EMMILC simulation work

Lucy Hodgins, Chris Freeman, Zehor Belkhatir

May 9, 2024

1 Introduction

This document outlines the preliminary simulations carried out to test the effectiveness of the EMMILC algorithm prior to physical implementation. It summarises the key findings from simulations that demonstrate the advantages of EMMILC over alternative algorithms. It focuses on the system behaviour in the absence of fatigue, although equivalent results were found in the presence of fatigue.

In [1] the motivation behind the use of EMMILC was two-fold. Firstly, the high accuracy of traditional ILC control of electrode arrays comes at the expense of lengthy identification procedures carried out between each trial. This involves the application of a separate ramp input for each electrode in the array, so that when using the full 6×4 electrode array each iteration would require 24 additional identification tests. A successful EMMILC controller must therefore converge faster than ID-based ILC in terms of the total number of tests. This is the first criteria on which EMMILC is evaluated in this work.

Secondly, section 2.4 of [1] highlighted the potential robustness issues associated with using a single nominal model to define the input update on each trial. In order to demonstrate the benefits of EMMILC over this design, it is necessary to show that there are plants within the uncertainty space that can be stabilised by EMMILC but not by a nominal model-based controller. This is therefore the second evaluation criteria.

This document begins with a brief overview of the simulation setup, before outlining the key EMMILC design decisions relating to defining the uncertainty space, controller structure, and model set. Following this, experimental results are presented comparing EMMILC to both ID-based and nominal model-based ILC.

2 Simulation setup

The model described in [1] was simulated in Matlab 2023b. The stimulation input was a truncated ramp input of 10s duration, which levelled out after 6s (see Figure 2a). This replicates a typical setup used in experimental trials, in which the gentle increase ensures that the stimulation remains comfortable for participants, with the time-period chosen to ensure that the joint reaches its final steady-state position. Inputs u_1 and u_2 governed the final height of the ramp applied to each electrode element[1], whilst the outputs q_1 and q_2 were the average joint angles over the final 2s of the simulation. White Gaussian noise with a signal-to-noise ratio of 30 was added to the output, although the overall effect of this was reduced through the averaging process described previously. The reference angles were $q_1 = q_2 = 60°$, corresponding to a pinch posture, as shown in Figure 2b.

3 EMMILC design for simulation

3.1 Defining the uncertainty space

The full system model presented in [1] contains over 70 parameters, from which two needed to be selected to define the simplified uncertainty space. When deciding which would be the most suitable, the following criteria were considered:

[1] With elements defined as shown in Figure 1

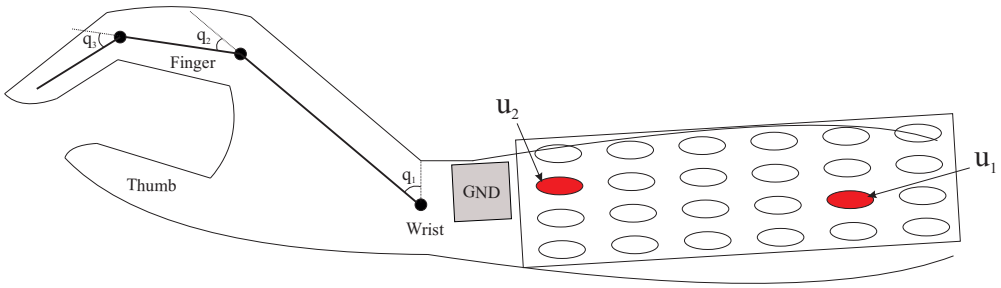
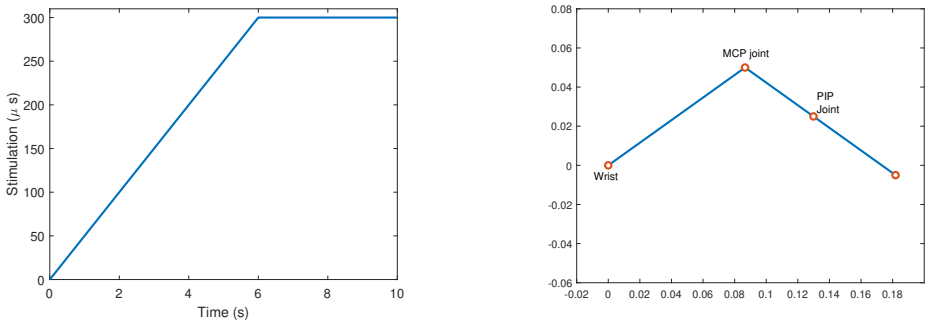


Figure 1: Hand and electrode array



(a) Example input signal

(b) Reference posture

Figure 2

- To what extent does changes in this parameter impact the system response?
- How easy is this parameter to identify? Some parameters, such as the default and cutoff joint angles, could be identified from a single ramp input (as outlined in [1]), removing the need to include them in the set of uncertain parameters.
- Is the range of this parameter well-defined? The static EMMILC algorithm used throughout this report requires the parameter range to lie within pre-specified bounds. For some parameters, such as the maximum force of each muscle, no information could be found regarding the uncertainty from the nominal value, making it much more difficult to apply EMMILC.

Taking each of these factors into consideration, the uncertain parameters were selected as the deadband and saturation values of the isometric recruitment curves (IRC), with these being assumed to be the same for all muscles for simplicity. The deadband and saturation parameters are an alternative to the parameters a_1, a_2, a_3 used to define the response of the IRC in (1). However, the physiological constraints of the system mean that these are a more natural choice. In this work the value of a_1 was set to be equal to the maximum torque, and a_2 and a_3 were determined by using the Matlab $lsqcurvefit$ function to fit (1) to a piece-wise function given by (2). This gave a straightforward mapping between the two parameter definitions.

$$z = h(u) = a_1 \frac{e^{a_2 u} - 1}{e^{a_2 u} + a_3} \tag{1}$$

$$z = \begin{cases} 0 & u < d \\ \frac{a_3}{s-d}(x-d) & d \leq u \leq s \\ a_3 & u > s \end{cases} \tag{2}$$

The deadband and saturation values were selected to define the uncertainty space as their variation could be quantified precisely (following results in [2]) and they were found in preliminary studies to result in significant changes in output for a given input.

The space $\mathcal{U}$ was defined as deadband values between 0-190μs and saturation values between 220-300μs. An additional constraint was put in place to restrict the difference between saturation and

deadband values to be greater than 80, in order to limit the steepness of the slope. These values were selected based on experimental data presented in [2].

3.2 Controller design procedure

One of the necessary conditions for stability of EMMILC is that the controller design procedure $K(p)$ produces a stable controller for all plants within the uncertainty space.Effectively this involves designing a controller capable of stabilising all regions of the uncertainty space given that perfect model information is known.

Given that a linearised ILC controller is to be used, the two remaining design decisions are what form of ILC to use (e.g. Newton, Gradient, Norm-optimal, etc.), and what the optimal value of γ is to enable fast convergence whilst ensuring stability over the entire uncertainty space.

Regarding the first decision, Newton ILC was selected due to its desirable convergence properties. Newton ILC is equivalent to inverse ILC in the linear context, with the update equation given by

$$u_{k+1} = u_k + \gamma(g(u_k))^{-1}e_k \tag{3}$$

Whilst this algorithm is generally less robust than other ILC approaches (such as gradient-based ILC), the structure of the EMMILC controller is designed to compensate for this by increasing the resolution of the model set. Additional modifications were made to the algorithm to account for input saturation due to physiological constraints. The approach used was based on the concept of parameter projection ([3]), but with adaptations made to account for the fact that the boundaries of the input space are not continuously differentiable. Full details can be found in Appendix A.

From initial simulations examining the performance of Newton ILC in all regions of the uncertainty space, a value of $\gamma = 0.2$ was found to provide the fastest convergence whilst producing a stabilising design in all regions of the uncertainty space. This was therefore used as the ILC gain in EMMILC, as well as for ID-based and nominal model-based ILC.

3.3 EMMILC model set design

As explained in [1], the EMMILC model distribution is based on the distance between neighbouring models, as measured using the gap metric, with parameter Δ determining the maximum gap between plants before a new model is introduced. Stability analysis over a range of values of Δ revealed the optimal value (i.e. the fewest number of models that ensured stability over the uncertainty space) to be 0.0025, resulting in the model set distribution shown in Figure 3.

3.4 Estimator design

The only parameter used in the estimator was λ, which governed how much residuals from previous trials were discounted. A value of $\lambda = 0.8$ was selected, as as a compromise between increased robustness no noise whilst allowing the system to adapt to changes in the plant over time, e.g. due to fatigue.

4 Results

4.1 True plant selection

In order to evaluate the performance of each algorithm, six prospective plants were selected from the uncertainty space. This allowed convergence results to be obtained that represent performance for a range of possible plants. The six plants were chosen from an initial sample of 40 randomly-distributed plants in order to give sufficient coverage of the uncertainty space. The distribution of these is shown in orange in Figure 3.

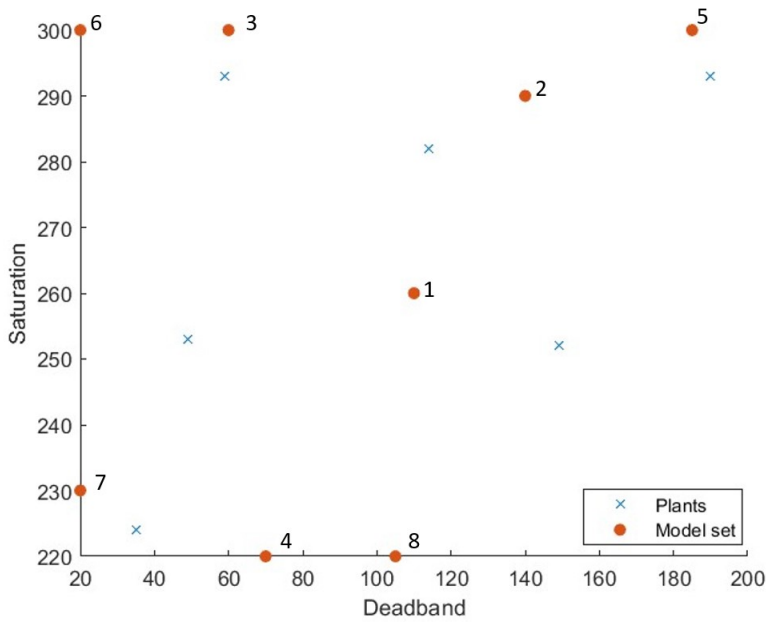


Figure 3: Distribution of the candidate model set and the 'true plant' set used to evaluate performance

5 Comparison between ILC algorithms

Figures 4 and 5 compare the convergence of EMMILC to that of ID-based ILC, and ILC with a nominal model, for the six candidate plants. Convergence is shown over the total number of tests (i.e. three tests for each iteration of ID-based ILC) and reveals that

1. EMMILC converges faster than ID-based Newton ILC for all except one of the plants tested.
2. EMMILC is stable in cases where ILC using a nominal model is not, specifically in the case of plants with deadband=190, saturation=293 and deadband=149, saturation=252.

This demonstrates the feasibility of EMMILC as well as its benefits over other existing designs. In examining the performance of EMMILC it is interesting to examine the models being used for control at each iteration. This is shown in Figure 6, which reveals that model switching typically occurs over the first few iterations, with most systems quickly settling on a stabilising model, and that this model is different in most cases (i.e. the algorithm converges to the model that's closest to the true plant).

6 Summary

This work has demonstrated the feasibility of EMMILC in simulation, as well as its improved convergence speed (in terms of the number of tests) when compared to identification-based ILC, and improved robustness compared to using a nominal model. The design made use of the deadband and saturation parameters of the isometric recruitment curve to define the uncertainty space, and used a Newton-based update step in the ILC control design.

A Saturation of Newton ILC algorithm

This section outlines the modifications made to the standard Newton ILC algorithm in order to ensure fast convergence in the presence of input constraints. The following explanation focuses specifically on the case of two inputs, but the algorithm extends naturally to higher dimensions, in which the input space takes the form of an n-dimensional hyper-cube.

Let τ represent the standard ILC update at each iteration, i.e.

$$\tau = \gamma(g'(u))^{-1}e_k \tag{4}$$

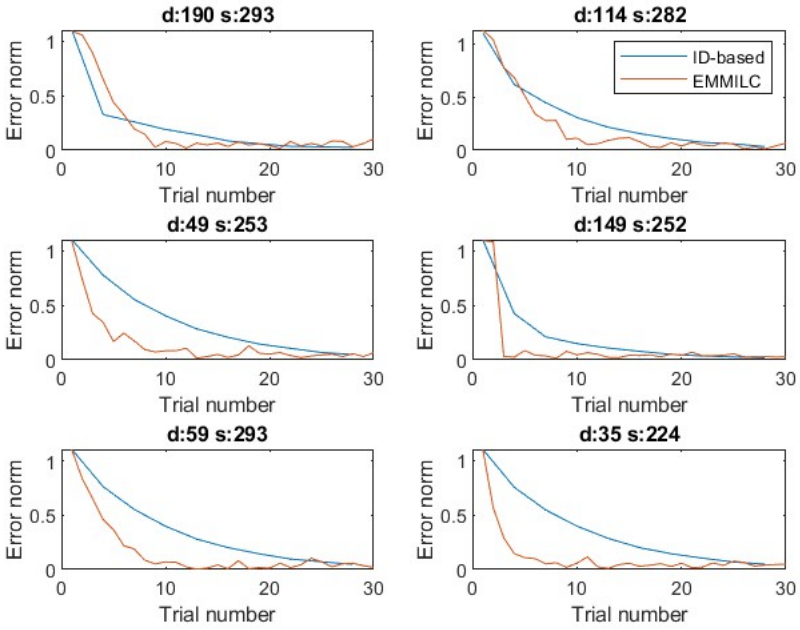


Figure 4: Convergence of EMMILC and identification-based ILC

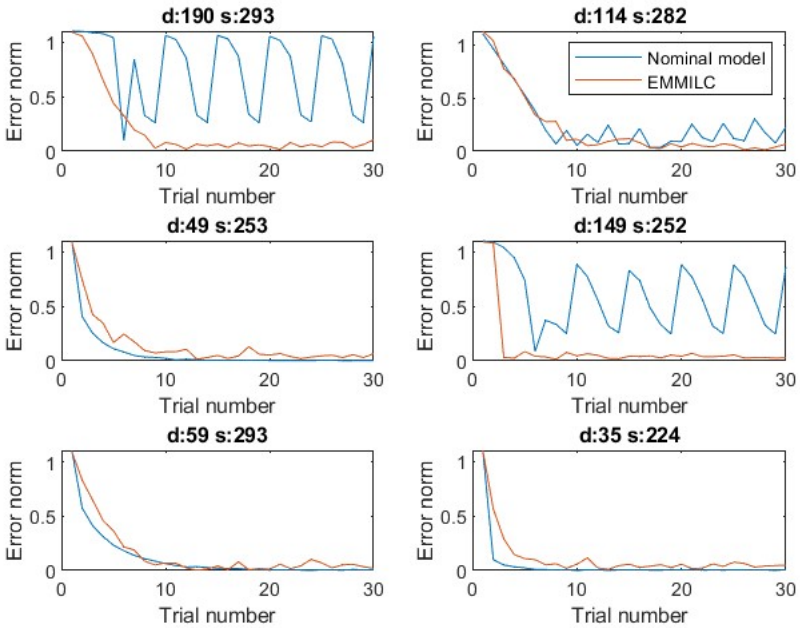


Figure 5: Convergence of EMMILC and nominal model-based ILC

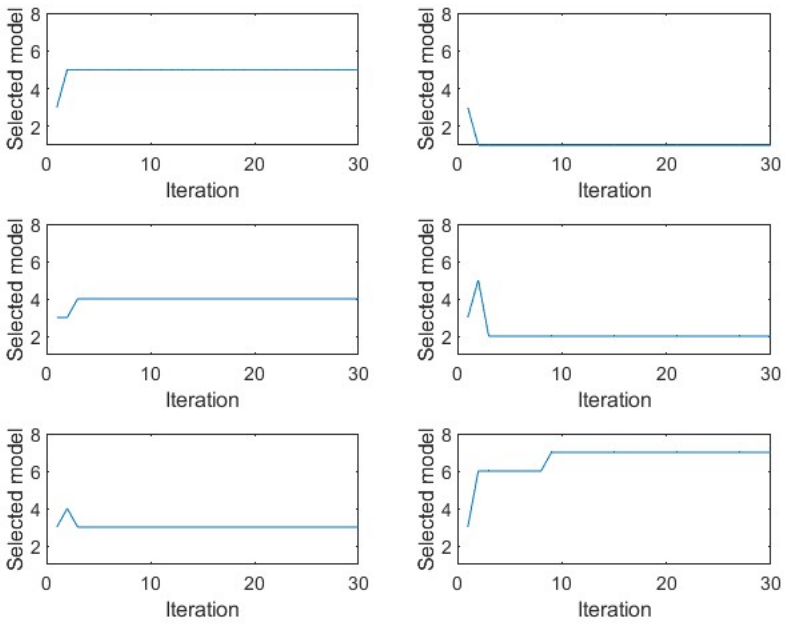


Figure 6: Model switching

for the case of Newton ILC. In cases where this vector update results in input on the following iteration (u_{k+1}) remaining within the input space, as shown in Figures 7a and 7b, the input update law takes the form of the standard Newton approach, $u_{k+1} = u_k + \tau$.

In the situation show in Figure 7c one of the inputs has saturated, and the standard input update τ results in u_{k+1} outside the input space. One option would be to calculate u_{k+1} using the component of τ along the unsaturated input, i.e. $u_{k+1} = u_k + \text{proj}(\tau)$, however implementation of this in simulation revealed that the resulting convergence was very slow. Instead the component of the gradient corresponding to the saturated input was removed from the calculation, giving a pseudo-gradient matrix $\bar{g}'(u_k)$ and corresponding input update

$$\bar{\tau} = \gamma(\bar{g}'(u_k))^{\dagger} e_k \tag{5}$$

which was then used to find u_{k+1}. For example, in the 2-input-2-output case where

$$g(u) = \begin{bmatrix} \frac{\partial g_1}{\partial u_1} & \frac{\partial g_1}{\partial u_2} \\ \frac{\partial g_2}{\partial u_1} & \frac{\partial g_2}{\partial u_2} \end{bmatrix} \tag{6}$$

and the input u_1 had saturated then the augmented matrix would be given by

$$g(u) = \begin{bmatrix} \frac{\partial g_1}{\partial u_2} \\ \frac{\partial g_2}{\partial u_2} \end{bmatrix} \tag{7}$$

If u_k occurs at a corner of the input space, as shown in 7d, the same procedure is used to calculate $\bar{\tau}_1$ and $\bar{\tau}_2$ in either direction, with u_{k+1} being calculated using whichever update allows it to remain within the input space. If both updates move the system outside of the input space this point will be a local minimum, and no change to the input is implemented.

References

[1] L. Hodgins, C. Freeman, and Z. Belkhatir, "Multiple model iterative learning control of FES electrode arrays," *Unpublished*, 2024.

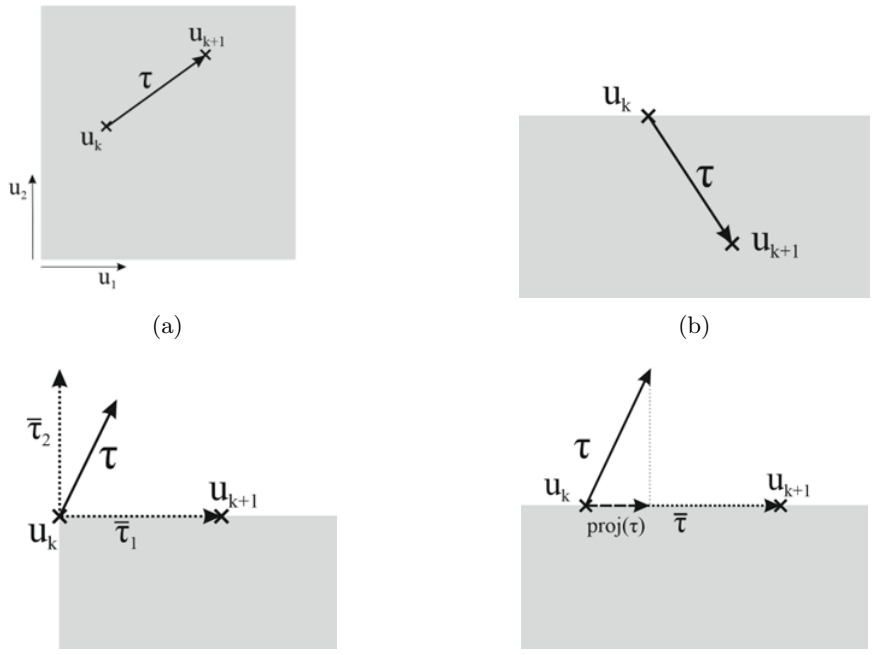


Figure 7: ILC update with input saturation

[2] O. W. Brend, *Implementation and Experimental Evaluation of Multiple Model Switched Adaptive Control for FES-based Rehabilitation.* PhD thesis, University of Southampton, 2013.

[3] M. Krstic, *Delay Compensation for Nonlinear, Adaptive, and PDE Systems.* Birkhäuser, 2009.