

ANALYSIS WINDOW INDUCED CONTROLLER DELAY FOR MULTIFUNCTIONAL PROSTHESES

Todd R. Farrell, Ph.D.¹⁻³ and Richard F. Weir, Ph.D.²⁻⁴

¹ Liberating Technologies ² Biomechatronics Development Laboratory, Rehabilitation Institute of Chicago ³ Northwestern University, Department of Biomedical Engineering ⁴ Jesse Brown VA Medical Center, Lakeside CBOC

INTRODUCTION AND VARIABLE DEFINITION: Many upper-limb, multifunctional prosthesis controllers analyze fixed segments of EMG data collected from the residual musculature in an attempt to discern the intended movement of the user. However, many researchers have designed controllers with little or no regard for the delay the controller will introduce when operated in real-time. If the delay is too large the prosthesis will feel sluggish and performance will suffer. Several attributes of the classifier affect the delay it will create.

State-based pattern recognition classifiers typically collect EMG data in 'analysis windows' whose length will be defined as T_a . Class decisions based upon these collected data cannot be generated instantaneously because time is required to both record and then process the EMG. The processing time (τ) is the time from the completion of data collection until a class decision is made. The length of the window being analyzed (T_a), the microprocessor used to perform the calculations, as well as the number of channels and the number and type of features being extracted will determine τ . Thus τ must be determined empirically for each classifier.

Both overlapped and disjoint analysis windows have been employed in experimental prosthesis controllers. Windows can be overlapped if the analysis windows are incremented by some amount of time (T_{new}) that is greater than the processing time (τ). Overlapping the windows increases the density of class decisions which will allow majority voting. Majority voting is a post-processing strategy that has been shown to increase classifier accuracy [1-2] by analyzing the current class decision along with the $n-1$ previous class decisions and selecting the class that occurs most frequently in those n decisions as the controller output.

The authors recently completed a study which found that 100 ms was the maximum amount of time that could be used to collect and analyze EMG signals (to maximize the classification accuracy) without substantially degrading the performance of the prosthesis [3]. This finding implies that the values of T_a , T_{new} , n and τ should be set to ensure that the amount of time from the user's intended change in class until the change in the output of the controller (i.e., the controller delay or 'D') is less than 100 ms. The goal of this work is to quantitatively define how each parameter (T_a , T_{new} , n and τ) affects the maximum delay as well as the range of delays introduced by the controller. Four controller configurations were examined including those that use overlapped or disjoint windows as well as those that did or did not use majority voting.

Note: the data are collected with a sampling period of T_s and a frequency of $1/T_s$ Hz.

EQUATION DERIVATION: For the sake of brevity, only the calculation of the delay for a controller utilizing a three-vote majority voting scheme with overlapped windows will be discussed in detail. However, Table I contains the results for all four controller configurations. The three votes considered in the majority vote are shown as small numbers to the right of each of the analysis and computation windows in Fig. 1. The current class decision is in bold and on top of the three numbers and the previous two class decisions are shown in the smaller, non-bolded font. The large bold number represents the classifier output, i.e. the majority vote winner. *Note:* The subsequent analyses examine state-select controllers without proportional control.

Two assumptions made the following analysis tractable. The first was that the output of the classifier will belong to the class whose data fills a majority of the analysis window. The second was that the change in the EMG reflecting a change in the intended movement of the user occurs instantly. These assumptions were found to be good first order approximations and the derived equations yielded good estimates of empirically measured controller delays (see below).

The controller delay can vary substantially from one contraction to the next because the change in the intended movement can happen anywhere within a particular analysis window. The worst case occurs if the intended change in class occurs just after half-way through the window (see the third analysis window of Fig. 1). Here, *more* than 50% of the data in this window belongs to Class 1 and *less* than 50% of the data belongs to Class 2. The assumption would then be that the output decision for this window would belong to Class 1. The controller does not produce a 'Class 2' decision until the next window of data is analyzed. The fourth analysis window

in Fig. 1 shows that although the current class decision is Class 2 (small bold number on top of the three), the two previous decisions belonged to Class 1. Therefore, Class 1 has the majority of the votes and is the output of the controller. It isn't until the next window produces an output of Class 2 that the winner of the majority vote is changed to Class 2.

Table I shows a summary of the equations for the worst-case, average, and best-case delays for the four controller configurations. Each of the equations assumes that the sampling period (T_s) is negligibly small. Additionally, the table provides the range of possible delays that could be experienced by the user. Ideally, this range would be very small. Small differences in the delay give the users a consistent response from their device. Performance in virtual environments was shown to degrade as the variability of the delay in the system increased [4]. The overlapping window approaches (T_{new}) shows a different delay range than non-overlapped approaches (T_a). From our experience, it is typical for T_{new} to be approximately an order of magnitude less than T_a . Therefore, for a given set of analysis window attributes, an overlapped approach will provide a more consistent response and should improve performance when compared to non-overlapped approaches. If the equations are examined closely, the overlapped window approach also reduces the maximum controller delay and limits the increase in the controller delay introduced by majority voting for a given set of analysis window attributes.

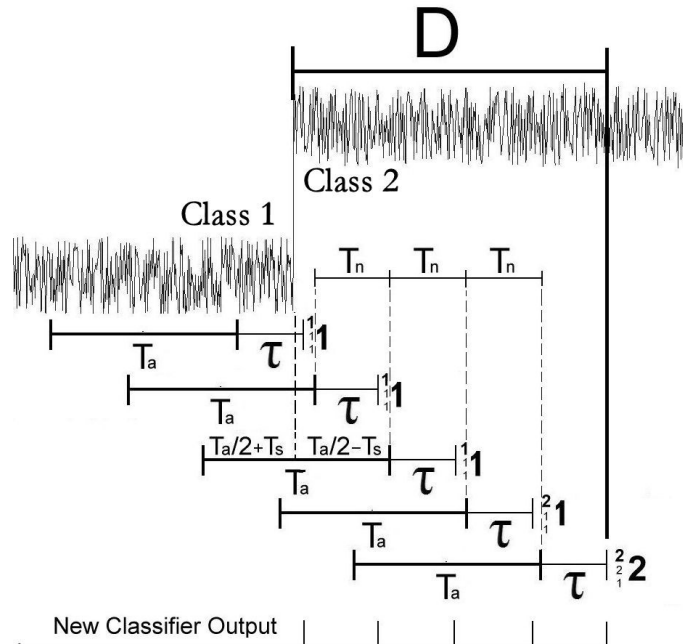


Fig. 1 - The maximum controller delay (D) is shown for a system that employs overlapped windows with a 3-vote majority voting scheme. The three votes are shown as the small numbers to the right of the analysis and computation windows. The top number of these three represents the current class decision. The output of the controller (winner of the majority vote) is shown as the large number to the far right of each window.

TABLE I

WORST-CASE, AVERAGE AND BEST-CASE DELAYS WITH DELAY RANGES				
Classifier Type	Worst-case Delay Equation	Average Delay Equation	Best-case Delay Equation	Difference Between the Best and Worst Cases
No overlap, No majority voting	$D = \frac{3}{2}T_a + \tau$	$D = T_a + \tau$	$D = \frac{1}{2}T_a + \tau$	T_a
No overlap, With majority voting	$D = \left(\frac{n}{2} + 1\right)T_a + \tau$	$D = \left(\frac{n+1}{2}\right)T_a + \tau$	$D = \left(\frac{n}{2}\right)T_a + \tau$	T_a
Overlap, No majority voting	$D = \frac{1}{2}T_a + T_{new} + \tau$	$D = \frac{1}{2}T_a + \frac{1}{2}T_{new} + \tau$	$D = \frac{1}{2}T_a + \tau$	T_{new}
Overlap, With majority voting	$D = \frac{1}{2}T_a + \left(\frac{n+1}{2}\right)T_{new} + \tau$	$D = \frac{1}{2}T_a + \left(\frac{n}{2}\right)T_{new} + \tau$	$D = \frac{1}{2}T_a + \left(\frac{n-1}{2}\right)T_{new} + \tau$	T_{new}

D = Controller Delay; T_a =Analysis Window Length; τ =Processing Time; T_{new} =Analysis Window Increment; n = Number of Majority Votes □

EQUATION VALIDATION: EMG data were collected as a subject transitioned between contraction classes to provide a validation of the derived equations. A linear discriminant analysis (LDA) classifier with overlapping windows performed pattern recognition on the RMS amplitude of a single EMG channel. This feature was chosen to keep the classification process as simple as possible and avoid potential confounding effects from the classifier. The subject produced constant extension of all four fingers for the duration of the trial and then alternated between relaxing or extending the wrist. These two movement classes were chosen because they have similar feature set variance and thus limit the effect that different feature set variances could have on the results. Two trials consisting of ten repetitions of the wrist extension movements were collected. Half of the data was used to train the classifier and half was used for testing. The results were then cross-validated by switching the training and testing data sets.

The test contractions were used to calculate the time from the change in the contraction class to the classifier producing the wrist extension output for five different analysis window lengths and majority vote combinations (Fig. 2). Fig. 2 also contains the estimated controller delays determined by the equations described in this paper (from Table I) as well as estimates using previously published methods [5]. The combinations of analysis window lengths and majority votes were chosen to keep the worst-case controller delay below the 100 ms value determined by Farrell and Weir [3]. The following analysis window lengths, number of majority votes and processing times/window shifts were investigated: 160 ms window, 1 vote, 10 ms shift; 120 ms window, 5 votes, 10 ms shift; 80 ms window, 21 votes, 5 ms shift; 40 ms window, 50 votes, 3 ms shift; and 20 ms window, 87 votes, 2 ms shift. The change in contraction classes were determined by visually examining the raw EMG signal.

Fig. 2 shows that the equations described in this paper (■) more accurately predicted the experimentally observed average prosthesis controller delay (Δ) than the previously published method (●), especially for larger analysis window lengths. The primary difference between our work and the previously published equations results from the fact that the previous work did not account for the fact that half of the analysis window needs to be filled with data from and new

class before the controller can recognize that a different movement is being produced.

Fig. 2 also indicates that there was a substantial amount of variability across the ten contractions as indicated by the wide standard deviations. The previous analysis did predict some variability in the controller delay but these values were lower than those that were observed. However, the previous analysis did predict an increase in variability with increasing window shift (T_{new} , see Table I). This increase in variability was observed as a monotonically increasing trend with increasing window shift. The increased variability in the delay calculations was likely primarily due to the variability in the author's estimate of the instant of the class change resulting from the transitory portions between the contraction classes not being nearly as consistent as the idealized step change in the EMG class. Regardless, the equations still do a good job of predicting the experimentally observed average controller delay.

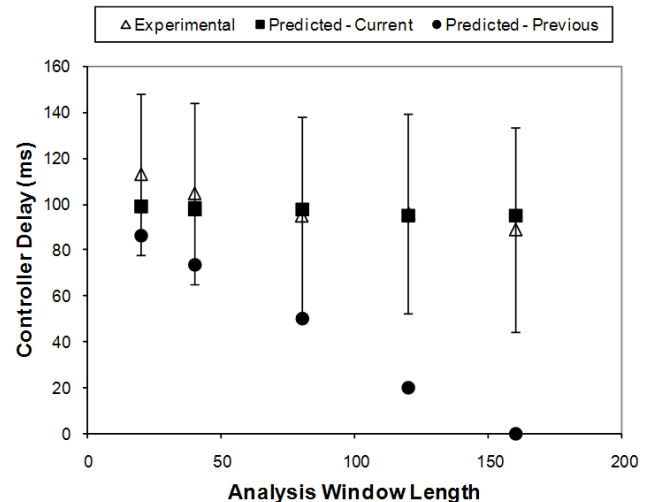


Fig 2 - Average time from the transition between contraction classes to the classifier producing the correct output. Experimental data from ten class transitions is shown (open triangles) as well as the predicted delays using both the methods described in this paper (filled squares) and by previous researchers (filled circles). The equations described in this paper provide more accurate estimates, especially for larger analysis windows.

CONCLUSION: A quantitative assessment of the effect of various analysis window attributes on the range of delays that can be produced between the intended change in movement class and the controller's associated classifier output decision was presented. Equations were derived for systems that used overlapped vs. non-overlapped windows as well as those that did or did not implement majority voting. These equations were validated and shown to produce more accurate estimates of the controller delay than previous efforts. This work highlighted that the delay is not simply a function of how fast the controller is able to produce a new output but instead is a function of the length of the analysis window length, the signal processing time and the number of majority votes being utilized. Finally, it was demonstrated that using an overlapped window approach produces a more consistent controller delay and decreases the maximum delay produced by the controller.

REFERENCES:

- [1] Englehart K, Hudgins B, Chan A. "Continuous multifunction myoelectric control using pattern recognition," *Technol Disabil.* 2003;15:95-103.
- [2] Huang Y, Englehart KB, Hudgins B, Chan AD., "A Gaussian mixture model based classification scheme for myoelectric control of powered upper limb prostheses," *IEEE Trans Biomed Eng.* 2005 Nov;52(11):1801-11.
- [3] Farrell TR, Weir RF., "The optimal controller delay for myoelectric prostheses," *IEEE Trans Neural Syst Rehabil Eng.* 2007 March;15(1):111-8.
- [4] Park KS, Kenyon RV, "Effects of network characteristics on human performance in a collaborative virtual environment," *IEEE Conference on Virtual Reality;* 1999 March 13th-17th; Houston, TX.
- [5] Englehart K, Hudgins B., "A robust, real-time control scheme for multifunction myoelectric control," *IEEE Trans Biomed Eng.* 2003 Jul;50(7):848-54.