Development Of Portable Actuator For

Exoskeleton With Visual Sensing Integration

BY

An-Chi He

B.S., National Chung-Cheng University, Chiayi, Taiwan, 2013

THESIS

Submitted as partial fulfillment of the requirements

for the degree of Master of Science in Mechanical Engineering

in the Graduate College of the

University of Illinois at Chicago, 2020

Chicago, Illinois

Defense Committee:

Myunghee Kim, Chair, Advisor

Miloš Žefran, Electrical and Computer Engineering

Pranav Bhounsule

Acknowledgments

Thank you Dr. Myunghee for guiding me through the phenomenal biomechanical world and provides me the cutting-edge knowledge, allowing me to explore my passion, support me all the way. This thesis has a humble beginning, starting as a course project which opens the road of the incredible wearable robotic world.

Thank you, Dr. Quintin, for supporting me from the first semester when I got to UIC, your touching yet firm words carry me to where I am. I wish I want to provide as much love as you do in my future.

Thank you, Kishan Patel, for cooperating with me on developing the controller, especially thank you for your 3D-sensing algorithm development, this study won't be able to finish without your invaluable contribution and I enjoy every moment of our discussion. Thank you, Cheng-En Wu, for drawing such a fantastic figure for the overall experiment setup.

Thank you to Prakyath, Seong-Mi, and Hyeong-Keun for discussing with me and gives me great ideas. Thank you to Yong-Seok for the hardware design, I believe this device will have outstanding progress in the future. Thank you, Dong-Sheng, for showing me the capability of the visual sensing camera.

Thank you to my mother, grandma, father, aunt, and Chien, I'm only here because your unconditional love is supporting me.

TABLE OF CONTENT

CHAPTER PAGE

1. INTORDUCTION ... 1
   - 1.1. Significance of lower-limb rehabilitation ... 1
   - 1.2. Significance of portable exoskeleton development ... 2
   - 1.3. Purpose of the study ... 2
   - 1.4. Contribution ... 3
   - 1.5. Summary ... 4

2. THE PORTABLE ACTUATOR SYSTEM ... 5
   - 2.1. Introduction ... 5
   - 2.2. Methods ... 7
     - 2.2.1. Communication protocol ... 8
     - 2.2.2. Hardware ... 9
   - 2.3. Results ... 11
     - 2.3.1. Motor response ... 11
     - 2.3.2. EtherCAT performance ... 13
   - 2.4. Discussion ... 14

3. ASSISTIVE WALKING CONTROLLER DESIGN ... 15
   - 3.1. Introduction ... 15
   - 3.2. Methods ... 17
     - 3.2.1. Gait detection ... 17
     - 3.2.2. Assistive torque trajectory ... 20
     - 3.2.3. Optimization ... 23
     - 3.2.4. Safety function ... 24
     - 3.2.5. Experimental Setup ... 24
   - 3.3. Results ... 27
     - 3.3.1. Gait detection ... 28
     - 3.3.2. Torque tracking, torque vs. gait ... 29
     - 3.3.3. Parameters ... 32
     - 3.3.4. Metabolic cost ... 34
   - 3.4. Discussion ... 35

4. SLOPE DETECTION ... 37
   - 4.1. Introduction ... 37
   - 4.2. Method ... 38
     - 4.2.1. Hardware ... 38

4.2.2. Slope detection algorithm 39
4.3. Result 40
4.4. Discussion 43

5. SUMMARY OF CONCLUSION 44

CITED LITERATURE 46

LIST OF TABLES

TABLE I..36

LIST OF FIGURES

2.1 Schematic of the experimental setup 9

2.2 Sensor displacement for gait detection 11

2.3 Motor step response 13

3.1 Illustration of the experimental setup 17

3.2 Gait detection rules based on sensor activation 20

3.3 Gait detection flow 20

3.4 Soleus muscle activation during gait 20

3.5 Gait time optimizer 21

3.6 Torque tracking performance 27

3.7 Gait detection by using two pressure sensors 29

3.8 Torque tracking performance 31

3.9 Actual measured torque vs. measured ankle angle and states 32

3.10 The effect of tuning the parameter compare to the actual measured torque 34

4.1 Slope detection using RGB-D camera with built-in IMU 41

4.2 Plane segmentation 41

4.3 Slope estimation 42

1. INTRODUCTION

1.1 Significance of lower-limb rehabilitation

Walking is an essential function in our daily life, stroke, one of the most common disease causes malfunctioning of walking impacts wide range of people (Dean et al, 2001), according to World Health Organization (Mackey et al, 2008), stroke can lead to walking disabilities such as balance disability and gait disturbance (Tyson et al, 2006; Mackey et al, 2008). Patients have to go through series trails of rehabilitation to regain the ability of walking, a powered wearable robotic rehabilitation has been proved as an effective approach (Shorter et al,2011; Díaz et al, 2011; Sankaranarayan et al, 2016). For the patients, rehabilitation can be a lengthy process it requires constant trails for months (Dean et al, 2001; Jørgensen et al, 1995; Kawahira et al, 2004; Lang et al, 2009). However, the location of the clinical site may restrict the patient who has to travel from distant places, study (Wade, 2003) shows that patients have better rehabilitation performance when in their environment; thus, a portable rehabilitation device is wanted to eliminate the restriction of the clinical site.

1.2 Significance of portable wearable device development

Technology evolvement brings a new relationship between robot and human, exoskeleton such as lower-limb orthosis serves well on applications such as human performance augmentation (Dollar et al, 2008; Zoss et al, 2006; Alan et al,2015; Kim et al, 2019) and rehabilitation (Shorter et al,2011). The benefit of the wearable robot combined the intelligence of wearer and the strength of the robot to assist humans in achieving a new height (Walsh et al, 2007). There are several commercially available products augment human performances such as lifting heavyweight (Walsh et al, 2007) or enhance locomotion ability (Lee et al., 2019). Ankle exoskeleton shows THE great potential of restoring locomotion ability for patients suffers from locomotive disorders (Sankaranarayan et al, 2016; Cain et al, 2007). With a portable wearable device, patients can have a more consistent rehabilitation trial, furthermore, the patient may explore the daily living environment with the assistance of the device, increasing the motivation and effectiveness of rehabilitation (Wade,2003). This can also reduce the labor-intense clinical rehabilitation by not requiring the governing of clinical staff.

1.3 Purpose of the study

This study aims to build an experimental platform for portable rehabilitation devices. We start with the essential sub-systems, which are controller, communication, control algorithm, and gait sensors(Chen et al., 2016). This experimental platform allows us to examine various control strategies and gait detection approaches. Other sub-systems can be added to the existing platform once the essential part of the system is proved working. We built the system with the commercially available motor, low-level controller and end-effector, uses a real-time Ethernet-based communication protocol between the computer and the low-level controller. The control algorithm is running on an isolated physical CPU while Windows OS running on other CPUs. The control algorithm is deployed into TwinCAT (The Windows Control and Automation Technology) program which serves as a target machine.

1.4 Contribution

An-Chi He:

- Designed the portable actuator structure
- Configured software communication structure
- Implemented EtherCAT protocol
- Implemented virtual target machine for Simulink model deployment
- Conducted motor response test
- Programmed assistive walking controller
- Programmed gait detection algorithm
- Programmed optimizers and safety functions in gait assistance algorithm
- Constructed experiment setup
- Conducted portable actuator bench-top test
- Assisted slope detection algorithm
- Analyzed experiment data
- All the unlisted works in this thesis

Kishan Patel:

- Programmed slope detection algorithm
- Conducted slope detection test
- Conducted portable actuator bench-top test
- Assisted gait sensor displacement

1.5 Summary

Overall, this study presented a novel approach of a portable actuator, greatly reduced the size of a portable device from previous studies while preserving the adjustability and expandable feature. Chapter 2 shows the progress and our approach to building communication between controllers. This is the frontier of building a full-scale portable device, the application of this system can also be expanded to other function which related to high-level control, such as robotic, human augmentation and motion control. Furthermore, the controller can be programmed in Simulink instead of low-level languages, which significantly improves the efficiency of programming the controller since Simulink is way easier to work with. To assist, the relationship between assistive motion and human state must be described, chapter 3 demonstrated the possibility of using two simple pressure sensors for efficient gait detection. The relationship of human gait and assistive torque is also described in chapter 3, finish with the metabolic cost measurement to prove the device is working properly. Chapter 4 is exploring the possibility of using an RGB-D camera with IMU sensor for environment detection, once the wearable device is augmented with visual sensing capability, features such as wall and plain detection can prevent the collision or adjust the assistive motion accordingly.

2. THE PORTABLE ACTUATOR SYSTEM

In this chapter, I will present the design of the portable actuator system and demonstrate its performance through a series of benchtop tests. The highlight of this chapter is the development process that I choose for the portable actuator system including communication protocol, sensors, and computation unit. Since it is time-consuming and labor-intense for developing a wearable device; thus, an efficient approach for the portable experimental platform with real-time computation ability is preferred. Physical training for stroke rehabilitation programs can be time-consuming and requires the therapist and the patient to spend a long time plus intensive labor on a clinical site. With a portable device, the patient may have rehabilitation in the household environment increase trials and ease the labor effort of rehabilitation. This study explored the portable actuator system, focusing on system computation, communication, and control algorithm design, presenting essential preliminary works of the system.

2.1 Introduction

Walking is a crucial function in our daily life, diseases such as stroke may lead to high disability according to World Health Organization (Mackey et al, 2008), symptoms such as balance disability (Tyson et al, 2006) and gait disturbance (Rathore et al, 2002) are not rare in stroke patient which effect daily walking. Rehabilitation with the wearable device on an upper limb has shown great potential in chronic stroke patients (N. S. K. Ho et al, 2011; Kerbs et al, 1998), improvement can also be found on lower limb rehabilitation (Shorter et al, 2011; Díaz et al, 2011). The requirement of the wearable device is significantly different from the off-board device. Clinical rehabilitation mostly using the off-board device in which all the equipment is located off-board from the user so weight, size of the motor and computation unit are not a concern; while the on-board device will require the user to carry the equipment. Considering the rehabilitation subject are patients with gait impairment and balance disability, carrying heavy wearable device is not an option due to potential risk, lightweight becomes the top priority for a portable actuator. Other than the weight, a real-time computation capability is preferred in a high-end motion control system such as a wearable device, this also allows us to implement multiple safety mechanisms in the controller to ensure the safety of using the device.

With a wearable rehabilitation device, rehabilitation sites can extend from clinical site to home, this encourages the user to explore the household environment and may motivate the patient by letting them regain the locomotive ability. A safe, lightweight portable rehabilitation device is crucial for such an application. Moreover, the development can be time-consuming for just configuring a suitable onboard computation unit, the computation capability can also prevent implement sophisticate control algorithms. To overcome the above difficulties, a low-cost experimental portable actuator design with benchtop test and the human walking test will be presented.

2.2 Methods

In this section, system construction is divided into two parts which are communication protocol and hardware, this system implemented a Real-Time Ethernet (RTE) communication protocol bridged the high and low-level controller guaranteed real-time communication, once the communication is ready, we move on to the computation part, reserved one physical CPU core make it serve as a target machine, then the high-level controller can be programmed in Simulink then deploy into the target machine to interact the other sub-systems (e.g.: motor unit, sensors). An RTE capable low-level controller is used to resolve the command signals via Ethernet for Control Automation Technology (EtherCAT) protocol while receives sensor signal by built-in digital I/O. Most of the components are commercially available which makes it an efficient approach in the manner of time and cost. The RTE protocol is preferred due to the real-time capability and high bandwidth.

2.2.1 Communication protocol

The communication protocol is the foundation of my approach due to it's the core of real-time communication and the ability to run a Simulink model in real-time. An RTE based protocol called EtherCAT (Ethernet for Control Automation Technology) is implemented to communicate between the control system, and the EPOS4 controller, it uses master and slave structure to denote the control system and low-level controller. RTE based protocols consist of advantages of low latency and high bandwidth, latency is under $3\ \mu s$ on 100 Mb/s bandwidth and under $1\ \mu s$ latency on 1 Gb/s bandwidth, the bandwidth is significantly better when comparing to CAN which has 1 MB /s bandwidth. Although the performance of EtherCAT controlling multiple devices is not the best among all RTE based protocols, the difference is neglectable when controlling under 20 devices based on study (Robert et al, 2012). Besides, the ability of capable to integrate Simulink under EtherCAT structure makes it stand out among all RTE protocols. EtherCAT uses Ethernet Standard IEEE 802.3 frame, under the EtherCAT structure as Fig 2.1 shows, while communicating between the control system and low-level controller, it's able to extract from and insert data into the frames. EtherCAT uses a distributed clock (DC) as a synchronization mechanism (Langloisa et al, 2018) for syncing all nodes in the network based on the DC of the control system (Cena et al, 2012). Beckhoff [Verl, Germany] provides TwinCAT (The Windows Control and Automation Technology) as a real-time operating system in the master device, TwinCAT reserve physical cores on the Windows computer for real-time computation and EtherCAT communication, while Windows OS running on other cores, enhance the performance of a personal computer. Under the EtherCAT structure, the controller can be written in C++, PLC and even Simulink, user can configure the personal computer into a real-time target machine for Simulink by using the TwinCAT program and TE1400 (Beckhoff, [Verl, Germany]) add-on.

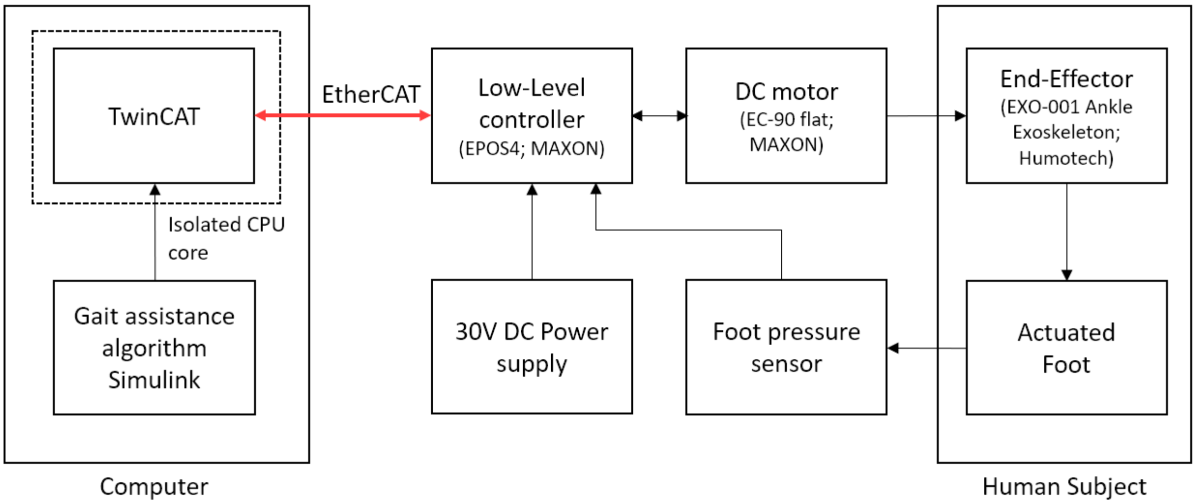

Figure 2.1: Schematic of the experimental setup. The controller is running on an isolated CPU core as a target machine, the control algorithm is deployed into the isolated CPU reads sensor signals from the low-level controller and compute command accordingly based on a pre-describe relationship of human gait and assistive torque.

2.2.2 Hardware

This system (Fig 2.1) demonstrated how it can serve as an ideal experimental platform for gait assisting algorithms by having real-time, virtual target machine and the I/O capabilities. The system consists an EC-90 motor (Maxon motor ag, [Switzerland]), an EPOS4 controller (Maxon motor ag, [Switzerland]), a desktop PC with Intel i7-7800X 3.50GHz processor and two pressure sensors (Sealed touchpad switches-Round; McMaster-Carr, [CA, USA]), although the system can be running on a laptop, here I use a desktop PC for equipment convenient.

The 48V EC-90 motor is attached to a benchtop test housing provides up to 3N-m on motor shaft although the nominal torque suggested by Maxon is 1N-m overclocking within a reasonable range hasn't shown any negative effect on the device. For the low-level controller, this system uses an EtherCAT capable low-level controller EPOS4 50/15 EtherCAT compact, EPOS4 resolve control frame from the master device via ethernet cable, it also contains a series of digital and analog I/O which is used read the sensor signals for gait detection. Two pressure sensors are placed under the left shoe at the heel and the front part of the foot as figure 2.2 shows covered with a 3-D printed sheet to prevent wear out from having direct contact with the ground, these sensors tell the control algorithm whether heel or the front part of the foot is contacting with the ground, five gait events, heel-strike, flat-foot, heel-off, toe-off and swing phases can be interpreted based on two pressure sensors. The computer runs the TwinCAT and Simulink gait assistance model, it receives sensor signals from EPOS4 calculate the assistive torque command accordingly, a detailed description will be presented in Chapter 3. The motor drives a commercially available ankle-foot orthosis (AFO) (EXO-001 Ankle Exoskeleton; Humotech, Inc [Pittsburgh, PA, USA]) by using the Bowden cable to achieve walking assistance. The EPOS4 uses torque control by using rotation speed, torque constant and current:

$$\tau = i * k_m \tag{2.1}$$

where τ is the estimated torque, i is input current, k_m is the torque constant respectively, MAXON provides EPOS Studio for autotune the torque constant, with this approach the system estimates the torque without using a torque sensor to close the loop, however, this approach does contain drawback such as inaccurate estimation on the early stage of torque development, although the torque reaches target torque eventually in a reasonable time.

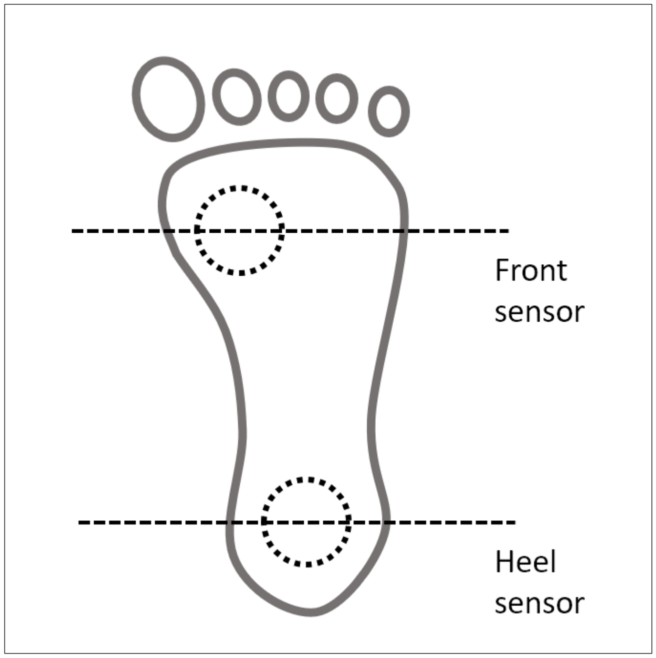

Figure 2.2: Sensor displacement for gait detection. The front sensor is aligned with the sesamoid rear sensor is aligned with heel respectively to detect gait event (e.g.: heel-off, flat-foot, heel-strike).

2.3 <u>Result</u>

This section is to understand the differences of estimated torque vs. actual measured torque, the experiment result is a good insight for compensating the difference between estimated torque and actual torque.

2.3.1 <u>Motor response</u>

We tested motor step response for the motor as fig 2.3 shows, the estimated torque is calculated by the low-level controller using equation 2.1, while the measured torque is measured on the ankle-foot orthosis. The measured torque is not able to reach the command torque due to frictions in the Bowden cable and the gap between torque estimation and actual torque. The response delay of torque command and estimated torque is 6ms, the delay between torque command and the first peak of measured torque is ~100ms, this high delay is due to it requires time to deliver the power through Bowden cable. There are some key points in this test and will be compensated to achieve desired assistive torque, the first key point is the motor is not able to hold the torque in the late stage, instead, using a gradually increasing desired torque trajectory will be better to hold the torque. Second, the jigger in the early stage, this is caused by the backlash while tensioning the cable from slack, we can add pretension to solve this issue in a later chapter. Third, the delay of the measured torque, by knowing it may take nearly 100ms for the torque to deliver, we have to initiate the desired assistive torque earlier than expected for compensating this difference.

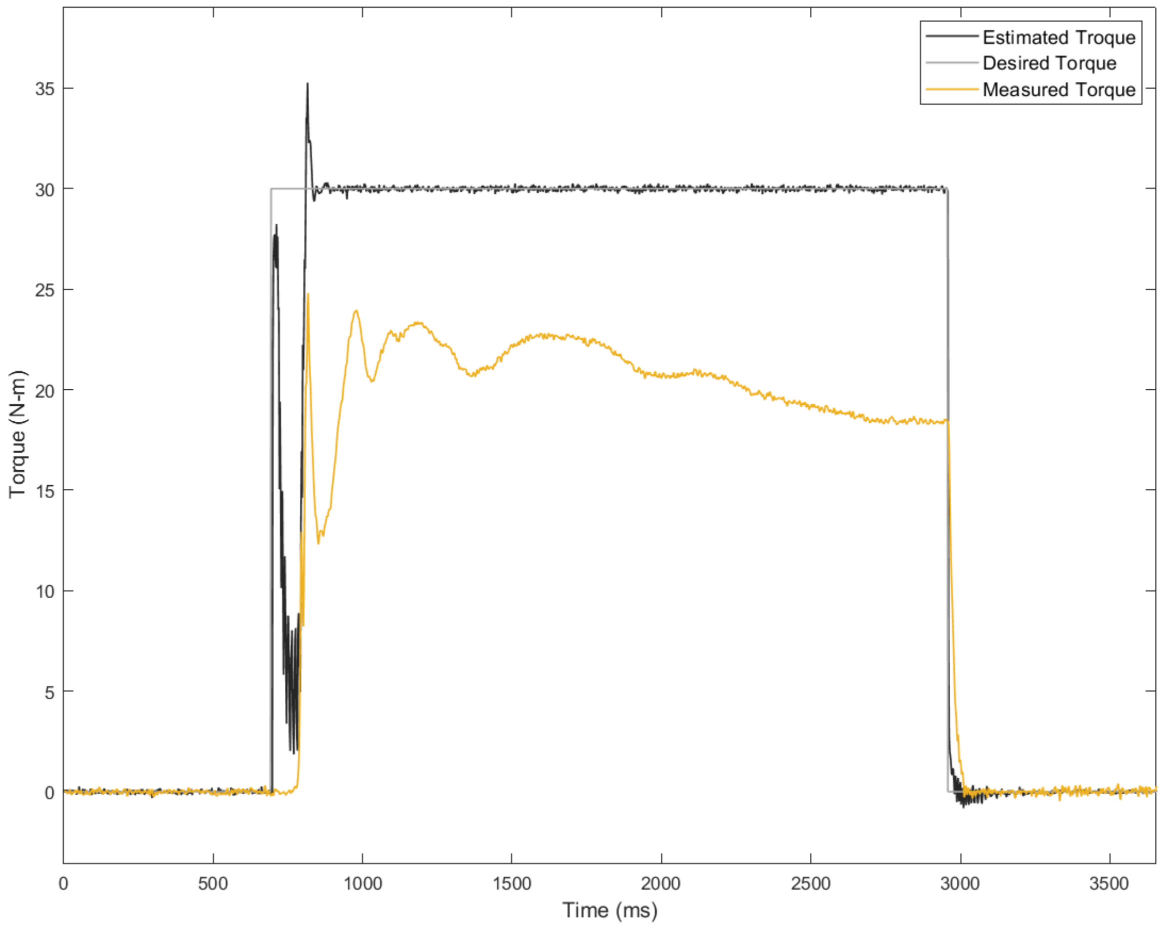

Figure 2.3: Motor step response. This figure comparing between estimated torque and actual measured torque due to the actual torque is not used as control feedback. Characteristics such as delay will be used for control compensation in chapter 3.

2.3.2 EtherCAT performance

Few key features come with EtherCAT communication protocol, 1. Created a target machine by isolated a physical CPU, 2. Simulink controller model can be deployed into the isolated CPU, 3. The Simulink is running with a 500Hz update rate. 4. 100 times communication bandwidth comparing the CAN. Proved EtherCAT can be an impressive alternative for wearable robotic communication.

2.4 Discussion

The system shows the capability of running the Simulink control model with 500 Hz. The low-level controller uses torque estimation for build-in torque control. In this section, the performance of using torque estimation has been examined, differences between measured torque and estimated torque will be compensated in later controller design, to achieve the actual desired assistive torque. Further studies are still required to close the gap between estimated torque and real measurement.

3. ASSISTIVE WALKING CONTROLLER DESIGN

In this chapter, we will investigate how the gait assistive controller algorithm was designed, the controller is a time-based controller to determine desired assistive torque according to the user's gait. Since the assistive motion of the ankle-foot orthosis is functionally the same as soleus muscle; thus, the relationship between time and desired torque is an interpretation of soleus muscle activation during the gait. While the user's gait is detected by two foot-pressure sensors, optimized by the previous two steps.

3.1 Introduction

The ankle-foot orthosis EXO-001 is a cable-driven device assist push-off. The control algorithm observes the human state and assists according to the desired relationship between gait and assistive torque. There are two types of controller commonly used for the ankle-foot orthosis, which are the time-based (Asbeck et al, 2015) and angle-based controller (Humotech, Inc [Pittsburgh, PA, USA]), time-based controller interprets human gait based on time and gait event (e.g.: heel-strike, flat-foot, heel-off, toe-off, swing) while angle based controller is based on ankle angle. A time-based controller is presented in this study, the relationship of time and assistive torque is described by soleus muscle activation during gait and gait time since soleus muscle is the primary muscle of the assistive motion. This time-based controller interpreted soleus muscle activation during gait (Duysens et al, 1991; Suzuki et al, 2014; Pasinetti et al, 2013), considering a full gait cycle count from heel-strike to next heel-strike on the same foot, the highest activation occurred during 50% gait cycle, also called the push-off state, the activation increasing drastically after the mid-stance (Fig 3.4). Based on the above principles, the controller is designed to provide assistive torque started at mid-stance, ended after push-off, however, each individual may have different gait patterns yields different timing of mid-stance and push-off, thus, the controller has to be able to adapt different individuals. To overcome the difficulty of controller adaptation for each individual, two optimization mechanisms are implemented in the controller, firstly, a moving average optimizer calculates the average gait event timing based on previous two steps, secondary, two tunable parameters "start percentage" and "peak percentage" is introduced for tuning the controller behavior. Finally, to explore the actual assistance that delivered to the human subject, the metabolic cost is used as the criteria for examining the result, the aim is to maximize the metabolic cost reduction and minimize the total work to ensure most of the work is helping. The overall system schematic is illustrated in fig. 3.1.

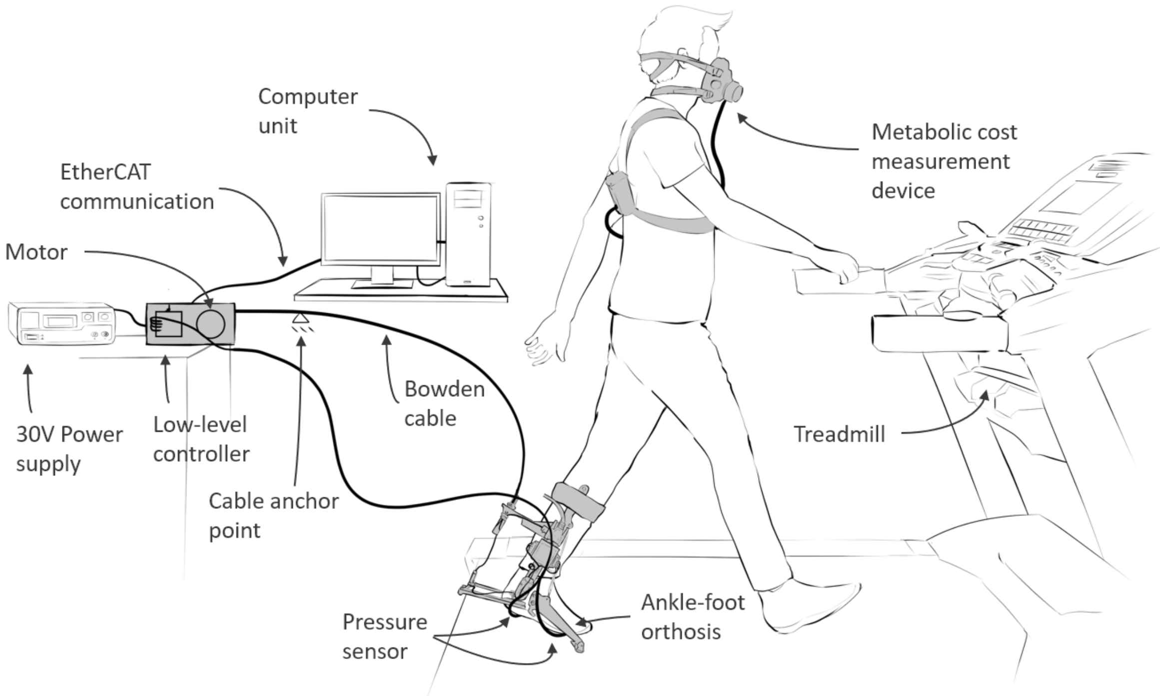

Figure 3.1: Illustration of the experimental setup. The ankle-foot orthosis is actuated by an off-board motor and controller, Bowden cable is used for power transmit. The computer runs the gait assistance controller and communicates with the low-level controller through EtherCAT communication, while gait state is detected by the pressure sensors placed under the shoe.

3.2 Method

An experimental time-based controller is used in this study, it controls push-off work by setting the preferred relationship between gait time and assistive torque, while the gait time is the accumulated time.

The relationship can also be described by:

$$\tau = f(Gait\ percentage) \tag{3.1}$$

while τ is the assistive torque and $gait\ percentage$ is describing the progression of each gait, calculated by:

$$Gait\ percentage = \frac{Gait\ time}{Expected\ step\ time} \tag{3.2}$$

While gait time is the accumulated time from the heel-strike to the swing phase, expected step time is an auto-tuned parameter in the controller.

3.2.1 Gait detection

Five gait events are detected by using two pressure sensors placed under the shoe, which are heel-strike, flat-foot, heel-off, toe-off and swing, the controller retrieves sensor signals and interprets the gait events shown as Figure 3.2. This approach is easy to implement but come with two challenges, first is sensor position, second is false sensor detection. Since this approach is relying on the two pressure sensors; thus, the displacement of sensors has a crucial influence on gait detection, misplacement of sensors could lead to inaccurate interpretation of each gait events. Sensors are deliberately placed under the heel and the position of sesamoid bones under the actuated shoes. The sensor may also have incorrect detection during the gait such as not detecting ground contact even the foot is on the ground, to overcome this, a rolling mechanism of gait events is introduced to the controller. The gait states are divided into heel-strike, flat-foot, heel-off and swing, each state may have three behaviors according to gait detection, the three behaviors are 1. proceed to the next state if conditions are satisfied, 2. stay in the same state 3. move to swing state. Firstly, states may only move in order (e.g.: swing → heel-strike → flat-foot → heel-off) with certain sensor conditions (Fig. 3.2). Second, stay in the same state if sensor signal is incorrect to move on to next state, this prevents false state transition due to incorrect sensor feedback, for example, if the front sensor is suddenly turned off during flat-foot, instead of falling back to the heel-strike state, the controller will stay in the flat-foot state. Third, any state will move to swing state if both sensor signals are off, swing state serves as a reset state if the controller stuck in any other state due to inaccurate gait detection waiting for heel-strike to initiate next gait cycle, another benefit is once the subject lifts their foot then the controller does zero torque control to ensure safety. The gait detection flow is presented as figure 3.3 shown.

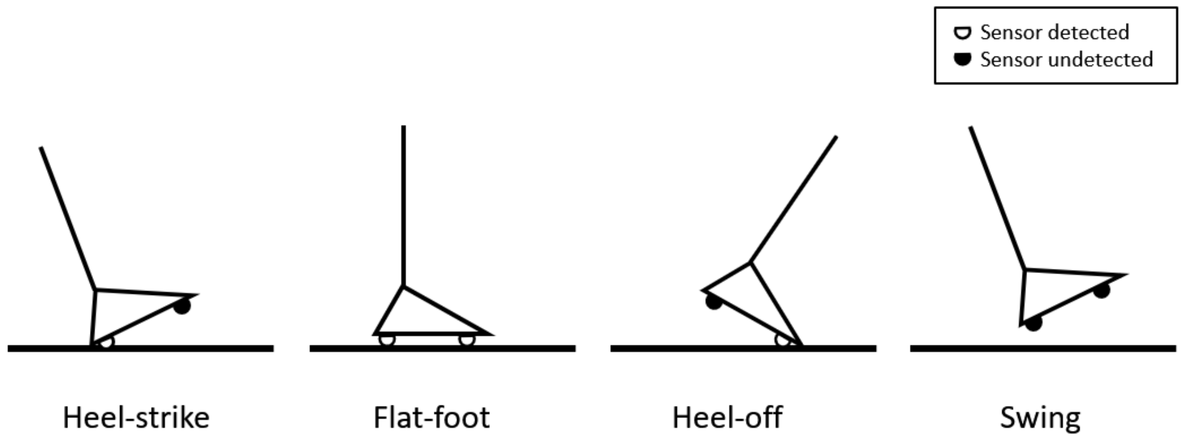

Figure 3.2: Gait detection rules based on sensor activation. This approach is cost-efficient and reliable for the experimental setup to prove the gait detection works.

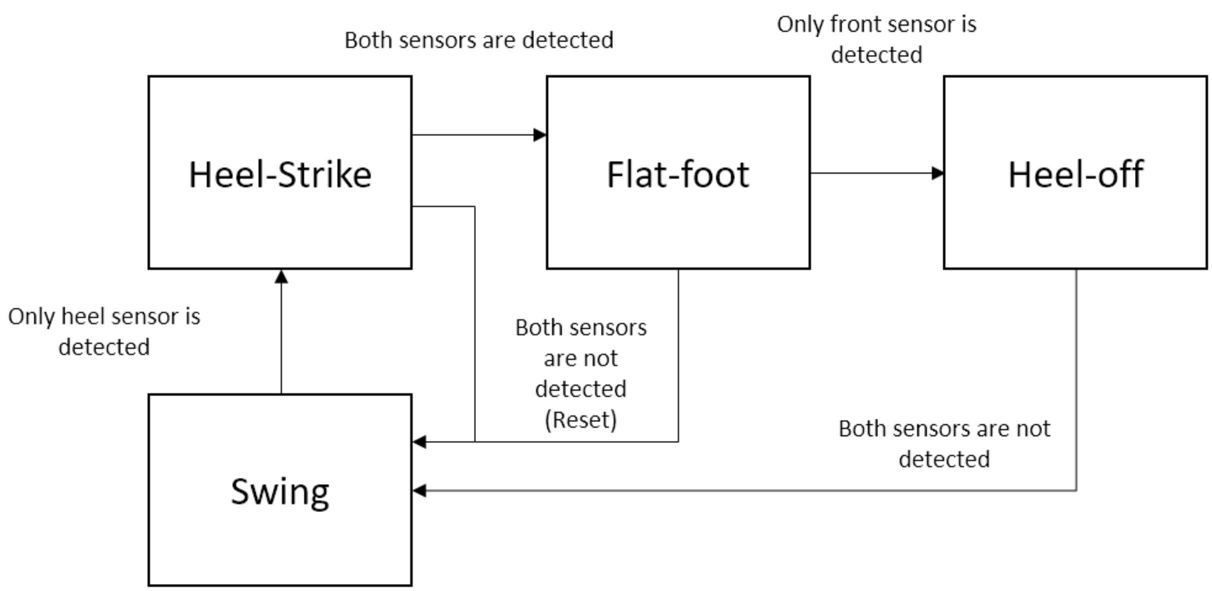

Figure 3.3: Gait detection flow: each state may have three behaviors based on different sensor condition 1. procced to the next state if the condition is satisfied, 2. Stay in the same state if the condition is not satisfied to move on, 3. reset to swing state if both sensors are undetected. This mechanism can well prevent false state transition such as Flat-foot to Heel-strike.

3.2.2 Assistive torque trajectory design

The assistive torque trajectory is designed to drive the primary assistive motion that the end-effector is designed for, in this study, the AFO is designated to assist plantar flexion. The cable used to drive AFO is functionally similar to soleus muscle due to its acts in parallel with the uni-articular soleus muscle. Based on Duysens's study of muscle activation during walking or running (Duysens et al, 1991), the soleus muscle shows higher activation than gastrocnemius during walking, while gastrocnemius shows higher activation during running, this study matches the anatomy perspective that soleus is the primary muscle for plantar flexion.

Pasinetti's study (Pasinetti et al, 2013) shows a novel gait cycle detection uses soleus EMG signal, this study provided good information of soleus muscle activation during the gait, it also indicated that soleus muscle activation is highly repetitive, this may suggest that the soleus muscle is mono-functional for walking.

Galle's study (Galle et al, 2015) also shows the EMG signal reduction of the soleus muscle is higher than gastrocnemius muscle while the push-off timing for their robotic exoskeleton is optimized. this is also a good indication that soleus is the primary muscle for plantar flexion. These studies could be a good indication that soleus muscle activation during gait is a decent approach for designing an assistive trajectory for plantar flexion. The controller is targeting plantar flexion, especially push-off phase, based on Pasinetti's study (Pasinetti et al, 2013) and Figure 3.4 we can see soleus activation increase drastically after mid-stance, reaches maximum activation during push-off and decrease after push-off, the assistive torque is following the same pattern that starts assistance at the mid-stance, reaches peak torque while heel-off, since the push-off will happen in a short time.

The target is assisting the push-off phase; however, each individual may have different gait patterns, so tuning for push-off timing is still required. To overcome this, two control parameters**, start percentage** and **peak percentage** are introduced (Fig. 3.4) and explored in this study, these parameters decide the timing of initiating the assistive torque and reaching to the peak torque respectively, it can be tuned according to different subjects. The assistive torque terminated when a swing state or false state is detected, the false state is triggered when torque dropping drastically indicated that the user may lift the foot while the swing state isn’t detected properly.

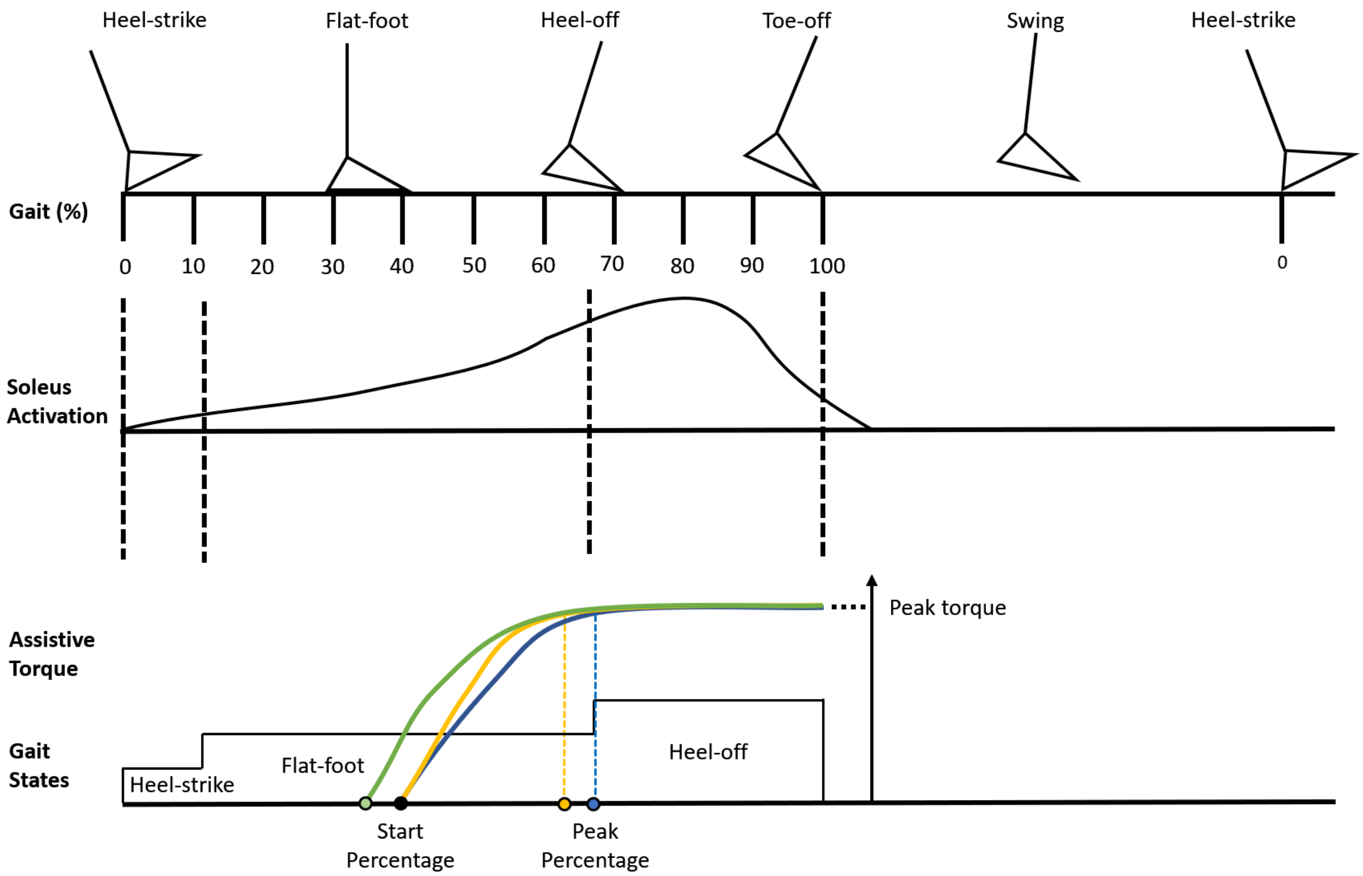

Figure 3.4: Soleus muscle activation during gait. Comparing to desired assistive torque command. Soleus muscle activated during the whole stance phase drastically increases after mid-stance and reaches the peak torque during the push-off phase. The assistive torque is following muscle activation trend which initiates during mid-stance and reaches peak torque at heel-off, it is terminated once toe-off is detected due to the following dorsi motion is conflicting the assistive motion. The Peak torque is around 25 N-m measured on the ankle-foot orthosis, which is lower than the actual push-off torque that the human leg is providing, thus the assistive torque remains maximum during the whole push-off phase.

3.2.3 <u>Moving average optimizer</u>

The moving average optimizer utilized the detected gait time then calculate the average gait time based on the previous two steps of the actuated leg. The averaged gait time is calculated by:

$$Gait\ time_n = \frac{(Gait\ time_{n-1} + Gait\ time_{n-2})}{2} \tag{3.3}$$

while $Gait\ time_n$ denote the expected step time of the current step from heel-strike to toe-off, $Gait\ time_{n-1}$ and $Gait\ time_{n-2}$ denote the recorded time from the previous two steps. The calculated gait time is used to determine gait percentage as Figure 3.5 shows. This optimizer brings two benefits, firstly, the controller can interpret the gait time for a different subject, second, the controller can adjust the gat time if the subject changes walking speed.

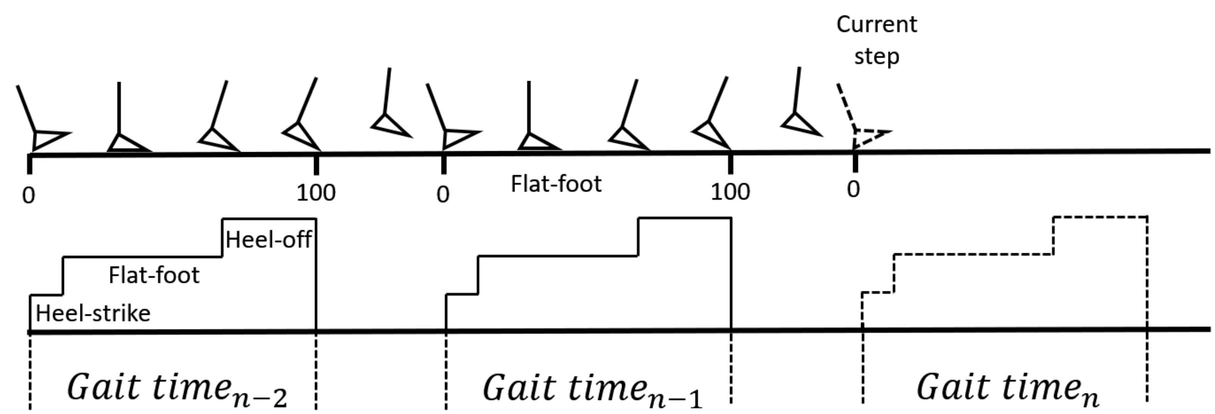

Figure 3.5: Gait time optimizer. Predicting gait time by using an average of the previous two steps.

3.2.4 Safety function

Any misfiring during the gait could lead to severe injuries such as falling; thus, safety mechanisms are introduced to the controller to prevent misfiring or any unwanted behavior. Firstly, if the estimated torque drops drastically then fall into an error state, terminate any torque command and wait for the next heel-strike to initiate the assistance. Second is the prevention of unwanted state transition (e.g.: toe-off → flat-foot), this false transition could happen when the sensor failed to detect correct gait state, instead of transit to incorrect state, the state will remain in the same state and wait until next swing to heel-strike transition is detected to initiate the next assistance.

3.2.5 Experimental setup

To understand how the assistive trajectory helping the user, the metabolic cost is one of the most common ways to do so, by comparing the metabolic cost of unpowered ankle-foot orthosis walking across different parameter sets. The system provided torque ranged from 0 to 25 N-m measured on the end-effector, the system consists of a computer, actuator, metabolic cost measurement device, and an exoskeleton. The metabolic data is measured by COSMED K5 (COSMED, [Rome, Italy]) and calculated by using the following equation:

$$metabolic\ cost\ (W) = 16.52 * \frac{VO_2}{60} + 4.56 * \frac{VCO_2}{60} \tag{3.4}$$

while VO_2 and VCO_2 denotes the flow rate of oxygen and carbon dioxide in each breath respectively. This gives us a great measurement of how the assistive device helping the subject.

Computer control and mechanical power of the ankle-foot orthosis were generated off-board and provided via a Bowden cable (Fig. 3.1). This is the preliminary test setup before the system may become on-board with future development. This experimental setup aims to observe the relationship between the tuning parameter and the measured torque, also measure the metabolic cost of different parameter settings. Four sets of control parameters were tested in this study as Table I and Figure 3.6 shows, while start percentage is a tunable parameter for initiate the assistive trajectory while peak percentage denotes the timing for assistive reaches to the peak torque, the peak torque is a constant torque 3 N-m on the motor shaft across the whole experiment.

Start Percentage	Peak Percentage
60	90
60	100
50	100

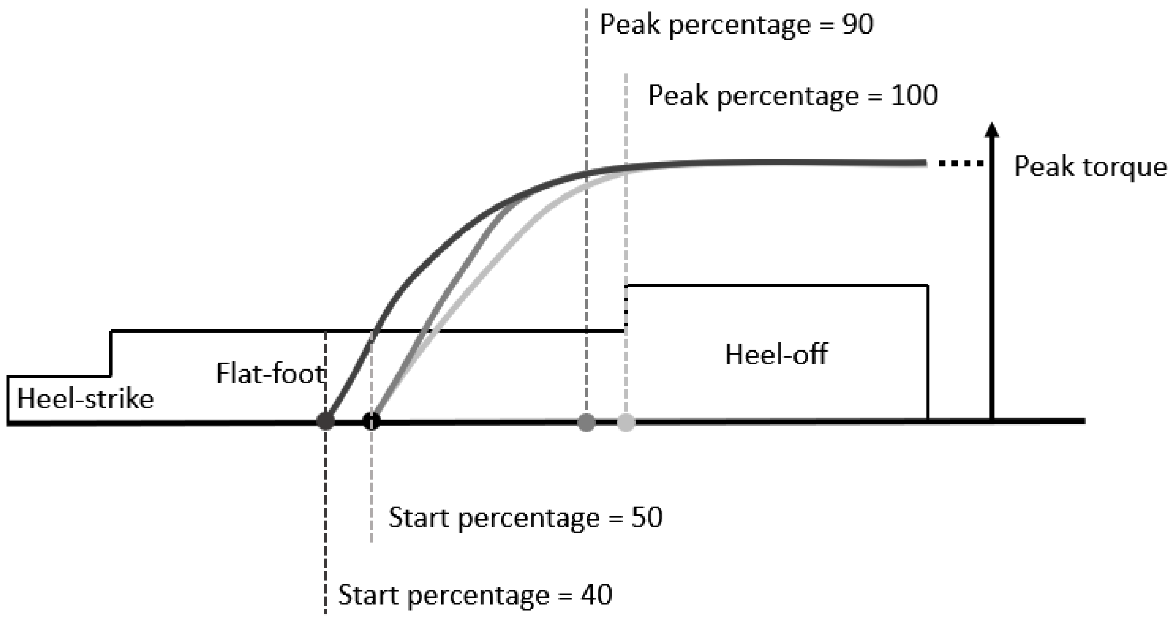

Figure 3.6: The percentage control law used for walking assistance. Peak desired torque is fixed at 3 N-m on the motor side. Start percentage denotes the start timing during the flat-foot phase, for example, the assistive torque will start in the 50% of the flat-foot phase if the Start percentage is set to 50. While the peak percentage determines when the torque will reach the peak, the assistive torque will reach the peak at 90% of the flat-foot phase if the peak percentage parameter is set to 90.

3.3 Result

The benchtop result without the motor on board the human walking test shows 10.82% of metabolic cost reduction achieved by using an EC motor (Maxon motor ag, [Switzerland]), providing 25N-m measured torque on a commercially available AFO. To understand the difference between actual measured torque on the AFO and estimated torque, we conducted several experiments with four parameters set as TABLE I shows. While changing the parameters, measured torque on the ankle-foot orthosis also changes accordingly, this is important since the system doesn't have the direct measurement, instead, the torque is estimated by motor current consumption and torque constant. The result shows a 10.82% metabolic cost reduction between powered and unpowered condition.

3.3.1 Gait detection

Our approach of using two pressure sensors to detect the switch shows the good result as Figure 3.7 shows, the maximum error shows in actual gait and detected state is 25ms, the actual gait is interpreted from ankle angle. The data is a randomly selected gait during the experiment to demonstrate the state detection, although the toe-off state is not precisely detection which contains about 40ms delay, this could be caused by the sensor displacement. Overall, the detection is reliable for gait assistance, implement force-sensitive resistor (FSR) may improve the performance.

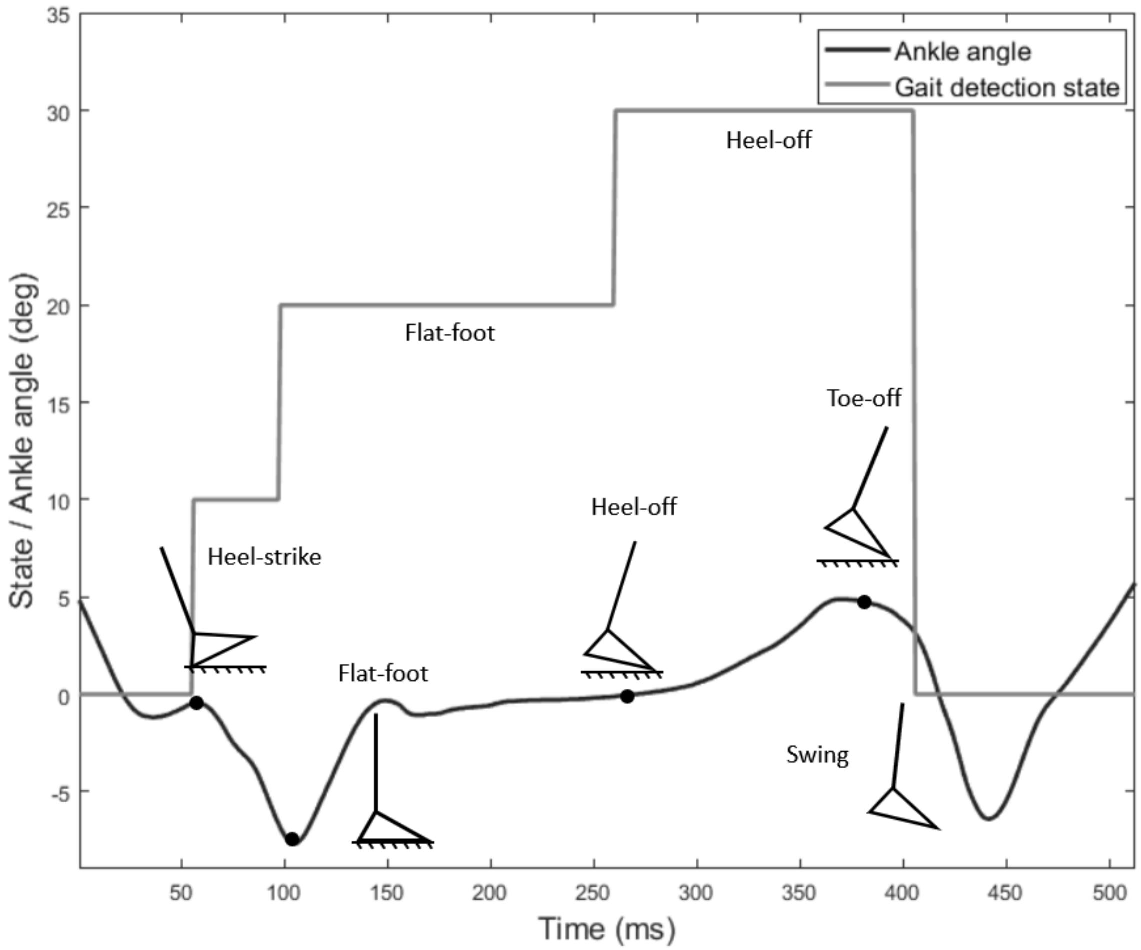

Figure 3.7: Gait detection by using two pressure sensors. The ankle angle shows the actual gait state while the gait detection states represent for four states, heel-strike, flat-foot, heel-off, swing is detected accordingly.

3.3.2 Torque tracking, torque vs. gait

The torque tracking section compares the estimated torque vs. actual measured torque to understand the performance of torque estimation. Figure 3.8 shows the torque tacking performance during gait. The torque command 1) started with 300 mN-m pre-tension once flat-foot state detected, 2) follow up with first sin wave torque-assist until heel-off is detected, 3) after heel-off activate another sin wave to hold the torque. However, the measured torque from the ankle-foot orthosis didn't show a good match due to the estimated torque is calculated by torque constant and current, while a part of torque is used to drive the motor to the desired velocity. At the end of the actual torque, it also shows the motor is not able to keep up with the desired torque. Figure 3.9 shows the measured torque and ankle angle to observe the relationship between torque and gait state, the peak force (~26.5 N-m) is measured on the ankle-foot orthosis and happened right before toe-off to assist the push-off motion correctly.

To overcome the poor torque tracking between estimated torque and actual torque, a compensation function must be added to close the gap.

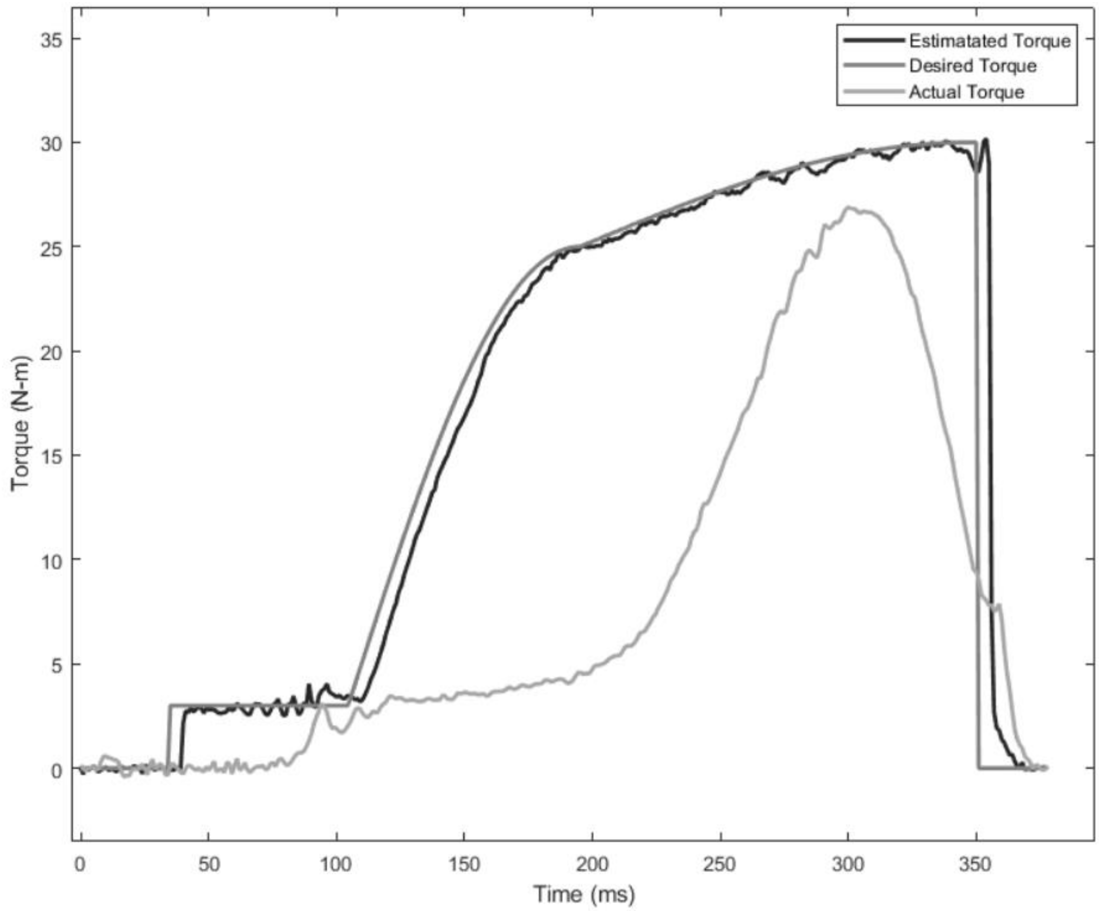

Figure 3.8: Torque tracking performance. Comparing between 1. Estimated torque, 2. Desired torque, 3. Actual torque. Although the tracking performance is not optimal, however, the command is tuned so that the measured torque is compensated for our desired assistive torque trajectory.

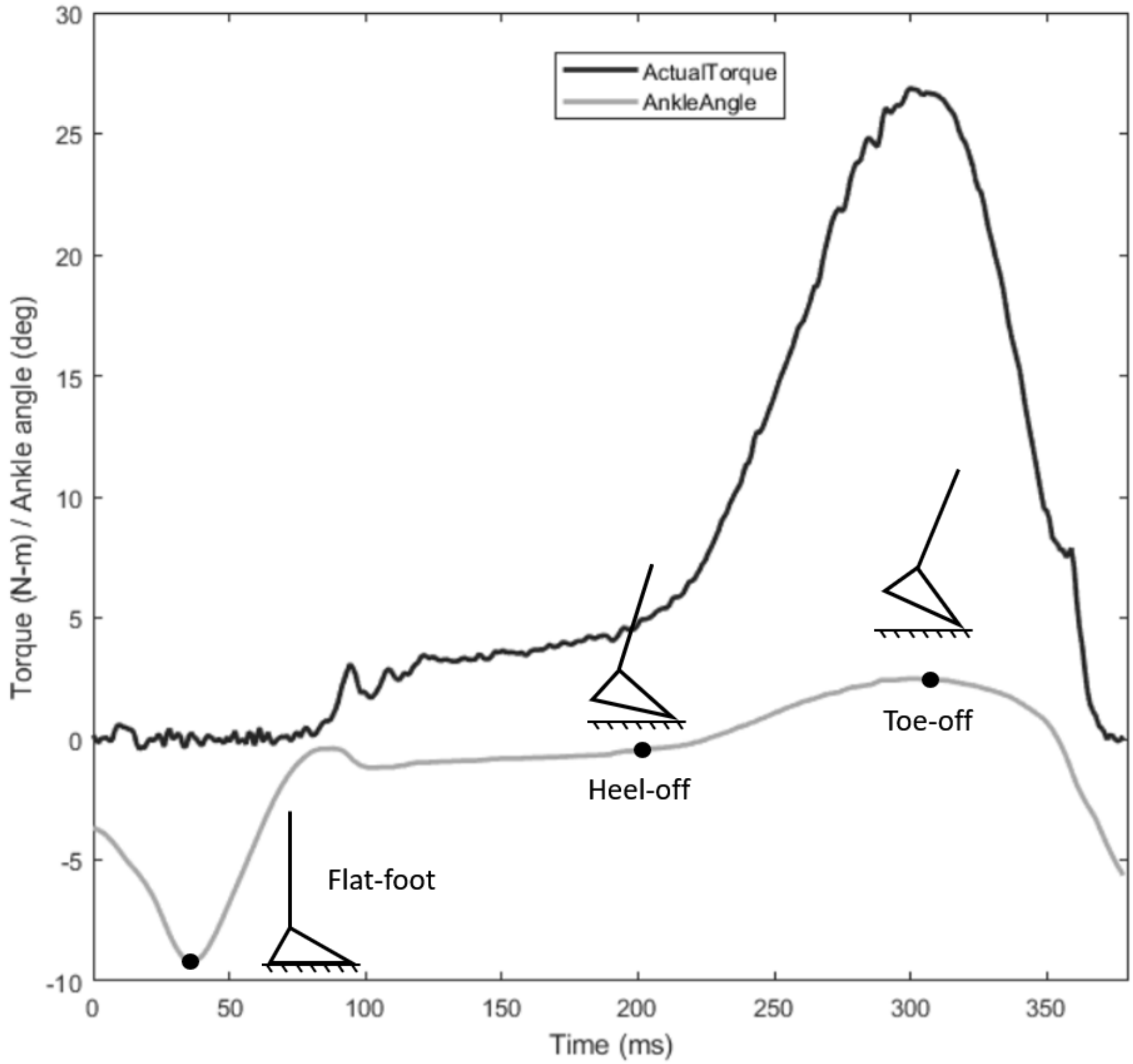

Figure 3.9: Actual measured torque vs. measured ankle angle and states. The result shows that the assistive torque is assisting push-off works.

3.3.3 Parameters

This section is investigating the relationship between the parameter and the measured torque. The tuning parameters affect the desired torque trajectory, however, the actual effect on the measured torque must agree the difference according to the tuned parameter. The performance is evaluated by comparing three different torque curves, determined by two-parameter sets of start and peak timings (Fig. 3.11). The definition of start percentage decides at which percentage to initiate assistance during the flat-foot state. In contrast, the peak percentage decides at which percentage to reach the peak assistive torque during the flat-foot state. (e.g., start percentage = 50, peak percentage = 100; means assistive torque will initiate during 50% of the flat-foot state, reaches the peak at the end of the flat-foot state). The parameter sets start with Start Percentage (startPer) 60 and Peak Percentage(peakPer) 90 as a baseline, the following two sets are 1. Increase peakPer by 10, 2. Decrease startPer by 10 accordingly. The measured torque data are the average of randomly picked 10 steps from each parameter set. With lower startPer, the assistance will start earlier, while with lower peakPer, the peak assistive torque will come faster, with above principle and figure 3.10, we can see startPer 50 and peakPer 90 covers bigger area since the assistance start earlier and it reaches the peak faster.

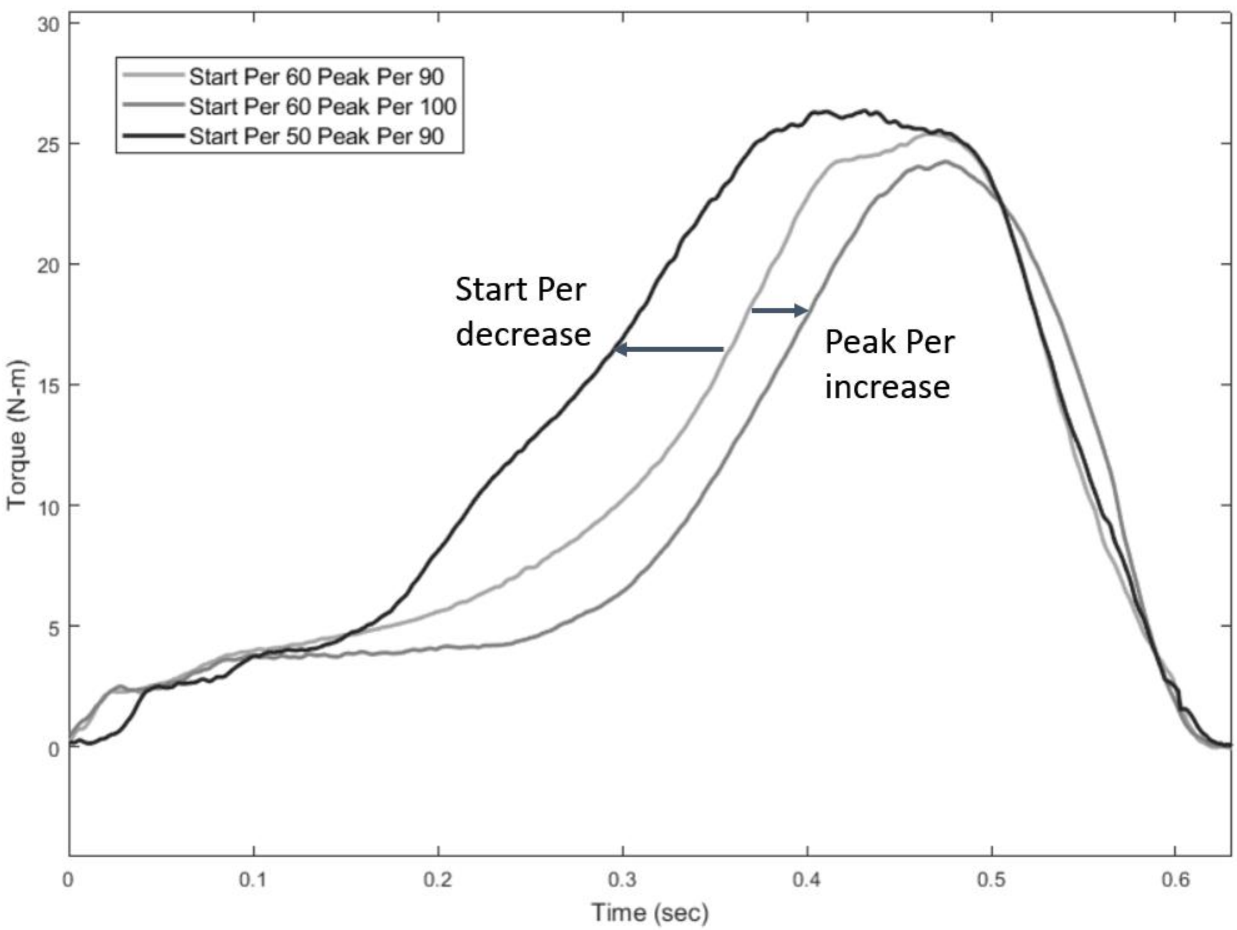

Figure 3.10: The effect of tuning the parameter compared to the actual measured torque. Based on the starting condition (startPer = 60, peakPer = 90), this graph compared the effect of reducing the start percentage and increase the peak percentage, it shows the combination of an early start and early peak covers the biggest area creates the highest work among all three sets. The measured torque trajectory is the average of randomly picked 10 steps for each different condition.

3.3.4 Metabolic cost

We further examined the effect of different parameter sets on the metabolic cost. Six conditions of experiment sessions are conducted, each session contained 5 minutes of walking, follow up with 5 minutes rest for the subject to recover. The average VO2 and VCO2 of the last minute are used for metabolic cost calculation calculated by using eq. 3.4. Table I shows the experiment result of the metabolic cost reduction according to different parameter sets. Condition 2 shows the best result, reducing the metabolic cost by 10.82% comparing to unpowered condition, although Fig 3.11 shows condition 1,4 and 5 may have higher total work than condition2 those work did not contribute to lowering metabolic cost, this suggested that rather than provide higher work during the walk, timing of assistive torque has greater effect for assisting human walking. This experiment result is limited to one subject to explore the relationship between metabolic cost and parameters, it cannot be concluded for all the human subjects since each person may consist of different preferences to the parameters. However, the result shows a promising achievement that the system can be used for walking assistance.

TABLE I

Metabolic cost reductions of different parameter conditions

	Condition	Metabolic cost (W)	Reduction
1	Start Per 60 / Peak Per 80	353.72	7.67%
2	Start Per 60 / Peak Per 90	341.66	10.82%
3	Start Per 60 / Peak Per 100	350.24	8.58%
4	Start Per 40 / Peak Per 90	357.88	6.58%
5	Start Per 50 / Peak Per 90	374.07	2.36%
6	Unpowered	383.10	Baseline

3.4 Discussion

This study demonstrated the core of a portable actuator that exhibited remarkable metabolic cost reduction with tunable assistive torque. Although the tracking performance of estimated torque and actual measured torque is far from optimal, the largest tracking error occur exhibited early in the early stage, the later stage also has a drastic fall before the desired torque goes down, further study is still required to compensate the difference. The best result of metabolic cost reduction is 10.82% lower comparing to unpowered condition, with the maximum assistive torque is 25 N-m measured on the ankle-foot orthosis, this shows the assistive torque-assist push-off works successfully. Optimization for different individuals is still required to personalize the assistive torque trajectory, human-in-the-loop optimization has shown great performance for customizing control parameters by including metabolic cost measurement from the subject (Kim et al, 2017). Overall, the system proved positively run in

EtherCAT communication, using Simulink as a controller model with a 500Hz update rate and can contribute to metabolic cost reduction with an off-board benchtop test. In the future work, I aimed to make the system on-board, other sub-systems such as battery system, motor housing, and a robust gait detection are still needed for development.

4. SLOPE DETECTION

4.1 Introduction

In the robotic field, sensor is one of the essential components for construct a robotic system, the sensor is mostly used for observing the system to provide essential information to the controller then compute the command accordingly. Visual sensing has been broadly used on robots for terrain detection or navigation (Wellhausen, et al; 2019) or other application such as autonomous vehicles, while in the wearable robot field, visual sensing has been used for gait mode detection (e.g.: walking on stair, uphill, downhill, level ground), Zheng's research uses laser distance sensor for detecting stair or level ground to recognize the gait mode (Zheng et al; 2011). To let the wearable device work with different environments, versatile environmental sensing can be used to ensure the robustness of the system. In this study, we want to explore the possibility of using an RGB-D camera to achieve environmental detection (e.g.: uphill, downhill, level ground), the advantage of the RGB-D camera is it's expendable nature since there are uncountable possibilities can be achieved by having the information from the RGB-D camera. In our approach, we used an Intel d435i RGB-D camera (Intel, [CA, USA]) which also has IMU sensor, we implemented a plain detection algorithm to detect gait modes such as uphill, downhill and level ground, a further application such as wall or stair detection can be added based on the current method. After having the information of different environmental conditions, the controller can adjust the programmed assistive torque accordingly.

4.2 Method

The need for visual sensing is this project is to provide the environment input for high-level gait assistance controller to optimize the control parameters to provide the best assistance according to the different terrain. The high-level gait assistance controller utilizes environmental sensing input from the depth-sensing camera Intel d435i RGB-D camera (Intel, [CA, USA]) with inertial measurement unit IMU.

4.2.1 Hardware & software

Intel realsense camera used in this project utilizes the Intel Realsense Software Development kit (SDK) an open-source software that includes librealsense2 library. This library has support for many popular programming languages (Python, MATLAB, Java, etc.) but we used C++ to get the full potential out of the camera. This library is able to collect different streams from the camera pipeline, for example, RGB images, Depth images, motion data, IR images, also some post-processing capabilities like filtering and point cloud generation. depth images and motion data were the two streams used in this project.

Point cloud library is another open-source library used for post-processing the point clouds. It has plenty of features for processing point clouds like filtering, feature detection, segmentation, sample consensus, etc.

4.2.2 Slope detection algorithm

The camera produces the point cloud from the depth image using librealsense SDK. The point cloud generated from librealsense library processed by Point Cloud Library and filtered using cut-off filter (threshold) to remove points outside of certain range to only use points up to 2-meter distance to focus on the terrain right in front of the subject(reduce the points for faster Plane segmentation). Plane segmentation for the terrain is performed using the PCL implementation of the Random Sample Consensus (RANSAC) algorithm. RANSAC is a model-fitting algorithm that can find the best fit for given model parameters and returns the model coefficients. For the model of a plane, RANSAC returns the normal vector of that plane in-camera reference frame. The orientation of the camera in the spatial reference frame was obtained using the IMU sensor. IMU sensor provides the acceleration vector which points to the direction opposite from gravitation acceleration. Finally, the slope of the terrain in the sagittal plane is estimated by finding the angle between plane-normal-vector and acceleration-vector, the approach is visualized with fig 4.1.

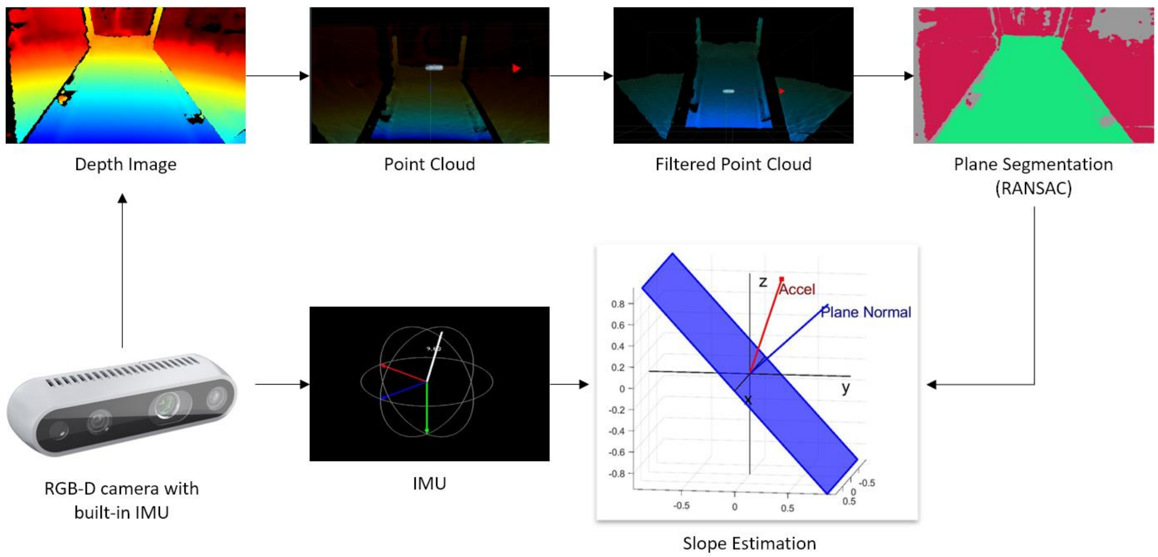

Figure 4.1: Slope detection using RGB-D camera with built-in IMU. The point cloud is filtered and segmented into planes; the ankle of the slope is computed according to IMU information.

4.3 <u>Result</u>

Plane segmentation: RANSAC method will try to find the best-fit plane for the given point cloud as shown in figure 4.2.

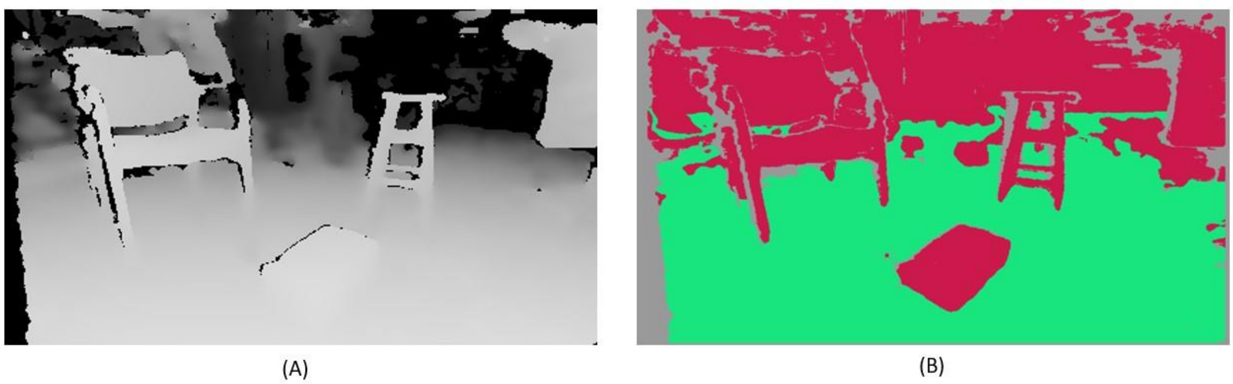

Figure 4.2: Plane segmentation. (A) shows the raw point cloud generated from depth image (B) segmentation of the floor from non-floor and shown with green color.

Figure 4.2 shows the plane segmentation performed on a single image (840×480) this same process when performed on live camera feed can get a segmentation update rate in the range of 10 to 15 frames per second without any optimization or downsampling.

Slope estimation: To understand the performance of slope estimation, the estimated degree measured by using the RGB-D camera is averaged by 3 measurements and compared with the actual degree on the treadmill (fig 4.3). The treadmill has several slope conditions represented by slope percentage as the treadmill grade in fig 4.3. The result shows errors less than 1 degree across all conditions, demonstrated a good approach of slope estimation while having an update rate of 15Hz.

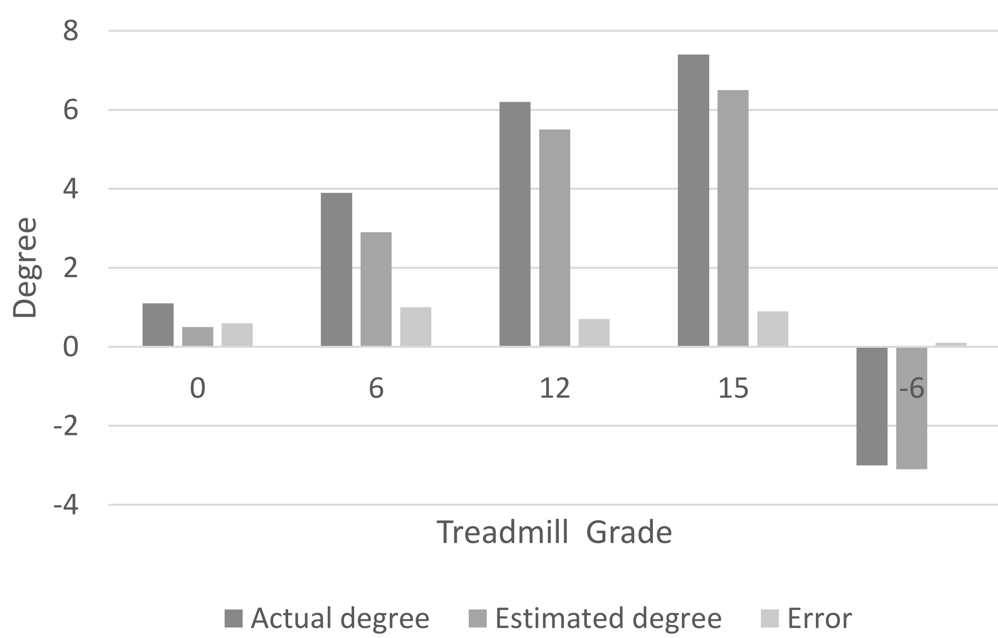

Figure 4.3: Slope estimation. The slope is estimated by using an RGB-D camera shows a reliable result of less than one-degree error.

4.4 Discussion

Although the result shows promising performance of slope detection, however, challenges such as camera shaking during human walking, local communication to the high-level controller are still needed to be solved. Improvements in the plane segmentation to remove false positive and robust camera orientation from IMU during walking needs to be done for future scope, while this are not the primary goals but enough to show the potential of this system

5. SUMMARY AND CONCLUSION

This study presented a well-functioning core of a portable system, we explored the communication, software structure, hardware setup, gait detection and gait assistive control algorithm with most commercially available components. This approach is cost-efficient and intelligent for exploring different possibilities of a portable gait assist device.

The Simulink control algorithm is deployed into a virtual target machine which is an isolated CPU, running with an update rate of 500Hz, this provides a versatile real-time programming platform to build a controller. Gait detection is simply achieved by using two pressure sensors and optimized by the control algorithm to prevent sensor failure and misfiring. The motor can provide up to 3-Nm from the motor shaft which is 25N-m measured on ankle-foot orthosis. Even with low assistive torque, the system still shows an impressive 10.82% metabolic cost reduction comparing to unpowered condition. The actuator system also presented a reasonable resolution, depending on the assistive torque trajectory parameters, the metabolic cost reduction was varied between 2~11%. This result suggests that this actuation system can be used to various control ideas including human-in-the-loop optimizations.

Visual sensing done with an RGB-D camera shows the capability of plane segmentation and slope detection with less than 1 degree of error, we only explored one of the applications while there are still uncountable applications can be achieved by using an RGB-D camera.

The current structure has proved as a good foundation for developing a portable gait assistance device, despite the positive result we have got, the system is still far from complete. The future work will include the battery system, motor housing, harness, algorithm refinement, human-in-the-loop optimization, and local communication.

CITED LITERATURE

Asbeck, A. T., Schmidt, K., & Walsh, C. J. (2015). Soft exosuit for hip assistance. Robotics and Autonomous Systems, 73, 102–110. https://doi.org/10.1016/j.robot.2014.09.025

Cain, S. M., Gordon, K. E., & Ferris, D. P. (2007). Locomotor adaptation to a powered ankle-foot orthosis depends on control method. *Journal of NeuroEngineering and Rehabilitation*, *4*, 1-13.

Cena, G., Bertolotti, I. C., Scanzio, S., Valenzano, A., & Zunino, C. (2012). Evaluation of EtherCAT distributed clock performance. *IEEE Transactions on Industrial Informatics*, *8*(1), 20–29.

Chen, B., Ma, H., Qin, L. Y., Gao, F., Chan, K. M., Law, S. W., … Liao, W. H. (2016). Recent developments and challenges of lower extremity exoskeletons. *Journal of Orthopaedic Translation*, 5, 26–37. Retrieved from https://doi.org/10.1016/j.jot.2015.09.007

Dean, C. M., Richards, C. L., & Malouin, F. (2001). Walking speed over 10 metres overestimates locomotor capacity after stroke. *Clinical Rehabilitation*, 15(4), 415–421.

Díaz, I., Gil, J. J., & Sánchez, E. (2011). Lower-Limb Robotic Rehabilitation: Literature Review and Challenges. *Journal of Robotics*, *2011*, 759-764.

Dollar, A. M., & Herr, H. (2008). Lower extremity exoskeletons and active orthoses: Challenges and state-of-the-art. *IEEE Transactions on Robotics*, *24*(1), 144-158.

Duysens, J., Tax, A. A. M., van der Doelen, B., Trippel, M., & Dietz, V. (1991). Selective activation of human soleus or gastrocnemius in reflex responses during walking and running. *Experimental Brain Research*, *87*(1), 193–204.

Galle, S., Malcolm, P., Collins, S. H., & De Clercq, D. (2017). Reducing the metabolic cost of walking with an ankle exoskeleton: interaction between actuation timing and power. *Journal of NeuroEngineering and Rehabilitation*, *14*(1), 1–16.

H. I. Krebs, N. Hogan, M. L. Aisen and B. T. Volpe, "Robot-aided neurorehabilitation," in *IEEE Transactions on Rehabilitation Engineering*, vol. 6, no. 1, pp. 75-87, March 1998.

Jørgensen, H. S., Nakayama, H., Raaschou, H. O., & Olsen, T. S. (1995). Recovery of walking function in stroke patients: The copenhagen stroke study. *Archives of Physical Medicine and Rehabilitation*, 76(1), 27–32.

Kawahira, K., Shimodozono, M., Ogata, A., & Tanaka, N. (2004). Addition of intensive repetition of facilitation exercise to multidisciplinary rehabilitation promotes motor functional recovery of the hemiplegic lower limb. *Journal of Rehabilitation Medicine*, 36(4), 159–164.

Kim, M., Liu, C., Kim, J., Lee, S., Meguid, A., Walsh, C. J., & Kuindersma, S. (2019). Bayesian optimization of soft exosuits using a metabolic estimator stopping process. *Proceedings - IEEE International Conference on Robotics and Automation*, *2019-May*, 9173–9179.

Kim, J., Lee, G., Heimgartner, R., Revi, D. A., Karavas, N., Nathanson, D., … Walsh, C. J. (2019). Reducing the metabolic rate of walking and running with a versatile, portable exosuit. *Science*, *365*(6454), 668–672.

Lang, C. E., MacDonald, J. R., Reisman, D. S., Boyd, L., Jacobson Kimberley, T., Schindler-Ivens, S. M., … Scheets, P. L. (2009). Observation of Amounts of Movement Practice Provided During Stroke Rehabilitation. *Archives of Physical Medicine and Rehabilitation*, 90(10), 1692–1698.

Langlois, K., Van Der Hoeven, T., Rodriguez Cianca, D., Verstraten, T., Bacek, T., Convens, B., … Vanderborght, B. (2018). EtherCAT tutorial: An introduction for real-time hardware communication on windows [tutorial]. *IEEE Robotics and Automation Magazine*, *25*(1).

Lee, Y., Lee, J., Choi, B., Lee, M., Roh, S., Kim, K., … Shim, Y. (2019). Flexible Gait Enhancing Mechatronics System for Lower Limb Assistance (GEMS L-Type). *IEEE/ASME Transactions on Mechatronics*, 24(4), 1520–1531.

Mackey, J., & Mensah, G. (2004). *WHO 2004_atlas oh heart disease and stroke.pdf* (p.9).

N. S. K. Ho., K. Y. Tong., X. L. Hu., K. L. Fung., X. J. Wei., W. Rong., E. A. Susanto. (2011). An EMG-driven exoskeleton hand robotic training device on chronic stroke subjects: Task training system for stroke rehabilitation," *2011 IEEE International Conference on Rehabilitation Robotics*, Zurich, 2011, pp. 1-5.

Pasinetti, S., Lancini, M., & Borboni, A. (2013). *EMG SIGNAL-ONLY GAIT CYCLE DETECTION ALGORITHM*. (August).

Rathore, S. S., Hinn, A. R., Cooper, L. S., Tyroler, H. A., & Rosamond, W. D. (2002). Characterization of incident stroke signs and symptoms findings from the atherosclerosis risk in communities' study. *Stroke*, *33*(11), 2718–2721.

Robert, J., Georges, J.-P., Rondeau, E., & Divoux, T. (2014). Minimum Cycle Time Analysis of Ethernet-Based Real-Time Protocols. *International Journal of Computers Communications & Control*, *7*(4), 744.

Sankaranarayan H, Gupta A, Khanna M, Taly AB, Thennarasu K. Role of ankle foot orthosis in improving locomotion and functional recovery in patients with stroke: A prospective rehabilitation study. J Neurosci Rural Pract. 2016;7(4):544–549.

Shorter, K. A., Kogler, G. F., Loth, E., Durfee, W. K., & Hsiao-Wecksler, E. T. (2011). A portable powered ankle-foot orthosis for rehabilitation. *Journal of Rehabilitation Research and Development*, *48*(4), 459–472.

Suzuki, T., Chino, K., Fukashiro, S. (2014). Gastrocnemius and soleus are selectively activated when adding knee extensor activity to plantar flexion. *Human Movement Science*, 36, 35-45.

Tyson, S. F., Hanley, M., Chillala, J., Selley, A., & Tallis, R. C. (2006). Balance disability after stroke. *Physical Therapy*, *86*(1), 30–38.

Walsh, C. J., endo, K. E. N., & HERR, H. (2007). A quasi-passive leg exoskeleton for load-carrying augmentation. *International Journal of Humanoid Robotics*, 04(03), 487–506.

Wade, D. T. (2003). Community rehabilitation, or rehabilitation in the community? *Disability and Rehabilitation*, 25(15), 875–881.

Wellhausen, L., Dosovitskiy, A., Ranftl, R., Walas, K., Cadena, C., & Hutter, M. (2019). Where should I walk (Predicting terrain properties from images via self-supervised learning. *IEEE Robotics and Automation Letters*, *4*(2), 1509–1516.

Zhang, Fan & Fang, Zheng & Liu, Ming & Huang, He. (2011). Preliminary Design of a Terrain Recognition System. Conference proceedings: ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference. 2011. 5452-5. 10.1109.

Zoss, A. B., H. Kazerooni and A. Chu, "Biomechanical design of the Berkeley lower extremity exoskeleton (BLEEX)," in IEEE/ASME Transactions on Mechatronics, vol. 11, no. 2, pp. 128-138, April 2006.

VITA

NAME: AN-CHI HE

EDUCATION: B.A., Mechanical Engineering, National Chung Cheng University, Chiayi, Taiwan, 2013

M.S., Mechanical Engineering, University of Illinois at Chicago, Chicago, Illinois, 2020

TEACHING: Department of Mechanical Engineering, University of Illinois at Chicago; Assistive Robotic course project mentor for Engineering Students, 2019