# Simulating Human Motor Learning

## An Old Solution in New Environments

Master's Thesis

Zhengping Gu


Delft University of Technology

TU Delft

# Simulating Human Motor Learning

## An Old Solution in New Environments

by

Zhengping Gu

(5281458)

| | |
|---|---|
| Instructor: | Prof.dr.ir. A.C. Schouten |
| Committee Members: | Dr.ir. W. Mugge |
| | Dr.ir. Y.B. Eisma |
| Project Duration: | Jul., 2022 - Apr., 2023 |
| Institution: | Delft University of Technology |
| Faculty: | Faculty of Mechanical, Maritime and Materials Engineering (3mE), Delft |
| Laboratory: | NeuroMuscular Control Lab |
| Degree: | Master of Science in Mechanical Engineering (BioMechanical Design - BioRobotics) |
| Presentation and Defense: | 16 May 2023, 14:30 - 17:00 |

TU Delft

# Acknowledgement

I would like to express my gratitude to my supervisor Prof. Alfred Schouten for his guidance. He offered me this great opportunity to work on this project and provided me with much advice and help with patience. He was always available when I had a question and he is a great teacher who encouraged me a lot. I would also like to thank Dr. Winfred Mugge, Dr. Mark van de Ruit, and Prof. Frans van der Helm for their insight and guidance. They, together with Alfred, gave me many much-needed suggestions about theory, programming, writing, and project management. I would like to thank Dr. Yke Eisma for joining the graduation committee.

I appreciate the company and support that my labmates Natasha and Marjolein warmheartedly offered. They were, most of the time, my motivation to appear on campus. Axel was working on a similar topic and gave me quite some nice insights.

Much thanks to those who offered time and help. I appreciate the mathematical explanation from Ekaterina and all the senior-year advice from Nihar. This project would have been much harder without the help from everyone who was willing to sit down and discuss my problems, including but not limited to Haoran, Xuanming, Quan, Thijs, Nick, Evelien and Vasileios.

I would like to thank all my friends who cared for me. Jeroen, Maria, Asit, Frank, Wenhao, Zhili, Simon, Karthik, Siyang, Xinmiao, Lennart and many others have made my past two and half years an unforgettable one. Special thanks to Floortje for her whiteboard. Best investment ever.

Most importantly, I would like to thank my parents for their endless love and support. Without them, non of these amazing journeys would have happened.

Zhengping Gu
Delft, May 2023

BLANK PAGE

# Simulating Human Motor Learning: an Old Solution in New Environments

Zhengping Gu
Supervisor: Prof.dr.ir. Alfred Schouten

***Abstract*—Feedback error learning (FEL) is a classical computational model that describes human motor learning. It consists of forward and inverse models representing internal dynamics and environmental disturbances. Such models can be used as controllers that represent the function of the motor cortex. On top of FEL, a model has been built with jointly trained feedforward and feedback controllers using a neural network. The controllers that actuated six muscles driving a two-degrees-of-freedom arm model were trained offline. This model successfully simulated human learning of point-to-point reaching movements in a horizontal plane when it was tasked to adapt itself in a null field (NF) and a velocity-dependent force field (VF). In this study, we further tested this model in the divergent force fields (DF) and a channelled force field (CF) to observe its performance. The comparison between the simulation results and the experimental evidence suggests that this model can predict some of the key features of the learning process, such as the kinematics, the muscle dynamics, and the impedance profiles. The learning decay was hindered when the lateral error was artificially eliminated by the CF, as reported in the literature. Overall, the model could converge towards realistic human reaching movements in all of the given environments.**

***Index Terms*—Neuralmuscular control, Arm impedance, Motor learning, Arm movement, Neural network.**

## I. Introduction

HUMANS control their limbs effectively in most of their daily activities, facing a variety of situations. Holding an item in hand, entering water, or fatiguing changes the physical dynamics externally or internally. The capability of humans to adapt their motor control strategy to novel force fields has been extensively studied. Shadmehr and Mussa-Ivaldi (1994) created a widely adopted paradigm for the experiment and simulations, asking the subjects to perform a series of point-to-point reaching motions and observing their trial-to-trial learning behaviour under novel dynamics. The deployed novel dynamics can be stable (Shadmehr and Mussa-Ivaldi, 1994; Lackner and Dizio, 1994) or unstable (Burdet et al., 2001; Franklin et al., 2003, 2007).

A major way of understanding the motor control performances and their adaptation behaviour is by building models and comparing the simulation to the experimental results. Since the learning process is explained as the gradual formation of the representation of the novel dynamics, namely the internal models, in the central nervous system (CNS) (Kawato, 1999), the models' convergence toward an optimum can be seen as a simulation of the progressing learning. Experiments have proven that the internal models consist of a forward and an inverse component. Models like optimal feedback control (OFC) (Todorov and Jordan, 2002; Izawa et al., 2008; Rigoux and Guigon, 2012; Aprasoff and Donchin, 2012; Ueyama, 2014; Razavian et al., 2015; Crevecoeur et al., 2019) focus on the role of the forward model in learning. OFC requires no desired trajectory and is robust to noise, yet its resultant trajectories are theoretically sub-optimal. However, OFC models only predict the post-learning behaviour instead of the learning process. Some other models consist only of the feedforward controller (Thoroughman and Shadmehr, 2000; Burdet et al., 2006; Franklin et al., 2007). Notwithstanding, it is common to include both the feedforward and feedback components in the model (Kawato et al., 1987; Stroeve, 1997; Haruno et al., 2001; Donchin et al., 2003; Kambara et al., 2009).

A straightforward goal for the simulations of learning is to minimize the trajectorial error. The mainstream methods of determining the desired trajectories typically seek the paths that minimize a parameter (Nishii and Taniai, 2009; Ohta et al., 2004), such as the movement jerk (Flash and Hogan, 1985), torque (Uno et al., 1989), variance (Harris and Wolpert, 1998), and so on. Physiologically, the cost also frequently involves energy-related terms such as the muscle activation level (Happee and Van der Helm, 1995) or muscle fatigue (Razavian et al., 2015). The error-effort trade-off is common among the cost functions, as it alleviates the co-contraction level as the training progresses. Some later research also revealed the retention phenomenon in learning on longer time scales (Kooij et al., 2016). The formation of this long term retention is known to be predicted by factors from the short term memory formation (Joiner and Smith, 2008). Models like FEL essentially change the motor command based on the error size (Franklin and Wolpert, 2011). The use of motor primitives in simulation explained some experimental discrepancies and was used to test the generalization of the model in state space, as adaptation is seen as linear functions of the primitives (Thoroughman and Shadmehr, 2000).

Being one of the first to investigate the learning process aside from the learning outcome, Stroeve (1999a) built a model on a simplified 6-muscle-2-link arm model (Winters and Stark, 1985) controlled by a shallow yet sufficient neural network (NN). This model followed the feedback error learning (FEL) scheme (Kawato et al., 1988). In general, although computationally demanding, FEL based on NNs can be expanded into more sophisticated and realistic control systems. Early versions of the FEL mainly focused on integrating the

Delft University of Technology, Delft, 2628CD, The Netherlands;
e-mail: z.gu-2@student.tudelf.nl

tasks that CNS must tackle in motor control into layers of a NN. The forward and inverse internal model pairs were saved into one layer and worked as a combined feedback and feedforward controller. It was noted that the FEL was oversimplified (Kawato et al., 1987; Stroeve, 1997) with the loop delay removed, muscle dynamics linearized, and the forward model omitted. Stroeve implemented these missing parts and specified the task design. In this new model, the NN was simplified to be a single task-specific internal model pair. Thus, the training essentially will not involve the cognitive selection of pre-trained internal models (Wolpert and Kawato, 1998; Haruno et al., 2001). Kambara et al. (2009) proposed a different approach, called the model-control-learning model (MCL), to complete the FEL construction by adopting the actor-critic method to train the feedback controller. This means that the training of the two internal models is separated from the training of the controller. As the model was trained in a reinforcement learning fashion, this method requests no desired trajectories, and the prior knowledge of the dynamics or the explicit expression of the cost function is not needed for its training. Although muscle dynamics and force feedback were not included in MCL, the model was sufficiently tested in various endpoint dynamics, including the velocity-dependent force field (or viscosity field, VF), the divergent force field (DF), and the channelled force field (CF) (Kambara et al., 2021).

In this paper, we revisited the model in Stroeve (1999b) and rebuilt it with up-to-date algorithmic toolboxes. The immediate goal of this project is to test if this newly built version of the model can perform as the previous version did, by comparing the dynamics of the simulation in the NF and the VF to the experiment and the simulation by Stroeve (1999a,b). The model has been validated with the classical experiment in Shadmehr and Mussa-Ivaldi (1994), and the simulated static and dynamic impedance matches the experiments (Gomi and Kawato, 1997). Then the model has been further evaluated with later experimental evidence, including the muscle activation profiles (Thoroughman and Shadmehr, 1999; Heald et al., 2018), new impedance data (Burdet et al., 2001; Tee et al., 2004; Burdet et al., 2013), adaptation index (Smith et al., 2006; Vaswani and Shadmehr, 2013) and so on.

In the following part of this paper, section II will provide the detailed methods to construct and test the model, section III will show the simulation result, and IV will discuss the results in comparison to the experiments, some phenomena observed, and the potential improvements for the model.

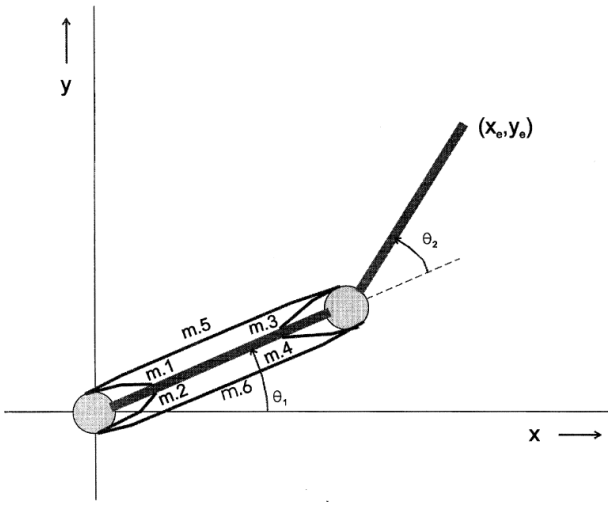



Fig. 1. The musculoskeletal system, adopted from Stroeve (1999a). The muscles are lumped in this simplified model. These six muscles include a pair of biarticular muscles and a monoarticular muscle pair attached to each joint (Burdet et al., 2013). The six muscles are: pectoralis major (m.1); posterior deltoid (m.2); brachioradialis (m.3); triceps lateralis (m.4); biceps (m.5); triceps longus (m.6). The shoulder is marked as joint 1 ($\theta_1$); the elbow is marked as joint 2 ($\theta_2$).

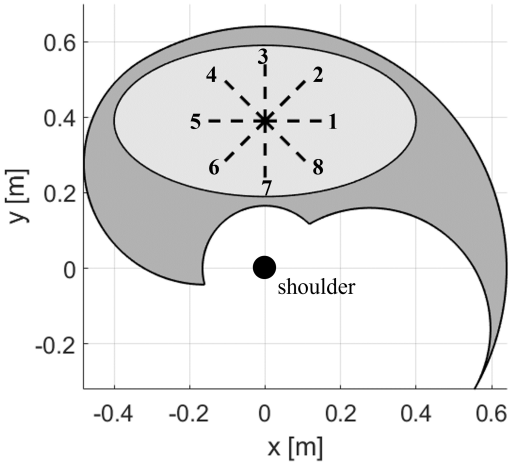



Fig. 2. The workspace is the coloured area in this plot, of which the light grey represents the goal space. The shoulder (joint 1) is placed at the origin (0,0) with the x-axis parallel to the frontal plane. The goal space does not necessarily has to be an ellipse but should cover at least a circle of 0.15m centering the fixed starting point (in Stroeve (1998b) it is rectangular). This workspace is essentially square in the polar system boxed by the maximum and minimum values of the joint angles. The eight desired trajectories are plotted in dash lines with their indices marked on their sides.

## II. Methodology

To simulate the motor learning of an arm (see Figure 1), the model by Stroeve (1999a) was constructed and will be explained in this section. All matrices and vectors are denoted with bold symbols to differ from scalars. In the simulation, the arm is required to make a point-to-point reaching movement within a horizontal plane. The time that the movements would take ranges in the interval [0.25,0.5]s. Upon finishing reaching, the hand should stay at the goal for an equal amount of time. This is called a "trial". The starting position of all trials is set to [0, 0.39]m, which is the center of an ellipse-shaped goal space (Figure 2, with the shoulder being set to the origin). The ellipse has a short axis of 0.2m and an eccentricity of 2.

The positions of the movement goals are distributed within this goal space. Depending on the task, the goal can be uniformly distributed in the goal space or assigned to the end of a standard trajectory. The desired trajectories are straight lines connecting the start and end points assigned. As a set of parameters, they include position, velocity, and acceleration information in Cartesian coordinate. Such desired trajectories feature minimum jerk movements (Flash and Hogan, 1985),

and have bell-shaped velocity profiles.

Eight standard trajectories were introduced, mainly to evaluate the in-and-post training performances. Each of these trajectories directs to a goal 0.15m from the starting position, with $45^\circ$ apart from one another in direction. The trajectory orienting in the positive x direction is named "traj1" with the number increasing until 8, counting the following trajectories counterclockwise. These trajectories are as well shown in Figure 2.

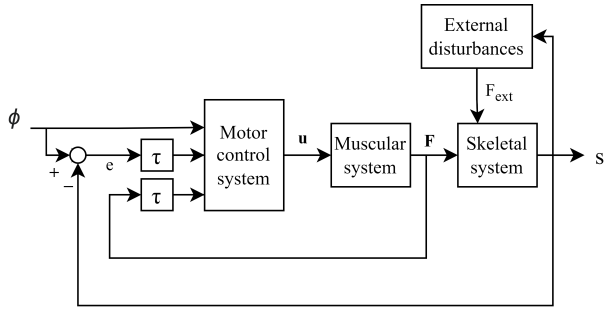



Fig. 3. Control system where $\boldsymbol{\phi}$ is the control input $\left[\phi_1;\phi_2;\dot{\phi_1};\dot{\phi_2};\ddot{\phi_1};\ddot{\phi_2}\right]$, $\mathbf{u}$ is the neural commands, $\mathbf{F}$ is the muscle forces, $\mathbf{F}_{ext}$ is the external force, $s$ is the output $\left[\theta_1;\theta_2;\dot{\theta_1};\dot{\theta_2}\right]$ that $\theta$ is the joint angle, $\mathbf{e}$ is the trajectorial error in the joint space, and $\tau$ is the feedback delay.

The model used for the simulation consists of a controller and a musculoskeletal system, as shown in Figure 3. Such system is based on the model presented in Stroeve (1998b,a, 1999a,b). In a trial, the movement intention from the higher cortex of the brain is represented by reference trajectory in the joint space $\boldsymbol{\phi} = \left[\phi_1;\phi_2;\dot{\phi_1};\dot{\phi_2};\ddot{\phi_1};\ddot{\phi_2}\right]$, and fed to the system as input. This is because the internal models are coded in the joint space (Burdet et al., 2013). The controller that represents the motor cortex receives $\boldsymbol{\phi}$, and the feedback values then output the level of neural command $u$, ranging from 0 to 1. Then the neural command, namely the excitation stimulates the muscles to generate the muscle force $\mathbf{F}$. The system states $\mathbf{s}$ in the joint space are the result of the forces working on the skeletal dynamics, which is also the system output. This includes the joint state $[\theta_1;\theta_2]$ and its time derivatives $\left[\dot{\theta_1};\dot{\theta_2}\right]$. Note that only the states are used as the feedback. $\tau$ is the 50ms feedback delay. The model is constructed in Simulink with an Euler integrator. The simulation is integrated over the generalized dynamics (Vallery and Schwab, 2020), and the offline training is coded in MATLAB (ver. R2022b).

### A. Musculoskeletal model

The musculoskeletal model is based on a Hill-type muscle model (Flash and Hogan, 1985) and an inverted double pendulum that represents the skeletal system. As shown in Figure 1, four monoarticular muscles and two biarticular muscles are attached to the joint with constant moment arms. The muscles have a first-order activation (40ms) and deactivation (70ms) time delay. The details of the analytical representation of the muscles and the limb are included in Appendix A.

With the shoulder resting at the origin, all possible configurations of the arm in the joint space form a workspace, which can be depicted in the Cartesian space, as in Figure 2.

### B. Controller

A multilayer perceptron network (MLP) is used as both the feedforward and feedback controller. The function of the internal models can be simulated by this MLP, whose input includes the desired trajectory, the trajectorial error per instance, and the force feedback. The neural command as the network output is sent to the musculoskeletal system. NNs with more than one layer are in theory capable of mapping nonlinear relationships. Stroeve (1999a) has concluded that an MLP with one hidden layer of 30 nodes is sufficient for the control of the aforementioned arm model. This three-layer network can be notated as:

$$\mathbf{u} = \mathbf{\Gamma}_2(\mathbf{W}_2\mathbf{\Gamma}_1(\mathbf{W}_1\mathbf{h} + \mathbf{b}_1) + \mathbf{b}_2), \tag{1}$$

where $\mathbf{u}$ is the neural commands, $\mathbf{h} = [\boldsymbol{\phi}, \mathbf{F}, \boldsymbol{\theta}]$ is the inputs, $\mathbf{W}$ is the weight matrices, $\mathbf{b}$ is the biased terms, and $\mathbf{\Gamma}$ is the sigmoidal activation function

$$\Gamma(x) = \frac{1}{e^{-x}+1}. \tag{2}$$

The musculoskeletal system can be generalized into a time-invariant system, since its update is off-line. The system has a state vector $\mathbf{x}$ and output $\mathbf{y}$:

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \\ \mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t)) \end{cases}. \tag{3}$$

The definition of the variables above can be found in Appendix B.

### C. Learning

Both the feedforward and feedback parts of the controller are optimized during the training toward the arm following the desired trajectories. This optimization problem, like any others, is determined by three components. Firstly, the decision variables are the weight matrices $\mathbf{W}$s. Secondly, the to-be-optimized weight matrices are not constrained. However, their initial values are kept low (in absolute value, yet non-zero) to avoid instant neuron saturation. Lastly, the performance of a trial of arm reaching movement is measured by a cost function that takes the trajectorial error and activation level into account:

$$J = \frac{1}{T}\int_0^T \left\{ \sum_{i=1}^{2} \left[\phi_i(t) - \theta_i(t)\right]^2 + \alpha \sum_{j=1}^{6} \frac{V_j}{V_{avg}} {a_j}^2(t) \right\} dt, \tag{4}$$

where $\theta$ is the position of the arm, which, as it has been concluded for motor control and learning, is in the joint space (Shadmehr and Mussa-Ivaldi, 1994; Burdet et al., 2013). $\alpha$ is the multiplier for the activation penalty to normalize it to the magnitude level of the trajectorial errors. Empirically its value is set to 0.01 as it determines the level of muscle co-activation,

which shall be maintained at 5% (Stroeve, 1999b). Gradient descent is used for the optimization, that is,

$$\mathbf{W}_{i,new} = \mathbf{W}_{i,old} - \eta \Delta \mathbf{W}_i, \tag{5}$$

with $\eta$ being the learning rate, which is set to 0.2 for this specific case. This is practically the highest learning rate that the optimizer allows without any significant divergent behaviour. $\Delta W_i$ (with $i \in [1, n_{layer} - 1]$) represents the effect of controller weights on $J$ averaged over time. This equation, from the perspective of training a neural network, stands for a batch update of a supervised learning session.

The means of obtaining $\Delta \mathbf{W}_i$ is by averaging the derivative of the cost function over the neural command u over the movement time $t$:

$$\frac{\partial J}{\partial \mathbf{u}}(t) \tag{6}$$

This term represents the direct effect of the muscle commands on the overall performance of the system. backpropagation through time (BTT) was used to calculate this Jacobian iteratively. Stroeve chose the ordered derivative to represent how $u$ at one instance can affect $J$ over the whole trial. The calculated $\frac{\partial J}{\partial \mathbf{u}}$ can then be linearized and used to calculate the Jacobians at the previous instance:

$$\frac{\partial J}{\partial x_j\,(N)} = \frac{\partial l\,(N)}{\partial x_j\,(N)} + \sum_{i=1}^{p} \frac{\partial l\,(N)}{\partial y_i\,(N)} \frac{\partial y_i\,(N)}{\partial x_j\,(N)} \tag{7}$$

$$\begin{aligned} \frac{\partial J}{\partial x_j\,(k)} = \frac{\partial l\,(k)}{\partial x_j\,(k)} + \sum_{i=1}^{p} \frac{\partial l\,(k)}{\partial y_i\,(k)} \frac{\partial y_i\,(k)}{\partial x_j\,(k)} + \\ \sum_{i=1}^{n} \frac{\partial J\,(k)}{\partial x_i\,(k+1)} \frac{\partial x_i\,(k+1)}{\partial x_j\,(k)} \end{aligned} \tag{8}$$

$$\begin{aligned} \frac{\partial J}{\partial u_j\,(k)} = \frac{\partial l\,(k)}{\partial u_j\,(k)} + \sum_{i=1}^{p} \frac{\partial l\,(k)}{\partial y_i\,(k)} \frac{\partial y_i\,(k)}{\partial u_j\,(k)} + \\ \sum_{i=1}^{n} \frac{\partial J\,(k)}{\partial x_i\,(k+1)} \frac{\partial x_i\,(k+1)}{\partial u_j\,(k)} \end{aligned} \tag{9}$$

where

$$\frac{\partial x_i\,(k+1)}{\partial x_j\,(k)} = \begin{cases} 1 + h \frac{\partial f_i}{\partial x_j}\left(\mathbf{x}\,(k), \mathbf{u}\,(k)\right), & i = j \\ h \frac{\partial f_i}{\partial x_j}\left(\mathbf{x}\,(k), \mathbf{u}\,(k)\right), & i \neq j \end{cases} \tag{10}$$

$$\frac{\partial x_i\,(k+1)}{\partial u_j\,(k)} = h \frac{\partial f_i}{\partial u_j}\left(\mathbf{x}\,(k), \mathbf{u}\,(k)\right). \tag{11}$$

The idea is that since for each term the Jacobian contains all Jacobians of the later instances. Therefore starting from the last trial, and passing the values to the previous trials accumulatively avoids repetitive calculation. The details of this backpropagation process are described in Stroeve (1999b) and partly explained in Appendix B.

### D. Task design

The tasks that involve arm-reaching movements are based on the paradigm by Shadmehr and Mussa-Ivaldi (1994). This paradigm is referred to for both learning and validation. The model was first randomly initialized and trained for 100,000 trials in the null field (NF), where no external force is applied. Then the model was trained for an equal amount of trials in a VF which imposes an external force on the endpoint of the arm. For both the NF and the VF sessions, the goals were randomly assigned in the goal space. The values used to define the VF in Stroeve (1999a) are the same as in Shadmehr and Mussa-Ivaldi (1994):

$$F_{ext,VF0} = \begin{bmatrix} 10.1 & 11.2 \\ 11.2 & -10.1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix}. \tag{12}$$

However, in this project, it was proven to be too big as the hand would be pushed to the edge of the workspace. Although eventually the model converged to the same optimum, the learning was slowed down, and affected by the activated spatial constraints. Therefore, for this project, the external viscosity was reduced by 1.5 times, getting the VF1:

$$F_{ext,VF1} = \begin{bmatrix} 6.73 & 7.47 \\ 7.47 & -7.4 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix}. \tag{13}$$

The force field was then removed after the training in the VF1 and retrained for 100,000 trials in the NF to observe the after-effect (AE). A second adaptation session in the VF1 was introduced after the AE. VF1 is the one that will mainly be referred to in this paper, while a different version was used, which is perpendicular to the hand velocity direction. This second VF (VF2) can be used for introducing lateral disturbance (Vaswani and Shadmehr, 2013; Kambara et al., 2021):

$$F_{ext,VF2} = \begin{bmatrix} 0 & 13 \\ 13 & 0 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix}. \tag{14}$$

With the NF training result being the baseline, the DF and the CF were applied in separate computational experiments to examine the model's adaptation performance. Unlike the VF, since the orientation of these force fields depends on the direction of the desired trajectory, only the trajectory in the positive y direction ("traj-3") was used for these training sessions. The value for the DF1 is taken from Kambara et al. (2021) and Burdet et al. (2001):

$$F_{ext,DF1} = \begin{bmatrix} 200 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{15}$$

for easy comparison. To explore the system's reaction to the DF1, the external stiffness was also tripled in another training session Burdet et al. (2001), resulting the DF2:

$$F_{ext,DF2} = \begin{bmatrix} 600 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}. \tag{16}$$

The CF is adopted from Kambara et al. (2021), which serves as an error clamp that artificially removes trajectorial error (Smith et al., 2006):

$$F_{ext,CF} = \begin{bmatrix} -600 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} -60 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix}. \tag{17}$$

The task design for the CF trials follows the scheme of the human experiment (Vaswani and Shadmehr, 2013). Based on the learnt model in the NF, a further 250 trials were performed in the NF but confined to traj-3. Then the model was trained for 300 trials in the VF2 before being put under the CF or NF for comparison. The complete scheme is summarized in Figure 4.

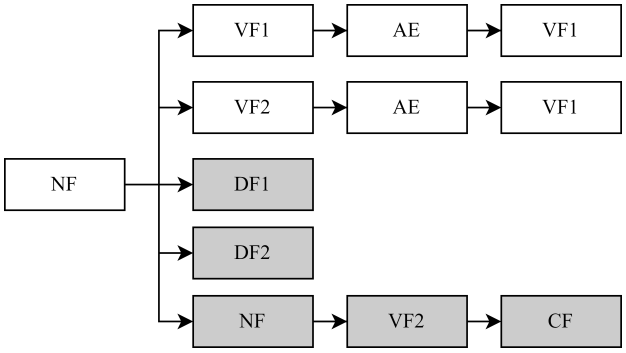



Fig. 4. Test scheme, the arrow indicates how the resulting model of one training session was used in the following sessions. The grey box indicates the trial used a fixed desired trajectory.

To produce comparable models in the NF and the VF, the simulation was run on traj-3 for 1000 trials each as well. This is not shown in Figure 4 or Section III, but was used for impedance evaluation shown in Appendix D.

### *E. Evaluation*

For the trials executed in the NF and the VF, the simulation was performed eight times over each of the standard desired trajectories with the controller weights $\mathbf{W}_1$ and $\mathbf{W}_2$ from a few selected trials throughout the training, so that important metrics and profiles describing the system dynamics can be recorded (Stroeve, 1999a), including the

- Cost
- Trajectorial error
- Activation level averaged over all muscles and all time instances,
- Muscle activation profiles
- Hand velocity profile in Cartesian space
- Impedance profile (intrinsic, reflexive, and empirical stiffness)
- Adaptation index (lateral force correlation coefficient for the CF session).

In the NF and the VF sessions when the goal space was used, the costs that represent the model performance, the trajectorial errors, and the activation levels were calculated by averaging over the eight trials that are based on the same weights. The other parameters were stored separately for each trajectory. For DF and CF sessions, since there was only one desired trajectory, such averaging was skipped.

In these evaluation trials, the endpoint stiffness was as well calculated accordingly to represent the system impedance. The details of the derivation can be found in Appendix B. This is mainly to evaluate the DF simulation results but NF and VF sessions were as well briefly included (see Appendix D for the results). The components that make up the impedance include the intrinsic impedance, the reflexive impedance, and the impedance induced by the feedforward behaviour of the controller. In this project, the stiffness was chosen to reflect the impedance profile in general, as the experiment suggested that the damping in the musculoskeletal system can be empirically calculated if the stiffness is known (Burdet et al., 2000).

Stroeve (1999b) calculated the intrinsic stiffness by deriving the sensitivity term of the control system. For this project, this was done by analytically calculating

$$\mathbf{Z}_k = \frac{d\mathbf{F}_{end}}{d\mathbf{p}_{end}} \tag{18}$$

where $\mathbf{F}_{end}$ is the sum of the active and passive forces projected to the end-point (hand), and $d\mathbf{p}_{end}$ is the infinitesimal displacement at the end-point position.

For the static case, the model was reduced to an automatic system with a sole feedback controller. The loop delay could be tackled with a Pate filter. In Stroeve (1999b,c), the system was linearized around a working point while deterministic multi-sine force impulses were applied as the external disturbance. The stiffness matrix was then acquired by substituting the frequency in the Laplace domain with zero. If the dynamic impedance characteristic were to be calculated, it would request additional simulation sessions and effective system identification to regress the model to a second-order one. Therefore, only the static impedance was adopted in this project, although the dynamic impedance profiles would have been analytically more accurate and interpretable. To calculate the reflexive stiffness with this feedback-only approach, simply a representation of the system in state space is needed. The stiffness would be the inverse of the system transfer function.

The impedance can also be approximated with an alternative approach (Tee et al., 2004; Burdet et al., 2013). This empirical method calculated the impedance as a linear function of the joint torque. This assumes that the static impedance can represent the dynamic impedance and acknowledges that the intrinsic impedance is not separable from the activation-dependent impedance.

The stiffness ellipses were then calculated with singular value decomposition (Gomi and Osu, 1998). The long and short axes of the ellipses are the maximum and minimum reaction force vectors in the given state. Their ratio gives the eccentricity and their perspective directions give the direction of the reaction forces.

For the CF trials, Smith et al. (2006) and Vaswani and Shadmehr (2013) used an index to represent the degree of adaptation. How the index should be calculated was not described in detail but it resembles the normalized correlation coefficient between the lateral component of the endpoint force profile and the ideal compensatory force in the VF2. Thus the index approaching 1 indicates that the profile is identical to the ideal case while -1 means the force compensates the VF2 in the opposite direction, with an identical but inverted force profile.

## III. Results

In the following subsections, the convergence characteristics and criteria selected for evaluations will be selectively dis-

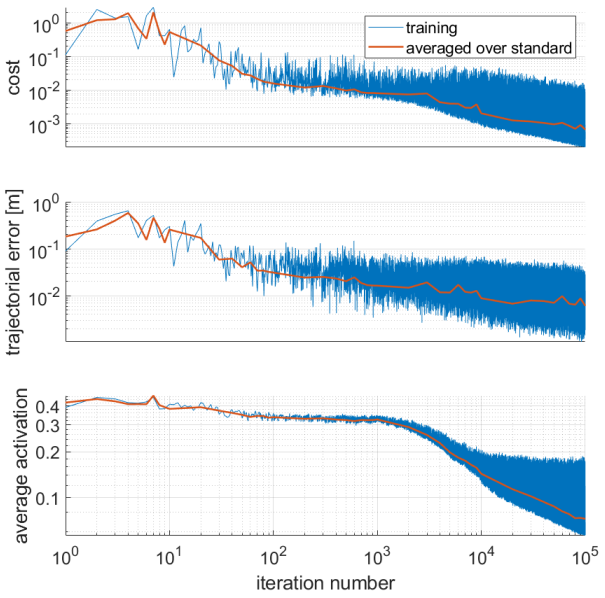



Fig. 5. Performance parameters during the NF training session with randomized desired trajectories in the goal space. The red line shows the same metrics from the trials that were simulated based on the controller weights from the respective iteration. Per iteration, the simulation was run eight times over all eight standard trajectories and the average values were taken from the resulting metrics.

cussed for each force field according to its respective scheme of training and testing. The coding process is explained in Appendix C and some additional results are presented in Appendix D.

### A. Learning in the NF

The system was with a controller that has randomly defined initial weights and was first trained in the NF as the baseline. The training process in which the model had to follow the randomized desired trajectories is illustrated in Figure 5. The learning effectively approaches its optima in the first 1000 trials. This agrees with Stroeve's comment on how fast the system can converge. The optimization process after the initial convergence sees the lower boundary of the cost further lowered while the upper boundary stayed at the same level. This, along with the oscillation of cost is related to the randomized goals and time intervals.

The cost per trial is primarily affected by the position of the goal of the desired trajectory. In general, the shorter the trajectory is, the lower the cost is. The oscillatory behaviour is mitigated when the desired trajectories are randomly selected from the eight standard trajectories and disappear when the desired trajectory is fixed (not shown in this paper). With a fixed starting point and a randomly assigned goal, the cost corresponding to a goal 20cm away from the starting point can be as high as 40 times the cost of a goal 2cm away. This relation is shown in Figure 6. The cost is especially high for traj-8 (the red region in the direction of the fourth quadrant). This might be because traj-8 is longer in the joint space

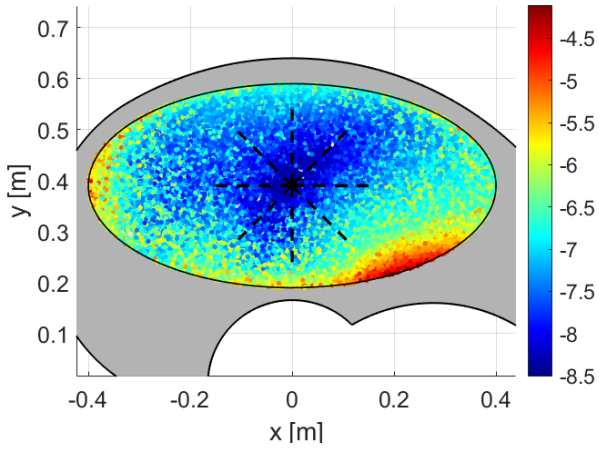



Fig. 6. Cost map in the workspace with no external forces applied to the hand (NF). Each dot marks the goal of a trial with the colour-coded cost in log scale with a base of 10.

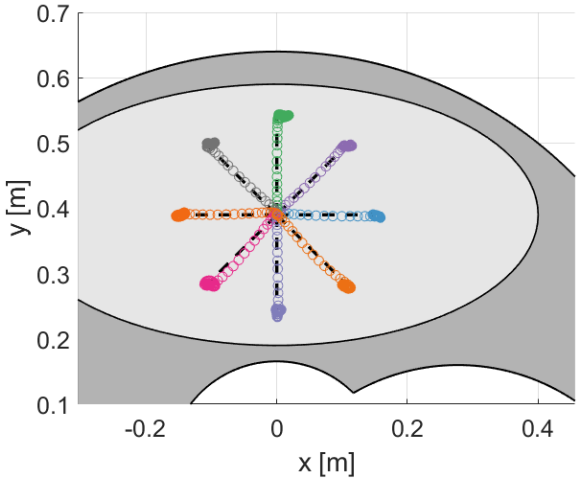



Fig. 7. Baseline performance of the model trained in NF. Each circle in the same color group is evenly distributed in reference to the movement time.

compared to the rest of the standard trajectories, as the edges of the high-cost regions in this cost map coincide with the grid lines in the joint space. The cost map in the VF is basically the same as this one. Presumably, a position-dependent force field such as the DF should drastically change the cost map terrain, if the DF was used in the entire goal space.

Figure 7 shows the final training outcome in the NF, as the model performed the task in all eight standard trajectories with the latest weights optimized for the controller. The system can converge toward a straight-line trajectory with a bell-shaped velocity profile close to the desired performance, as Figure 8 demonstrates.

As the baseline, the muscle activation profiles and in reference to the movement time was as well calculated (Figure 9. The impedance profile in the NF will be discussed together with the training in other force fields in the following subsections. The more generalized way of demonstrating the

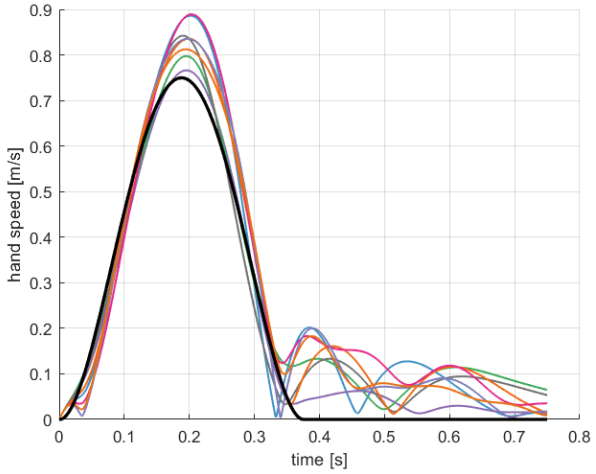



Fig. 8. Velocity profile by the end of the NF training in eight directions with the same colour code in Figure 7, and the desired velocity profile derived from Flash and Hogan (1985) (black). The velocity here is the modular length of the velocity vector thus no negative value is possible. This figure also shows the endpoint oscillation around the goal once it reaches it. Instead of stopping still, the arm hovers around as a result of the cost-performance trade-off.

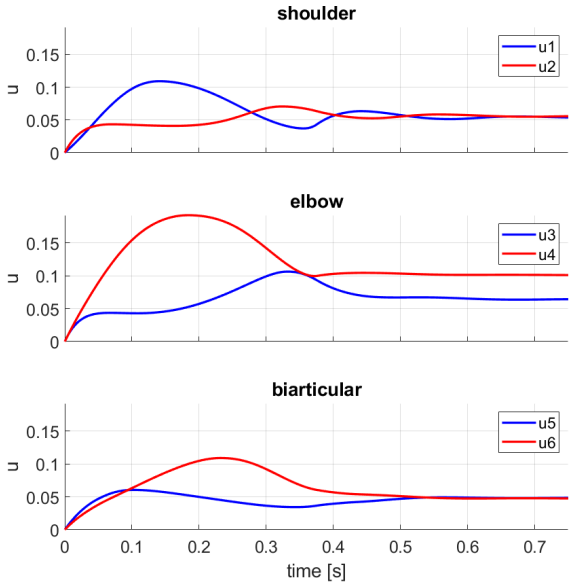



Fig. 9. Neural command profile by the end of training in the NF, traj-3 was given as the desired trajectory. The muscle pairs are separated into three subplots, with u5 and u6 being the muscle activation of the biarticular muscles while the others being the monoarticular muscle activation level for the shoulder (u1 and u2) and the elbow (u3 and u4) respectively.

convergence in the NF, namely averaging all parameters over the trials performed on standard desired trajectories, is shown in Figure 10 together with the VF and another NF to see the AE. The cost was, in general, decreasing over the 100,000 trials, yet per component inside the cost did not decrease uniformly, at least in the first 10,000 trials.

### *B. Learning in the VF*

The model was trained with the NF-VF-NF-VF scheme. These four training sessions are sequential. The training resulted in movement performances converging toward the optima in all four sessions, meaning the trajectories for the standard trials would converge toward the baseline shown in Figure 7. Figure 10 shows the change in the three evaluation criteria during the optimization. Since the NF training started with $\mathbf{W}$, which is randomized close to zero, the cost at the beginning stayed high. The rest three sessions also experienced relatively higher costs at the beginning, yet the increment from the previous optima is relatively low. The activation level stayed low even during training in all force fields but the NF. This implies that the change of force field did not pull the cost too far away from the optima, thus the control strategy the model learns in one force field (including the NF) is transferable in another force field. The convergence of the activaton profile in the VF1 is shown in Figure 11.

The trajectories upon the first exposures in the VF1 for the first and second time, along with the after-effect after training in the VF1 for the first time, are shown in Figure 12. It can be observed that the novel dynamics caused lateral movements in reference to the desired trajectories. Since the model hasn't learnt to counter the yet unknown force field at its first exposure, the remedy to this deviated trajectory resulted in a hook-like trajectory (Shadmehr and Mussa-Ivaldi, 1994).

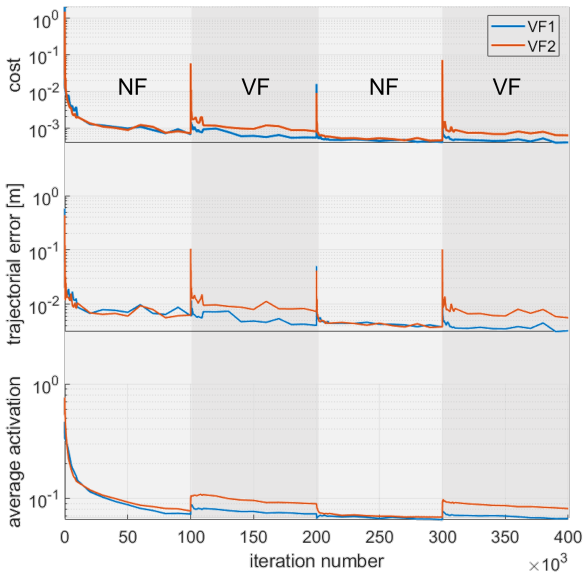



Fig. 10. The convergence of the performance parameters during the NF-VF-NF-VF training sessions averaged over eight standard desired trajectories. The trial number refers to the trial from which the weights of the NN were collected and used for the standard trials. The training results for the first 100,000 iterations in the NF were not identical in the two attempts since the system at the first iteration was initialized with different and random controller weights.

After the model had adapted to the VF1, the moment the force field was removed, the after-effect appeared as a deviation in the opposite direction, only this time the trajectories tended not to overshoot the goal so the hooks seemed smaller. After the model adapted to the NF for the second time, the model

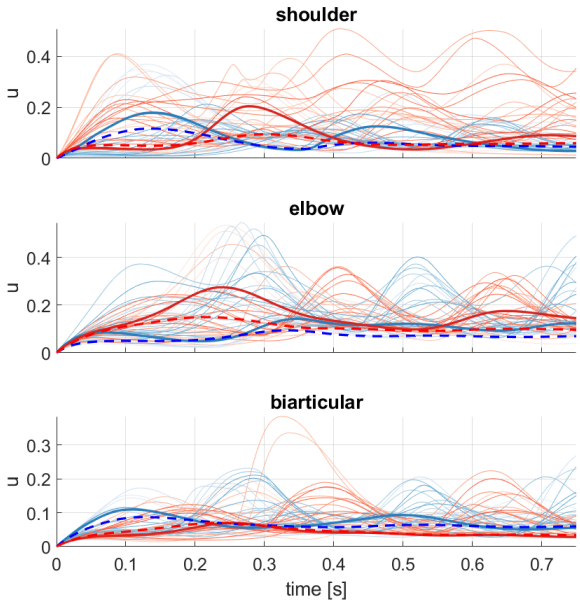



Fig. 11. Neural command profiles throughout the training in VF1, the line colours darken with the trial number progresses logarithmically. Blue stands for flexive activation and red stands for extensive activation. The activation profiles of the muscles in a trained model are marked with bold lines. Dash lines stand for the activation profile in the trained model for the NF before VF1 was imposed.

was again exposed to VF. Notably, the deviations upon this second exposure in the VF1 are not entirely the same as that for the first time. The system took a longer detour and failed to eventually stop at the goal for some of the standard desired trajectories. This is possibly because the muscle activation was lower after the training in the AE compared to the end of the first adaptation in the NF. Thus the introduction of the VF1 made the system temporarily unstable.

Judging by the trajectory profile for the first exposures, it is possible that the feedforward controller contributed to most of the eliminated trajectorial errors, as the feedback controller only actuated a slower motion that mainly forms the "hook" after the desired trajectory as the system input went to constant.

Figure 10 also illustrates the convergence performance and the disturbed trajectories with VF2 being applied. It appears that the force field perpendicular to the endpoint velocity direction caused less disturbance than VF1 in general. The change of force field caused approximately equal amounts of increment in the cost, regardless of which force field it is. This trend is roughly the same as in the changes in the trajectorial errors. Since the activation did not increase once the model was trained once in the NF, it can be said that the rise in cost is mainly caused by the trajectorial error, and the change of control strategy did not involve the rise of control commands to the muscles. This time the trajectory remained the same for both exposures in the VF2, and the deviation in the aftereffect seems larger than in the VF1.

### C. Learning in the DF

The two sessions of training in the DFs shown in this part are not consecutive. Instead, they both started after a trained NF session. This assumes that all attempts to adapt in the NF end in the same optimum since the models always converge for different initial $\mathbf{W}$s, and the trained performance are similar to each other. The convergence of the cost, trajectory error and average activation is illustrated in Figure 13. Since the goal is no longer random, the evaluation of this training did not involve any simulation over the other standard trajectories. The oscillation in this figure is partly because of the random movement time for each trial and partly because the hand position started to oscillate around the goal once the reaching movement was finished and the model started to maintain the posture. The cost decreased as it would with the force fields previously covered, and so did the trajectorial error. However, unlike in the VFs, the activation level increased at the beginning of the training and peaked at around 1000 trials. The activation only fell back gradually after that and converged toward the minima. The neural command profiles for the DF1 are shown in Figure 14.

The trajectories pre- and post-exposure to DF1 and DF2 are shown in Figure 15. The red line is captured at 1400 trials into the optimization, where the activation level averaged throughout a trial is close to its highest. The major part of this optimization is to make the hand eventually stop at its goal. Reaching straightly from the starting point toward the goal was even possible in the first trial upon exposure. Most of the trajectorial errors are the result of the model drawing the hand back to the goal after passing it. Deviations during the movement are also not as large as the ones when the arm was trying to maintain the posture. This part of the trajectory is not shown in the figure. Naturally, it is intuitive to find that the lateral deviation of DF2 is larger than DF1, as the external force is larger for the same amount of deviation.

The stiffness ellipses are presented in Figure 15. They show the end-point stiffness eclipses of the model at the beginning of the selected trial, when it was halfway to the goal, and when it first arrived at the desired $y_t$, ignoring the $x$ component of its position. The full profile of the stiffness is included in the appendix D. The impedance changes when the training progresses and differs per time step in a trial. The system always started the movement with a similar stiffness as in the trained NF model, where the stiffness stayed roughly the same throughout the motion. In general, the stiffness ellipses are significantly distorted when the hand moves away from the desired trajectory. This happens when the model could not follow the straight line trajectory and steadily hold the final position before it was optimized. Note that the empirical stiffness shown in Figure 15 excluded the extra term representing the cocontraction's effect (Tee et al., 2004).

### D. Learning in the CF

For the CF, the training scheme is executed with all randomized factors in the simulations removed. The training process is shown in Figure 16. Instead of the trajectorial error and the activation level, the lateral force correlation coefficient

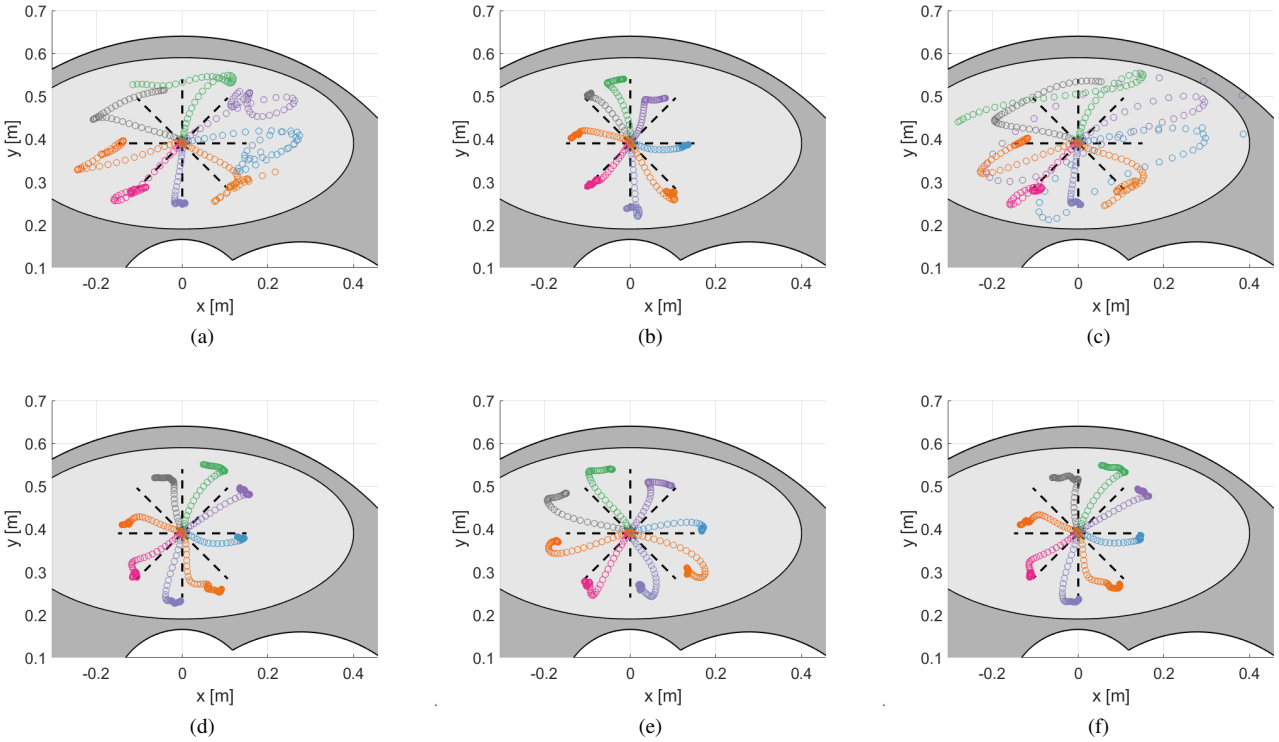



Fig. 12. Trajectories of the model following the eight standard desired trajectories upon (a) the first exposure in the VF1; (b) the after-effect of the post-VF1-adaptation system in the NF; (c) the first exposure in the VF1 for the second time. The endpoint of the limb actually went so far that it hit the workspace boundary again; (d) the first exposure in the VF2; (e) the after-effect of the post-VF2-adaptation system in the NF; (f) the first exposure in the VF2 for the second time.

against the lateral force profile from the end of VF2 training is shown, together with the maximum lateral deviation from the desired trajectory before the desired $y$ position was reached. The numbers of trials are adopted from Vaswani and Shadmehr (2013). For one fixed trajectory, the model converges much faster. It can be seen that for a model pre-trianed in the NF, the training without the error clamp converges within 200 trials. With the CF introduced, although the extent of learning is observably mitigated, the 600-trial learning session still well covered the adaptation process.

## IV. Discussion

A motor control model was constructed with an MLP as the controller. With the BTT algorithm, the feedforward and feedback controllers that the MLP represents have been jointly trained. The training has been performed according to the scheme presented in Figure 4. Trajectories and parameters that are used to evaluate the performance of the model were derived and specifically, the impedance was calculated for the DF sessions and the force-based learning index for the CF sessions. In this section, the simulation result will be discussed and qualitatively compared to existing experimental results. Appendix D includes some of the illustrated comparisons.

### A. Adaptation

The NF-VF-NF-VF convergence characteristics, including the aforementioned velocity profiles, are consistent with Stroeve (1999a), and therefore agree with Shadmehr and Mussa-Ivaldi (1994). Figure 12 shows both the trajectories disturbed by the VF for the first time and the ones with the first removal of the VF. It is clear in the figure that the lateral deviation between the trajectories and their respective desired trajectories are in opposite directions for VF first exposure and for the AE. This suggests that adapting to the VF means that the system compensates for the external force-induced errors (Shadmehr and Mussa-Ivaldi, 1994).

During the training, the trajectory error declined before the activation. This suggests that the primary goal of the optimization is to follow the trajectory and the value of $\alpha$ should not be too high. A higher $\alpha$ punishes the activation more strictly and thus will force the system to prioritize lowering the neural command to the muscles. This might result in converged trajectories that do not follow the desired ones but minimizes the actuation effort. The error-effort trade-off can be seen in the end-of-training trajectories in the VF2, which slightly deviated from the desired trajectory while such deviation did not happen in the NF sessions.

The results of the training in the NF and the VF resemble realistic human movement. The bell-shaped endpoint velocity profile Shadmehr and Mussa-Ivaldi (1994) was achieved. The direction of the movement did not induce many observable differences in the hand velocity profiles, thus an averaged profile is representative. The velocity curve from the experiment was much smoother in the NF. The simulated NF velocity is closer to the experimental velocity in the VF, which is less stable

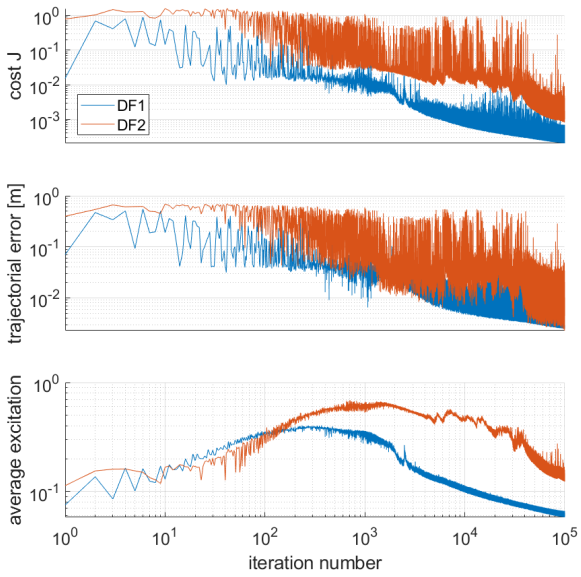



Fig. 13. The change of the evaluation metrics during the DF training sessions. The NF part is not included since it is identical to the VF cases. Only traj-3 is used as the desired trajectory thus the results were not averaged in any way.

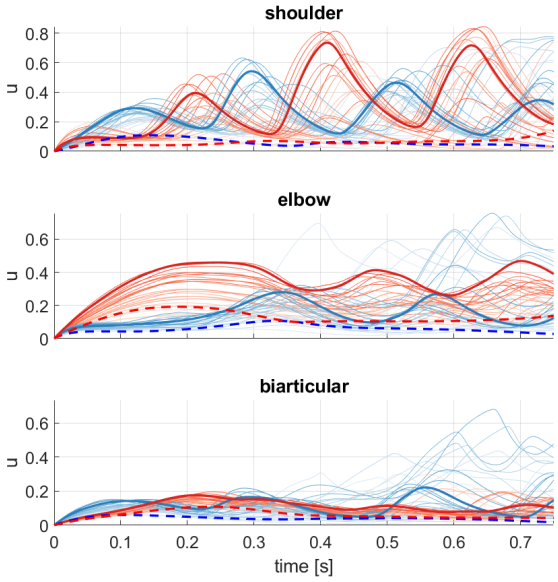



Fig. 14. Neural command profiles for the throughout the training in DF1, the line colours darken with the trial number progresses logarithmically. Blue stands for flexive activation and red stands for extensive activation. The activation profiles of the muscles in a trained model are marked with bold lines. Dash lines stand for the activation profile in the trained model for the NF (Figure 9).

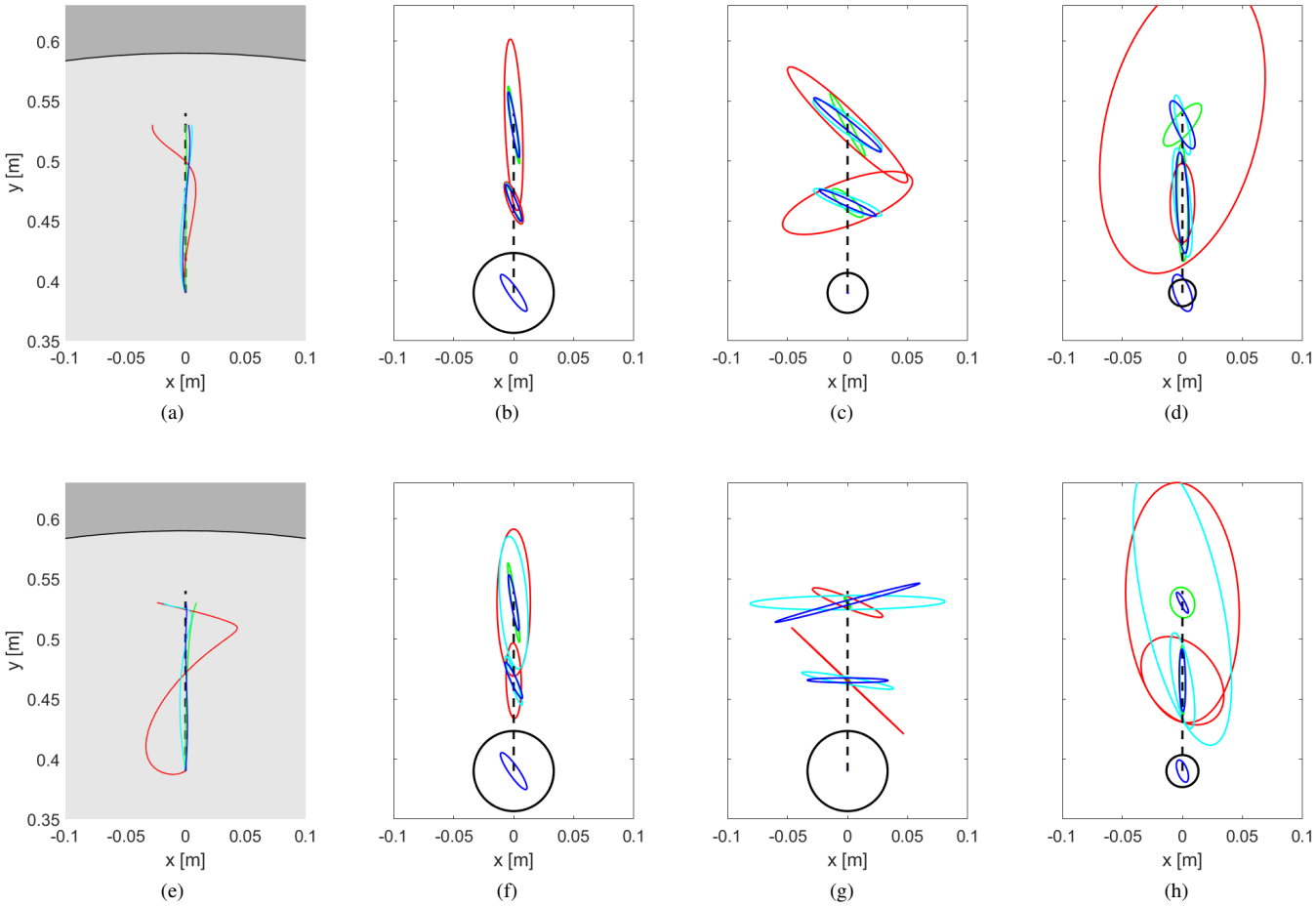



Fig. 15. Trajectories (a & b) and impedance ellipses (others) in the DF1 (first row) and the DF2 (second row), upon exposure (green), 1,400 trials into adaptation (red), 30,000 trials into adaptation (cyan), and post adaptation (blue). The subfigures demonstrate the intrinsic (b & f), reflexive (c & g), and empirical (d & h) stiffness. These figures do not show the parts of the trajectories overshooting the destined $y_t$. After reaching the goal in the DF, oscillation around it started. The black circle in the middle stands for: (b) 100N/m reference for DF1 intrinsic stiffness; (c) 10N/m reference for DF1 reflexive stiffness; (d) 100N/m reference for DF1 empirical stiffness; (f) 100N/m reference for DF2 intrinsic stiffness; (g) 100N/m reference for DF2 reflexive stiffness; (h) 200N/m reference for DF2 empirical stiffness.

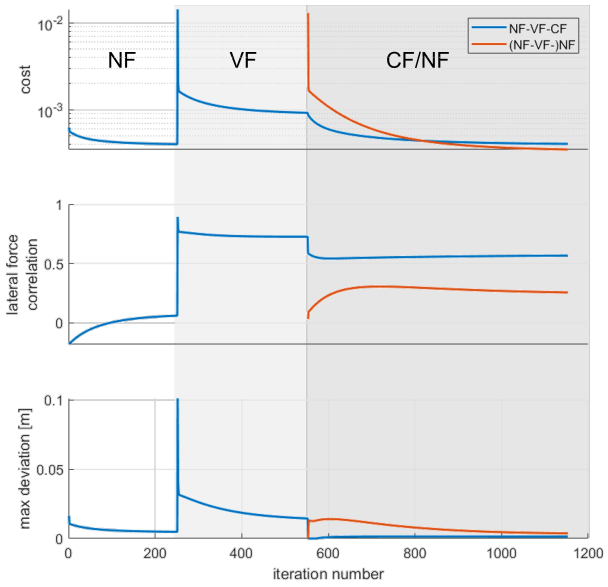



Fig. 16. Changes of the cost, the correlation coefficient of the lateral force (with the optimal lateral force in the VF2) and the maximum lateral deviation throughout the optimization in the CF. The blue line shows the result with the error clamp and the red line without. The training outcome for the red before trial 550 is identical to that for the blue line, thus is not shown.

upon reaching the goal. The effect of the feedback component can be observed from the simulated velocity profile as well. In Figure 8, the speed of the hand experienced a minor sudden change around 0.05s, which is the duration of the feedback delay. This is a clear sign of the participation of the feedback controller.

The activation profiles (Figure 11 and 14) are also similar to the actual EMG profiles from the experiments (see Appendix D). A discrepancy in the setup was that the reference trajectory in this simulation is traj-3, while in Thoroughman and Shadmehr (1999) is traj-4, and Heald et al. (2018) averaged over four directions. The trained activation profiles for the NF and the VF1 also resemble the experimental data from Franklin et al. (2003), except for the major activation recorded for Posterior Deltoid. In both the simulation and the experiment, it can be found that the activation level slightly increased in the VF1 (post-adaptation) compared to that in the NF. However, the activation level in the early phase of training seems rather unstable in the simulation. Since the desired trajectories stayed at the goal for another 0.375s for an evaluated standard trial, it can be regarded that it was mainly the feedback controller that contributed to the post-reaching correction. The two figures clearly indicate that this correction appeared to be an oscillatory motion around the target, seemingly resulting from the reciprocal activation of the agonist and antagonist muscle pairs. The activation of all muscles had never decreased below 5% in this second half of motion, which is the target co-activation level that Stroeve (1999a) intended to achieve by tuning the cost weight $\alpha$. With learning proceeding, the VF1 model could perform accurate reaching that no correction is needed, while even in the optima, the DF models failed to stop at the target.

However, not stopping at the target did not affect the evaluation of the DF sessions much, as trials that induced large lateral deviation had been excluded, and the motion upon reaching a 1 cm circle around the goal was not analysed in the experiment (Burdet et al., 2001). Most importantly, the training did offer a straight-line trajectory. From the performance perspective, further tuning the model and the task parameter might result in more stable posture holding once the hand reaches the goal. The DF trials were simulated after similar NF sessions. Due to the fact that each attempt of NF training started in different configurations, the resulting models are not identical to each other. The resulting metrics, the trajectories, and the dynamic profiles, nonetheless, suggested these models should fall in the same optima. Increased activation levels were found in the 0.375s post-adaptation motions. Although the profiles only resemble the experimental data (Franklin et al., 2003) to a limited degree, it showed that coactivation is one of the major features of the reaching phase of the motion. Since the activation profile converged toward a significantly higher value after the adaptation, the decrease in averaged muscle activation shown in Figure 13 is more likely due to the mitigated oscillation in the second half of the motion once the trajectorial tracking was improved. The endpoint error of the DF1 had a similar magnitude as the VF1 during and after training and was smaller than that of the DF2, which is also consistent with the data by Franklin et al. (2003), although the inversed exponential-shaped lower bound that appeared after 2000 trials in Figure 13 was not seen in the literature.

The model trained in the VF2, however, was not able to converge toward a certain activation profile. It was clear in its training convergence record (see Appendix D). With speculation, the cause of this result might be that the control strategy for a goal cannot be generalized toward other goals. In each trial, the model was updated to the optima of that very configuration of the desired trajectory. Such optima might be far away from the optima for other desired trajectories. Statistics based on the randomized NF and VF data sets show that the distance between the goals is positively relevant to the rise in the cost, but not as significant as the positions of the goals as shown in Figure 6. Therefore more likely it was because the feedback control behaviour played a more active role in the VF2 adaptation.

### B. *Impedance characteristics*

The impedance was mainly evaluated for the DF trials. It appears that the strategy the model used was to eliminate the trajectorial error so that the lateral force could be minimized and the impedance would therefore be reduced to roughly the pre-exposure level. When the endpoint deviated from the desired trajectory, the stiffness matrix changed toward the singularity. In this case, the stiffness ellipse can hardly be reasoned. Judging by Figure 14, the simulation showed that the model would eventually hover around the goal even before the designated movement time of 0.375 s ended. This indicates that the impedance captured in Figure 15 does not perfectly

represent the impedance profiles during the reaching motions. In this regard, the half-trained models provided less adequate and comparable impedance profiles than the fully trained ones, and the DF1 model should give better results than the one in the DF2. The calculation of dynamic impedance in the reflexive and empirical way linearized the force-position relation with reference to the desired trajectory (Stroeve, 1999b; Burdet et al., 2000). This explained the large distortion for the DF2 stiffness during the training.

It has been remarked that the intrinsic stiffness contributed to less than 10% of the reaction force (Burdet et al., 2000). Ignoring the magnitude of the impedance, the intrinsic stiffness per trial changes along the trajectory in a way that typical stiffness would (Stroeve, 1999b; Tee et al., 2004). The DF1 also induced enlarged reflexive and empirical stiffness in the direction of the force field. This agrees with what Burdet et al. (2001) observed. The trajectory in the 1400th trial in the DF1 (red) significantly deviated from the desired one when its empirical impedance was enlarged by a degree of magnitude.

Possibly because the intrinsic and reflexive stiffness profiles did not take the feedforward behaviour into account, the general magnitude of the impedance, and thus the sizes of the stiffness ellipses are much smaller than that in the literature. Although, from the opposite perspective, the reflexive stiffness might provide a good indication of how active the feedback controller was. The empirical approach by Tee et al. (2004) did not show much about the increased lateral stiffness without the additional term $\mathbf{K}_i$ (see Appendix B). It does produce good impedance profile for longitudinal movement both before and after the adaptation (see Appendix D). Since the figure excluded the cocontraction-induced impedance as it would be just artificially adding a major yet fixed-value horizontal stiffness to the already existing stiffness profile, it shows that the reciprocal contraction is only affected by the DFs to a limited level compared to the effect of the VF1. Although the change of empirical profile regarding the VF1 cannot be validated by experiment as this simulation result clearly approached singularity during the training and hardly returned to normal post-adaptation.

### C. Motor memory

The CF sessions are simulated with a deterministic task, meaning that all randomised factor was eliminated when the desired trajectory was computed, including the movement time, which was set to 750 ms. A better simulation would include not only the changing movement time but also slightly altered starting points and goals randomly assigned within a 1 cm radius from the traj-3 standard positions (Vaswani and Shadmehr, 2013). The error clamp's direction would have changed accordingly in order to stay perpendicular to the desired trajectory. Yet such randomization would induce much noise in the parameters such as the cost and the lateral fore correlation, making the reasoning of the result less straightforward.

For the CF trials, the learning curve represented by the lateral force correlation matches the experiment reported by Vaswani and Shadmehr (2013)in general. The index increased as the VF2 session progressed and dropped when the VF2 was removed. The transient appears to be exponential-like except for the first exposure in the VF2, likely to result from an utterly unstable trial that touched the edge of the workspace. Upon exposure to the CF after the brief training in the VF2, the decay of the correlation coefficient also did not return to the level that the experiment showed. The correlation coefficient decreased even faster than when it was exposed in the NF. This is likely to be because the learning based on the trajectorial error was effectively halted due to the error clamp and the learning only progressed for the activation sake, which is two orders of magnitude slower than the error-based learning. The delay of the decay was also not reflected by the simulation. It means that the model does not simulate the two-factor learning Smith et al. (2006).

### D. Future direction

Tuning the controller, in this case, means finding the optimal settings, including the initial weights and the learning rates so that the model can be properly optimized. Networks with more layers have been explored while the model was constructed in this project. No significant improvement in performance has been found. Nevertheless, the deep networks were only roughly tuned manually. Therefore, a well-optimised controller might still outperform the current MLP in use. Using adaptive or multiple learning rates might be a minor improvement. Although the standard decreasing $\eta$ worked modestly on the current model as its only effect was slowing down the training, applying different rates of learning on the two controllers or even on individual weights can be an alternative. The time-sensitive learning could also be an issue. Under the current batch update, each instance in a reaching movement contributes equally toward the gradient. Yet it is apparent that the reaching and holding position has different requirements for the feedforward and feedback controllers respectively. Investigation into this matter might result in increased learning efficiency or unnecessary complexity. Additionally, if the run time allows, the fourth-order Runge–Kutta method would have been a more up-to-date choice than an Euler integrator.

The use of randomized desired trajectories, including their movement time, directly affected the performance and can be shown with parameters such as cost. Once the tasks were defined with fixed variables, the convergence became deterministic, and the learning curves became smooth. With the goal being too close to the starting point, it is debatable if the motion still counts as reaching or not. The strategy for posture control can well be different from motion control and can involve different controllers or synergy. Defining a lower limit for the trajectory length can be an improvement to the task design. The tasks can also be redesigned to reflect the actual experiments more accurately. For instance, the experiment tasking humans to train in one direction (Vaswani and Shadmehr, 2013) introduced force fields that are not perfectly perpendicular to the y direction, forcing the subject to generalize over different movement directions. Although this angular deviation is small, such modification to the simulation can improve its fidelity.

The model is inherently based on a series of assumptions and simplifications. Most importantly, the largely linearized Hill's model. It simplifies the, particularly analytical, calculation complexity, making the gradient descent a viable way of training. This is common among the models that adopt supervised learning (Kawato et al., 1987; Haruno et al., 2001). Gradient-based learning rules work better with linearized models and provide limited capability to react to unstable dynamics. If more biologically plausible components are to be used in this model, the calculation of the Jacobian will be increasingly challenging, and alternative learning policies might be needed. Other models known to include Hill-type muscles either applied reinforcement learning (Kambara et al., 2009) or specific learning rules based on muscle dynamics (Franklin et al., 2007). Gründemann (2023) successfully implemented the Huxley model within a simple feedback control loop and trained it with the genetic algorithm. This proof-of-concept attempt can be potentially incorporated into a more sophisticated control system, possibly with NNs deployed.

## V. Conclusion

A model has been reconstructed, consisting of a combined feedback and feedforward controller, a Hill-type muscular model, and simplified skeletal dynamics. This model was first trained in the NF to perform point-to-point reaching tasks, achieving straight trajectories with bell-shaped velocity profiles. On top of this, the model successfully adapted to the VFs, managing to follow the desired trajectories and resulting in after effect upon the removal of the VFs. The model was able to adapt to the DFs, demonstrating increased contraction level and struggle in stablizing around the targets. The CF hindered the recovery of the behaviour after the removal of the VF. However, the speed of such decay and the absence of the delay suggests that additional mechanisms are required for this model to simulate the two-rate learning behaviour. The learning process showed that trajectory tracking was prioritized over decreasing muscle activation. The calculated stiffness profiles only roughly resemble the experimental results but successfully demonstrated the model's reaction to the DFs. The endpoint stiffness expanded during the training and gradually returned to the pre-training level after the trajectorial error had been eliminated. Overall, human motor learning can be simulated with this model in an imposed novel dynamics.

## References


Aprasoff, J. and Donchin, O. (2012). Correlations in state space can cause sub-optimal adaptation of optimal feedback control models. *Journal of Computational Neuroscience*, 32:297–307.

Burdet, E., Franklin, D., and Milner, T. (2013). *Human Robotics*. MIT Press.

Burdet, E., Osu, R., Franklin, D., Yoshioka, T., Milner, T., and Kawato, M. (2000). A method for measuring endpoint stiffness during multi-joint arm movements. *Journal of Biomechanics*, 33:1705–1709.

Burdet, E., Osu, R., Franklin, D. W., Milner, T. E., and Kawato, M. (2001). The central nervous system stabilizes unstable dynamics by learning optimal impedance. *Nature*, 414:446–449.

Burdet, E., Tee, K. P., Mareels, I., Milner, T. E., Chew, C. M., Franklin, D. W., Osu, R., and Kawato, M. (2006). Stability and motor adaptation in human arm movements. *Biological Cybernetics*, 94:20–32.

Crevecoeur, F., Scott, S. H., and Cluff, T. (2019). Robust control in human reaching movements: A model-free strategy to compensate for unpredictable disturbances. *The Journal of Neuroscience*, 39:8135–8148.

Donchin, O., Francis, J. T., and Shadmehr, R. (2003). Quantifying generalization from trial-by-trial behavior of adaptive systems that learn with basis functions: Theory and experiments in human motor control. *The Journal of Neuroscience*, 23:9032–9045.

Flash, T. and Hogan, N. (1985). The coordination of arm movements: an experimentally confirmed mathematical model. *The Journal of Neuroscience*, 5:1688–1703.

Franklin, D. and Wolpert, D. (2011). Computational mechanisms of sensorimotor control. *Neuron*, 72:425–442.

Franklin, D. W., Burdet, E., Osu, R., Kawato, M., and Milner, T. E. (2003). Functional significance of stiffness in adaptation of multijoint arm movements to stable and unstable dynamics. *Experimental Brain Research*, 151:145–157.

Franklin, D. W., Liaw, G., Milner, T. E., Osu, R., Burdet, E., and Kawato, M. (2007). Endpoint stiffness of the arm is directionally tuned to instability in the environment. *Journal of Neuroscience*, 27:7705–7716.

Gomi, H. and Kawato, M. (1997). Human arm stiffness and equilibrium-point trajectory during multi-joint movement. *Biological Cybernetics*, 76:163–171.

Gomi, H. and Osu, R. (1998). Task-dependent viscoelasticity of human multijoint arm and its spatial characteristics for interaction with environments. *The Journal of Neuroscience*, 18:8965–8978.

Gründemann, A. (2023). How muscle stiffness affects neural control parameters: Short-range stiffness improves stability and feedback robustness of musculoskeletal models. Master's thesis, Delft University of Technology.

Happee, R. and van der Helm, F. C. T. (1995). The control of shoulder muscles during goal directed movements, an inverse dynamic analysis. *Journal of Biomechanics*, 28:1179–1191.

Harris, C. M. and Wolpert, D. M. (1998). Signal-dependent noise determines motor planning. *Nature*, 394:780–784.

Haruno, M., Wolpert, D. M., Haruno, M., Wolpert, D., and Kawato, M. (2001). Mosaic model for sensorimotor learning and control. *Neural Computation*, 13:2201–2220.

Heald, J. B., Franklin, D. W., and Wolpert, D. M. (2018). Increasing muscle co-contraction speeds up internal model acquisition during dynamic motor learning. *Scientific Reports*, 8:16355.

Izawa, J., Rane, T., Donchin, O., and Shadmehr, R. (2008). Motor adaptation as a process of reoptimization. *Journal of Neuroscience*, 28:2883–2891.

Joiner, W. M. and Smith, M. A. (2008). Long-term retention explained by a model of short-term learning in the adaptive control of reaching. *Journal of Neurophysiology*, 100:2948–

2955.

Kambara, H., Kim, K., Shin, D., Sato, M., and Koike, Y. (2009). Learning and generation of goal-directed arm reaching from scratch. *Neural Networks*, 22:348–361.

Kambara, H., Takagi, A., Shimizu, H., Kawase, T., Yoshimura, N., Schweighofer, N., and Koike, Y. (2021). Computational reproductions of external force field adaption without assuming desired trajectories. *Neural Networks*, 139:179–198.

Kawato, M. (1999). Internal models for motor control and trajectory planning. *Current Opinion in Neurobiology*, 9:718–727.

Kawato, M., Furukawa, K., and Suzuki, R. (1987). A hierarchical neural-network model for control and learning of voluntary movement. *Biological Cybernetics*, 57:169–185.

Kawato, M., Uno, Y., Isobe, M., and Suzuki, R. (1988). Hierarchical neural network model for voluntary movement with application to robotics. *IEEE Control Systems Magazine*, 8:8–15.

Kooij, K., Overvliet, K. E., and Smeets, J. B. J. (2016). Temporally stable adaptation is robust, incomplete and specific. *European Journal of Neuroscience*, 44:2708–2715.

Lackner, J. R. and Dizio, P. (1994). Rapid adaptation to coriolis force perturbations of arm trajectory. *Journal of Neurophysiology*, 72:299–313.

Nishii, J. and Taniai, Y. (2009). Evaluation of trajectory planning models for arm-reaching movements based on energy cost. *Neural Computation*, 21:2634–2647.

Ohta, K., Svinin, M. M., Luo, Z., Hosoe, S., and Laboissière, R. (2004). Optimal trajectory formation of constrained human arm reaching movements. *Biological Cybernetics*, 91:23–36.

Razavian, R., Mehrabi, N., and McPhee, J. (2015). A model-based approach to predict muscle synergies using optimization: application to feedback control. *Frontiers in Computational Neuroscience*, 9.

Rigoux, L. and Guigon, E. (2012). A model of reward- and effort-based optimal decision making and motor control. *PLoS Computational Biology*, 8.

Shadmehr, R. and Mussa-Ivaldi, F. (1994). Adaptive representation of dynamics during learning of a motor task. *The Journal of Neuroscience*, 14:3208–3224.

Smith, M. A., Ghazizadeh, A., and Shadmehr, R. (2006). Interacting adaptive processes with different timescales underlie short-term motor learning. *PLoS Biology*, 4:e179.

Stroeve, S. (1997). A learning feedback and feedforward neuromuscular control model for two degrees of freedom human arm movements. *Human Movement Science*, 16:621–651.

Stroeve, S. (1998a). An analysis of learning control by backpropagation through time. *Neural Networks*, 11:709–721.

Stroeve, S. (1998b). Neuromuscular control model of the arm including feedback and feedforward components. *Acta Psychologica*, 100:117–131.

Stroeve, S. (1999a). Analysis of the role of proprioceptive information during arm movements using a model of the human arm. *Motor Control*, 3:158–185.

Stroeve, S. (1999b). Impedance characteristics of a neuromusculoskeletal model of the human arm i. posture control. *Biological Cybernetics*, 81:475–494.

Stroeve, S. (1999c). Impedance characteristics of a neuromusculoskeletal model of the human arm ii. movement control. *Biological Cybernetics*, 81:495–504.

Tee, K. P., Burdet, E., Chew, C. M., and Milner, T. E. (2004). A model of force and impedance in human arm movements. *Biological Cybernetics*, 90:368–375.

Thoroughman, K. A. and Shadmehr, R. (1999). Electromyographic correlates of learning an internal model of reaching movements. *The Journal of Neuroscience*, 19:8573–8588.

Thoroughman, K. A. and Shadmehr, R. (2000). Learning of action through adaptive combination of motor primitives. *Nature*, 407:742–747.

Todorov, E. and Jordan, M. I. (2002). Optimal feedback control as a theory of motor coordination. *Nature Neuroscience*, 5:1226–1235.

Ueyama, Y. (2014). Mini-max feedback control as a computational theory of sensorimotor control in the presence of structural uncertainty. *Frontiers in Computational Neuroscience*, 8.

Uno, Y., Kawato, M., and Suzuki, R. (1989). Formation and control of optimal trajectory in human multijoint arm movement. *Biological Cybernetics*, 61:89–101.

Vallery, H. and Schwab, A. L. (2020). *Advanced Dyanmics*. Stichting Newton-Euler, 3rd edition.

Vaswani, P. A. and Shadmehr, R. (2013). Decay of motor memories in the absence of error. *The Journal of Neuroscience*, 33:7700–7709.

Winters, J. M. and Stark, L. (1985). Analysis of fundamental human movement patterns through the use of in-depth antagonistic muscle models. *IEEE Transactions on Biomedical Engineering*, BME-32:826–839.

Wolpert, D. and Kawato, M. (1998). Multiple paired forward and inverse models for motor control. *Neural Networks*, 11:1317–1329.

## APPENDIX A
## THE MUSCULOSKELETAL MODEL

Stroeve (1999a) referred to Winters and Stark (1985) for the muscle model, which simplified the muscle activation to a first-order relation in reference to the excitation. The limb dynamic is calculated with the so-called TMT method summarized by Vallery and Schwab (2020).

### A. Muscular model

For the muscles, only the contractile element is included, with the force-velocity relation being Hill-type and the force-length relation being Gaussian. The muscle model takes $\mathbf{u}$ as the input, $\mathbf{a}$ as the state variable, and $\mathbf{F}$ as the output. The activation dynamics has:

$$\dot{a}_i = \frac{u_i - a_i}{\tau_i} \tag{19}$$

where $i \in [1, 6]$ and

$$\tau_i = \begin{cases} \tau_{ac} & u > a \\ \tau_{da} & u < a \end{cases}$$

$\tau_{ac} = 40ms$ and $\tau_{da} = 70ms$ are the time constants for activation and deactivation of the muscles, respectively.

The general expression for the muscle force for one muscle is:

$$\mathbf{F} = \mathbf{a}\mathbf{F}_{lce}\mathbf{F}_{vce}\mathbf{F}_{max} \tag{20}$$

$\mathbf{F}_{max}$ is the maximum force for the muscles, $\mathbf{F}_{lce}$ is the force-length relation, and $\mathbf{F}_{vce}$ is the force-velocity relation. The muscle length can be derived from the joint angle:

$$\mathbf{l}_m = \mathbf{l}_r - \sum_{j=1}^{2} \mathbf{r}_j \left(\theta_{\mathbf{j}} - \theta_{\mathbf{rest,j}}\right) \tag{21}$$

$$\dot{\mathbf{l}}_m = -\sum_{j=1}^{2} \mathbf{r}_j \dot{\theta}_{\mathbf{j}} \tag{22}$$

$$\mathbf{l}_{ce} = \mathbf{l}_m - \mathbf{l}_t \tag{23}$$

$$\dot{\mathbf{l}}_{ce} = \dot{\mathbf{l}}_m \tag{24}$$

Where $\mathbf{r}$ is the moment of arm, $\theta_{rest}$ is the rest position of the arm, which is not necessarily where the muscles are at the rest lengths, $\mathbf{l}_t$ is the tendon length. Substituting $\theta$ in equation 21 with $\theta_{max}$ or $\theta_{min}$ leads to $l_{max}$ or $l_{min}$ per muscle. With these terms derived, for $\mathbf{F}_{lce}$ we have:

$$\mathbf{F}_{lce} = e^{-\left(\frac{\mathbf{l}_{ce}-\mathbf{l}_{ce0}}{\mathbf{l}_{cesh}}\right)^2} \tag{25}$$

where

$$\mathbf{l}_{ce0} = \mathbf{l}_{min} + L_{opt}\left(\mathbf{l}_{max} - \mathbf{l}_{min}\right) - \mathbf{l}_t \tag{26}$$

$$\mathbf{l}_{cesh} = L_{sh}\left(\mathbf{l}_{max} - \mathbf{l}_{min}\right) \tag{27}$$

$$\tag{28}$$

For $\mathbf{F}_{vce}$:

$$\mathbf{F}_{vce} = \begin{cases} 0 & \dot{\mathbf{l}}_{ce} \leq -\mathbf{v}_{max} \\ \frac{V_{sh}\left(\mathbf{v}_{max}+\dot{\mathbf{l}}_{ce}\right)}{V_{sh}\mathbf{v}_{max}-\dot{\mathbf{l}}_{ce}} & -\mathbf{v}_{max} < \dot{\mathbf{l}}_{ce} \leq 0 \\ \frac{V_{sh}V_{shl}\mathbf{v}_{max}+V_{ml}\dot{\mathbf{l}}_{ce}}{V_{sh}V_{shl}\mathbf{v}_{max}+\dot{\mathbf{l}}_{ce}} & \dot{\mathbf{l}}_{ce} > 0 \end{cases} \tag{29}$$

where

$$\mathbf{v}_{max} = V_{vm}\left(1 - V_{er}\left(1 - \mathbf{a}\dot{\mathbf{l}}_{ce}\right)\right) \tag{30}$$

The values of all constants can be found in Stroeve (1999a).

### B. Skeletal model

The TMT method is used to perform Newton's second law in a generalized coordinate, which is, in this case, the joint space. Thus we have:

$$\ddot{\mathbf{q}} = \mathbf{M}_{red}^{-1}\mathbf{F}_{red} \tag{31}$$

where $\ddot{\mathbf{q}} = \left[\ddot{\theta}_1, \ddot{\theta}_2\right]$ for $\mathbf{q} = [\theta_1, \theta_2]$, $\mathbf{M}_{red}$ is the reduced mass matrix and $\mathbf{F}_{red}$ is the reduced force vector, namely the resultant forces imposed on the system projected to this local space. With $\mathbf{T} = [x_{m1}, y_{m1}, x_1, y_1, x_{m2}, y_{m2}, x_2, y_2, \theta_1, (\theta_1 + \theta_2)]$,

$$\mathbf{F}_{red} = \frac{\partial \mathbf{T}}{\partial \mathbf{q}}\left(\mathbf{F}_{ext} - \mathbf{M}\mathbf{g}_{conv}\right) + \mathbf{T}_{pas} + \mathbf{T}_{mus} \tag{32}$$

and

$$\mathbf{M}_{red} = \frac{\partial \mathbf{T}}{\partial \mathbf{q}}\mathbf{M}\frac{\partial \mathbf{T}}{\partial \mathbf{q}} \tag{33}$$

where $x$ and $y$ are the position of the end of the limbs, $x_m$ and $y_m$ are the positions of their centres of mass. $\mathbf{M}$ is the mass matrix that has the same number of dimensions as $\mathbf{T}$. $\mathbf{F}_{ext}$ is the external force in Cartesian coordinate, $\mathbf{g}_{conv}$ is the convective term in TMT. $\mathbf{T}_{pas}$ is the passive torques which is treated as a univariate function of time. $\mathbf{T}_{mus}$ is the resultant muscle torques in the joint space. The expressions follows:

$$\mathbf{g}_{conv} = \frac{\partial \frac{\partial \mathbf{T}}{\partial \mathbf{q}}\dot{q}}{\partial q}\dot{q} \tag{34}$$

$$\mathbf{T}_{pas} = -\mathbf{B}\dot{\theta} - sgn\left(\theta - \theta_{rest}\right)\frac{\mathbf{T}_{max}}{e^{\mathbf{PE}_{sh}-1}}\left(e^{\frac{\mathbf{PE}_{sh}}{\mathbf{PE}_{xm}}|\theta-\theta_{rest}|-1}\right) \tag{35}$$

$$\mathbf{T}_{mus,j} = \sum_{i=1}^{6} \mathbf{F}_i \mathbf{r}_{ij} \tag{36}$$

where $\mathbf{B}$ is the joint damping constant, $\mathbf{PE}_{sh}$ and $\mathbf{PE}_{sm}$ are the shape factors for the passive torques, and $\mathbf{r}_{ij}$ is the rotational moment of arm of muscle $i$ on joint $j$.

## APPENDIX B
## MODEL SENSITIVITY AND BACKPROPAGATION

### A. Backpropagation through time

The batch update of the weight used in the controller is described as

$$\mathbf{W}_{i,new} = \mathbf{W}_{i,old} - \eta\Delta\mathbf{W}_i \tag{5 revisited}$$

$$\Delta\mathbf{W}_i = \frac{1}{N}\sum_{k=\mathbf{0}}^{N} \nabla_{\mathbf{W}_i} J\left(k\right) \tag{37}$$

for $i = 1, 2$ being the MLP network layer number, k is the discrete instance number, and N is the time vector's length. Here the BTT can be taken into two parts: the neural network part and the musculoskeletal part:

$$\nabla_{\mathbf{W}_i} J\left(k\right) = \frac{\partial \mathbf{u}}{\partial \mathbf{W}_i}\nabla_{\mathbf{u}} J\left(k\right) \tag{38}$$

At any given time in a trial, the equations of the backpropagation for the neural netwrok part of the system stay the same. This is the partial derivative of equation 1 and its sub-function. Here we only include the 1-hidden-layer case:

$$\frac{\partial \mathbf{u}}{\partial \mathbf{W}_1} = \boldsymbol{\phi}\frac{\mathbf{z}_1}{1-\mathbf{z}_1}\mathbf{W}_2\frac{\mathbf{u}}{1-\mathbf{u}} \tag{39}$$

$$\frac{\partial \mathbf{u}}{\partial \mathbf{W}_2} = \mathbf{a}_2\frac{\mathbf{u}}{1-\mathbf{u}} \tag{40}$$

where $\mathbf{z}_1$ is the output signals of the hidden layer and $\mathbf{a}_2$ is the input signals to the output layer, the latter has one more bias term than the former. Calculating the other term in equation 38, requires iterative execution of equation 7, equation 8 and equation 9:

$$\frac{\partial^+ J}{\partial x_j(k)} = \frac{\partial l(k)}{\partial x_j(k)} + \sum_{i=1}^{p} \frac{\partial l(k)}{\partial y_i(k)} \frac{\partial y_i(k)}{\partial x_j(k)} + C_1 \sum_{i=1}^{n} \frac{\partial^+ J(k)}{\partial x_i(k+1)} \left( C_2 + h \frac{\partial f_i}{\partial x_j}(\mathbf{x}(k), \mathbf{u}(k)) \right) \quad (7\ 8, \text{revisited})$$

$$\frac{\partial^+ J}{\partial u_j(k)} = \frac{\partial l(k)}{\partial u_j(k)} + \sum_{i=1}^{p} \frac{\partial l(k)}{\partial y_i(k)} \frac{\partial y_i(k)}{\partial u_j(k)} + \sum_{i=1}^{n} \frac{\partial^+ J(k)}{\partial x_i(k+1)} h \frac{\partial f_i}{\partial u_j}(\mathbf{x}(k), \mathbf{u}(k)) \quad (9, \text{revisited})$$

where $C_1 = 0$ when $k = N$, otherwise $C_1 = 1$; $C_2 = 1$ when $i = j$, otherwise $C_2 = 0$. Consistent with equation 3:

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \\ \mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t)) \end{cases} \quad (3, \text{revisited})$$

we define the system states as follows:

$$\mathbf{x} = \left[ a_1, a_2, a_3, a_4, a_5, a_6, \theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2 \right]$$
$$\mathbf{y} = \left[ F_1, F_2, F_3, F_4, F_5, F_6, \theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2 \right]$$
$$\mathbf{u} = [u_1, u_2, u_3, u_4, u_5, u_6]$$

For $J$ being discretized into $l$:

$$J = \sum_{k=\mathbf{0}}^{N} l(\mathbf{x}(t), \mathbf{y}(t), \mathbf{u}(t)) \quad (41)$$

we have:

$$\frac{\partial l}{\partial \mathbf{x}} = [2\mathbf{F}_{max} \alpha \mathbf{a} \mathbf{l}_r, 2e_1, 2e_2, 2\dot{e}_1, 2\dot{e}_2] \quad (42)$$
$$\frac{\partial l}{\partial \mathbf{y}} = [\mathbf{0}_{1\times 6}, 2e_1, 2e_2, 2\dot{e}_1, 2\dot{e}_2] \quad (43)$$
$$\frac{\partial l}{\partial \mathbf{u}} = \mathbf{0}_{1\times 6} \quad (44)$$

where $e$ is the trajectorial error in the joint space and $\dot{e}$ is its time derivative.

### B. Sensitivity model

Now the only terms that remain are $\frac{\partial \mathbf{f}}{\partial \mathbf{x}}$, $\frac{\partial \mathbf{f}}{\partial \mathbf{u}}$, $\frac{\partial \mathbf{g}}{\partial \mathbf{x}}$, and $\frac{\partial \mathbf{g}}{\partial \mathbf{u}}$. These are the jacobians of the musculoskeletal system. For $\mathbf{s} = \left[ \theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2 \right]$ and therefore $\dot{\mathbf{s}} = \left[ \dot{\theta}_1, \dot{\theta}_2, \ddot{\theta}_1, \ddot{\theta}_2 \right]$, the matrices are:

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial \dot{\mathbf{a}}}{\partial \mathbf{a}} & \mathbf{0}_{6\times 4} \\ \frac{\partial \dot{\mathbf{s}}}{\partial \mathbf{a}} & \frac{\partial \dot{\mathbf{s}}}{\partial \mathbf{s}} \end{bmatrix} \quad (45)$$

$$\frac{\partial \mathbf{f}}{\partial \mathbf{u}} = \begin{bmatrix} \frac{\partial \dot{\mathbf{a}}}{\partial \mathbf{u}} \\ \mathbf{0}_{4\times 6} \end{bmatrix} \quad (46)$$

$$\frac{\partial \mathbf{g}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial \mathbf{F}}{\partial \mathbf{a}} & \frac{\partial \mathbf{F}}{\partial \mathbf{s}} \\ \mathbf{0}_{4\times 6} & \mathbf{I}_{4\times 4} \end{bmatrix} \quad (47)$$

$$\frac{\partial \mathbf{g}}{\partial \mathbf{u}} = \begin{bmatrix} \mathbf{0}_{10\times 6} \end{bmatrix} \quad (48)$$

Note that although the external forces $\mathbf{F}_{ext}$ are included in the vector $u$ in Stroeve (1999b), which means that the terms $\frac{\partial \mathbf{f}}{\partial \mathbf{F}_{ext}}$ and $\frac{\partial \mathbf{g}}{\partial \mathbf{F}_{ext}}$ are included in the sensitivity matrices $\frac{\partial \mathbf{f}}{\partial \mathbf{u}}$ and $\frac{\partial \mathbf{g}}{\partial \mathbf{u}}$, respectively, these terms are not part of $\frac{\partial J}{\partial \mathbf{u}}$ as they are not explicitly a function of $u$ and thus are ignored in this paper since the external forces are treated as black boxes.

Inside the jacobians, the muscle sensitivity models are: $\frac{\partial \dot{\mathbf{a}}}{\partial \mathbf{a}}$, $\frac{\partial \dot{\mathbf{a}}}{\partial \mathbf{u}}$, $\frac{\partial \dot{\mathbf{a}}}{\partial \mathbf{s}} = \mathbf{0}_{6\times 4}$, $\frac{\partial \dot{\mathbf{F}}}{\partial \mathbf{a}}$, $\frac{\partial \mathbf{F}}{\partial \mathbf{u}} = \mathbf{0}_{6\times 4}$, and $\frac{\partial \dot{\mathbf{F}}}{\partial \mathbf{s}}$,of which, the first two are diagonal matrices that:

$$\frac{\partial \dot{\mathbf{a}}_i}{\partial \mathbf{a}_i} = -\frac{1}{\tau_i} \quad (49)$$
$$\frac{\partial \dot{\mathbf{a}}_i}{\partial \mathbf{u}_i} = \frac{1}{\tau_i} \quad (50)$$

with $\tau_i$ being either the activation constant or deactivation constant of muscle $i \in [1, 6]$ depending on its current activation level $a_i$ and the value of $u_i$. $\frac{\partial \mathbf{F}}{\partial \mathbf{a}}$ is a 6-by-6 diagonal matrix. $\frac{\partial \mathbf{F}}{\partial \mathbf{s}}$ is converted from the muscle space to the joint space in Stroeve (1999b) when calculating the jacobian. For the simplicity of calculation, in this paper, both the muscles and the skeletal dynamics are represented in joint space since the muscle lengths and the joint angles, along with their time derivatives, are mutually interchangeable.

The rest six terms from the four derivatives make the limb dynamics. These are $\frac{\partial \dot{\mathbf{s}}}{\partial \mathbf{a}}$, $\frac{\partial \dot{\mathbf{s}}}{\partial \mathbf{s}}$, $\frac{\partial \dot{\mathbf{s}}}{\partial \mathbf{u}} = \mathbf{0}_{4\times 6}$, $\frac{\partial \mathbf{s}}{\partial \mathbf{a}} = \mathbf{0}_{4\times 6}$, $\frac{\partial \mathbf{s}}{\partial \mathbf{s}} = \mathbf{I}_{4\times 4}$, and $\frac{\partial \mathbf{s}}{\partial \mathbf{u}} = \mathbf{0}_{4\times 6}$. Here,

$$\frac{\partial \dot{\mathbf{s}}}{\partial \mathbf{a}} = \frac{\partial \dot{\mathbf{s}}}{\partial \mathbf{T}} \frac{\partial \mathbf{T}}{\partial \mathbf{F}} \frac{\partial \mathbf{F}}{\partial \mathbf{a}} \quad (51)$$
$$\frac{\partial \dot{\mathbf{s}}}{\partial \mathbf{s}} = \frac{\partial \dot{\mathbf{s}}}{\partial \mathbf{s}_{direct}} + \frac{\partial \dot{\mathbf{s}}}{\partial \mathbf{T}} \frac{\partial \mathbf{T}}{\partial \mathbf{F}} \frac{\partial \mathbf{F}}{\partial \mathbf{s}} \quad (52)$$

where $\mathbf{T}$ is the two joint torques, $\mathbf{F}$ is the six muscle forces, and $\frac{\partial \dot{\mathbf{s}}}{\partial \mathbf{s}_{direct}}$ is the direct derivative of the joint angular acceleration over the states (since $\frac{\partial \theta}{\partial \mathbf{s}} = \mathbf{0}_{1\times 4}$).

More analytical details for all the above-mentioned expressions can be found in Stroeve (1999b). For this project, these terms were symbolically derived using MATLAB. Although the software can in theory provide the lumped function between the neural command and the joint acceleration (and its derivative), it takes a long time used to execute such an algorithm. As a result, deriving the mathematical expressions at the above-provided level seems to balance the programming effort and the computer run-time the best.

### C. Impedance

*1) Intrinsic stiffness:* The equation for the intrinsic stiffness is per definition

$$\mathbf{Z}_k = \frac{d\mathbf{F}_{end}}{d\mathbf{p}_{end}} \quad (18, \text{revisited})$$

As mentioned $\mathbf{F}_{end}$, in the Cartesian space, is composed of two parts, thus

$$\mathbf{Z}_k = \frac{d\mathbf{F}_{pas}}{d\mathbf{p}_{end}} + \frac{d\mathbf{F}_{mus}}{d\mathbf{p}_{end}} - \frac{\partial^2 \mathbf{F}_{end}}{\partial \mathbf{T} \partial \mathbf{q}} \quad (53)$$

Breaking down this expression we have

$$\frac{d\mathbf{F}_{pas}}{d\mathbf{p}_{end}} = \frac{\partial \mathbf{F}_{pas}}{\partial \mathbf{T}_{pas}} \frac{\partial \mathbf{T}_{pas}}{\partial \mathbf{s}} \frac{\partial \mathbf{s}}{\partial \mathbf{p}_{end}} \tag{54}$$

$$\frac{d\mathbf{F}_{mus}}{d\mathbf{p}_{end}} = \frac{\partial \mathbf{F}_{mus}}{\partial \mathbf{T}_{mus}} \frac{\partial \mathbf{T}_{mus}}{\partial \mathbf{F}} \frac{\partial \mathbf{F}}{\partial \mathbf{s}} \frac{\partial \mathbf{s}}{\partial \mathbf{p}_{end}} \tag{55}$$

where

$$\frac{\partial \mathbf{F}_{pas}}{\partial \mathbf{T}_{pas}} = \frac{\partial \mathbf{F}_{mus}}{\partial \mathbf{T}_{mus}} = \frac{\partial \mathbf{q}}{\partial \mathbf{p}_{end}} \tag{56}$$

that

$$\frac{\partial \mathbf{s}}{\partial \mathbf{p}_{end}} = \begin{bmatrix} \frac{\partial \mathbf{q}}{\partial \mathbf{p}_{end}} \\ \mathbf{0}_{2\times 2} \end{bmatrix} \tag{57}$$

The term $\frac{\partial \mathbf{q}}{\partial \mathbf{p}_{end}}$ describes the jacobian of the inversed kinematics of the skeletal system, which should be the inverse of the forward kinematics jacobian according to the inverse function theorem. The other terms in equation 54 and 55 are available upon BTT being calculated. One assumption made here is that the muscle dynamics won't change at a given state from lengthening to shortening, and the effect of $v_{max}$ is neglected (Stroeve, 1999b).

*2) Reflexive stiffness:* The stiffness induced by the feedback loop is essentially calculated by wiring the system in equation 3 into a feedback loop (Stroeve, 1999b):

$$\mathbf{u}_1 = \mathbf{G}\mathbf{y}_1 (t - \tau) \tag{58}$$

Where

$$\begin{aligned} \mathbf{G} &= \mathbf{k}_{fb} \frac{\partial \mathbf{u}}{\partial \mathbf{h}_2} \\ &= \mathbf{k}_{fb} \mathbf{W}_1 \frac{\mathbf{z}_1}{1 - \mathbf{z}_1} \mathbf{W}_2 \frac{\mathbf{u}}{1 - \mathbf{u}} \end{aligned} \tag{59}$$

is the transfer function for the controller, which can be found in the MLP backpropagation, $\mathbf{k}_{bf}$ is the feedback gain, $\mathbf{h}_2 = [\mathbf{s}, \mathbf{F}]$ are the loop feedback and $\tau = 0$ for simplicity. If the delay is to be considered in this case, a Pate filter should be used. The stiffness is expressed by the inverse of the transfer function of this feedback loop.

*3) Empirical stiffness:* With the empirical model (Tee et al., 2004; Burdet et al., 2013) basedon the experimental results (Burdet et al., 2000, 2001; Franklin et al., 2003), the stiffness can as well be calculated as a function as joint torques, given the external force is known. The torques is calculated as

$$\begin{aligned} \mathbf{T} &= \mathbf{T}_{ext} + \mathbf{T}_{mus} \\ &= \frac{\partial \mathbf{T}}{\partial \mathbf{q}} \mathbf{F}_{ext} + \mathbf{T}_{mus} \end{aligned} \tag{60}$$

then the joint stiffness

$$\mathbf{K}_{joint} = \begin{bmatrix} 10.8 + 3.18|\tau_1| & 2.83 + 2.15|\tau_2| \\ 2.51 + 2.34|\tau_2| & 8.67 + 6.18|\tau_2| \end{bmatrix} \tag{61}$$

One additional increase of stiffness was assumed to account for the effort to counter the effect of the DF. This term $\mathbf{K}_i$ is equal to the DF stiffness presented in equation 15 and 16. With $\mathbf{F}_{ext}$ and $\mathbf{K}_i$ substituted by 0, we have the static endpoint stiffness. Adding $\mathbf{F}_{ext}$, the stiffness reflects the reciprocal activation. $\mathbf{K}_i$ represents coactivation. $\mathbf{K}_{joint}$ can be transformed to Cartesian space in the same fashion of equation 54, then

$$\mathbf{Z}_{empirical} = \mathbf{K}_{joint,cart.} + \mathbf{K}_i \tag{62}$$

## APPENDIX C
## ALGORITHM DEVELOPMENT

The construction of the model started at the musculoskeletal dynamics. Torque input was used to test the skeletal dynamics by observing if the kinematic was reasonable regarding the input given. Then the muscles were added and simple flexion and extension could be performed. Once the physical part of the system was convincingly good (later the result was cross checked against the model by Gründemann (2023)), controller was added, starting with a feedback controller and PID. The result was far from adequate as the model was full of bugs and suboptimal coding, which dragged the run speed down to a few seconds per trial.

Without prior success in controlling the model, the MLP wsa still written into the model with the feedback loop ignored. MATLAB does offer a few reinforcement learning packages and NN examples. However, these resources were not volatile enough to be integrated into this project, which is, as mentioned, the downside of using MATLAB instead of Python. The latter offers a variety of machine learning tools, while the former does not. The reason for choosing MATLAB is the numerical integration and the management of simulation, in general, was largely empowered by Simulink. The MLP was as easy as two lines of code but learning math about forward and back propagation took some time. The BTT algorithm was implemented in two ways. One was by keying all the derived equations directly into the code, the other by presenting all forward dynamics in the symbolic form and allowing the computer to automatically inverse the math by calculating the Jacobians. The former method was much faster during the simulation but it was rather hard to ensure all code was correct and well maintained. The other option, however, was extremely slow. Eventually, the final solution stood on the middle ground. The BTT algorithm was broken down into several small yet tedious computational tasks to be machine-derived, then these tasks were manually assembled into the Jacobian of the entire musculoskeletal system.

It had been so far fairly fast and straightforward to have all the components added to the model, yet after all syntax errors were eliminated, the removal of all logical errors from the code has proven to be the most grinding and time-consuming part of the project. First, the saturation problem in the neural signals was solved by limiting the initial weight of the controllers. Then the overly large workspace constraint was reduced with the border slightly expanded to avoid the desired trajectory reaching outside the domain. The identification of the position of the errors was the main hurdle. Combined with yet-to-be-optimized run speed, the progress was slow, although some common issues had been tackled, such as avoiding for-loops. In order to find the error, the model was again simplified with the temporary removal of the controller and later the removal of the muscle, as nothing was found but a few wrong constants. The skeletal dynamics as first confirmed to be correct since, with the absence of the muscles, the desired torque could be calculated. The torque drove the hand into the desired trajectories. Two major problems were eventually found. The main bug was the wrong propagation direction

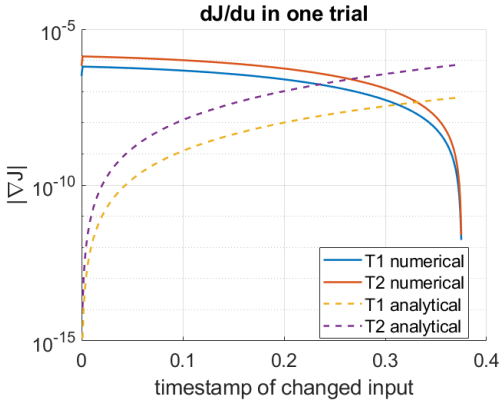



Fig. 17. The comparison between the numerical differentiation and the analytical derivatives. Each point in the numerical curve was calculated by observing the change of $J$ when a unit impulse of $T$ was sent to the system at the respective time. The solid and dash lines have similar values yet close-to-symmetric profiles. This is a strong indication that the propagation direction was wrong.

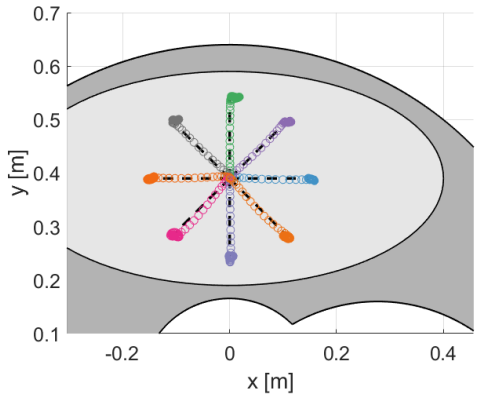
of BBT (Figure 17). This was the reason why the optima did not result in straight-line trajectories. This was found by comparing the numerical differentiation of the system ($\frac{\partial J}{\partial T_{1,2}}$) and its analytical derivation result (namely the BTT). After the BTT was corrected, the model was to be re-complicated. The muscle was given it's individual s-function so that the model reflects Stroeve's derivation more closely. This is where the other bug appeared. The issue was that there was an algebraic loop in the Simulink model. This was identified with the inbuilt profiler, as the output sub-functions in the s-functions were called an abnormally larger number of times.

Alongside numerous minor bug-fixing, further efforts to complete the model mainly included the introduction of the activation penalty, the force feedback, the external forces, multiple (standard and randomized) desired trajectories and so on. Simultaneously, tuning the controller has always been a problem, especially when the controller was expanded into deep learning NN. This was only made easier after the bug and the run time issues were fixed so that fast and relatively reliable simulations could be executed to find a good configuration with trial and error. This mainly regarded the learning rate $\eta$, the initial weights and the cost activation weight $\alpha$.

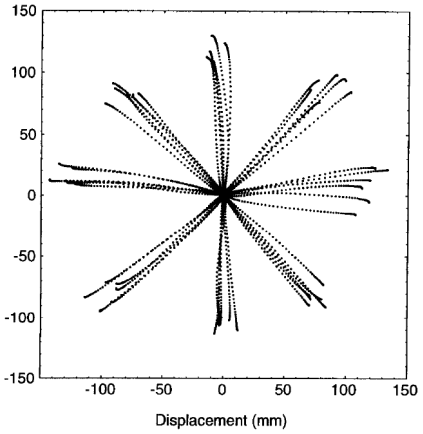
The task design was relatively easy once the model was decently working. With a good computer, the run speed was risen up to roughly 0.1s per second, which is close to the ideal speed expected. Most of the evaluation metrics were straightforward. Only it seems rather challenging to calculate the impedance properly.

(See next page for Appendix D)

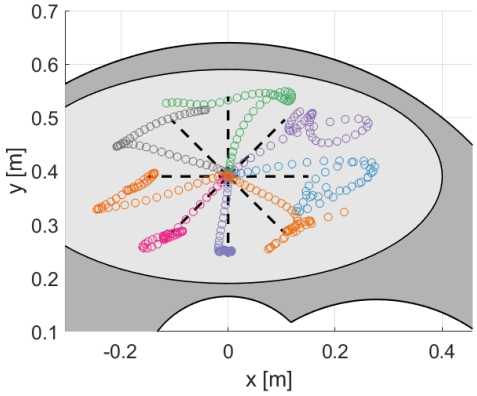
# APPENDIX D
## SUPPLEMENTARY FIGURES



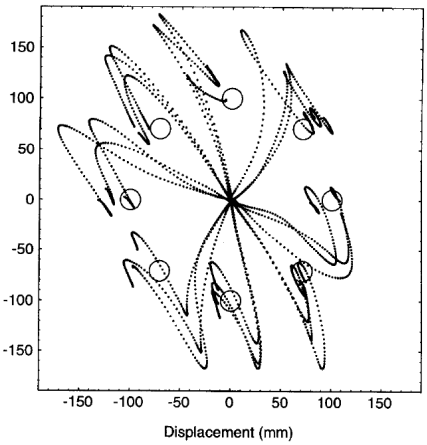

(a) Simulation result, trained in the NF.



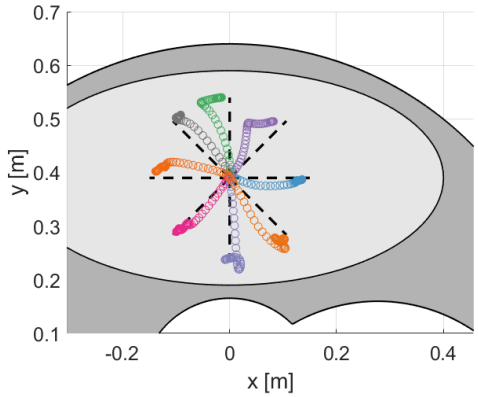

(b) Experiment, trained in the NF.



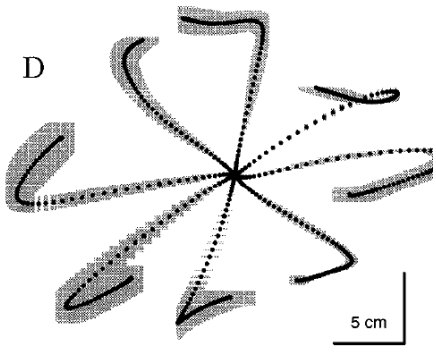

(c) Simulation result, first exposure in the VF1.




(d) Experiment, first exposure in the VF.

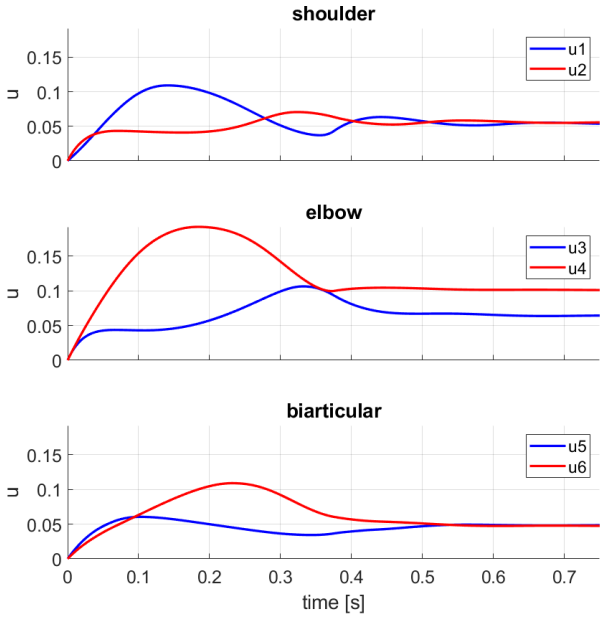



(e) Simulation result, after effect of model trained in the VF1

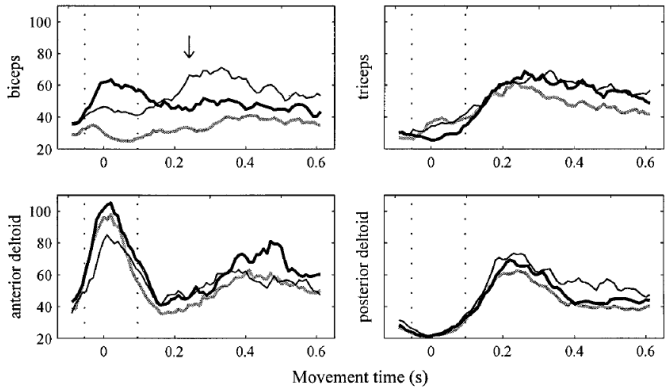



(f) Experiment, after effect of model trained in the VF.

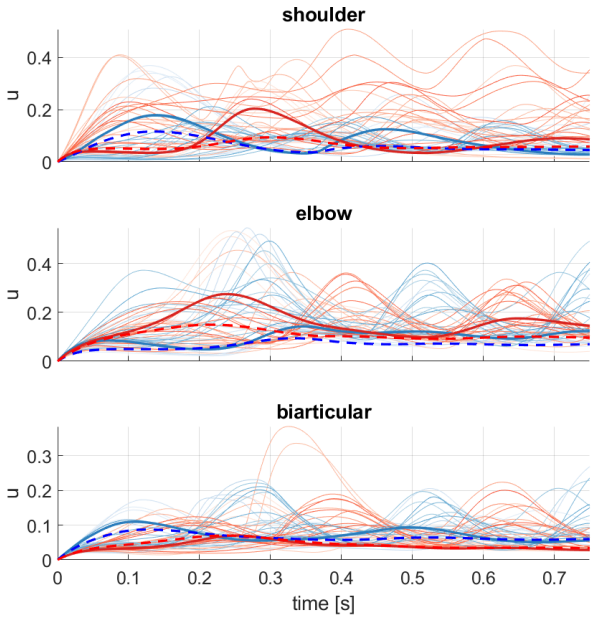
Fig. 18. Comparison between the simulated trajectories and the experiment by Shadmehr and Mussa-Ivaldi (1994), from which the figures are adopted.



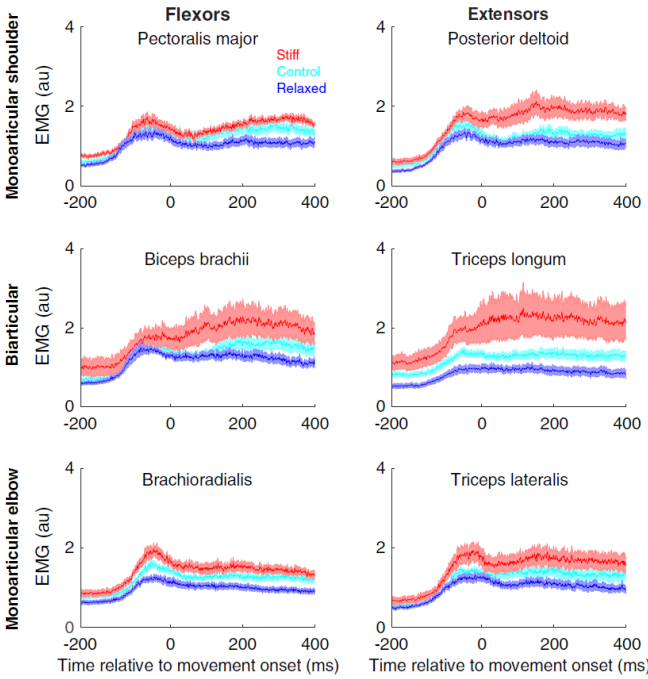

(a) Simulation result, trained in the NF.




(b) Experiment in the NF (Thoroughman and Shadmehr, 1999). The grey line is the EMG data in the NF based on traj-4, the thin black lines stand for pre-adaptation VF EMG data, and the thick black lines stand for the post-adaptation VF EMG data.

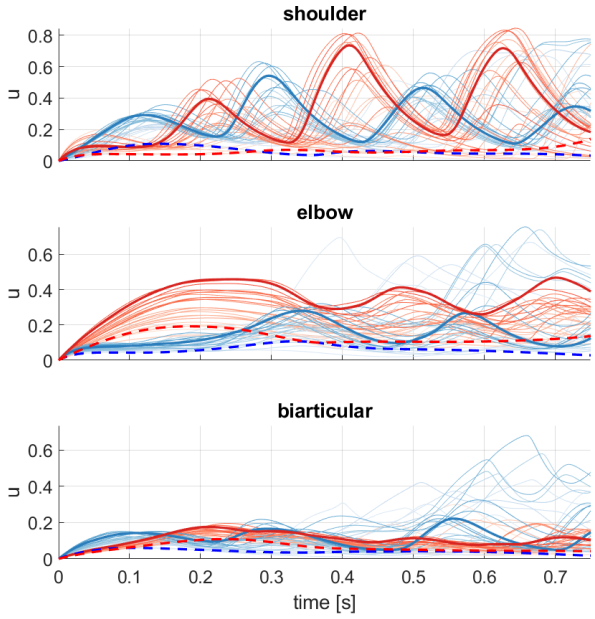



(c) Simulation result, in the VF1.

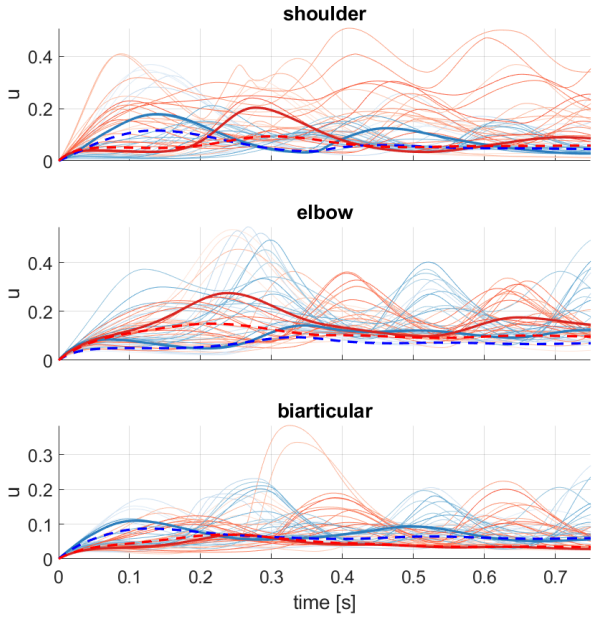



(d) Experiment in the VF (Heald et al., 2018). The profile is averaged over four directions.

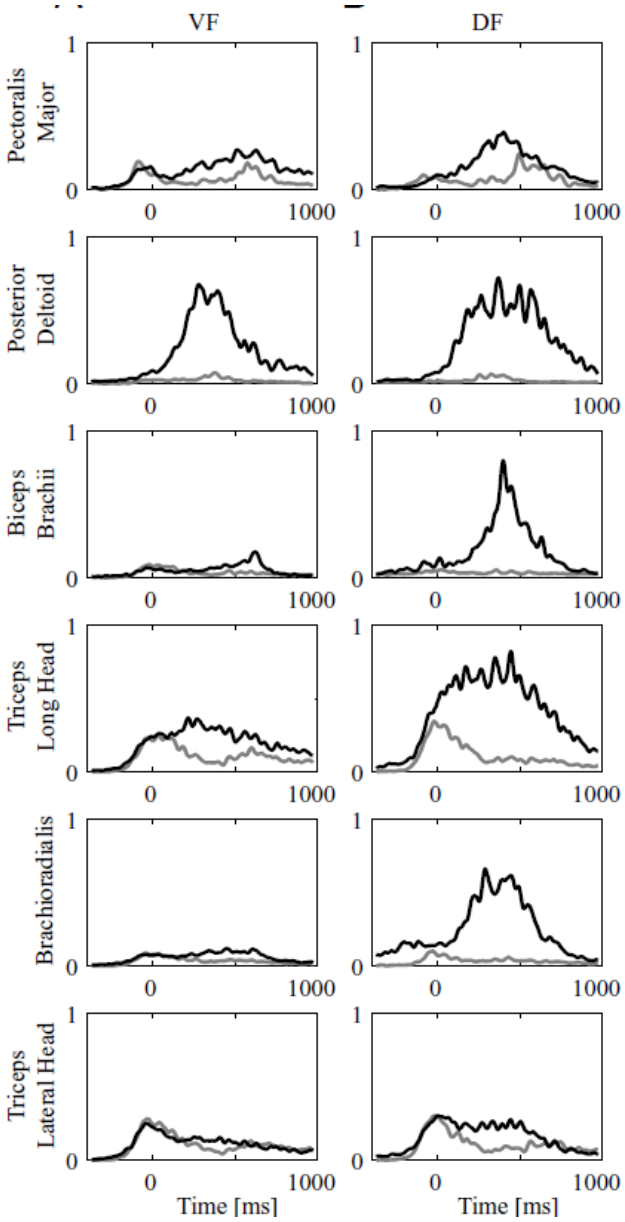
Fig. 19. Comparison of the activation profiles in the NF and the VF between the simulation and the experiments, from which the figures are adopted.




(a) Simulation result, trained in the DF1.




(b) Simulation result, trained in the VF2.




(c) Experiment in the VF1 (Franklin et al., 2003). The EMG levels in the NF are represented by the grey lines.

Fig. 20. Comparison of the activation profiles in the VF2 and DF between the simulation and the experiments (NF cases also included in Franklin's figure), from which the figures are adopted.

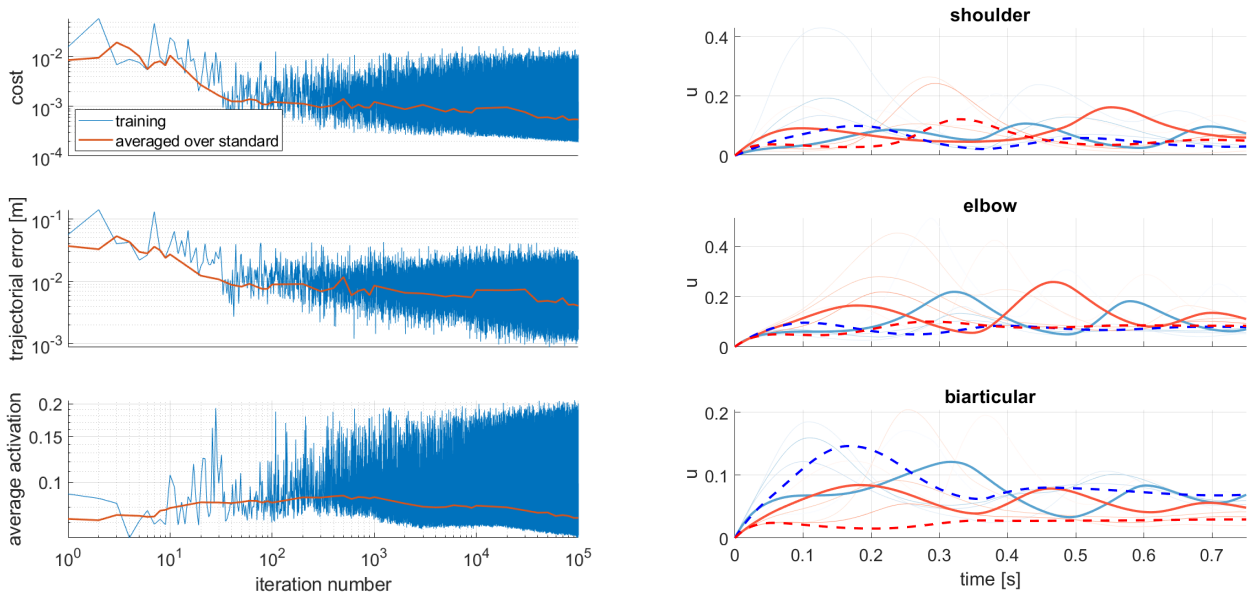



Fig. 21. Left: the tracking of the VF2 training process and its evaluation over the standard trajectories; Right: activation profiles of the VF2 training

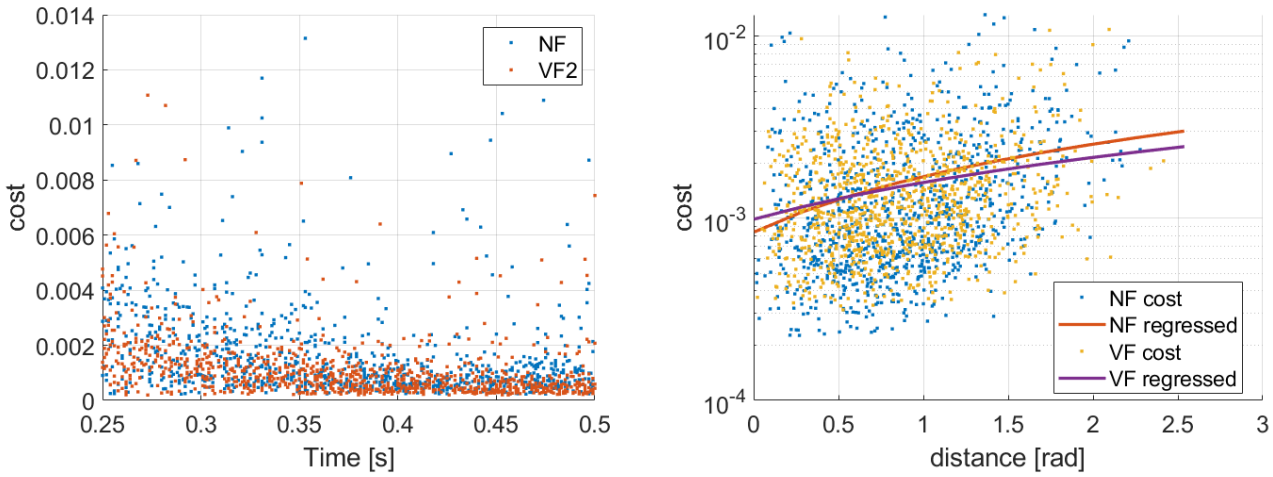
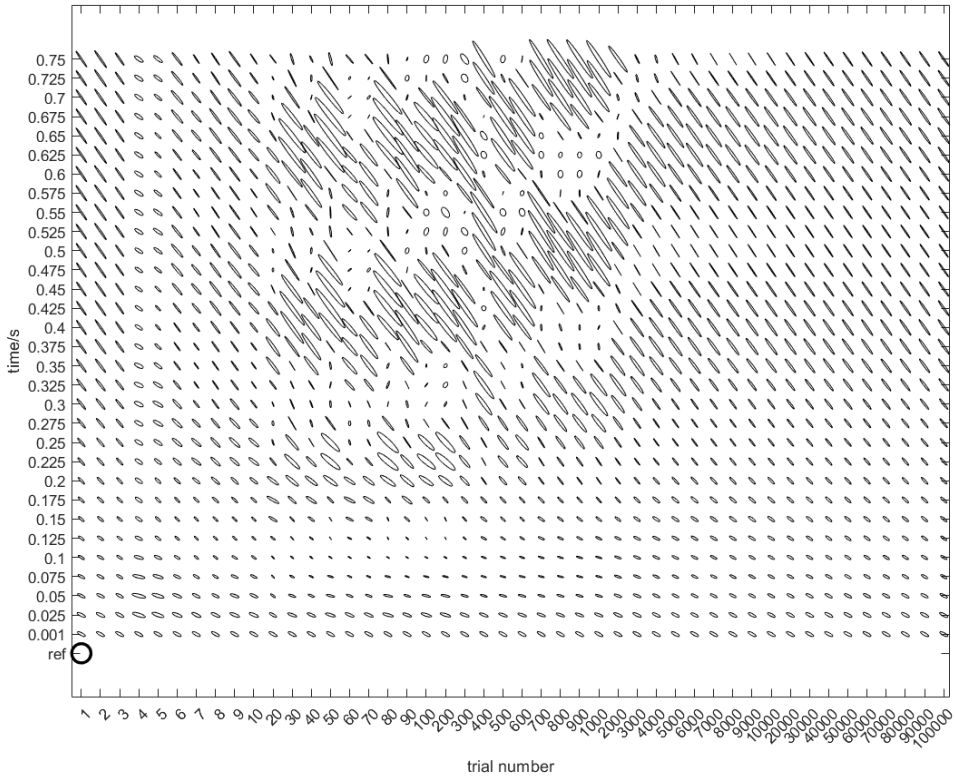
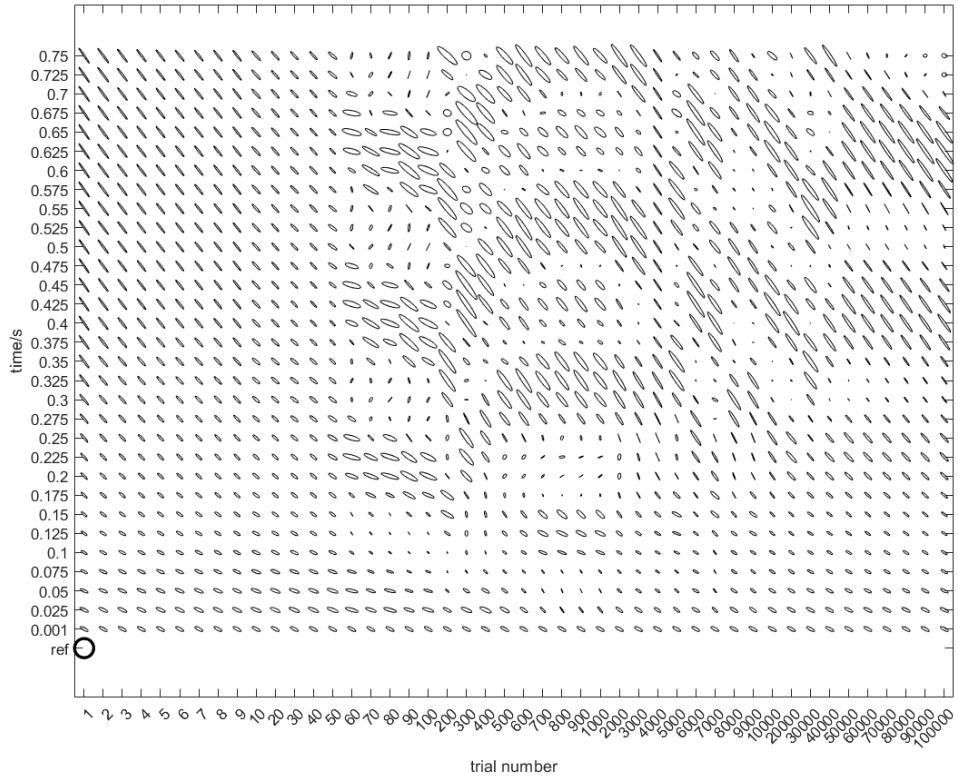



Fig. 22. Left: The relation between the cost and predetermined movement time; Right: Ther relation between the cost and the length of the desired trajectory (the distance between the start and the goal).

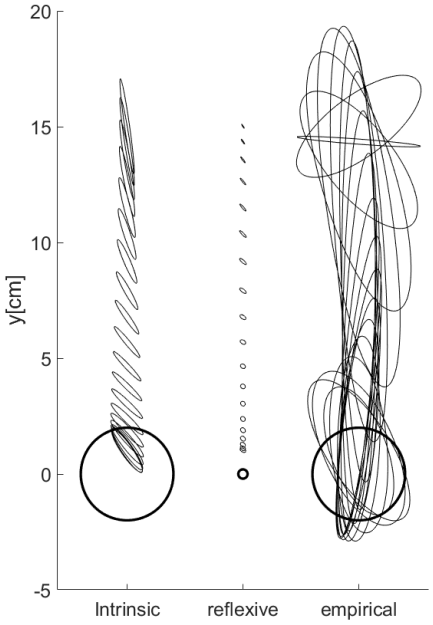



(a) Intrinsic stiffness ellipses for the DF1

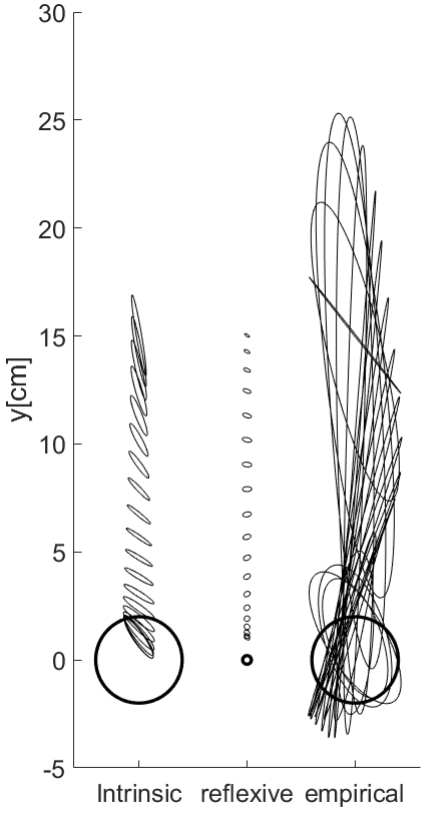



(b) Stiffness ellipses for the DF2

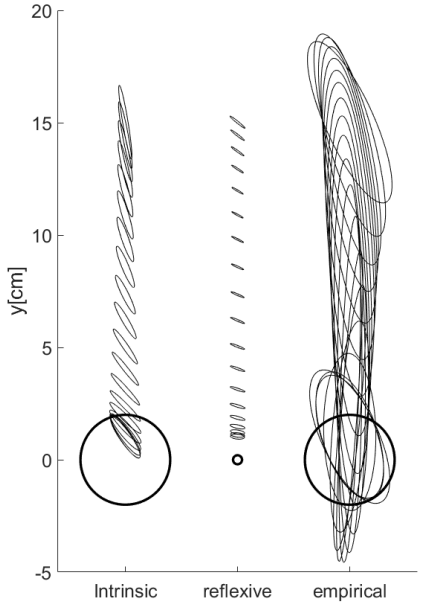
Fig. 23. Intrinsic stiffness profiles in the DFs, the reference is 100 N/m



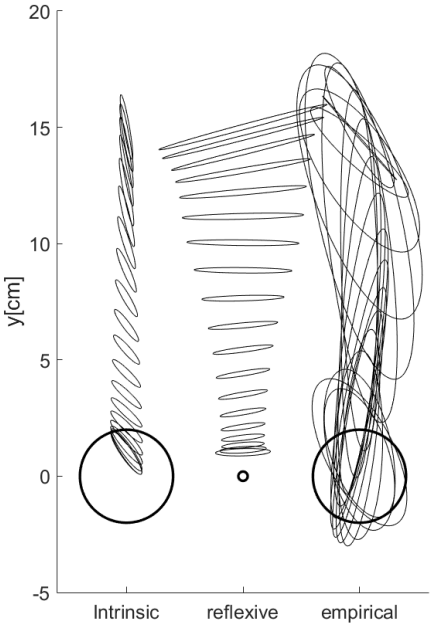

(a) Stiffness of the learnt motion in the NF




(b) Stiffness of the learnt motion in the VF1

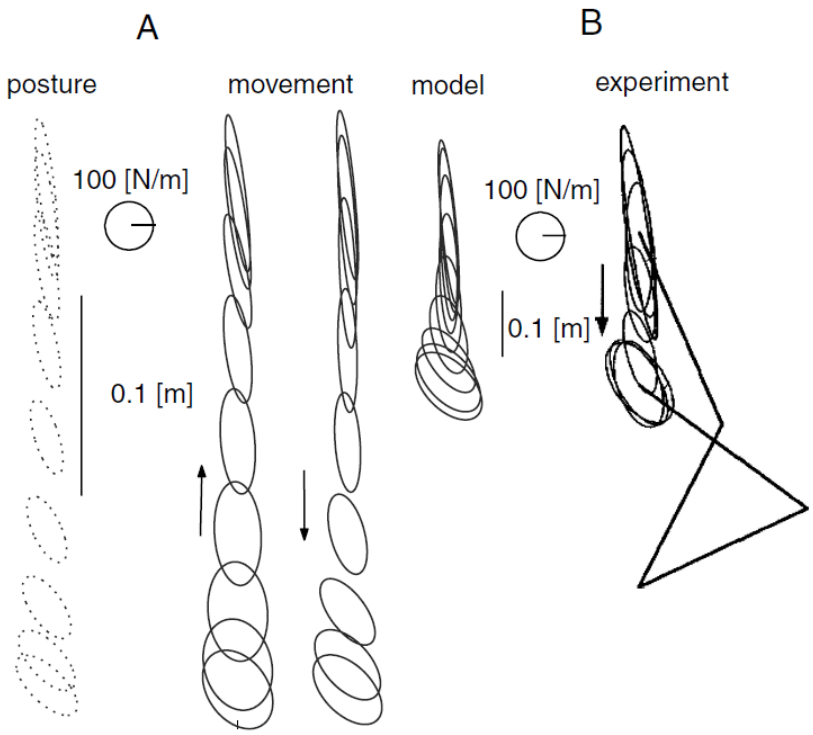



(c) Stiffness of the learnt motion in the DF1

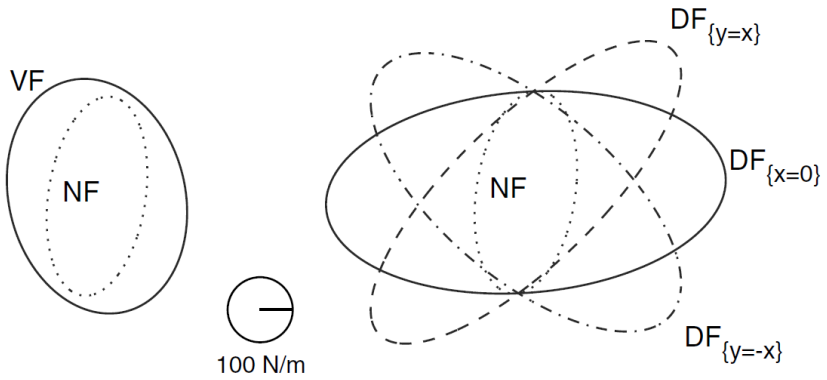



(d) Stiffness of the learnt motion in the DF2

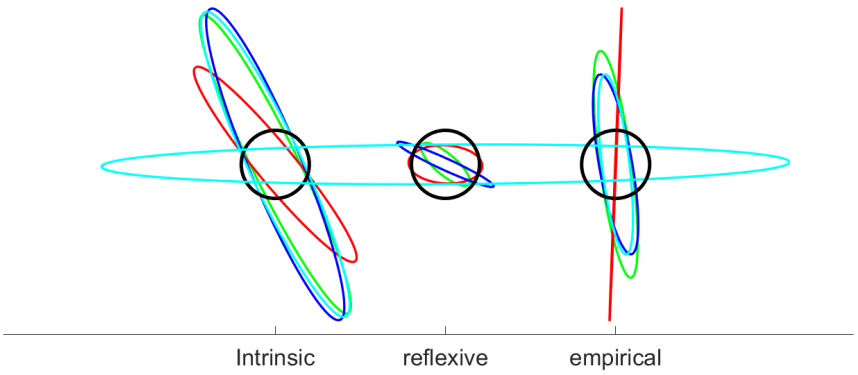
Fig. 24. Intrinsic stiffness profiles in four force fields. The left, middle, right reference circles in all plots stand for 100N/m, 10N/m, and 100N/m stiffness.




(a) A: stiffness profiles in the NF by (Tee et al., 2004); B: stiffness profiles in the NF by (Gomi and Kawato, 1997). Both figures were adopted from the former.

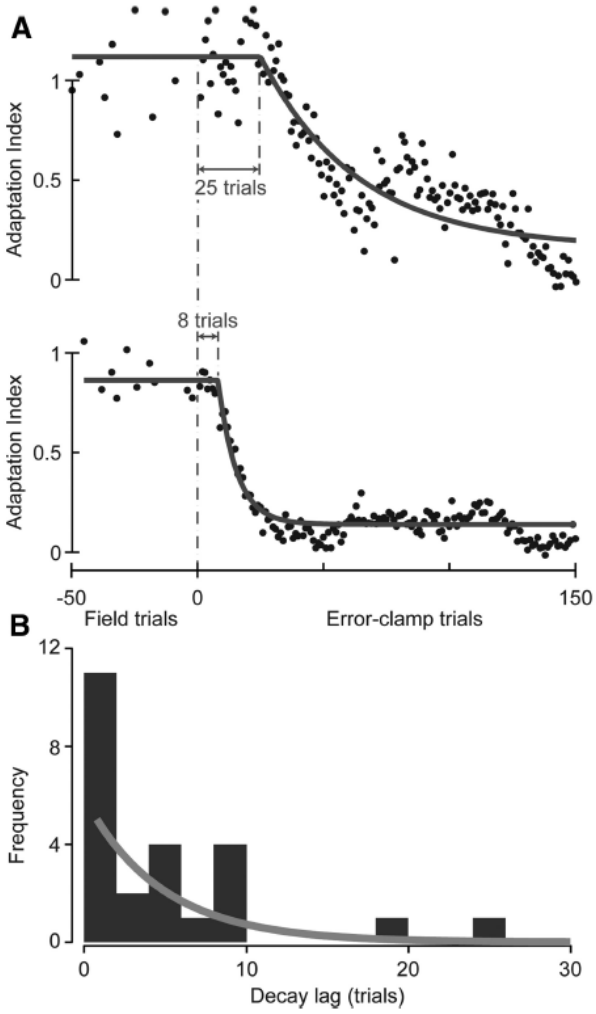



(b) Impedance in NF, VF, and DF by (Tee et al., 2004)

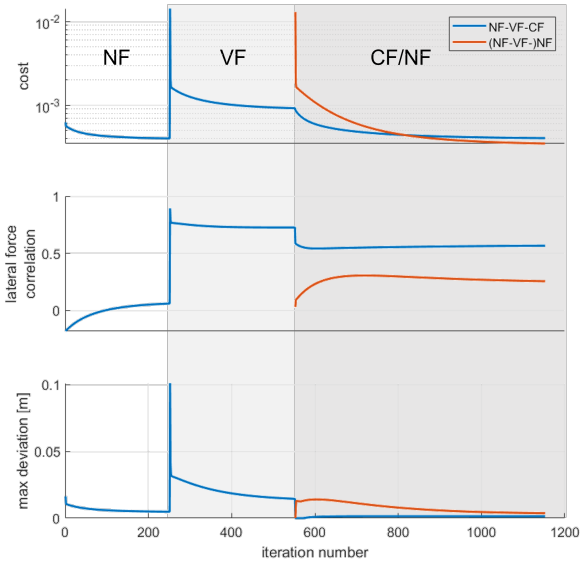



(c) Stiffness at the midway of the motion in NF (green), VF (red), DF1 (blue) and DF2 (cyan) from the simulation. The reference circles (black) from left to right represent 10N/m, 10N/m, and 100N/m.

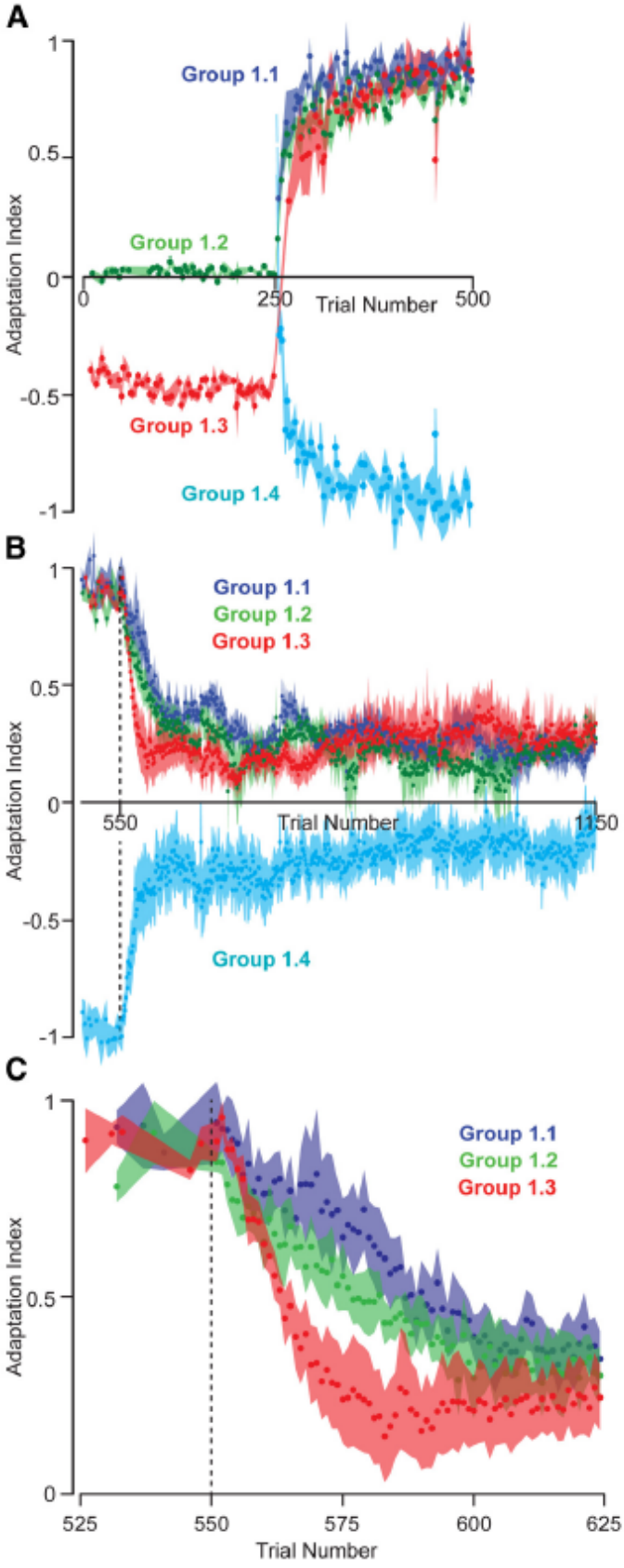
Fig. 25.




(a) The lag in memory decay observed by (Vaswani and Shadmehr, 2013).




(b) Simulation result, trained in the NF-VF2-CF sequence.




(c) Experiment by (Vaswani and Shadmehr, 2013), group 1.2 has the same task design as the simulation.

Fig. 26. Comparison of the change of the learning index based on lateral force correlation coefficient in reference to the trained model in VF2.