

Continuous Gait Phase Estimation using LSTM for Robotic Transfemoral Prosthesis Across Walking Speeds

Jinwon Lee, *Member, IEEE*, Woolim Hong, *Student Member, IEEE*, and Pilwon Hur, *Member, IEEE*

Abstract—User gait phase estimation plays a key role for the seamless control of the lower-limb robotic assistive devices (e.g., exoskeletons or prostheses) during ambulation. To achieve this, several studies have attempted to estimate the gait phase using a thigh or shank angle. However, their estimation resulted in some deviation from the actual walking and varied across the walking speeds. In this study, we investigated the different setups using for the machine learning approach to obtain more accurate and consistent gait phase estimation for the robotic transfemoral prosthesis over different walking speeds. Considering the transfemoral prosthetic application, we proposed two different sensor setups: i) the angular positions and velocities of both thigh and torso (S1) and ii) the angular positions and velocities of both thigh and torso, and heel force data (S2). The proposed setups and method are experimentally evaluated with three healthy young subjects at four different walking speeds: 0.5, 1.0, 1.5, and 2.0 m/s. Both results showed robust and accurate gait phase estimation with respect to the ground truth (loss value of S1: 4.54e-03 Vs. S2: 4.70e-03). S1 had the advantage of a simple equipment setup using only two IMUs, while S2 had the advantage of estimating more accurate heel-strokes than S1 by using additional heel force data. The choice between the two sensor setups can depend on the researchers' preference in consideration of the device setup or the focus of the interest.

Index Terms—Transfemoral prosthesis, gait phase estimation, machine learning.

I. INTRODUCTION

HUMAN gait phase estimation has been studied for clinical or rehabilitation purpose [1]–[4] and for developing assistive robotic devices, such as powered prostheses [5]–[9] or exoskeletons [10], [11]. For the clinical purpose, a set of vision-based motion capture systems and force plate is conventionally utilized to evaluate the rehabilitation process or to monitor patient's gait abnormalities by observing patient's motion in the well-controlled space [2], [4]. Gait phase estimation is also crucial to properly control assistive devices based on the user's walking state [5]–[9]. Even though the aforementioned built-in system has the advantage of precision and accuracy, this setup is not suitable for assistive devices due to the requirement of an on-board estimation

Jinwon Lee is with the School of Mechanical Engineering, Korea University, Seoul, 02841, South Korea. (e-mail: jinwonlee@korea.ac.kr).

Woolim Hong is with the Department of Mechanical Engineering, Texas A&M University, College Station, TX 77840, USA (ulim8819@tamu.edu).

Pilwon Hur is with the School of Mechanical Engineering, Gwangju Institute of Science and Technology, Gwangju 61005, South Korea. He is also affiliated with the Department of Mechanical Engineering, Texas A&M University, College Station, TX 77843, USA (email: pilwonhur@gist.ac.kr, pilwonhur@tamu.edu).

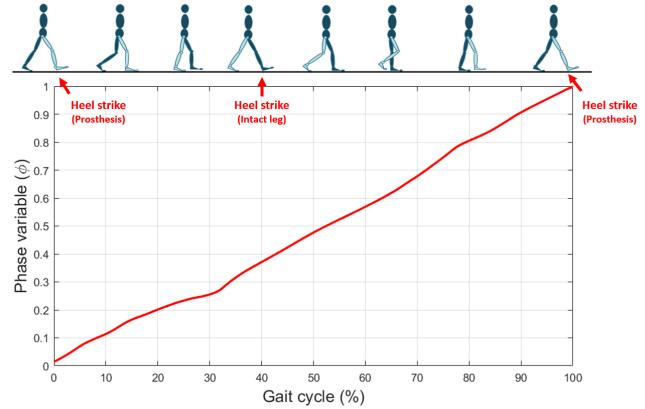


Fig. 1. A phase variable ($\Phi \in [0,1]$) can be mapped into a human gait cycle (x-axis: 0–100%), indicating the user's walking progression

of the user's gait phase in the daily living. Thus, human gait phase estimation using wearable sensors has been widely studied to control assistive devices. In this study, we limit our scope to the robotic prosthesis for developing the gait phase estimation model. To achieve the seamless control for such robotic devices, researchers have attempted to parameterize the wearable sensor data into a single kinematic variable (i.e., phase variable) to represent the user gait progression (see Fig. 1). The phase variable is bounded on $[0,1]$, and is a function of gait kinematic variables. It represents the gait phase in a continuous and monotonic manner [6]–[9].

To obtain the phase variable for the robotic prosthesis control, many researchers attempted to use a thigh angle [5]–[7] or a linearized hip position [8], [9] of the user. For example, Thatte et al. [5] utilized a sensor-fusion approach using angular position and velocity from the inertia measurement unit (IMU) at the thigh and the encoders at the ankle and the knee, respectively. They achieved a robust and adaptive gait phase estimation during the stance phase via an extended Kalman filter (EKF). However, due to the missing estimation at the swing phase, its application would be limited when the swing phase estimation is needed. To estimate the user's gait phase over the entire gait cycle, a linearized hip position was proposed as the phase variable in [9]. The linearized hip position is known as one of the suitable candidates to represent the human gait phase since it monotonically varies along with the user's movement [8], [9]. It can be calculated by the inverse kinematics using the angles from the shank

TABLE I
GAIT PHASE ESTIMATION STUDIES FOR THE ROBOTIC ASSISTIVE DEVICE

Reference	Application	Method	Sensor input	Walking speed
Thatte et al. [5]	Transfemoral prosthesis	Extended Kalman filter (EKF)	Thigh (IMU) Ankle & knee (encoder) Heel (force sensor)	0.8, 1.2, 1.6 m/s
Zhao et al. [9]	Transfemoral prosthesis	Linearized hip position	Thigh & shank (IMU) Ankle & knee (encoder) Heel (force sensor)	1.3 mph (0.58 m/s)
Quintero et al. [6]	Transfemoral prosthesis	Thigh angle estimation	Thigh (IMU) Heel (force sensor)	1.5, 2.0, 2.5, 2.7 mph (0.67, 0.89, 1.12, 1.21 m/s)
Zago et al. [12]	Normal gait	Decision trees for gait event detection	Foot (IMU) Ground reaction forces	0.6 - 2.0 m/s
Xu et al. [13]	Transfemoral prosthesis	Adaptive Oscillators	Shank (IMU)	Self-selected three speeds (slow, normal, fast)
Seo et al. [10]	Ankle exoskeleton	Long short-term memory (LSTM) & Gated recurrent unit (GRU)	Shank (IMU) Heel & toe (force sensor)	Varying speeds (no details)
Kang et al. [11]	Hip exoskeleton	Long short-term memory (LSTM)	Thigh & torso (IMU) Hip (encoder) Heel (force sensor)	0.6 - 1.0 m/s with 0.05 increment
Vu et al. [14]	Transtibial prosthesis	Recurrent neural network (RNN)	Shank (IMU) Heel & toe (force sensor)	2.2, 2.6, 3.2, 3.8 m/s

and the thigh. Yet, two IMUs are required at the intact leg of the user to obtain the hip position. This could limit the autonomy of the prosthesis by mirroring the information from the intact leg to control the prosthesis at the other side. Also, using sensors on the intact side would not be preferred by the user because having a sensor set on the intact side may cause discomfort during daily use of the prosthesis. Instead, Villarreal et al. [15]–[17] proposed a different sensor location (i.e., thigh) to parameterize the gait phase without the sensor set at the intact side. The thigh angle has been widely used as the phase variable because it shows a reasonable estimation with a single IMU only at the residual leg even in several walking scenarios: slope [6], [7], [18], perturbation [16], and different speeds [6], [19]. However, an early saturated phase variable (around 80-90%) was found in [18], [19] due to a varying range of motion of the thigh for each step. Moreover, the thigh angle could highly vary under the perturbation or the disturbance during walking, particularly in the swing phase [16], [18]. Thus, relying on only a single sensor input may not be the best idea for a robust gait phase estimation.

Thanks to rapid advancements in machine learning, gait phase estimation methods have evolved using multiple sensor information. For instance, Seo et al. [10] estimated the gait phase for their ankle exoskeleton, relying on a shank-mounted IMU. To achieve a real-time estimation for their phase-based control, they proposed to use Recurrent Neural Networks (RNNs) with additional information from the foot pressure sensors during their model training. Similarly, in [11], researchers also utilized a neural network learning model to estimate the gait phase based on sensor-fusion: the encoders at the hip, IMUs at the thigh and trunk. Regardless of the participants, they both achieved more robust and accurate estimation

of a continuous variable with the learning technique [10], [11]. These methods, however, are not applicable to the lower-limb prostheses due to the difference between the exoskeleton and the prosthesis. For instance, the proposed exoskeletons were designed to assist the ambulation of individuals with paraplegia, thus the researchers more focused on estimating the gait phase at slow walking speeds [10], [11]. This is because the exoskeletons for paraplegics are prone to be operated in relatively lower speeds [20]–[24] than the prostheses [6], [25] due to the limited ambulation of paraplegics. Even though some exoskeletons can achieve faster ambulation, the crutches are required for those systems [26]–[28]. So, a gait phase estimation at faster walking speeds for the prosthesis application is needed. More importantly, a transfemoral prosthesis user has limited control over the shank's orientation. For the prosthesis application, [14] reported a good estimation result using a deep learning algorithm; however, researchers still rely on the data from the lower shank when they estimated the user gait. Lower shank information can be used only with the transtibial amputees, not with the transfemoral amputees for the gait phase estimation. This is because, in the case of transfemoral amputees, the shank part is dictated by the prosthetic control, not by the user him/herself due to the knee joint loss. Thus, considering the transfemoral prosthetic control, we need to find an appropriate sensor location which can be fully controlled by the user regardless of the limb-loss.

To the authors' knowledge, there are no machine learning-based gait phase estimation models for transfemoral prosthesis using a wearable sensor set. Thus, the aim of this study is to implement a Long Short-Term Memory (LSTM) method for continuous gait phase estimation based on appropriate sensor sets for a transfemoral prosthesis. In Section II, we introduce

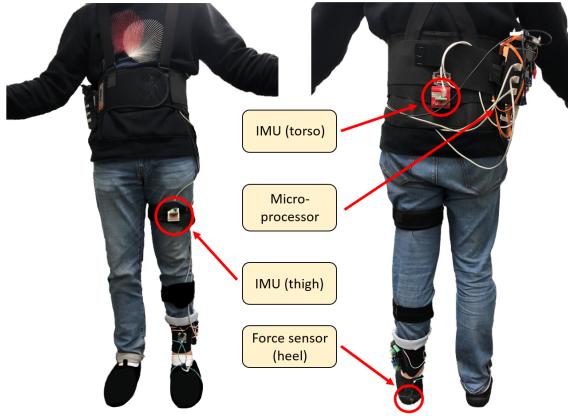


Fig. 2. A wearable sensor set for the gait phase estimation: two IMUs at the thigh and torso, and a force sensor at the heel

the wearable sensor setup candidates (2 sensors Vs. 3 sensors) to estimate the user gait phase. Also, a training process with the proposed neural network model is described based on the nominated sensor setup. The estimation model is evaluated and tested in Section III with discussion. Finally, we conclude this article in Section IV.

II. METHODS

In this section, we introduce a wearable sensor setup for estimating the user's gait in the prosthesis application. Then, we explain how the gathered data is differed by the gait phase and how to label it. We describe the network architecture composed of LSTMs and bidirectional-LSTMs (Bi-LSTMs) for the gait phase estimation.

A. A wearable sensor set for gait phase estimation

A wearable sensor set to obtain the input data for the learning of gait phase is described in Fig. 2. Two 9-axis IMUs (MPU9150, SparkFun Electronics, USA) were located at the subject's thigh and torso to measure their global angles. Also, a force-sensitive resistor (FSR: FlexiForce A502, Tekscan, USA), attached under the shoe insole, was used during locomotion to detect the heel-strike event for initializing the gait cycle. All the sensor data was collected by a micro-processor (BeagleBone Black, Texas Instruments, USA) at 200 Hz.

B. Gait phase division

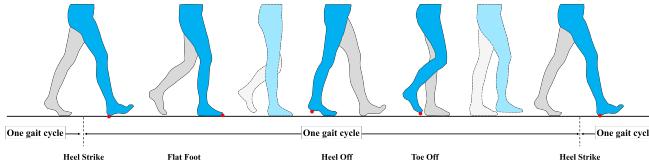


Fig. 3. Human gait cycle with important kinematic changes

Human walking can be defined as a single gait cycle. A gait cycle is conventionally defined from a heel-strike to the next heel-strike on the same leg. This gait cycle also can be divided

into several sub-phases as in Fig. 3: heel-strike (HS), flat-foot (FF), heel-off (HO), toe-off (TO). Gait phase percentage differs depending on the walking speed and personal gait characteristics. However, in general, the faster the individual walks, the smaller the percentage of HO and TO becomes [29]. As per [29], at 0.6 m/s of the walking speed, HO and TO occur at 51% and 67%, respectively, while a walking speed of 1.6 m/s results in HO at 31% and TO at 53%. In the case of HS and HO, the gait phase can be found through the force sensor at the heel, even though it is difficult to determine the percentage of HO that depends on walking speed and gait characteristics. Thus, some adaptive estimation methods based on the simultaneous kinematics/kinetics sensor data are needed to achieve more robust estimation.

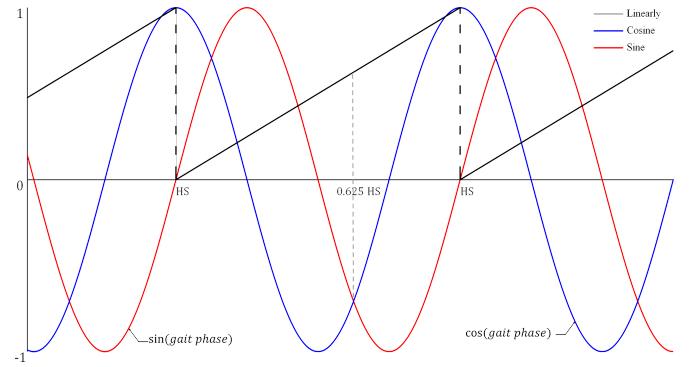


Fig. 4. Gait phase transformation. To eliminate the discontinuity, gait phase is transformed into cosine and sine. For example, the gait phase of 0.625 is the same as the pair of cosine and sine, (-0.707, -0.707).

With two IMUs (e.g., thigh and torso) and heel FSR sensor shown in Fig. 2, position and velocity of the thigh and torso, and heel force data are recorded in 200 Hz. The HS is widely used as a gait phase initialization because the gait cycle is conventionally defined from HS on the same leg to another HS. The heel FSR was used to find the heel-strike event. When the heel force sensor value exceeded a certain threshold, we set this point as the HS. Taking into account the characteristics of heel FSR, which is dependent on mass, 15% of the weight was set as the threshold. We label the data in two different ways with the point of HS: i) a linear interpolation method, and ii) a polar coordinate encoding method. The linear interpolation method uses the threshold of the heel sensor to obtain HS points, and then linearly connects two consecutive HS points. As a result, the estimated gait phase linearly increases from 0 to 1, and resets to 0 at the end of the gait cycle. As shown with the black line in Fig. 4, when the linear interpolation is used, the discontinuity cannot be avoidable at the HS. This discontinuity around HS causes unexpected problems when calculating the loss value in the training session. Loss value is used to evaluate the estimation result based on the difference between the ground truth and the prediction. When HS occurs, the label instantly turns into 0 from 1, meaning that even a single delayed sample (5 ms) at HS could result in a large difference between the ground truth and prediction. As the error due to discontinuity is much more significant than other points, the network predominantly tends to find the HS only. In order to solve this discontinuity issue, we transform the labels

into the polar coordinates [10], [11]. The percentage of the gait phase, P , can be represented as an angle θ between 0 and 2π with equation 1. And then, we transform the polar coordinate with equation 2 and 3 as new labels x_1 and x_2 . They represent a continuous sinusoidal function, which is bounded in -1 and 1.

$$\theta = P \times \frac{2\pi}{100} \quad (1)$$

$$x_1 = \cos \theta \quad (2)$$

$$x_2 = \sin \theta \quad (3)$$

The output of the neural network is represented by the polar coordinate, (y_1, y_2) . To be used as a phase variable for the prosthesis control, the estimated phase should be bounded in $[0,1]$. Thus, additional transformation is needed as explained in equations 4 and 5 [10], [11].

$$\|\hat{\theta}\| = \tan^{-1}(\frac{y_2}{y_1}) \quad (4)$$

$$\hat{P} = \begin{cases} \frac{\|\hat{\theta}\|}{2\pi} & y_1 > 0 \& y_2 \geq 0 \\ \frac{\|\hat{\theta}\|}{2\pi} + \frac{1}{2} & y_1 < 0 \\ \frac{\|\hat{\theta}\|}{2\pi} + 1 & y_1 > 0 \& y_2 < 0 \end{cases} \quad (5)$$

C. Neural networks training

1) *Long short-term memory (LSTM)*: RNN has been widely used in the field of translation, text, and time series prediction. In this type of data, past events may affect future results. However, RNN suffers from vanishing gradients as the number of steps increases. In order to resolve the vanishing gradient issue, the LSTM was proposed [30].

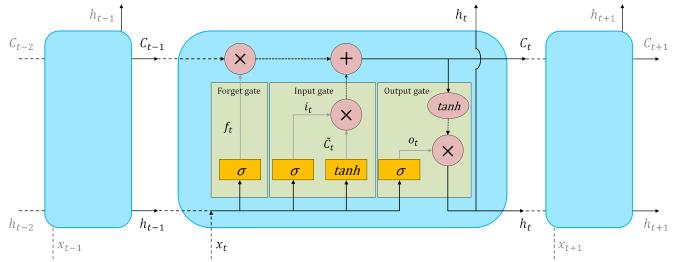


Fig. 5. Typical structure of LSTM block. Yellow squares and pink circles represent neural network layer and pointwise operation, respectively.

LSTM learns by adding or removing information through the cell state, c_t , just like a conveyor belt, and three gates: forget f_t , input i_t , and output o_t . The forget gate (f_t) determines what information to forget and what information to keep, while the input gate (i_t) determines whether new information is stored in the cell state. The output gate (o_t) determines what information is output from the cell state. Input gates are determined by concatenating the previous output (h_{t-1}) and the current input (x_t). The input value that passed through both the forget and the input gates becomes a new cell state (C_t) with the previous cell state (C_{t-1}) and operation. The new cell state is transferred to the next block, and at the same

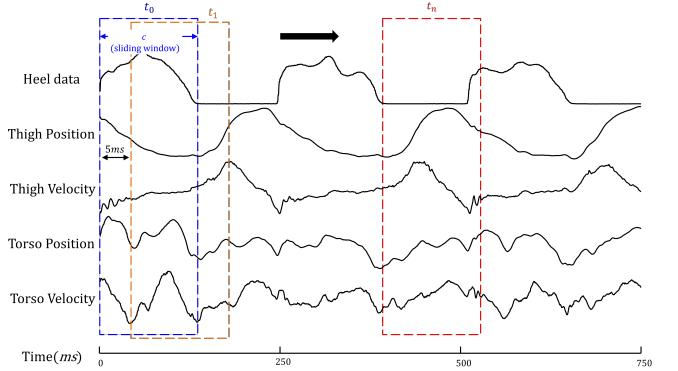


Fig. 6. Sliding window sequence from the obtained input data. c is the length of sliding windows sequence.

time, an output value (h_t) is generated through the output gate (o_t).

However, LSTM has a limitation in that the output tends to converge based on a straight pattern since the input order is chronological. In order to resolve this problem, Bi-LSTM added a backward learning to the original forward learning, which learns the past information from the future information [31]. It has an advantage for long data sequences due to bidirectional learning. Since our dataset is long, Bi-LSTM is a suitable network.

2) *Proposed Networks*: In order to learn how LSTM changes over time, a sequence called the sliding window is needed. Sliding windows bind a sequence of a certain size as a unit for training LSTM. It is important to find appropriate sizes because the results of estimation vary depending on the size of the sliding window. Fig. 6 shows sliding window for input data. Since we sampled the data at 200 Hz, there is a difference of 5 ms between the sliding window at time t_0 and the sliding window at time t_1 . As input data, we prepared datasets with two different labelings: i) a polar coordinate with heel force data, and ii) a polar coordinate without heel force data. Note that the polar coordinate without heel force data refers to the labeling with the polar coordinate along with the variables for thigh position, thigh velocity, torso position and torso velocity. We name this sensor set of four variables as S1 throughout this paper. The polar coordinate with heel force data is the dataset with the heel force data added to S1. This sensor set of five variables are named as S2.

Fig. 7 shows the stacked LSTM network architecture. If the range of raw data is too large or too small, it will not be properly trained. In order to resolve the problem, pre-processing was performed to place the raw data between 0 and 1. Several raw data are grouped as a sequence with a length of c and transferred to the LSTM layer. There are five layers for the LSTM to train the networks for gait phase estimation. In the LSTM layer, the numbers of cells (u_i) are 256, 256, 128, and 128, respectively. The unit of Fully Connected (FC) layer is two, the size of the input labels. In the output layer, we select the last value in the sequence to get the gait phase at time t .

We use Adam optimizer [32] with a learning rate of 0.001 for training. The batch size is 128, and the sequence size

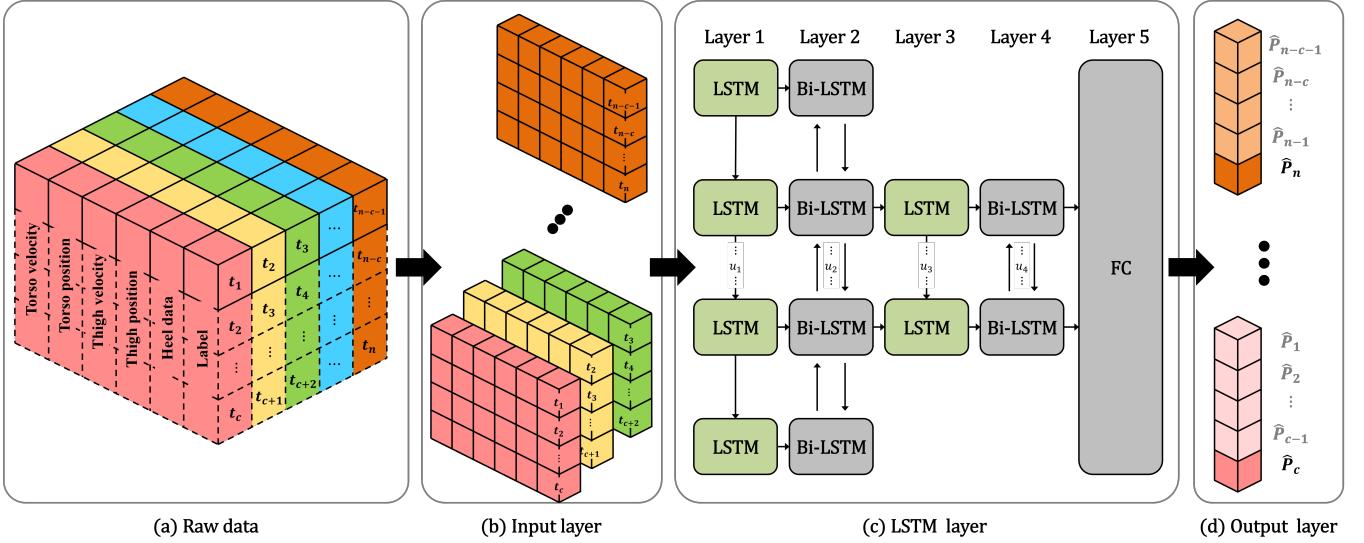


Fig. 7. Stacked network architecture. (a) Solid line is raw data recorded with sensors. Dash line is temporary data for sequence. t_n has five recorded data (i.e., heel, thigh position, thigh velocity, torso position, torso velocity) and its representative label at time n . Also, c indicates the sequence length. Label can be either one or two depending on datasets. (b) The input layer creates a sequence by raw data. Each layer consists of u_i cell. (c) The LSTM layer consists of a stacked-layer with multiple LSTMs, Bi-LSTMs, and Fully Connected (FC). (d) P_n indicates the gait phase estimation at time n .

is 300. All models are trained using the mean squared error (MSE) as the loss function, shown in equation 6.

$$MSE = \frac{1}{n} \sum_{t=1}^n (P_t - \hat{P}_t)^2 \quad (6)$$

In order to prevent overfitting, the model is trained for a maximum of 100 epochs, stopping early if the validation loss do not continue to decrease in 10 epochs. Also, we perform shuffles and randomization for the data. For each sensor setup, we divide the dataset into three subsets: 70% of the total data is used for training, 20% is used for validation, and 10% is used for testing.

III. EXPERIMENT

A. Experimental protocol

To train and validate our network model, we conducted the walking experiment on a treadmill with three healthy subjects (age of 28.3 ± 1.5 years, height of 1.70 ± 0.15 m, and weight of 65.0 ± 3.0 kg). Prior to the experiment, the subjects gave informed consent to the protocol which was reviewed and approved by the Institutional Review Board (IRB) at Texas A&M University (IRB2015-0607F). In the experiment, we have a total of five walking speed conditions: four constant speeds (0.5, 1.0, 1.5, and 2.0 m/s) and one varying speed condition: increasing from 0.5 to 2.0 m/s then decreasing to 0.5 m/s with the increment/decrement of 0.5 m/s. We changed the walking speed conditions in random order to avoid any bias in the experiment resulting from the influence of unknown factors that may affect the experiment. Before the data recording, 10 minutes of practice walking was provided to each subject to get used to the experimental setup. The calibration process for IMU was also preceded for 10 seconds at the neutral position of each subject. The proposed models were trained on a computer with I7-9700 CPU (2.6 GHz) and NVIDIA

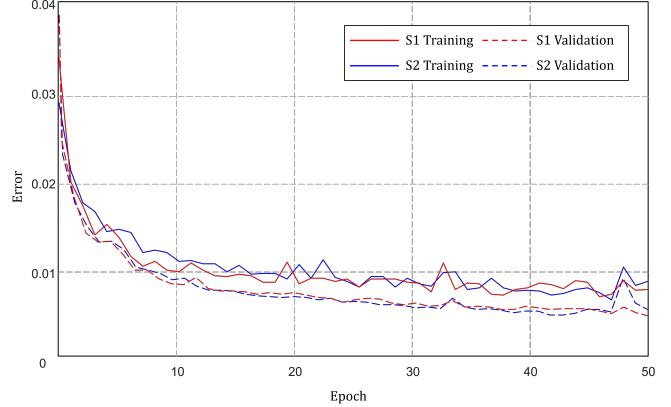


Fig. 8. MSE loss value curve for epoch. Solid lines and dashed lines indicate training and validation, respectively.

RTX 2060m GPU (6 GB). More details of the proposed work can be found in [33] as an open-source for encouraging future developments.

B. Experimental result

As we mentioned in Section II.C, we trained the network model in the polar coordinate. Fig. 8 shows the loss value curve for S1 and S2. Based on the loss value, the validation results for both S1 and S2 were similar to the ground truth with small errors (S1: 4.54e-03 Vs. S2: 4.70e-03). Both models were appropriately trained without under- and over-fitting. The trainable parameters of S1 and S2 are 957,698 and 958,722, respectively.

Fig. 9 depicts the gait phase estimation error between the ground truth and the testing prediction for 16 slow steps ((a) 0.5 m/s) and 16 fast steps ((b) 1.5 m/s). Negative errors at slow walking speed suggest that the predicted value is

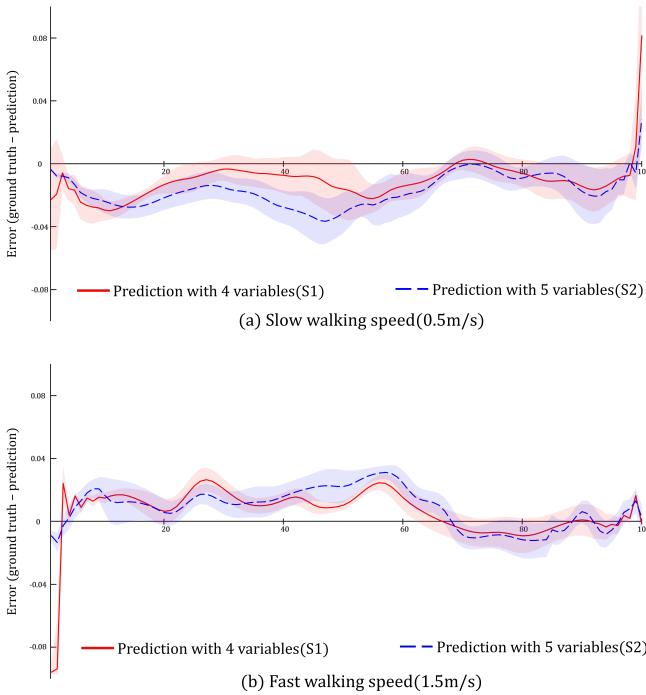


Fig. 9. Gait phase estimation error between ground truth and prediction at two different walking speeds: (a) 0.5 and (b) 1.5 m/s. Red indicates the error of S1 setup, and blue indicates the error of S2 setup. All the lines and shaded regions represent the mean and ± 1 standard deviation of 16 steps, respectively.

greater than the ground truth during slow walking. On the other hand, positive errors at fast walking speed mean the predicted is smaller than the ground truth during fast walking. This indicates that the prediction is either faster or slower than the ground truth, depending on the speed. The average error for the slow walking speed was $1.67 \pm 1.36\%$, while for the fast walking speed, the error was $1.45 \pm 1.47\%$. At the slow walking, the largest differences were found during the mid-stance (40-60%) of the gait cycle with the largest standard deviation. This is because the sensor inputs (i.e., thigh and torso) deviated more when humans tried to balance themselves during the load transferring at the mid-stance phase. As the walking becomes slower, the stance phase maintains longer, thus the sensor inputs might deviate more due to the longer mid-stance phase. On the other hand, at the fast walking, there was no noticeable difference with a small standard deviation. Regardless of walking speed, S1 had a greater error than S2 near 0 or 100% of the gait phase.

Table II shows the averaged HS detection errors of 12 consecutive steps at from 0.5m/s to 2.0 m/s. It is clearly shown that the errors in S1 are notably larger (4-6 times) than those of S2 in the HS timing. Since S2 directly used the heel force data as a parameter, it was not surprising that the trained network including the heel sensor detects the HS more accurately. Compared to S2, the network trained with S1 had to track the HS only with the IMU data. Even though the IMUs at the thigh and torso could detect the HS based on the accelerometer information, they were less reliable compared to a direct detection from the heel force sensor. Still, the temporal error of S1 ($\simeq 30$ ms) may be considered acceptable for the

TABLE II
MEAN AND ONE STANDARD DEVIATION OF THE DIFFERENCE BETWEEN GROUND TRUTH AND PREDICTION AT THE HEEL-STRIKE FOR 12 CONSECUTIVE GAIT CYCLES

Walking speed	4 variables (S1)	5 variables (S2)
0.5 m/s	18.75 ± 10.68 ms	3.75 ± 3.11 ms
1.0 m/s	17.08 ± 10.75 ms	2.08 ± 2.57 ms
1.5 m/s	20.83 ± 13.29 ms	5.00 ± 3.02 ms
2.0 m/s	22.50 ± 8.12 ms	3.75 ± 3.11 ms

gait phase because it was less than a short-latency response time ($\simeq 40$ -50 ms) from the reflex pathways of lower-limb [34]. In addition, the error of gait phase that occurred for 30 ms in HS was a small of $\pm 3\%$.

Fig. 10 describes the estimation of subject's gait phase in two different walking speeds (0.5 and 1.5 m/s) using two polar coordinate datasets, S1 and S2. As shown in Fig. 10(a) and (b), there was an apparent difference between S1 and S2 when the HS occurs. In both walking speeds, S2 had smaller detection errors (5 and 10 ms) at the HS. In the case of S1, relatively greater detection errors (25 and 30 ms) were shown when the HS occurs.

Fig. 10(c) shows the estimation results when the walking speed was continuously changed. Regardless of the walking speed, both configurations (i.e., S1 and S2) showed robust estimation without noticeable differences throughout the entire gait cycle. As we already mentioned, when the walking speed was increased, larger deviations around the HS were found. On the other hand, at the slower walking speed, larger deviations appear at the mid-stance (40-60%).

IV. DISCUSSION

According to the results, the heel force sensor shows its own benefit in finding the accurate HS. However, the heel force sensor is relatively sensitive to the foot placement for obtaining the sensor data in a robust manner. In this paper, the heel sensor input can be robustly given regardless of the foot placement because we put the sensor beneath the healthy subject's compliant foot. However, differing from human, a prosthetic foot is usually made by relatively stiff material (e.g., Acrylonitrile Butadiene Styrene (ABS)) or designed as a rigid body. In this case, if the prosthetic foot has a poor heel-contact, the heel force sensor would have a problem detecting the heel-strike correctly.

It is difficult to directly compare our results with previous studies due to discrepancies in the sensor setups and sensor-specific output range. However, our study has some merits in gait phase estimation compared to previous studies. For constant walking speed, the error of Kang et al. [11] is $4.83 \pm 0.62\%$, whereas our study shows a difference of $1.67 \pm 1.36\%$. Networks from our study and [11] have the same sliding window size. Using MSE (our study) rather than RMSE ([11]) as a loss function and having a deeper layer for LSTM may have reduced the mean error with slight increases in the variability. Vu et al. [14] have, similar to our study, used an RNN with one IMU and two force sensors in a transtibial

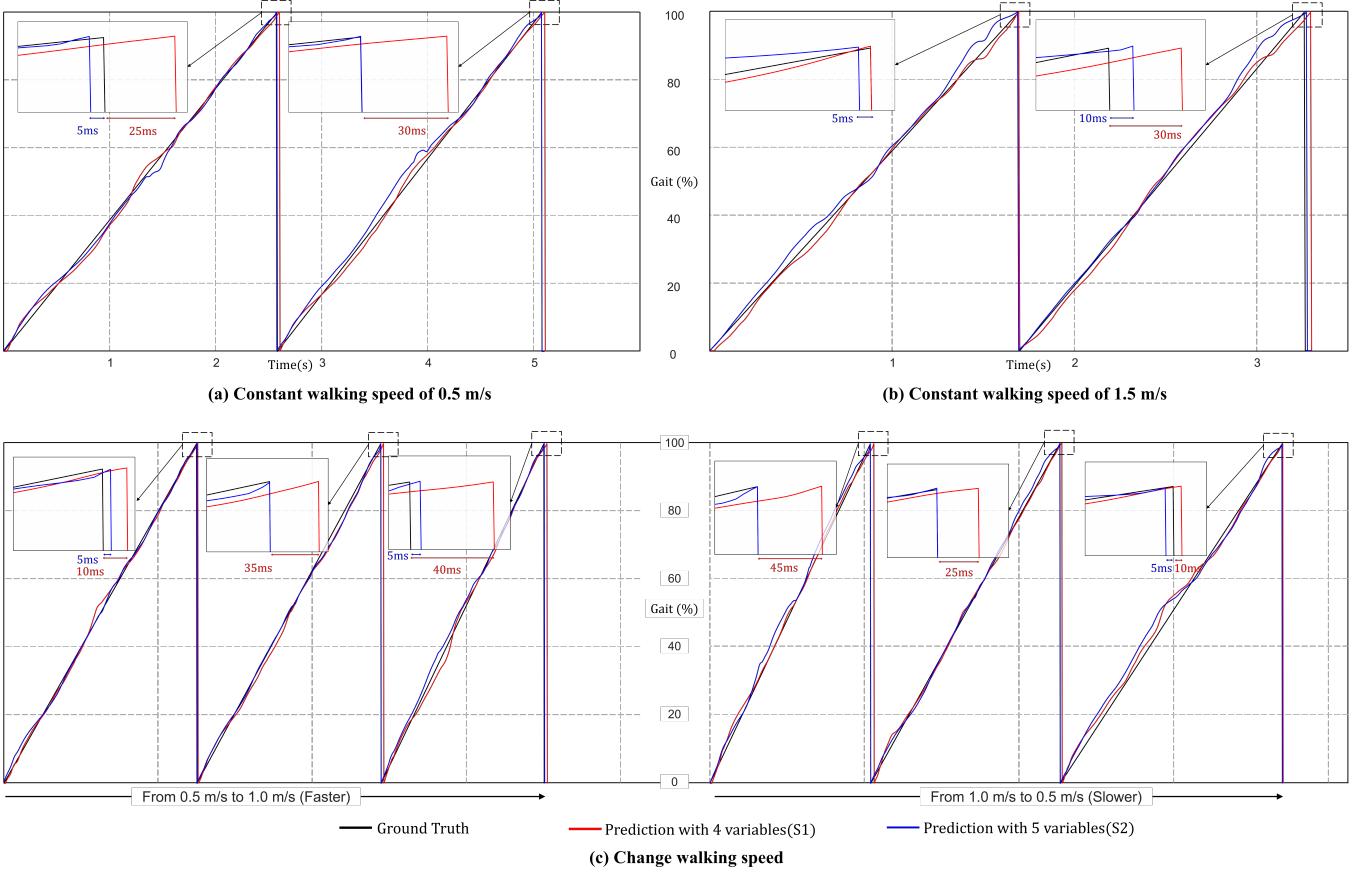


Fig. 10. Robustness of gait phase estimation according to walking speed. (a) and (b) show gait phase estimation at two different constant walking speeds: 0.5 and 1.5 m/s. In particular, the differences between ground truth and two data setups are highlighted around the HS. (c) shows the results in increasing walking speed from 0.5 to 1.0 m/s, and decreasing walking speed from 1.0 to 0.5 m/s.

prosthesis application for gait phase estimation. The estimation error is $2.1 \pm 0.1\%$, which is a slightly larger error than ours but with consistent prediction. We postulate that two force sensors helped stable and consistent prediction, but fewer IMU sensors and shallower network layers caused higher errors in multiple subjects.

There were some limitations in this study. Firstly, the heel force sensor may have a durability issue due to the frequent exposure of large impact at the HS. Secondly, during the data analysis, we found the high-frequency noise in a particular part of the torso data from one subject. We suspect that the torso IMU was slightly vibrated on the subject's clothes during walking because the sensor was not fully tightened. Having the noisy data in our analysis may reduce the reliability of the data and hinder network learning. However, due to the worldwide pandemic COVID-19, further experiments to gather sufficient data are not possible. Also, the actual walking experiment using the prosthesis is restricted. Although we could not conduct an additional experiment for the amputee subjects, we expect that the amputee subjects wearing the robotic transfemoral prosthesis have a similar gait pattern of healthy subjects. This is why we used the proposed sensor sets from the healthy subjects. In the future study, experiments will be conducted with the amputee subjects with the given custom-built powered transfemoral prosthesis, AMPRO II (Fig. 11),

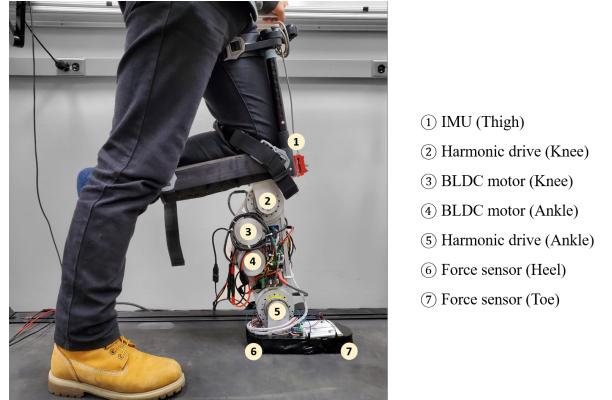


Fig. 11. A custom-built powered transfemoral prosthesis, AMPRO II

to validate the proposed idea. Also, to obtain a more robust estimation, additional network training is planned with the increased number of subjects' datasets. Thirdly, although the LSTM-based algorithm takes less time than the short-latency response time, performance improvement is necessary because it requires a lot of computing power in real-time.

