Target Prediction for Icon Clicking by Athetoid Persons

Kevin C. Olds, Sara Sibenaller, Rory A. Cooper, Dan Ding, and Cameron Riviere

Abstract—We present an investigation into modeling of athetoid motion and prediction of user intent, for use in assistive computer interfaces during icon-clicking tasks. Data were recorded from three athetoid patients during unassisted icon-clicking trials with an isometric joystick. In order to facilitate development and testing of filter designs without the difficulty of repeated testing with human subjects, a quantitative model of the recorded patient data was developed using pseudoinverse methods. Using this model within the visuomotor control loop for the icon-clicking task, a prediction filter was then developed to reduce the target acquisition time. The filter is based on a novel "autoregressive stretching window" model which selects five data points evenly distributed across the input and output histories to predict the intended target, together with a second-order system that smoothes the movement of the cursor. On average, the filter demonstrated a reduction of target acquisition time by a factor of 2.7 in experiments with the patient models.

I. INTRODUCTION

ATHETOID cerebral palsy can greatly impair control of hand movement, making many everyday tasks difficult or impossible. Interaction with computers is a particularly important problem, since computers have become a fundamental occupational tool, and standard interface devices such as mice, keyboards, and joysticks generally require high motor function. Isometric or force-sensing joysticks [1, 2] may be easier for persons with athetosis to use [3]. However, even in this case, cursor control can be painfully slow: athetosis both reduces the bandwidth of purposeful movement and corrupts the remaining bandwidth with involuntary movement at the same frequencies [4].

A variety of movement disorders exists. In the case of pathological tremor, numerous linear and nonlinear approaches have been implemented as input filters for computer interfaces [5-7]. However, such filtering approaches are not particularly suited to the present case.

Targeting accuracy is of course crucial, but it may already be adequate: in a previous study involving an icon-clicking task, subjects generally succeeded in clicking the intended icon, given sufficient time [8]. To improve the productivity of persons with athetosis, reduction of the time required for target acquisition is needed. Hence, prediction seems to be the greatest need for this application, rather than smoothing the path to the target.

This paper describes recent progress toward the development of an appropriate prediction method for icon-clicking by athetoid users. Data for development of the method were acquired in an icon-clicking experiment completed by three athetoid subjects [8]. The subjects were asked to use an isometric joystick to move a cursor and click a target circle on a computer screen. The locations of the starting points and target circles were random. A click was registered if the cursor remained inside the target circle for two seconds. Because the closed control loop via the vision of the user is an integral part of such a task, and due to the cost and inconvenience of retesting filter algorithms with athetoid subjects, we developed a model of each athetoid subject using the data from [8]. Using this model, a prediction method using a novel "stretching average" approach has been developed. This paper describes the development of the models and the predictor, and presents the results of closed-loop icon-clicking simulations using the predictor with the models obtained from the three athetoid subjects.

II. PATIENT MODEL

The suitability of linear systems for modeling the response of healthy human subjects during pursuit tracking tasks with a joystick has been established previously [9]. Given the experimental data for three athetoid subjects collected in [8] (one female, age 24, and two males, ages 38 and 49), we established a model based on the following state vector, $\mathbf{z}[n]$:

$$\mathbf{z}[n] = \begin{bmatrix} x[n] \\ y[n] \\ \dot{x}[n] \\ \dot{y}[n] \end{bmatrix}, \qquad (1)$$

where the states are the position and velocity of the cursor in x and y, measured relative to the target location. The first step in creating the patient model was to identify the system matrix $\mathbf{M}$ of the linear system shown in (2) from

the patient data.

$$\mathbf{z}[n+1] = \mathbf{M} \cdot \mathbf{z}[n] \qquad (2)$$

Since the data sets were small, we solved for $\mathbf{M}$ using the pseudoinverse. The matrix of inputs, $\mathbf{IN}$, and corresponding outputs, $\mathbf{OUT}$, for each of the N time steps of the data, shown in (3) and (4), which are related by (5), were used in order to solve for $\mathbf{M}$ as shown in (6).

$$\mathbf{IN} = [\mathbf{z}[1] \quad \ldots \quad \mathbf{z}[N-1]] \qquad (3)$$
$$\mathbf{OUT} = [\mathbf{z}[2] \quad \ldots \quad \mathbf{z}[N]] \qquad (4)$$
$$\mathbf{M} \cdot \mathbf{IN} = \mathbf{OUT} \qquad (5)$$
$$\mathbf{M} = \mathbf{OUT} \cdot \mathbf{IN}^{+} \qquad (6)$$

This method produced a solution for $\mathbf{M}$ that yielded the minimum norm least squares error when mapping all the inputs in $\mathbf{IN}$ to all the outputs in $\mathbf{OUT}$.

The error vector, $\mathbf{e}[n]$, which is the error between the output of the linear model and the actual patient data, was then computed as follows.

$$\mathbf{e}[n+1] = \mathbf{z}[n+1] - \mathbf{M} \cdot \mathbf{z}[n] \qquad (7)$$

In models such as [9], in addition to the linear transfer function of the human controller, it is common to include an additive noise which is also considered part of the human response. This is particularly appropriate in the case of athetosis, which is known to exhibit a large stochastic component [4]. Given the existing literature in spectral quantification of athetotic movement [10], we used a frequency domain approach based on a noise-shaping filter to simulate the random component of the subjects' movements in the recorded data for use as additive noise in the overall model. The Yule-Walker parametric power spectrum estimation method was used to generate a white-noise-shaping filter, $\mathbf{f}[n]$, which would output a signal with approximately the same power spectrum as $\mathbf{e}[n]$. We chose the order of the noise-shaping filter for each coordinate of each patient by choosing the lowest order for which the filter power spectrum is inside the 95% confidence interval of the power spectrum of $\mathbf{e}[n]$. This resulted in a filter of third order for patient 1, second order for patient 3, and for patient 2, fifth order in x and third order in y. Since the velocities in our data were derived from position data using a forward difference, the first two rows of $\mathbf{e}[n]$ (the x and y position) have almost no error, so we only characterized the power spectrum of the errors in the velocities. Thus, the output of the human model at time $n+1$ is given by (8).

$$\mathbf{z}[n+1] = \mathbf{M} \cdot \mathbf{z}[n] + \mathbf{f}[n+1] \qquad (8)$$

The joystick that was used in the experiments features a dead zone, so this was incorporated this into the model as well, using the simple rule that if (9) is true, then we apply (10), and likewise for $y[n]$.

$$|x[n] - x[n+1]| < 1 \qquad (9)$$
$$x[n+1] = x[n] \qquad (10)$$

The stochastic noise in athetoid movement has been found to be positively correlated with the amount of force exerted by the subject [11, 12]. Therefore, we also included a scale factor for the $\mathbf{f}[n+1]$ term in (8), which is linearly related to the norm of $\mathbf{z}[n]$, as shown in (11).

$$\mathbf{z}[n+1] = \mathbf{M} \cdot \mathbf{z}[n] + a \cdot (\|\mathbf{z}[n]\| + b) \cdot \mathbf{f}[n+1] \qquad (11)$$

The constants a and b were tuned by trial and error so that the time to target (click) and the click success rate of the model matched those of the real patients, and the model trajectories qualitatively resembled those of the patients. The results are summarized in Table I, which compares the performance of the model to that of the real patients. The columns marked "real" indicate the performance of the patient in the 100 trials in [8]. The model performance in the table represents 1000 trials using the same starting points and target locations as for the patients; i.e., 10 simulated trials were generated for each of the original pairs of start and end points.

TABLE I
AVERAGE TARGET ACQUISITION TIMES AND SUCCESS RATES FOR MODELS AND PATIENTS

Patient	Target Acquisition Time (s)		Success Rate	
	Real	Model	Real	Model
1	12.5	13.3	85%	86%
2	9.3	9.4	98%	97%
3	6.7	6.6	100%	100%

The data for each patient included 100 trials. The model data involved 1000 runs, using the same start/endpoints as the actual patient runs.

As expected, the model trajectories for patients 2 and 3 were rather similar to the patient trajectories; those for patient 1 were less so, since the recorded data from patient 1 displayed more evidently nonlinear behavior. Fig. 1 shows a recorded trajectory from patient 3, together with the corresponding model trajectory. Since the model is stochastic, no two trajectories will be the same, but a certain qualitative similarity can be seen.
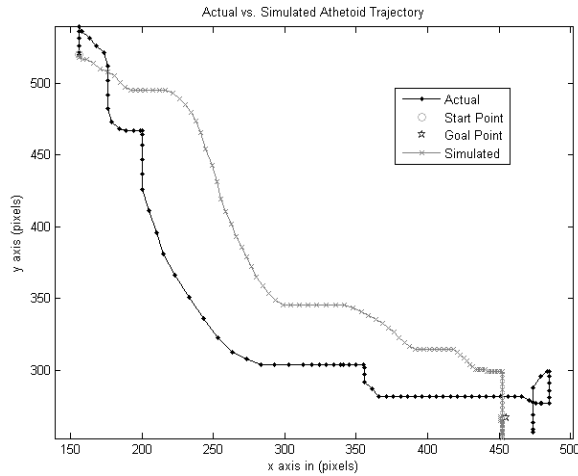
Fig. 1. One of patient three's trajectories along with the model's simulated trajectory. Each dot on the trajectory indicates a single time step, which is 0.05s.

III. FILTER DEVELOPMENT

The athetoid patient model described above allows development and testing of the filter in a realistic closed-loop environment. The goal of the filter is to predict the intended target and make the patient's target acquisition as fast as possible. A two-stage filter was developed: the first stage predicts the intended goal, and the second stage smoothes the approach to the goal somewhat in order to avoid jumpy cursor movement, which may be disconcerting for the user. Fig. 2 shows the overall block diagram of the system.
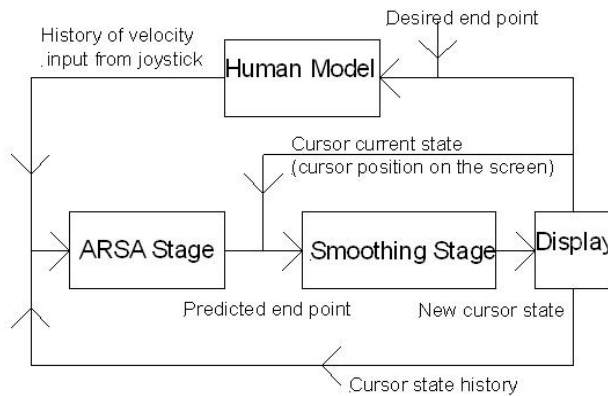

Fig. 2. Overall block diagram of the system, including both the human model and the two stages of the filter. (The ARSA and Smoothing Stages accept multiple inputs; when arrows merge this does not represent summation.)

A. The ARSA Stage

The purpose of the ARSA stage is to predict the user's intended goal. Preliminary experimentation made it clear that a moving average system with a relatively short window would not work well, since longer-term trends, such as the general direction of the target, are important for the task in question. On the other hand, a method that utilizes all the data samples at every time step is problematic due to the time-varying number of inputs and

the increased computational complexity. Our solution to this trade-off is the "auto-regressive stretching average" (ARSA) filter. Instead of a standard moving window, this filter takes as inputs the first and last samples in the input and output time histories, as well as some number of samples linearly distributed in between, and outputs a linear combination of these. However, since this filter is in discrete time, it is not always possible to have the filter inputs exactly equally distributed between the first and last samples. Therefore we used the pattern depicted in Fig. 3 to determine what samples would be taken as inputs.
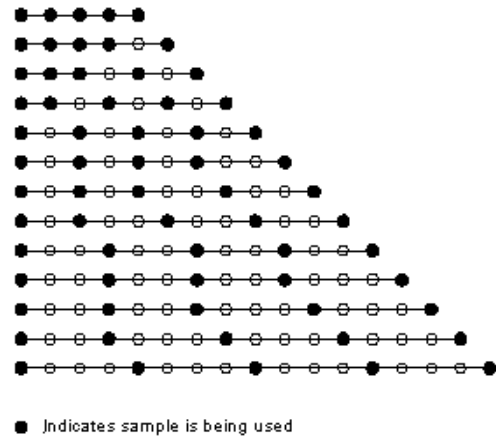

Fig. 3. Depiction of ARSA filter input sampling as time advances. Each row shows all of the samples up to that time, with time increasing to the right. As time moves forward (moving down through the rows), different samples are used (black dots).

To further clarify the concept, Fig. 4 shows a third-order ARSA system in direct form II. We found empirically that an ARSA filter of length 5 (as in Fig. 3) worked well for this application. After some experimentation, we found that it worked well to let the system first run with the filter off for 9 time steps, and then begin filter operation with the tenth sample.
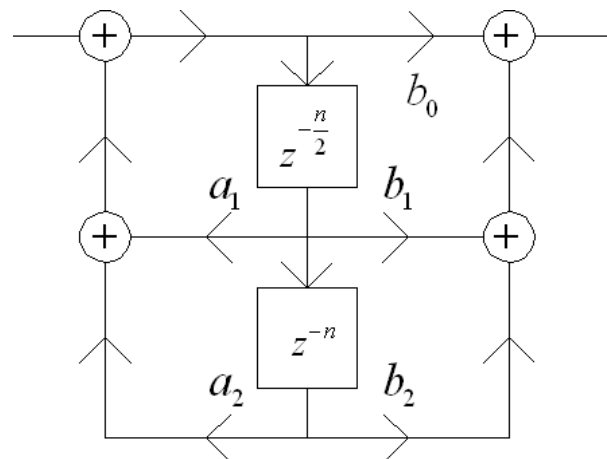

Fig. 4. A length three ARSA filter shown in direct form II. Input on the left, output on the right. The ARSA filter differs from the classical ARMA filter in that the delays increase linearly with time.

For simplicity, the set of 5 filter weights was trained once for a given patient model, and then held constant. First,

define $\mathbf{W}[n]$ to be the combination of the cursor state and the joystick inputs $j_x[n]$ and $j_y[n]$, shown in (12).

$$\mathbf{W}[n] = \begin{bmatrix} \mathbf{z}[n] \\ j_x[n] \\ j_y[n] \end{bmatrix} \tag{12}$$

Let $n_0 = 0$, n_1, n_2, n_3, $n_4 = n$ be the correct five values of n chosen according to the pattern in Fig. 3. We then define $\mathbf{S}[n]$, the total state of the ARSA system, which consists of the five $\mathbf{W}$ vectors at the correct n values, shown in (13).

$$\mathbf{S}[n] = \begin{bmatrix} \mathbf{W}[0] \\ \mathbf{W}[n_1] \\ \mathbf{W}[n_2] \\ \mathbf{W}[n_3] \\ \mathbf{W}[n] \end{bmatrix} \tag{13}$$

Each $\mathbf{S}[n]$ is a single input to the ARSA system. Since each $\mathbf{S}[n]$ has 30 elements, and we desire a two-element point,

$$\mathbf{p}[n] = \begin{bmatrix} x[n] \\ y[n] \end{bmatrix}, \tag{14}$$

as output, the ARSA filter weight matrix, $\mathbf{A}$, must be 2 × 30. This is too many elements to tune by hand. However, this problem is quite similar to the problem of identifying the system matrix for the patient model; the pseudoinverse method seems attractive once again. The problem is that this time, the input and output matrices are not clearly defined, since there are no pre-generated data as there were before. It is also impossible to simply run the filter with the human model and generate data, since the resulting data depend on the running of the closed-loop system of which $\mathbf{A}$ is a part. However, we have recourse to a simple alternative that allows us to generate a useable solution: since the only purpose of the ARSA stage is to predict the user's intended endpoint, and we know what the user's endpoint is (we have to feed it to the human model to run the simulation), we simply assume that the ARSA stage's guess will be correct, and substitute in the $\mathbf{p}$ that we know, thus cutting $\mathbf{A}$ out of the loop. This approximation allows us to run the whole system and generate an input matrix, $\mathbf{U}$, and an output matrix, $\mathbf{T}$, and then use the pseudoinverse to generate $\mathbf{A}$ as the matrix that will map $\mathbf{U}$ to $\mathbf{T}$, as shown by (15)-(18), where N is the total number of time steps we ran the system for.

$$\mathbf{U} = \begin{bmatrix} \mathbf{S}[10] \ldots \mathbf{S}[N] \end{bmatrix} \tag{15}$$

$$\mathbf{T} = \begin{bmatrix} \mathbf{p}[10] \ldots \mathbf{p}[N] \end{bmatrix} \tag{16}$$

$$\mathbf{A} \cdot \mathbf{U} = \mathbf{T} \tag{17}$$

$$\mathbf{A} = \mathbf{T} \cdot \mathbf{U}^+ \tag{18}$$

Thus, the approach attempts to train $\mathbf{A}$ such that (19) will result.

$$\mathbf{A} \cdot \mathbf{S}[n] = \mathbf{p}[n] \tag{19}$$

It is known, of course, that the subject intends to move the cursor to the target, but almost certainly he or she does not intend to reach the target in a single time step, as in (19). Thus, this approach involves a rough approximation of user intent. However, since the main goal of the work is to speed up the target acquisition as much as possible, this approximation has proved useful, as the testing results will show. Closed-loop iteration to refine $\mathbf{A}$ would be possible, but so far performance has been adequate without it. The only significant drawback is the possibility of jerky cursor movement, which has been resolved by adding a "smoothing" stage to the prediction filter.

B. The Smoothing Stage

This stage is merely a second-order model constrained to move the cursor directly toward the target at each step. At each time step, given the current cursor position and the endpoint location predicted by the ARSA stage, this stage steps the cursor toward the endpoint. It would be possible to simply have the cursor go directly to the predicted endpoint, but this would likely result in "jerky" motion that would be difficult to control. As a second-order model, the stage has simple, intuitive parameters which can be used to tune the movement to fit a patient's preferences.

First, define $\mathbf{x}[n]$ to be the current cursor position, $\mathbf{v}[n]$ to be the current cursor velocity, and $\mathbf{p}[n]$ to be the user's intended goal (the output of the ARSA stage). Define the error vector $\mathbf{r}[n]$ as

$$\mathbf{r}[n] = \mathbf{x}[n] - \mathbf{p}[n]. \tag{20}$$

Next, create a unit vector in the direction of the error vector.

$$\hat{\mathbf{r}}[n] = \frac{\mathbf{r}[n]}{\|\mathbf{r}[n]\|} \tag{21}$$

Now we can find $\mathbf{v}'[n]$, the component, $\mathbf{v}[n]$, that points in the direction of the desired endpoint.

$$\mathbf{v}'[n] = (\mathbf{v}[n] \bullet \hat{\mathbf{r}}[n]) \cdot \hat{\mathbf{r}}[n] \tag{22}$$

Now that we have position and velocity terms, we can define the "force," $\mathbf{f}[n]$, that our second-order model is exerting on the cursor. It is as though the cursor had a mass m, with a spring of constant k and damper of constant b connecting it to $\mathbf{p}[n]$.

$$\mathbf{f}[n] = -b \cdot \mathbf{v}'[n] - k \cdot \mathbf{r}[n] \tag{23}$$

Now that we have $\mathbf{f}[n]$, we can find $\mathbf{a}[n]$, the acceleration of the cursor.

$$\mathbf{a}[n] = \frac{\mathbf{f}[n]}{m} \qquad (24)$$

With $\mathbf{a}[n]$, we can use Euler integration to obtain the next position and velocity of the cursor, where h is the time step.

$$\mathbf{v}[n+1] = \mathbf{v}'[n] + h \cdot \mathbf{a}[n] \qquad (25)$$
$$\mathbf{x}[n+1] = \mathbf{x}[n] + h \cdot \mathbf{v}'[n] \qquad (26)$$

IV. FILTER TESTING

To test the effectiveness of the filter, we used it with the patient models as shown in Fig. 2. We used four different testing scenarios designed to resemble situations that commonly occur in normal computer use.

1) *Grid*. A 3x3 square grid of buttons was used, like icons on a desktop. The button the cursor started on and the desired destination were randomly chosen from the grid. In this scenario, the only way the trial could fail was if it timed out by running longer than 20 s before clicking the correct button. In all four test scenarios, a click is registered by 2 s of uninterrupted dwell time.

2) *Grid with All Icons Active*. The second test used the same grid of buttons and the same clicking criteria, but now all of the buttons were active and could be clicked. This means that if the cursor stayed in any of the buttons for two seconds, a click was registered. If an incorrect button was clicked, the run was recorded as a failure.

3) *Random*. The third test used randomly located buttons. This is the scenario that was used in the collection of the original data from the athetoid subjects [8]. The cursor started in a random location, and the desired button was randomly located on the screen. The same timeout and clicking criteria as before were applied.

4) *Random with All Icons Active*. The fourth test used the same randomly located target buttons, but this time it was possible to click anywhere on the screen where there could potentially exist a button that did not overlap the target button. If the cursor dwelled for two seconds within any such hypothetical button, then the system registered a click and the run ended in failure. Obviously, this test was by far the most difficult since it was possible to falsely click almost anywhere on the screen; this is reflected in the much lower success rate for this scenario.

The results for all these scenarios are shown in Table II. In every case the filter decreased the target acquisition time, and increased or maintained the success rate. The decrease in target acquisition time ranged from a factor of 1.7 to a factor of 4.7, depending on the patient and scenario, with the average reduction among the 9 patient/scenario combinations being a factor of 2.7. For the fourth scenario, which is arguably the most realistic, and which produced the lowest reduction factors, the average reduction in target

TABLE II
AVERAGE TARGET ACQUISITION TIMES AND SUCCESS RATES FOR MODELS
WITH AND WITHOUT FILTERING

Patient/ Test Scenario	Target Acquisition Time (s)		Success Rate	
	Model	Model with Filter	Model	Model with Filter
1-G	15.4	3.3	83%	100%
2-G	10.3	3.5	99%	100%
3-G	7.1	3.5	100%	100%
1-GA	13.6	3.4	73%	100%
2-GA	10.5	3.6	98%	100%
3-GA	7.0	3.6	100%	100%
1-R	13.7	4.0	87%	98%
2-R	9.3	3.6	98%	99%
3-R	6.7	3.4	100%	100%
1-RA	8.5	3.6	5%	92%
2-RA	6.7	3.5	35%	95%
3-RA	6.0	3.5	75%	100%

Each patient/test scenario involved 240 trials. G indicates a 3x3 grid of buttons with only the correct button active. GA indicates the same grid with all buttons active. R indicates randomly located buttons with only the correct button active. RA indicates a randomly located target button, where it is possible to click on any hypothetical button that does not overlap the target.

acquisition time was a factor of 2.0. Fig. 5 shows an example of filtered and unfiltered trajectories for a typical pair of start and end points.

V. DISCUSSION

The results demonstrate the general feasibility of the approach. In the future, in addition to matching the target acquisition time and frequency content of the raw data, we plan to evaluate the models using appropriate trajectory
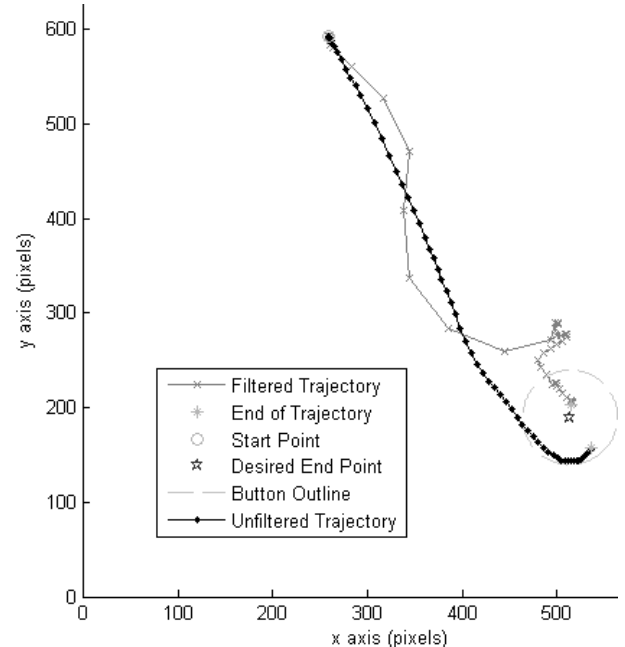

Fig. 5. Filtered and unfiltered simulated trajectories for patient three. Each dot on the trajectories marks one time step, which is 0.05 s. In this case, the unfiltered trajectory took 4.25 s, and the filtered trajectory took 2.75 s.

similarity measures [13]. This will be followed by testing of the predictor system with the human users on whom the models are based.

The characteristics of the human controller do not remain constant. It was demonstrated in [9] that a person's internal controller changes depending on the system the person is trying to control, i.e., the type of computer task. Furthermore, even if the task in which the filter is to be used is the same as the one described here, there are sure to be variations over time within any given human subject that will affect the parameters of the internal controller. The degree to which a filter trained as presented here will yield an overall closed-loop system that remains useful or even stable when used at different times or for different tasks remains a subject for further investigation.

As mentioned earlier, the filter as presented here was trained in a single step, using the approximation in (19). Iterative training is possible, however, and may result in improved performance.

REFERENCES

[1] R. A. Cooper, D. K. Jones, S. Fitzgerald, M. L. Boninger, and S. J. Albright, "Analysis of position and isometric joysticks for powered wheelchair driving," *IEEE Transactions on Biomedical Engineering,* vol. 47, pp. 902-910, 2000.

[2] R. A. Cooper, D. M. Spaeth, D. K. Jones, M. L. Boninger, S. Fitzgerald, and S. Guo, "Comparison of virtual and real electric powered wheelchair driving using a position sensing joystick and an isometric joystick," *Medical Engineering and Physics,* vol. 24, pp. 703-708, 2002.

[3] S. Sibenaller, D. Ding, B. Dicianno, R. A. Cooper, and C. Riviere, "Kinematic characteristics of athetoid movement during target acquisition," in *RESNA Annu. Conf.*, 2008, submitted.

[4] P. D. Neilson, "Voluntary control of arm movement in athetotic patients," *Journal of Neurology, Neurosurgery and Psychiatry,* vol. 37, pp. 162-170, 1974.

[5] J. G. Gonzalez, E. A. Heredia, T. Rahman, K. E. Barner, and G. R. Arce, "Optimal digital filtering for tremor suppression," *IEEE Transactions on Biomedical Engineering,* vol. 47, pp. 664-673, 2000.

[6] P. O. Riley and M. J. Rosen, "Evaluating manual control devices for those with tremor disability," *Journal of Rehabilitation Research and Development,* vol. 24, pp. 99-110, 1987.

[7] C. N. Riviere and N. V. Thakor, "Modeling and canceling tremor in human-machine interfaces," *IEEE Engineering in Medicine and Biology Magazine,* vol. 15, pp. 29-36, 1996.

[8] J. J. Vázquez López, S. Sibenaller, D. Ding, and C. Riviere, "Toward filtering of athetoid motion with neural networks," in *Proc. 28th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Lyon, France, 2007, pp. 1434-1436.

[9] B. Kuchen, K. Henning, H. Rake, and O. Schäfer, "Frequency response of the human controller for various processes and stochastic testsignals," *Biological Cybernetics,* vol. 27, pp. 33-39, 1977.

[10] S. Niku and J. M. Henderson, "Toward quantification of athetotic movements by frequency spectrum analysis," *Journal of Biomechanics,* vol. 18, pp. 71-76, 1985.

[11] P. D. Neilson, "Measurement of involuntary arm movement in athetotic patients," *Journal of Neurology, Neurosurgery and Psychiatry,* vol. 37, pp. 171-177, 1974.

[12] S. Niku and J. M. Henderson, "Determination of the parameters for an athetotic arm model," *Journal of Biomechanics,* vol. 18, pp. 209-215, 1985.

[13] M. Vlachos, G. Kollios, and D. Gunopulos, "Discovering similar multidimensional trajectories," in *Proc. 18th Int. Conf. Data Eng.*, San Jose, Cal., 2002, pp. 673-684.