

**Neural Engineering Laboratory, George Mason University,
Microelectronics Laboratory, Thomas Jefferson High School for Science and Technology
2019-2020 Mentorship Second Semester Paper
Comparing Muscle Spindle Afferent Models
Stephanie Hachem
shachenc027@gmail.com**

Abstract

Proprioception, the internal sense of where your body parts are relative to each other, is essential for many, particularly bimanual, daily activities. Unfortunately, because modern prosthetics lacks this sense it is often difficult or impossible to perform hand-eye coordination tasks with them, and thus upper extremity prosthetics can become a nuisance or burden amputees frequently abandon at home (Biddiss & Chau, 2007). Towards creating a naturally-functional prosthetic able to provide proprioception, this project aims to compare computational models of muscle spindle afferents, using experimental data, in hopes that the best computational model could later be used to predict what voltages should be provided to which afferent nerves in a residual limb.

Two muscle spindle models were compared using experimentally-measured afferent and muscle length data digitally extracted from figures in 10 articles. Data was from cats and humans. Both models implement the same formulas in either MATLAB or Python, take muscle length as an input, and can provide primary and/or secondary afferent output. Comparing the experimentally-measured and the predicted afferents, the more recent, Python model was found to provide more accurate afferent output.

Keywords: muscle spindle afferents; muscle spindle model; proprioception; proprioceptor; Neural Simulation Tool

1. INTRODUCTION

1.1. Muscle spindles

Muscle spindles are sacks of sensitive sensory fibers situated inside capsules parallel to muscle fibers in a muscle belly, an arrangement allowing the spindles to be stretched in tandem with the muscle. Each spindle holds several such sensitive fibers, in three types: nuclear chain fibers which consistently monitor muscle length, static nuclear bag fibers also consistently monitoring length, and dynamic nuclear bag fibers primarily monitoring velocity of change in length, and secondarily consistently monitoring length. Sensory information encoded by those sensitive fibers is transmitted by two main types of nerves: primary (Ia) afferents, where afferent here refers to nerves sending information towards the spinal cord and brain, wrapped around all three fiber types to output length and velocity; and secondary (II) afferents wrapped around the nuclear chain and static nuclear bag fibers to output length alone. Thus, when a muscle is stationary, both the Ia and II afferents are sending signals encoding the muscle's length, while when a muscle is in the process of stretching, Ia output increases drastically to indicate the occurrence and velocity of the stretch while II output increases proportional to the actual stretch.

1.2. Microneurography

Microneurography is the safe and painless technique of inserting electrode-tipped needles into muscles to monitor nerve activity, including proprioceptor activity, while actual muscle activity is typically recorded through surface EMG, the technique of placing electrodes on skin to monitor muscle contraction.

1.3. Literature review

1.3.1. Recordings of human muscle spindle afferents using microneurography

Jones, Wessberg, and Vallbo (2001) recorded and analyzed microneurographic recordings of proprioceptor response in the human wrist to wrist movements of volunteers in order to determine if muscle length and bodily movement direction is encoded in afferent output, as was demonstrated. This study revealed the more accurate velocity output of spindles in muscles which constantly counteract gravity, while spindles in muscles with a smaller role in counteracting gravity were shown to be equally sensitive to velocity and stretch in accordance with the roles of Ia and II afferents. It was concluded that despite the small amount of data gathered, wrist joint position could be predicted well from spindle output.

1.3.2. An accurate muscle spindle model

From the considerable consistency of resulting spindle afferent data across several studies similar to the above, the question of whether or not afferent activity is predictable arose. In response, several studies containing mathematical models describing spindle input based on movement and excitation of the muscle the spindle is situated in were published. In particular, the muscle spindle model of Mileusnic, Brown, Lan, and Loeb (2006) has yielded remarkable accuracy in predicting spindle input based on muscle dynamics. The model was created to closely resemble the physiological elements of spindles, and was validated by experimental data for ability to reproduce experimentally-observed spindle characteristics and behavior under numerous conditions. Today, Mileusnic et al.'s (2006) model remains one of the more widely used and accurate spindle afferent output prediction models.

1.3.3. Cat muscle spindle data and models can apply to human data

To compare the predictive power in this rising number of spindle input algorithms, Malik, Jabakhanji, and Jones (2015) analyzed those models with the MATLAB and Simulink softwares. Specifically, they juxtaposed the ability of 6 algorithms from the literature to predict actual spindle input results based on kinematics for a set of human arm data from a past study (Jones, Wessberg & Vallbo, 2001). The three algorithms which predicted results best were then used with wrist kinematic data generated through simulating random rotations around the wrist joint axes with a tendon displacement model also from the literature, and the study concluded that muscle spindle models originally made with data from cat hindlegs have predictive power for human muscle spindles.

1.3.4. An open-source muscle spindle model

Vannucci, Falotico, and Laschi (2017) programmed a muscle spindle model accessible through the Neural Simulation Tool (Gewaltig, Morrison, & Plessner, 2012). The spindle model's algorithms are based off of simplifications, which attempt to minimize loss in prediction accuracy, of algorithms presented by Mileusnic et al. (2006). The spindle model was verified through and applied in robotic and biological applications, is open-source (Vannucci, Falotico, & Laschi, 2018), and will be used in this study.

1.3.5. An open-source dataset of muscle spindle afferents

Blum, D'Incamps, Zytynski, and Ting (2017) measured muscle spindle afferent output from cat hindlegs during a variety of muscle stretch kinematic conditions. Results suggested that muscle spindle afferent output is a function of whole-muscle tendon force, not muscle length as previous models have

assumed. Muscle spindle afferent output files are open-source (Blum, Lamotte D'Incamps, Zytnicki, & Ting, 2018) and will be used in this study.

2. METHODS

Note that from this point on, the word "afferent(s)" is always used to refer to muscle spindle afferent firing rate, and OFL refers to optimal fiber length.

2.1. Required software

Install MATLAB 2017b version 9.3.0 or above on a Windows 64-bit platform, and the Neural Simulation Tool and dependencies (Python 3) on a Oracle Linux 7 Red Hat Enterprise operating system (this study used Anaconda 3 and Oracle VM VirtualBox version 5.2.12). Install the muscle spindle module (Vannucci, Falotico, & Laschi, 2018) on the VM. Install PlotDigitizer version 2.6.8 (Huwaldt & Steinhorst, 2015) on Windows.

2.2. Data sources

Eleven sources of data were simulated with afferent models. Data measured by Blum et al. (2018) was obtained from supplemental files (.m) provided with Blum et al.'s (2017) article, and all other data was obtained from digitizing figures from 10 articles in the literature (Dimitriou, 2016; Matthews, 1962; Marasco, Bourbeau, Shell, Granja-Vazquez, & Ina, 2017; Matthews, 1963; Blum, D'Incamps, Zytnicki & Ting, 2017; Prochazka & Gorassini, 1998; Matthews & Stein, 1969; Day, Bent, Birznieks, Macefield, & Cresswell, 2017; Nichols & Cope, 2004; Edin & Vallbo, 1990). See Table 1 for metadata of each data source.

2.3. Blum et al.'s data

Blum et al.'s (2018) data was ran with 2 afferent models: Schneider's (2013) model, and Vannucci et al.'s model.

2.3.1. Blum et al.'s data: Schneider's model

Muscle lengths of Blum et al.'s data were converted to a decimal of that muscle's OFL and written to files which were fed to Schneider's Simulink model in MATLAB. Blum et al.'s data also provided the time of every afferent action potential; this data was converted to firing rates (to be compared with Vannucci et al.'s model output of firing rates) through averaging the number of action potentials within a 0.0005-second timestep.

2.3.2. Blum et al.'s data: Vannucci et al.'s model

Muscle lengths of Blum et al.'s data, with a 0.0005-second timestep, were converted to decimals of OFL and written to files (.txt) which were fed to Vannucci et al.'s model. Vannucci et al.'s model requires the NEST software to run, which requires Linux, so the Vannucci et al.'s model was used on a Linux VM (see the Required Software section), and files were shared by shared folders using VirtualBox Guest Additions. As in Table 1, Blum et al.'s data was recorded only from Ia fibers, so only Ia fiber types were used to simulate the output, and predicted afferents were recorded with a multimeter and spike detector, and written to file (.dat). The NEST kernel was reset every time a new simulation was run in order to possibly speed up simulations, and a population size of 1 Ia/II fiber was used to reduce simulation time and memory needed for output files.

2.4. Digitized article data

Experimentally-measured muscle lengths and afferents data was obtained from manually and automatically digitizing figures from the literature.

2.4.1. Digitized article data: Manual digitization

The software PlotDigitizer version 2.6.8 (Huwaldt & Steinhorst, 2015) was used in Windows. A screenshot of each graph, as zoomed up as possible while still remaining completely on the computer screen, was taken using the software Snipping Tool; the width or height, of width/height reference lines (sometimes axes, sometimes a separate line on the side of the graph) whose unit and width/height was known, was measured in pixels with the software Paint, and used to calculate the length/height in units (not pixels) of the x and y axes (if the reference width/height wasn't the axes). Units were always time (s) for the x-axis, and either afferents (impulses/s, Hz), muscle length (m), or muscle length displacement (m), for the y-axis. These length/heights in units were used to calibrate PlotDigitizer, with which users open a file, calibrate axes, and click on each point of interest in the graph; points are then saved to file (.csv). See Appendix A for a detailed explanation of how to manually digitize.

2.4.2. Digitized article data: Automatic digitization

Because manual digitization is time-consuming, Open Source Computer Vision Library (openCV) (Bradski, 2000) and Python 3 were used with the Spyder IDE (Raybaut & Cordoba, 2009) to automatically digitize files. OpenCV converted images to grayscale and recorded the xy coordinates of each pixel whose grayscale value was greater than some threshold; thresholds were adjusted for some figures. The length between the left-most x-coordinate and the image's left edge was subtracted from all x coordinates to align the data to the starting time. Length/heights (in pixels and in units) of reference lines were used to calculate scale factors, and scale factors were used to convert aligned coordinates into units, and these converted and aligned coordinates were written to files (.txt). See Appendix A for a detailed explanation of how to automatically digitize.

2.4.3. Digitized article data: Schneider's model

Muscle lengths (and the corresponding time) of digitized article data were converted to a decimal of that muscle's OFL and written to files (along with the corresponding time) which were fed to to Schneider's Simulink model in MATLAB.

2.4.4. Digitized article data: Vannucci et al.'s model

Muscle lengths were converted to decimal of OFL and written to files (.txt) which were fed to to Vannucci et al.'s model. Since there isn't data at constant timesteps in many of the digitized graphs, the NEST Simulate function's timestep argument was calculated for each timestep. Ia or II fibers (both with a population of 1 fiber) were used for each graph depending on which type of fiber (Ia or II) was experimentally measured in the article. Optimal fiber lengths (OFL), which were assumed to be a muscle's resting length, were taken from the article, from other articles, or was estimated (see next paragraph); decimal of OFL were then input to Vannucci et al.'s model. As when simulating Blum et al.'s data, Vannucci et al.'s model was used on a Linux VM, shared folders were used, predicted afferents were recorded with a multimeter and spike detector and written to file (.dat), and the NEST kernel was reset every time a new simulation was run.

2.4.5. Digitized article data: Estimating OFL

The OFLs of muscles whose muscle lengths and afferents were measured by Marasco et al. (2017), Prochazka & Gorassini (1998), and Edin & Vallbo (1990) couldn't be found in the article or literature, so the length was estimated: for graphs which didn't provide units, the limits (maximum and minimum y coordinates) of the muscle length graph were assumed to be the resting length (that is, the OFL) and the muscle's limit of stretch, respectively. The muscle was assumed to have a range of stretch that is from 100% of the OFL to 108% of the OFL, since these are the limits of the Vannucci et al.'s model's input OFLs in that model's example file (.py) and those are the limits of that model's input OFLs before the simulation is killed by the Linux Out of Memory killer. Therefore, the muscle's OFL was calculated given that this range of stretch (limit of stretch y coordinate - OFL y coordinate, where OFL y coordinate is 0) is 0.08 of the muscle's OFL.

3. RESULTS

3.1. Color legend

For Figures 1, 2, 3, 4, 10, and 12, the experimentally-measured OFL are yellow, the experimentally-measured afferents are red, Vannucci et al.'s predicted afferents are blue, Schneider's predicted afferents are green, and (only in the plots of Blum et al.'s data) the averages of Blum et. al's experimentally-measured afferents and Schneider's predicted afferents are orange.

3.2. General remarks

For Blum et al.'s data, experimentally-measured afferents are generally in horizontal lines, without curves, are far above Schneider's predicted afferents, and aligned with or slightly above Vannucci et al.'s predicted afferents. For digitized article data, experimentally-measured afferents are far above Schneider's predicted afferents, and generally aligned with or below Vannucci et al.'s predicted afferents. Examples of graphs of Blum et al.'s data are in Figure 1, and examples of graphs of digitized article data are in Figure 2.

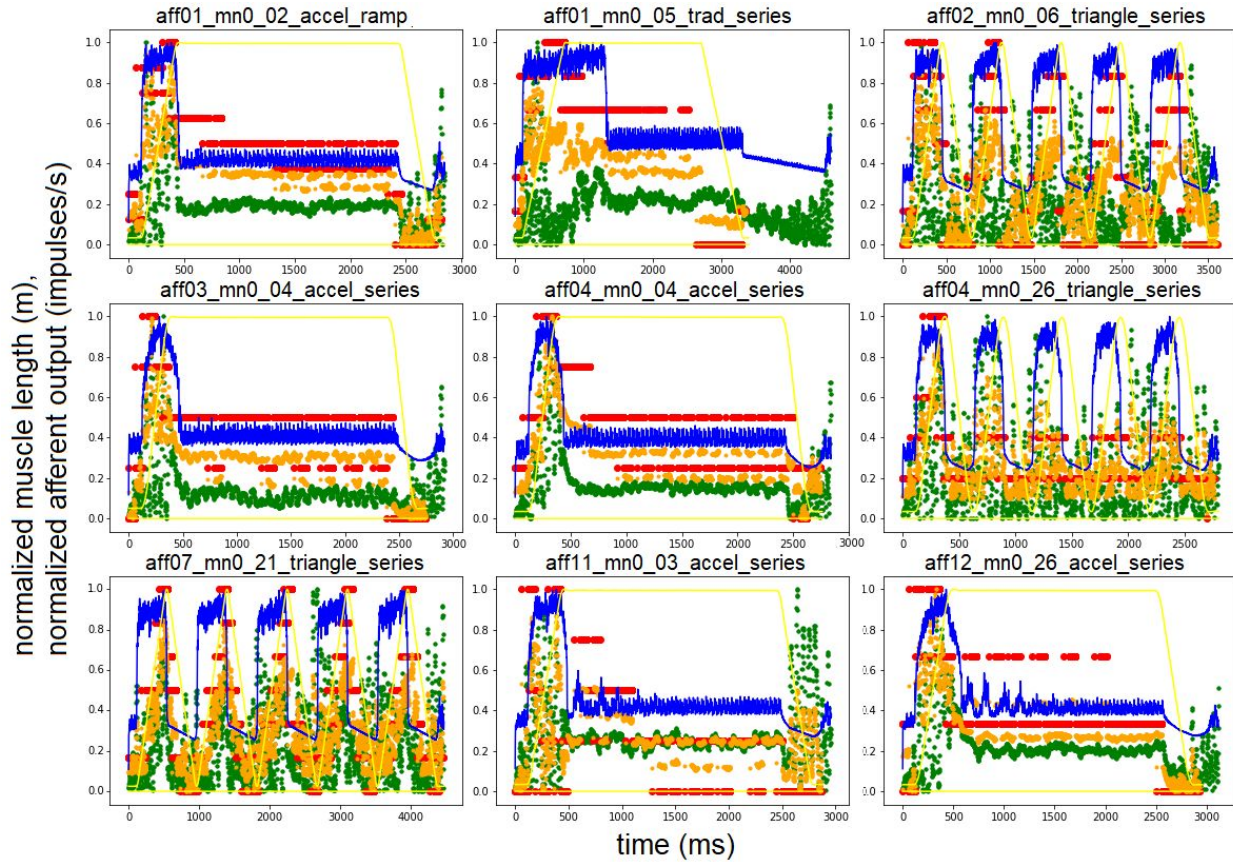


Figure 1. Examples of graphs of Blum et al.'s data. The experimentally-measured OFL (m/m; unitless) are yellow, the experimentally-measured afferents ((impulses/s)/(impulses/s); unitless) are red, Vannucci et al.'s predicted afferents ((impulses/s)/(impulses/s); unitless) are blue, Schneider's predicted afferents ((impulses/s)/(impulses/s); unitless) are green, and (only in the plots of Blum et al.'s data) the averages of Blum et al.'s experimentally-measured afferents ((impulses/s)/(impulses/s); unitless) and Schneider's predicted afferents ((impulses/s)/(impulses/s); unitless) are orange. All y-values are normalized and therefore unitless.

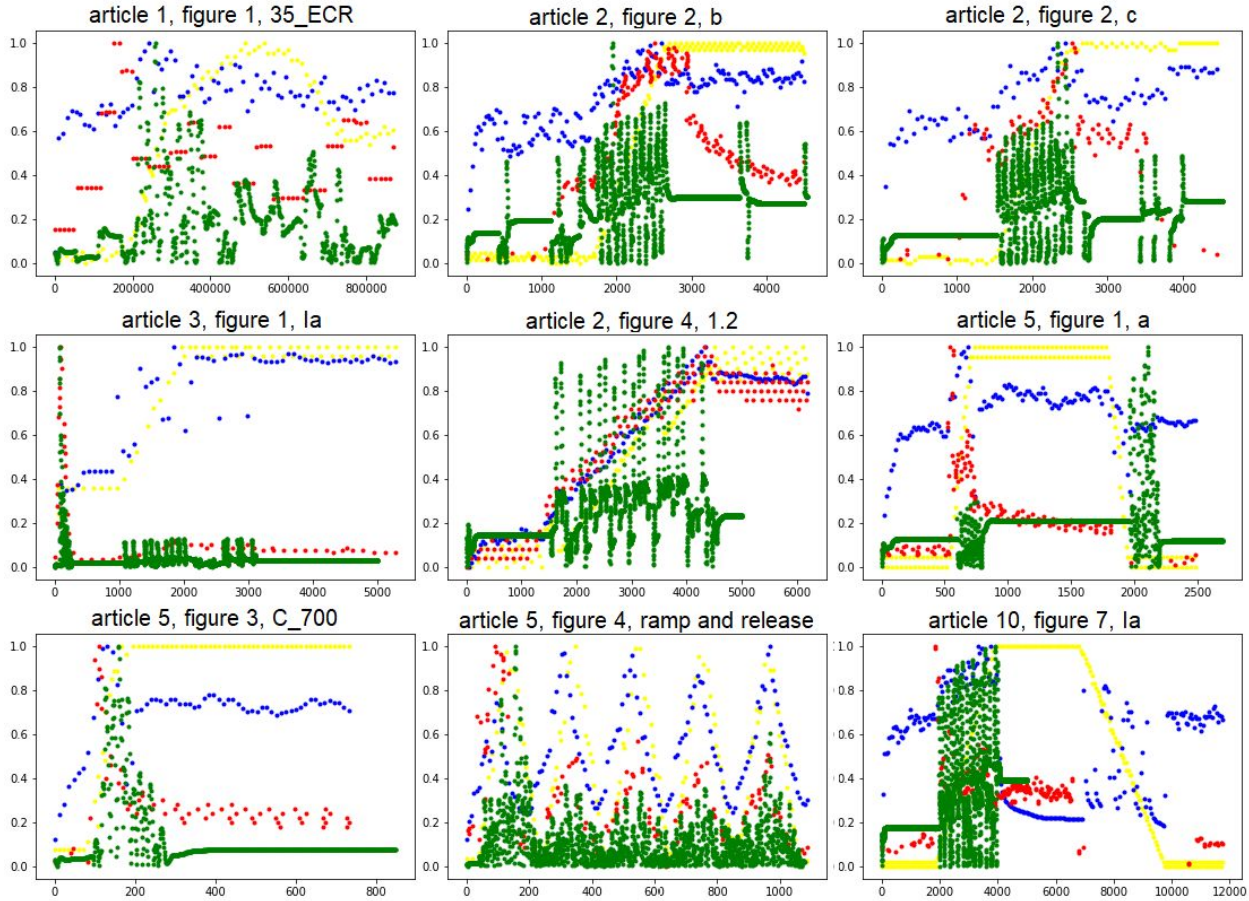


Figure 2. Examples of graphs of digitized article data. The experimentally-measured OFL (m/m; unitless) are yellow, the experimentally-measured afferents ((impulses/s)/(impulses/s); unitless) are red, Vannucci et al.'s predicted afferents ((impulses/s)/(impulses/s); unitless) are blue, Schneider's predicted afferents ((impulses/s)/(impulses/s); unitless) are green.

3.3. Using both models for more accurate predictions of experimentally-measured afferents for Blum et al.'s data

In graphs of Blum et al.'s data, it was observed that Vannucci et al.'s predicted afferents sometimes fell between the experimentally-measured afferents and Schneider's predicted afferents, and thus perhaps the average of the experimentally-measured afferents and Schneider's predicted afferents would be a good approximation of Vannucci et al.'s predicted afferents. Then,

("~=" means "approximately equals")

Vannucci et al.'s predicted afferents \sim average of Schneider's predicted afferents and experimentally-measured afferents

$= (\text{Schneider's predicted afferents} + \text{experimentally-measured afferents}) * .5$

$2 * \text{Vannucci et al.'s predicted afferents} \sim \text{Schneider's predicted afferents} + \text{experimentally-measured afferents}$

$2 * \text{Vannucci et al.'s predicted afferents} - \text{Schneider's predicted afferents} \sim$

experimentally-measured
afferents

In this way, Vannucci et al.'s predicted afferents and Schneider's predicted afferents could be used to approximate experimentally-measured afferents. However, the average of Schneider's predicted afferents and experimentally-measured afferents wasn't a good approximation of Vannucci et al.'s predicted afferents, and thus those graphs weren't included here.

3.4. RMS error

3.4.1. Average RMS error with Blum et al.'s data

The average root mean squared (RMS) error (across all Blum et al. data graphs) by which Vannucci et al.'s predicted afferents fail to match the experimentally-measured afferents is 57.844 impulses/(s*points). The average RMS error (across all Blum et al. data graphs) by which Schneider's predicted afferents fail to match the experimentally-measured afferents is 86.120 impulses/(s*points). The average RMS error (across all Blum et al. data graphs) by which Blum et al.'s experimentally-measured afferents and Schneider's predicted afferents fail to match Vannucci et al.'s predicted afferents in Blum et al.'s data is 52.687 impulses/(s*points). This error is less than the former two RMS errors (57.844 impulses/(s*points) and 86.120 impulses/(s*points)), and this error should be the same as the error between 2*Vannucci et al.'s predicted afferents - Schneider's predicted afferents and experimentally-measured afferents, so this error being lower than the former two RMS errors shows that, for Blum et al.'s data, using Vannucci et al.'s predicted afferents and Schneider's predicted afferents together predicts experimentally-measured afferents better than using either alone. See Table 2 for RMS errors for all Blum data graphs, and see Figure 3 for the Blum data graphs with the minimum/maximum RMS errors for Vannucci's model, Schneider's model, and the average of the experimentally-measured afferents and Schneider's models' predictions..

3.4.2. Average RMS error with digitized article data

The average RMS error (across all digitized article data graphs) by which Vannucci et al.'s predicted afferents fail to match the experimentally-measured afferents is 32.941 impulses/(s*points). The average RMS error (across all digitized article data graphs) by which Schneider's predicted afferents fail to match the experimentally-measured afferents is 33.205 impulses/(s*points). See Table 3 for RMS errors for all digitized article graphs, and see Figure 4 for the digitized data article graphs with the minimum/maximum RMS errors for Vannucci and Schneider's models.

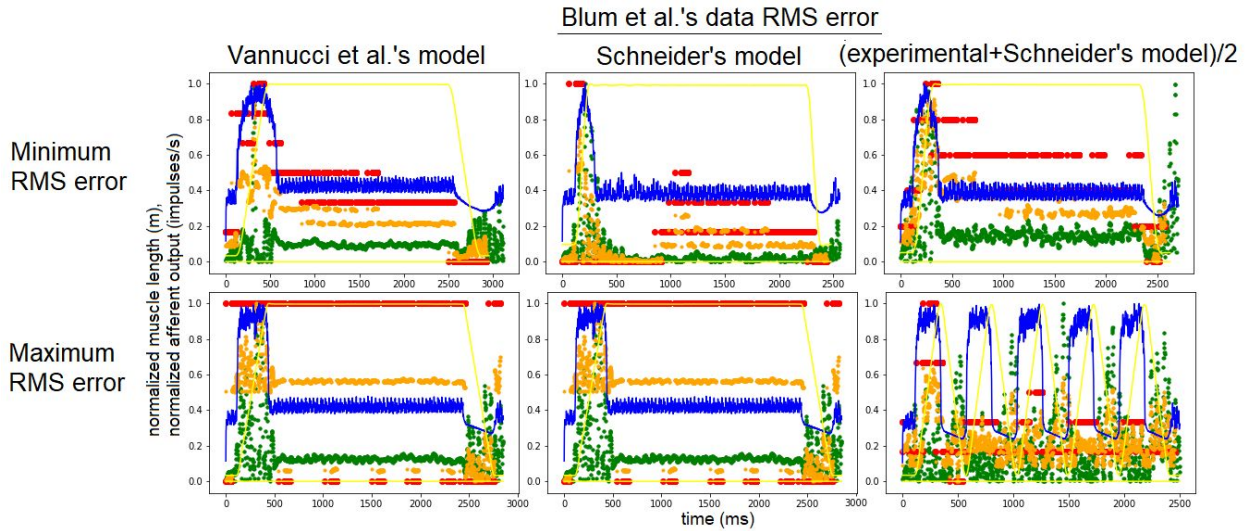


Figure 3. For Blum et al.'s data, the graphs with the minimum and maximum RMS errors by which Vannucci et al.'s predicted afferents fail to match the experimentally-measured afferents, the graphs with the minimum and maximum RMS errors by which Schneider's predicted afferents fail to match the experimentally-measured afferents, and the graphs with the minimum and maximum RMS errors by which Blum et. al's experimentally-measured afferents and Schneider's predicted afferents fail to match Vannucci et al.'s predicted afferents.

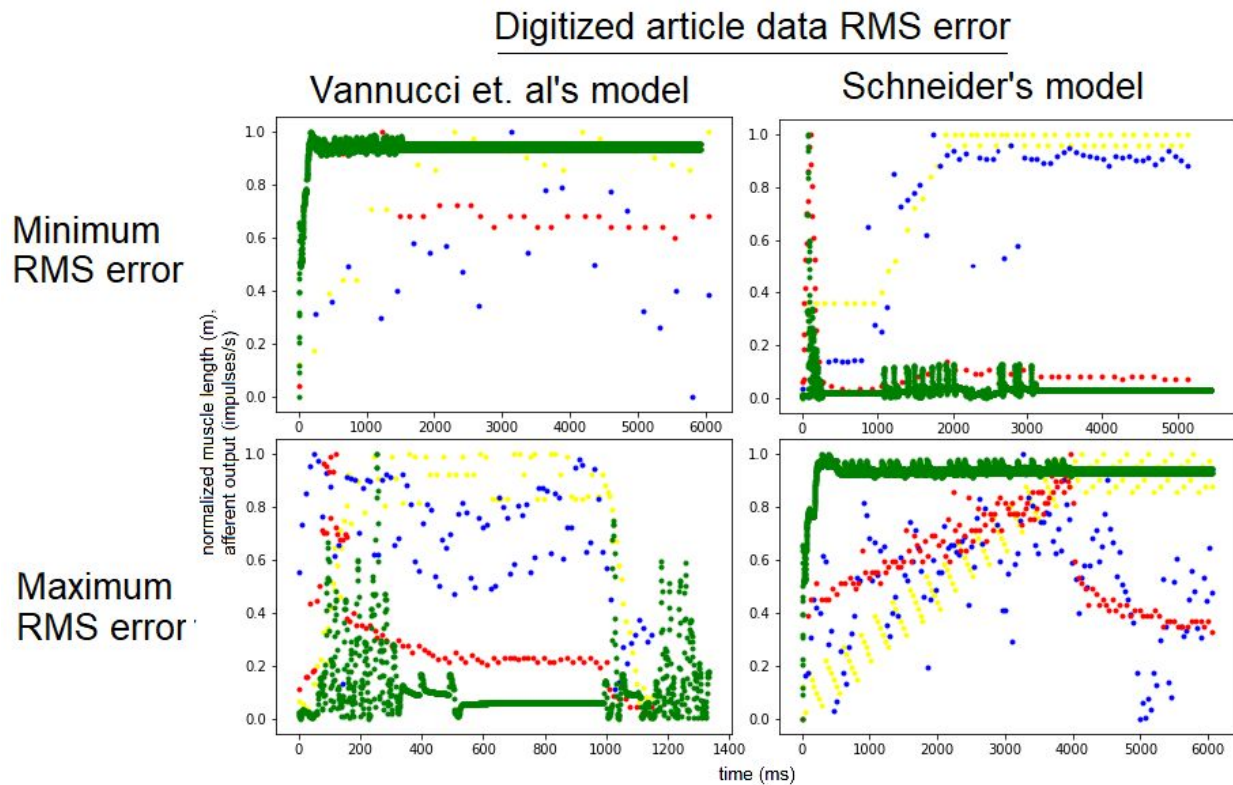


Figure 4. For digitized article data, the graphs with the minimum and maximum RMS errors by which Vannucci et al.'s predicted afferents fail to match the experimentally-measured afferents, and the

graphs with the minimum and maximum RMS error by which Schneider's predicted afferents fail to match the experimentally-measured afferents.

4. DISCUSSION

4.1. Similarities in average RMS error between Blum et al.'s data and digitized article data

With both Blum et al.'s data and digitized article data, the average RMS error by which Vannucci et al.'s predicted afferents fail is more than the average RMS error by which Schneider's predicted afferents fail. While the shape of Schneider's predicted afferents is often similar to experimentally-measured afferents, the height is typically off, which can cause a greater RMS error than having shape be off. The shape of Vannucci et al.'s predicted afferents is similar to experimentally-measured afferents less often than Schneider's predicted afferents is similar to experimentally-measured afferents, but the height of Vannucci et al.'s predicted afferents is often more close to that of experimentally-measured afferents; this is likely the cause of the lower RMS error of Vannucci et al.'s data.

4.2. Differences in average RMS error between Blum et al.'s data and digitized article data

The average RMS error by which both models fail with digitized article data is less than the average RMS error by which both models fail with Blum et al.'s data. This could be because the experimentally-measured muscle lengths and afferents of some digitized article data graphs are in humans or rats (see Table 1), while the experimentally-measured muscle lengths and afferents of Blum et al.'s data are in cats. However, Malik, Jabakhanji, and Jones (2015) demonstrated that muscle spindle models originally made based on cat experiments have strong predictive value for modeling human muscle spindle afferents. Moreover, of the 10 articles from which data was digitized, 6 have data from measurements taken in cats.

5. CONCLUSION

I obtained experimentally-measured muscle lengths and corresponding experimentally-measured muscle spindle afferents from online article supplemental information or from digitizing data in graphs of online articles, and compared experimentally-measured muscle spindle afferents with afferents predicted from Schneider's model and Vannucci et al.'s model, when providing those models the input of the experimentally-measured muscle lengths. With both Blum et al.'s data and digitized article data, the average RMS error by which Vannucci et al.'s predicted afferents fail is more than the average RMS error by which Schneider's predicted afferents fail. This is likely because the height of Vannucci et al.'s predicted afferents is more accurate than that of Schneider's predicted afferents, although the shape of Schneider's predicted afferents is more accurate than that of Vannucci et al.'s predicted afferents, by observation. The error between $2 \times$ Vannucci et al.'s predicted afferents - Schneider's predicted afferents and experimentally-measured afferents is less than the error between Vannucci et al.'s predicted afferents and Blum et al.'s experimentally-measured afferents, and is less than the error between Schneider's predicted afferents and Blum et al.'s experimentally-measured afferents, and therefore using Vannucci et al.'s predicted afferents and Schneider's predicted afferents together

predicts experimentally-measured afferents better than using either alone.

References

- Biddiss, E. A., & Chau, T. T. (2007). Upper limb prosthesis use and abandonment: a survey of the last 25 years. *Prosthetics and orthotics international*, 31(3), 236-257. doi:10.1080/03093640600994581
- Blum, K. P., D'Incamps, B. L., Zytnicki, D., & Ting, L. H. (2017). Force encoding in muscle spindles during stretch of passive muscle. *PLoS Computational Biology*, 13(9), e1005767. doi:10.1371/journal.pcbi.1005767
- Blum, K. P., Lamotte D'Incamps, B., Zytnicki, D., & Ting, L. H. (2018, September). Data from: *Force encoding in muscle spindles during stretch of passive muscle*. Retrieved from <https://datadryad.org/stash/dataset/doi:10.5061/dryad.pd40m>.
- Bradski, G. (2000). The OpenCV Library, opencv-python version 4.1.2.30. *Dr. Dobb's Journal of Software Tools* [Computer software]. Retrieved from <https://opencv.org/releases/>.
- Day, J., Bent, L. R., Birznieks, I., Macefield, V. G., & Cresswell, A. G. (2017). Muscle spindles in human tibialis anterior encode muscle fascicle length changes. *Journal of Neurophysiology*, 117(4), 1489-1498. doi:10.1152/jn.00374.2016
- Dimitriou, M. (2016). Enhanced muscle afferent signals during motor learning in humans. *Current Biology*, 26(8), 1062-1068. doi:10.1016/j.cub.2016.02.030
- Edin, B. B., & Vallbo, A. B. (1990). Dynamic response of human muscle spindle afferents to stretch. *Journal of Neurophysiology*, 63(6), 1297-1306. doi:10.1152/jn.1990.63.6.1297
- Frigon, A., Johnson, M. D., & Heckman, C. J. (2012). Differential modulation of crossed and uncrossed reflex pathways by clonidine in adult cats following complete spinal cord injury. *The Journal of Physiology*, 590(4), 973-989. doi:10.1113/jphysiol.2011.222208
- Gewaltig, M. O., Morrison, A., & Plesser, H. E. (2012). NEST by example: an introduction to the neural simulation tool NEST. In *Computational Systems Neurobiology* (pp. 533-558). Springer, Dordrecht. doi:10.1007/978-94-007-3858-4_18
- Huwaldt, J. A., & Steinhorst, S. (2015). PlotDigitizer version 2.6.8 [Computer software]. Retrieved from <https://sourceforge.net/projects/plotdigitizer/>.
- Jones, K. E., Wessberg, J., & Vallbo, Å. B. (2001). Directional tuning of human forearm muscle afferents during voluntary wrist movements. *The Journal of Physiology*, 536(2), 635-647. doi:10.1111/j.1469-7793.2001.0635c.xd
- Malik, P., Jabakhanji, N., & Jones, K. E. (2015). An Assessment of Six Muscle Spindle Models for Predicting Sensory Information during Human Wrist Movements. *Frontiers in Computational Neuroscience*, 9. doi:10.3389/fncom.2015.00154
- Marasco, P. D., Bourbeau, D. J., Shell, C. E., Granja-Vazquez, R., & Ina, J. G. (2017). The neural response properties and cortical organization of a rapidly adapting muscle sensory group response that overlaps with the frequencies that elicit the kinesthetic illusion. *PloS One*, 12(11). doi:10.1371/journal.pone.0188559
- Matthews, P. B. C. (1962). The differentiation of two types of fusimotor fibre by their effects on the dynamic response of muscle spindle primary endings. *Quarterly Journal of Experimental Physiology and Cognate Medical Sciences: Translation and Integration*, 47(4), 324-333.

doi:10.1113/expphysiol.1962.sp001616

Matthews, P. B. C. (1963). The response of de-efferented muscle spindle receptors to stretching at different velocities. *The Journal of Physiology*, 168(3), 660-678.

doi:10.1113/jphysiol.1963.sp007214

Matthews, P. B. C., & Stein, R. B. (1969). The sensitivity of muscle spindle afferents to small sinusoidal changes of length. *The Journal of Physiology*, 200(3), 723-743.

doi:10.1113/jphysiol.1969.sp008719

Mileusnic, M. P., Brown, I. E., Lan, N., & Loeb, G. E. (2006). Mathematical models of proprioceptors. I. Control and transduction in the muscle spindle. *Journal of Neurophysiology*, 96(4), 1772-1788.

doi:10.1152/jn.00868.2005

Nichols, T. R., & Cope, T. C. (2004). Cross-bridge mechanisms underlying the history-dependent properties of muscle spindles and stretch reflexes. *Canadian Journal of Physiology and Pharmacology*, 82(8-9), 569-576. doi:10.1139/Y04-074

Prochazka, A., & Gorassini, M. (1998). Ensemble firing of muscle afferents recorded during normal locomotion in cats. *The Journal of Physiology*, 507(1), 293-304.

doi:10.1111/j.1469-7793.1998.293bu.x

Puljak, L. (n.d.). Extracting data from figures using software. Retrieved from

https://training.cochrane.org/sites/training.cochrane.org/files/public/uploads/resources/downloadable_resources/2016_11_webinar_Puljak-extracting-data-from-figures.pdf.

Raybaut, P. & Cordoba, C. (2009). Spyder: The Scientific Python Development Environment, version 3.3.3, with Python 3.7.3, Qt 5.9.6, PyQt5 5.9.2, and Windows 10 . [Computer software]. Retrieved from <https://github.com/spyder-ide/spyder>.

Rosenthal, N. P., McKean, T. A., Roberts, W. J., & Terzuolo, C. A. (1970). Frequency analysis of stretch reflex and its main subsystems in triceps surae muscles of the cat. *Journal of Neurophysiology*, 33(6), 713-749. doi:10.1152/jn.1970.33.6.713

Saul, K. R., Hu, X., Goehler, C. M., Vidt, M. E., Daly, M., Velisar, A., & Murray, W. M. (2015). Benchmarking of dynamic simulation predictions in two software platforms using an upper limb musculoskeletal model. *Computer Methods in Biomechanics and Biomedical Engineering*, 18(13), 1445-1458. doi:10.1080/10255842.2014.916698

Schneider, O. (2013). *An Interactive Simulation of The Muscle Spindle*. Retrieved from <https://github.com/oschneid/RTMuscleSpindle>.

Spector, S. A., Gardiner, P. F., Zernicke, R. F., Roy, R. R., & Edgerton, V. R. (1980). Muscle architecture and force-velocity characteristics of cat soleus and medial gastrocnemius: implications for motor control. *Journal of Neurophysiology*, 44(5), 951-960. doi:10.1152/jn.1980.44.5.951

Vannucci, L., Falotico, E., & Laschi, C. (2017). Proprioceptive Feedback through a Neuromorphic Muscle Spindle Model. *Frontiers in Neuroscience*, 11, 341. doi:10.3389/fnins.2017.00341

Vannucci, L., Falotico, E., & Laschi, C. (2018). NeuralModels. Retrieved from <https://gitlab.com/sssa-humanoid-robotics/NeuralModels>.

Appendix A: How to manually digitize data in figures of scientific articles

Puljak's (n.d.) instructions on slides 13-28 (inclusive) explain this process well. I add to that explanation with Figures 5 and 6, which show how to manually digitize a figure. The notes below explain each step of Figures 5 and 6.

Steps

(Note that the figure being digitized in this example is digitized article 2, figure 1.)

1. Open the file in PlotDigitizer. Click "Digitize." Click on the left-most end of the x-axis.
2. Type the x-axis minimum. Click "Okay."
3. Click on the right-most end of the x-axis.
4. Type the x-axis maximum. Click "Okay." (The x-axis maximum was calculated by finding the number of pixels in a line with a width or height of known units which aren't pixels. For the image shown, I determined that the 1-second measuring bar in this figure (digitized article 2, figure 1) is 68 pixels, and the x-axis length in the figure being digitized is 300 pixels, so $(300 \text{ pixels})(1 \text{ s}/68 \text{ pixels})=4.412 \text{ s}$, which is the x-axis maximum given 0 is the x-axis minimum.)
5. Click on the left-most end of the y-axis.
6. Type the y-axis minimum. Click "Okay."
7. Click on the right-most end of the y-axis.
8. Type the y-axis maximum. Click "Okay."
9. Type the x-axis units. Click "OK."
10. Type the y-axis units. Click "OK."
11. Click "Zoom: In" until zoomed in as much as possible (700%).
12. Click on each point of interest in the graph.
13. Finish clicking on each point of interest in the graph. Click "Done."
14. Click "File" in the pop-up containing the (x,y) coordinates of all the digitized points.
15. Click "Save As..." in the drop down from the pop-up. Save the file.

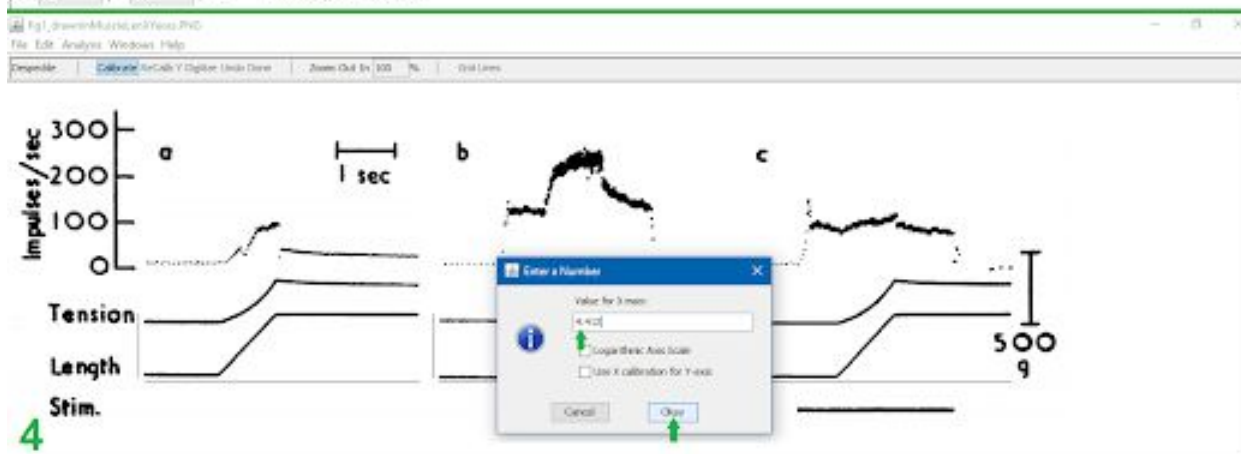
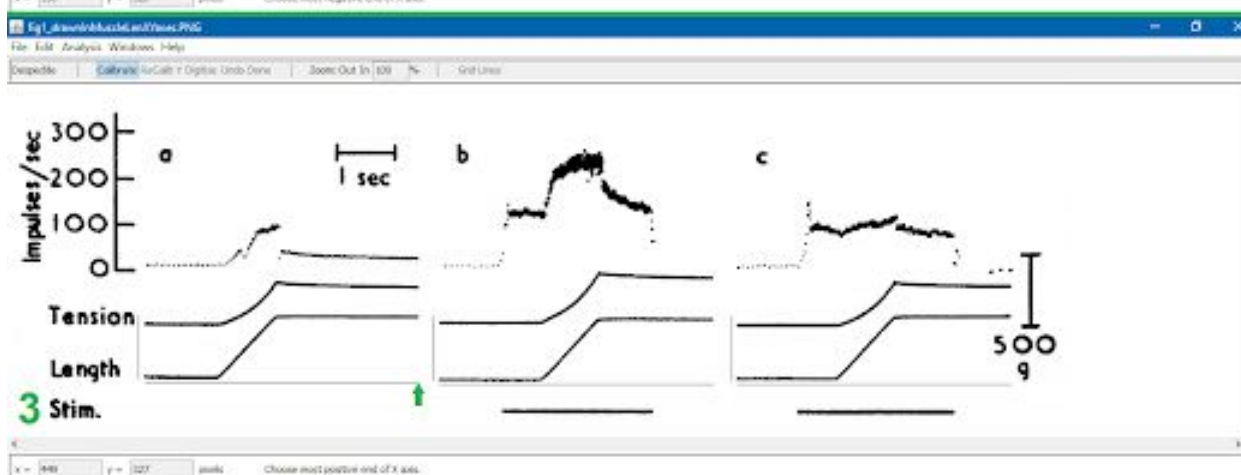
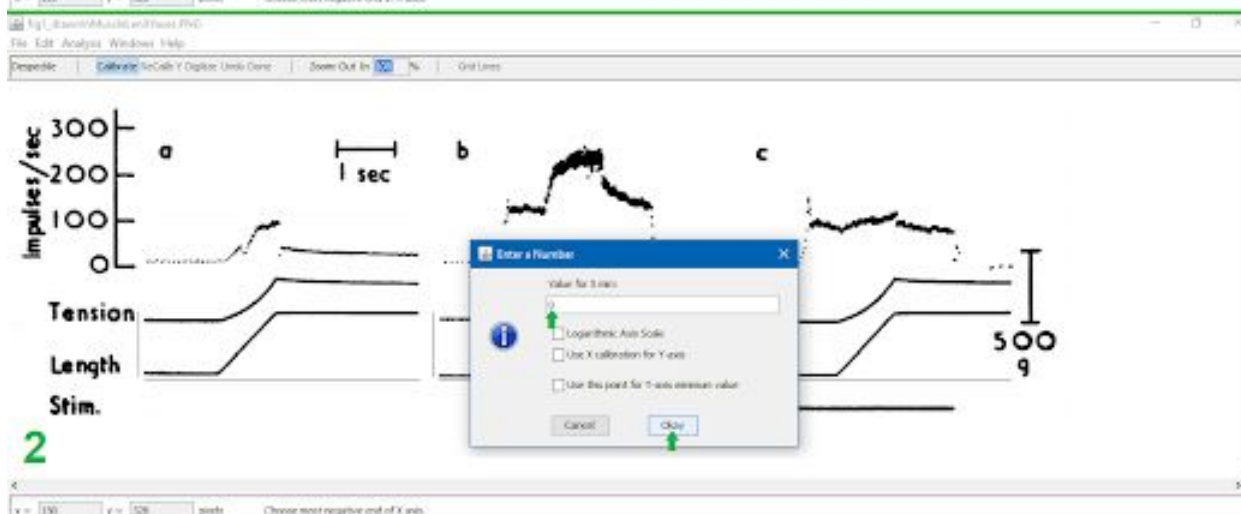
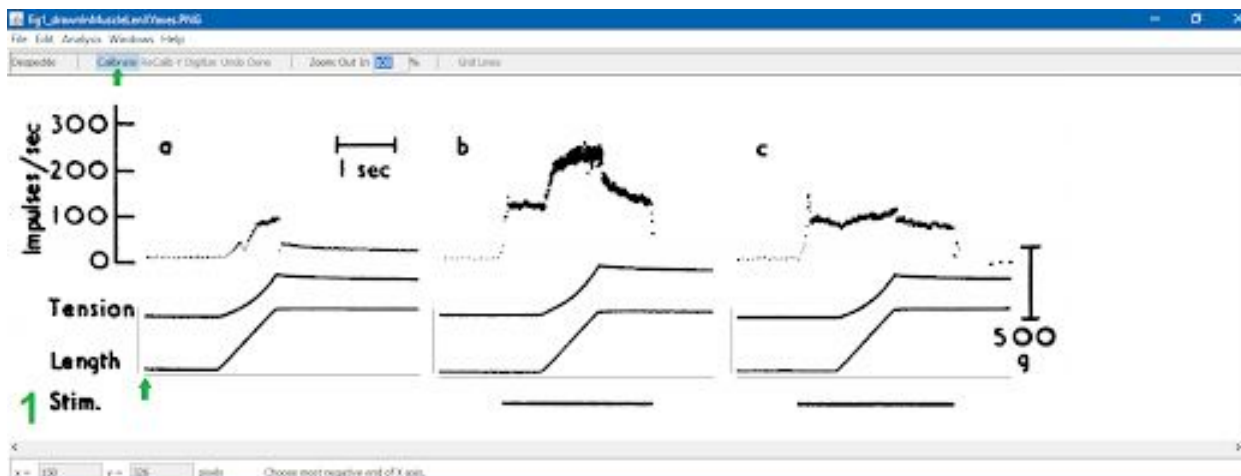


Figure 5. A walkthrough on manual digitization. Steps 1-8 (inclusive) explain the figure.

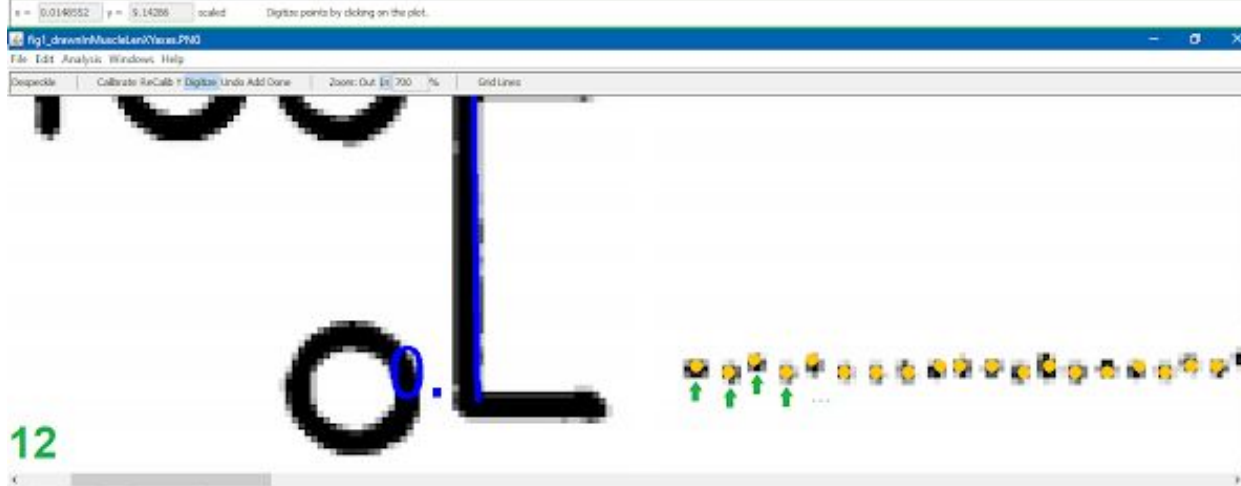
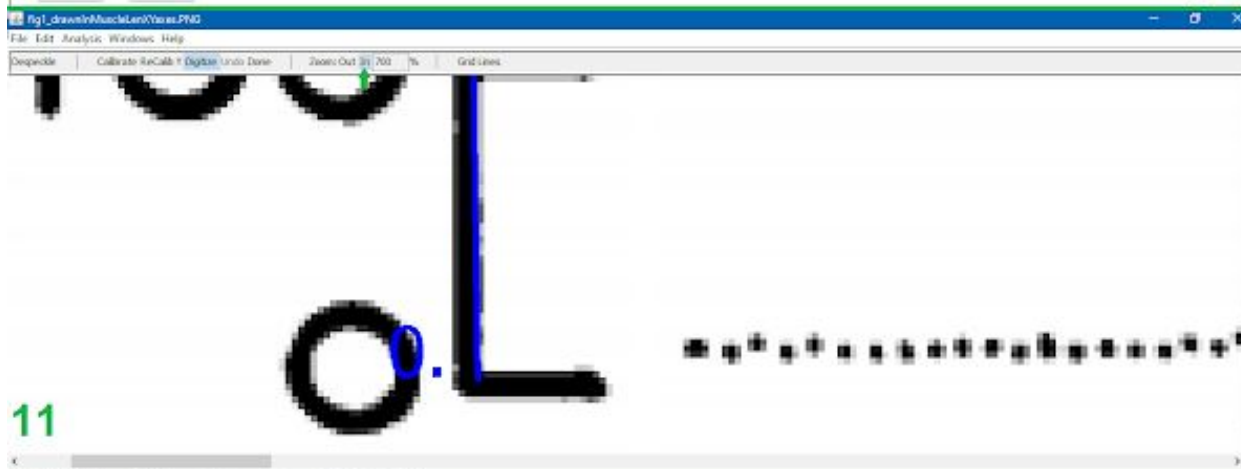
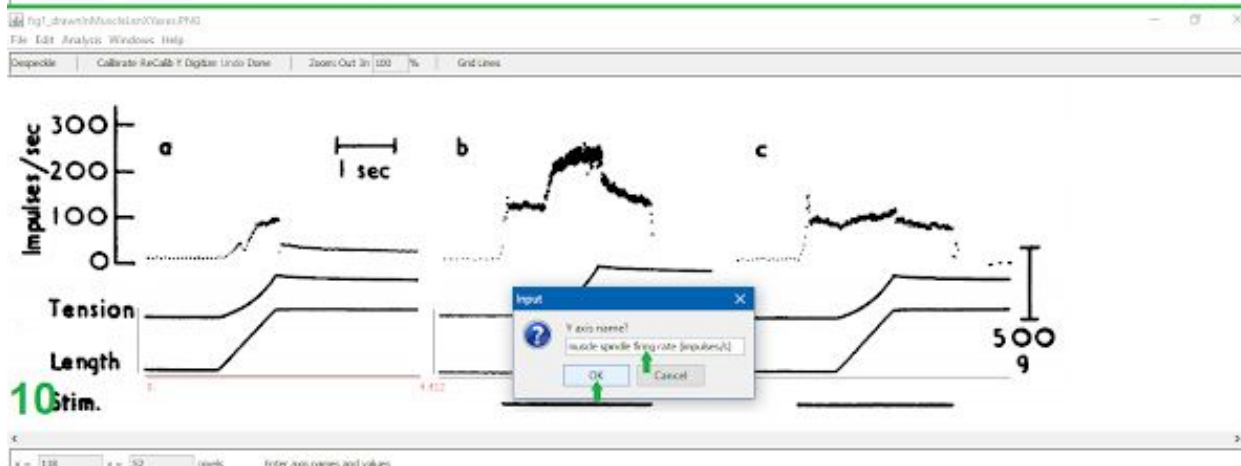
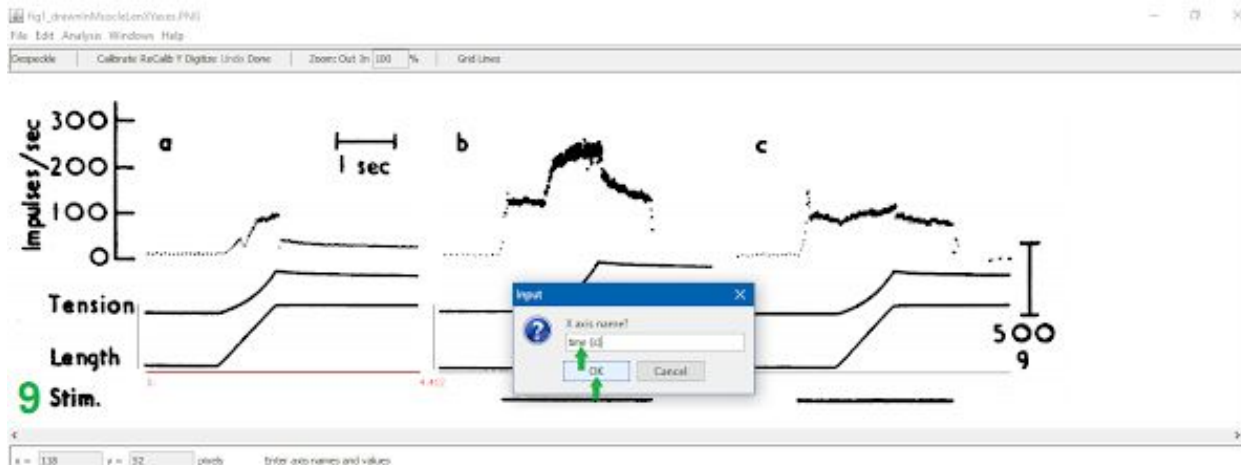


Figure 6. The continuation of a walkthrough on manual digitization. Steps 9-15 (inclusive) explain the figure.

Appendix B: How to automatically digitize data in figures of scientific articles

Get a scale

All images used were .png format, and were screenshots (taken with Microsoft Snipping Tool) of .pdf files of research papers (viewed with Google Chrome). Open the image in Microsoft Paint, use the Select Tool, and measure the length (height or width, depending on how the measuring line is aligned) (in pixels) of some line in the image which indicates how many graph units the line is. This line may be the axes (as in purple number 1, 2, 3, and 4 of Figure 7), or may be a separate bar on the side (as in purple number 5 and 6 of Figure 7). Axes may apply to several graphs (as in all graphs of Figure 7), including the graph of interest, so you may have to screenshot several unrelated graphs so that your screenshot includes the graph.

Remove lines

Remove any axes, labels, titles, legends, graph lines, or anything except the datapoints to be digitized. See Figures 8 and 9 for examples of removing guiding dots and lines which aren't datapoints, but are the same color of the datapoints.

Run the program

Run digitize.py, which searches for the color with the blue-green-red (BGR) lower and upper bounds specified in the code, and writes each point of that color to a comma-separated values (.csv) file. See Figures 10, 11, and 12 for examples of datapoints of a certain color extracted from a graph which has curves of various colors.

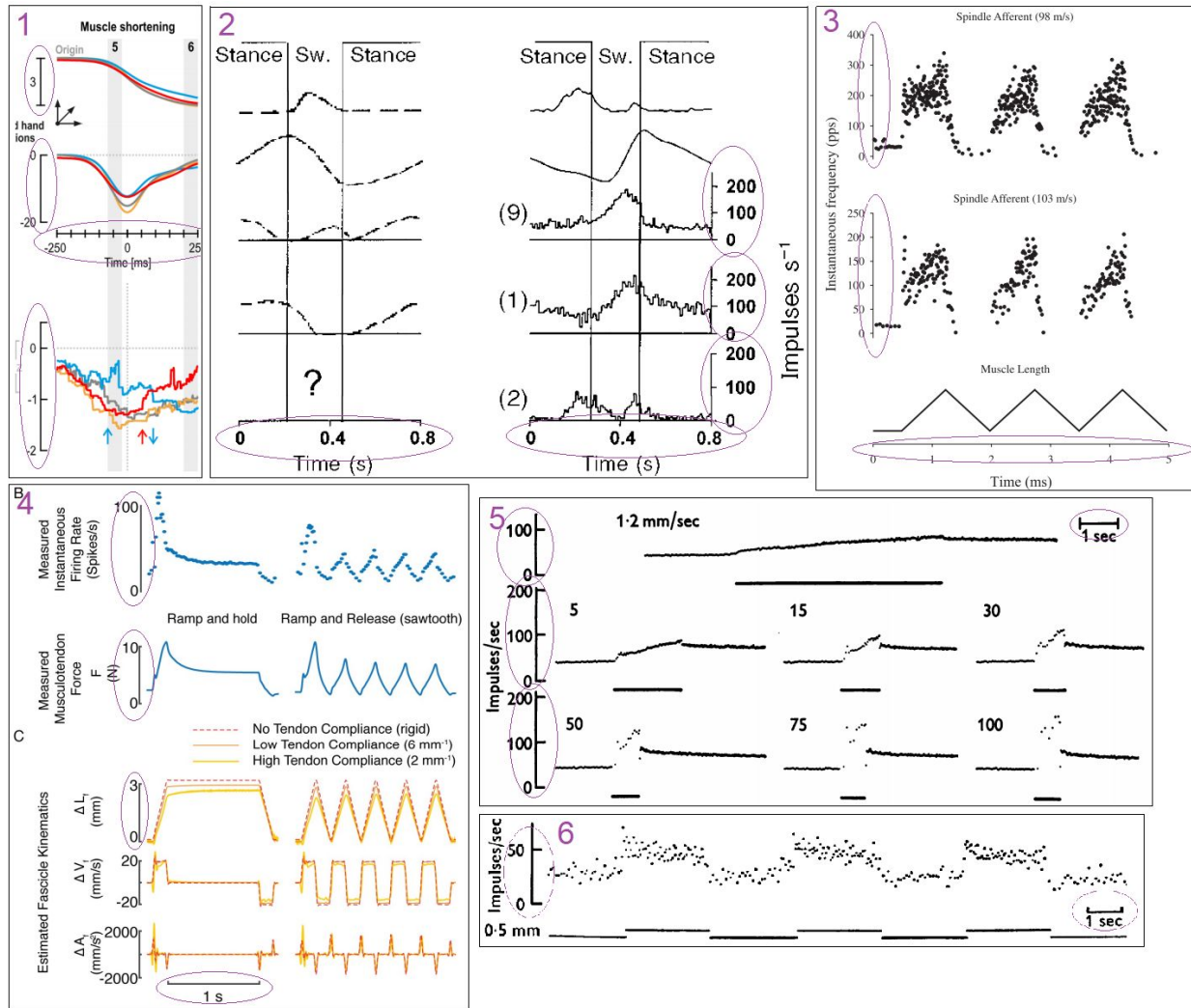


Figure 7. Examples of some line (circled in purple) in the image which indicates how many graph units the line is. This line may be the axes (as in purple number 1, 2, 3, and 4), or may be a separate bar on the side (as in purple number 5 and 6). Examples of graphs where x or y axes apply to multiple graphs, not just the graph closest to the axes.

- Purple number 1. Data from article 1, figure 4, muscleShortening
- 2. Data from article 6, figure 6, Sartorius and Posterior hamstrings
- 3. Data from article 9, figure 2.
- 4. Data from article 4, figure 2.
- 5. Data from article 5, figure 4.
- 6. Data from article 7, figure 7.

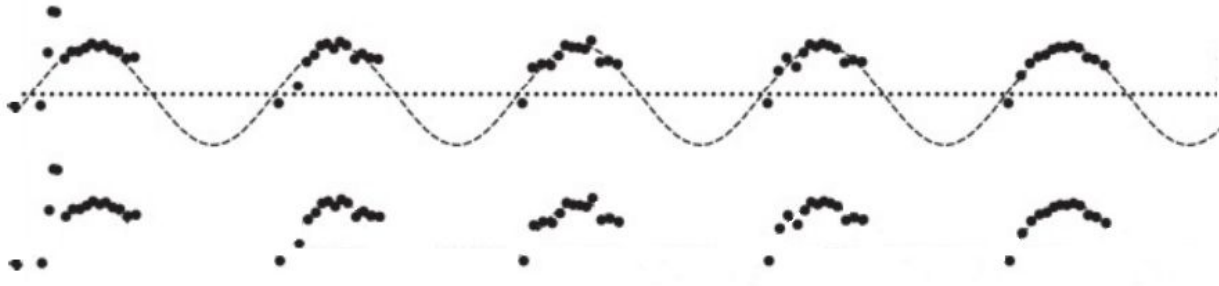


Figure 8. Examples of removing guiding dots which aren't datapoints, but are the same color of the datapoints. The figure is from digitized data article 8, figure 4.

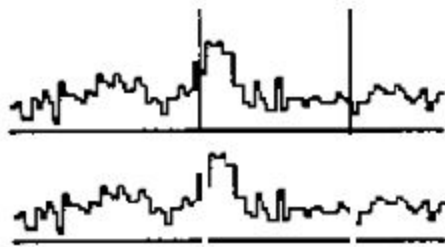


Figure 9. Examples of removing guiding lines which aren't datapoints, but are the same color of the datapoints. The figure is from digitized data article 6, figure 6, extensor digitorum longus Ia afferents.

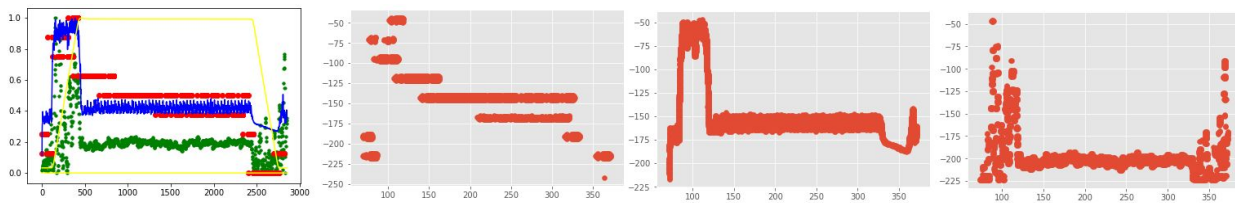


Figure 10. Examples of datapoints of a certain color extracted from a graph which has curves of various colors.

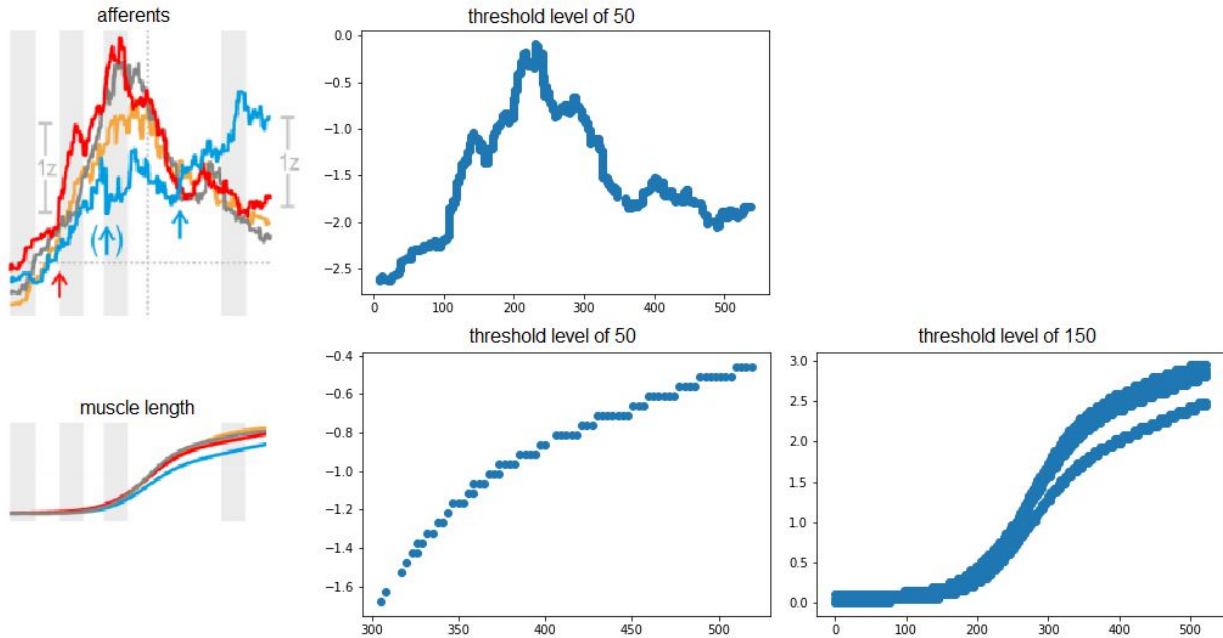


Figure 11. Article 1 figure 4 muscleLengthening graph, where a threshold level of 50 captures the grey line, but some of that line is covered by other colors, producing an incorrect graph. Therefore, although the threshold level of 150 captures multiple lines, I used that threshold level in order to get a graph with y values for each x value, for smooth simulation with Vannucci et al.'s and Schneider's models.

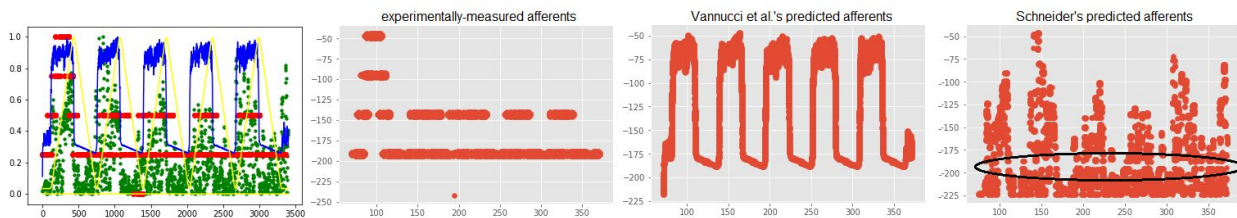


Figure 12. An example of getting graphs with holes, since colors cover other colors, when extracting datapoints of a certain color from a graph which has curves of various color.