SEOUL NATIONAL UNIVERSITY

BACHELOR THESIS

Reconstructing locomotion in VR from WIP (Walking-In-Place) motion : an IMU-based, inside-out approach

Author:
JiGang KIM

Supervisor:
Prof. Frank C. PARK

A thesis submitted in fulfillment of the requirements for the degree of Bachelor of Engineering

in the

Robotics Laboratory
Department of Mechanical & Aerospace Engineering

December 21, 2017

Declaration of Authorship

I, JiGang KIM, declare that this thesis titled, "Reconstructing locomotion in VR from WIP (Walking-In-Place) motion : an IMU-based, inside-out approach" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

SEOUL NATIONAL UNIVERSITY

Abstract

Department of Mechanical & Aerospace Engineering

Bachelor of Engineering

Reconstructing locomotion in VR from WIP (Walking-In-Place) motion : an IMU-based, inside-out approach

by JiGang KIM

WIP (Walking-In-Place) motion based VR (Virtual Reality) locomotion techniques simulate horizontal motion in virtual environments from vertical foot motion. Due to its similarity to real walking, WIP provides intuitive and immersive user interface for navigating vast virtual environments within the confines of limited physical space. Numerous past literatures on this topic implemented costly and complex systems to demonstrate their algorithms. The proposed approach requires a simple setup of two IMU (Inertial Measurement Unit) sensor modules attached to each foot and a Android phone capable of running Google Cardboard. With the proposed setup it is possible to track WIP motion as well as walking motion. This has been possible by incorporating key ideas from PDR (Pedestrian Dead Reckoning) algorithms, which aim to accurately track the position of a pedestrian without receiving any information from external sources. This paper presents a low-cost, natural interface for navigating VEs by synthesizing a personalized locomotion from WIP motion with offline gait analysis.

KEYWORDS: Walking-In-Place (WIP), Inside-Out, Virtual Reality (VR), Virtual Environment (VE), Virtual Locomotion, Inertial Measurement Unit (IMU)

Acknowledgements

This research has been funded by Korea Institute of Science and Technology (KIST). Special thanks to Dr. Kim and Dr. Moon for their advice and assistance and Mr. Jeong for providing source code and assets for Unity client and TCP server.

Contents

Declaration of Authorship **i**

Abstract **ii**

Acknowledgements **iii**

1 Introduction **1**
1.1 State of VR 1
1.2 Gait Analysis 1
1.2.1 Gait Event Detection 2
1.2.2 Gait Feature Extraction 3
1.3 Locomotion Generation in VR 4
1.3.1 WIP Techniques 5
1.4 Sensor Fusion Orientation Filter 6

2 Methods **7**
2.1 System Setup 7
2.2 Offline Gait Analysis 8
2.2.1 Event Detection 8
2.2.2 Tracking 9
2.3 Real-time WIP Tracking 10
2.3.1 FSM 10
2.3.2 Position Tracking 11
2.4 Locomotion Generation 11
2.4.1 Direct Approach 12
2.4.2 Indirect Approach 13
2.4.3 Heading 13

3 Results and Discussion **14**
3.1 Offline Gait Analysis 14
3.1.1 Event Detection 14
3.1.2 Trajectory 15
3.2 Locomotion Generation 16
3.2.1 WIP Event Detection 16
3.2.2 WIP Sessions 17

4 Conclusion **21**

Bibliography **22**

List of Figures

1.1 Gait event definition 2
1.2 Taxonomy of virtual travel techniques 4

2.1 System overview 7
2.2 Sensor module with custom brackets 8
2.3 Gait event detection 8
2.4 Kitagawa's method 9
2.5 FSM diagram for WIP motion 10
2.6 Sample velocity profile for walking and WIP motion 12

3.1 Gait event detection results 14
3.2 Gait feature extraction results 15
3.3 WIP event detection results 16
3.4 WIP event detection with quaternion decomposition 17
3.5 WIP session 1 18
3.6 WIP session 2 18
3.7 WIP session 3 19
3.8 WIP session 4 19
3.9 Demo 20

Chapter 1

Introduction

1.1 State of VR

In recent years VR (Virtual Reality) has become widely accessible with the advent of sub-1000$, enthusiast-grade headsets such as the HTC VIVE and the Occulus Rift, followed by the adoption from a major console platform PlayStation with the PSVR and a slew of low-cost, smartphone-based mobile offerings from startups. On the software front Microsoft is working on Windows Mixed Reality and the two major mobile OSes Android and iOS both have launched developer kits for AR (Augmented Reality) and VR.

Whereas in the past there has been a focus on cutting down cost with breakthroughs in hardware, inside-out VR has become an emerging trend this year. This is evident from CES (Consumer Electronics Show) 2017 with many companies showcasing their inside-out VR solutions. The term inside-out refers to how the headset position is tracked. In contrast to outside-in VR which require external sensor arrays for point of reference, inside-out VR only uses on-board sensor without the need of external systems. This is achieved by monocular or binocular visual SLAM (Simultaneous Localization and Mapping) complemented with IMU (Inertial Measurement Unit) sensors. The technology is expected to trickle-down to consumers in few years.

1.2 Gait Analysis

Gait analysis is a heavily researched topic in many disciplines. From medical sciences including physiology, pathology and gerontology to computer animation and pedestrian tracking, there has been numerous research on this topic. Medical science is where human gait has been extensively studied and where most of the terminology originates from. Research interests range from enhancing the performance of athletes to diseases prediction and rehabilitation. In computer animation gait analysis is used to achieve natural perceived motion in techniques such as motion blending and motion warping which manipulates existing motion capture data to generate motion. Pedestrian tracking requires gait analysis to detect steps and accurately measure stride length.

1.2.1 Gait Event Detection

Most commonly recognized gait events are heel-strike, foot-flat, heel-off and toe-off which is well represented in a figure from a book by Whittle, 2006 (Figure 1.1). Heel-strike is when the foot impacts the ground followed by foot-flat when the foot has made full contact with the ground. Subsequent events heel-off and toe-off occurs when the foot is detached from the ground heel to toe.

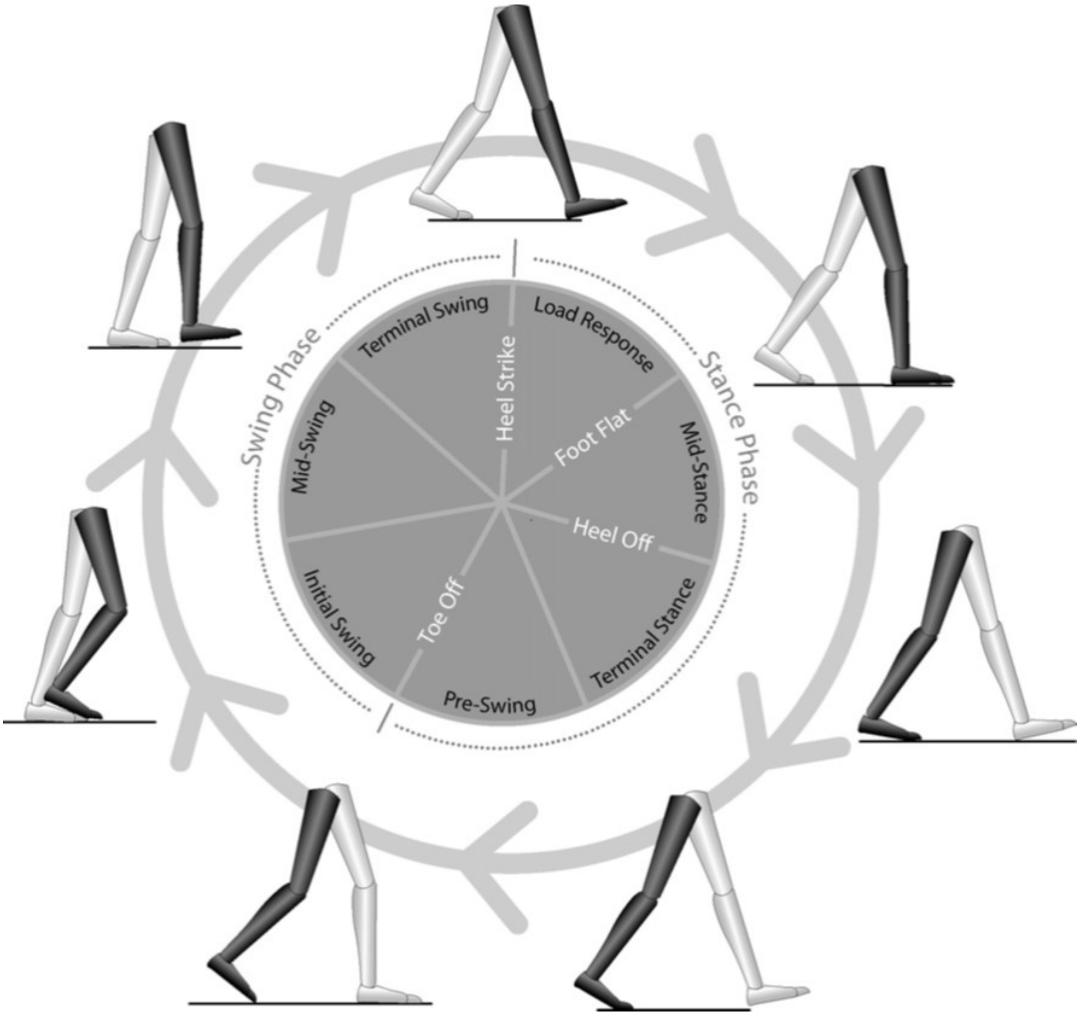

FIGURE 1.1: Gait events as defined in Whittle, 2006

There are various gait event detection methods with different types of sensors and sensor configurations. This is well-documented in a meta-analysis by Rueterbories et al., 2010. In most cases, recent research on gait analysis utilize either force-based sensor, IMU sensor, magnetometer or some combination of aforementioned devices to detect gait events. While it is also possible to obtain gait events from motion capture systems such as the VICON or from depth cameras, these devices are overly sophisticated for the sole purpose of gait event detection.

Force-based sensors which measure ground reaction force, range from simple on-off switches to pressure sensitive sensors either in a standalone mat form factor or embedded within the shoe. MEMS (Micro-Electro-Mechanical Systems) IMU sensors and magnetometers are often packaged onto a single board and measure acceleration, angular velocity and magnetic field. Acceleration data is well suited for detecting shocks resulting from contact such as the heel-strike event. For functional analysis angular velocity data is preffered as angular velocity magnitude is invariant to sensor placement, under the reasonable assumption that the sensor is attached to a rigid body (i.e. shoe). Magnetometers can be prone to local magnetic disturbances, especially so in an indoor settings and must be used with caution. Detailed description of gait event detection using IMU sensors is presented by Jasiewicz et al., 2006.

Different techniques can be deployed to detect gait events from sensor data. Rueterbories et al., 2010 presents a summary of commonly used gait detection methods. With force-based sensors simple thresholding can get the job done. In many cases, functional analysis of either raw sensor data or derived data coupled with FSM (Finite State Machine) can provide reliable gait event detection. With the recent explosion of various learning techniques, data-driven approach can outperform traditional techniques in some cases.

1.2.2 Gait Feature Extraction

Gait features can be classified into two categories: spatio-temporal and kinematic. The former is represented by a single parameter and the latter is represented by a time series or a waveform. Spatio-temporal features are parameters pertaining to space (spatial) and time (temporal). Spatial parameters include but not limited to step length, stride length, clearances (e.g. MTC, minimum toe clearance), maximum and minimum heights. Temporal parameters include but not limited to cadence, speed and duration between gait events. Kinematic features can be any time series or waveform related to gait that may or may not be represented by wavelets. Kinematic features include but not limited to time series of joint angles, limb angular velocity and limb acceleration.

Focusing on IMU-based gait feature extraction techniques, the traditional approach is to obtain gravity-compensated accelerometer data represented in the ground frame with the knowledge of sensor orientation and perform successive integration with constraints provided by gait events to obtain drift-compensated velocity and position (Rampp et al., 2015; Kitagawa and Ogihara, 2016). Recently there has been data-driven approach to extract gait features directly from IMU sensor data with deep convolutional neural networks (Hannink et al., 2017). If the successive stride vectors can be reliably measured and the initial position is given, PDR (Pedestrian Dead Reckoning) is possible. A detailed overview on this topic is provided by Woodman, 2010. State of the art PDR algorithm considers heel-strike and toe-off phase and improves upon conventional INS-EKF-ZUPT (Inertial Navigation System - Extended Kalman Filter - Zero Velocity Update) algorithm (Ju et al., 2016).

For the purpose of this research, spatio-temporal parameters are needed. Kitagawa and Ogihara, 2016 presents a reliable method to obtain foot trajectory with IMU sensors only. It relies on double integration of the gravity-compensated and orientaion-corrected acceleration with some strong assumptions on the underlying motion to eliminate drift. For gait feature extraction, this research largely adopts the method described in Kitagawa and Ogihara, 2016.

1.3 Locomotion Generation in VR

There are various locomotion generation schemes depending on the input used and the system setup. Inputs include but not limited to gamepad or joystick inputs, headset positional tracking, gesture inputs such as tapping, gaze inputs in stare-to-move and look-down-to-move techniques, WIP (Walking-In-Place) and RDW (Redirected Walking, Razzaque, Kohn, and Whitton, 2001). A brief taxonomy of techniques is presented by Nilsson, Serafin, and Nordahl, 2016 (Figure 1.2).

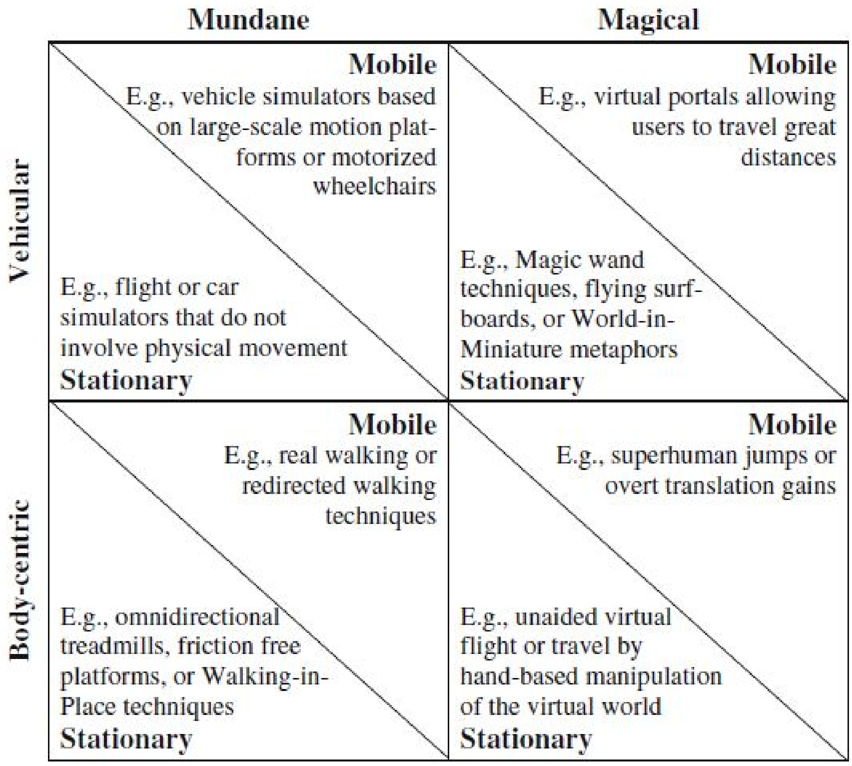

FIGURE 1.2: Taxonomy of virtual travel techniques presented by Nilsson, Serafin, and Nordahl, 2016

One of the strong selling points of WIP techniques over headset positional tracking is that the user can navigate infinite virtual space within a limited physical space. Compared with other techniques, WIP techniques provide a interface that is most similar to real walking which is more natural and may alleviate VR sickness. One can argue that RDW which extends the size of the virtual space from a given physical space while retaining the benefits of real walking strikes a nice balance between WIP techniques and headset positional tracking. However, virtual environments have to be designed in such a way that the user following a slightly curved path is convinced to be walking in a straight line without inducing VR sickness. This research seeks to improve upon existing WIP techniques by achieving similar performance with low-cost, inside-out, lightweight IMU-based setup.

1.3.1 WIP Techniques

Earlier attempts to synthesize locomotion in VR from WIP motion date back to the 1990s. Surprisingly, earlier attempts include using feed-forward neural networks to determine whether or not the participant is performing WIP motion (Slater, Usoh, and Steed, 1995). Though the reported accuracy of around 90% may not be adequate for pratical use. In the last decade there has been resurgence of research on this topic. Feasel, Whitton, and Wendt, 2008 focused on improving two performance criterion - latency and smoothness (LLCM-WIP, Low-Latency-Continuous-Motion WIP). This was followed by GUD-WIP (Gait-Understanding-Driven WIP) which aimed to generate actual walking-like velocity profile with the knowledge of gait characteristics (Wendt, Whitton, and Brooks, 2010). Bruno, Pereira, and Jorge, 2013 improved upon GUD-WIP by using maximum heel height (amplitude) as a metric of locomotion speed and claims less travel time over long distances and improved accuracy in short distances when compared to GUD-WIP (SAS-WIP, Speed-Amplitude-Supported WIP).

LLCM-WIP emphasizes low starting/stopping latency and smoothness of the generated locomotion. Heel height signal of left and right foot is conditioned with numerical differentiation, smoothing, DC-bias removal, scaling, etc. which is then summed to be the locomotion speed. Heel height is obtained from magnetic foot trackers. Orientation of the locomotion velocity vector is determined by the chest-orientation tracker. Such implementation, as mentioned by the authors, is prone to magnetic disturbances.

Whereas LLCM-WIP relied on signal conditioning of heel height to generate locomotion, GUD-WIP analyzes gait event sequences of walking motion and WIP motion and comes up with a rather sophisticated FSM-based locomotion scheme. The authors claim a smoother, more natural locomotion over LLCM-WIP. However, due to the inherent limitation of state machines, stopping latency is worse than that of LLCM-WIP. GUD-WIP was implemented with optical trackers.

SAS-WIP attempts to improve the user experience of the WIP technique by proving the user with more control over locomotion speed which results in reduced user fatigue and better accuracy. Performance is improved (faster long distance travel and more precise short distance travel) over GUD-WIP. SAS-WIP was also implemented with optical trackers.

Recently smartphone-based mobile solutions have emerged. Such solutions require minimal hardware and while they may not be as sophisticated as solutions with dedicated hardware, they are more than capable of performing simple functions. Tregillus and Folmer, 2016 synthesizes gaze-directed locomotion when resulting head bobbing motion from WIP triggers an event. Numerous related software packages exist such as the assets found on the Unity Asset Store with varying degrees of success.

1.4 Sensor Fusion Orientation Filter

Orientation can be obtained from multiple sensor data. Accelerometer data provide pitch and roll angles by measuring the direction of gravitational acceleration. Gyroscope data provide orientation relative to initial configuration by integrating angular velocity. Magnetometer data provide a reference direction for yaw angle. Due to sensor characteristics, each has its own strengths and weaknesses. Orientation obtained from accelerometer data is fast and responsive but cannot provide yaw angle and also prone to disturbances such as linear acceleration. Gyroscope data can provide rotations in all directions but is prone to drifting from the integration process. Magnetometer data provide reference heading but more often than not magnetic disturbances, especially in indoor settings, require special consideration.

It is common to combine the data from various sensors with sensor fusion algorithms to obtain responsive and robust orientation. IMU-based sensor fusion algorithms require accelerometer and gyroscope data and is used when absolute heading is not required and slight yaw drift is allowed. MARG (Magnetic Angular Rate Gravity) filters incorporate magnetometer data in addition to IMU data to determine the absolute heading but may under-perform compared to IMU filters under high magnetic noise environments. These filters can be implemented with Kalman filters or their variants which is the most widely used approach. Kalman filters require matrix computation which can be computationally expensive. Quaternion-based, gradient descent based orientation filter has been suggested to overcome this computational disadvantage (Madgwick, Harrison, and Vaidyanathan, 2011). In this research, Madgwick's algorithm implemented in C was repackaged into C++ class and used to obtain orientation quaternion from IMU sensor data.

Chapter 2

Methods

2.1 System Setup

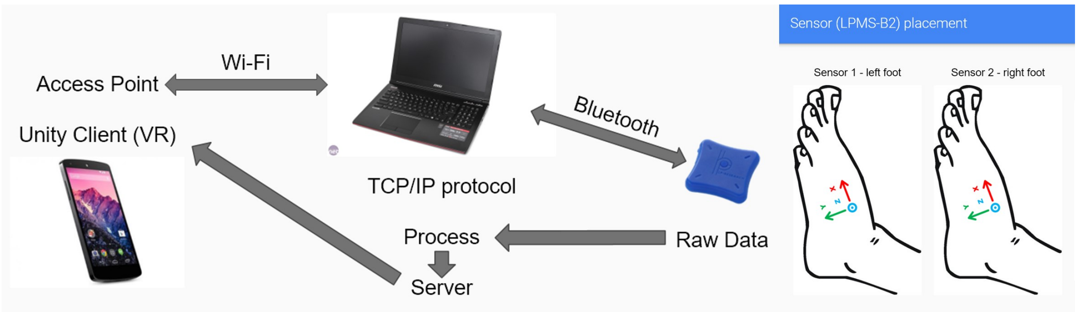

FIGURE 2.1: System overview

System overview is represented in the Figure 2.1. System setup consists of a smartphone running Android 6.0 Marshmallow (Nexus 5), a PC running Windows 10 (MSI GE62) and two IMU sensors (LPMS-B2) mounted on each foot as shown in Figure 2.2. For the purpose of this research, IMU sensors are configured to run at 100Hz sampling rate with gyroscope range of $2000°/s$ and accelerometer range of 8G. Though it should be noted that sampling rate of 200Hz is achievable under current configuration. Refer to the datasheet from the manufacturer for detailed specifications. PC acts as a central hub, processing the data received from the IMU sensors and sending the virtual velocity to the smartphone (VR headset). PC and smartphone is connected by Wi-Fi with smartphone providing connection via hotspot and IMU sensors are connected to PC with Bluetooth. Software was developed on Visual Studio 2015 and tested on Unity 5.6.1f1 with sample VR scenes from Udacity. Google Cardboard enables Unity VR projects to run on smartphones running Android 4.4 KitKat or above.

There are three processes - WIP client, Unity client and server. Algorithm presented in this paper is implemented in the WIP client which processes data from IMU sensors and generates virtual velocity. Unity client receives said velocity from the WIP client and displays the virtual environment accordingly. Server maintains a communication session between WIP client and Unity client. WIP client and server runs on the PC and Unity client runs on the smartphone. WIP client is implemented in C++ and the rest is implemented in C#.

FIGURE 2.2: Sensor module with custom brackets

2.2 Offline Gait Analysis

Offline gait analysis consist of two parts - event detection and tracking. Gait event detection is implemented with custom FSM (Finite State Machine) with four states and five transitions. Gait tracking borrows from Kitagawa and Ogihara, 2016 with minor modifications. Gait event detection was not necessary in implementing WIP technique and has been used as a reference only. Kitagawa's algorithm is implemented in the WIP client and executed with 'kita' command (on pre-recorded dataset) or 'calibrate' command (on current recording session).

2.2.1 Event Detection

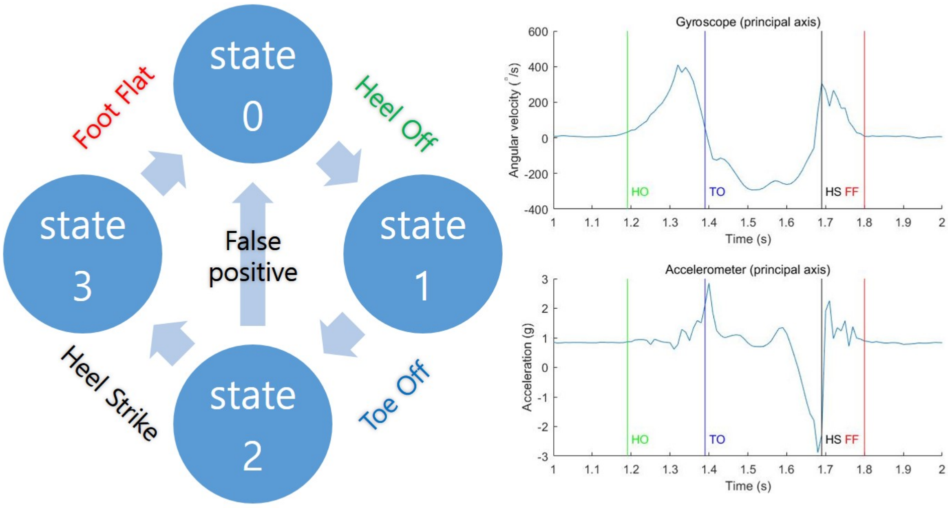

FIGURE 2.3: Gait event detection (Right: FSM diagram, Left: gait events superimposed on acceleration and angular velocity along the principal axis)

Gait event detection relies on gyroscope data only. This is because acceleration can vary depending on sensor placement but angular velocity is independent of sensor placement under the rigid body assumption. Angular velocity along the principal axis, which is defined as the one of the three sensor axes with the largest range of angular velocity, was used to determine the states. This provides better results when compared with using the magnitude of angular velocity - which ensures coordinate invariance - without adding too much complexity. For optimal results it is recommended to roughly align one of the sensor axes to the ankle joint axis.

FSM was devised with functional analysis of the principal axis angular velocity time series (Figure 2.3). Initial state (i.e. foot is stationary) is state 0. Transition from state 0 to state 1 occurs when heel-off event is detected. Heel-off event is defined as when the angular velocity exceeds $30°/s$ and is increasing with time. Toe-off event, which is defined as when zero crossing has occurred and the time derivative is less than $-2000°/s^2$, triggers the transition from state 1 to state 2. Two transitions are defined for state 2. Transition to state 3 occurs when heel-strike is detected. Heel-strike is defined as when zero crossing in the other direction has occurred and the time derivative is greater than $2000°/s^2$. Transition to state 0 occurs when there is a false positive and gait is not recognized. This condition is satisfied to be when the angular velocity is greater than zero for certain time steps prior to heel-strike. Foot-flat event, which is defined as when angular velocity is below $10°/s$, triggers the transition from state 3 to state 0. The sign of the principal angular velocity might have to be flipped to match the graph in Figure 2.3 if the sensor placement does not follow the one shown in Figure 2.1.

2.2.2 Tracking

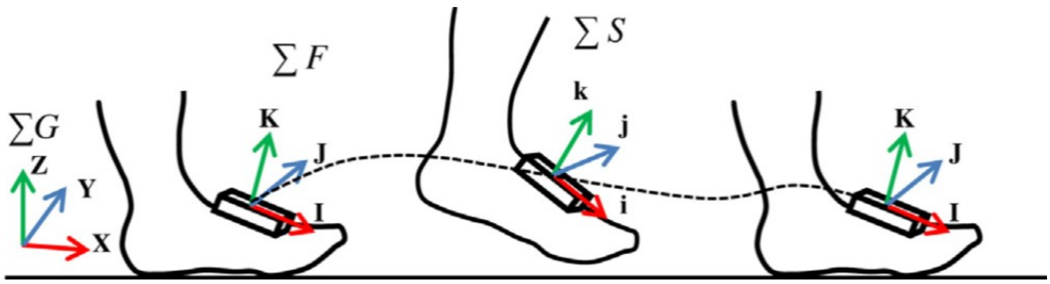

FIGURE 2.4: Figure from Kitagawa and Ogihara, 2016. Ground frame G is fixed to the ground. Sensor frame S is fixed to the sensor and F is the sensor frame at foot-flat.

Gait tracking algorithm follows the method described in Kitagawa and Ogihara, 2016 but implemented with quaternions instead of rotation matrices for coordinate transformations. Kitagawa's method performs double integration with velocity and position drift correction for accurate gait trajectory tracking. From the gait trajectory, spatio-temporal gait features can be extracted. For detailed description of the algorithm, refer to Kitagawa's paper. A summary is provided below.

1. Find foot-flat and foot-off time with simple thresholding of angular velocity to obtain T_0 and T_f.
2. Save acceleration a_{F0} (gravity) and orientation quaternion q_G^F at T_0 .
3. Find linear acceleration in G coordinates between T_0 and T_f.
   a. Represent raw sensor acceleration in F coordinates (a_{Ft}).
   b. Subtract a_{F0} to obtain gravity compensated linear acceleration.
   c. Represent said value in G coordinates.
4. Integrate linear acceleration by time and apply velocity drift correction such that the velocity is zero at T_f.

5. Integrate once more and apply positional drift correction such that the vertical component of position is zero at T_f.

2.3 Real-time WIP Tracking

Real-time WIP tracking can be difficult due to the real-time aspect. Unlike offline analysis where the entire time series is available, real-time applications have to work with data up to the current time point. Moreover, system lag and overall latency also becomes an issue for real-time applications. Specifically for tracking WIP in real-time, estimating velocity drift rate which affects the tracking accuracy of the consequent step is crucial. This can be done by taking into account the history of past velocity drift rates with a feedback loop. With the feedback loop in place complications such as ensuring the stability of the system arise.

2.3.1 FSM

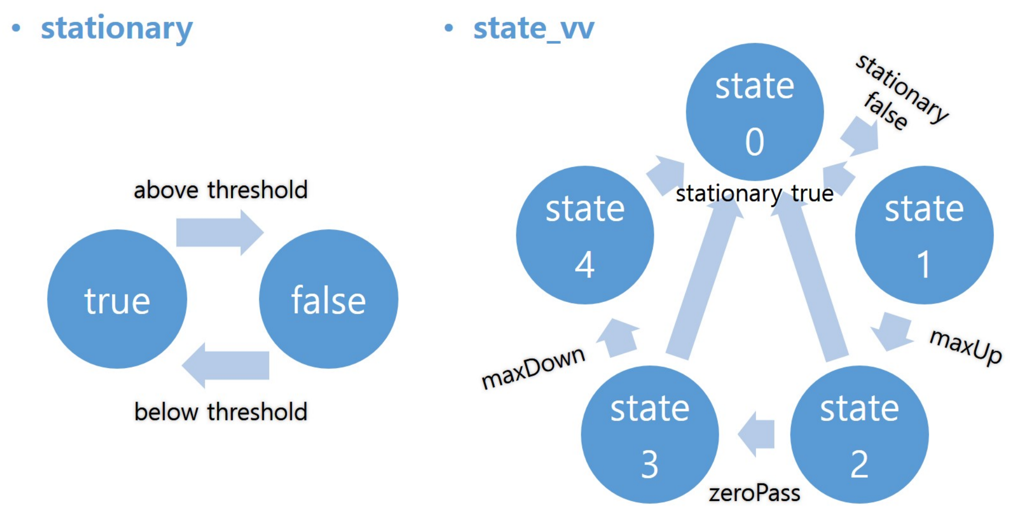

FIGURE 2.5: FSM diagram for WIP motion (Left: stationary, Right: state_vv)

There are two FSMs for real-time WIP motion tracking as shown in Figure 2.5. FSM for stationary utilizes simple thresholding for state transition. Thresholding is performed on band-pass filtered acceleration and low-pass filtered angular velocity. If the filtered values are under the user configurable thresholds, foot is assumed to be stationary. Through trial and error optimal threshold was set as 0.04G and $30°/s$ for acceleration and angular velocity thresholds, respectively. Band-pass filter for acceleration was implemented as subtracting known offset of 1G (gravity) from raw acceleration magnitude which was then processed with first order digital IIR (Infinite Impulse Response) low-pass filter with a cutoff frequency of 5Hz and a sampling rate of 100Hz.

FSM for state_vv was devised with functional analysis of the vertical component of the foot velocity. Under WIP motion, foot travels up and down resulting in a vertical velocity profile that is sine-like. When foot is stationary, state_vv is at state 0. Otherwise state_vv follows a clockwise state transition triggered by events maxUp, zeroPass, maxDown (Figure 2.5). maxUp event is when foot reaches maximum upwards velocity, followed by zeroPass event when the foot reaches the apex and heads back down. maxDown event is when foot reaches maximum downwards velocity.

2.3.2 Position Tracking

The basic idea behind real-time WIP position tracking algorithm is similar to Kitagawa's method used for gait tracking. However, the real-time aspect introduces some challenges, particularly with drift correction and lag from application of filters in the FSM. Pseudocode of the algorithm is presented below.

1. Represent the raw acceleration measured in sensor frame in ground coordinates and subtract gravity (0, 0, 1)G to obtain gravity compensated linear acceleration a_G.
2. During non-stationary periods integrate velocity drift rate corrected linear acceleration by time to obtain velocity in ground coordinates v_G.
3. Integrate once more during non-stationary periods to obtain position in ground coordinates x_G. If the vertical component of position falls below zero said component is set to zero. Also, during stationary periods x_G is set to zero.
4. Let v_G at transition from non-stationary to stationary be drift velocity v_{drift} and the duration of the last non-stationary session be Δt. Then the velocity drift rate is updated by the following equation in feedback form.

$$\dot{v}_{drift} = \dot{v}_{drift} + v_{drift}/\Delta t$$

When compared with Kitagawa's method there is a minor difference in deriving gravity compensated linear acceleration. This is because the definition of T_0 is complicated to implement in a real-time fashion. Major difference is in drift correction. With real-time tracking it is impossible to know the exact the value of the velocity drift rate beforehand, thus velocity drift rate from the previous session is assumed as the velocity drift rate. This results in reduced accuracy when compared with Kitagawa's method and is a trade-off for running in real-time.

2.4 Locomotion Generation

A direct and an indirect approach for locomotion generation scheme has been devised. Direct approach reconstructs horizontal velocity of the foot under normal gait from the vertical position of the foot in WIP motion. Simulated horizontal velocity of each foot is then manipulated to generate locomotion in the virtual environment. This approach is largely inspired by the method described in Feasel, Whitton, and Wendt, 2008 which derives locomotion from signal processing of heel height. Indirect approach generates locomotion from spatio-temporal parameters extracted from WIP motion such as maximum foot height and non-stationary duration. This approach is more in line with recent research such as Bruno, Pereira, and Jorge, 2013 and Tregillus and Folmer, 2016.

2.4.1 Direct Approach

Direct approach is based on the simple assumption of the WIP motion. One of the simple ways of modelling ankle trajectory during gait is assuming a cycloid trajectory $(t - sin(t), 1 - cos(t))$. Then the time derivative of horizontal and vertical component of the cycloid follow $1 - cos(t)$ and $sin(t)$, respectively. Note that the vertical component of the cycloid trajectory and the horizontal component of the time derivative follow the same profile $1 - cos(t)$. Thus, the horizontal velocity of the foot can be reconstructed from vertical position of the foot with an appropriate scale factor. This scale factor can be determined from the results of offline gait analysis. Assuming one-to-one correspondence between vertical components of walking and WIP motion, it is possible to reconstruct horizontal motion from WIP motion. Aforementioned assumptions have been backed with actual data as shown in Figure 2.6. Note that the vertical component of walking is not entirely sine-like which is due to the fact that the sensor was mounted at the dorsum of the foot. With reconstructed horizontal motion for both feet, torso velocity is determined by the average of reconstructed feet speeds.

$$
\begin{aligned}
v_{z,walking} &= V_z sin\left(\frac{2\pi}{T}t\right) \\
v_{r,walking} &= V_r\left[1 - cos\left(\frac{2\pi}{T}t\right)\right] \\
p_{z,walking} = \int v_{z,walking}dt &= \frac{TV_z}{2\pi}\left[1 - cos\left(\frac{2\pi}{T}t\right)\right] \\
scalefactor &= \frac{v_{r,walking}}{p_{z,walking}} = \frac{2\pi V_r}{TV_z}
\end{aligned}
$$

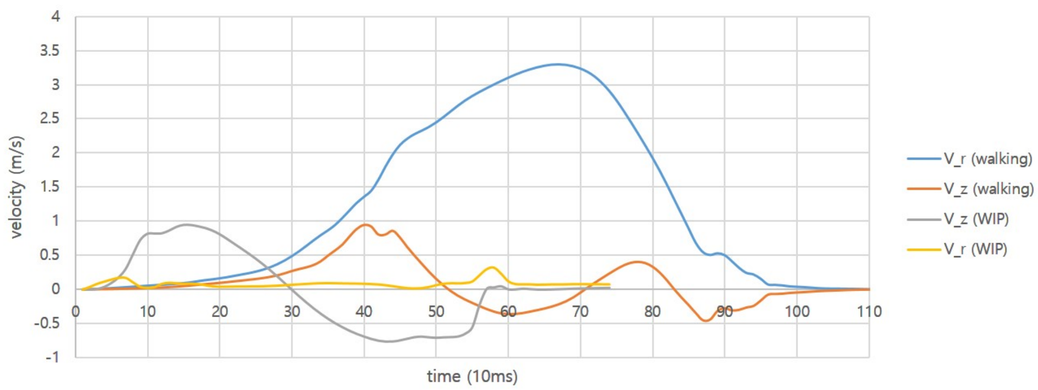

FIGURE 2.6: Sample velocity profile for walking and WIP motion

Practical issues arise during implementation. Simple averaging of reconstructed foot motion results in jerkiness. This is due to brief double stance phase causing the motion to abruptly stop midway. Similar issues were discussed by (Feasel, Whitton, and Wendt, 2008) and was mitigated by adopting a low-pass filter to smooth out the locomotion with minor compromise regarding increased stopping latency. In this paper, extra measures were taken to prevent jerkiness and to keep the stopping latency relatively low. When state_vv is at state 4 ($t_4 \leq t$), extra padding is applied to prevent speed from falling to zero. The equation is provided below.

$$v_{r,reconstructed} = scalefactor \times p_{z,WIP}(t) \qquad (t_1 \leq t \leq t_4)$$

$$v_{r,reconstructed} = 0.5 \times scalefactor \times (p_{z,WIP}(t) + p_{z,WIP}(t_4)) \qquad (t_4 \leq t)$$
$$v_{locomotion} = 0.5 \times (v_{r,reconstructed(left)} + v_{r,reconstructed(right)})$$

2.4.2 Indirect Approach

Unlike direct approach, indirect approach does not utilize the entire time series of the foot vertical position. It only requires maximum foot height and non-stationary duration. Higher maximum foot height and shorter non-stationary duration corresponds to faster locomotion. Maximum foot height and non-stationary duration is updated when either one of the foot is at zeroPass (state_vv transition from state 2 to state 3). The foot is at maximum height at zeroPass and the non-stationary duration is assumed to be double the duration between foot-off and zeroPass. When WIP motion is initiated maximum foot height and non-stationary duration is unknown, thus until one of the feet has reached zeroPass direct approach is used for locomotion. If both feet remains stationary for longer than the user configurable timeout, it is assumed that the user intends to stop. This value will determine the frequency of unintended stops and the stopping latency. In order to distinguish between rotation-in-place and WIP, if maxHeight is under certain threshold (of 0.02-0.03m) speed is set to zero. The equation for indirect approach is presented below.

$$v_{locomotion} = scalefactor \times maxHeight \times f(WIPPeriod),$$

where $f(WIPPeriod)$ is a function that increases as $WIPPeriod$ decreases.

2.4.3 Heading

Locomotion heading can be determined either by user's gaze direction obtained from the headset or by the midway direction of the feet. The latter requires syncing (offset compensation) between the virtual environment frame and the ground frame. Gaze-directed heading is easy to implement as it is provided by the Google Cardboard API. Feet-directed heading needs to be implemented in the WIP client. Pseudocode of said algorithm is provided below.

1. Compute relative orientation quaternion to initial orientation quaternion.
2. Decompose said quaternion and obtain the two-dimensional direction vector (representing the yaw angle) of both feet.
3. Add the direction vector of both feet to obtain midway direction vector.

Chapter 3

Results and Discussion

3.1 Offline Gait Analysis

3.1.1 Event Detection

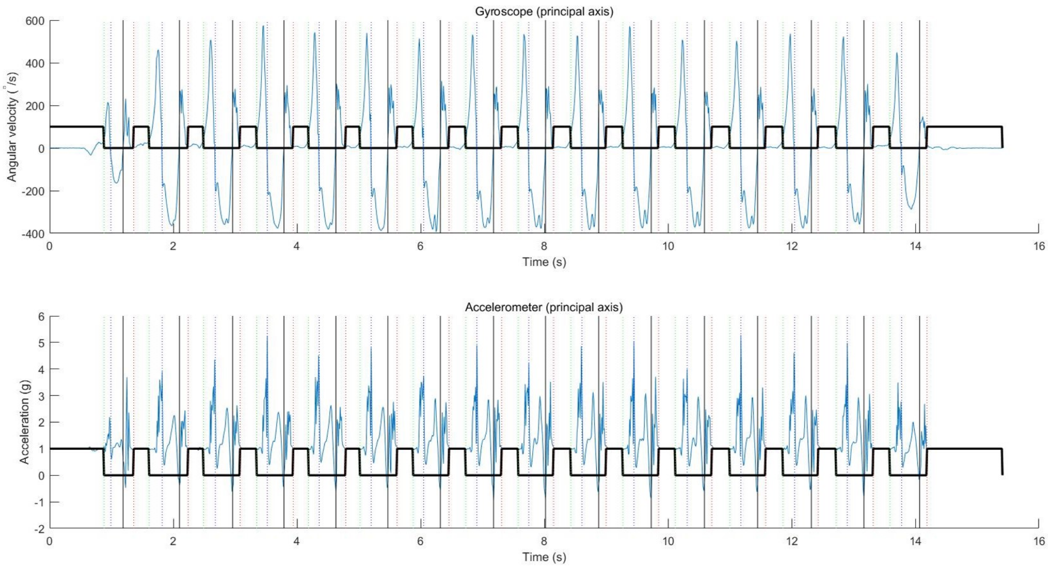

FIGURE 3.1: Gait event detection for a sample walking session. Color-coded vertical lines indicate gait events (black: heel-strike, red: foot-flat, green: heel-off, blue: toe-off). Thick black line represents stationary state with zero being stationary.

FSM described in Chapter 2 was used to generate results shown in Figure 3.1. Gait events were successfully recognized. Gait event detection was only used as a reference and was not required for WIP technique described in this paper.

3.1.2 Trajectory

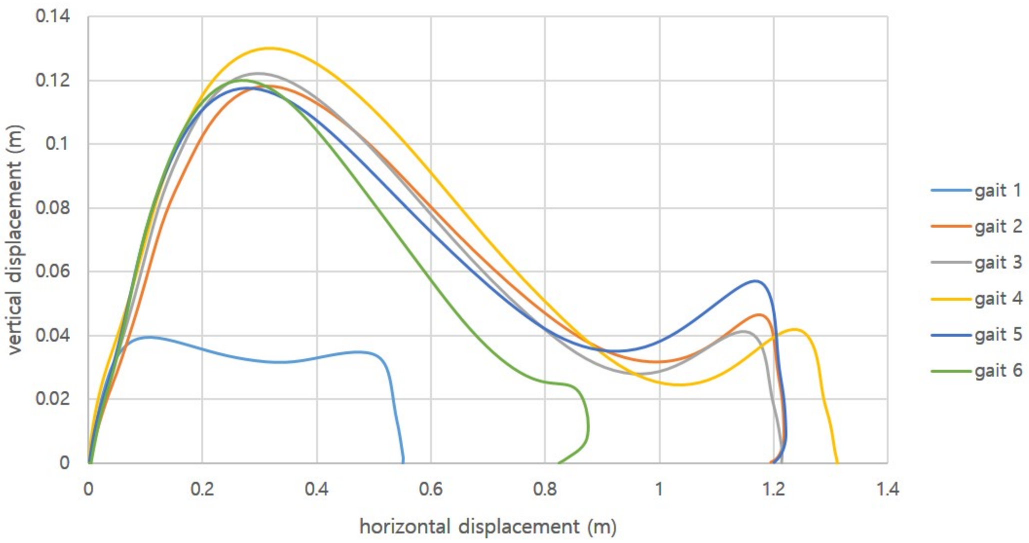

FIGURE 3.2: Gait trajectory for a sample walking session. Gait 1 and 6 are stating and stopping gait sequences, respectively and the rest are steady state gait sequences. Similar results were obtained when compared with Kitagawa and Ogihara, 2016.

Kitagawa's method summarized in Chapter 2 was used obtain gait trajectory shown in Figure 3.2. Time derivative of the horizontal and vertical component of the trajectory yields V_r and V_z. Period T can also be determined from the results. These parameters provide a baseline for the $scale factor$.

3.2 Locomotion Generation

3.2.1 WIP Event Detection

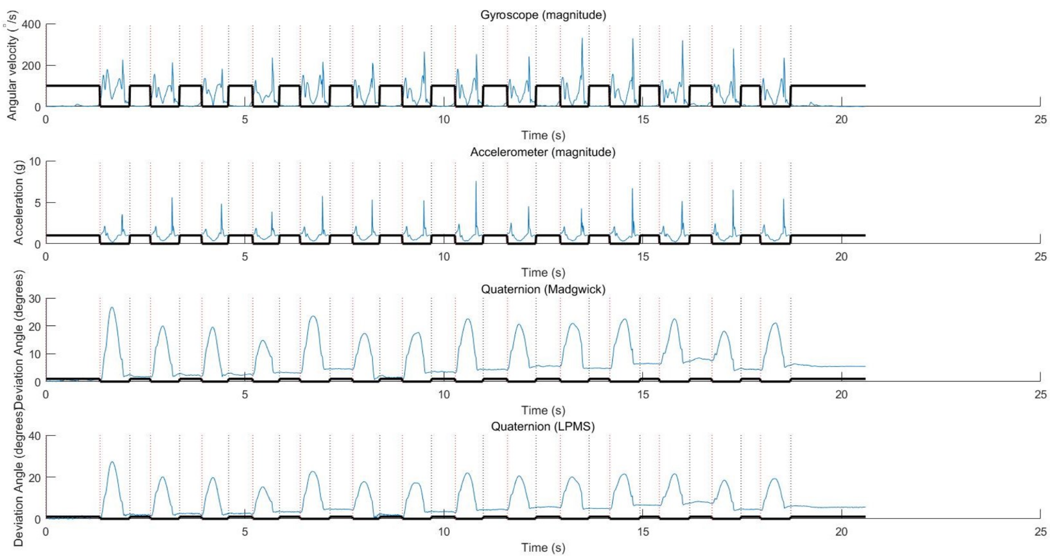

FIGURE 3.3: WIP event detection for a sample WIP session. Color-coded vertical lines indicate state transition (black: foot-strike, red: foot-off). Thick black lines represent stationary state with zero being stationary.

WIP event detection is shown in Figure 3.3. Stationary state is determined by a FSM described in Chapter 2, with simple thresholding for state transition. Deviation angles shown in Figure 3.3 are derived from axis-angle representation of relative quaternions to initial orientation. In order to verify Madgwick's algorithm, it was compared against quaternion readings from the LPMS-B2 IMU sensor.

Figure 3.4 overlays WIP events with xy (pitch and roll) and z (yaw) components of deviation angle. xy component ϕ_{xy} and z component ϕ_z can be obtained from relative quaternion to initial orientation $q = [q_0, q_1, q_2, q_3]$ with the following equation.

$$\phi_z = 2atan2(q_3, q_0) \qquad -\pi \leq \phi_z \leq \pi$$

$$\phi_{xy} = 2acos\left(\frac{q_0}{cos(\phi_z/2)}\right) \quad \text{or} \quad 2acos\left(\frac{q_3}{sin(\phi_z/2)}\right) \qquad -\pi \leq \phi_{xy} \leq \pi$$

As shown in Figure 3.4, xy component of deviation angle (in yellow) starts rising right after foot-off event but settles slightly before foot-strike event, suggesting a slight lag in the foot-strike event detection. This is due to use of filtered values for thresholding in the stationary FSM as mentioned in Chapter 2. By adopting xy component of deviation angle as a means of detecting foot-strike event, lag can be reduced at the expense of added complexity of the FSM. However, for the purpose of implementing the WIP technique, slight lag is encouraged to prevent unintended stops in the generated locomotion. Thus, a simple thresholding of filtered acceleration and angular velocity is used for stationary FSM.

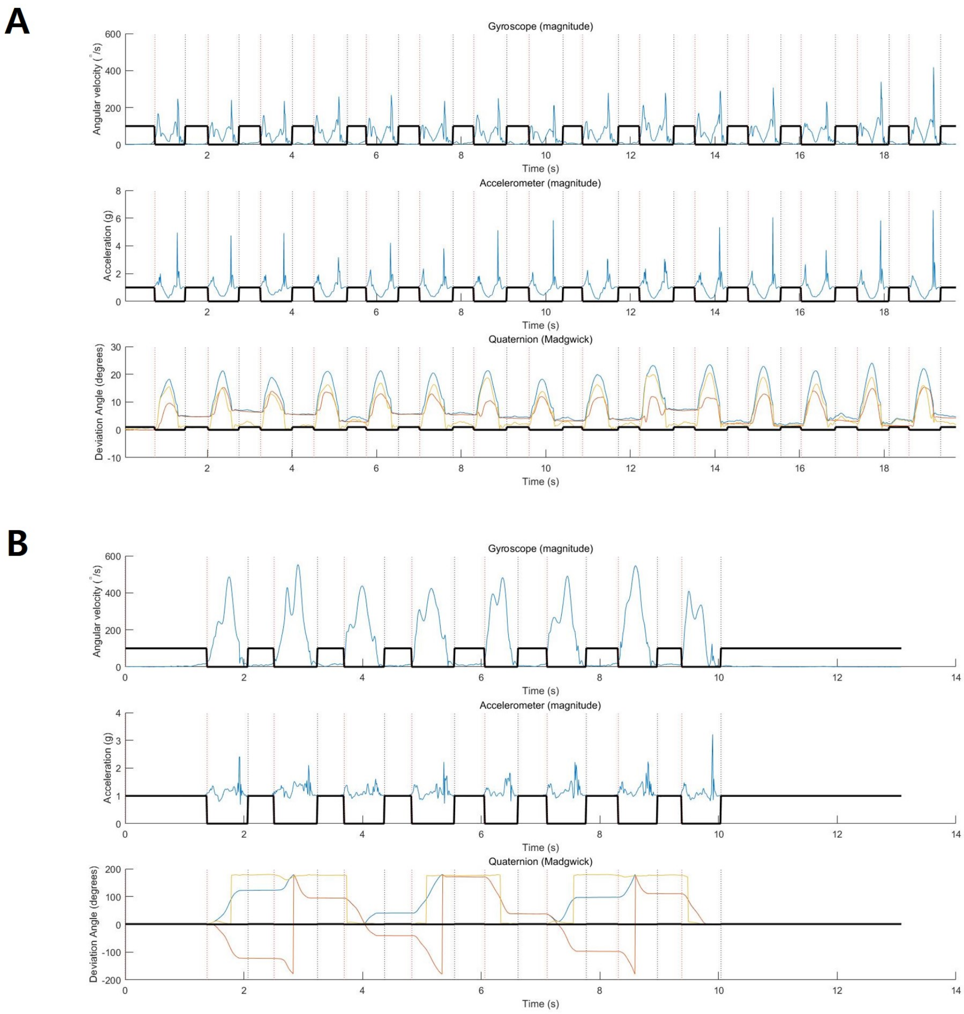

FIGURE 3.4: WIP event detection for A. simple WIP motion, B. WIP motion with rotation. Color-coded vertical lines indicate state transition (black: foot-strike, red: foot-off). Thick black lines represent stationary state with zero being stationary. Angles obtained from axis-angle representation of the relative quaternions to initial orientation can be decomposed into xy and z component as shown in the third row of A and B (blue: combined, red: z component, yellow: xy component).

3.2.2 WIP Sessions

Direct and indirect approach for locomotion generation was conducted on four different WIP sessions - natural WIP, leisurely WIP, marching WIP and natural WIP with rotation. With offline analysis of gait, the $scalefactor$ was appropriately set to generate locomotion that is similar to that of walking. The results are presented in Figure 3.5 through 3.8. From the figures it can be seen that the indirect approach maintains identical starting and stopping latencies with the added benefit of reduced jerkiness and the detection of rotation-in-place as seen in Figure 3.8. Demonstration of proposed WIP technique is shown in Figure 3.9.

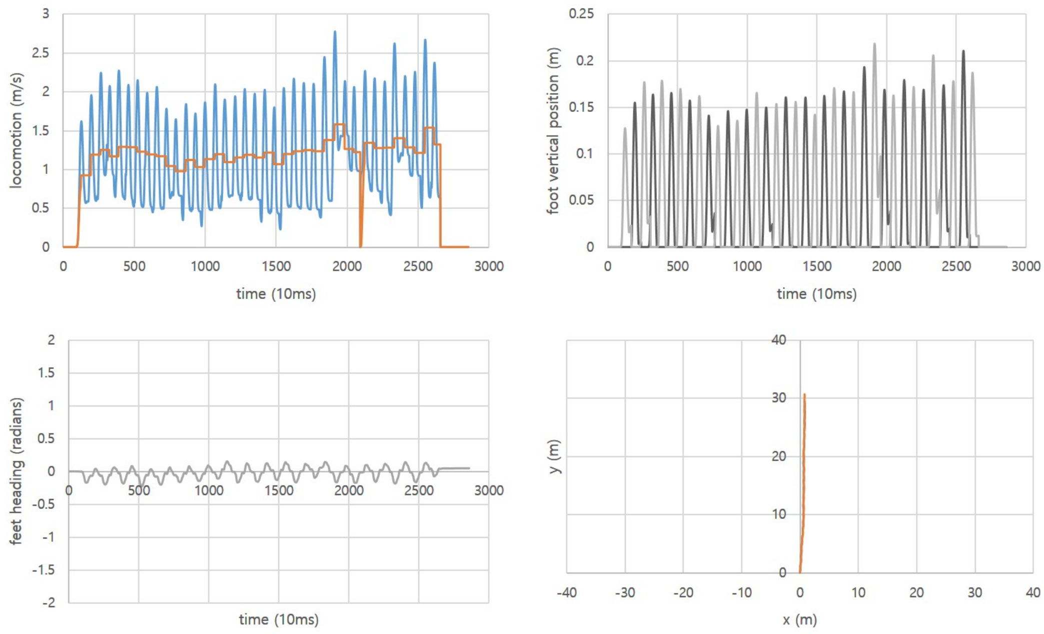

FIGURE 3.5: Locomotion generation for WIP session 1 (natural WIP, medium amplitude). Direct approach is in blue and indirect approach is in orange. Left foot and right foot is in black and grey, respectively.

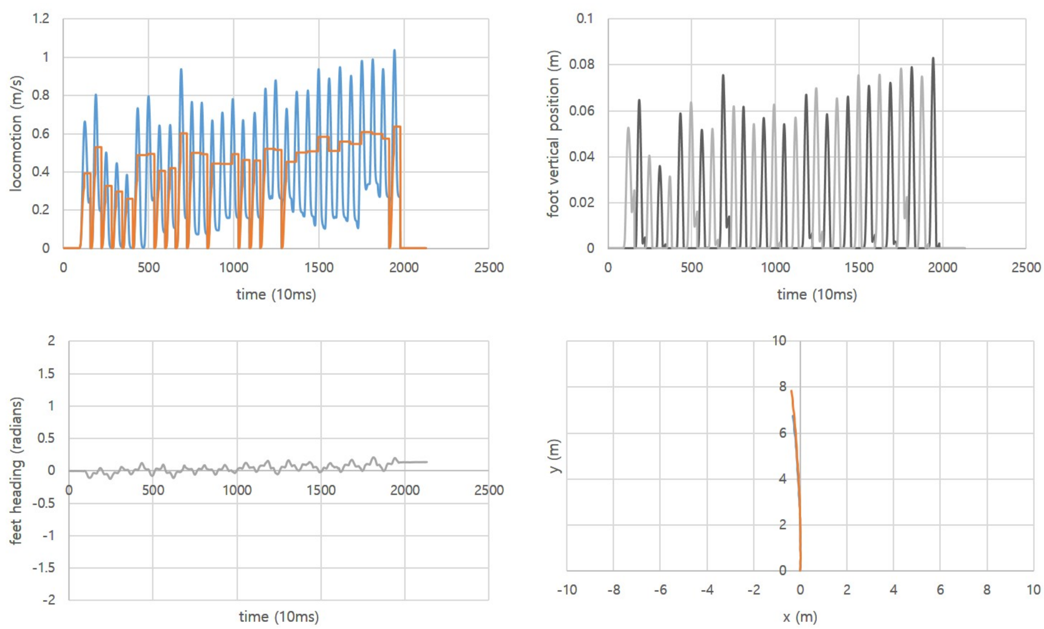

FIGURE 3.6: Locomotion generation for WIP session 2 (leisurely WIP, small amplitude). Direct approach is in blue and indirect approach is in orange. Left foot and right foot is in black and grey, respectively.

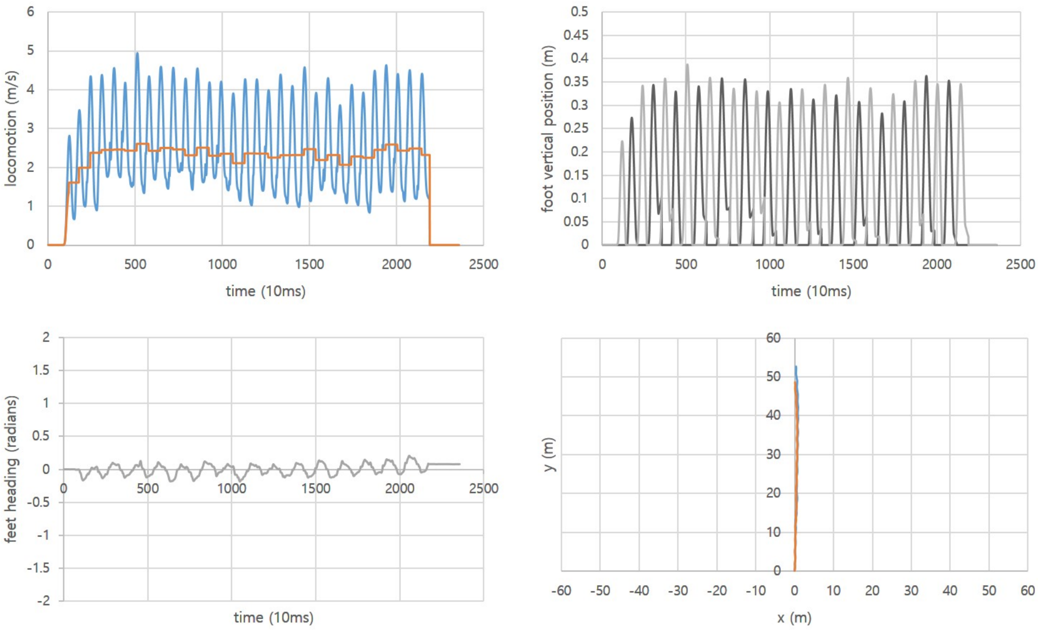

FIGURE 3.7: Locomotion generation for WIP session 3 (marching WIP, large amplitude). Direct approach is in blue and indirect approach is in orange. Left foot and right foot is in black and grey, respectively.

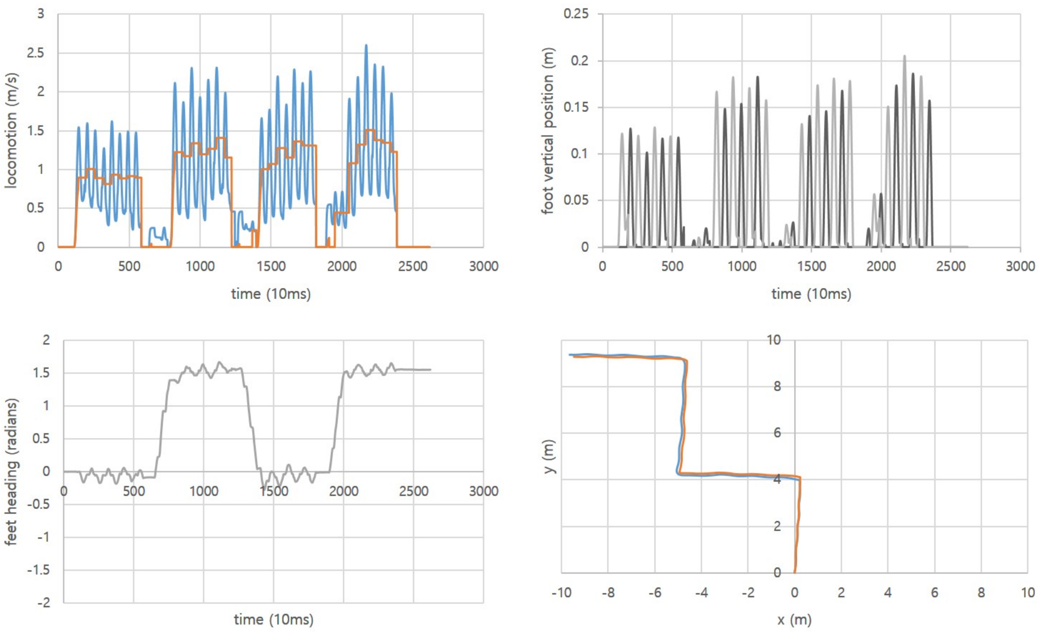

FIGURE 3.8: Locomotion generation for WIP session 4 (natural WIP with rotation). Direct approach is in blue and indirect approach is in orange. Left foot and right foot is in black and grey, respectively.

FIGURE 3.9: Demonstration of proposed WIP technique with Unity (sample scene from Udacity)

Chapter 4

Conclusion

Generating locomotion from WIP motion has been achieved with various setups. LLCM-WIP (Feasel, Whitton, and Wendt, 2008) used magnetic sensors mounted to the knees to track WIP motion and to the chest to determine the heading. GUD-WIP (Wendt, Whitton, and Brooks, 2010) and SAS-WIP (Bruno, Pereira, and Jorge, 2013) both used optical motion capture system with trackers attached to the shins (GUD-WIP) and to the the feet (SAS-WIP), respectively. Bruno et al., 2017 improved upon SAS-WIP and used a commercial hardware (Microsoft Kinect) instead of an optical motion capture system. Recently there has been attempts such as the VR-STEP (Tregillus and Folmer, 2016) to implement the WIP technique with on-board smartphone sensors only. Though it lacks sophistication such as variable locomotion speeds with spatio-temporal WIP parameters achieved in other setups, no custom hardware is required.

The setup proposed in this paper retains the level of sophistication of the most recent research and at the same time reduces the bulk and the cost of additional hardware required. Moreover, unlike some setups in the literature, the inside-out nature of this setup does not require external sensor arrays which allows for a hassle free user experience. For research purposes LPMS-B2 modules costing 300$ each was used. However, consumer-grade MEMS IMU sensor which is inexpensive and small enough to be integrated in the shoe, is adequate for the setup proposed in this paper. Future studies might focus on developing custom hardware and conducting user surveys to further iron out issues and improve performance metrics.

Bibliography

Bruno, Luís C., João Pereira, and Joaquim Jorge (2013). "A New Approach to Walking in Place". In: *Human-Computer Interaction - INTERACT 2013: 14th IFIP TC 13 International Conference Proceedings*. Vol. 8119. 3, pp. 370–387. URL: https://doi.org/10.1007/978-3-642-40477-1_23.

Bruno, Luís C. et al. (2017). "Hip-directed walking-in-place using a single depth camera". In: *Int. J. Hum.-Comput. Stud.* 105, pp. 1–11. URL: https://doi.org/10.1016/j.ijhcs.2017.03.006.

Feasel, Jeff, Mary C. Whitton, and Jeremy D. Wendt (2008). "LLCM-WIP: Low-latency, continuous-motion walking-in-place". In: *Proceedings of the 2008 IEEE Symposium on 3D User Interface*. Vol. 3DUI '08, pp. 97–104. URL: http://ieeexplore.ieee.org/document/4476598/.

Hannink, J. et al. (2017). "Sensor-Based Gait Parameter Extraction With Deep Convolutional Neural Networks". In: *IEEE Journal of Biomedical and Health Informatics* 21.1, pp. 85–93. URL: https://www.ncbi.nlm.nih.gov/pubmed/28103196.

Jasiewicz, Jan M. et al. (2006). "Gait event detection using linear accelerometers or angular velocity transducers in able-bodied and spinal-cord injured individuals". In: *Gait & Posture* 24.4, pp. 502–509. URL: https://www.ncbi.nlm.nih.gov/pubmed/16500102.

Ju, Hojin et al. (2016). "A pedestrian dead-reckoning system that considers the heel-strike and toe-off phases when using a foot-mounted IMU". In: *Measurement Science and Technology* 27.1, p. 015702. URL: http://stacks.iop.org/0957-0233/27/i=1/a=015702.

Kitagawa, Naoki and Naomichi Ogihara (2016). "Estimation of foot trajectory during human walking by a wearable inertial measurement unit mounted to the foot". In: *Gait & Posture* 45, pp. 110–114. URL: https://www.ncbi.nlm.nih.gov/pubmed/26979891.

Madgwick, Sebastian O. H., Andrew J. L. Harrison, and Ravi Vaidyanathan (2011). "Estimation of IMU and MARG orientation using a gradient descent algorithm". In: *2011 IEEE International Conference on Rehabilitation Robotics*. Vol. ICORR '11, pp. 1–7. URL: http://ieeexplore.ieee.org/document/5975346/.

Nilsson, Niels Christian, Stefania Serafin, and Rolf Nordahl (2016). "Walking in Place Through Virtual Worlds". In: *Proceedings of the 18th International Conference on Human-Computer Interaction. Interaction Platforms and Techniques*. Vol. 9732. 2, pp. 37–48. URL: http://dx.doi.org/10.1007/978-3-319-39516-6_4.

Rampp, Alexander et al. (2015). "Inertial sensor-based stride parameter calculation from gait sequences in geriatric patients". In: *IEEE Transactions on Biomedical Engineering* 62.4, pp. 1089–1097. URL: https://www.ncbi.nlm.nih.gov/pubmed/25389237.

Razzaque, Sharif, Zachariah Kohn, and Mary C. Whitton (2001). "Redirected Walking". In: *Eurographics 2001 - Short Presentations*. URL: http://diglib.eg.org/handle/10.2312/egs20011036.

Rueterbories, Jan et al. (2010). "Methods for gait event detection and analysis in ambulatory systems". In: *Medical Engineering & Physics* 32.6, pp. 545–552. URL: https://www.ncbi.nlm.nih.gov/pubmed/20435502.

Slater, Mel, Martin Usoh, and Anthony Steed (1995). "Taking Steps: The Influence of a Walking Technique on Presence in Virtual Reality". In: *ACM Transactions on Computer-Human Interaction* 2.3, pp. 201–219. URL: http://doi.acm.org/10.1145/210079.210084.

Tregillus, Sam and Eelke Folmer (2016). "VR-STEP: Walking-in-Place Using Inertial Sensing for Hands Free Navigation in Mobile VR Environments". In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. Vol. CHI '16, pp. 1250–1255. URL: http://doi.acm.org/10.1145/2858036.2858084.

Wendt, Jeremy D., Mary C. Whitton, and Frederick P. Brooks (2010). "GUD WIP: Gait-Understanding-Driven Walking-In-Place". In: *Proceedings of the 2010 IEEE Virtual Reality Conference*. Vol. VR '10, pp. 51–58. URL: http://dx.doi.org/10.1109/VR.2010.5444812.

Whittle, Michael (2006). *An Introduction to Gait Analysis*. 4th ed. Butterworth-Heinemann. ISBN: 9780750688833.

Woodman, Oliver (2010). "Pedestrian localisation for indoor environments". PhD thesis. University of Cambridge.