



Todd R. Farrell, PhD

Determining delay created by multifunctional prosthesis controllers

I am writing to express a concern that my colleagues from previous laboratories at Northwestern University/Rehabilitation Institute of Chicago (RIC) and I have shared for a number of years related to multifunctional prosthesis control. Previous investigations have used a variety of analysis window attributes for their multifunctional prosthesis controllers [1–8]. Researchers have varied the length of the analysis window, the amount of overlap between consecutive windows, and the number of majority votes used in the postprocessing of the classifier decisions. However, we believe that many researchers have made decisions about these attributes with little regard for the overall delay created in the real-time system. (Note that the term *classifier* typically refers to an element of the controller that uses the inputs provided to it, e.g., electromyographic [EMG] signals, force sensor data, and position sensor data to decide which joint(s) of the prosthesis should be actuated.) For example, Peleg et al. performed classification decisions on data up to 1.4 seconds after the onset of the contraction [9]. This classifier would require its user to wait on the order of seconds for Peleg et al.’s prosthesis to respond, which would likely be quite frustrating for the user. We are not suggesting that new algorithms should not be explored simply because they may create substantial delays. However, we do believe that these delays should be considered and discussed in each article that is published on this topic. While a particular classifier may create a 1 percent increase in classification accuracy, if it cannot add this increase in accuracy in a reasonable amount of time, it may be a “non-starter.” Continuing in this vein, we would like to discuss some findings that we believe will allow prosthesis controller designers to better understand how their controllers will behave in real time. (For the purpose of this editorial, further use of “we” and “our” indicates my colleagues and me.)

This editorial focuses on analysis window issues from the perspective of EMG-based multifunctional prosthesis control. However, the work described can be extended to other forms of window-based biosignal classifiers, such as brain machine interfaces that use neural spike-counting algorithms for intracortical data [10–12] or those that use electroencephalogram recordings [13–14].

BACKGROUND AND VARIABLE DEFINITION

Many readers may be familiar with multifunctional prosthesis classifiers, but I would like to provide a brief introduction for those who are new to the area. Many multifunctional prosthesis controllers analyze EMG signals collected from the residual limb in an attempt to determine the intended

movement of the user. These controllers use EMG data to decide which degree of freedom will be actuated and then produce motor-drive signals for the corresponding joint of the prosthesis. Classifiers typically make these decisions by comparing data collected in real time to data collected during a training session. Generally, the movement of the prosthesis whose training data best match the current real-time data sample is then selected as the output of the classifier. These class decisions can then be used for determining the “classification accuracy” of a classifier, which is defined as the percentage of the time that the output of the controller matches the intended movement of the user. Huang et al. provide a detailed description of the various steps of the classification process [15].

Some previous work has shown that classification accuracy increases when EMG feature extraction and pattern recognition are performed on larger data windows [1]. However, when more EMG data are collected for analysis, more time is required to collect and then process the larger data set. This collection and processing time increases the delay between the user producing a command and the controller responding appropriately. If this delay becomes excessive, it can make the prosthesis feel sluggish and unresponsive to the user. As such, a trade-off is made between the length of time needed to decipher the user’s intent accurately and the need for a quick and responsive device. I will focus the discussion on how several variables associated with the classification process affect the controller delay of a system. We define the controller delay as the amount of time from the user’s intended change in movement class (as reflected by a change in the EMG signal) until the classifier produces the correct output.

In pattern recognition-based controllers, the EMG data are frequently collected in “analysis windows” whose lengths are defined as T_a . Class decisions (i.e., decisions produced by the classifier indicating which prosthesis movement will be actuated) based on these collected data cannot be generated instantaneously because time is required to both record and process the EMG (e.g., extract signal features and/or perform pattern recognition). 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), as well as the number of channels and the number and type of features being extracted, will determine τ .

Overlapped Windows

Englehart and Hudgins explain that the processing time (τ) required to analyze the EMG can be much less than the length of the analysis window (T_a), causing the processor to lie idle a majority of the time [3]. **Figure 1** shows an example of using disjoint windows and visually demonstrates that class decisions (labeled as d1, d2, and d3 in **Figures 1** and **2**) can be produced relatively infrequently and that the processor is not being used to its full capacity.

To use the capabilities of the processor fully, Yamada et al. [16] and Englehart et al. [2–3] suggested increasing the frequency of class decisions by “sliding” the window along at increments that are slightly larger than τ . In this way, as soon as a class decision is generated, the controller immediately begins processing the next class decision on a “new” window of data. **Figure 2** (note the differences in lengths of time in the x -axes between **Figure 1** and **Figure 2**) shows how the use of overlapped windows increases the frequency of which class decisions are made and nearly maximizes the use of the processor.

When the classifier uses overlapped windows, the amount of new data that will be added to each new analysis window (i.e., the amount of “shift”) will be defined as T_{new} . Theoretically, minimum overlap can be achieved when $T_{\text{new}} = \tau$; however, in practice, T_{new} will be slightly larger than τ .

Majority Voting

A technique that takes advantage of the increased frequency of class decisions afforded by overlapping windows is “majority voting.” Majority voting is a postprocessing strategy that can increase the overall classifier accuracy [2,15,17]. For small analysis window lengths, it has been shown to be particularly effective at improving classifier robustness by smoothing out relatively noisy, but high density, streams of class decisions.

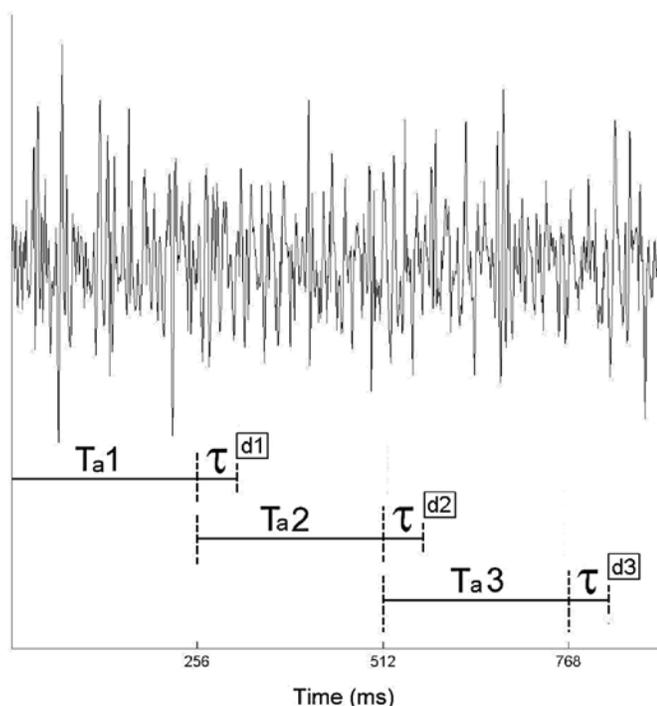


Figure 1.

Example of disjoint windowing of single channel of electromyographic data ($T_a = 256$ ms, $\tau = 64$ ms). Each analysis window (T_a) takes finite time to process (τ) before a decision (d1, d2, and d3) can be produced. Note that time required to process each window (τ) is much less than analysis window length and therefore processor is lying idle a majority of time. Figure is modified version. *Source:* Englehart K, Hudgins B. A robust, real-time control scheme for multifunction myoelectric control. *IEEE Trans Biomed Eng.* 2003;50(7):848–54. [PMID: 12848352] DOI:10.1109/TBME.2003.813539

The current class decision along with the $n-1$ previous class decisions (where n = number of majority votes) is analyzed, and the class that occurs most frequently in those n decisions is selected as the controller output.

I should point out that other groups are exploring new and innovative ways of further postprocessing the classifier outputs. For example, researchers at the Center for Bionic Medicine at RIC are currently working on “decision-based velocity ramps” that tie the speed of the selected movement to the number of consecutive decisions belonging to a particular movement class [18]. The velocity ramp causes the speed of any movement to start slowly and then increase to its maximum, but it also causes spurious misclassifications to create

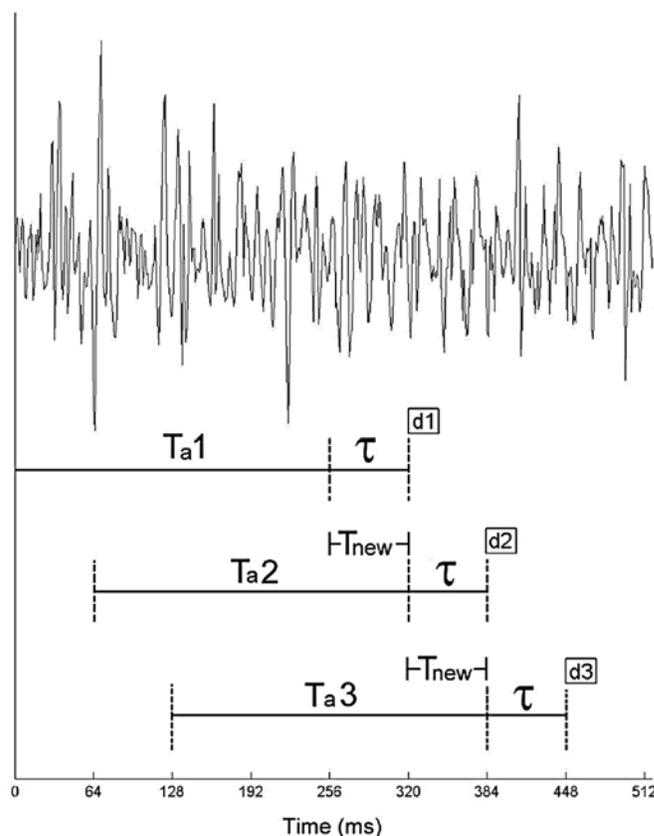


Figure 2.

Example of overlapped windowing of single channel of electromyographic data ($T_a = 256$ ms, $\tau = T_{\text{new}} = 64$ ms). When overlapped windows are used, analysis window slides along at relatively small increments, adding newly collected data and discarding oldest data. In this example, amount of new data in each window (T_{new}) is exactly equal to amount of time required to process data from previous analysis window (τ). Setting amount of overlap equal to processing time allows controller to begin processing next class decision immediately when decision on previous window's data has been completed. Overlapped windows increase frequency of class decisions, which makes postprocessing technique of majority voting tractable. Figure is modified version. *Source:* Englehart K, Hudgins B. A robust, real-time control scheme for multifunction myoelectric control. *IEEE Trans Biomed Eng.* 2003;50(7):848–54. [PMID: 12848352] DOI:10.1109/TBME.2003.813539

relatively small movements of the unintended degrees of freedom. This additional level of postprocessing would theoretically take longer to complete a given predetermined set of movements if the user were to create perfectly repeatable patterns of EMG activity. However, in reality, no user can produce perfectly repeatable contractions with each movement. The RIC team showed that subjects

completed tasks at higher rates with the velocity ramp postprocessing than without it because of the velocity ramps' potential to limit the deleterious effects of misclassifications. While the team at RIC is making great strides to improve function in the clinical environment with other postprocessing techniques, I will be focusing on the "classic" approaches that have been used extensively in the literature.

Optimal Controller Delay

My colleagues and I performed a study that examined the effects of different controller delays on the performance of an externally powered prosthesis [19]. We defined the "optimal controller delay" as the maximum amount of time that could be used to collect and analyze EMG signals (so as to increase the classification accuracy) without substantially degrading the performance of the prosthesis as measured using a functional task. Not surprisingly, we found an inverse relationship between performance on the Box and Block Test and the amount of delay that was present in the controller. While any delay in the controller caused a decrease in performance, this decrease was not found to be statistically or clinically significant until at least 100 to 125 ms of delay was present for an above-average (90th percentile) user. Therefore, we proposed that the controller delay that allowed the most time for signal collection and analysis (to maximize classification accuracy) without significantly decreasing prosthesis performance was found to be approximately 100 ms.

Our results support Smith et al.'s recently completed study that took the experiment a step further by examining the effects of window length on both classification accuracy and controller delay and then determining how these two factors combined to affect performance on virtual tasks [20]. Smith et al. concluded that, depending on the performance metric, the optimal controller delay ranged from 88 to 138 ms, which correlates well with our findings.

The results of our previous study and Smith et al.'s work imply that the ideal scenario would be setting the values of T_a , T_{new} , n , and τ to ensure that the longest amount of time from the intended change in class until the change in the output of the

controller (i.e., the controller delay [D]) is approximately 100 ms. Next, I will discuss the approach that was developed to improve on current methods of defining how each parameter (T_a , T_{new} , n , and τ) will affect both the maximum delay and range of possible delays introduced by the controller.

WORST-CASE ANALYSES

In the case of commonly used windowing strategies, the amount of time between the user's intended change in movement and the controller producing the correct output (i.e., the controller delay [D]) can vary substantially. First, I will examine the largest delays that would be expected for a controller using a given set of analysis window attributes. The following section will then discuss the best-case and average expected delays as well as the range of possible delay values.

Three assumptions were made for the equations I am presenting. 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 (e.g., from Class 1 to Class 2) occurs instantaneously. In reality, a short transitory phase exists in which the EMG signals resemble the steady state of neither Class 1 nor Class 2. However, I believe that the assumption of an instantaneous class change is reasonable as a first-order approximation and it was made to simplify the calculations and make the analysis more tractable. The equation validation experiment discussed later will demonstrate that accurate predictions of the controller delay are possible without considering the transition portion of the contraction. Finally, I am assuming that user intent for movement is represented by the onset of the EMG signal. While research has shown that EMG signals are created before movement of the limb [21], I believe that the EMG signal provides us an easily accessible signal that clearly indicates the timing of the user's intended movements.

Note that these assumptions only provide a *starting point* for determining how any particular attributes of a controller will affect the delay that is produced by

it. As I will discuss in detail later, several issues may affect the assumptions just listed. One example is related to thresholds for how well the output of a particular data window needs to match training data from a particular contraction class. Again, it should be stressed that researchers should consider how their specific controller will operate in real time.

Figures 3 to 6 that follow show an idealized, instantaneous change in the EMG signal produced during two different movements (Class 1 or 2). In these figures, the output class decision from the controller is given in a large bold font to the right of the analysis and computation windows. Also, the tick marks on the bottom of **Figures 3 to 6** represent the instances in which the controller produces a new class decision, i.e., these ticks indicate when the commands are sent to the motors and their frequency represents what my colleagues and I refer to as the “update rate” of the system. Note that for this discussion, the data are assumed to be collected with a sampling period of T_s and, therefore, at a frequency of $1/T_s$ Hz.

Disjoint Windows with No Majority Vote

First, consider the case in which only a single class decision is being used (no majority voting) and disjoint (nonoverlapped) windows of data are being examined. The worst case would occur if the intended change in class occurred just after halfway through the window (see the third analysis window of **Figure 3**). In this case, >50 percent of the data in this window belong to Class 1 and <50 percent of the data belong to Class 2. We would then assume that the output decision for that window would be Class 1. Therefore, the controller would not produce the correct class decision until the next full window of data was analyzed.

Based on these observations, the maximum time from the intended change in movement to the controller producing the correct class decision (D) would be

$$D = \frac{3}{2}T_a - T_s + \tau . \quad (1)$$

Since the sampling period T_s is much smaller than the other elements of **Equation 1** and can be

considered negligible, **Equation 1** can reasonably be approximated by

$$D \approx \frac{3}{2}T_a + \tau . \quad (2)$$

The sampling period T_s is shown in **Figure 3** to demonstrate that the class change occurs just after the midpoint of the analysis window; it will not be included in the final equations and is not included in **Figures 4 to 6**.

Disjoint Windows with Majority Vote

If majority voting were used on disjoint windows of data, the time necessary for the classifier to produce a correct decision would increase. As stated before, majority voting stipulates that the class that occurs most frequently in the previous n decisions will be the output of the classifier. Simply having the *current* class decision change to a new class would be insufficient to alter the overall output of the controller because the controller requires a *majority* of decisions to belong to the new class.

Consider the previous controller, but with a 3-vote majority voting scheme instead of the 1-vote approach. In **Figure 4**, the three votes considered in the majority-voting scheme are shown as the small numbers to the right of the analysis and computation windows. The current class decision is in bold and is the top number of the three stacked numbers. The previous two class decisions are the two smaller unbolded stacked numbers. Again, as in **Figure 3**, the large bold number represents the classifier output, i.e., the winner of the majority vote. The fourth analysis window in **Figure 4** shows that although the current class decision is Class 2, the two previous decisions belonged to Class 1, meaning that Class 1 has the majority of the votes and therefore is the output of the controller.

When the classifier uses disjoint windows with an n -vote majority voting scheme, the maximum time from the intended change in class to the controller correctly producing that class as an output (again, assuming that T_s is considered negligibly small) is

$$D \approx \left(\frac{n}{2} + 1\right)T_a + \tau . \quad (3)$$

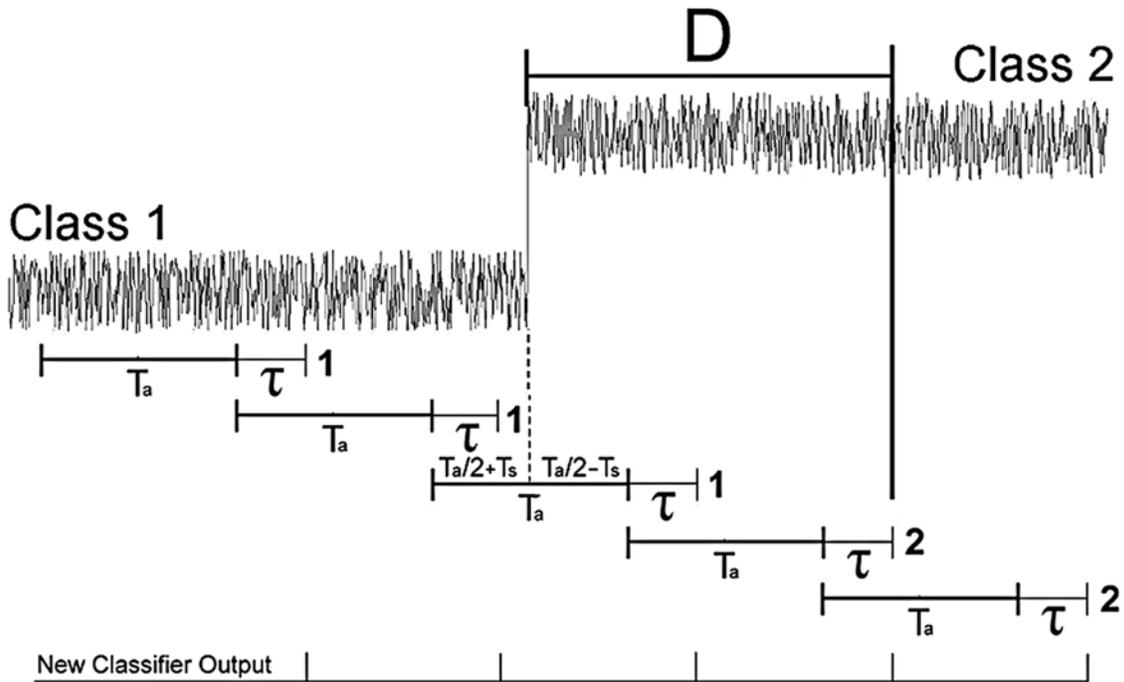


Figure 3.

Maximum controller delay (D), i.e., time from intended change in movement class (indicated by step change in electromyographic [EMG] signal) to controller producing correct output class, is shown for a system that employs disjoint windows with no majority voting. Output of controller is shown as large bold number to far right of each window and tick marks on bottom of figure indicate “update rate,” i.e., frequency with which new output is produced by controller. Example shown is “worst-case” scenario in which change in EMG occurs just after midpoint of analysis window. Since a majority of third analysis window is filled with data from Class 1, output of that window is assumed to belong to Class 1.

Note that if n is set to 1 (i.e., only a single vote is used and thus no majority voting exists), then **Equation 3** reduces to **Equation 2**.

Overlapped Windows with No Majority Vote

Overlapping the analysis windows reduces the maximum amount of time between an intended class change and the controller producing the correct output. **Figure 3** shows that if the intended class change occurs just after the midpoint of the analysis window, the controller output will not be correct until after the next full window of data is analyzed. However, when overlapped windows are employed, the window slides in increments of T_{new} . Therefore, if the class change occurs slightly after the midpoint of one analysis window, then after the analysis window has been shifted, a majority of the next window’s data will belong to the new class and the controller output will be changed. An example

is shown in **Figure 5**. The bottom trace of **Figure 5** also shows that the controller using overlapped windows can produce new outputs more frequently; i.e., the “update rate” is increased.

When the classifier uses overlapped windows with no majority voting, the maximum time from the intended change in class to the controller producing the correct output is

$$D \approx \frac{1}{2}T_a + T_{\text{new}} + \tau. \quad (4)$$

Overlapped Windows with Majority Vote

If majority voting is included in the overlapped window approach, then an additional amount of time will be required before the votes of the correct class can become a majority (**Figure 6**). In this case,

$$D \approx \frac{1}{2}T_a + \left(\frac{n+1}{2}\right)T_{\text{new}} + \tau. \quad (5)$$

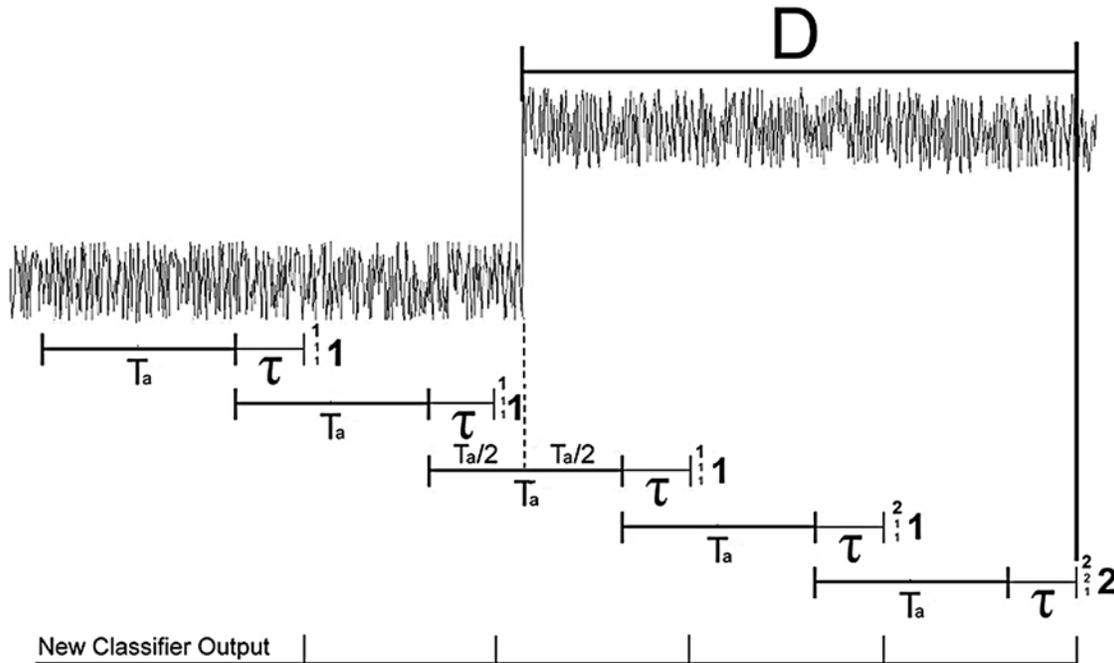


Figure 4.

Maximum controller delay (D) is shown for system that employs disjoint windows with 3-vote majority-voting scheme. Three votes are shown as small numbers to right of analysis and computation windows. Top number of these three is in bold and represents current class decision. Output of controller (winner of majority vote) is shown as large number to far right of each window.

The equations for worst-case delays for each of the controller types are summarized in **Table 1**.

BEST-CASE/AVERAGE ANALYSES AND DELAY RANGES

While it is beneficial to know the longest delays introduced by a controller, determining the range of delays that can be introduced is also worthwhile. Since the change in EMG has a uniform probability of occurring at any point within a particular analysis window, some variability will be found in the delays the user experiences and the average delay will simply be the mean of the best and worst cases. Ideally, the difference in delay between the best and worst cases would be very small. Small differences in the delay allow the users to get a consistent response from their devices. Research in virtual environments has shown that performance tends to degrade as the variability of the delay in the system increases [22–23].

For the worst-case (maximum delay) scenario, the change in class was assumed to have occurred just after halfway through the analysis window. For the best-case (minimum delay) scenarios, the class change was assumed to occur just before the midpoint of the analysis window. In this case, just over half of the window would contain data from the new class, and therefore, the output of the controller can be assumed to belong to the new class. Again, this assumption only provides a first-order approximation and will be affected by several factors. The equations for the average and best-case delays were derived similarly to the equations for worst-case delays and are shown in **Table 1**.

Table 1 also summarizes the difference between the best- and worst-case delays using the derived equations and thus determines the possible range of delays the users could experience. Note that when an overlapping window approach is used, the range of delays is equal to the amount of window shift (T_{new}), while a nonoverlapped approach has a possible delay range that is equal to the length of the analysis

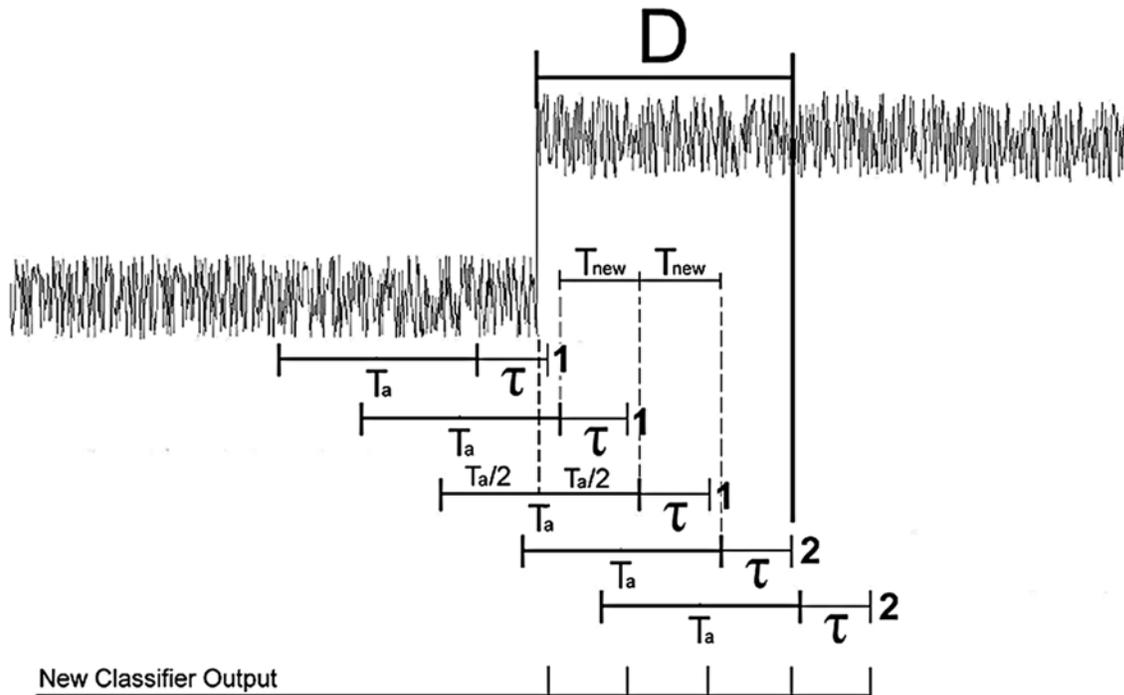


Figure 5.

Maximum controller delay (D) is shown for system that employs overlapped windows with no majority voting. Output of controller is shown as large number to far right of each window. Note that frequency of decisions indicated by tick marks on bottom of figure is increased when compared with **Figures 3** and **4**.

window (T_a). The difference between T_a and T_{new} is determined by the features that are being extracted for analysis, number of channels, pattern recognition algorithm, etc. From my experience, a general rule of thumb is that, using modern microprocessors, T_{new} will be at least an order of magnitude less than T_a . Therefore, one can conclude that using an overlapped approach will have a smaller range of controller delays and will provide the user with a more consistent response. Therefore, the more consistent response of the overlapped windows should improve prosthesis performance when compared with the nonoverlapped approach for a given set of analysis window attributes.

AN EXAMPLE

One may find that visualizing the differences between the different classifiers through an example is easier. If we use the equations presented in **Table 1**

and assume that $T_a = 150$ ms, $T_{new} = 12$ ms, $\tau = 10$ ms, and $n = 3$, we can compare the expected delays from controllers using overlapped and nonoverlapped windows as well as those using majority voting. **Table 2** shows the worst-case, average, and best-case delays as well as the theoretical range of delays for the four windowing conditions.

These examples show that using the overlapped approach substantially decreases the maximum and average delays produced by the controller for a given set of analysis window attributes. They also show that majority voting increases this delay. However, the increase due to majority voting is much more pronounced when disjoint windows are used than when overlapped windows are used. Additionally, while increasing the number of votes increases the delay for a given analysis window length, it also increases the classification accuracy [2,16,18], which improves the robustness of the system, and reduces the effect of spurious incorrect classification decisions.

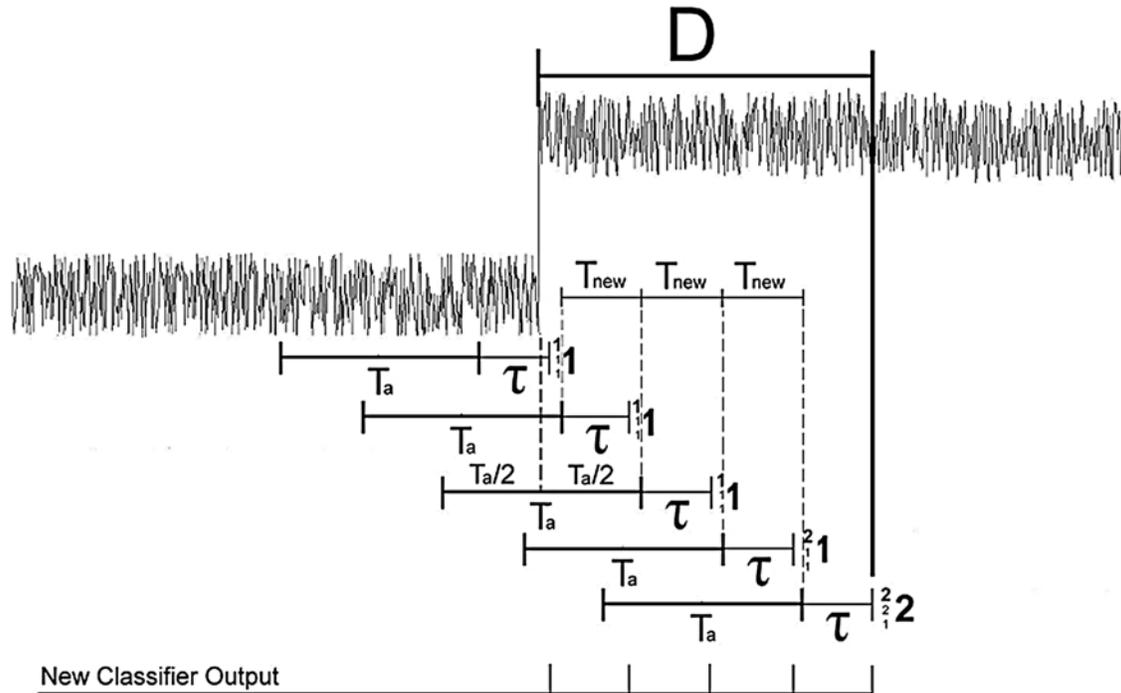


Figure 6.

Maximum controller delay (D) is shown for system that employs overlapped windows with 3-vote majority-voting scheme. Three votes are shown as small numbers to right of analysis and computation windows. Top number of these three is shown in bold and represents current class decision. Output of controller (winner of majority vote) is shown as large number to far right of each window.

Table 1.

Worse-case, average, and best-case delay equations with delay ranges.

Classifier Type	Worst-Case Delay	Average Delay	Best-Case Delay	Difference Between Best & 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}

Note: Equations to estimate worst-case, average, and best-case controller delay as well as difference between best- and worst-case controller delays. General rule of thumb is that T_{new} is approximately an order of magnitude less than T_a , meaning that users should experience more consistent delay with overlapped windows. D = maximum delay between users intended movement and controller producing correct output class, n = number of majority votes, T_a = analysis window length, τ = processing time, T_{new} = amount of window overlap.

Therefore, my colleagues and I recommend overlapped windows because they reduce the maximum controller delay, reduce range of delays experienced by the user, and limit the

increase in the controller delay introduced by majority voting. Therefore, in the examples that follow, the overlapped window strategy will be implemented.

Table 2.

Predicted best-case, average, and worst-case delays as well as delay range for four investigated classifiers. Overlapped approach is shown to substantially reduce maximum and average delays produced by controller and reduce increase in delay created by majority voting.

Classifier Type	Worst-Case Delay	Average Delay	Best-Case Delay	Delay Range
No Overlap, No Majority Voting	235	160	85	150
No Overlap, with Majority Voting	385	310	235	150
Overlap, No Majority Voting	97	91	85	12
Overlap, with Majority Voting	109	103	97	12

Note: Representative values $T_a = 150$ ms, $T_{new} = 12$ ms, $\tau = 10$ ms, and $n = 3$ votes (when applicable) were chosen for each analysis window attribute.

PRACTICAL IMPLEMENTATION

As mentioned previously, some window attributes are related. The amount of window shift (T_{new}) is related to the processing time (τ), which is based on the analysis window length (T_a) as well as the processor, memory, type of EMG features, algorithms used to extract EMG features/perform pattern recognition, and number of EMG channels. Therefore, the two variables under the designer's direct control for a given feature set are the analysis window length (T_a) and number of majority votes (n). Once the T_{new} and τ are approximated for each analysis window length and feature set, the previously discussed equations can be used to calculate the number of majority votes that can be used to produce the desired controller delay (D). Both T_a and n can be varied to arrive at a number of combinations of analysis window lengths and majority votes that will produce the desired controller delay. The controller designer should determine if any relative advantages exist when using a large window with a small number of votes or a small window with a large number of votes. However, previous work has indicated that no substantial difference exists between these two approaches with regard to classification accuracy [3,24]. This issue is discussed in more detail later.

COMPARISON TO PREVIOUS WORK

Englehart and Hudgins were one of a few groups who have examined the effect of analysis window attributes on real-time prosthesis control and used this analysis in their controller designs [3]. We greatly respect the decades of work that the group at the University of New Brunswick performed in this area, and they should be commended for their awareness of the importance of this issue and their investigation into it. In a previous article [3], they proposed that the response time of a system using majority voting would be

$$D \geq \frac{(n-1)}{2} \tau . \quad (6)$$

(For clarity, variable names have been converted to those used in this guest editorial.)

Comparing the previously reported approach (**Equation 6**) with the calculations performed using the methods described in this editorial (**Equation 5**) shows that the two methods give different estimates for the number of allowable votes (n) for a given combination of T_a , τ , and D . Note that for the data presented in **Figures 7–8**, the idealized case will be examined, in which the amount of overlap is exactly equal to the processing time (i.e., $T_{new} = \tau$).

Figure 7 compares the two methods when the number of majority votes that could be used if the acceptable delay limit was 128 ms is estimated with the use of the processing time data from Englehart and Hudgins [3]. One can observe that these two methods differ substantially, especially when the analysis window lengths increase. For example, with a 256 ms analysis window, an acceptable delay of 128 ms, and a processing time of 16 ms, the previously published work [3] would contend that 17 votes could be used, while the analysis presented in this editorial would determine that a 128 ms delay with an analysis window of that size could not be guaranteed. If the equations derived here are applied to the prior work's 17-vote estimate, the resulting predicted delay is 288 ms, which is more than double the desired delay.

The difference in the two methods results from the previously published work assuming that the

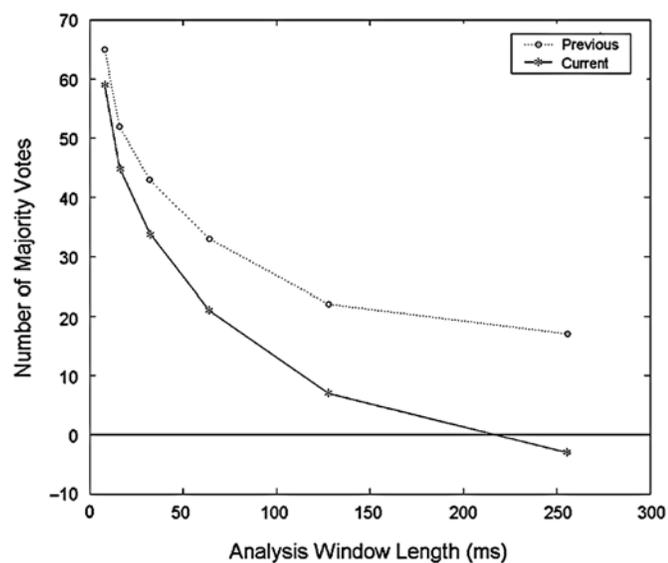


Figure 7.

Comparison of estimates of maximum number of majority votes that will create maximum controller delay of 128 ms for analysis windows of different lengths. Equations derived in this editorial give much more conservative estimates (*) than previously published work (○), especially for larger analysis windows. (Source: Englehart K, Hudgins B. A robust, real-time control scheme for multifunction myoelectric control. IEEE Trans Biomed Eng. 2003;50(7):848–54. [PMID:12848352].) A large majority of differences between two methods are result of accounting for time required to fill at least half of analysis window with data from correct class.

instant the EMG changed from Class 1 to Class 2, the classifier was producing a correct decision [3]. This editorial postulates that class changes cannot be detected instantaneously. A vast majority of the differences between the two methods results from accounting for the time required to fill at least half of the analysis window with data from the correct class (noted by Chu et al. [25]). This time to fill the analysis window is why the differences in the two methods increase with increasing analysis window length. Other differences between the two methods include the time needed to allow processing of the first analysis window that produced a correct class decision and time for an additional vote that may be required if the change in class occurs one sample after halfway through the analysis window. Accounting for these three factors should allow one to estimate more precisely the number of majority votes required to achieve a desired controller delay.

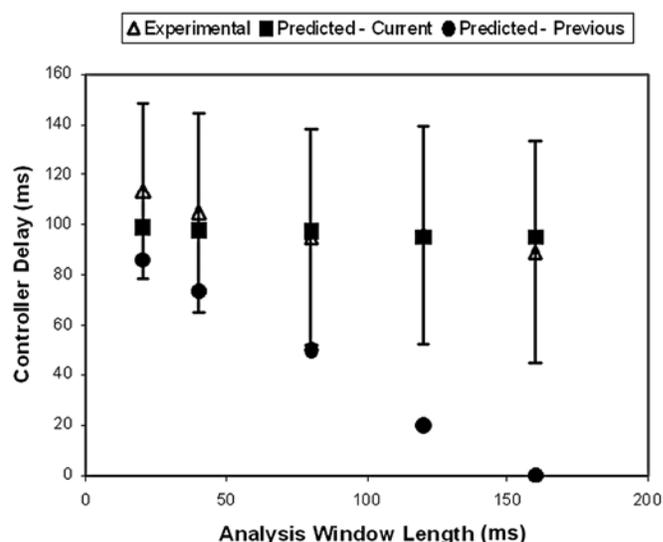


Figure 8.

Average time from transition between contraction classes and classifier producing correct output. Experimental data from 10 class transitions is shown (Δ) as well as predicted delays using both methods described in this editorial (\blacksquare) and by previous researchers (\bullet). (Error bars indicate standard deviations.) Equations described in this editorial provide better estimates of average controller delay, especially with larger analysis windows.

EQUATION VALIDATION

After deriving the equations and comparing them to a previous work, my colleagues from my previous laboratory and I collected data from a subject to try to verify that our approach was realistic. We certainly realize that “one subject does not a study make,” but we believe the results provide compelling support for the derived equations.

A linear discriminant analysis (LDA) classifier with overlapping windows performed pattern recognition on the root-mean-square 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. A subject was instructed to produce a series of 4-second-long contractions that alternated between two contraction classes. The subject produced constant extension of all four fingers for the duration of the trial and alternated between combining finger extension with either wrist relaxation or wrist extension.

These two movement classes were chosen because they occupy relatively similar portions of the feature space and therefore reduce the possibility of the feature set variances confounding the results. Two trials consisting of 10 repetitions each of the wrist extension movements were collected. The first set of 10 contractions was used to train the classifier, and the second set was used to test the classifier. Then we repeated the procedure by switching the training and testing data sets to cross-validate the results. The changes in contraction classes were determined through a visual examination of the raw EMG signal.

The EMG data were used to calculate the time from the change in the contraction class to the classifier producing the wrist extension class output for five different analysis window length and majority vote combinations (**Figure 8**). The combinations of analysis window lengths and majority votes were chosen with use of the previously derived equations, and each of these combinations was chosen to keep the controller delay below the 100 ms value (i.e., worst case is 100 ms) that Farrell and Weir [19] and Smith et al. [20] determined. Analysis window lengths of 160, 120, 80, 40, and 20 ms were investigated along with their corresponding window shifts/processing times and number of majority votes. **Figure 8** also contains the estimated controller delays determined by the equation described in this editorial (**Equation 5**) as well as estimates using the previously published method (**Equation 6**).

Figure 8 shows that, for this specific example, methods described here (■) more accurately predicted the experimentally observed average prosthesis controller delay (△). These equations predicted the controller delay more accurately than the previously published method (●), especially for larger analysis window lengths. The wide standard deviations (error bars) shown in **Figure 8** indicate that a substantial amount of variability was found across the 10 contractions. Our analysis did predict some variability in the controller delay; however, these predicted values were much lower than the observed data.

The increased variability in the delay was likely primarily due to the variability in the author's estimate of the timing of the class change. The variability in the class change estimate likely resulted from the

transitory portions between the contraction classes not being nearly as consistent as the idealized step change between the EMG classes used to derive the equations. Even if the transition between classes had been included in the equations derived in this editorial, the estimate of the time of the change in class would have been a rigid one (e.g., the transition occurs halfway between the two steady state contraction classes) that could not have accounted for the fact that real EMG data are stochastic and their variability is difficult to model reliably. Also, the increase in variability could be a result of the real data being much less separated in the feature space than the theoretical data that were used to derive the equations. However, the equations appear to accurately predict the experimentally observed average controller delay.

EFFECT OF WINDOW SIZE ON CLASSIFICATION ACCURACY

The equations derived here were shown to accurately predict the classifier delay that will be introduced into the system. However, a more important topic may be whether these combinations of analysis window lengths and majority votes produce different classification accuracies.

The effect of analysis window length on classification accuracy was found to depend not only on analysis window length but also on other variables as well. Previous research conducted by Farrell [24] and Farrell and Weir [26] has shown that the best combination of window length and number of votes can depend on the type of features extracted from the EMG signal. The impetus for this prior work was to examine the effect of electrode targeting and implantation on classification accuracy. Classification accuracies were calculated for several different types of classifiers as subjects produced 12 different hand and wrist movements. These classifiers included those that used EMG data from different types of electrodes, analysis window/majority vote combinations, feature sets, pattern recognition mechanisms, and training data sets. Farrell [24] and Farrell and Weir [26] provide details of the methodology of these experiments.

As part of this previous analysis, we calculated classification accuracies with the use of an LDA classifier using EMG data from each of four electrode conditions for the five analysis window/majority vote conditions shown in **Figure 8**. Since each of the four electrode conditions had similar trends, the average classification accuracies for the different window sizes using (1) only EMG amplitude metrics and (2) EMG amplitude measures combined with autoregressive and time-domain features are shown in **Figure 9**.

Results showed that higher classification accuracies were obtained with the longer analysis windows/smaller number of majority votes when additional EMG features were combined with amplitude measures (right side of **Figure 9**). This trend makes intuitive sense because the autoregressive models become more robust when they are created on a larger number of EMG samples. A repeated measures analysis of variance (ANOVA) conducted on the data shown on the right side of **Figure 9** found that the 160 ms analysis window condition was statistically superior to all the other conditions. (Even though the standard error bars are relatively large, repeated measures ANOVAs can find small but consistent differences between conditions.)

Alternately, when the amplitude measurements were not complimented by autoregressive and time-domain features, higher accuracies tended to be seen with smaller analysis windows (20 and 40 ms) and larger numbers of votes. For example, a repeated measures ANOVA conducted on the data shown on the left side of **Figure 9** indicated that the 40 ms analysis window length was statistically superior to the 80, 120, and 160 ms windows but statistically equivalent to the 20 ms window.

I must stress that while the differences between the analysis window lengths are statistically significant, the change in accuracy is *quite small*. While a 1 percent change in accuracy may be statistically significant, whether this difference will have a notable effect on the user's clinical performance is unclear. Previous studies have shown that prosthesis performance is more than just a function of classification accuracy [27–29]. Therefore, based upon these data sets, my colleagues and I believe that the analysis window length/majority vote combination

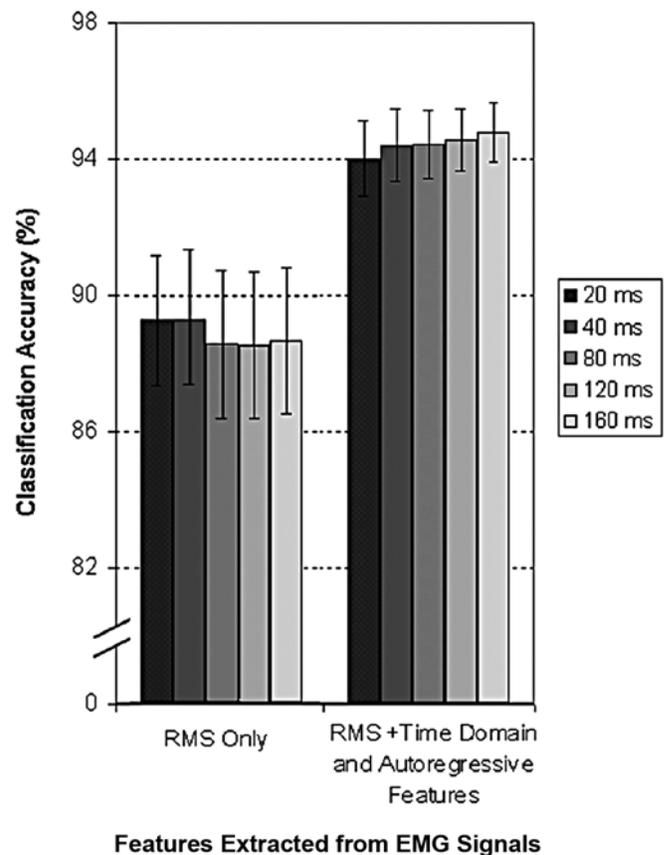


Figure 9.

Classification accuracies for each analysis window length/majority vote combination used to calculate data shown in **Figure 8** for average of 4 different electrode conditions as users performed 12 different movement classes. Left cluster of plots shows accuracies when classifier uses only electromyographic (EMG) amplitude metrics, while right shows accuracies resulting from amplitude measures along with time-domain and autoregressive signal features. (Source: Park KS. Effects of network characteristics and information sharing on human performance in COVE [master's thesis]. [Chicago (IL)]: University of Illinois at Chicago; 1997.) Note that when additional signal features are used, longer windows produce higher classification accuracy, while smaller windows perform better when only EMG amplitude metrics are used. Note: Nonzero y-axis is shown for clarity, and error bars show standard error. RMS = root-mean-square.

may ultimately have a relatively small effect on how the prosthesis performs during actual use.

These data demonstrate that different feature sets play a role in which analysis window length/majority vote combination produces the highest classification accuracy. Additionally, different classifiers (i.e., classifiers other than the LDA used in these examples) could possibly produce different optimal window lengths. These observations indicate that the appropriate

combinations of analysis window length/number of majority votes for a given desired delay can be determined with the equations in this editorial, but the combination of window length and majority votes that produces the highest classification accuracy should be determined empirically for each classifier.

DISCUSSION

The findings presented in this editorial demonstrated that the delays present in a multifunctional prosthesis controller depend on a number of variables. The findings also demonstrated that it is essential to consider the length of the time required to fill the analysis window in addition to the other window parameters for one to accurately estimate the controller delay.

Two potential limitations of the presented equations result from assumptions made to make the derivation of the equations more tractable: (1) the transition between the contraction classes was instantaneous and (2) half of the analysis window needed to be filled with data from the new contraction class before the output of that window could be recognized as belonging to the new class. Obviously, an instantaneous change in contraction class does not occur, but the validation experiment showed this to be a reasonable first-order approximation that produced accurate estimates of the controller delay.

Additionally, while the assumption appears reasonable, it may not be true that half of the analysis window needs to be filled with data from the new contraction class before that window is recognized by the classifier as belonging to the new class. The LDA classifier that was used in the validation experiments was trained on strictly steady state data, and no threshold was set for “how well” the output of a particular data window needed to match the training data from a particular contraction class. That is, there is no “default to off” for ambiguous classifications; the output of the classifier was simply the class which best matched the current window. In this case, to assume that half of the window needs to be filled with data from the new class makes sense, but this may not hold for other classifiers.

For example, if classifiers are heavily biased toward the “off” class to attempt to avoid unintentional

movements of the prosthesis, more than half of the analysis window may need to be filled with data from the new class so that the classifier is “sure” that a particular window of data belongs to a contraction class. Alternately, the off-class data may possibly be clustered tightly around the origin of the feature space. Therefore, any small deviation from the origin may cause a particular window to be recognized as belonging to a contraction class. This scenario may require less than half of the analysis window to be filled with data from the new class because filling only a small percentage of the window with nonrest data may cause the features to deviate from the origin of the feature space enough to be recognized as belonging to a contraction class. Both of these scenarios seem possible, which indicates that the controller delay may be affected by different characteristics of the classifier in addition to the analysis window attributes.

Equations that robustly and accurately relate analysis window attributes to the induced controller delay would be useful for the designers of prosthesis controllers. While my colleagues and I believe that the equations presented in this editorial predict these delays better than previously published predictors of controller delay, it is only a first step. We believe that further study of the relationships between different control strategies and the delays imparted to real-time systems is essential.

CONCLUSIONS

Many researchers have made decisions about analysis window attributes with little regard for the overall delay created in the real-time system, and these delays should be considered and discussed in each article that is published on this topic. In this vein, this editorial provides 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 output decision for one type of controller. Equations were derived for systems that used overlapped versus nonoverlapped windows as well as those that did or did not implement majority voting. My colleagues and I recommend using an

overlapped window approach because it produces a more consistent controller delay and decreases the maximum delay produced by the controller. Additionally, majority voting was shown to increase the controller delay; however, since majority voting can increase classification accuracy, it may be worth the cost of the increased delay.

A preliminary validation was performed so that the derived equations could be verified to more accurately estimate the controller delay than previous efforts. The delay from a class change to the classifier output decision was shown not to be instantaneous. The delay is not simply a function of how fast the controller produces a new output but instead is a function of the length of the analysis window, the signal processing time, and the number of majority votes used.

ACKNOWLEDGMENTS

Funding/Support: This material was based on work supported in part by the National Institutes of Health under grant 1 R01 EB01672-01 and by the Department of Veterans Affairs (VA) Rehabilitation Research and Development Service and was administered through the Jesse Brown VA Medical Center, Chicago, Illinois. This work was also partially funded by the National Institute on Disability and Rehabilitation Research of the Department of Education under grant H133E030030.

Additional Contributions: I would like to thank Dr. Richard Weir for his years of mentorship and assistance throughout my training in general and this project in particular, and I would like to acknowledge Dr. Andrew Hansen for his useful contributions to this work.

Disclaimer: The opinions in this article are those of the grantee and do not necessarily reflect those of the Department of Education.

Todd R. Farrell, PhD

Senior Research Engineer, Liberating Technologies, Inc; Holliston, MA

Email: todd.farrell@liberatingtech.com

REFERENCES

1. Englehart K, Hudgins B, Parker PA. A wavelet-based continuous classification scheme for multifunction myoelectric control. *IEEE Trans Biomed Eng.* 2001;48(3):302–11. [\[PMID: 11327498\]](#)
[DOI:10.1109/10.914793](#)
2. Englehart K, Hudgins B, Chan AD. Continuous multifunction myoelectric control using pattern recognition. *Technol Disabil.* 2003;15(2):95–103.
3. Englehart K, Hudgins B. A robust, real-time control scheme for multifunction myoelectric control. *IEEE Trans Biomed Eng.* 2003;50(7):848–54.
[\[PMID: 12848352\]](#)
[DOI:10.1109/TBME.2003.813539](#)
4. Gallant PJ, Morin EL, Peppard LE. Feature-based classification of myoelectric signals using artificial neural networks. *Med Biol Eng Comput.* 1998;36(4):485–89. [\[PMID: 10198534\]](#)
[DOI:10.1007/BF02523219](#)
5. Chang CG, Kang WJ, Luh JJ, Cheng CK, Lai JS, Chen JJ, Kuo TS. Real-time implementation of electromyogram pattern recognition as a control command of man-machine interface. *Med Eng Phys.* 1996;18(7):529–37. [\[PMID: 8892237\]](#)
[DOI:10.1016/1350-4533\(96\)00006-9](#)
6. Micera S, Sabatini AM, Dario P. On automatic identification of upper-limb movements using small-sized training sets of EMG signals. *Med Eng Phys.* 2000;22(8):527–33. [\[PMID: 11182577\]](#)
[DOI:10.1016/S1350-4533\(00\)00069-2](#)
7. Hudgins B, Parker P, Scott RN. A new strategy for multifunction myoelectric control. *IEEE Trans Biomed Eng.* 1993;40(1):82–94. [\[PMID: 8468080\]](#)
[DOI:10.1109/10.204774](#)
8. Zardoshti-Kermani M, Wheeler B, Badie K, Hashemi R. EMG feature selection for movement control of a cybernetic arm. *Cybernet Syst.* 1995;26:189–210.
[DOI:10.1080/01969729508927496](#)
9. Peleg D, Braiman E, Yom-Tov E, Inbar GF. Classification of finger activation for use in a robotic arm. *IEEE Trans Neural Syst Rehabil Eng.* 2002;10(4):290–93. [\[PMID: 12611366\]](#)
[DOI:10.1109/TNSRE.2002.80631](#)
10. Isaacs RE, Weber DJ, Schwartz AB. Work toward real-time control of a cortical neural prosthesis. *IEEE Trans Rehabil Eng.* 2000;8(2):196–98.
[\[PMID: 10896185\]](#)
[DOI:10.1109/86.847814](#)
11. Serruya M, Hatsopoulos N, Fellows M, Paninski L, Donoghue J. Robustness of neuroprosthetic decoding algorithms. *Biol Cybern.* 2003;88(3):219–28.

- [PMID: 12647229]
[DOI:10.1007/s00422-002-0374-6]
12. Serruya MD, Hatsopoulos NG, Paninski L, Fellows MR, Donoghue JP. Instant neural control of a movement signal. *Nature*. 2002;416(6877):141–42. [PMID: 11894084] [DOI:10.1038/416141a]
 13. Krusienski DJ, Schalk G, McFarland DJ, Wolpaw JR. A mu-rhythm matched filter for continuous control of a brain-computer interface. *IEEE Trans Biomed Eng*. 2007;54(2):273–80. [PMID: 17278584] [DOI:10.1109/TBME.2006.886661]
 14. McFarland DJ, Wolpaw JR. Sensorimotor rhythm-based brain-computer interface (BCI): Feature selection by regression improves performance. *IEEE Trans Neural Syst Rehabil Eng*. 2005;13(3):372–79. [PMID: 16200760] [DOI:10.1109/TNSRE.2005.848627]
 15. 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;52(11):1801–11. [PMID: 16285383] [DOI:10.1109/TBME.2005.856295]
 16. Yamada M, Niwa N, Uchiyama A. Evaluation of a multifunctional hand prosthesis system using EMG controlled animation. *IEEE Trans Biomed Eng*. 1983;30(11):759–63. [PMID: 6662533] [DOI:10.1109/TBME.1983.325192]
 17. Englehart K, Hudgins B, Stevenson M, Parker PA. Classification of myoelectric signal burst patterns using a dynamic neural network. *Proceedings of the 21st Annual IEEE Northeast Bioengineering Conference*; 1995 May 22–23; Bar Harbor, ME. Los Alamitos (CA): IEEE; 1995. p. 63–64. [DOI:10.1109/NEBC.1995.513734]
 18. Simon AM, Hargrove LJ, Lock BA, Kuiken TA. A strategy for minimizing the effect of misclassifications during real time pattern recognition myoelectric control. *Conf Proc IEEE Eng Med Biol Soc*. 2009;2009:1327–30. [PMID: 19964513]
 19. Farrell TR, Weir RF. The optimal controller delay for myoelectric prostheses. *IEEE Trans Neural Syst Rehabil Eng*. 2007;15(1):111–18. [PMID: 17436883] [DOI:10.1109/TNSRE.2007.891391]
 20. Smith LH, Hargrove LJ, Lock BA, Kuiken TA. Determining the optimal window length for pattern recognition-based myoelectric control: Balancing the competing effects of classification error and controller delay. *IEEE Trans Neural Syst Rehabil Eng*. 2011;19(2):186–92. [PMID: 21193383] [DOI:10.1109/TNSRE.2010.2100828]
 21. Hoffman DS, Strick PL. Step-tracking movements of the wrist in humans. II. EMG analysis. *J Neurosci*. 1990;10(1):142–52. [PMID: 2299389]
 22. Park KS. Effects of network characteristics and information sharing on human performance in COVE [master's thesis]. [Chicago (IL)]: University of Illinois at Chicago; 1997.
 23. Park KS, Kenyon RV. Effects of network characteristics on human performance in a collaborative virtual environment. *Proceedings of the IEEE Virtual Reality Conference*; 1999 Mar 13–17; Houston, TX. 1999. p. 104–11.
 24. Farrell T. Multifunctional prosthesis control: The effects of targeting surface and intramuscular electrodes on classification accuracy and the effect of controller delay on prosthesis performance [dissertation]. [Evanston (IL)]: Northwestern University; 2007.
 25. Chu JK, Moon I, Kim SK, Mun MS. Control of multifunction myoelectric hand using a real-time pattern recognition. *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*. Los Alamitos (CA): IEEE; 2005. p. 3511–16.
 26. Farrell T, Weir RF. A comparison of the effects of electrode implantation and targeting on pattern classification accuracy for prosthesis control. *IEEE Trans Biomed Eng*. 2008;55(9):2198–2211. [PMID: 18713689] [DOI:10.1109/TBME.2008.923917]
 27. Hargrove L, Losier Y, Lock B, Englehart K, Hudgins B. A real-time pattern recognition based myoelectric control usability study implemented in a virtual environment. *Conf Proc IEEE Eng Med Biol Soc*. 2007;2007:4842–45. [PMID: 18003090]
 28. Lock BA. Design and interactive assessment of continuous multifunction myoelectric control systems [master's thesis]. [New Brunswick (Canada)]: University of New Brunswick; 2005.
 29. Hargrove LJ, Scheme EJ, Englehart KB, Hudgins BS. Multiple binary classifications via linear discriminant analysis for improved controllability of a powered prosthesis. *IEEE Trans Neural Syst Rehabil Eng*. 2010;18(1):49–57. [PMID: 20071277] [DOI:10.1109/TNSRE.2009.2039590]

This article and any supplementary material should be cited as follows:

Farrell TR. Determining delay created by multifunctional prosthesis controllers. *J Rehabil Res Dev.* 2011; 48(6):xxi–xxxviii.

DOI:10.1682/JRRD.2011.03.0055

ResearcherID: Todd R. Farrell, PhD: C-9124-2011

