized Total Time is always one. Two subjects’ per-trial data was incomplete due to system errors, and were not included in this analysis.

**WIP system:** No significant difference was found with WIP system on any of these metrics (Figure 4.5). There were no significant interaction effects including WIP system.

- Total Time:  $F(2, 19) = 0.072, p = 0.930$
- Collision Time:  $F(2, 19) = 0.88, p = 0.431$
- Stopping Error:  $F(2, 19) = 2.319, p = 0.126$
- Normalized Total Time:  $F(2, 19) = 0.69, p = 0.52$

I performed a power analysis to determine how likely statistically significant results would be for any of these measures. For the following power analysis results, the first two numbers are the Cohen’s  $f$  distribution score for the observed mean and standard deviations

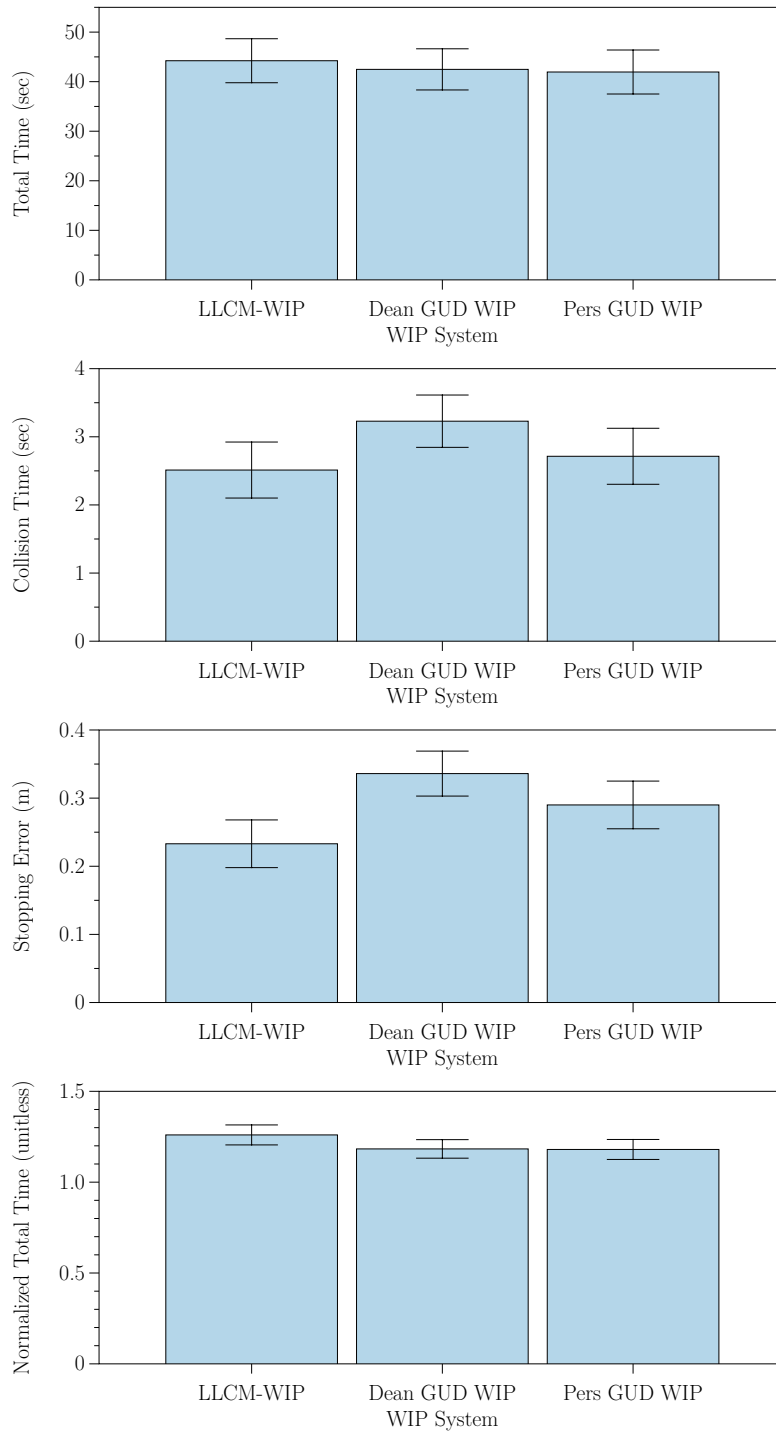


Figure 4.5: **Main effects by WIP system:** The not-significant means and standard errors among WIP systems as measured by Total Time, Collision Time, Stopping Error, and Normalized Total Time.

for eight subjects per condition, and the associated Type II error ( $\beta$ ) for this experiment. The third number (N) is the number of subjects per condition that would be required to find a statistically significant result (at  $\beta = 0.8$ ) assuming the means and standard deviations remained the same.

- Total Time: Cohen's  $f = 0.09$ ,  $\beta = 0.07$ ,  $N = 459$
- Collision Time: Cohen's  $f = 0.3$ ,  $\beta = 0.21$ ,  $N = 41$
- Stopping Error: Cohen's  $f = 0.49$ ,  $\beta = 0.5$ ,  $N = 15$
- Normalized Total Time: Cohen's  $f = 0.19$ ,  $\beta = 0.12$ ,  $N = 60$

**Training effect:** There was a significant training effect for Total Time, Collision Time, and Normalized Total Time; a trend for Stopping Error. Since the data failed the Mauchly sphericity test, I used the Greenhouse-Geisser values. Figure 4.6 shows these results with their associated statistically-significant fits: Subjects improved on all three non-normalized metrics with later sets, nearly stabilizing for Total Time at the end. The Normalized Total Time's initial increase is caused by within-first-set training. In the first set, the first hallway experienced is the Straight Hallway. Since subjects improved drastically within the first set, the first set's Normalized Total Time is less than one. During sets two and three, even though the subjects sped up overall, the time spent in the Acute, Obtuse, and Curved Hallways was approximately the same relative to the Straight Hallway time. By the final set, they were improving with respect to the Straight Hallway.

- Total Time:  $F(1.575, 29.922) = 49.897$ ,  $p < 0.001$ ,  $\omega = 0.73$ 
  - Within-Subjects Contrasts, Linear:  $F(1, 19) = 102.040$ ,  $p < 0.001$
  - Within-Subjects Contrasts, Quadratic:  $F(1, 19) = 13.416$ ,  $p = 0.002$
- Collision Time:  $F(2.032, 38.613) = 14.803$ ,  $p < 0.001$ ,  $\omega = 0.62$ 
  - Within-Subjects Contrasts, Linear:  $F(1, 19) = 25.449$ ,  $p < 0.001$
- Stopping Error:  $F(2.687, 51.062) = 2.228$ ,  $p = 0.095$ ,  $\omega = 0.23$ 
  - Within-Subjects Contrasts, Linear:  $F(1, 19) = 5.400$ ,  $p = 0.031$
- Normalized Total Time:  $F(2.308, 43.861) = 18.936$ ,  $p < 0.001$ ,  $\omega = 0.67$ 
  - Within-Subjects Contrasts, Linear:  $F(1, 19) = 20.211$ ,  $p < 0.001$
  - Within-Subjects Contrasts, Quadratic:  $F(1, 19) = 53.602$ ,  $p < 0.001$

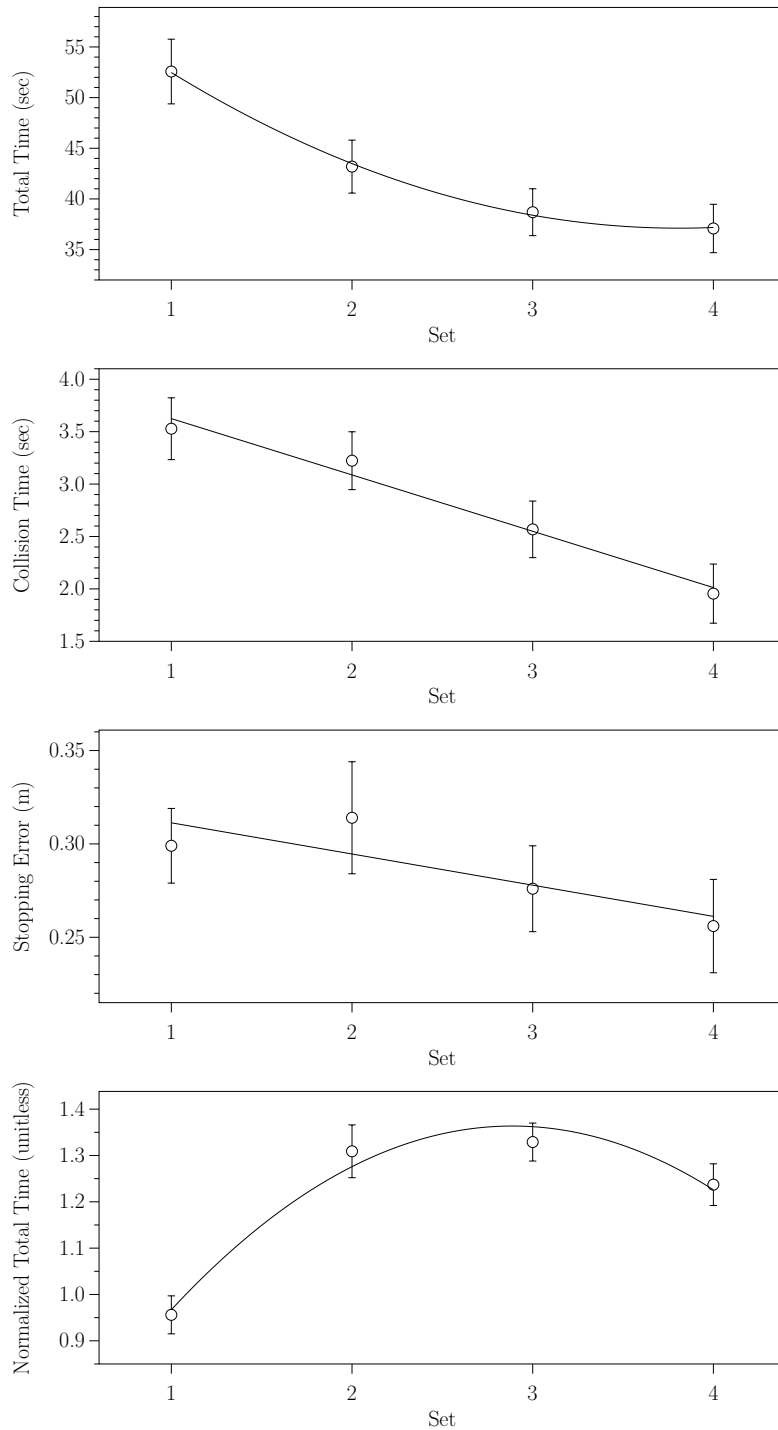


Figure 4.6: **Training effects:** The means and standard errors for Total Time, Collision Time, Stopping Error, and Normalized Total Time by set with statistically-significant fits for all WIP interfaces. Total Time, Collision Time, and Normalized Total Time show statistically significant training effects. Stopping Error shows a trend.

**Hallway:** There was a main effect for the hallways for Total Time, Collision Time, and Normalized Total Time, but no significance found for Stopping Error. Since the data failed the sphericity test, I used the Greenhouse-Geisser values. Figure 4.7 shows these results with their associated statistically-significant fits: Acute Hallways took longer than the rest; subjects' Collision Time increased by hallway in the following order: Straight, Curved, Obtuse, Acute; Stopping Error did not appear to be affected by hallway type.

- Total Time:  $F(2.667, 50.664) = 30.245$ ,  $p < 0.001$ ,  $\omega = 0.76$ 
  - Within-Subjects Contrasts, Linear:  $F(1, 19) = 57.64$ ,  $p < 0.001$
  - Within-Subjects Contrasts, Quadratic:  $F(1, 19) = 9.187$ ,  $p = 0.007$
  - Within-Subjects Contrasts, Cubic:  $F(1, 19) = 5.084$ ,  $p = 0.036$
- Collision Time =  $F(2.597, 49.345) = 31.995$ ,  $p < 0.001$ ,  $\omega = 0.76$ 
  - Within-Subjects Contrasts, Linear:  $F(1, 19) = 78.103$ ,  $p < 0.001$
- Stopping Error =  $F(2.805, 53.299) = 0.117$ ,  $p = 0.942$
- Normalized Total Time:  $F(1.787) = 40.510$ ,  $p < 0.001$ ,  $\omega = 0.80$ 
  - Within-Subjects Contrasts, Linear:  $F(1, 19) = 56.906$ ,  $p < 0.001$
  - Within-Subjects Contrasts, Quadratic:  $F(1, 19) = 13.181$ ,  $p = 0.002$

**Interaction - Training by hallway:** There was an interaction between the set number (training) and the hallways for Total Time (Figure 4.8) and Collision Time (Figure 4.9). Since the data failed the sphericity test, I used the Greenhouse-Geisser values. The Total Time interaction is almost certainly caused by the crossover for Straight Hallways between the first and second sets. As previously mentioned, this is caused by within-set training in the first set: Since subjects always experienced the Straight Hallway first, they improved by the time they experienced the Curved and Obtuse Hallways within the same set. The differences in Collision Time by Hallway probably refers to the difference in when different hallways have derivative changes. Thus, neither is of important practical significance.

- Total Time:  $F(4.349, 82.628) = 9.021$ ,  $p < 0.001$ ,  $\omega = 0.52$
- Collision Time:  $F(4.159, 79.02) = 2.624$ ,  $p = 0.039$ ,  $\omega = 0.26$

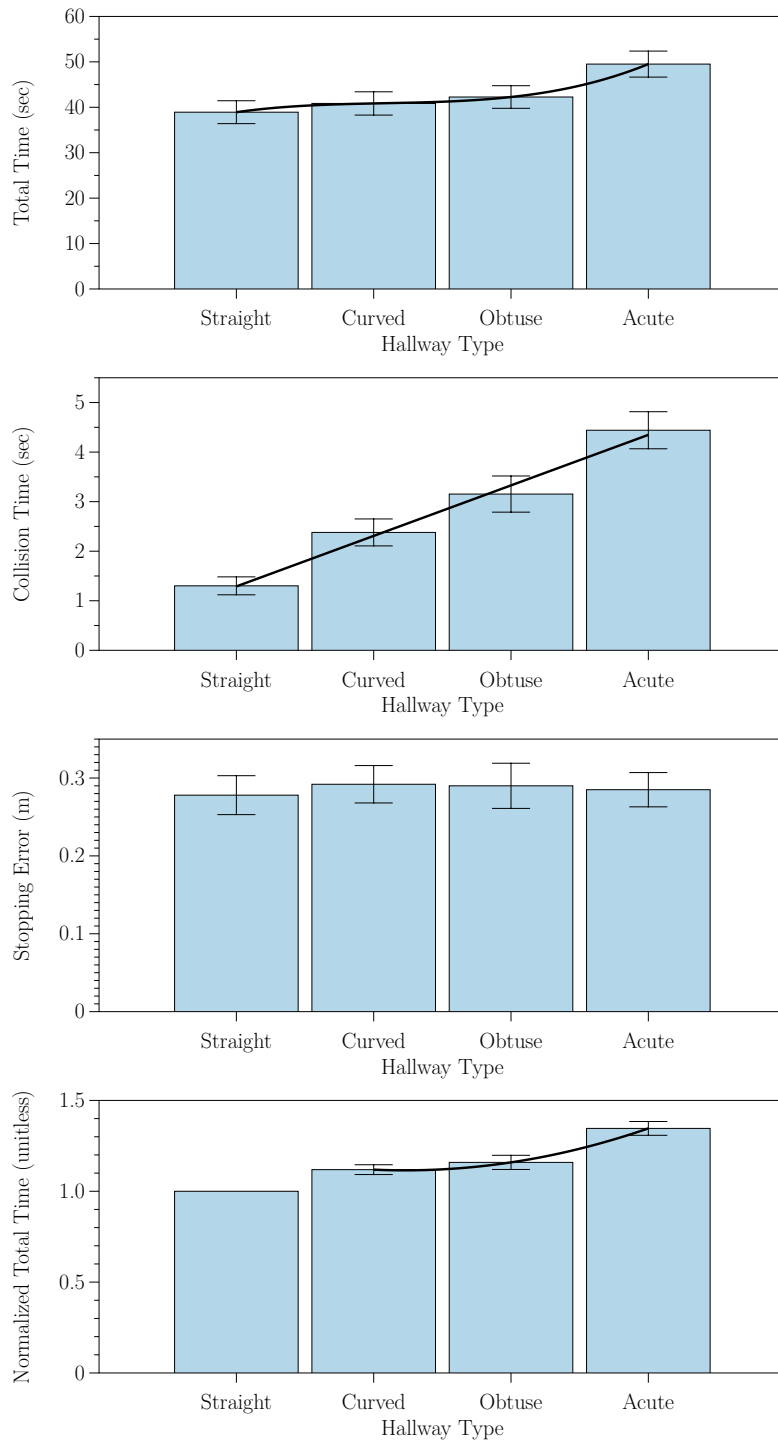


Figure 4.7: **Effect of hallway type:** The means and standard errors for Total Time, Collision Time, Stopping Error, and Normalized Total Time by hallway for all WIP interfaces. Total Time, Collision Time, and Normalized Total Time have a main effect by hallway and are shown with their statistically significant fits. Stopping Error has none.

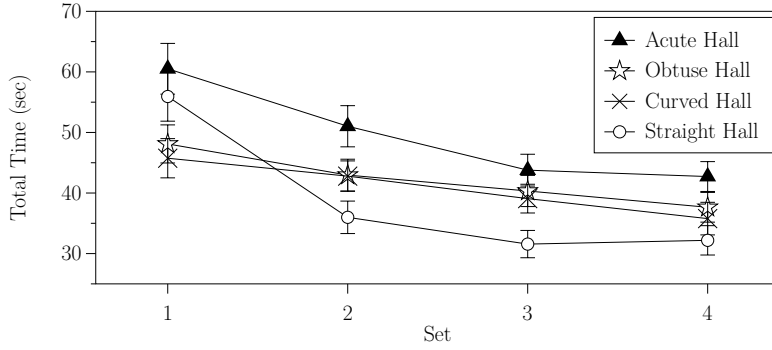


Figure 4.8: **Total Time interaction – training and hallway:** Interaction between training effect (X axis) and hallway effect (different lines) for Total Time. Mean and standard error shown for each datapoint. The statistically significant interaction is almost certainly explained by the cross-over interaction for the Straight Hallway with the Curved and Obtuse Hallways between the first and second sets.

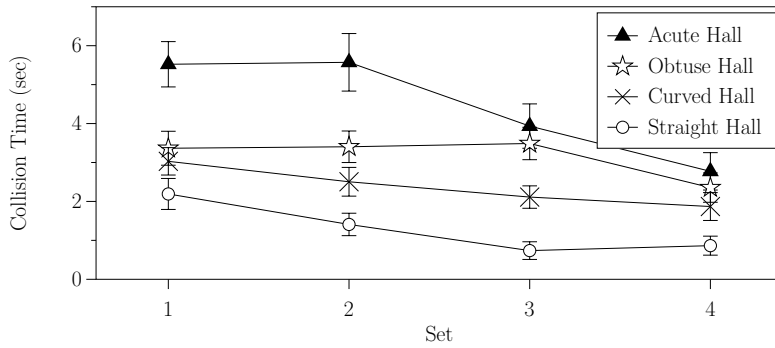


Figure 4.9: **Collision Time interaction – training and hallway:** Interaction between set number (X axis) and hallway (different lines) for Collision Time. Mean and standard error shown for each datapoint. The statistically significant interaction probably refers to the difference in when different hallways have derivative changes.

#### 4.2.2 Simulator Sickness

I attribute simulator sickness in WIP systems to sensory conflict: The eyes report motion that the inner ear does not. Because the visual motion characteristics of the Dean GUD WIP and Personalized GUD WIP systems are nearly identical, for this analysis, I grouped the results of both of these systems and compared them to LLCM-WIP.

Since I hypothesized LLCM-WIP would cause worse nausea than the two GUD WIP systems, I used a one-tailed ANOVA of the four simulator sickness scores. The ANOVA demonstrates a significant effect for Nausea (Figure 4.10), but no significant effect for Occulomotor, Disorientation, or overall sickness. Because jerky visual motions can cause sickness,



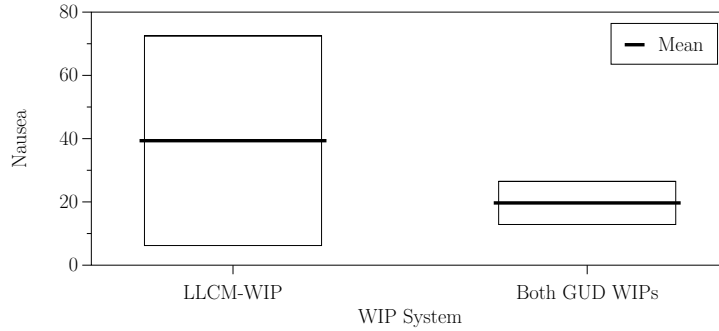


Figure 4.10: **Simulator sickness by WIP system:** The means (thick horizontal line) and 95% confidence intervals for the mean (boxes) of LLCM-WIP and combined GUD WIP systems. This difference is statistically significant: LLCM-WIP can cause worse nausea than GUD WIP. Kevin Arthur reports that five previous studies had mean nausea scores between 7 and 27 (Arthur, 2000).

System	Complained	Balance	Jerkiness
LLCM-WIP	Yes	2	2
LLCM-WIP	No	6	6
GUD WIP (both)	Yes	0	0
GUD WIP (both)	No	16	16
$\chi^2$ Significance		0.029	0.029

Table 4.2: **Complaint frequency – balance and jerkiness:** Number of subjects who complained during free response periods about balance problems and jerky visuals split by LLCM-WIP and combined GUD WIP systems. Statistical significance of each result shown on the bottom line.

and lack of balance can be a symptom of simulator sickness, I ran separate Chi-squared tests for the free responses for these categories. Both cases are significant (Table 4.2).

- Nausea:  $F(1, 22) = 1.838$ ,  $p = 0.04$ ,  $\omega = 0.31$
- Occulomotor:  $F(1, 22) = 0.938$ ,  $p = 0.18$
- Disorientation:  $F(1, 22) = 0.567$ ,  $p = 0.29$
- Overall sickness:  $F(1, 22) = 0.988$ ,  $p = 0.17$
- Jerky visuals:  $\chi^2(1) = 4.771$ ,  $p = 0.029$ ,  $r = 0.37$
- Balance problems:  $\chi^2(1) = 4.771$ ,  $p = 0.029$ ,  $r = 0.37$

### 4.2.3 Presence

The SUS presence questionnaire contains six questions on a seven-point Likert scale. A presence score is the sum of all responses over a predetermined threshold. I counted scores

Abbreviation	Full Question
Curved Halls Easy	Curved paths were easy to complete.
As Expected	The technique worked as I expected.
Moved ...	The technique moved me... (Too fast, Just right, Too slow).
Started Quickly	The technique started moving when I wanted it to.
Stopped Quickly	The technique stopped moving when I wanted it to.
Acute Halls Easy	Sharp turns were easy to complete.
Obtuse Halls Easy	Soft turns were easy to complete.

Table 4.3: **Full questions to abbreviations:** Figure 4.11 uses these shortened versions of the longer questions listed here.

of five or greater. A Kruskal-Wallis test found no significant difference in reported presence scores by WIP interface.

- Presence:  $H(2) = 0.143$ ,  $p = 0.931$

#### 4.2.4 Self-Reported Usability

**Likert scores:** The subjects answered several questions about the WIP system on a seven-point Likert scale (Questionnaire, Section II, Table 4.3, Figure 4.11). A Kruskal-Wallis test of all responses demonstrated the following:

- **Significant:** Subjects reported that they moved too slowly in LLCM-WIP and closer to their intended speed in both GUD WIP systems. Post hoc tests only find a statistically significant difference between Personalized GUD WIP and LLCM-WIP (Figure 4.11, “Moved ...”).
  - Moved...:  $H(2) = 7.228$ ,  $p = 0.027$ ,  $r = 0.39$ 
    - LLCM-WIP to Dean GUD WIP:  $H(1) = 3.392$ ,  $p = 0.066$
    - LLCM-WIP to Pers GUD WIP:  $H(1) = 6.729$ ,  $p = 0.009$
    - Dean GUD WIP to Pers GUD WIP:  $H(1) = 0.515$ ,  $p = 0.473$
- **Significant:** GUD WIP subjects felt curved hallways were easier to travel than LLCM-WIP subjects (Figure 4.11, “Curved Halls Easy”).
  - Curved halls easy:  $H(2) = 11.955$ ,  $p = 0.003$ ,  $r = 0.56$ 
    - LLCM-WIP to Dean GUD WIP:  $H(1) = 8.597$ ,  $p = 0.003$
    - LLCM-WIP to Pers GUD WIP:  $H(1) = 6.333$ ,  $p = 0.012$
    - Dean GUD WIP to Pers GUD WIP:  $H(1) = 3.589$ ,  $p = 0.058$

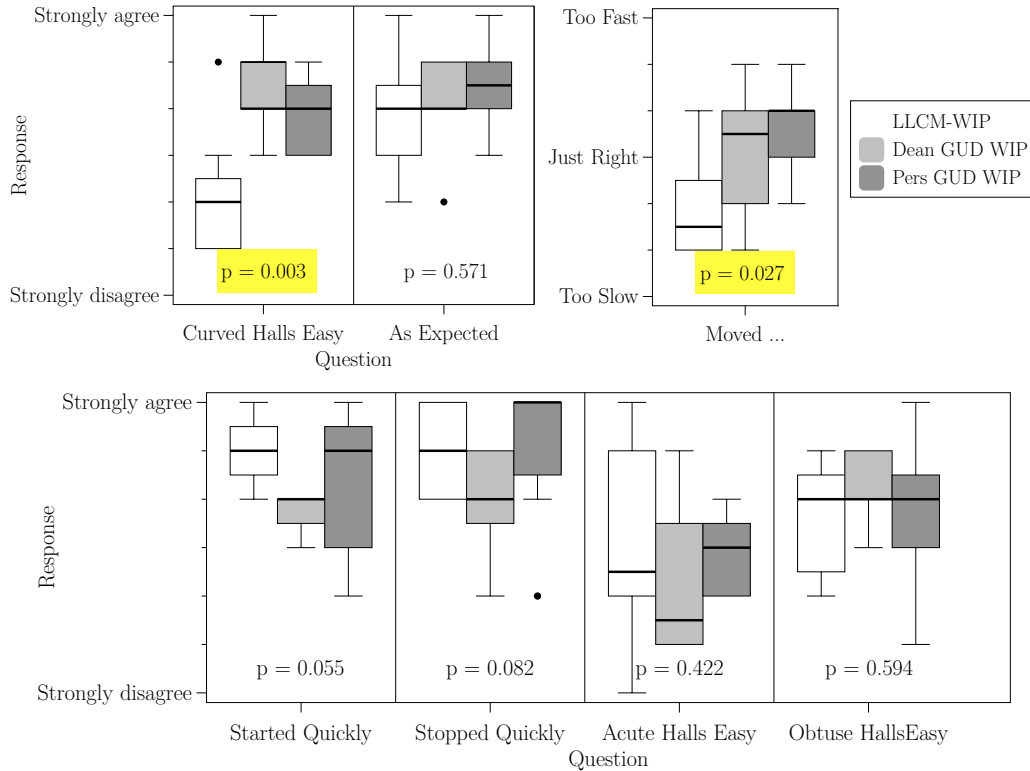


Figure 4.11: **Questionnaire responses:** Box plots of all subjects’ WIP judgements questionnaire responses (Section II). LLCM-WIP scores are the white boxes, Dean GUD WIP are the medium gray boxes, and Personalized GUD WIP are the darker boxes. Statistical significance is shown below each grouping – significant results are highlighted.

- **Trend:** LLCM-WIP and Personalized GUD WIP subjects felt starting and stopping latency was somewhat better than Dean GUD WIP (Figure 4.11, “Started Quickly”). Post hoc tests indicate this to be a strong difference only between LLCM-WIP and Dean GUD WIP for starting latency.

- Started quickly:  $H(2) = 5.794$ ,  $p = 0.055$ ,  $r = 0.33$ 
  - LLCM-WIP to Dean GUD WIP:  $H(1) = 8.538$ ,  $p = 0.003$
  - LLCM-WIP to Pers GUD WIP:  $H(1) = 0.441$ ,  $p = 0.507$
  - Dean GUD WIP to Pers GUD WIP:  $H(1) = 0.988$ ,  $p = 0.320$
- Stopped quickly:  $H(2) = 4.999$ ,  $p = 0.082$ ,  $r = 0.28$ 
  - LLCM-WIP to Dean GUD WIP:  $H(1) = 2.902$ ,  $p = 0.088$
  - LLCM-WIP to Pers GUD WIP:  $H(1) = 0.464$ ,  $p = 0.496$
  - Dean GUD WIP to Pers GUD WIP:  $H(1) = 4.017$ ,  $p = 0.045$

System	Complained	No Short Steps	Bad Step-to-Speed	Direction
LLCM-WIP	Yes	0	4	5
LLCM-WIP	No	8	4	3
Dean GUD WIP	Yes	4	6	4
Dean GUD WIP	No	4	2	4
Pers GUD WIP	Yes	5	0	6
Pers GUD WIP	No	3	8	2
$\chi^2$ Significance		0.006	0.002	0.582

Table 4.4: **Complaint frequency – short steps, step-to-speed function, direction:** Number of subjects who complained during free response periods about inability to take short steps, a poor step-to-speed function, and direction problems – split by LLCM-WIP and Dean GUD WIP, and Personalized GUD WIP systems. Each result’s Chi-square test’s statistical significance shown on the bottom line.

- No significant differences were found for Acute and Obtuse Hallway easiness, or if the system worked as expected.
  - Acute halls easy:  $H(2) = 1.726$ ,  $p = 0.422$
  - Obtuse halls easy:  $H(2) = 1.041$ ,  $p = 0.594$
  - As expected:  $H(2) = 1.121$ ,  $p = 0.571$

**Free response:** Free response statements showed that GUD WIP’s subjects wanted the ability to take small steps. This was significant for Personalized GUD WIP, a trend for Dean GUD WIP. Both LLCM-WIP and Dean GUD WIP subjects complained more about problems in how their in-place steps translated to forward speed, whereas Personalized GUD WIP subjects did not. This difference was significant for Dean GUD WIP, a trend for LLCM-WIP. All WIP systems contained subjects who had difficulty with shin-directed walking (no significant difference).

- Small steps:  $\chi^2(2) = 10.080$ ,  $p = 0.006$ ,  $r = 0.40$ 
  - LLCM-WIP to Dean GUD WIP:  $\chi^2(1) = 5.333$ ,  $p = 0.021$
  - LLCM-WIP to Pers GUD WIP:  $\chi^2(1) = 7.273$ ,  $p = 0.007$
  - Dean GUD WIP to Pers GUD WIP:  $\chi^2(1) = 0.254$ ,  $p = 0.614$
- In-place steps to speed:  $\chi^2(2) = 12.514$ ,  $p = 0.002$ ,  $r = 0.49$ 
  - LLCM-WIP to Dean GUD WIP:  $\chi^2(1) = 1.067$ ,  $p = 0.302$
  - LLCM-WIP to Pers GUD WIP:  $\chi^2(1) = 5.333$ ,  $p = 0.021$
  - Dean GUD WIP to Pers GUD WIP:  $\chi^2(1) = 9.600$ ,  $p = 0.002$

- Shin direction problems:  $\chi^2(2) = 1.082$ ,  $p = 0.582$

## 4.3 Discussion

### 4.3.1 Walk Speed, Accuracy, and Stopping

The lack of significant WIP-system-caused difference in Total Time, Collision Time, and Stopping Error – while somewhat disappointing – is understandable. The LLCM-WIP multiplier was chosen to set LLCM-WIP’s average speed to approximately the same as the output GUD WIP speed, leading to highly similar Total Times.

Between-subject variability likely increased because subjects received no WIP-system-specific training on how their assigned WIP system interpreted their in-place steps. This was intentional: I wanted to know what users automatically do when taking in-place steps, not what they can be trained to do.

Furthermore, I instructed subjects to attempt to minimize Total Time, Collision Time, and Stopping Error and did not tell them what a “good” score on any of them would be. When people really walk, they would weight these metrics differently. Merging these measures with arbitrary weightiness would create an artificial relationship among these measures that would lead subjects to skew their behaviors away from naturalness.

The associated power analysis provided the following insights. Stopping Error is the most powerful difference, requiring only 15 subjects per walk interface to find a significant difference with 80% probability ( $\beta$ ). The other per-trial measures in decreasing power order are Collision Time, Normalized Total Time, and Total Time. Assuming this test were rerun with more subjects, the only measure I believe likely to provide an ecologically significant result would be Stopping Error. The currently-not-significant difference for Stopping Error – if made significant by more subjects – would demonstrate that LLCM-WIP performs better by this metric than the GUD WIP systems. I believe this is caused by the lack of “small steps” in the GUD WIP systems.

### 4.3.2 Hallways

The four hallways were designed to test different things. The Straight Hallway measures speed and accuracy when little direction control is required. The Curved Hallway measures how well small changes in direction can be performed during continuous walking. The Obtuse and Acute Hallways are two levels of the same metric – measuring how well users can travel to a location, turn, then continue their travel.

The results confirm that these hallways served their purposes: The Straight Hallway was the fastest and most accurate – even considering that it was always the first hallway experienced. The Curved Hallway was generally the second fastest, and second most accurate – permitting users to walk quickly and accurately while making continual minor adjustments in direction. The Acute Hallway was harder than the Obtuse.

Since all hallways ended identically in a resting room with a target, it makes sense that there would be no statistical difference by hallway for Stopping Error.

### 4.3.3 Training

Training effects appear to be inherent in Walking-In-Place systems. From the results shown in Figure 4.6, it appears that subjects improve drastically on walking speed first: Total Time decreases between sets one and two, Normalized Total Time decreases within set one (set one has an average score below 1). Total Time appears to stabilize over by Set 4. Accuracy appears to improve throughout: Collision Time and Stopping Error decrease among sets two, three, and four; and since Normalized Total Time decreases between sets three and four, they walked closer to Straight Hallway speeds in accuracy-requiring hallways. Since none of these latter scores have stabilized between the third and fourth sets, I think it likely that subjects' accuracy could improve further before stabilizing.

### 4.3.4 Simulator Sickness

Some users are much more prone to simulator sickness than others. Hence the wide variation of responses for LLCM-WIP users on the Nausea score: Although not measured, it is possible that those prone to visual-motion-induced nausea felt high nausea after LLCM-

WIP; those not prone felt almost none.

LLCM-WIP's sensory conflicts are the opposite of those on a boat. When one is visually enclosed within a boat, visuals remain relatively stable (one's head moves much as the walls around him), but one's inner ear reports rocking. When employing LLCM-WIP, one's inner ear reports relatively stable positions, but visuals report high-frequency, high-magnitude forward velocity changes.

Since three separate measures (SSQ's Nausea score, free-response jerkiness, and free-response balance problems) all indicate that LLCM-WIP causes more nausea than GUD WIP, this is an important result. Any widely-used system will be used by some nausea-prone users: Developers will want to avoid sickening a significant portion of their users.

#### **4.3.5 Direction from shins**

The direction-from-shin complaints for all WIP systems indicate that further research on specifying WIP system direction is required. Several subjects commented that they were accustomed to gaze-directed motion from their video game experience. However, since the task involved in this study did not require exploring their environment, subjects had no reason to want to look elsewhere than their walking direction. Based on that, I expect that in a wayfinding or exploring task, gaze-directed WIP systems would receive more complaints and fare poorer in the task.

#### **4.3.6 Short Steps**

This was the only metric on which LLCM-WIP was statistically better than GUD WIP (Table 4.4). While watching subjects, I noted that when they wanted to move a small distance, subjects took very small, quick in-place steps. In GUD WIP, these are high-frequency steps – leading to very quick forward speeds.

Short steps are particularly important during precise motions. I believe that LLCM-WIP's better (albeit, not statistically significantly better) Collision Time and Stopping Error scores are caused by this difference in ability to take small steps. In fact, I believe that adding some indication of head position (such as a small head-position shadow) would

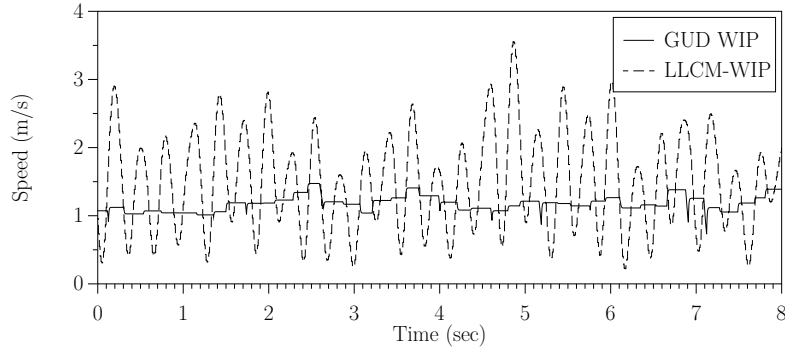


Figure 4.12: **Instantaneous speed comparison:** Given the same inputs over a period of eight seconds, this subject (if experiencing LLCM-WIP) would experience his speed change from around 0.5m/s to 3m/s multiple times per second. A GUD WIP subject would experience a relatively consistent output speed.

make the Stopping Error difference larger: As currently implemented, I believe LLCM-WIP will fare better than GUD WIP on this metric.

In Chapter 5, I discuss improvements to GUD WIP that I believe would reduce this difference.

### 4.3.7 Average Speed

LLCM-WIP subjects rated the system as moving slower than they wanted it to move. Figure 4.5 shows the not-significant differences between WIP interface for Total Time. Subsection 4.3.1's power analysis demonstrated that any visible differences are very weak.

Therefore, between WIP interface, the speeds were much the same. However, LLCM-WIP's subjects felt they moved too slowly. I suggest that this disparity may be caused by the large fluctuations in LLCM-WIP output speed. Figure 4.12 demonstrates how frequently LLCM-WIP speeds vary – on the order of two cycles per second. I know of no real-world equivalent where forward motion continually changes so drastically so quickly. I believe that one's perception is not able to estimate overall walk speed when presented with such fluctuations. Therefore, the subjects self-rated their speed as less accurate. I do not know why they felt it was slower instead of faster.



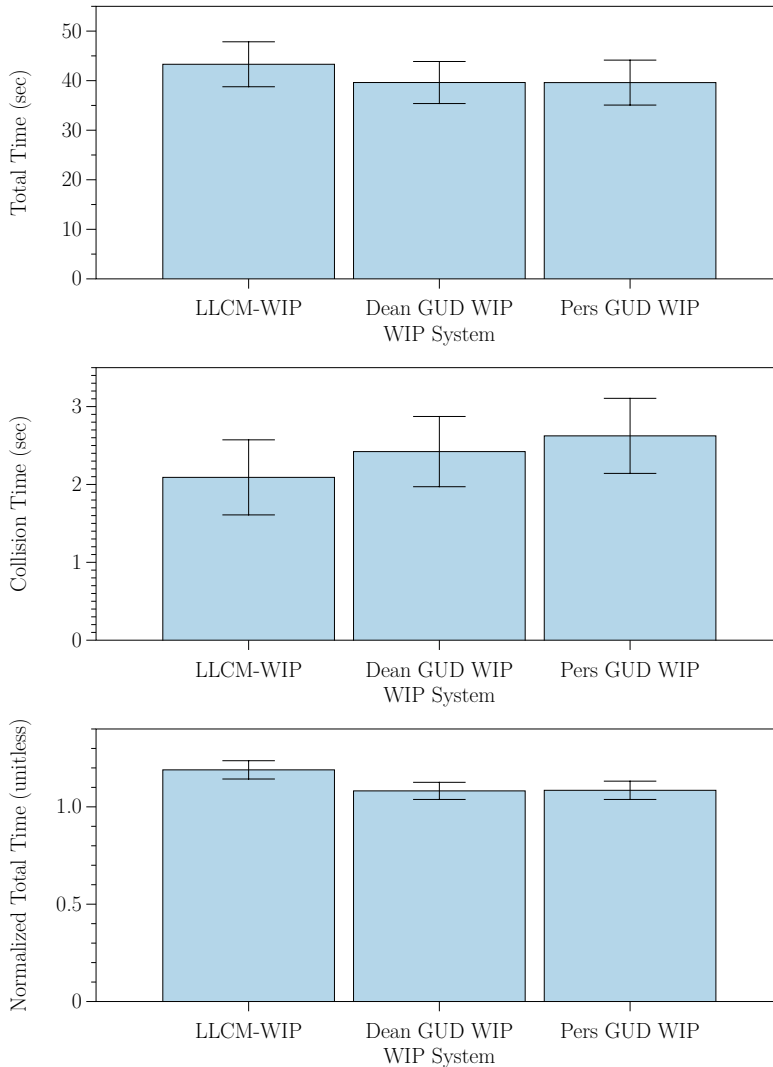


Figure 4.13: **Curved hallway results:** The not-statistically-significant differences for subjects on the Curved Hallways by system (means and standard errors shown).

### 4.3.8 Curved Paths

On the questionnaire, both Dean and Personalized GUD WIP subjects self rated their performance on the Curved Hallways as better than LLCM WIP subjects. Figure 4.13 shows the not-significant differences between the three WIP systems on the Curved Hallways for Total Time, Collision Time, and Normalized Total Time. It shows that GUD WIP subjects both went slightly faster (Total Time) and slightly closer to their Straight Hallway speed (Normalized Total Time). However, they also collided more during that period.

Unlike the Acute and Obtuse Hallways, the Curved Hallway permitted subjects to walk

at a continual (or rhythmic) pace throughout the trial: In the angled hallways, subjects generally stopped at the corners. Unlike the Straight Hallway, the Curved Hallway required subjects to continually make fine adjustments to their walk direction. I suggest that GUD WIP users felt they did very well in Curved Hallways (although the numbers do not suggest any real improvement) because the GUD WIP model measures step frequency and outputs a rhythmic walk speed: In the Curved Hallway, subjects had a feel for what speed they meant to be going, what speed they were going, and felt they could make the continual adjustments necessary to travel that path.

#### **4.3.9 In-Place-Stepping-to-Speed Conversion**

During free response periods, half of LLCM-WIP subjects complained about the way their in-place steps were converted to forward speed; three quarters of Dean GUD WIP subjects complained; no Personalized GUD WIP subjects complained.

The complaints were of three types: Some LLCM-WIP subjects complained that larger, low-frequency steps led to faster speed than smaller, high-frequency steps; some Dean GUD WIP subjects complained that step height didn't appear to affect walk speed; and some Dean GUD WIP subjects complained about not being able to properly control their speeds. The first two types of complaints conflict: This may mean that some users think in terms of step length while others think in terms of step frequency.

However, the most interesting fact is that these complaints are completely missing when subjects experienced Personalized GUD WIP – a Walking-In-Place system that has been tuned to their personal step-frequency-to-walk-speed patterns. I believe that this is the result of extensive walking experience: People are familiar with how their steps translate into forward speed, and when a Walking-In-Place system matches their personal experience, they don't complain.

#### **4.3.10 Forward Walk Model and User Variations**

When measuring personal step-frequency-to-walk-speed functions, the Expedited Method worked very well for most subjects in this study. However, it did not work at all for

one subject. She was unable to complete the study since her step-frequency-to-walk-speed function could not be measured – her head didn't bob. Some people are trained to minimize their head bobs while walking: Finishing schools place books on young women's heads to train them to reduce bobbing. While I believe the Forward Walk Model is accurate for the majority of healthy people, it is not so for all.

#### **4.3.11 Step-Frequency-to-Walk-Speed Function Curvature**

The real walking curves from GUD WIP's validation analysis (Section 2.5, Figure 2.16) contained both positive and negative curvatures. I hypothesized that this was caused by requiring the subjects to walk at specified step frequencies – both higher and lower than the step frequency than the subject would normally choose. In this study, I asked the users to walk at their own defined slow, medium, and fast speeds. The best-fit quadratic fits from the 24 subjects were 23 positive curvatures ( $d^2y/dx^2 = 0.28\text{--}1.32$ ), and one nearly linear ( $d^2y/dx^2 = 0.02$ ). These are more in-line with what the biomechanics literature suggests.



## CHAPTER 5

# Conclusion

This dissertation has demonstrated the truth of the following thesis: *Walking-In-Place systems are measurably improved by using real-walking-based models.*

- Chapter 2 introduces a model called GUD WIP for converting in-place steps to forward walking speeds. This model introduces the in-place-stepping gait cycle, measures step frequency, and proposes that step frequency is a sufficient parameter to generate accurate virtual walking speeds.
- Chapter 3 introduces a model called the Forward Walking Model that describes how the head moves while one walks forward to extract personalized walking patterns from head track data alone. The chapter includes an analysis that demonstrates that the extracted parameters – step frequency, walk speed, and direction – are approximately as accurate as values derived from manually-tagged data.
- These two models have been shown to improve Walking-In-Place systems along the following metrics:
  - Like real walking’s speeds, GUD WIP’s output speeds are more within-step consistent than the outputs from previous systems (Figures 2.1, 2.2 and 4.12).
  - Like real walking, GUD WIP’s step-frequency-to-walk-speed is more consistent than LLCM-WIP’s (Subsection 2.5.2, Table 2.1).
  - GUD WIP causes less nausea than LLCM-WIP (Subsection 4.3.4, Figure 4.10, Table 4.2).
  - Users rate GUD WIP’s resulting speed as more accurate than LLCM-WIP (Subsection 4.3.7, Figure 4.11).
  - Users rate curved paths as easier in GUD WIP than LLCM-WIP (Subsection 4.3.8, Figure 4.11).

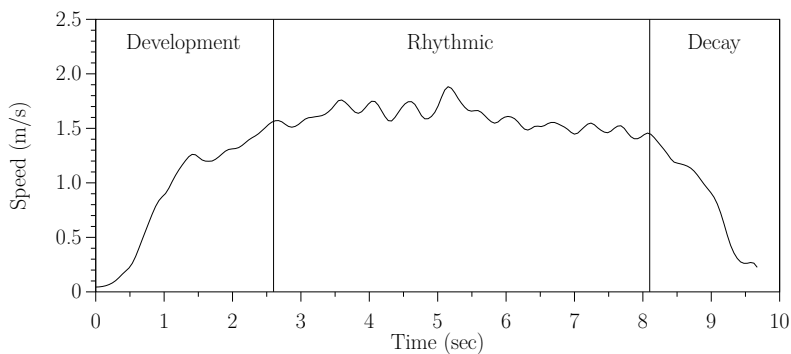


Figure 5.1: **The walking phases:** Development, rhythmic, and decay phases shown against a smoothed plot of walking speed for a straight-line path. This dissertation focuses mainly on improving Walking-In-Place systems for rhythmic speeds. Little is known about the characteristics of development and decay.

- Users complain less about the way steps translate to speeds when using Personalized GUD WIP than Dean GUD WIP or LLCM-WIP: When the relationship between in-place step frequency and forward speed matches their real walking relationship, users are happier (Subsection 4.3.9, Table 4.4).

Moreover, LLCM-WIP’s one statistically significant win over GUD WIP (complaints about lack of small steps in GUD WIP, Subsection 4.3.6, Table 4.4) actually reinforces this thesis: Dean and Personalized GUD WIP both lack a real-walking-based model for starting and stopping: Subjects’ complaints note its absence. GUD WIP measures in-place step frequency and outputs the intended *rhythmic* walking speed. However, when beginning and ending a path, one does not walk at rhythmic speeds, but at slower accelerating *development* and decelerating *decay* speeds (Figure 5.1). The subject-wished-for small steps appear most often when beginning motion, ending motion, or maneuvering to avoid obstacles – at the precise moments that development and decay speeds would be active.

Unfortunately, walk speed’s mathematical characteristics during development, decay, and maneuvering periods are little known. When beginning this dissertation, I hoped that I would derive a mathematical representation for these elements and measure their effect on Walking-In-Place systems. However, the models developed within this dissertation are required prerequisites for such a system. The missing models for development and decay phases are left as future work.

## 5.1 Future Work

The work presented herein opens several areas of potential further research.

### 5.1.1 GUD WIP

**Development and decay:** GUD WIP estimates rhythmic walking speeds. However, walking includes slower development and decay phases at the beginning and end of any path (Figure 5.1). The usability study illustrated that WIP-system users want the ability to take smaller steps (Subsection 4.3.5): Adding development and decay models would satisfy most of their desires.

Although the biomechanics literature names these phases, I have found no mathematical definitions that specify the precise transition moments for these phases. Moreover, the characteristics of the development and decay phases are unknown: How many steps does each phase require? Do different step frequencies or desired walk speeds alter the duration of development and decay phases? What mathematical function would best fit these properties?

Another inherent problem impedes adoption of decay-phase characteristics: Whereas it is easy to identify the first steps in a path as they occur, I know of no way to identify the last steps in a path as they occur. Lee suggests that  $\tau$  – the *time-to-collision* metric – dictates slowing behaviors (Lee, 1980). Whereas this metric could aid in any path that includes stopping before hitting a wall or stopping at a known target position, the situation where a user wishes to stop accurately at a user-chosen location in an open space remains: There would be no system-knowable position to measure  $\tau$  for.

**Non-kinematic measurements:** GUD WIP’s implementation for identifying in-place gait events (Subsection 2.4.1) uses stepping kinematics – tracker-measured body positions and motions. Using kinematics led to two important tradeoffs in the implementation. First, to permit local maneuvering and turning while still measuring forward-motion-causing steps, the user’s foot must exceed a minimum height and speed to be identified as a forward-motion-causing step. Second, there is no way to differentiate between-step double support periods from after-last-step double support phases – leading to approximately a half second’s

stopping latency. Could other measurements remove either or both of these problems? Two possibilities are kinematic measurements through electromyography (EMG, leg muscle activations), and electroencephalographic (EEG) brain measurements. The HMD's magnetic fluctuations might disrupt the EEG measurements.

**Running-In-Place:** GUD WIP is a *Walking-In-Place* system. Thus, the gait cycle implementation described in Subsection 2.4.2 outputs zero step frequency when in-place running steps are identified. While not a significant problem with the current GUD WIP system, during the user study described in Chapter 4, I did notice one subject who would occasionally build to in-place running steps and find that his forward motion suddenly stopped.

There are two problems that must be solved for a GUD-WIP-like Running-In-Place system. First, while step length and step frequency are approximately linearly correlated during normal walking – permitting WIP systems to measure only step frequency – I do not know that this relationship continues while running. Second, the relative difficulty of taking in-place walking steps and real walking steps is different from the relative difficulty of taking in-place running steps and real running steps: It is much harder to maintain in-place running steps. I believe overcoming both of these problems could be a complete dissertation.

**WIP direction:** This dissertation doesn't address virtual walk directions, but uses the average of the user's shins' direction as direction. As discussed in Subsection 4.3.5, subjects complained about specifying their direction by the direction of their shins in all WIP systems. Arechavaleta found that shoulder direction best matched walk direction in real walking (Arechavaleta et al., 2006). It may be worth moving the forward-direction-specifying trackers to the shoulders.

A separate, but related, issue is how to specify non-forward walk directions. Several subjects asked how to take backward steps. It would be interesting to attempt Gaiter-like leg gestures to specify direction within GUD WIP (Templeman et al., 1999).

**Low-cost WIP:** My GUD WIP implementation requires costly, accurate shin tracking. Adapting GUD WIP to cheaper technology such as the Wii Fit™ or pressure sensors could decrease cost and increase usage. Three problems arise: First, is there a simple way to keep



the user over the correct pressure sensors while the HMD blocks the user’s view of where he is standing? Second, how does the user specify forward direction in such a system? The pressure sensors only measure if they are being depressed, not what direction the body is facing or which foot is depressing it. Finally, without foot position measurements, how would maximum foot height be identified? How could starting latency of the first step be decreased?

### 5.1.2 Forward Walk Model

**Models for in-place steps:** At present, GUD WIP requires costly, accurate shin tracking. Two adaptations of the Forward Walk Model could eliminate these expensive trackers:

- I believe an in-place-stepping head model – similar to the Forward Walking Model – could estimate in-place step frequency from head-track data alone. Unlike Slater’s head-based Virtual Treadmill (Slater and Usoh, 1994), this would estimate more than “stepping identified”.
- An in-place-stepping pressure-plate model could estimate in-place step frequency without any position tracking for tracker-free CAVEs<sup>TM</sup>.

**Analyzing real walking:** The Forward Walk Model is an inexpensive method if you have a large head-tracked space. However, it is not the simplest tool for measuring personal step-frequency-to-walk-speed functions. This was not the Forward Walk Model’s initial purpose. For a simpler personal step-frequency-to-walk-speed device, I recommend a GAITRite Portable Walkway System. It provides a 2-dimensional array of pressure sensors along a four-meter-long path (GAITRite, 2010). The GAITRite System can measure the timing and distance between footfalls – sufficient information to measure step frequency and walk speed to personalize GUD WIP systems.

The Forward Walk Model was originally intended as the first stage in analyzing real walking head-track logs to derive the mathematical model for head motion during development and decay phases, etc. Separating the head bobbing from the smoothed speed and direction enables identifying smaller changes in forward and angular velocity. Furthermore, step frequency, step length, and step timings are each important information when creating

such a model. Future research should use the Forward Walk Model for its intended purpose.

**Improving approximation methods:** The approximation methods proposed with the Forward Walk Model work rather well. However, several components could be improved:

- The model employs sine curves to approximate head bobbing components. However, head bobs are not precisely sine curves. I hypothesize that the difference between a sine curve and the vertical bob's actual path contributes to the approximation methods' (Section 3.3) requiring at least 1.2 cycles. Appendix B shows that Levenberg-Marquardt requires only a half cycle when the model-function and the input-function are both sine curves. If the model employed a curve that better matched the bob's actual path, the window size could be decreased.

To better understand the vertical bob's shape, consider the vertical bob and the walking gait cycle (Figure 2.3). The vertical bob's minimum height occurs when both feet are spread farthest apart – during Double Support. Figure 2.4 illustrates that the Double Support period's gait-cycle percent time decreases with increased step frequency – while other periods' percent time increases. Therefore, the relative amount of time spent around the bob's minimum height decreases at faster step frequencies: The lower trough sharpens at fast step frequencies.

- Figures 3.3 and 3.6 illustrate that the approximation methods don't converge to real-walking minima during transitions between not walking and walking – losing approximately half a step's real-walking data. In these boundary situations, the data window contains linear (not walking) and sinusoidal (walking) data. This problem could be reduced by weighting portions of the window on walking probability. This would allow the sine-curved-based model to fit walking portions more closely, and not be pulled to a false minimum by non-walking portions.
- Convergence to the horizontal components of the Forward Walk Model failed at times – requiring that vertical error be more heavily weighted. This may indicate inaccuracies with the horizontal components of the walk model. Asymmetric gait is not accounted for by the current model, and I have seen evidence of asymmetry in non-pathological walkers. Further research is necessary to determine if asymmetry or other factors hurt

horizontal-component convergence.

- The approximation methods’ window sizes include 1.2 full cycles at the lowest expected step frequency. As was shown in the analysis (Section 3.6), at higher step frequencies, the window contains so many cycles that the methods are unable to find a good fit for the entire window. An adaptive window size could solve this problem.

**Further validation:** Section 3.6 demonstrates that the approximation methods measure average walk speed, walk direction, and step frequency approximately as well as manually tagging head tracker logs. I believe this demonstrates that these approximation methods generally work. However, further validation is necessary. A GAITRite System could be used to validate step frequency, step length, walk speed, walk direction, and step timings at a finer scale. I know of no technique other than manually tagging tracker logs that could validate the bob-amplitude parameters. I know of no good way to validate the direction estimations for curved paths.

**Other applications:** Section 3.1.2 proposes that the approximation methods could be used for real-time animated walking avatars and for improving Redirected Walking techniques. Realizing either will require significant research.

### 5.1.3 User Study

**Rerunning the study:** While running the user study and analyzing the results, I realized that a few simple changes could improve the strength of the results. For any who replicate this between-subjects user study, I would recommend two changes.

First, different subjects have different propensities for simulator sickness. These propensities could serve as covariates to the simulator sickness scores. Lacking other ways to measure this, I would add three questions to the questionnaire (with Likert responses between “strongly agree” and “strongly disagree”):

- I get sick traveling in cars and airplanes.
- I get sick when traveling by boat.
- I get sick when on spinning carnival rides.

Second, the lack of a visual cue for the subject’s horizontal position made getting a

perfect Stopping Error score very unlikely. In the real world, people can ascertain horizontal position by seeing their feet on the ground. A simple VE technique is to draw a circular shadow under the user's head location. The Stopping Error task would then be the task of centering one's shadow over the target.

**Other metrics:** This dissertation demonstrates that including real-walking models measurably improves WIP systems. In Section 1.4, I state that WIP-system-caused distraction decreases other VE task performance. Further user studies could test if GUD WIP measurably improves users' ability to perform training, wayfinding, or other higher-level-cognitive tasks.

#### 5.1.4 Joystick

Current Joystick walking interfaces (JS) simply scale the joystick's displacement to create forward-walking speeds. These speeds are very different from Real Walking's. Several user studies have demonstrated that JS is worse than other walking interfaces (Usoh et al., 1999; Whitton et al., 2005). However, due to their low cost and simplicity, JS is still one of the most widely deployed walking interfaces. Therefore, even small JS improvements could benefit a variety of applications. I would recommend incorporating real-walking models like the following:

- Joystick pressure drives step frequency – indirectly altering speed.
- The development and decay phases are important components of any path: Resulting Joystick speeds – similar to WIP speeds – must include them.

# Appendices



# GUD WIP Implementation

Chapter 2 describes the GUD WIP model (Section 2.3) and provides some of the UNC GUD WIP implementation’s details (Section 2.4). In this appendix, the phrase “UNC GUD WIP” is shorthand for the “UNC GUD WIP implementation”. This appendix provides detailed implementation documentation for those who maintain UNC GUD WIP or want to replicate it.

## A.1 Overview

Figure A.1 shows the data flow through UNC GUD WIP’s computation thread. A separate thread maintains the user interface. Each box represents a C++ class that performs computations described in sections below.

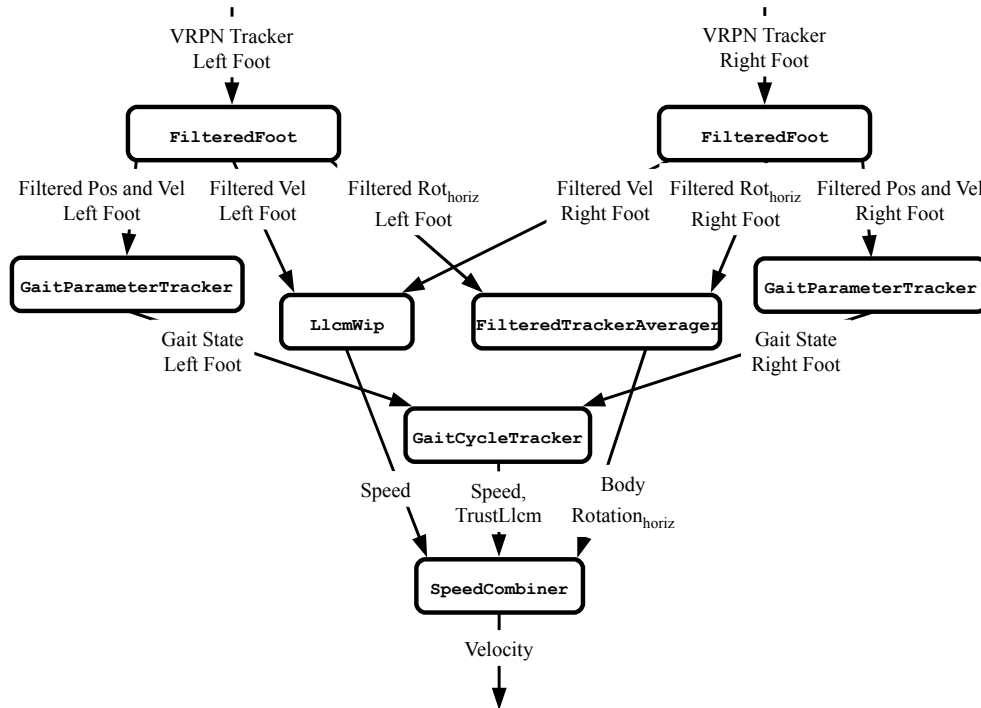


Figure A.1: **UNC GUD WIP system overview**: This diagram illustrates the data flow through UNC GUD WIP. Each box represents a class that processes its input data and outputs data for later classes.

Each section follows the same pattern:

- Purpose
- Inputs
- Computation
- Outputs

Some class names differ from the best descriptor of their final operation. Each class-describing section title contains a descriptor of the function, and the class name in parenthesis.

## A.2 Filtering Shin Data (`FilteredFoot`)

**Purpose:** Tracker noise obscures some GUD WIP state transitions (see Figure 2.10). The `FilteredFoot` class transforms and filters shin tracker data to aid later classes (Figure A.2).

**Inputs:** This class receives `PhaseSpace` tracker data via `VRPN` (Taylor II et al., 2001) for the position (3d point) and orientation (quaternion) of one of the user’s heels. The `PhaseSpace` tracker system is initialized to have the same orientation as the `HiBall` ceiling’s, but the origin is approximately the center of the cameras surrounding the user. A previous class estimates the heel’s position by transforming the shin position downward 35 cm. A separate instance exists for each heel.

**Computation:** To decrease noise, these input data are filtered by a Kalman filter (Kalman, 1960; Welch and Bishop, 2006). Kalman filters smooth data by estimating the system’s state via combining input tracker data with a state-progressing model. I examined the following models: position-velocity (PV), position-velocity-acceleration (PVA), and mixed PV/PVA. The mixed-PV/PVA-model filter runs both the PV- and PVA-model filters, outputting a weighted average of both. For the vertical component of my in-place stepping motions, the mixed model favored the PV model: The average weight for PV-model was  $\bar{x} = 0.79$  with  $s = 0.19$ ; for PVA-model,  $\bar{x} = 0.21$  with  $s = 0.19$ . I could see no input-data trend for the cases when the mixed model chose PVA over PV. This indicated that the PV model more closely approximated the input data. Since it provided acceptable results by itself, the final system filters with only the PV Kalman filter.



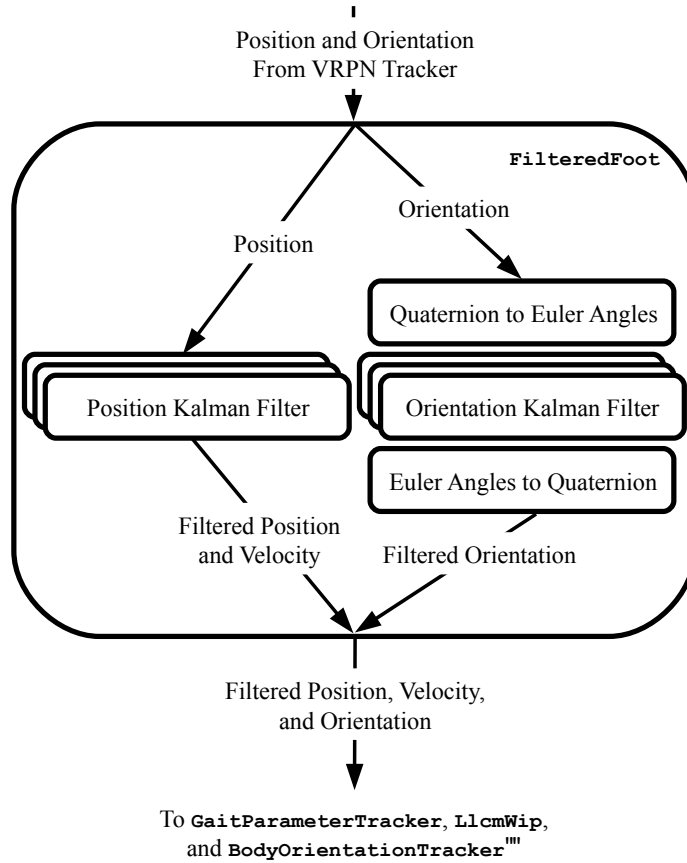


Figure A.2: **The FilteredFoot class:** The VRPN-input shin position is translated to the foot position, and each component is filtered. The VRPN-input quaternion rotation is transformed to Euler angles, filtered, and transformed back to quaternions.

This class employs 6 filters – three for each component of position (meters, subscript  $p$ ), and three for each Euler-angle rotation (degrees, subscript  $o$ ). For my trackers, the following measurement noises ( $r$ ) and process noises ( $q$ ) yielded good results:

$$r_p = 0.005 \quad q_p = 0.1 \tag{A.1}$$

$$r_o = 0.02 \quad q_o = 1.0 \tag{A.2}$$

**Outputs:** This class outputs the foot’s filtered position (3d point) and velocity (3d vector), and the shin’s orientation (quaternion).

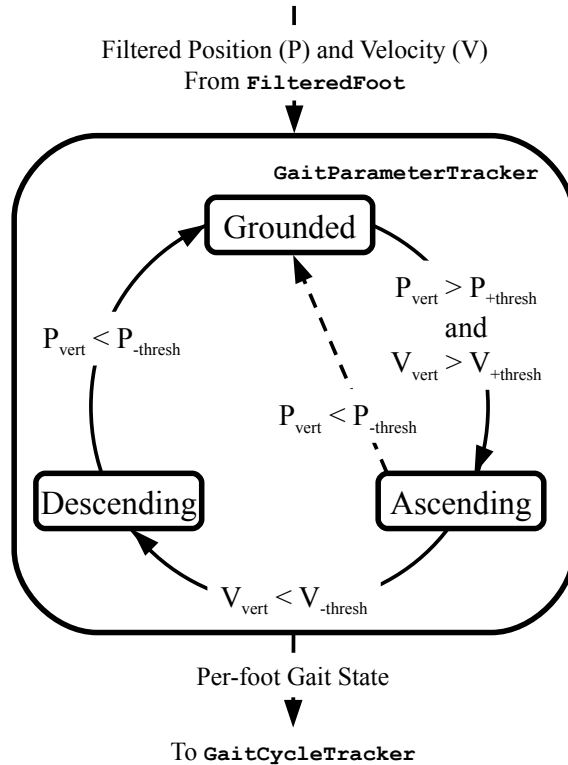


Figure A.3: **The `GaitParameterTracker` class:** The state transition diagram for a single foot as it takes in-place steps as implemented by the `GaitParameterTracker` class. The current per-foot gait state (`Grounded`, `Ascending`, or `Descending`) is maintained unless an exit criterion (on transitions) is fulfilled. The dashed transition handles possible input errors: It is not a part of steady-state operation.

### A.3 Per-Foot Gait State (`GaitParameterTracker`)

**Purpose:** The Gait Cycle state machine’s code (see Section A.4) was became too large to easily maintain as a single class. The `GaitParameterTracker` class simplifies the Gait Cycle calculation by classifying a single foot – be it `Grounded`, `Ascending`, or `Descending` (Figure A.3).

**Inputs:** This class receives input from an instance of `FilteredFoot`.

**Computation:** This class’s computation is described in Section 2.4.1.

**Outputs:** One of three per-foot gait states: `Grounded`, `Ascending`, or `Descending`.

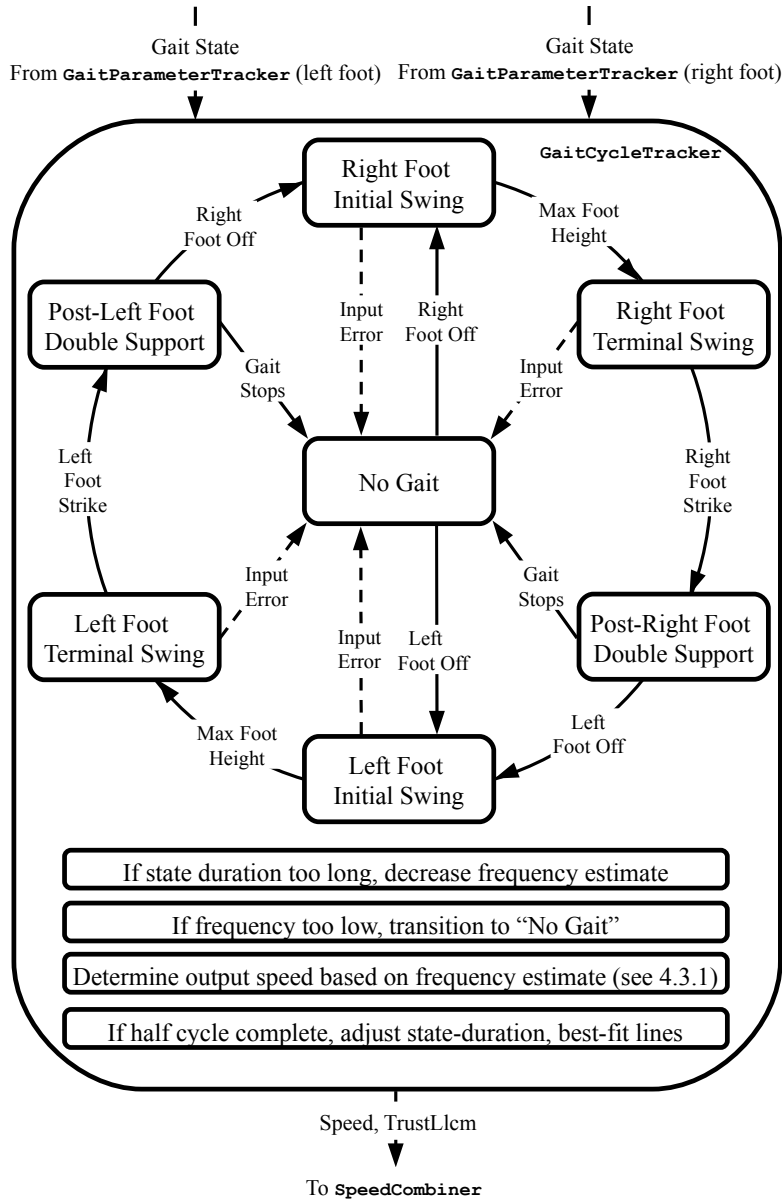


Figure A.4: **The `GaitCycleTracker` class:** The fundamental GUD WIP state machine. The current state is maintained unless an exit criterion is fulfilled. The dashed transitions are for software robustness on input errors.

## A.4 Gait Cycle Tracker (`GaitCycleTracker`)

**Purpose:** The `GaitCycleTracker` class implements the state machine described in Section 2.3, given the per-foot gait state (Figure A.4).

**Inputs:** Receives per-foot state from the left and right feet's `GaitParameterTrackers`.

**Computation:** Function already described in Section 2.3 and Subsection 2.4.2 is not re-

peated here. This section describes a robustness-enhancing feature: per-user gait-state-duration fitting.

UNC GUD WIP performs real-time line fitting to improve gait-state-duration estimation. The relationship between gait-phase-percent-time and step frequency is linear (Equation 2.3, Figure 2.4). The per-user UNC-GUD-WIP-measured in-place relationship varies from the walking averages provided in Equations 2.4–2.6. The following are possible sources of this variation:

- **Among-user variation:** The values in Equations 2.4–2.6 are across-user averages.
- **Real-to-in-place-walking variation:** Walking steps use different muscles and perform different motions than in-place steps.
- **Faulty assumption:** Equations 2.4 and 2.5 assume that initial and terminal swing evenly split the in-place swing; although approximately accurate, this is not precisely true.
- **Imprecise measurement:** The system does not precisely measure the instant the foot leaves the ground, lands on the ground, or reaches maximum height.

To overcome this variation, after the user completes a half gait cycle (initial swing, terminal swing, and double support), the percent of the cycle spent in each of these states is calculated and added to a per-user-session data store: With each new data point, the per-gait-state lines' slopes and intercepts are rediscovered through linear, least-squares fit to the updated data. Thus UNC GUD WIP becomes better the more a user uses it within a session.

**Outputs:** This class outputs the speed derived from the estimated step frequency and Equation 2.2, and a boolean (`TrustL1cm`) used to decrease starting latency (Subsection 2.3.2). `TrustL1cm` indicates when walking is detected, but no gait information is yet available. If a per-user step-frequency-to-walk-speed function is provided, it also outputs the Personalized GUD WIP speed.

## A.5 LLCM-WIP (`L1cmWip`)

**Purpose:** To decrease starting latency, GUD WIP requires LLCM-WIP (Feasel et al., 2008) (Figure A.5).

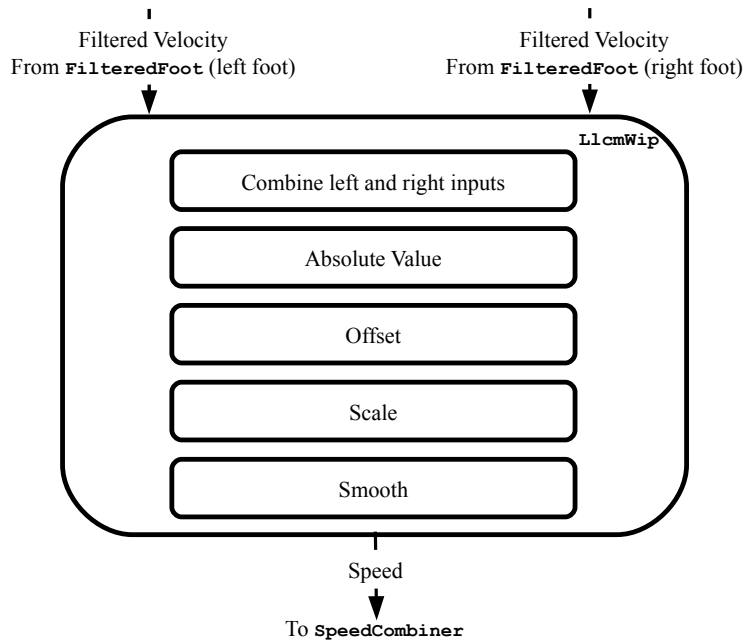


Figure A.5: **The L1cmWip class:** Implements Feasel’s LLCM-WIP algorithm (Feasel et al., 2008).

**Inputs:** The L1cmWip class takes inputs from the filtered foot tracker (Section A.2) – including the foot’s vertical velocity.

**Computation:** This class performs LLCM-WIP’s calculations. It scales, offsets, absolute values, smooths, and combines the vertical velocities of the user’s feet. I employ a fourth order Butterworth filter set to 5 Hz to smooth, and 0.1 m/s as the offset value. These calculations are performed at all times, although only sometimes used by later classes.

**Outputs:** The LLCM-WIP-predicted virtual walking speed.

## A.6 Walk Direction (FilteredTrackerAverager)

**Purpose:** WIP systems must output a horizontal-plane walk direction and speed. This class provides UNC GUD WIP’s shin direction to be used as the walk direction, averaged over the two shins (Figure A.6). This class is more general than its current usage: It calculates the average of all tracker values.

**Inputs:** Filtered tracker data for both shins.

**Computation:** It calculates the mean position and orientation of the shins.

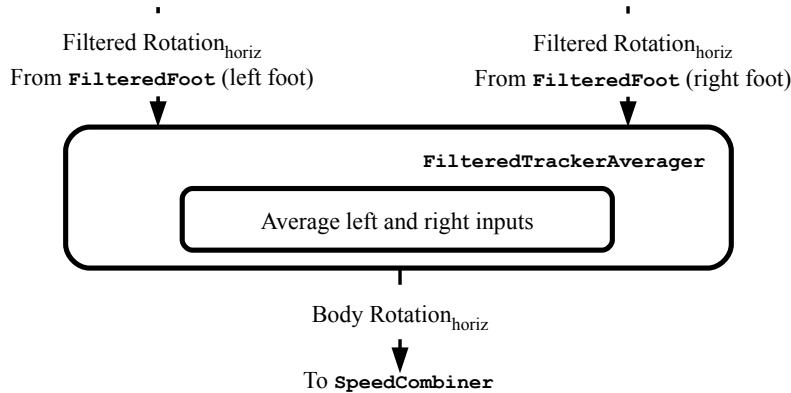


Figure A.6: **The FilteredTrackerAverager class:** This class averages the position and orientation of the user. UNC GUD WIP only uses the average horizontal rotation to estimate walk direction.

Alternate direction-setting techniques could be employed with little change to UNC GUD WIP: torso directed, chest directed, or something far more elegant.

**Outputs:** The input trackers’ averaged position and orientation. Of these, only the horizontal rotation component is used as the virtual walk direction.

## A.7 Creating Final Velocity (SpeedCombiner)

**Purpose:** The SpeedCombiner class provides direction and speed inputs to any VE system which needs virtual walking speeds (Figure A.7). The outputs are delivered via VRPN.

**Inputs:** It receives speeds from the LlcMwip, and GaitCycleTracker classes. Direction comes from FilteredTrackerAverager.

**Computation:** For GUD WIP speeds, during steady state, the GaitCycleTracker’s output speed is used as the WIP speed. During the initial swing phase of the first step, the GaitCycleTracker identifies that the user intends to move but can’t estimate intended speed. Thus, it sets TrustLlcM to true and LLCM-WIP’s speed is sampled and the maximum speed while TrustLlcM is true is used as the WIP speed.

To smooth hard transitions between LLCM-WIP-provided values and Gait Cycle Tracker values – and step-frequency updates at the completion of states – this class smooths the combined speed with a final Kalman filter ( $r = 100$ ,  $q = 1$ ).

**Outputs:** This class provides VE systems with three three-channel VRPN Analog Servers.

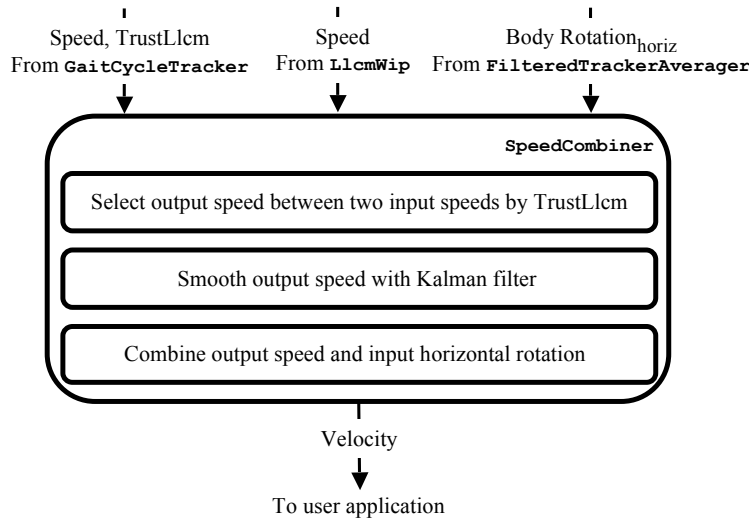


Figure A.7: **The `SpeedCombinerClass` class:** This class receives speed and direction inputs and determines the final GUD WIP velocity.

At each update, three floating point values (the three channels) are available for each of the following WIP systems:

- LLCM-WIP (passed directly from the `LlcmWip` class),
- Dean’s average-equation GUD WIP, and
- Personalized step-frequency-to-walk-speed GUD WIP (if personalized function available).

The three channels are each a three-space walking velocity vector (the vertical component is always zero). Providing all three WIP system speeds enables comparison user studies (Chapter 4).





# Forward Walking Model Analysis: Levenberg-Marquardt Fitting to Perfect Sinusoids

## B.1 Introduction

Chapter 3 describes how I use Levenberg-Marquardt Analysis (LMA) to fit head-track data to the Forward Walking Model. In this appendix, I analyze how well LMA fits a sinusoidal model to sinusoid input data. This analysis provides insights into LMA's characteristics when fitting the head's vertical bob.

This appendix assumes the reader understands Chapter 3, especially how LMA is used for the Forward Walking Model's approximation methods. In this appendix, errors below five decimal places are considered sufficiently accurate.

## B.2 Analysis

In this analysis, I examine how well LMA performs under varying conditions. Both the to-be-fit model and input data use the same equation:

$$y = a \cdot \sin(2\pi \times ft + \phi) + c \quad (\text{B.1})$$

where  $y$  is the simulated head height (meters),  $a$  is the amplitude (meters),  $f$  is the frequency (Hz),  $t$  is time (seconds),  $\phi$  is the phase (radians), and  $c$  is the offset (meters).

I use `lmfit` – an implementation of the Levenberg-Marquardt Algorithm in C (Wuttke, 2009). `lmfit` requires a function that calculates the residual ( $\epsilon$ ) at every the input data element:

$$\epsilon_i^2 = (y_i - a_j \cdot \sin(2\pi \times f_j t_i + \phi_j) + c_j)^2 \quad (\text{B.2})$$

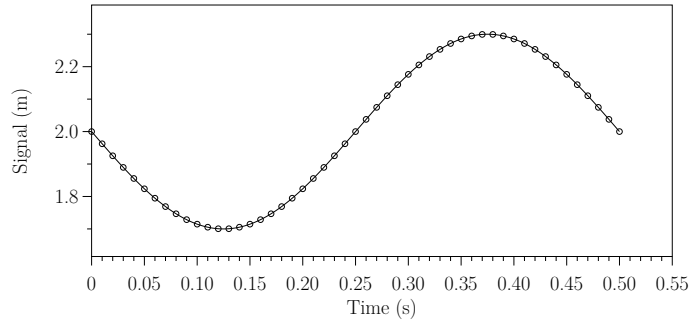


Figure B.1: **lmfit results:** Given input data (points), **lmfit** approximates model parameters for the best fit (sinusoidal curve). Input data and resulting model fit have amplitude=0.3 m, frequency=2 Hz, phase= $\pi$  radians, and offset=2 m.

where  $x$  and  $y$  are the input data, subscript- $i$  denotes the  $i$ -th element in these arrays, and subscript- $j$  denotes the parameter estimates at the  $j$ -th execution of this method.

Herein, I constrain the amplitude and frequency to be positive. For details on parameter constraining, see Subsection 3.3.1.

The inputs to my **lmfit**-calling code are as follows:

- The to-be-fitted data: In this analysis, I made the points always evenly spaced along a perfect sine curve with the following varying characteristics:
  - **Window Size**
  - **Sample Resolution**
  - **Model Parameters:** The input data's amplitude ( $a$ ), frequency ( $f$ ), phase ( $\phi$ ), and offset ( $c$ )
- **Initial Estimates:** The initial values for amplitude ( $a_0$ ), frequency ( $f_0$ ), phase ( $\phi_0$ ), and offset ( $c_0$ )

In each of the following analyses, the calling code varies one of the bold-faced input values. The variations cover the space of expected real-walking values. The errors shown herein are the two-norm ( $L_2$ ) of the residual of the fit data (Equations B.2). Each analysis follows the same pattern:

- Purpose
- Process
- Results

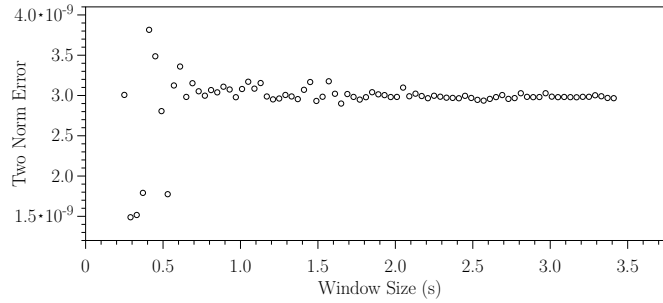


Figure B.2: **Varying window size:** For all window sizes, convergence was excellent: Even before it converges at approximately 1.0 seconds, the errors are very small. Window size varies (0.25...3.5 sec). Resolution: 256 samples per window. Parameters: amplitude = 0.03 m, frequency = 1.0 Hz, phase = 1.0 radians, and offset = 1.8 m. Initial estimates: 10% error added to all parameters.

- Conclusion

### B.2.1 Varying Window Size

**Purpose:** Larger input windows have higher smoothing for time-varying data (if the frequency changes on the window), and higher latency (for the center time on the window). This seeks the minimum acceptable window size to decrease both.

**Process:**

- *Window size:* Varying from 0.25 sec to 3.5 sec
- *Resolution:* Fixed at 256 samples / window
- *Parameters:* Fixed at amplitude = 0.03 m, frequency = 1.0 Hz, phase = 1.0 radians, offset = 1.8 m
- *Initial estimates:* Fixed at 10% error added to true values

**Results** (Figure B.2): Since the input data was precisely a 1.0 Hz sinusoid, the seconds in the window are also the number of cycles in the window. Although the error converges to a fixed error at window sizes greater than one cycle (1 second), even the “un-converged” errors (< 1 second) are still better than 8 digits of accuracy, and some are lower errors than the “converged” errors.

**Conclusion:** I use a 0.5 second window size for all following analyses.

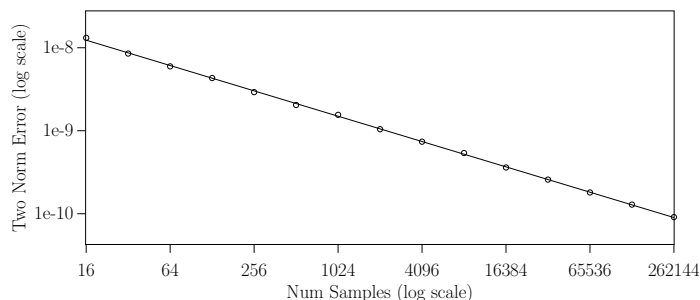


Figure B.3: **Varying sampling resolution:** A log-log plot. For all resolutions, convergence was excellent. The fit-line's slope (-0.507) indicates approximate square-root convergence with increased resolution. Window size: 0.5 sec. Resolution varies (16, 32, ...  $2^{18}$ ). Parameters: amplitude = 0.03 m, frequency = 1.5 Hz, phase = 1.0 radians, offset = 1.8 m. Initial estimates: 30% error added to all parameters.

## B.2.2 Varying Sampling Resolution

**Purpose:** Determine `lmfit`'s convergence properties with varying sample resolution.

**Process:**

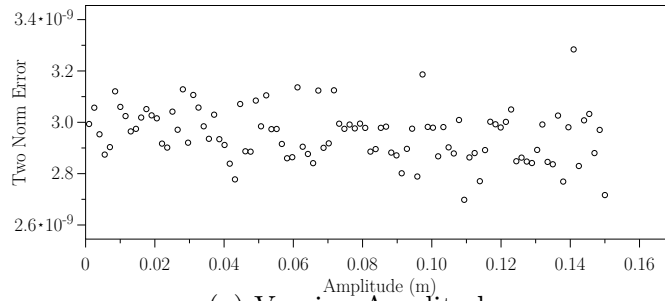
- *Window size:* Fixed at 0.5 sec
- *Resolution:* Varying: 16, 32, 64, ...  $2^{18}$  samples
- *Parameters:* Fixed at amplitude = 0.03 m, frequency = 1.5 Hz, phase = 1.0 radians, offset = 1.8 m
- *Initial estimates:* Fixed at 30% error added to true values

**Results** (Figure B.3): This figure shows that as the number of samples increases, the precision increases. The fit-line's slope (-0.507) indicates that the resulting fit is improved by approximately the square root of the increased number of samples: Doubling the number of samples improves the accuracy by approximately 1.414 times.

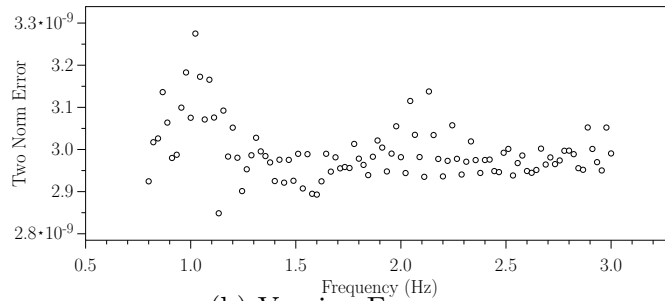
**Conclusion:** Since all tests are accurate to at least seven digits, as few as 16 samples per cycle suffices. In all remaining tests, I employ a relatively modest number of samples per window (256).

## B.2.3 Varying Model Parameters

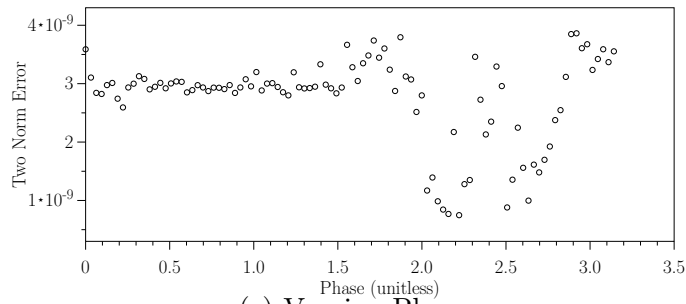
**Purpose:** Determine `lmfit` convergence at selected window size and resolution for all real-walking-range parameter values.



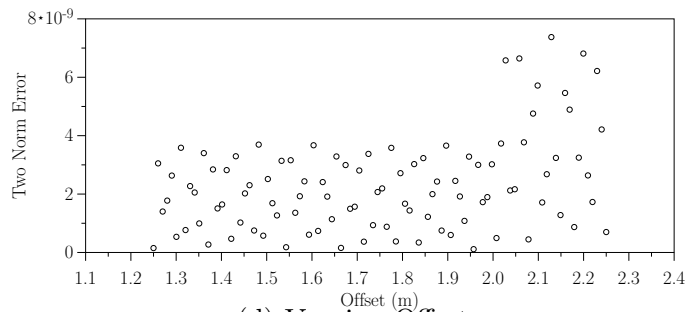
(a) Varying Amplitude



(b) Varying Frequency



(c) Varying Phase



(d) Varying Offset

Figure B.4: **Varying parameter values:** Varying (a) amplitude, (b) frequency, (c) phase, and (d) offset. All results are better than required accuracy ( $10^{-6}$  would likely be sufficient). Window size: 0.5 sec. Resolution: 256 samples. Parameters (when not the varying parameter): amplitude = 0.03 m, frequency = 1.5 Hz, phase = 1.0 Hz, offset = 1.8 m. Initial estimates: 30% error added to all parameters.

**Process:**

- *Window size:* Fixed at 0.5 sec
- *Resolution:* Fixed at 256 samples
- *Parameters:* Each varied individually. When not varying, amplitude = 0.03 m, frequency = 1.5 Hz, phase = 1.0 radians, offset = 1.8 m. When varying:
  - amplitude = (0.0...0.15) m
  - frequency = (0.8...3.0) Hz
  - phase = (0.0... $\pi$ ) radians
  - offset = (1.25...2.25) m
- *Initial estimates:* Fixed at 30% error added to true values

**Results** (Figure B.4): `lmfit` converges to sufficiently accurate results for all parameters for values within real-walking ranges.

**Conclusion:** All real-walking-valid parameter values should be fine for `lmfit`.

## B.2.4 Varying Initial Estimates

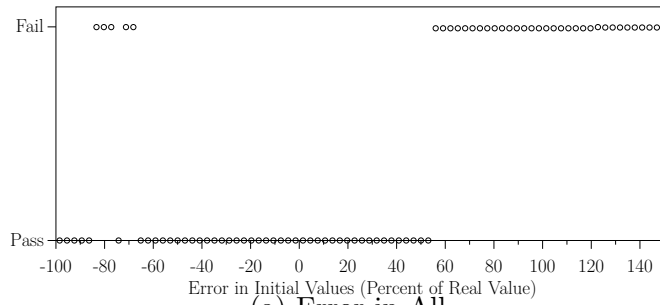
**Purpose:** Determine how initial estimate errors affect `lmfit` convergence. Determine if any parameter's estimate is more important than others.

**Process:**

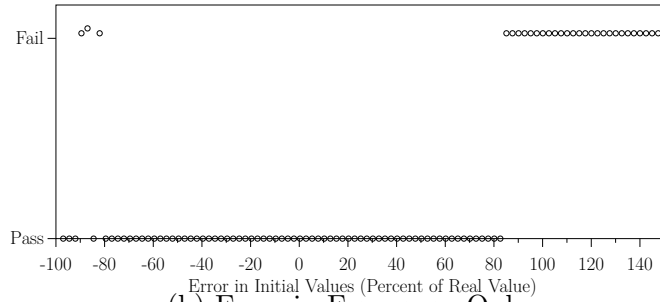
- *Window size:* Fixed at 0.5 sec
- *Resolution:* Fixed at 256 samples
- *Parameters:* Fixed at amplitude = 0.03 m, frequency = 1.5 Hz, phase = 1.0 radians, offset = 1.8 m
- *Initial estimates:* Varied from -100%...+150% of the correct value. For not-varying parameters, initial estimates were set to be precisely correct.

**Results** (Figure B.5): In (a), all parameter's initial estimates vary. When all parameters are within about +/-50% of their correct values, `lmfit` converged to the global minimum. Outside that region, convergence to local minima occurred often.

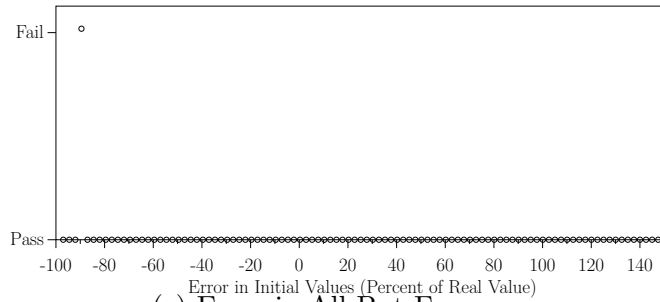
In (b), only frequency's initial estimate varies. The global-convergence range is approximately +/-80%. In (c), all except frequency vary. The global-convergence range is



(a) Error in All



(b) Error in Frequency Only



(c) Error in All But Frequency

Figure B.5: **Varying initial estimates:** Varying (a) all parameters, (b) frequency only, and (c) all parameters except frequency. Two-norm errors at zero converged to the global minimum. All others converged to an incorrect local minimum. Window size 0.5 s. Resolution: 256 samples. Parameters: amplitude = 0.03 m, frequency = 1.5 Hz, phase = 1.0 radians, offset = 1.8 m. Initial estimates: When not varying, 0% added.

approximately -90%...+150%. Figure B.6 shows what one of the failed fits looked like. Note that the fit's frequency was double that of the input data, but that the phase was such that the fit's maximum and minimum approximately matched locations of the input data's maximum and minimum.

**Conclusion:** Figure B.5(b) and (c) demonstrate that frequency is the most important estimate. When only the frequency estimate varied from the true value, there were still many cases where `lmfit` converged to the wrong minimum. When all other estimates

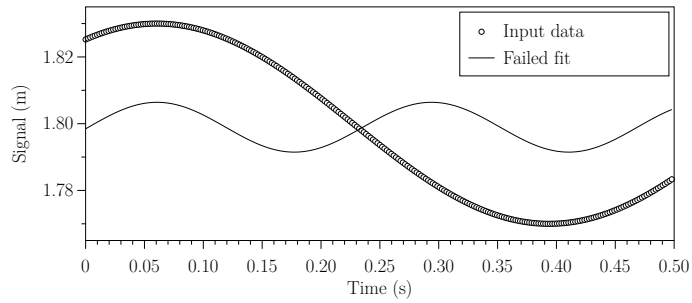


Figure B.6: **Failed fit:** When the frequency was set to  $\sim 1.5$  times the actual input frequency, `lmfit` went to a local minimum fit with frequency approximately twice that of the input data. Note that the phase that resulted places the fit's maximum and minimum approximately in line with the input data's.

except the frequency estimate varied from the true value, only one case did not converge correctly.



# Forward Walking Model Analysis: Fourier Fitting to Perfect Sinusoids

## C.1 Introduction

Chapter 3 presents the Forward Walking Model and two Levenberg-Marquardt-Algorithm-based approximation methods. In Section 3.5, I briefly mention the problems encountered when attempting a Fourier-based approximation method for the Forward Walking Model. This appendix provides greater details on this attempt.

## C.2 Discrete Fourier Principles

This brief review of Discrete Fourier Principles comes from Smith's text (Smith, 1997), personal exploration, and discussions with Gary Bishop.

Fourier principles state that any discrete signal may be decomposed into a set of discretely sampled sine waves. The Discrete Fourier Transform (DFT) provides the mechanism for discovering the frequency, magnitude, and phase of these frequencies. Thus, the DFT transforms any time-sampled signal from the time domain to the frequency domain (where the signal is considered the sum of a discrete series of sine waves).

The DFT can be written as a matrix operation. Given some discretely sampled input function (fn), the Fourier transform (ft) can be obtained through the following matrix multiplication:

$$\begin{bmatrix} \text{dft}(f_0, t_0) & \text{dft}(f_0, t_1) & \cdots & \text{dft}(f_0, t_{n-1}) \\ \text{dft}(f_1, t_0) & \text{dft}(f_1, t_1) & \cdots & \text{dft}(f_1, t_{n-1}) \\ \vdots & \vdots & \ddots & \vdots \\ \text{dft}(f_{\frac{n}{2}}, t_0) & \text{dft}(f_{\frac{n}{2}}, t_1) & \cdots & \text{dft}(f_{\frac{n}{2}}, t_{n-1}) \end{bmatrix} \times \begin{bmatrix} \text{fn}(t_0) \\ \text{fn}(t_1) \\ \vdots \\ \text{fn}(t_{n-1}) \end{bmatrix} = \begin{bmatrix} \text{ft}(f_0) \\ \text{ft}(f_1) \\ \vdots \\ \text{ft}(f_{\frac{n}{2}}) \end{bmatrix} \quad (\text{C.1})$$

where  $t_i$  are the sampling times of the input function,  $f_i$  are the basis frequencies (Subsection C.2.1), and  $\text{dft}(f_i, t_i)$  is shorthand for the following:

$$\text{dft}(f_i, t_i) = \frac{\cos(2\pi \times f_i \times t_i)}{(\|\cos(2\pi \times f_i \times t)\|_2)^2} + i \frac{\sin(2\pi \times f_i \times t_i)}{(\|\sin(2\pi \times f_i \times t)\|_2)^2} \quad (\text{C.2})$$

where  $\|\cdot\|_2$  is the  $L_2$  norm. Many texts replace the normalizing denominators in this equation with  $N/2$  and two special-cases of  $N$ . These are the final values for the two-norms on the DFT's basis frequencies. Equation C.2's more general form is used later in this appendix.

Each  $\text{ft}(f_i)$  is composed of a real ( $\text{ft}_{\text{re}}(f_i)$ ) and imaginary ( $\text{ft}_{\text{im}}(f_i)$ ) component. The magnitude and phase of each Fourier component can be directly obtained:

$$\text{magnitude}(\text{ft}(f_i)) = \sqrt{(\text{ft}_{\text{re}}(f_i))^2 + (\text{ft}_{\text{im}}(f_i))^2} \quad (\text{C.3})$$

$$\text{phase}(\text{ft}(f_i)) = \tan^{-1} \left( \frac{\text{ft}_{\text{im}}(f_i)}{\text{ft}_{\text{re}}(f_i)} \right) \quad (\text{C.4})$$

### C.2.1 Basis frequencies

The basis frequencies for a Discrete Fourier Transform are chosen such that there are an integral number of cycles within each row of the DFT matrix (see Eq. C.1). For each  $f_i$ , the number of cycles within each row is precisely  $(i) - \text{Zero}$  in the first row, one in the second, two in the third,  $\dots$ ,  $\frac{n}{2}$  in the last.<sup>1</sup> Each  $\text{ft}(f_i)$  describes how much the  $i$ -th frequency contributes to the input function.

These basis frequencies limit the DFT's resolution. If the input is a sine curve with a number of cycles not represented by the basis frequencies (say 3.5 cycles), the DFT can only determine that one of the two neighboring basis frequencies best matches (3 or 4 cycles).

---

<sup>1</sup>Note this does not state these frequencies in Hz. The input function's window size ( $t_0, t_{n-1}$ ) affects the selected frequencies. If the input function has precisely one second's data, then the number of cycles per row is also the frequency. Otherwise, the frequencies must be scaled appropriately to discover the frequency represented.

## C.2.2 Possible Solutions

If greater frequency resolution is desired, some change must be made to either the algorithm or the input. Three variation classes are discussed herein. Section C.3 discusses various ways to increase the window size. Section C.4 discusses an algorithm that employs DFT principles but alters the basis frequencies for a fixed window size. Section C.5 describes an algorithm that combines both of the previous two: not directly employing the DFT while altering the window size.

## C.3 Increase the Window Size

Increasing the window size directly increases the DFT's frequency-resolving power by the following relationship:

$$\text{Resolution} = \frac{1}{\text{Window Size}} \quad (\text{C.5})$$

In this section, three different techniques for increasing the window size are considered briefly. None of these were directly employed in my analysis, for the reasons stated with each.

### C.3.1 Increase the Sample Duration

The most straightforward way of increasing the window size is to increase the sampling duration of the input function: Instead of inputting one second's data, input two seconds' data.

A problem with this approach is that one is no longer analyzing the frequency of the original function, but a larger function. If the function were a single time-varying frequency, then although increasing the window size increases the frequency resolution, it decreases the time resolution.

This problem is compounded by the fact that with an increased window size, the increased frequency resolution is quite small. In order to increase a single significant digit in frequency resolution, ten times the input duration is required.

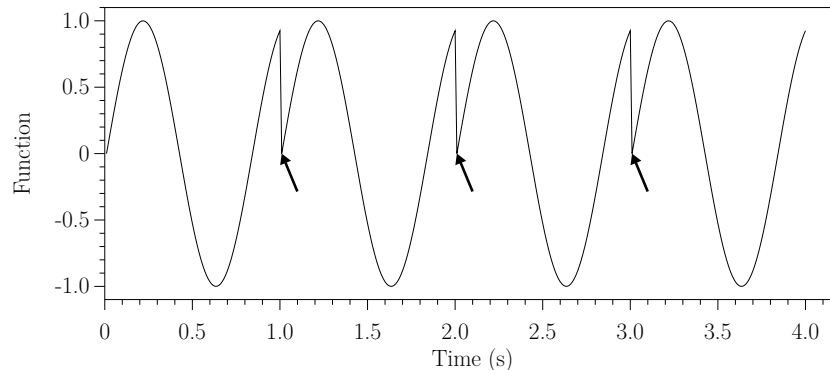


Figure C.1: **Repeating the input signal:** This causes discontinuities (at arrows) in the input function that show up as high-frequency components, and obscures the original function's phase.

### C.3.2 Repeat the Input Sample

Figure C.1 demonstrates a window-size-increasing algorithm that avoids the time-averaging effect described in the previous section: Repeat the input signal some number of times.

While this does provide increased frequency-resolution without diminished time resolution, there are at least two problems with this approach:

- **High frequency noise:** Discontinuities introduced at the repeat locations (arrows in Figure C.1) now show up as high-frequency noise in the Fourier transform. This can make the input signal's frequency harder to identify.
- **Phase lost:** Since the repeated signals do not repeat at precise full-cycle positions, the input signal's phase is altered with each repeated value.

### C.3.3 Repeat and Mirror the Input Sample

To avoid function discontinuities (Figure C.1, arrows), the repeated signal can be mirrored at every-other repeated instance (Figure C.2). Unfortunately, these mirror images still cause first-derivative discontinuities, still insert high frequency noise, and still obscure the phase.

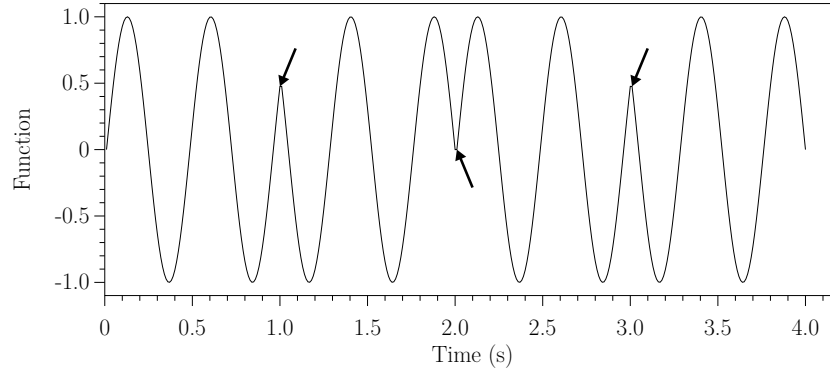


Figure C.2: **Repeating and mirroring the input signal:** Although this avoids function discontinuities, it still causes first-derivative discontinuities (at arrows) in the input data that show up as high-frequency components, and obscures the original function’s phase.

## C.4 Vary the Basis Frequencies

### C.4.1 Method Description

This technique keeps the input signal fixed and alters the DFT basis frequencies.

The principal insight for this algorithm is that each  $ft(f_i)$  in Equation C.1 is the product of the corresponding row of the DFT matrix and the input function. Mathematically (using the same notation as Equations C.1 and C.2):

$$ft(f_i) = \frac{\cos(2\pi \times f_i \times t) \circ fn(t)}{\|\cos(2\pi \times f_i \times t)\|_2} + i \frac{\sin(2\pi \times f_i \times t) \circ fn(t)}{\|\sin(2\pi \times f_i \times t)\|_2} \quad (C.6)$$

where  $\circ$  denotes the dot product between the two functions (iterating over  $t$ ).

The basis frequencies ( $f_i$ ) chosen by the DFT are such that there are an integral number of cycles in each row of the matrix. To increase the frequency resolution without changing the window size, this algorithm chooses non-integer-cycle basis frequencies – allowing for any frequency to be dotted with the input signal via Equation C.6.

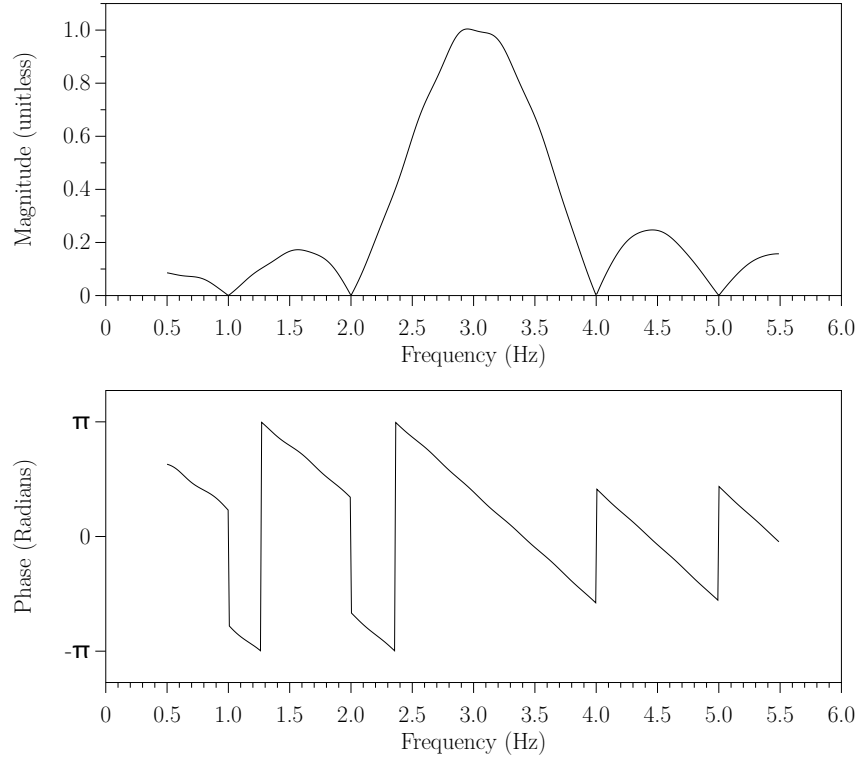


Figure C.3: **Vary basis frequency – magnitude and phase:** An absolute value sinc function and linearly varying phase. This example shows the result of testing 512 equally-spaced frequencies between 0.5 and 5.5 Hz with Equation C.7. The magnitude peak is highest near 3 Hz, with approximately a correct magnitude, and phase.

## C.4.2 Results

**Positive:** I created a program that calculates Equation C.6 for variable frequency. For all tests, I fit a simple input function:

$$\text{fn}(t) = \sin(2\pi \times f \times t + \phi) \quad (\text{C.7})$$

With  $t$  sampled regularly on a sample window.

Figure C.3 demonstrates the results of running this program against input with the following characteristics ( $f = 3.0$  Hz,  $\phi = 1.22$  radians, and window size = 1 sec). This was tested against 512 equally-spaced frequencies (0.5...5.5 Hz).

Considering the input function and the operations performed on it in both the spatial and frequency domains demonstrates that this figure represents what would be expected to

occur. Since the input spatial-domain function is a pure sine wave of a single frequency, its frequency-domain representative should be a two spikes at  $+f$  and  $-f$ . However, limiting the input function to a window multiplies it by a box filter in the spatial domain – convolving the frequency-domain spikes with a sinc function.

All this is visible in Figure C.3. The resulting magnitude is similar to the absolute value of a sinc function. The irregularities in the sinc function (non-symmetry, lumpy appearance – particularly in the largest hump, etc.) come from the addition of two sinc functions from the two frequency-domain spikes. The discontinuities in the phase estimates occur at one of two places: at the sinc function’s zero crossings, or when the phase transitions from  $-\pi$  to  $\pi$ .

The maximum-magnitude frequency from this algorithm approximates the input sine wave’s frequency. The magnitude and phase at that frequency approximate the magnitude and phase of input function.

**Negative:** Unfortunately, the largest hump’s lumpiness causes the maximum-magnitude frequency to be not the input function’s actual frequency, but a slightly lower frequency ( $\sim 2.95\text{Hz}$ ). In other cases, the lumpiness causes the maximum-magnitude frequency to be a slightly higher frequency. The maximum magnitude frequency moves due to variations of the input function’s non-frequency parameters.

**Variation 1, number of cycles input:** I ran this basis-function-changing algorithm against a fixed sine wave (frequency 1.0 Hz, magnitude = 1.0, phase = 0.0 radians, varying window size = (0.5 . . . 4.5) cycles in window, Figure C.4).

With less than one cycle, the best-identified frequency is very inaccurate. The visible discontinuities (at approximately 1.05, 1.5, 1.95, 2.4, etc. cycles) occur when the higher-frequency lump suddenly becomes larger than the lower-frequency lump. Although it converges to the correct frequency, it does so at a slow pace: The discontinuities are still visible above four cycles.

**Variation 2, input’s phase:** I ran the basis-function-changing algorithm against an input sine wave (frequency = 2.0 Hz, magnitude = 1, varying phase (0 . . .  $\pi$ ) radians, window size = 1 cycle, Figure C.5).

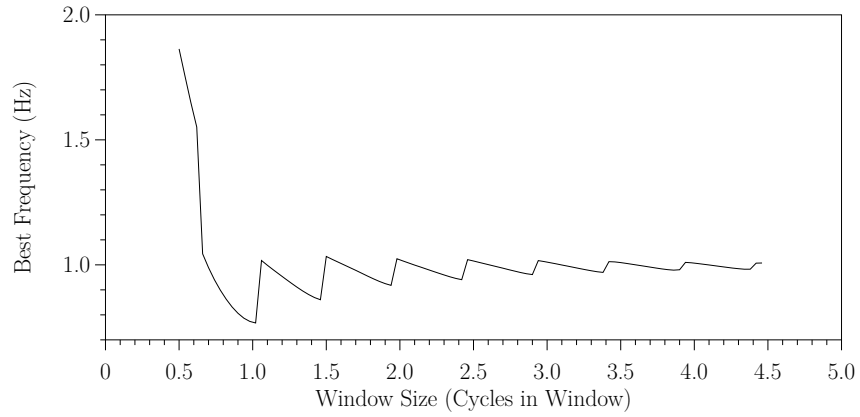


Figure C.4: **Maximum magnitude frequency with varying window size:** The error's shape repeats but decreases in size with increased window size. It converges to the correct value.

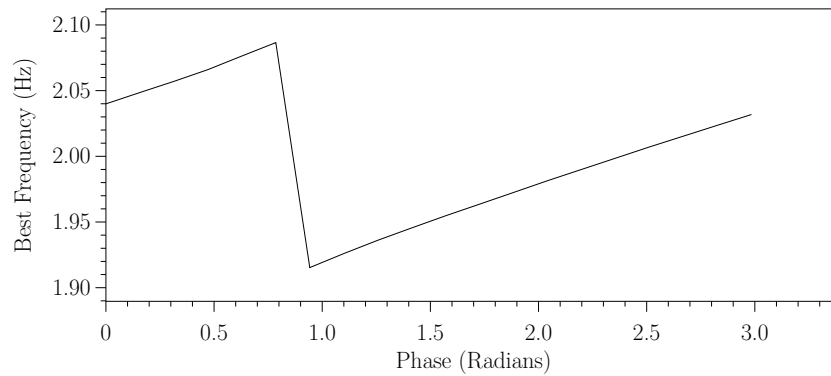


Figure C.5: **Phase affects maximum magnitude frequency:** The maximum-magnitude frequency for a 2.0 Hz sine with varying phase (x-axis) and a fixed window size (1.0 cycle).



The input data's phase affects the method's accuracy.

### C.4.3 Discussion

I believe that the discontinuities in Figures C.4 and C.5 are caused by the repeated-input problem (Subsection C.3.2, Figures C.1 and C.2): There are high-frequency components affecting the result.

When the input signal is repeated, the high frequency appears at repetition boundaries. In this case, the high-frequency is in the rows of the DFT matrix. Fourier principles assume that the contents of that row repeat infinitely, and in this method, the rows do not contain an integral number of cycles. Thus when the row implicitly repeats, it creates discontinuities.

## C.5 Vary the Basis Frequencies and the Window Size

### C.5.1 Method Description

I attempted a third frequency-resolution method, using both an integer number of cycles in the basis functions and not repeating the input function. I accomplished this by varying the window size at different frequencies.

An example will help clarify this method: If the input is a two second data window, sampled at 20 samples per second, the following frequencies can be analyzed:  $2 \text{ cycles}/2\text{s} = 1 \text{ Hz}$ ,  $2 \text{ cycles}/1.95 \text{ s} = 1.025 \text{ Hz}$ ,  $2 \text{ cycles}/1.9 \text{ s} = 1.052 \text{ Hz}$ , ... At every possible window size, the input is tested with precisely two cycles. (Any integer could be used. Two was chosen to permit relatively low frequencies (0.67 Hz) at relatively small window sizes (3 sec).)

### C.5.2 Results

**Positive:** Figure C.6 shows a single execution's output from this method against an input sine (frequency = 1.75 Hz, amplitude = 1.0, phase = 1.22 radians).

This figure is harder to explain than Figure C.3. Again, the spatial-domain input is a single frequency (two spikes in the frequency domain). However, now the window size

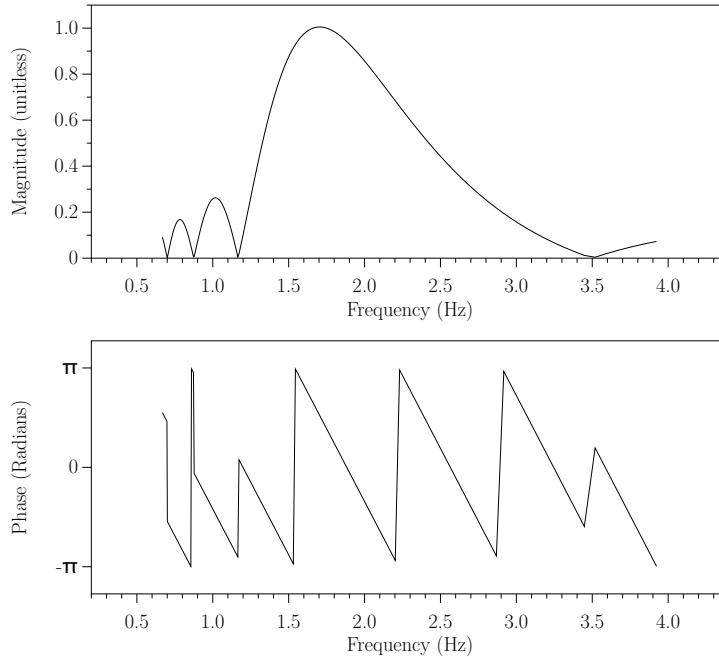


Figure C.6: **Varying basis frequency and window size method – magnitude and phase:** Input is a fixed sine curve (frequency = 1.75 Hz, amplitude = 1.0, phase = 1.22 radians, window size = 3.0 sec). The maximum magnitude is highest near 1.75 Hz, with approximately a correct amplitude and phase.

(multiplication by a box in spatial domain – convolution by a sinc in the frequency domain) is changing for each tested frequency. Therefore, each magnitude value in Figure C.6 is a single sample, each from a slightly different spike-convolved-with-sinc function. Thus, the resulting sinc appears compressed at lower frequencies and stretched at higher frequencies. Just as with Figure C.3, the phase is discontinuous only when the sinc is zero, or when it transitions from  $-\pi$  to  $\pi$ .

**Negative:** Unlike Section C.4’s method, this method’s frequency resolution depends on the input’s spatial resolution. I tested this against a 2.0 Hz sine input with varying sampling resolution (Figure C.7). As the resolution increases, the method’s best fit converges to  $\sim 1.93$ Hz. Unfortunately, although it converges, it does not converge to the correct answer (2.0 Hz). This is because each magnitude value’s sinc function still has lumps in it.

Although this technique eliminates the phase-caused changes’s discontinuity, the result still varies (compare Figures C.5 and C.8).

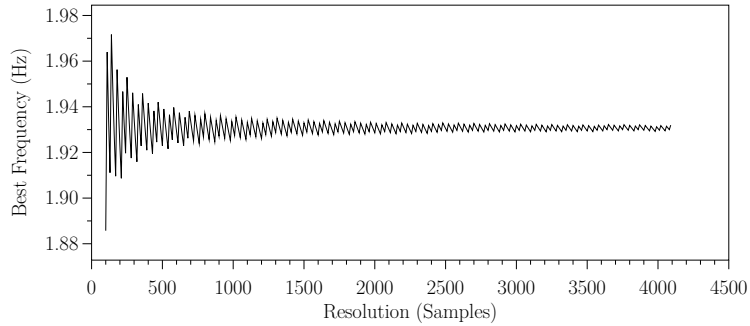


Figure C.7: **Input resolution changes method’s accuracy:** The maximum-magnitude frequency for a 2.0 Hz signal with increasing spatial resolution.

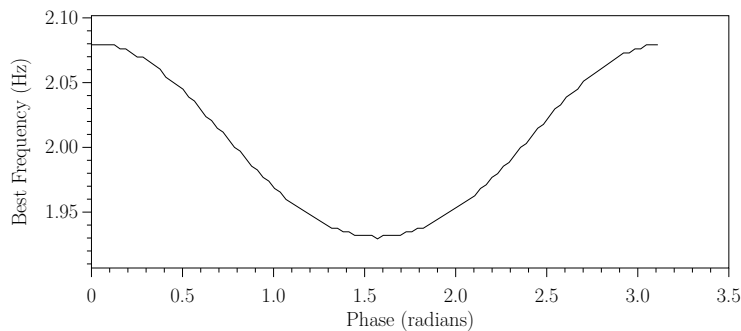


Figure C.8: **Phase affects maximum-magnitude frequency:** Input is a single sine (frequency = 2.0 Hz, varying phase ( $0 \dots \pi$ ) radians, window size = 3.0 sec).

### C.5.3 Discussion

This method can be improved. For instance, Section C.4 shows that increasing the input’s window size improves accuracy. Therefore, trying three or more input cycles should improve Figure C.7’s converged-to solution’s accuracy, and decrease Figure C.8’s variation. However, increasing number of cycles at lower frequencies requires larger input windows – causing increased averaging and latency.

If various number of integer cycles were attempted at each window size, frequency resolution could improve without increasing spatial resolution. This would introduce two difficulties: The first is increased implementation complexity. This is negligible if the benefit is high. Second, analyzing the output would be considerably more complex. Figure C.6 shows the compressed and expanded sinc function that results when each higher frequency is the result of a slightly smaller window size. If multiple frequencies were tested at each

window size, the resulting sinc would have different window sizes for different resolutions with no simple description of how the window size changes with increased resolution. Neither of these difficulties should stop those interested in improving the frequency resolution.

## C.6 Conclusion

Each of the Fourier-based methods result in significant errors. These errors appear inherent to Fourier methods:

- More input signal provides a better output for a fixed input frequency, but increases averaging for time-variable input frequencies.
- The input's phase affects the result.

The analysis in Appendix B demonstrates that Leveberg-Marquardt-based methods are more effective at fitting sines: The errors are orders of magnitude smaller.

## APPENDIX D

# Usability Study Questionnaire

Please enter answers to the following questions.

Participant number:

SECTION I. In this section, you will describe your current physical condition. For each of the following conditions, please indicate how you are feeling right now, on a scale of “none” through “severe”. Here are definitions for some of the conditions: Fatigue - Weariness or exhaustion of the body. Eye Strain - Weariness or soreness of the eyes. Nausea - Stomach distress. Vertigo - Surroundings seem to swirl. Stomach Awareness - Just a short feeling of nausea. Fullness of Head - Sinus pressure. Example: 1 - None, 2 - Slight, 3 - Moderate, 4 - Severe.

1. General Discomfort

- 1 (None)
- 2
- 3
- 4 (Severe)

2. Fatigue

- 1 (None)
- 2
- 3
- 4 (Severe)

3. Headache

- 1 (None)
- 2
- 3
- 4 (Severe)

4. Eye Strain

- 1 (None)
- 2
- 3
- 4 (Severe)

5. Difficulty Focusing

- 1 (None)
- 2
- 3
- 4 (Severe)

6. Increased Salivation

- 1 (None)
  - 2
  - 3
  - 4 (Severe)
7. Sweating
- 1 (None)
  - 2
  - 3
  - 4 (Severe)
8. Nausea
- 1 (None)
  - 2
  - 3
  - 4 (Severe)
9. Difficulty Concentrating
- 1 (None)
  - 2
  - 3
  - 4 (Severe)
10. Fullness of Head
- 1 (None)
  - 2
  - 3
  - 4 (Severe)
11. Blurred Vision
- 1 (None)
  - 2
  - 3
  - 4 (Severe)
12. Dizzy (with your eyes open)
- 1 (None)
  - 2
  - 3
  - 4 (Severe)
13. Dizzy (with your eyes closed)
- 1 (None)
  - 2
  - 3
  - 4 (Severe)
14. Vertigo
- 1 (None)
  - 2
  - 3
  - 4 (Severe)
15. Stomach Awareness
- 1 (None)

- 2
  - 3
  - 4 (Severe)
16. Burping
- 1 (None)
  - 2
  - 3
  - 4 (Severe)
17. Please list any additional symptoms you are experiencing
- (Big box to write in.)

SECTION II. In this section, you will rate the locomotion technique you used.

1. The technique started moving when I wanted it to.
  - 1 (Strongly disagree)
  - 2
  - 3
  - 4
  - 5
  - 6
  - 7 (Strongly agree)
2. The technique stopped moving when I wanted it to.
  - 1 (Strongly disagree)
  - 2
  - 3
  - 4
  - 5
  - 6
  - 7 (Strongly agree)
3. Which of the following is most accurate: The technique moved me
  - 1 (Too slow)
  - 2
  - 3
  - 4 (Just right)
  - 5
  - 6
  - 7 (Too fast)
4. Sharp turns (less than 90 degrees) were easy to complete.
  - 1 (Strongly disagree)
  - 2
  - 3
  - 4
  - 5
  - 6
  - 7 (Strongly agree)
5. Soft turns (greater than 90 degrees) were easy to complete.
  - 1 (Strongly disagree)

- 2
  - 3
  - 4
  - 5
  - 6
  - 7 (Strongly agree)
6. Curved paths were easy to navigate.
- 1 (Strongly disagree)
  - 2
  - 3
  - 4
  - 5
  - 6
  - 7 (Strongly agree)
7. The technique worked as I expected.
- 1 (Strongly disagree)
  - 2
  - 3
  - 4
  - 5
  - 6
  - 7 (Strongly agree)

SECTION III. In this section, you will describe your experience in the Virtual Environment.

1. Please rate your sense of walking in the environment, on a scale of 1 to 7, where 7 represents your natural experience of walking in a real environment.
- 1 (Not natural)
  - 2
  - 3
  - 4
  - 5
  - 6
  - 7 (Very natural)
2. To what extent did the experience within the virtual environment seem like reality for you?
- 1 (Not reality)
  - 2
  - 3
  - 4
  - 5
  - 6
  - 7 (Reality)
3. When you think back to the experience, do you think of the virtual environment more as images that you saw or more as somewhere that you visited?
- 1 (Images)
  - 2



- 3
  - 4
  - 5
  - 6
  - 7 (Somewhere you visited)
4. During the time of your experience, did you often think to yourself that you were actually in the virtual environment?
- 1 (Never in the virtual environment)
  - 2
  - 3
  - 4
  - 5
  - 6
  - 7 (Always in the virtual environment)
5. Consider your memory of being in the virtual environment. How similar in terms of the structure of the memory is this to the structure of the memory of other places you have been today? By “structure of the memory” consider things like the extent to which you have a visual memory of the virtual environment, whether that memory is in color, the extent to which the memory seems vivid or realistic, its size, location in your imagination, the extent to which it is panoramic in your imagination, and other such structural elements.
- 1 (Not similar)
  - 2
  - 3
  - 4
  - 5
  - 6
  - 7 (Very similar)
6. During the time of the experience, which was the strongest on the whole, your sense of being in the virtual environment or of being in the real-world lab?
- 1 (In the lab)
  - 2
  - 3
  - 4
  - 5
  - 6
  - 7 (in the virtual environment)

SECTION IV. In the next series of questions, you will describe your background as it relates to this experiment. Questions 2 to 4 are asked so that we can describe the diversity of our participants in our reports to our funding agency, the National Institutes of Health.

1. How old are you?
2. Are you:
  - Male
  - Female

- I prefer not to answer
3. Are you of Hispanic or Latino descent?
    - Yes
    - No
    - I prefer not to answer
  4. Race/ethnicity: (select the ONE option that fits you best)
    - American Indian or Alaskan Native
    - Asian (includes Philippine Islanders)
    - Black or African-American
    - Native Hawaiian or other Pacific Islander
    - White
    - More than one race
    - Unknown or prefer not to answer
  5. Which eye is your dominant eye? If you do not know which eye is dominant, please ask the experimenter how to determine it.
    - Left eye
    - Right eye
  6. Over the past two years, what is the most you have played video games in a single week? Select one:
    - Less than 1 hour
    - 1 - 2 hours
    - 2 - 5 hours
    - 5 - 10 hours
    - 10 - 20 hours
    - 20 - 40 hours
    - 40 or more hours
  7. In what percent of your computer game-playing do you use a gamepad or joystick to move around in the environment? Select one:
    - 0% (Never)
    - 20%
    - 40%
    - 60%
    - 80%
    - 100% (Always)
  8. How many times have you experienced a virtual environment using a head-mounted display or a CAVE-like system? Select one:
    - 0 (Never)
    - 1
    - 2
    - 3
    - 4
    - 5
    - 6 (or more) times
  9. When was the last time you experienced a virtual environment using a head-mounted display or CAVE-like system? Select one:
    - This week

- Within the last month
- Between 1 month and a year
- More than a year ago
- Never

SECTION V. Free responses.

1. Please tell us what you thought of the locomotion technique that you used. For instance, anything you noticed that was particularly like, or unlike, really walking.
  - (Big box to write in.)
2. Please provide feedback on the experiment and your experience in the virtual environment.
  - (Big box to write in.)



# BIBLIOGRAPHY

- Allison, R. (2008). Personal communication.
- Arechavaleta, G., Laumond, J.-P., Hicheur, H., and Berthoz, A. (2006). The nonholonomic nature of human locomotion: a modeling study. In *Biomedical Robotics and Biomechatronics, 2006 (IEEE BioRob 2006)*, pages 158–163.
- Arthur, K. W. (2000). *Effects of Field of View on Performance with Head-Mounted Displays*. PhD thesis, University of North Carolina at Chapel Hill.
- Blackburn, T. (2008). Personal communication. Defining Locomotion for dissertation.
- Bouguila, L., Ishii, M., and Sato, M. (2002). Realizing a new step-in-place locomotion interface for virtual environment with large display systems. In *Proceedings of the workshop on virtual environments 2002*, volume 23, pages 197–207. ACM.
- Bowman, D., Kruijff, E., LaViola, J. J., and Poupyrev, I. (2005). *3D User Interfaces: Theory and Practice*. Addison-Wesley Professional.
- Brooks, Jr., F. P. (1987). Walkthrough—a dynamic graphics system for simulating virtual buildings. In *Proceedings of the 1986 Workshop on Interactive 3D graphics '86*. ACM Press.
- Brooks, Jr., F. P., Airey, J., Alspaugh, J., Bell, A., Brown, R., Hill, C., Nimscheck, U., Rhengans, P., Rohlf, J., Smith, D., Turner, D., Varshney, A., Wang, Y., Weber, H., and Yuan, X. (1992). Six generations of building walkthroughs – final technical report to the national science foundation. Technical Report TR92-026, Department of Computer Science, The University of North Carolina at Chapel Hill.
- Chai, J. and Hodgins, J. K. (2005). Performance animation from low-dimensional control signals. In *Proceedings of ACM SIGGRAPH 2005*, volume 24, pages 686–695.
- Darken, R. P., Cockayne, W. R., and Carmein, D. (1997). The omni-directional treadmill: a locomotion device for virtual worlds. In *Proceedings of the 10th annual ACM symposium on User interface software and technology*, pages 213–221.
- Dean, G. A. (1965). An analysis of the energy expenditure in level and grade walking. *Ergonomics*, 8(1):31–47.
- Feasel, J. (2007). Personal communication.
- Feasel, J., Wendt, J. D., and Whitton, M. C. (2008). LLCM-WIP: Low-latency, continuous-motion walking-in-place. In *Proceedings of the IEEE 3D User Interfaces (3DUI '08)*, pages 97–104.
- Feasel, J., Whitton, M. C., Kassler, L., Brooks, Jr., F. P., and Lewek, M. D. (2010). The integrated virtual environment rehabilitation treadmill system. In Review.
- Field, A. and Hole, G. (2003). *How to Design and Report Experiments*. SAGE Publications Ltd.

- Finley, F. R. and Cody, K. A. (1970). Locomotive characteristics of urban pedestrians. *Archives of Physical Medicine and Rehabilitation*, 51:423–426.
- GAITRite (2010). GAITRite systems. <http://www.gaitrite.com/Products/index.html>.
- Gibson, J. J. (1986). *The Ecological Approach to Visual Perception*. Lawrence Erlbaum.
- Hillaire, S., Lecuyer, A., Cozot, R., and Casiez, G. (2008). Using an eye-tracking system to improve camera motions and depth-of-field blur effects in virtual environments. In *Proceedings of the IEEE Virtual Reality, 2008*.
- Inman, V. T., Ralston, H. J., and Todd, F. (1981). *Human walking*. Williams and Wilkins.
- Innovative Sports Training (2010). Motionmonitor version 8. <http://www.innsport.com/>. Retrieved January 2010.
- Interrante, V., Ries, B., and Anderson, L. (2007a). Seven league boots: A new metaphor for augmented locomotion through moderately large scale immersive virtual environments. In *IEEE Symposium of 3D User Interfaces 2007*, pages 167–170.
- Interrante, V., Ries, B., Lindquist, J., and Anderson, L. (2007b). Elucidating the factors that can facilitate veridical spatial perception in immersive virtual environments. In *Proceedings of the IEEE Virtual Reality 2007*, pages 11–18.
- Iwata, H. (1999). Walking about virtual environments on an infinite floor. In *Proceedings of the IEEE Virtual Reality, 1999*, pages 286–293.
- Iwata, H., Yano, H., Fukushima, H., and Noma, H. (2005). Circulafloor. *IEEE Computer Graphics and Applications*, 25(1):64–67.
- Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the SME – Journal of Basic Engineering*, 82:35–45.
- Kennedy, R. S., Lane, N. E., Berbaum, K. S., and Lilienthal, M. G. (1993). Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness. *The International Journal of Aviation Psychology*, 3(3):203–220.
- Lecuyer, A., Burkhardt, J.-M., Henaff, J.-M., and Donikian, S. (2006). Camera motions improve the sensation of walking in virtual environments. In *Proceedings of the IEEE Virtual Reality 2006*, pages 11–17.
- Lee, D. N. (1980). *Tutorials in Motor Behavior*, chapter Visuo-Motor Coordination in Space-Time, pages 281–295. North-Holland Publishing Company.
- Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares. *The Quarterly of Applied Mathematics*, 2(2):164–168.
- Liu, G., Zhang, J., Wang, W., and McMillan, L. (2006). Human motion estimation from a reduced marker set. In *Proceedings of the ACM 2006 Symposium on Interactive 3D Graphics and Games*, pages 35–42.
- Marquardt, D. (1963). An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, 11:431–441. Wikipedia condenses Levenberg-Marquardt optimization concisely.

- Meehan, M., Insko, B., Whitton, M., and Frederick P. Brooks, J. (2002). Physiological measures of presence in stressful virtual environments. In *Proceedings of the ACM SIGGRAPH 2002*, pages 645–652.
- Molen, N. H. and Rozendal, R. H. (1966). Some factors of human gait. *Koninklijke Nederlandse Akademie van Wetenschappen*, 69:522–527.
- Molen, N. H., Rozendal, R. H., and Boon, W. (1972). Fundamental characteristics of human gait in relation to sex and location. *Koninklijke Nederlandse Akademie van Wetenschappen*, 75:215–223.
- Murray, M. P., Drought, A. B., and Kory, R. C. (1964). Walking patterns of normal men. *Journal of Bone and Joint Surgery*, 46-A(2):335–360.
- Murray, M. P., Kory, R. C., Clarkson, B. H., and Sepic, S. B. (1966). Comparison of free and fast speed walking patterns of normal men. *American Journal of Physical Medicine*, 45(1):8–24.
- Murray, M. P., Kory, R. C., and Sepic, S. B. (1970). Walking patterns of normal women. *Archives of Physical Medicine and Rehabilitation*, 51:637–650.
- Nitzsche, N., Hanebeck, U. D., and Schmidt, G. (2004). Motion compression for telepresent walking in large target environments. *Presence: Teleoperators and Virtual Environments*, 13(1):44–60.
- Öberg, T., Karsznia, A., and Öberg, K. (1993). Basic gait parameters: Reference data for normal subjects. *Journal of Rehabilitation Research*, 30(2):210–223.
- Peck, T. C., Fuchs, H., and Whitton, M. C. (2010). Improved redirection with distractors: A large-scale-real-walking locomotion interface and its effect on navigation in virtual environments. In *Proceedings of the IEEE Virtual Reality 2010*.
- Peck, T. C., Whitton, M. C., and Fuchs, H. (2008). Evaluation of reorientation techniques for walking in large virtual environments. In *Proceedings of the IEEE Virtual Reality 2008*.
- Razzaque, S. (2005). *Redirected Walking*. PhD thesis, Department of Computer Science, The University of North Carolina at Chapel Hill, <http://www.cs.unc.edu/Publications/PHDAbstracts.html#Razzaque>. Retrieved May 2006.
- Razzaque, S. (2007). Personal communication.
- Razzaque, S., Swapp, D., Slater, M., Whitton, M. C., and Steed, A. (2002). Redirected walking in place. In Mueller, S. and Stuezlinger, W., editors, *Proceedings of the 8th Eurographics Workshop on Virtual Environments*, pages 123–130.
- Rose, J. and Gamble, J. G., editors (2006). *Human Walking (3rd ed.)*. Lippincott, Williams & Wilkins.
- Ruddle, R. A. and Lessels, S. (2009). The benefits of using a walking interface to navigate virtual environments. *ACM Transactions on Computer-Human Interaction*, 16(1):5:1–5:18.

- Slater, M. (2007). Personal communication.
- Slater, M., Steed, A., McCarthy, J., and Maringelli, F. (1998). The influence of body movements on subjective presence in virtual environments. *Human Factors*, 40(3):469–477.
- Slater, M., Steed, A., and Usoh, M. (1995a). The virtual treadmill: a naturalistic metaphor for navigation in immersive virtual environments. In *Selected papers of the Eurographics workshops on Virtual Environments '95*, pages 135–148.
- Slater, M. and Usoh, M. (1994). *Artificial Life and Virtual Reality*, chapter Body Centred Interaction in Immersive Virtual Environments, pages 125–147. John Wiley and Sons.
- Slater, M., Usoh, M., and Steed, A. (1995b). Taking steps: The influence of a walking technique on presence in virtual reality. *ACM Trans. on Computer-Human Interaction*, 2(3):201–219.
- Smith, S. W. (1997). *Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Publishing.
- Steed, A. (2008). Personal communication.
- Stoakley, R., Conway, M. J., and Pausch, R. (1995). Virtual reality on a WIM: Interactive worlds in miniature. In *ACM CHI'95 Conference on Human Factors in Computer Systems*.
- Suma, E., Finkelstein, S., Clark, S., and Wartell, Z. (2009). An approach to redirect walking by modifying virtual world geometry. In *Proceedings of the IEEE VR 2009 Workshop on Perceptual Illusions in Virtual Environments*, pages 16–18.
- Taylor II, R. M., Hudson, T. C., Seeger, A., Weber, H., Juliano, J., and Helser, A. T. (2001). Vrpn: A device-independent, network-transparent vr peripheral system. In *Proceedings of the ACM Symposium on Virtual Reality Software & Technology (VRST)*, pages 55–61.
- Templeman, J. (2008). Personal communication.
- Templeman, J. N. (2003). United states patent 6646643: “User control of simulated locomotion”.
- Templeman, J. N., Denbrook, P. S., and Sibert, L. E. (1999). Virtual locomotion: Walking in place through virtual environments. *Presence*, 8(6):598–617.
- Templeman, J. N., Sibert, L. E., and Page, R. C. (2006). Immersive simulation to train urban infantry combat. In *Virtual Media for Military Applications*, pages 23.1–23.16. NATO Research Technology Organisation, HFM-136.
- Terziman, L., Lecuyer, A., Hillaire, S., and Wiener, J. M. (2009). Can camera motions improve the perception of traveled distance in virtual environments. In *Proceedings of the IEEE Virtual Reality 2009*, pages 131–134.
- Usoh, M., Arthur, K., Whitton, M. C., Bastos, R., Steed, A., Slater, M., and Brooks, Jr., F. P. (1999). Walking > walking-in-place > flying, in virtual environments. In *Proceedings of the ACM SIGGRAPH 1999*, pages 359–364.



- Virtusphere (2008). Virtusphere – a new locomotion technology for immersing in virtual reality. <http://www.virtusphere.com/>. Retrieved March 2008.
- Welch, G. and Bishop, G. (2006). An introduction to the Kalman Filter. Technical Report TR 95-041, Department of Computer Science, University of North Carolina at Chapel Hill.
- Whitton, M. C., Cohn, J. V., Feasel, J., Zimmons, P., Razzaque, S., Poulton, S. J., McLeod, B., and Brooks, Jr., F. P. (2005). Comparing VE locomotion interfaces. In *Proceedings of the IEEE Virtual Reality 2005*, pages 123–130.
- Whitton, M. C. and Razzaque, S. (2008a). *HCI Beyond the GUI: Design for Haptic, Speech, Olfactory and Other Nontraditional Interfaces*, chapter Locomotion Interfaces. Morgan Kaufmann.
- Whitton, M. C. and Razzaque, S. (2008b). Locomotion interfaces case study (Companion case study to Whitton and Razzaque 2008a). Web supplement to *HCI Beyond the GUI: Design for Haptic, Speech, Olfactory and Other Nontraditional Interfaces*. Retrieved April 7, 2010 from [http://books.elsevier.com/companions/9780123740175/casestudies/Case\\_Study\\_04.pdf](http://books.elsevier.com/companions/9780123740175/casestudies/Case_Study_04.pdf).
- Williams, B., Narasimham, G., McNamara, T. P., Carr, T. H., Rieser, J. J., and Bodenheimer, B. (2006). Updating orientation in large virtual environments using scaled translational gain. In *Proceedings of the 3rd ACM symposium on Applied perception in graphics and visualization*, pages 21–28. ACM.
- Williams, B., Narasimham, G., Rump, B., McNamara, T. P., Carr, T. H., Rieser, J., and Bodenheimer, B. (2007). Exploring large virtual environments with an hmd when physical space is limited. In *Proceedings of the 4th ACM symposium on Applied perception in graphics and visualization*, pages 41–48.
- Wuttke, J. (2009). lmfit – a C/C++ routine for Levenberg-Marquardt minimization with wrapper for least-squares curve fitting, based on work by B. S. Garbow, K. E. Hillstrom, J. J. More, and S. Moshier. version 2.5. <http://www.messen-und-deuten.de/lmfit/>. Retrieved September 2009.
- Yan, L., Allison, R. S., and Rushton, S. K. (2004). New simple virtual walking method - walking on the spot. In *8th Annual Immersive Projection Technology (IPT) Symposium Electronic Proceedings*.
- Zarrugh, M. Y., Todd, F. N., and Ralston, H. J. (1974). Optimization of energy expenditure during level walking. *European Journal of Applied Physiology and Occupational Physiology*, 33:293–306.