Estimating Peak Transverse Plane Moment and Whole-Body Angular Momentum for Transtibial Amputees with Python

ME 396P Final Project

Etse Campbell, Lindsey Lewallen, and Avery Pendley

Introduction

Every year, nearly 185,000 persons in the United States are subject to limb amputation, adding to the approximately 1.2 million Americans disabled by some form of limb loss (Ephraim et al., 2005). Among these numbers, transtibial amputations (i.e., below knee), make up for about half of all lower limb amputations (Sun and Voglewede, 2004). In general, lower limb loss greatly effects mobility, specifically the ability to walk.

Turning while walking requires an increased rotation of the body about the transverse plane, compared to that of straight walking (Glaister et al., 2007) For transtibial amputees, this can be challenging due to the nature of many prostheses being stiff and inhibiting transverse plane rotation (Hafner et al., 2002). As a result, shear pressure between the residual limb and prosthetic socket can cause mobility issues and residual limb pain (Ephraim et al., 2005). Furthermore, the ability to turn while walking requires an increase in dynamic balance, more than that required for straight-line walking, that is estimated with whole body angular momentum (Nolasco et al., 2019). Because transtibial amputation leads to a functional loss of ankle muscles, lower-limb amputees have more difficulty regulating dynamic balance (Silverman and Neptune, 2011), ultimately making turning a difficult task. With about 40,000 transtibial amputations a year, creating a functional, yet comfortable, prosthesis is necessary (Sun and Voglewede, 2004).

In an effort to ease the difficulty of turning, a transverse rotation adapter can be attached to a prosthesis, permitting internal and external rotation of the prosthetic socket relative to the prosthetic foot (Lamoureux and Radcliffe, 1977), helping to reduce shear pressure and increase dynamic balance. Although useful in theory, these devices are typically set at one optimal stiffness level and cannot be changed during movement, which does not allow for maximum rotation abilities. Due to the differing dynamics of turning and straight-line walking, the ability to adjust the stiffness of the rotation adapter during walking could potentially increase the mobility of the user.

Being so, a prototype for a variable stiffness transverse place adapter, called VSTA II, was recently designed for a lower limb prosthesis (Pew and Klute, 2017). The VSTA II has five springs in a parallel that can be adjusted while the user is in motion, allowing for the adapter to alternate between stiffness levels with the goal of easing walking, specifically the ability to turn. The ability to adjust the stiffness levels during motion theoretically allows for a greater reduction in shear pressure and change in dynamic balance than a single stiffness adapter would.

The scope of this project is as follows: to predict peak transverse plane moment (PTPM) and peak-to-peak whole body angular momentum (WBAM) with varying stiffness levels of the VSTA II. To complete the desired goals, two supervised machine learning algorithms were created. These algorithms predicted four measures (external PTPM of the transverse rotation adapter, internal transverse PTPM of the transverse rotation adapter, WBAM in the coronal plane, and WBAM in the transverse plane).

Methodology & Results

Dataset preparation

We assembled data for whole body angular momentum and transverse plane moment utilizing a 3×3×3 factorial experiment, the experimental design is summarized in Table 1. Data were collected from 10 individuals with a unilateral (one-side) transtibial (below-knee) amputation wearing the VSTA II. Amputees performed three activities: walking straight (ST), turning with their prosthesis on the inside limb (IN), and turning with their prosthesis on the outside limb (OUT). They walked at three different speeds: slow (SLW), comfortable (SSW), and fast (FSW). The stiffness was set at three different levels: 0.25 Nm/º (Stiffness A), 0.75 Nm/º (Stiffness B), and 1.25 Nm/º (Stiffness C).

Table 1. Experimental design for WBAM and PTPM data collection

Factors	***Levels***		
	Low	***Medium***	***High***
Activity	Straight walking (ST)	Inside turn (IN)	Outside turn (OUT)
Speed	Slow (SLW)	Moderate (SSW)	Fast (FSW)
Stiffness	0.25 Nm/° (A)	0.75 Nm/° (B)	1.25 Nm/° (C)

Experimental data were used calculate four measures for each step taken: external PTPM of the transverse rotation adapter, internal transverse PTPM of the transverse rotation adapter, WBAM in the coronal plane, and WBAM in the transverse plane. These measures were calculated for each step, therefore, there are multiple steps and multiple values for each condition.

Visualizing Data

We used the **matplotlib.pyplot** library to visualize PTPM and WBAM data (Figure 1 and Figure 2). These graphs show all of the inputs plotted against our outputs. Each subject is shown as a different color. The jitter function was used spread out data points to better visualize them in the figures.

Generally, it's difficult to see trends in these plots. We have several variables that seem to be affecting our outputs. The machine learning algorithms can help us capture trends in this data.

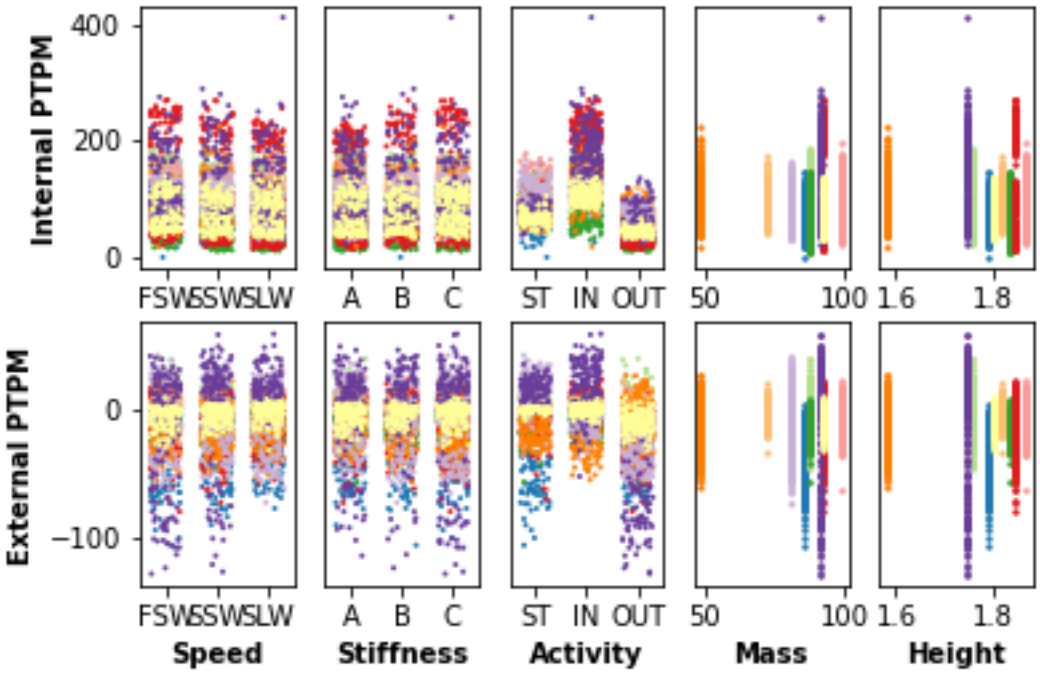

Figure 1. Effect of Speed, Stiffness, Activity, Mass, and Height on Internal and External PTPM

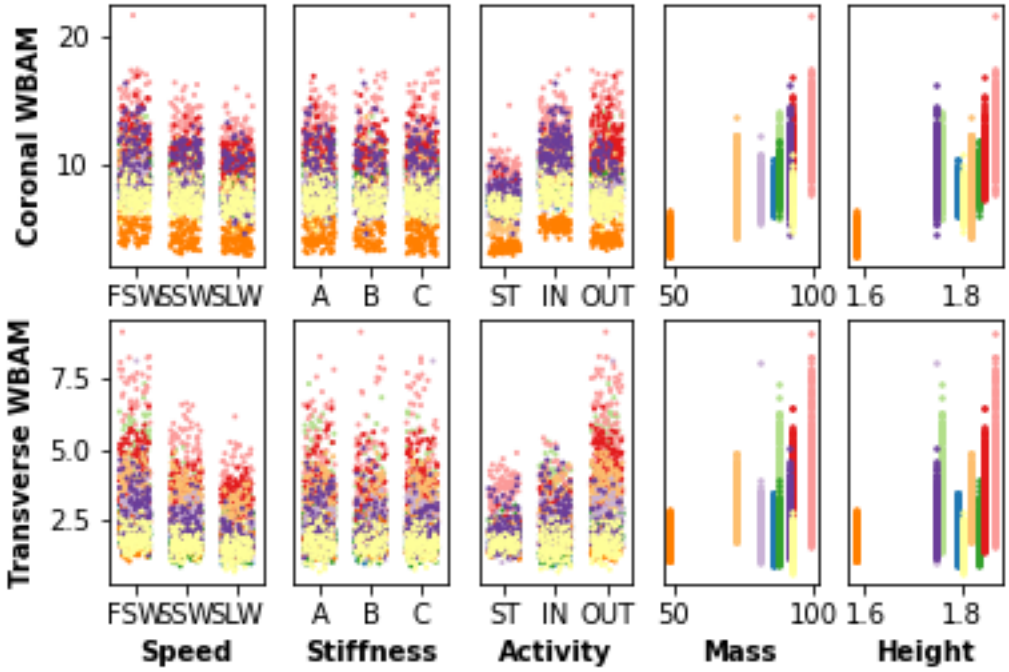

Figure 2. Effect of Speed, Stiffness, Activity, Mass, and Height on Coronal and Transverse Plane WBAM

Data preprocessing
We used **Pandas** library to check for normality, organize the dataset by the subject number, and convert the dataset into a DataFrame object.

Machine Learning Algorithms
Experimental data were used to create four linear mixed effects models and four linear models that predicted one of the four outputs (i.e., external/internal PTPM of the transverse rotation adapter, coronal/transverse WBAM). The syntax for the linear model (`lme`) in Python is as follows:

```
f1 = 'Output~mass+height+OUT+ST+SLW+SSW+B+C'
lm = sm.ols(formula=f1, data=dataset).fit()
```

Where `dataset` is the text file containing the dataset after it has been cleaned and normalized. To create the basic linear model, we specified a formula and a dataset using **statsmodels.formula.api** as it supports pandas data frames. In the formula, we used five inputs (i.e., height, mass, activity type: `OUT+ST`, transverse rotation adapter stiffness: `B+C`, and speed: `SLW+SSW`) as model predictors. Activity, speed, and stiffness are categorical variables (i.e., they do not have a numerical value), therefore, we created dummy variables for these inputs. Our output is one of the four outputs we mentioned previously.

The syntax for the linear mixed effects (`lme`) model in Python is as follows:

```
f2 = 'Output~OUT+ST+SLW+SSW+B+C+mass*height'
lme = sm.mixedlm(formula=f2,data=dataset_JL,groups=dataset_JL['sub'],re_formula='~SLW+SSW').fit()
```

A linear mixed effects model is an extension of the linear model because we can specify the inputs/outputs like above while also accounting for fixed and random effects that could influence our data. Fixed effects are variables that we expect to have an effect on the response variable (e.g., Coronal WBAM). Because we are interested in determining how height, mass, activity (`OUT+ST`), speed (`SLW+SSW`), and stiffness (`B+C`) affect WBAM and PTPM, these are labeled as the fixed effects. Because height and weight may be confounding variables (i.e., they could be correlated), we added an interaction term between these variables to the model (i.e., mass*height). Random effects are the grouping factors. In this case, data were grouped by the subject number (labeled as factors 1-10), classified as a random effect (i.e., groups = dataset['sub']. It is also important to consider that speed (labeled just as fast, slow, and comfortable), will vary among subjects. Therefore, we added speed as a random slope parameter (i.e., `re_formula='~SLW+SSW'`).

Algorithm evaluation and testing

Using **KFold** from **sklearn**, *the ten-fold cross validation error* was computed for each of the eight models to compare the models. Ten-fold cross validation error is commonly used in machine learning to compare models with different complexity. We split observations into ten partitions then trained the model using nine of them (Figure 3). We predicted the algorithm error using the one remaining partition. This was repeated ten times, holding out a different partition each time. The result was averaged to get a final error.

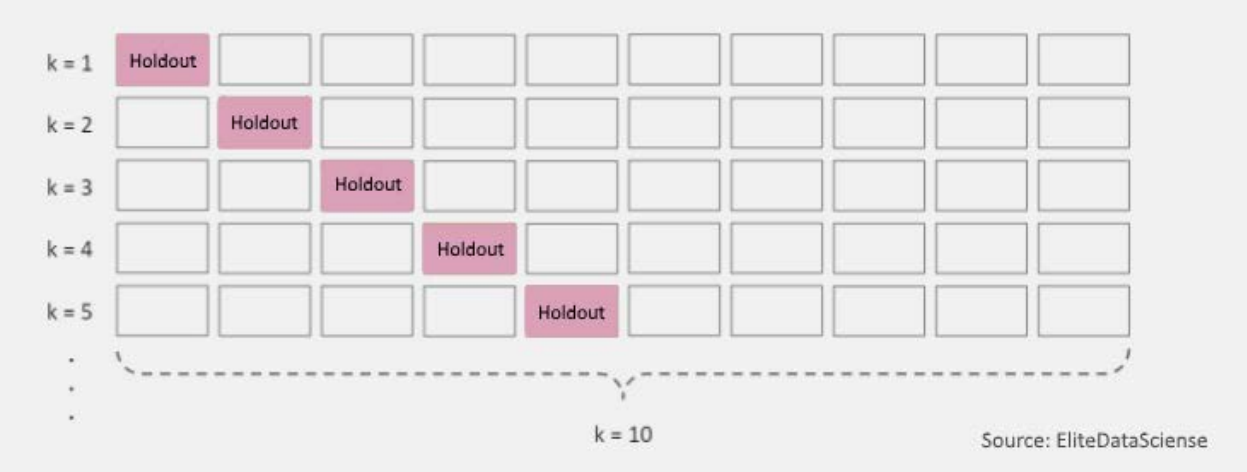

Figure 3. Ten-fold cross validation error process

Results from ten-fold cross validation error are shown in Table 2.

Table 2. Results from ten-fold cross validation error

Model	***Coronal WBAM (kg-m²/s²)***	***Transverse WBAM (kg-m²/s²)***	***Internal PTPM (Nmm/kg)***	***External PTPM (Nmm/kg)***
Linear	1.50	0.94	34.49	19.65
Linear Mixed Effects	1.48	0.97	34.76	19.89

Both models produced reasonable errors for the range of our data. Our linear mixed effects model, the more complex algorithm, but we did not see a significant reduction in error compared to the linear model.

Representative Subject

Finally, we used the linear model to predict output measures for a representative subject with a mass of 80 kg and height of 1.7m. This subject walked straight at a self-selected pace with stiffness B. Figure 4 shows the output from Python when we input values for our representative subject.

```
In [1]: runfile('C:/Users/lkl444/Desktop/Python/Final Project/Final_Code.py',
wdir='C:/Users/lkl444/Desktop/Python/Final Project')
The predicted Coronal Plane WBAM is 6.33.
The predicted Transverse Plane WBAM is 1.75.
The predicted Internal PTPM is 87.49.
The predicted External PTPM is -5.3.
```

Figure 4. Predicted Coronal WBAM, transverse plane WBAM, internal PTPM and external PTPM using representative subject

Summary and Conclusions

In conclusion, we were successful at predicting Whole Body Angular Momentum and Peak Transverse Plane Moment. In the future, we hope to explore more machine learning algorithms and create an optimization algorithm that will choose an optimal stiffness based on a set of inputs. This could be used as a tool for clinicians to help improve selection of stiffness values for the VSTA II.

Several packages were used to develop the machine learning program for this project. The Pandas package helped us clean and organize our data, Statsmodels created our two machine learning algorithms, and Sklearn helped evaluate the algorithms developed for this program. Additionally, NumPy was utilized to process numerical data. Finally, we used two packages to visualize data: Matplotlib and Seaborn. Matplotlib helped us create the basic figures to visualize data, while Seaborn helped us better visualize our categorical data

References

Ephraim, PL, Wegener, ST, MacKenzie, EJ, Dillingham, TR, Pezzin, LE, 2005. Phantom pain, residual limb pain, and back pain in amputees: Results of a national survey. Arch. Phys. Med. Rehabil.

Sun, Jinming, and Philip A Voglewede. “Powered Transtibial Prosthetic Device Control System Design, Implementation, and Bench Testing.” *Journal of medical devices* 8.1 (2014): n. pag.Web.

Glaister, BC, Bernatz, GC, Klute, GK, Orendurff, MS, 2007. Video task analysis of turning during activities of daily living. Gait Posture.

Hafner, BJ, Sanders, JE, Czerniecki, JM, Fergason, J, 2002. Transtibial energy-storage-and-return prosthetic devices: A review of energy concepts and a proposed nomenclature 39, 1–11.

Nolasco, LA, Silverman, AK, Gates, DH, 2019. Whole-body and segment angular momentum during 90-degree turns. Gait Posture 70, 12–19.

Silverman, AK, Neptune, RR, 2011. Differences in whole-body angular momentum between below-knee amputees and non-amputees across walking speeds. J. Biomech. 44, 379–385.

Lamoureux, LW, Radcliffe, CW, 1977. Functional analysis of the UC-BL shank axial rotation device. Prosthet. Orthot. Int. 1, 114–118.

Pew, C, Klute, GK, 2017. Second generation prototype of a variable stiffness transverse plane adapter for a lower limb prosthesis. Med. Eng. Phys. 49, 22–27.