

# Adaptive auto-regressive proportional myoelectric control

Carles Igual, Jorge Igual, Janne M. Hahne and Lucas C. Parra

**Abstract**—In proportional myographic control one can control either position or velocity of movement. Here we propose to use adaptive auto-regressive filters, so as to gradually adjust between the two. We implemented this in an adaptive system with closed-loop feedback, where both the user and the machine simultaneously attempt to follow a cursor on a two-dimensional arena. We tested this on 15 able-bodied and three limb-deficient participants using an 8-channel myoelectric armband. The human-machine pairs learn to perform smoother cursor movements with a larger range of motion when using the auto-regressive filters, as compared to our previous efforts with moving-average filters. Importantly, the human-machine system converges to an approximate velocity control strategy resulting in faster and more accurate movements with less muscle effort. The method is not specific to myoelectric control and could be used equally well for motion control using high-dimensional signals from reinnervated muscles or direct brain recordings.

**Index Terms**—Adaptive linear filtering, Recursive Least Squares, Electromyography, Control, Prosthesis.

## I. INTRODUCTION

Electromyographic (EMG) signals are small electric potentials generated during muscle contractions [1]. They can be measured non-invasively on the surface of the skin. As their amplitude increases with increasing muscle force, EMG signals can be utilized for proportional control. In rehabilitation this is successfully used to control electrically powered hand and arm prostheses from EMG-signals of the residual muscles [2]. In conventional myoprostheses, two bipolar EMG-signals are placed on antagonistic muscle-groups, such as the wrist extensors and flexors, and are used to control the velocity of one degree of freedom (DOF) [2]. Extending this concept to more DOFs is usually not directly possible because typically not enough independent control signals are available. In commercially available prostheses, cumbersome switching concepts are used to control multiple functions sequentially.

Research efforts over the past decade have extracted more complex control information of a larger number of EMG-sensors with machine learning techniques [3]–[6]. Most work focused on classification-based approaches, which in its original form were still restricted to sequential on/off control of each individual function. Extensions of this work allow for a proportional control [7] and combined activation of multiple functions [8], but the highest flexibility is obtained by a continuous mapping of EMG features into control signals using regression techniques [9]–[14].

A challenge in most mapping algorithms is obtaining reliable labels for supervised training. While in able-bodied individuals the

kinematics [11], [12] or forces [9], of the actual limb can be measured, in prosthetic end-users this is not possible. One approach is to perform bilateral mirrored contractions, but this implies certain errors and is limited to unilateral amputations [15]. Alternatively, one may rely on visual cues given to the participants as targets under the assumption that participants can reliably follow these cues [16]. However, “blindly” generating consistent muscle contractions is difficult and so here we provide real-time visual feedback to help participants follow a desired target movement. In this approach, both the user and the learning algorithm attempt to follow a common target, whereby humans adjust muscle force in real-time and the machine simultaneously adapts its control parameters. As a result, the human and the machine can in principle concurrently adapt to converge to a common control strategy.

The present work follows our previous efforts to learn continuous movement control in 2D [12], while providing closed-loop, real-time feedback to the user [17]. We ask able-bodied participants to generate muscle contractions that result in 2D wrist movement (wrist flexion/extension, ulnar/radial deviation). Myoelectric activity is recorded from the forearm by a wearable armband with 8 channels. This activity is then used to predict an intended movement target on a circular arena shown on a computer screen (see Figure 2). Previously, we have used linear regression to predict location from instantaneous EMG-amplitudes. With such linear proportional control, stronger muscle contractions lead to larger cursor displacements, i.e., muscle contractions control the position of the target [12].

In position control, the position is maintained as long as the user maintains the muscle contraction. This can be tiring and would quickly cause fatigue when holding objects. Therefore, in most commercial prosthetic devices, the velocity is controlled proportionally to the EMG amplitude instead, i.e., the strength of the contraction controls the speed of movement. If the user relaxes, the prosthesis remains in the current position and an antagonistic contraction is required to revert the movement. However, because it is difficult to visually estimate and replicate the velocity of an object, training of regression algorithms by visual cues are typically done in a position control mode. In the current work we present a novel, more general control concept, that is not restricted to either position control or velocity control. The algorithm is capable of incorporating both control schemes including intermediate combinations of both. The goal is for the control strategy to emerge naturally from a closed-loop interaction of the human and machine, rather than imposing position or velocity control arbitrarily through the design.

## II. AUTO-REGRESSIVE APPROACH

Our approach is to explicitly use the current position to predict the next intended location. This leads to an auto-regressive predictor that is more flexible than either position or velocity control. To clarify the importance of using an auto-regressive filter consider the following. Denote the 2-dimensional position that we would like to control as  $\mathbf{y}(t)$ , and the  $M$ -dimensional myoelectric control signal as  $\mathbf{x}(t)$  (typically related to the EMG signal power). In its simplest form, proportional controls implies  $\mathbf{y}(t) = \mathbf{B}\mathbf{x}(t)$  (ordinary linear regression), which is what we implemented in [12]. To implement velocity control, the input has to be able to modify the difference of

Manuscript received June 21, 2018. This work is partially supported by Ministerio de Educacion, Cultura y Deporte (Spain) under grant FPU15/02870.

Carles Igual is with the Department of Communications, Universidad Politecnica de Valencia, Valencia, Spain (e-mail: carigba@etsii.upv.es).

Jorge Igual is with the Department of Communications, Universidad Politecnica de Valencia, Valencia, Spain (e-mail: jigual@com.upv.es).

Janne M. Hahne is with the Applied Rehabilitation Technology Lab, Department of Trauma Surgery, Orthopedic Surgery and Hand Surgery, University Medical Center Göttingen, Germany (e-mail: janne.hahne@med.uni-goettingen.de)

Lucas C. Parra is with the Department of Biomedical Engineering, City College of New York, NY 10031 USA (e-mail: parra@ccny.cuny.edu)

the current from the previous position,  $\mathbf{y}(t) - \mathbf{y}(t-1) = \mathbf{B}\mathbf{x}(t)$ . In other words, we need an auto-regressive structure:  $\mathbf{y}(t) = \mathbf{y}(t-1) + \mathbf{B}\mathbf{x}(t)$ . To gradually adjust between position and velocity control we should allow for additional coefficients:  $\mathbf{y}(t) = \mathbf{A}\mathbf{y}(t-1) + \mathbf{B}\mathbf{x}(t)$ . When  $\mathbf{A} = \mathbf{0}$  we have pure position control, when  $\mathbf{A} = \mathbf{I}$  we have pure velocity control. More generally, we will allow these coefficients  $\mathbf{A}$  and  $\mathbf{B}$  to be multi-input multi-output (MIMO) filters: multiple temporal inputs are filtered in time to generate multiple outputs in time (not just instantaneous mapping). In doing so we can filter the input, for instance, to smooth the noisy fluctuations of myographic activity (with  $q$  tabs of a moving-average (MA) filter:  $\mathbf{B}_k, k = 0 \dots q$ ). With an auto-regressive (AR) filter we can take a variable history into account for computing velocity or acceleration on a variable time-scale (with  $p$  filter tabs:  $\mathbf{A}_k, k = 1 \dots p$ ). The most important aspect here is that these filters are not fixed, but instead, they should be adapted to best match the behavior of the human when presented with the task. In total, we are proposing and adaptive ARMA-MIMO system that attempts to predict the desired locations  $\mathbf{y}(t)$  recursively from the myographic signals  $\mathbf{x}(t)$ :

$$\mathbf{y}(t) = \sum_{k=1}^p \mathbf{A}_k(t)\mathbf{y}(t-k) + \sum_{k=0}^q \mathbf{B}_k(t)\mathbf{x}(t-k) \quad (1)$$

The mathematical derivations that follow are established theory of adaptive IIR filtering [18]–[20]. We reproduce this theory here to tie it into the context of myographic motor control, to motivate the choices we made among various recursive algorithms, and to provide explicit equations for implementation. Note that during training the filter matrices  $\mathbf{A}_k(t)$  and  $\mathbf{B}_k(t)$  are themselves dependent on time as they will be adjusted so that  $\mathbf{y}(t)$  matches a desired target location  $\mathbf{d}(t)$ . Figure 2 shows a snapshot of such a target  $\mathbf{d}(t)$  as a green circle and the current position  $\mathbf{y}(t)$  as a red cross. While the adaptive filter algorithm tunes filters  $\mathbf{A}_k(t)$  and  $\mathbf{B}_k(t)$ , the user is concurrently generating myoelectric signals  $\mathbf{x}(t)$  to move the red cross to the green target. Generally the user will vary  $\mathbf{x}(t)$  on a rapid time scale of less than a second, whereas the filter parameters are adjusted on a slower time scale of many seconds or minutes. The human learner can also adjust strategy of movement on this slower time scale. The concurrent learning system is expected to converge due to the common training goal (reducing the distance of current position to target) and assuming an appropriate choice of learning constants to prevent instabilities (we touch on this in more detail in the Discussion section).

For myoelectric control the feature-vector  $\mathbf{x}(t)$  will be a nonlinear function of the raw myoelectric signals, such as the root-mean-square of the broad-band signals or more complex time or frequency domain features [12]. Note that when  $p = 0$  in eq. (1), the output has a finite impulse response (FIR) and when  $p = q = 0$ , we obtain the instantaneous linear regressor proposed in [17] (although there the temporal filter was fixed to be an exponentially MA). Here we extend over this prior work by adding the auto-regressive filter  $\mathbf{A}_k$ , which results in an infinite impulse response (IIR). In all experiments we will compare the performance of the FIR system with the new adaptive IIR filter structure proposed here.

In order to simplify the problem, we will assume that  $y_1(t)$  and  $y_2(t)$  are independent; this means that the axes that determine the two wrist angles, the flexion-extension axis, and the radial-ulnar axis are independent [12]. Assuming this independence, the matrices  $\mathbf{A}_k(t)$  are diagonal, and each angle  $y_i(t)$   $i = 1, 2$  can be estimated separately from previous positions of the same DOF and input signals:

$$y_i(t) = \sum_{k=1}^p a_{i,k}(t)y_i(t-k) + \sum_{k=0}^q \mathbf{b}_{i,k}^T(t)\mathbf{x}(t-k), \quad i = 1, 2 \quad (2)$$

where  $a_{i,k}(t)$  are the corresponding diagonal entries in  $\mathbf{A}_k(t)$ , and  $\mathbf{b}_{i,k}(t)$  are the corresponding rows in the filter matrices  $\mathbf{B}_k(t)$ .

We can express equation (2) in a compact form such as:

$$y_i(t) = \beta_i^T(t)\mathbf{z}_i(t), \quad i = 1, 2 \quad (3)$$

with the coefficients vector  $\beta_i(t)$  and the data vector  $\mathbf{z}_i(t)$ , both of which are column vectors of length  $p + (q + 1)M$ , and are defined as:

$$\beta_i(t) = [a_{i,1}(t), a_{i,2}(t), \dots, a_{i,p}(t), \mathbf{b}_{i,0}(t), \mathbf{b}_{i,1}(t), \dots, \mathbf{b}_{i,q}(t)]^T \quad (4)$$

$$\mathbf{z}_i(t) = [y_i(t-1), y_i(t-2), \dots, y_i(t-p), \mathbf{x}^T(t), \dots, \mathbf{x}^T(t-q)]^T \quad (5)$$

The learning task is to find the  $\hat{\beta}_i(t)$  that minimizes the mean squared error  $\varepsilon_{MSE}(t)$  between the output of the system  $\mathbf{y}(t)$  and the desired position  $\mathbf{d}(t)$ , i.e.:

$$\varepsilon_{MSE}(t) = \sum_{i=1}^2 E \left\{ (d_i(t) - y_i(t))^2 \right\} \quad (6)$$

This is called the output-error formulation [20], since the filters are estimated using the mean squared-error of the output  $\mathbf{y}(t)$ . Once the filters  $\hat{\beta}_i(t)$  are calculated, the estimate of the current position  $\hat{\mathbf{y}}(t)$  is obtained using eq. (2) and the error can be obtained.

However, notice that with definition (5)  $\mathbf{z}(t)$  depends on the history of  $\mathbf{y}(t)$  and thus it itself depends of the parameters  $\beta(t)$ . Through this recursive dependence the error is a non-linear function of the parameters  $\beta_i$ . Nevertheless, since eq. (3) resembles a linear regression problem, it is called a pseudolinear regression. The nonlinearity implies that the cost function is not a quadratic function, so the linear estimate can be suboptimal.

### III. ADAPTIVE FILTERING OF THE EMG SIGNALS

An adaptive approach to the problem is particularly important in the context of closed-loop feedback. The user can in principle change the control strategy in real-time, and so the optimal mapping between EMG signal and target location should be able to adjust to the current control strategy. Our goal is to continuously adapt the coefficient sample by sample, instead of recalculating the coefficients with a batch of training data, and then having the user adjust to the new set of coefficients as in our previous work [17].

The main idea behind the adaptive methods is that the new estimate is obtained from the previous estimate by moving in the direction that minimizes the MSE in eq. (6). Since the negative gradient vector points in that direction, we just have to calculate the partial derivatives of the MSE with respect to the coefficients of the system.

The updating rule is:

$$\beta(t+1) = \beta(t) - \mu \nabla \varepsilon_{MSE}(t), \quad (7)$$

where  $\nabla \varepsilon_{MSE}(t)$  is the gradient and  $\mu$  the step size. The gradient requires the calculation of the expected values. Since we do not know the distributions, these expectations must be estimated. The

simplest solution is to remove the expectation operator; i.e., to use the instantaneous value, obtaining the least mean squares LMS algorithm [18]. Calculating the gradient vector, the updating rules for the coefficients in  $\beta_i$  are:

$$a_{i,k}(t+1) = a_{i,k}(t) + \mu e_i(t) \frac{\partial y_i(t)}{\partial a_{i,k}(t)}, \quad k = 1, \dots, p \quad (8)$$

$$b_{i,k}^j(t+1) = b_{i,k}^j(t) + \mu e_i(t) \frac{\partial y_i(t)}{\partial b_{i,k}^j(t)}, \quad (9)$$

$$k = 0, \dots, q; j = 1, \dots, M$$

where  $e_i(t)$  is the instantaneous error at time  $t$ , i.e.,  $e_i(t) = d_i(t) - \beta_i^T(t) \mathbf{z}_i(t)$ . The partial derivatives in the preceding equations are given by:

$$\frac{\partial y_i(t)}{\partial a_{i,k}(t)} = y_i(t-k) + \sum_{l=1}^p a_{i,l}(t) \frac{\partial y_i(t-l)}{\partial a_{i,k}(t)}, \quad (10)$$

$$k = 1, \dots, p$$

$$\frac{\partial y_i(t)}{\partial b_{i,k}^j(t)} = x_j(t-k) + \sum_{l=1}^p a_{i,l}(t) \frac{\partial y_i(t-l)}{\partial b_{i,k}^j(t)}, \quad (11)$$

$$k = 0, \dots, q; j = 1, \dots, M.$$

The second term in the right hand side of eq. (10,11) is due to the recursion model, since the estimated position at time  $t$  depends on the  $p$  previous positions, and, each of these depends on the coefficients.

In eq. (10,11) we have an additional problem. The equations are not recursive; i.e., they depend on the present value of  $a_k(t)$  and  $b_k(t)$  (do not confuse a recursive system with a recursive algorithm). If we use a small step size  $\mu$ , we can assume that the coefficients are changing slowly, so:

$$\frac{\partial y_i(t-l)}{\partial a_{i,k}(t)} \simeq \frac{\partial y_i(t-l)}{\partial a_{i,k}(t-l)} \quad (12)$$

$$\frac{\partial y_i(t-l)}{\partial b_{i,k}^j(t)} \simeq \frac{\partial y_i(t-l)}{\partial b_{i,k}^j(t-l)} \quad (13)$$

Using this approximation in eq. (10,11), we obtain:

$$\frac{\partial y_i(t)}{\partial a_{i,k}(t)} \simeq y_i(t-k) + \sum_{l=1}^p a_{i,l}(t) \frac{\partial y_i(t-l)}{\partial a_{i,k}(t-l)} \quad (14)$$

$$k = 1, \dots, p$$

$$\frac{\partial y_i(t)}{\partial b_{i,k}^j(t)} \simeq x_j(t-k) + \sum_{l=1}^p a_{i,l}(t) \frac{\partial y_i(t-l)}{\partial b_{i,k}^j(t-l)} \quad (15)$$

$$k = 0, \dots, q; j = 1, \dots, M.$$

So we can obtain the approximations of the derivatives  $\psi_{a_{i,k}}(t) = \frac{\partial y_i(t)}{\partial a_{i,k}(t)}$  and  $\psi_{b_{i,k}^j}(t) = \frac{\partial y_i(t)}{\partial b_{i,k}^j(t)}$  in a recursive calculation:

$$\psi_{a_{i,k}}(t) = y_i(t-k) + \sum_{l=1}^p a_{i,l}(t) \psi_{a_{i,k}}(t-l) \quad (16)$$

$$k = 1, \dots, p$$

$$\psi_{b_{i,k}^j}(t) = x_j(t-k) + \sum_{l=1}^p a_{i,l}(t) \psi_{b_{i,k}^j}(t-l) \quad (17)$$

$$k = 0, \dots, q; j = 1, \dots, M$$

Note that the derivatives in eq. (16,17) are delayed versions of  $y_i(t)$  and  $x_j(t)$  filtered by the time-varying recursive filter  $a_{i,k}(t)$ .

We call this the IIR LMS algorithm. This algorithm estimates  $p + q \times M - 1$  parallel filters at every iteration; this requires a lot of

storage and computational resources. With the assumption that the step size  $\mu$  is small, we can obtain a simplified IIR LMS algorithm. Since the coefficients  $a_{i,k}(t)$  do not vary too much in intervals of length  $p$ ,  $a_{i,k}(t) \simeq a_{i,k}(t-1) \simeq \dots \simeq a_{i,k}(t-p)$ , we can assume that they are time invariant in that period, and we can exchange the order of filtering and delay operations in eq. (16,17). It means that we can first filter the input and output signals for  $k = 1$  and  $k = 0$ , respectively,

$$\tilde{y}_i(t) = \psi_{a_{i,1}}(t) \quad (18)$$

$$\tilde{x}_i^j(t) = \psi_{b_{i,0}^j}(t) \quad (19)$$

and, then, approximate the other elements in the gradient vectors as delayed versions of them. This is called the filtered IIR LMS algorithm. It requires only  $M + 1$  filters to approximate the derivatives.

The LMS algorithm updates the parameters according to the gradient of the instantaneous squared-error (a stochastic gradient descent method). Another option is to use the recursive Gauss-Newton RGN algorithm that improves the convergence rate using sample covariance matrices to control the direction during the updating step. The algorithm is more complicated, since in every iteration the inverse of the covariance matrix must be also updated. Nevertheless, the inversion of the matrix is avoided thanks to the matrix inversion lemma, reducing the computational cost [19].

The general updating rule for the RGN algorithm for the equation and output error formulations is:

$$\beta_i(t+1) = \beta_i(t) + \mu \mathbf{P}_i(t+1) \tilde{\mathbf{z}}_i(t) e_i(t), \quad (20)$$

where  $\tilde{\mathbf{z}}_i(t)$  is the filtered version of the data vector,  $e_i(t)$  is the error with the current coefficients  $e_i(t) = d_i(t) - \beta_i^T(t) \mathbf{z}_i(t)$ , and  $\mathbf{P}_i^{-1}(t)$  an estimate of the Hessian matrix that is updated by:

$$\mathbf{P}_i^{-1}(t+1) = \lambda \mathbf{P}_i^{-1}(t) + \gamma \tilde{\mathbf{z}}_i(t) \tilde{\mathbf{z}}_i^T(t) \quad (21)$$

with  $\lambda$  the forgetting factor that controls the weight of previous values in the current estimate. Typical values (see [20]) are  $\lambda = 0.9, \dots, 1$ , and  $\lambda = 1 - \mu$ ,  $\gamma = 1$ .

The inversion of the matrix is avoided using the matrix inversion lemma, and the updating rule becomes:

$$\mathbf{P}_i(t+1) = \lambda^{-1} \left( \mathbf{P}_i(t) - \frac{\mathbf{P}_i(t) \tilde{\mathbf{z}}_i(t) \tilde{\mathbf{z}}_i^T(t) \mathbf{P}_i(t)}{\lambda/\gamma + \tilde{\mathbf{z}}_i^T(t) \mathbf{P}_i(t) \tilde{\mathbf{z}}_i(t)} \right) \quad (22)$$

The difference with the LMS algorithm is due to the  $\mathbf{P}_i(t)$  matrices. So, if these matrices are equal to the identity matrix, both algorithms are the same and depending on the equation or output-error formulation that is chosen, we get the LMS algorithms explained previously.

The implementation of the filtered IIR RGN algorithm is summarized as follows:

Initialization,

Iteration ( $t = 0, 1, \dots$ ),

for  $i = 1, 2$ :

Error estimate:

$$y_i(t) = \beta_i^T \mathbf{z}_i(t) \quad (23)$$

$$e_i(t) = d_i(t) - y_i(t) \quad (24)$$

Filter signals:

$$\tilde{y}_i(t) = y_i(t) + \sum_{l=1}^p a_{i,l}(t) \tilde{y}_i(t-l) \quad (25)$$

$$\tilde{x}_i^j(t) = x_j(t) + \sum_{l=1}^p a_{i,l}(t) \tilde{x}_i^j(t-l) \quad (26)$$

$$\tilde{\mathbf{z}}_i(t) = [\tilde{y}_i(t-1), \tilde{y}_i(t-2), \dots, \tilde{y}_i(t-p), \tilde{\mathbf{x}}^T(t), \dots, \tilde{\mathbf{x}}^T(t-q)]^T \quad (27)$$

Update:

$$\mathbf{P}_i(t+1) = \lambda^{-1} \left( \mathbf{P}_i(t) - \frac{\mathbf{P}_i(t)\tilde{\mathbf{z}}_i(t)\tilde{\mathbf{z}}_i^T(t)\mathbf{P}_i(t)}{\lambda/\gamma + \tilde{\mathbf{z}}_i^T(t)\mathbf{P}_i(t)\tilde{\mathbf{z}}_i(t)} \right) \quad (28)$$

$$\beta_i(t+1) = \beta_i(t) + \mu\mathbf{P}_i(t+1)\hat{\mathbf{z}}_i(t)e_i(t) \quad (29)$$

Updates of the AR coefficients are only executed if they lead to stable recursion (i.e. the poles of the AR coefficients remain within the unit circle).

Note that if we remove the filtering step, we obtain a pseudolinear regression algorithm. If we substitute the output signal by the observed output, i.e., we follow an equation-error instead of the output-error formulation,  $p = q = 0$  (there is no feedback nor memory in the system) and  $\mu = \gamma = 1$ , the algorithm is the same that the exponentially weighted recursive least squares algorithm. The cost function is a modification of the least-squares cost function, by incorporating a forgetting factor  $\lambda$  so recent samples are weighted more strongly in the error computation:

$$e_{ERLS}(T) = \sum_{t=1}^T \lambda^{T-t} |d(t) - y(t)|^2 \quad (30)$$

Note that eq. (30) is the cost function used in [17], i.e., the recursive version of the linear regression solution. The algorithm in [17] is adaptive, but the system is not recursive, so there is no feedback between the past and present positions, nor between the past EMGs observations and the current position. The exponential factor allows to obtain a weighted covariance matrix in order to accommodate the time varying nature of the signals.

## IV. EXPERIMENTAL PARADIGM

### A. Data Acquisition

We use a Myo Armband from Thalmics to acquire the EMG signals. It has a flexible diameter to fit a forearm circumference between 7.5 to 13 inches. It has eight bipolar EMG electrodes and samples the EMG at 200 Hz at 8 bit resolution. This does not cover the entire EMG-spectrum and would be insufficient for more complex feature-extraction, but is sufficient for extracting simple amplitude features. Signals are transferred to the computer via Bluetooth. A Matlab program is executed in order to acquire and process the data in real-time. We use Matlab 16a 64 bit version running on a 2.6 GHz personal laptop with 8 GB RAM.

We use log-variance of the EMG signals as the input feature vector  $\mathbf{x}(t)$ , with variance computed in a time window of 200 ms and updated every 40 ms as in [12] ( $t$  is sampled at 25 Hz). The sensors are placed in the same position and orientation for all participants (upper part of the forearm, close to the elbow with the LED light of the device pointing to the same direction). Fifteen able-bodied participants were tested (8 males, 7 females) with ages ranging between 20 to 50 years. We also tested two male individuals and a female with limb deficiency; one amputee (with a 20 cm stump; Figure 1) and two congenital (with a 10 cm and a 25 cm stump respectively). All individuals provided written informed consent before the experiment. The experiments were in accordance with the declaration of Helsinki and were approved by the UPV ethics committee, approval number P11-23-03-18.

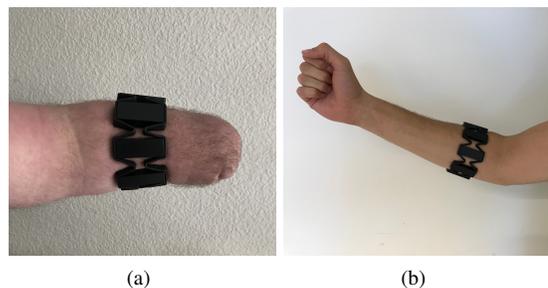


Fig. 1. (a) Amputee participant using the Myo Armband. (b) Able-bodied participant using the Myo Armband.

### B. Study Design

Participants sat in a comfortable position in front of a computer screen, with the elbow of the arm resting on the table and flexed by nearly 90 degrees. The able-bodied ones were instructed to relax the hand so that forearm activity was only dedicated to wrist motions. After the armband was placed on the forearm of the participant and connected to the computer, the device was initialized. The experiment consisted of a training and testing phase as follows.

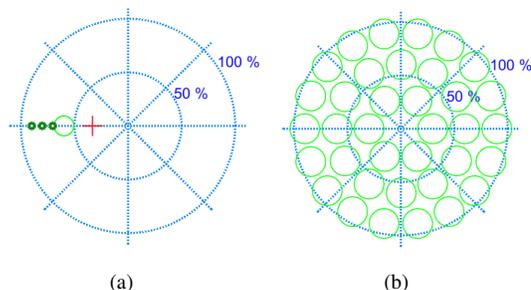


Fig. 2. (a) User-feedback screen during training experiment. Red cross: estimated position. Large green circle: current target position. Small green circles anticipate future target positions. (b) The 36 targets placed in 3 different radii (0.3, 0.6, 0.9) for the test experiment. There are 6 targets in the inner circle, 12 in the intermediate and 18 in the outer circle.

1) **Training phase:** Figure 2(a) shows the user-screen during training. The center of the coordinate system corresponds to the neutral position and the two axes to the two DOFs controlled in this study. The green circle indicates the desired target position  $\mathbf{d}(t)$  and the red cross is the current estimated position  $\mathbf{y}(t)$ . This display is updated at the same 25 Hz rate (40 ms) as the adaptive filter equations. Before the start of the training phase participants are familiarized with the closed-loop user feedback shown in Figure 2(a).

**Training of the IIR system:** The three small circles indicate the direction of movement for the upcoming target positions. This helps the user prepare for the upcoming movement and maintain muscle contractions synchronized with the desired target locations. We defined a set of simple trajectories in the flexion-extension (right-left) and radial-ulnar (up-down) axes that were paced with a constant speed that participants could easily follow without significant delays. The target starts in the center. Then it moves during 6 seconds from center to the right side. Once it reaches the right-most position, it returns back to the center in 6 seconds without stopping at any location. The same is done for the up, left and down directions. This four movement directions defines a lap. The training experiment consists on five consecutive laps, totaling 240 s of training. The order and time of these target movements are identical across participants. The users were instructed to move their wrist so that the red cursor

follows the green circle. For example, in Figure 2(a), participant would have to conduct a left movement with his/her wrist. They were instructed not to worry even by larger deviations from the target, but to try their best and remain focused on the task. For the able-bodied individuals it was easy for the experimenter to monitor their effort. During the training phase the  $\mathbf{A}_k$  and  $\mathbf{B}_k$  parameters are continuously adjusted using the filtered IIR RGN algorithm explained in the previous section. Here we used  $p = 1, q = 0$ , i.e. we used the immediately preceding position and the instantaneous input (without temporal filtering).

**Training of the FIR system:** As comparison to the new recursive IIR algorithm we test a FIR structure with the same training procedure as in [17]. Briefly, a target circle is first shown at the center to start at a neutral position. When the red cross is at the center a target appears at one of the outer-most positions, directly to the right, up, left or down. The participant has to move the forearm till it arrives at this target. The users have 20 seconds to hit the target, after which the next target appears. After reaching the target and maintaining the red cross inside the green circle for 1 s (a hit) the target circle jumps back to the center. If the target is not hit within 20 seconds, it also jumps back to the center. The process repeats with the other three directions. This is considered one lap and total training consists of 5 repeated laps. During the training phase the  $\mathbf{B}_k$  parameters of the algorithm are continuously adjusted using the FIR algorithm in [17]. That algorithm is the same as the one in our model in eq. (1) with  $p = 0, q = 0$ , i.e. no recursive feedback of position is used, but the input is filtered, and an additional post-processing step (an exponential moving average filter) is used to smoothen the output (see [17] for the details).

**2) Test phase:** The test phase starts after the 5 training runs were completed and is identical for the IIR and FIR systems. During the test-phase the parameters of the algorithm were kept constant. In the test phase, the goal for the participants is to move the red cursor to various target locations (indicated by the green circle) and maintain this location for at least one second. The task was similar as in the training FIR phase, targets were static. If the target was reached in less than 20 seconds and maintained for 1 second it was counted as a hit, otherwise it was counted as a missed target. After each hit or miss, a new target was shown at a new pseudo-random location. Eventually a total of 36 uniformly distributed target positions were presented with no repetition (Figure 2(b)). The targets were shown in the same pseudo-random order for all participants. Note that the targets include regions that have not been explored during training. Therefore, we are also testing the ability of the algorithm to generalize and to avoid over-fitting.

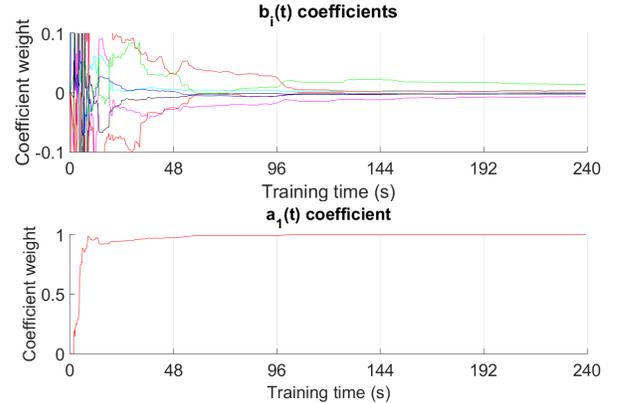
To control the effects of fatigue and practice, we divided participants into two groups. One group first trains and tests with the FIR system and after that with the IIR system. The other group does the opposite. We set the learning constants to  $\mu = 1$  and  $\lambda = 1$  during training so the algorithm is effectively integrating across all samples.

## V. RESULTS

### A. Real-time adaptation during training

To gain a sense for the speed of adaptation of the system during closed-loop training, we first show the learning process for one user in Figure 3. The panels show the FIR and IIR coefficients ( $b_i$  and  $a_1$ ) for the flexion-extension direction as they adapt in time. In this example, the FIR coefficients start to converge after the first lap (48 seconds), while the IIR coefficient is learned in just 15 seconds. In Figure 4 we show the position and error during training. The first two panels show the moving target (black) and the cursor position (color) generated by two users during training. The position error

increases whenever the target moves ahead of the users response, but decays over the total duration of the training session. The same trend is observed for the instantaneous error averaged over the 15 participants as shown in the third panel. A repeated measures ANOVA on the mean position error of the five repeated training laps shows that position error is reduced over time ( $F(14) = 32.9, p = 3 \cdot 10^{-7}$ , with time coded as a continuous predictor variable). In particular, there is a reduction of error between the last two laps (paired t-test,  $t(14) = 3.2, p = 0.007$ ).



**Fig. 3.** Parameter adaptation during training. The first row shows the eight FIR coefficients  $b_i$  as they develop in time during training for one representative able-bodied participant. The second row is the IIR coefficient  $a_1$  in that same time period. Coefficients are shown here for the flexion-extension direction only. Note that the IIR coefficient converges almost immediately to  $a_1 = 1$ , which corresponds to velocity control. Results are similar for other participants and in radial-ular direction. Gray vertical lines indicate start/end of the five repeated training laps with identical target trajectories.

### B. Performance gains of IIR system during the test phase

We observed during the test phase that participants produced smoother trajectories and wider range of movement when using the IIR system. In contrast, the FIR system required more force and some participants had a limited range of movement in some directions.

To quantify performance numerically we measured the completion rate, defined as the number of targets hit (reached within 20 seconds and held for 1 second) over the total number of targets (36). To evaluate the efficiency and ease-of-use we also calculated the path-length during the entire test session, including hits and misses. Secondary outcome measures were path efficiency (path length over shortest distance to target, averaged over all targets), completion time (average time to a target, missed targets count as 20 seconds), and attempt ratio (number of times entering the target area over number of hits).

Test phase performance of the closed-loop learning algorithm was evaluated for all participants with both FIR and IIR algorithms. Figure 5 shows that the rate of targets hit increases in all but one participant by addition of the auto-regressive filter structure (IIR). At the same time, the length of the path to reach the targets is reduced for all participants. A Wilcoxon signed rank test shows that difference in rate of hits as well as in path length are statistically significant ( $p = 0.005$  and  $p = 0.00006$ , respectively with  $N = 15$ ).

With the IIR method, 95% of the targets were hit, with perfect performance for half the users (no targets were missed). Of the 5% that were missed, 90% were missed due to the 20 seconds time limitation. A color map of the interface is shown in Figure 6. Only one position was missed by three out of fifteen participants (orange

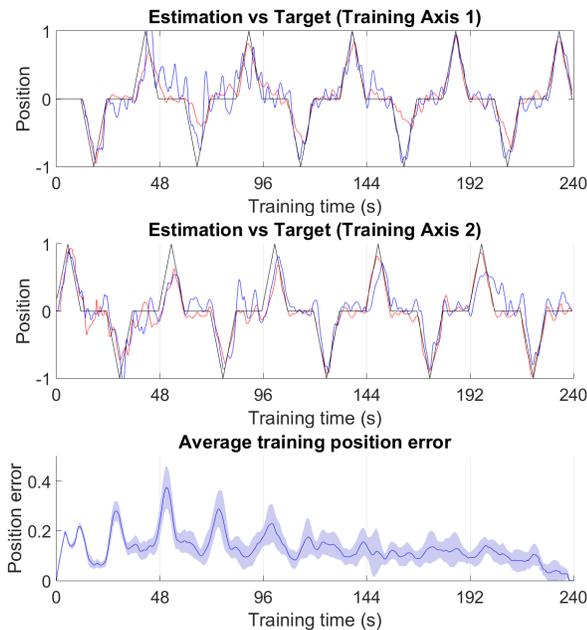


Fig. 4. Performance during training. First and second row indicate the instantaneous position of the user-machine system during training for both axes. Able-bodied participant (red), a participant with limb deficiency (blue) and target (black). The third row shows the instantaneous error during training averaged over all 15 able-bodied participants. Shaded area indicates standard error of the mean across participants. Gray vertical lines indicate start/end of the five repeated training laps with identical target trajectories.

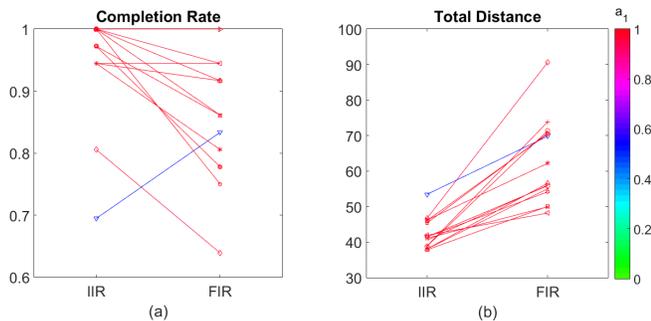


Fig. 5. Test phase performance in able-bodied individuals. Each line is a participant. Completion rate increases or stays the same for IIR structure. Average path length to reach a target is shortened for all 15 participants when using the IIR structure. Line color indicates the value  $\alpha_1$  learned by the IIR filter (in the IIR condition) for one axis. Evidently almost all participants learned velocity control, i.e.  $\alpha_1 \approx 1$ . There is a single exception with  $\alpha_1 = 0.54$  (the blue line).

circle) while many targets were hit by everyone (electric green circles) and some of them were missed by one or two participants (clear green circles). For the FIR algorithm, hit rate was around 85% and there are targets that were missed by 3 people (orange), 4 (light red) and 5 or more times (dark red). Only one participant got a perfect score. The FIR errors were located in the outer positions indicating the FIR learning algorithm is limited in gain. This could be the result of a non-linear relation between EMG and displacement, whereby the linear gain  $b_0$  is suitable in the inner range, but not for larger displacements. In contrast, the IIR system has no range limitation as the cursor can in principle continue moving if the user/machine pair learned velocity control, i.e.,  $a_1 = 1$ .

The results on the secondary outcome measures are shown in Figure 7. Path efficiency is improved similarly to overall path-

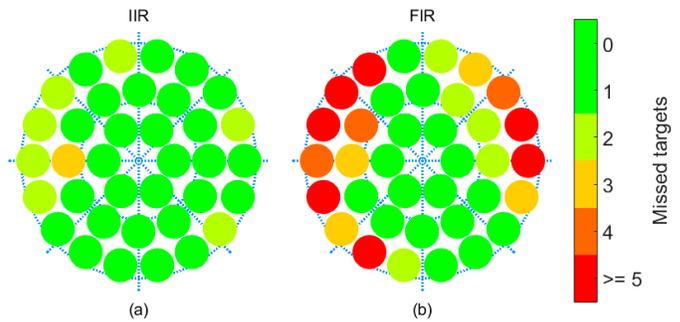


Fig. 6. Number of missed targets over all 15 able-bodied participants during the test phase.

length, as it is the same measure except that it is normalized by the shortest length to reach a target. Completion time is faster for most participants when using the IIR system (11 out of 15). In general, movement was slower and better controlled, which explains why some participants did not gain in speed despite improving on all other measures. Attempt ratio captures the ability to reach and maintain the target for the prerequisite 1 second. A ratio of 1 indicates that the participant never exited the target area prematurely. Evidently, the IIR system allows significantly better control to hold the position, despite velocity control. To visualize the difference in path efficiency, we plot in Figure 8 the trajectories of one participant for both algorithms.

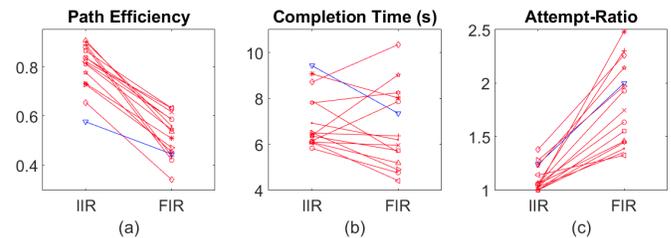


Fig. 7. Secondary outcome measures for able-bodied individuals during test phase. Each line/symbol is a participant. IIR structure improves the path efficiency and reduced the number of attempts needed to hit a target for all users. Completion time does give mixed results. Same color map used as in Figure 5.

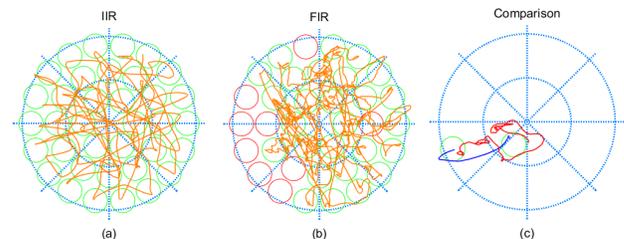


Fig. 8. Trajectories following the 36 targets during the test phase for one user. The IIR trajectories are smoother and shorter than for the FIR filter. This participant does not reach some areas in the FIR case. The right panel compares the trajectory from one target to the next for another participant. The blue line is the IIR method; the red line is the FIR algorithm. The IIR trajectory (blue curve) is smoother than the FIR trajectory (red curve) and takes less time and effort.

In Figure 8 we also compare the paths from an inner to an outer target achieved with both methods (blue: IIR, red: FIR). With both algorithms, the user was able to hit the outer target, but the FIR path is much more erratic and clearly shows the effort that was required. As an example of the user experience, we show the last seconds of the

testing phase for the FIR and IIR algorithms for the same participant in the video [21].

### C. Results on participants with limb deficiency

Able-bodied individuals rely on actual wrists movements during training and testing. To demonstrate that this is not required with the closed-loop feedback, we recruited three individuals with limb deficiency (Figure 1). The identical training and testing was used as before. These individuals relied purely on visual feedback on the screen to guide their muscle contractions. Performance is numerically lower in these individuals, as compared to able-bodied participants. We ascribe this to the lower EMG signal strength we observed in these participants, in particular for the individual with a shorter stump.

All performance metrics showed a performance benefit with the IIR system (Figure 9). The number of hits increases, the total path length, is reduced almost by 40% and the efficiency of the trajectories increases accordingly. The attempt ratio is improved in all participants and is close to 1 with the IIR. This indicates that when the target was reached, it was easy to maintain the position. When inspecting the coefficients found by the IIR algorithm we find again that the  $a_1 \approx 1$ , meaning that this human-machine pair again learned velocity control.

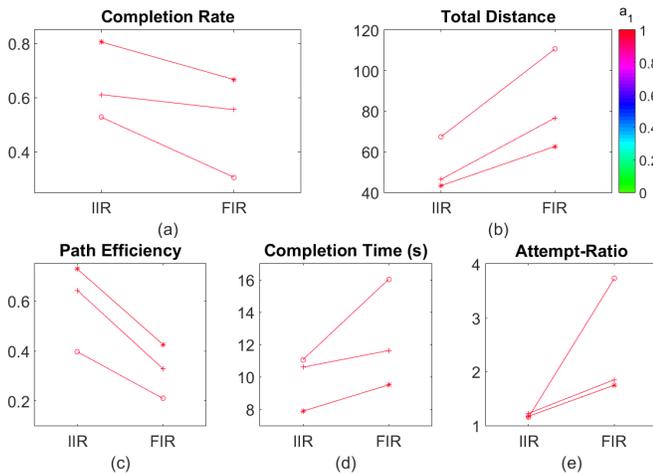


Fig. 9. Performance metrics for the participants with limb deficiencies. Same metrics as in Figure 5 and Figure 7. Same color map is used also for  $a_1$  values.

## VI. DISCUSSION

We have demonstrated here the benefits of learning recursive filters for proportional myographic control. The recursive filter allowed us to seamlessly titrate between position and velocity control. Given these options, the human-machine system naturally converged to a velocity-control strategy. Velocity-control is known to have a number of benefits in practice [22] such as less overall effort for the user and no limitations of the range of motion (e.g. due to fatigue). However, to our knowledge no adaptive strategy has yet been proposed to directly learn proportional velocity control, and our own previous efforts had been limited to position control [17]. By introducing a recursive structure we were able to readily incorporate more general control strategies into a closed-loop learning mechanism. To do this we leverage established theory of adaptive IIR filtering [18]–[20]. The novelty lies in relating this theory to myoelectric control, which allows for a gradual transition between position and velocity control and an efficient closed-loop training procedure without the need for manual parameter adjustments.

In the case of the FIR filter, we only learned instantaneous regression coefficients  $\mathbf{b}_0$  and used a post-processing filter to smooth the output signals, following [17]. In the IIR approach proposed in this paper, this post-processing is not necessary since it is implicitly implemented in the output recursion. In addition, the human-machine pair naturally learned a velocity control strategy. Since we are using a model with only one recursion and instantaneous input coefficients, this limited number of free parameters allowed us to learn the coefficients with relatively short training session of a few minutes.

Another practical benefit is that velocity or gain-factors do not need to be adjusted manually as in other approaches [14], since all factors are learned during the process of real-time, closed-loop adaptation.

An important caveat of this work is that we have tested the system on a somewhat artificial 2D cursor movement task and that we have focused mostly on able-bodied individuals. The preliminary results with limb deficient participants are nevertheless encouraging. In these individuals, myographic signals are typically weaker and electrodes more difficult to place. Despite overall lower performance we find that the adaptive IIR filter still shows an improvement in this target group over to the FIR filter. Larger number of participants have to be tested to determine if these results can be replicated across the more diverse physiology in this group. Similarly, the proposed strategy should be tested on a realistic motor control task, ideally using an actual prosthetic device, e.g. [14], [23].

Here we used a closed-loop learning system where in principle the human and machine can simultaneously adapt. In contrast to [17], parameter adaptation is ongoing during the entire closed-loop training period. The learning rules proved stable in practice despite concerns that such concurrent adaptations can become unstable [24]. Note that in the present work we did not quantify or demonstrate human adaptation (as e.g. done in [25]) and so discussion of co-adaptive learning are theoretical in nature. The issue of stability of co-adaptive learning has been studied previously on a theoretical level [26]. The main observation of that work is that stable co-adaptation may be achieved as long as one learner adapts slower than the other. In the example shown in Figure 3, adaptation of the controller happened relatively quickly (15-50 seconds). This is evidently slower than the time constant of human motion control ( $<1$  second), but faster than the time constant a human may use for adapting movement strategy. This means that the machine is slow enough to allow the participants to control the specific movement trajectory, yet fast enough to adapt to the strategy the user is trying to implement. From the controllers standpoint, the human is quasi-stationary, and from the humans standpoint, the controller is quasi-stationary as well (since it stabilizes so fast). The net result is a stable system despite the closed-loop interaction of the human-machine controllers. For this stability it is necessary that the user generates consistent muscle contractions on the time scale of parameter adaptation (approximately 120 seconds here). We believe that providing clear instructions to the users at the beginning is important in this regard, as well as the game-like interface which keeps participant motivated to follow those instructions. The most important aspect of this interface is the real-time feedback coupled with a consistent goal, namely, the machine and human continuously attempt to reduce the same error.

Note that the guidance to the user can be suggestive of position control (“to move further out, try to make a larger effort”) or it can suggest velocity control (“to move in a given direction, flex the muscle and just wait for the cursor to move”). We only briefly experimented with acceleration control, which can be implemented by allowing two tap-delays  $A_1$  and  $A_2$  in eq. (1). This can combine  $\mathbf{y}(t-1)$  and  $\mathbf{y}(t-2)$  with the current position  $\mathbf{y}(t)$  to compute acceleration. Our initial findings suggest that the algorithm performs equally well, although the overall human-machine control tends to

overshoot. Admittedly, controlling acceleration may be more challenging for the user as short flexion forces are all that is needed to start and stop a ballistic movement. We can also envision controlling wrist rotation, which is common in prosthetic devices. In that case we anticipate that participants will require more careful instructions in order to consistently perform muscle contractions that were not previously associated with wrist rotation. Future work may explore this and other training protocols with concurrent human-machine learning [17], [26], and test for human adaptation following [25], [27], [28]. In particular, it may be interesting to test whether the closed-loop strategy used here with concurrent parameter adaptation leads to co-adaptive learning, and whether this results in performance gains as compared to off-line training [17], [29].

Finally, we note that nothing about the proposed approach is specific for myoelectric control. The method could be used equally well for motion control using high-dimensional signals from re-innervated muscles [30] as these behave similarly to conventional myographic signals where signal amplitude increases with effort. The approach could also be used in brain-machine interfaces [31], [32]. These capture neuronal firing directly from the motor cortex, which is known to encode both position and velocity of movement [33].

## VII. ACKNOWLEDGEMENTS

We want to thank Brian Aguilar Avila for careful proofreading of this manuscript, as well as Jonathon Sensinger and anonymous reviewers who provided insightful comments which focused the message of this paper.

## REFERENCES

- [1] R. Merletti and P. A. Parker, *Electromyography: physiology, engineering, and non-invasive applications*. John Wiley & Sons, 2004, vol. 11.
- [2] A. Muzumdar, *Powered upper limb prostheses: control, implementation and clinical application*. Springer, 2004.
- [3] P. Parker, K. Englehart, and B. Hudgins, "Myoelectric signal processing for control of powered limb prostheses," *Journal of Electromyography and Kinesiology*, vol. 16, no. 6, pp. 541–548, Dec. 2006.
- [4] M. A. Oskoei and H. Hu, "Myoelectric control systems - a survey," *Biomedical Signal Processing and Control*, vol. 2, no. 4, pp. 275–294, Oct. 2007.
- [5] E. Scheme and K. Englehart, "Electromyogram pattern recognition for control of powered upper-limb prostheses: state of the art and challenges for clinical use," *Journal of Rehabilitation Research and Development*, vol. 48, no. 6, pp. 643–659, Sep. 2011.
- [6] B. Peerdeman, D. Boere, H. Witteveen, R. Huis in 'tVeld, H. Hermens, S. Stramigioli, H. Rietman, P. Veltink, and S. Misra, "Myoelectric forearm prostheses: State of the art from a user-centered perspective," *The Journal of Rehabilitation Research and Development*, vol. 48, no. 6, p. 719, 2011.
- [7] C. Castellini and P. van der Smagt, "Surface EMG in advanced hand prosthetics," *Biological Cybernetics*, vol. 100, no. 1, pp. 35–47, 2009.
- [8] A. Young, L. Smith, E. Rouse, and L. Hargrove, "Classification of simultaneous movements using surface EMG pattern recognition," *Biomedical Engineering, IEEE Transactions on*, vol. 60, no. 5, pp. 1250–1258, May 2013.
- [9] N. Jiang, K. B. Englehart, and P. A. Parker, "Extracting simultaneous and proportional neural control information for multiple-DOF prostheses from the surface electromyographic signal," *IEEE Transactions on Bio-Medical Engineering*, vol. 56, no. 4, pp. 1070–1080, Apr. 2009.
- [10] A. Fougner, O. Stavadahl, P. Kyberd, Y. Losier, and P. Parker, "Control of upper limb prostheses: Terminology and proportional myoelectric control #x2014;a review," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 20, no. 5, pp. 663–677, 2012.
- [11] S. Muceli and D. Farina, "Simultaneous and proportional estimation of handkinematics from emg during mirrored movements at multipledegrees-of-freedom," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 20, no. 3, pp. 371–378, May 2012.
- [12] J. Hahne, F. Biessmann, N. Jiang, H. Rehbaum, D. Farina, F. Meinecke, K.-R. Müller, and L. Parra, "Linear and nonlinear regression techniques for simultaneous and proportional myoelectric control," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 22, no. 2, pp. 269–279, 2014.
- [13] A. Ameri, E. Scheme, E. Kamavuako, K. Englehart, and P. Parker, "Real-time, simultaneous myoelectric control using force and position-based training paradigms," *Biomedical Engineering, IEEE Transactions on*, vol. 61, no. 2, pp. 279–287, Feb. 2014.
- [14] J. M. Hahne, M. A. Schweisfurth, M. Koppe, and D. Farina, "Simultaneous control of multiple functions of bionic hand prostheses: Performance and robustness in end users," *Science Robotics*, vol. 3, no. 13, Jun. 2018.
- [15] J. L. Nielsen, S. Holmgard, N. Jiang, K. B. Englehart, D. Farina, and P. A. Parker, "Simultaneous and proportional force estimation for multifunction myoelectric prostheses using mirrored bilateral training," *Biomedical Engineering, IEEE Transactions on*, vol. 58, no. 3, pp. 681–688, Mar. 2011.
- [16] A. Ameri, E. N. Kamavuako, E. J. Scheme, K. B. Englehart, and P. A. Parker, "Real-time, simultaneous myoelectric control using visual target-based training paradigm," *Biomedical Signal Processing and Control*, vol. 13, pp. 8–14, Sep. 2014.
- [17] J. M. Hahne, S. Dähne, H.-J. Hwang, K.-R. Müller, and L. C. Parra, "Concurrent adaptation of human and machine improves simultaneous and proportional myoelectric control," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 23, no. 4, pp. 618–627, 2015.
- [18] B. Widrow and S. D. Stearns, *Adaptive signal processing*. Prentice-Hall, 1985.
- [19] P. S. R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation*. Springer, 2013.
- [20] J. J. Shynk, "Adaptive iir filtering," *ASSP Magazine, IEEE*, vol. 6, no. 2, pp. 4–21, 1989.
- [21] (2018) User experience video. [Online]. Available: <https://youtu.be/HH8xyOrLBWk>
- [22] E. D. Engeberg, S. G. Meek, and M. A. Minor, "Hybrid force–velocity sliding mode control of a prosthetic hand," *IEEE Transactions on Biomedical Engineering*, vol. 55, no. 5, pp. 1572–1581, 2008.
- [23] S. Amsuess, P. Goebel, B. Graimann, and D. Farina, "A multi-class proportional myocontrol algorithm for upper limb prosthesis control: Validation in real-life scenarios on amputees," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 23, no. 5, pp. 827–836, 2015.
- [24] M. Rupp and A. H. Sayed, "Robustness of gaussnewton recursive methods: A deterministic feedback analysis," *Signal Processing*, vol. 50, no. 3, pp. 165–187, 1996.
- [25] A. W. Shehata, E. J. Scheme, and J. W. Sensinger, "Evaluating internal model strength and performance of myoelectric prosthesis control strategies," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 26, no. 5, pp. 1046–1055, 2018.
- [26] J. S. Müller, C. Vidaurre, M. Schreuder, F. C. Meinecke, P. Von Büna, and K.-R. Müller, "A mathematical model for the two-learners problem," *Journal of neural engineering*, vol. 14, no. 3, p. 036005, 2017.
- [27] M. Ison, I. Vujaklija, B. Whitsell, D. Farina, and P. Artemiadis, "High-density electromyography and motor skill learning for robust long-term control of a 7-dof robot arm," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 24, no. 4, pp. 424–433, 2016.
- [28] J. M. Hahne, M. Markovic, and D. Farina, "User adaptation in Myoelectric Man-Machine Interfaces," *Scientific Reports*, vol. 7, no. 1, p. 4437, Jun. 2017.
- [29] M. Couraud, D. Cattaert, F. Paquet, P. Y. Oudeyer, and A. d. Ruyg, "Model and experiments to optimize co-adaptation in a simplified myoelectric control system," *Journal of Neural Engineering*, vol. 15, no. 2, p. 026006, 2018.
- [30] T. A. Kuiken, G. Li, B. A. Lock, R. D. Lipschutz, L. A. Miller, K. A. Stubblefield, and K. B. Englehart, "Targeted muscle reinnervation for real-time myoelectric control of multifunction artificial arms," *Jama*, vol. 301, no. 6, pp. 619–628, 2009.
- [31] L. R. Hochberg, M. D. Serruya, G. M. Fiehs, J. A. Mukand, M. Saleh, A. H. Caplan, A. Branner, D. Chen, R. D. Penn, and J. P. Donoghue, "Neuronal ensemble control of prosthetic devices by a human with tetraplegia," *Nature*, vol. 442, no. 7099, p. 164, 2006.
- [32] M. Velliste, S. Perel, M. C. Spalding, A. S. Whitford, and A. B. Schwartz, "Cortical control of a prosthetic arm for self-feeding," *Nature*, vol. 453, no. 7198, p. 1098, 2008.
- [33] L. Paninski, M. R. Fellows, N. G. Hatsopoulos, and J. P. Donoghue, "Spatiotemporal tuning of motor cortical neurons for hand position and velocity," *Journal of neurophysiology*, vol. 91, no. 1, pp. 515–532, 2004.