Supplementary Information: Quantifying Generalization from Trial-by-Trial behavior of Adaptive Systems that Learn with Basis Functions

Opher Donchin, Joseph T. Francis, and Reza Shadmehr

July 31, 2003

S.1 The artificial adaptive controller

Making movements with the adaptive controller

In our task, a subject holds the handle of a robotic arm and makes movements in the horizontal plane to reach visually presented targets. The paradigm involves a four-link system described in Fig. 1A of the paper, where two of the links are formed by the subject's arm and the other two are formed by the robotic arm. The physics of this coupled system can be expressed as follows:

$$\begin{cases} I_r(\mathbf{p})\ddot{\mathbf{p}} + G_r(\mathbf{p},\dot{\mathbf{p}})\dot{\mathbf{p}} = \mathbf{E}(\mathbf{p},\dot{\mathbf{p}}) + J_r^T(\mathbf{p})\mathbf{F}_{\text{handle}} \\ I_s(\mathbf{q})\ddot{\mathbf{q}} + G_s(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} = \mathbf{C}(\mathbf{q},\dot{\mathbf{q}},\mathbf{q}^*(t)) - J_s^T(\mathbf{q})\mathbf{F}_{\text{handle}} \end{cases} \tag{S1}$$

Where r is robot, s is subject, $\mathbf{p}$ is robot joint angles, $\mathbf{q}$ is subject joint angles, I is an inertial matrix, G is a coriolis-centripetal matrix, and $\mathbf{E}$ and $\mathbf{C}$ are forces actively generated by the robot and subject respectively. $\mathbf{F}_{\text{handle}}$ is a coupling term that represents the force that the robot and subject apply to each other. The active subject torques, called the controller, are governed by both the actual position and velocity and the desired trajectory $\mathbf{q}^*(t)$. The desired trajectories we used were minimum-jerk for displacements of 10cm of 0.5s (Flash and Hogan, 1985).

Previous simulations (Shadmehr and Brashers-Krug, 1997) suggest that for 0.5s reaching movements, human hand trajectories can be reasonably accounted for by a controller with a feed forward internal model of the task's dynamics and a simple feedback mechanism to stabilize the limb:

$$\mathbf{C} = \hat{I}_s(\mathbf{q}^*)\,\ddot{\mathbf{q}}^* + \hat{G}_s(\mathbf{q}^*,\dot{\mathbf{q}}^*)\dot{\mathbf{q}}^* + \hat{J}_s^T(\mathbf{q}^*)\hat{\mathbf{F}}_{\text{handle}} - K\,(\mathbf{q}-\mathbf{q}^*) - R\,(\dot{\mathbf{q}}-\dot{\mathbf{q}}^*)$$

where K and R are empirical estimates of a typical subject's joint stiffness (while at posture) and viscosity matrices. A hat ($\hat{\square}$) indicates a controller estimate rather than an actual value. We assume that the controller represents the robot's imposed forces in hand-centered rather than joint coordinates. This is only to simplify presentation, and we show in section S.6 that this simplification does not affect our conclusions.

$\hat{\mathbf{F}}_{\text{handle}}$ combines an estimate of the robot's passive inertial properties $\hat{\mathbf{F}}_{\text{robot arm}}$ with an estimate of the active torques imposed by the robot $\hat{\mathbf{F}}_{\text{field}}$. (We will generally write $\hat{\mathbf{F}}$ for $\hat{\mathbf{F}}_{\text{field}}$.) These forces are also additive:

$$\hat{\mathbf{F}}_{\text{handle}} = \hat{\mathbf{F}}_{\text{robot arm}} + \hat{\mathbf{F}}$$

Because all of our subjects practiced for approximately 600 movements in the null field before making any fielded movements, we assume that the controller has an accurate internal model of the inertial properties of the its own arm as well as the robot arm.

Eq. S1 was numerically integrated with a time step of 1 ms to produce human-like movements.

Learning with basis functions in the adaptive controller

To adapt to the force field, our controller must approximate the force field's map from velocity to force. Because the controller is primarily feed forward, the approximate map must use desired velocity and not actual velocity, so $\hat{\mathbf{F}}(\dot{\mathbf{x}}^*) \approx \mathbf{F}(\dot{\mathbf{x}})$ after learning. To build this map, we rely on the the theory of basis functions (Poggio, Fahle, and Edelman, 1992; Sanner and Kosha, 1999). We assume that the space of velocities is encoded using a collection of basis elements g_j each of which has a receptive field covering a specific piece of the space. (The shape of the basis elements and the tiling we used is specified in the Methods.) We also associate a force vector $\mathbf{w}_j$ with each basis element. The estimate of force for the internal model is then (repeating Eq. 2 from the paper)

$$\begin{aligned} \mathbf{g} &= [g_1(\dot{\mathbf{x}}), \ldots, g_m(\dot{\mathbf{x}})]^T \\ W &= \begin{bmatrix} w_{x1} & \ldots & w_{xm} \\ w_{y1} & \ldots & w_{ym} \end{bmatrix} \\ \hat{\mathbf{F}}(\dot{\mathbf{x}}) &= W\mathbf{g}(\dot{\mathbf{x}}) \end{aligned} \tag{S2}$$

The internal model will need to adapt so that the estimate $\hat{\mathbf{F}}(\dot{\mathbf{x}}^*)$ can approximate the actual force field $\mathbf{F}(\dot{\mathbf{x}})$. We assume that adaptation is through changes in the force vectors, w_{ij}, associated with each basis element, and that the receptive field $\mathbf{g}$ is fixed. It is also possible to consider other forms of adaptation (such as modifying the receptive field while leaving the force vectors fixed), but analysis of the other options is much less simple. We further assume that during a movement, no changes occur in w_{ij}. All changes are confined to the period after the movement has ended and before the next movement has started.

We use gradient descent to change the force vectors w_{ij}. We define a positive definite functional over the error in predicted force ($\tilde{\mathbf{F}}(t) = \mathbf{F}(t) - \hat{\mathbf{F}}(\dot{\mathbf{x}}^*(t))$):

$$e = \frac{1}{2}\int_0^T \tilde{\mathbf{F}}(t)^T\tilde{\mathbf{F}}(t)\,dt$$

Where T is the total time of the movement: 500ms in our paradigm. The length of the desired trajectory is fixed across movements, so the limits of integration are well defined. We also tested the model using the time of the actual movement rather than the desired trajectory, but this did not have a significant effect on the results. Using this error functional, we can create an update rule for the w_{ij}

$$w_{ij}^{(n+1)} = w_{ij}^{(n)} - \eta\frac{\partial e}{\partial w_{ij}}$$

(Superscript (n) means movement number n.) We can plug in definitions of $\hat{\mathbf{F}}$ (Eq. S2) and $\tilde{\mathbf{F}}$ above and carry out the differentiation to find that

$$w_{ij}^{(n+1)} = w_{ij}^{(n)} + \eta\int_{t=0}^{T} \tilde{f}_i^{(n)}(t) g_j(\dot{\mathbf{x}}^*_{k^{(n)}}(t))\,dt$$

$\dot{\mathbf{x}}^*_{k^{(n)}}(t)$ is the desired velocity in direction $k^{(n)}$, the direction of the nth movement, considered as a function of time. $\tilde{f}_i$ is the ith component of vector $\tilde{\mathbf{F}}$. It is more convenient to write this equation in matrix notation:

$$W^{(n+1)} = W^{(n)} + \eta \int_{t=0}^{T} \tilde{\mathbf{F}}^{(n)}(t)\mathbf{g}(\dot{\mathbf{x}}^*_{k^{(n)}}(t))\, dt \tag{S3}$$

We have found the rule with which the adaptive controller will update its parameters after each movement. Thus, to generate a sequence of movements, we start with an initial randomized W matrix, we then integrate Eq. S1 to produce a movement. Then, we plug the $\tilde{\mathbf{F}}$ experienced on that movement into Eq. S3 to update the weights. This is followed by another movement with the new W, and another update using the new $\tilde{\mathbf{F}}$. The whole process is iterated for however many movements are required.

S.2 The dynamical model

Here we show that despite nonlinear dynamics of the system and the complexities of the artificial adaptive controller, the trial-by-trial performance of the system can be approximated with linear dynamics. We begin by writing Eq. S3 as an equation in the internal model's output, $\hat{\mathbf{F}}$, by picking an arbitrary point in velocity space $\dot{\mathbf{x}}_a$ and multiplying both sides by $\mathbf{g}(\dot{\mathbf{x}}_a)$

$$\hat{\mathbf{F}}^{(n+1)}(\dot{\mathbf{x}}_a) = \hat{\mathbf{F}}^{(n)}(\dot{\mathbf{x}}_a) + \eta \int_{t=0}^{T} \left[\mathbf{g}(\dot{\mathbf{x}}^*_{k^{(n)}})^T \mathbf{g}(\dot{\mathbf{x}}_a)\right] \tilde{\mathbf{F}}^{(n)}(t)\, dt$$

The equation describes how a force error $\tilde{\mathbf{F}}^{(n)}(t)$ experienced over the trajectory of movement n in direction $k^{(n)}$ changes the internal model $\hat{\mathbf{F}}$ at some arbitrary point in velocity space, $\dot{\mathbf{x}}_a$. A number of approximations will be used to simplify this equation.

For the force fields used in this paper, magnitude of force is always proportional to speed. Our first approximation is that, along a straight line in velocity space (constant velocity direction, such as the desired trajectory) the magnitude of the force error will be also proportional to the speed. If we use $\dot{x}_{\max}$ for the peak speed, $t_{\max}$ for the time of the peak speed, and $\dot{\mathbf{x}}_{k^{(n)}}$ for the velocity at peak speed on the n-th movement:

$$\tilde{\mathbf{F}}(t) \approx \frac{\tilde{\mathbf{F}}(t_{\max})}{\dot{x}_{\max}} \cdot \dot{x}(t)$$

We have slipped in a second approximation: the peak speed is the same and occurs at the same time from movement to movement. We can now move the constants outside the integral.

$$\hat{\mathbf{F}}^{(n+1)}(\dot{\mathbf{x}}_a) = \hat{\mathbf{F}}^{(n)}(\dot{\mathbf{x}}_a) + \eta \frac{\tilde{\mathbf{F}}(t_{\max})}{\dot{x}_{\max}} \int_{t=0}^{T} \left[\mathbf{g}(\dot{\mathbf{x}}^*_{k^{(n)}}(t))^T \mathbf{g}(\dot{\mathbf{x}}_a)\right] \dot{x}(t)\, dt \tag{S4}$$

The next approximation is based on the fact that speed of reaching movements typically follows a bell-shaped profile. We replace this bell-shaped function with a triangular function of slope

$$a = \frac{\dot{x}_{\max}}{T/2} \tag{S5}$$

This triangular function peaks midway in the movement (at $t_{\max}$). Making this change introduces an average deviation from the bell shaped profile of only $0.07 \times \dot{x}_{\max}$. Under this approximation, the integral in Eq. S4 can be rewritten as an integral over speed rather than

time (where a well defined conversion from speed to velocity is possible because the desired trajectories only visit one velocity at each speed):

$$\hat{\mathbf{F}}^{(n+1)}(\dot{\mathbf{x}}_a) = \hat{\mathbf{F}}^{(n)}(\dot{\mathbf{x}}_a) + \frac{2\eta\tilde{\mathbf{F}}(t_{\max})}{a\dot{x}_{\max}} \int_{\dot{x}=0}^{\dot{x}=\dot{x}_{\max}} \left[\mathbf{g}\left(\frac{\dot{\mathbf{x}}_{k^{(n)}}}{\dot{x}_{\max}} \cdot \dot{x} \right)^T \mathbf{g}(\dot{\mathbf{x}}_a) \right] \dot{x}\, d\dot{x}$$

Since $\dot{\mathbf{x}}_a$ is an arbitrary point in velocity space, we are free to select it in whatever way is convenient to us. We choose it to be the peak velocity of the desired trajectory associated with some movement direction, l. Then, using $\dot{\mathbf{x}}_l$ for $\dot{\mathbf{x}}^*_l(t_{\max})$:

$$\hat{\mathbf{F}}^{(n+1)}(\dot{\mathbf{x}}_l) = \hat{\mathbf{F}}^{(n)}(\dot{\mathbf{x}}_l) + \frac{2\eta\tilde{\mathbf{F}}(t_{\max})}{a\dot{x}_{\max}} \int_{\dot{x}=0}^{\dot{x}=\dot{x}_{\max}} \left[\mathbf{g}\left(\frac{\dot{\mathbf{x}}_{k^{(n)}}}{\dot{x}_{\max}} \cdot \dot{x} \right)^T \mathbf{g}(\dot{\mathbf{x}}_l) \right] \dot{x}\, d\dot{x} \tag{S6}$$

We define a function of the two directions, l and $k^{(n)}$:

$$B_{l,k^{(n)}} = \frac{2\eta}{a\dot{x}_{\max}} \int_{x'=0}^{x'=\dot{x}_{\max}} \left[\mathbf{g}\left(\frac{\dot{\mathbf{x}}_{k^{(n)}}}{\dot{x}_{\max}} \cdot x' \right)^T \mathbf{g}(\dot{\mathbf{x}}_l) \right] x'\, dx'$$

This is the definition of $B_{l,k}$ given in the text (Eq. 4) with $\alpha = \frac{2\eta}{a\dot{x}_{\max}}$. We can now rewrite equation S6 in terms of $B_{l,k}$ (understanding $\tilde{\mathbf{F}}$ to be evaluated at $t_{\max}$)

$$\hat{\mathbf{F}}^{(n+1)}(\dot{\mathbf{x}}_l) = \hat{\mathbf{F}}^{(n)}(\dot{\mathbf{x}}_l) + B_{l,k^{(n)}}\tilde{\mathbf{F}}$$

This B function captures those aspects of the update rule that depend only on the directions of movement. Note that the equation should hold true for any choice of target direction, l. We therefore have a separate equation for each possible l. In all of the experiments reported in the paper, we have 8 different target directions. Thus, it makes sense instead of $\hat{\mathbf{F}}(\dot{\mathbf{x}}_l)$ to write $\hat{\mathbf{F}}_l$ which will be a 2×1 vector for each direction of movement.

$$\hat{\mathbf{F}}^{(n+1)}_l = \hat{\mathbf{F}}^{(n)}_l + B_{l,k^{(n)}}\tilde{\mathbf{F}}^{(n)} \qquad l = 1, \ldots, 8$$

which asserts that the force error $\tilde{\mathbf{F}}^{(n)}$ experienced at peak velocity will change the internal model $\hat{\mathbf{F}}_l$ for all possible movement directions, $l = 1, \ldots, 8$. This effect of $\tilde{\mathbf{F}}^{(n)}$ is modulated by a *generalization function* $B_{l,k}$. The generalization function describes how errors experienced in direction k affect the internal model for any other direction l. Because there are 8 directions of movement, $B_{l,k}$ is an 8×8 matrix.

Our task is to estimate the generalization function B from the movements themselves. However, while we know both $\hat{\mathbf{F}}^{(n)}_l$ and $\tilde{\mathbf{F}}^{(n)}(t)$ at every time point for the artificial adaptive controller, neither can be easily measured from experimental data in real subjects. That means that $\tilde{\mathbf{F}}^{(n)}$ would be difficult to assess cleanly in our experiments. An easier variable to measure would be error in the movement trajectory. We can define movement error, $\mathbf{y}$, as hand displacement from a desired trajectory at the point of maximum velocity. In the simulation, it can be shown empirically that for perturbed reaching movements $\mathbf{y}$ and $\tilde{\mathbf{F}}$ are linearly related across the 8 directions of movement (Donchin and Shadmehr, 2002). The linear relationship is captured by a compliance-like 2×2 matrix we call D, so that $y = D\tilde{\mathbf{F}}$. We now have a system of nine vector equations:

$$\begin{cases} \mathbf{y}^{(n)} = D(\mathbf{F}^{(n)} - \hat{\mathbf{F}}^{(n)}_{k^{(n)}}) \\ \hat{\mathbf{F}}^{(n+1)}_l = \hat{\mathbf{F}}^{(n)}_l + B_{l,k^{(n)}}\tilde{\mathbf{F}}^{(n)} & l = 1, \ldots, 8 \end{cases}$$

We introduce one more new variable, $\mathbf{z}^{(n)}_{k^{(n)}} \equiv D\hat{\mathbf{F}}^{(n)}_{k^{(n)}}$. Intuitively, $\mathbf{z}$ can be thought of as the displacement that would have been experienced during a movement n if the internal model had not compensated for the expected field. We replace $\hat{\mathbf{F}}$ with $D^{-1}\mathbf{z}$ and after some simplifications we have:

$$\begin{cases} \mathbf{y}^{(n)} = D\mathbf{F}^{(n)} - \mathbf{z}^{(n)}_{k^{(n)}} \\ \mathbf{z}^{(n+1)}_l = \mathbf{z}^{(n)}_l + B_{l,k^{(n)}}\mathbf{y}^{(n)} \quad l = 1, \cdots, 8 \end{cases} \tag{S7}$$

This is the dynamical system of the paper's Eq. 3 that is used throughout the text.

Our sequence of approximations can be justified by showing that this dynamical model is a good fit to the data generated by the adaptive controller. Performance in each movement was quantified using hand position error at peak velocity (the vector $\mathbf{y}$). These errors are displayed for a simulation with Gaussian basis elements of width $\sigma = 0.2$ m/s in Supplementary Figure S1 (blue lines, catch trials indicated by circles). The figure displays the x and y components of the error vector as a function of movement number. In the outer subplots, movements are grouped according to direction. In the center subplot, the errors for the first 75 movements (in their actual sequence) are illustrated. The effect of the arm's anisotropic inertia and compliance, combined with the effect of the force field, results in different patterns of error in different directions.

We took the entire sequence of errors (192 movements) and fitted them to the dynamical system of Eq. S7. We then used the parameters of the fit to iterate Eq. S7 and produce a predicted sequence, $\hat{\mathbf{y}}^{(n)}$. In Fig. S1, this predicted sequence is shown in red while the measured sequence is shown in blue. The r^2 of the fit between model and data for the entire 192 target sequence is 0.920. The fit is remarkable because the simulation consisted of a set of non-linear differential equations that described limb dynamics and adaptation. However, the resulting sequence of errors were fit to a simple set of equations (Eq. S7), which contained essentially 12 unknown parameters (8 in B and 4 in D). The high degree of fit suggests that the simple equations compactly described the trial-to-trial pattern of movement errors in this adaptive system.

Fig. S2 compares the quality of fit for simulations with wide basis functions ($\sigma = 0.3$ m/s) to a simulation with narrower basis functions ($\sigma = 0.1$ m/s). For each simulation, the sequence of errors $\mathbf{y}$ were fit to the dynamical system, parameters B and D were found, and $\hat{\mathbf{y}}$ was estimated. This figure displays $\mathbf{y}$ and $\hat{\mathbf{y}}$ by projecting these vectors onto the parallel direction (a line connecting start point to target of each movement) and the perpendicular to it. This is a more natural representation than the x and y components. As before, catch trials are indicated with circles. Fig. S2A and C show the fit for all movements. The r^2 of the fits were 0.981 and 0.967.

S.3 Comparing the model to Thoroughman and Shadmehr (2000)

Here we list the differences between the dynamical model we reached by derivation (Eq. S7 in this document and Eq. 3 in the text) and the dynamical model used by Thoroughman and Shadmehr (2000) (their Eq. 4).

1. In the current model error and force are measured as vectors. In the Thoroughman and Shadmehr (2000) approach, error was measured as a scalar that indicated perpendicular displacement. In the scalar representation, error was generalized so that it was always perpendicular to the direction of motion. That is, it rotated around the circle with the direction of movement. In the current representation, the error vector for any given

direction is only scaled when it is generalized to any other direction. The two approaches make different experimental predictions that we can test. In the results section we show that generalization is best represented as scaling of a vector error.

An additional advantage of using vectors is that they permit us to generalize the theory to force fields other than the curl field. In the current paper, we use assistive and resistive fields that produce errors parallel to the direction of movement. A scalar representation of error would be insufficient for analyzing generalization in these fields.

2. The old model is in a canonical form. The updated value of the hidden state is an arbitrary linear combination of the old state and the newly experienced forces. In Eq. S7, the update rule is not arbitrary and, even more important, it is the *error* experienced in the current movement that drives the update and not the *force*.

3. Thoroughman and Shadmehr (2000) actually fit their Eq. 4 to the data separately for each direction of movement. This prevents a true trial-by-trial evaluation of the hidden state, and creates a danger of over-fitting. Since our model is derived explicitly from the behavior of an adapting internal model, it naturally describes the trial-by-trial evolution of that internal model.

4. Since our model explains performance across different directions, it contains a single 2×2 parameter, D, that captures the biomechanical contribution to variations in performance. There is no equivalent parameter in the previous model.

5. Since the previous model was fit separately for each direction, the value of the previous force was actually a vector of values for forces experienced in the different directions between every two movements in the direction that was being fit. Let's call these movement n and m. This elements of this vector were always either -1, 1 or 0. The value for direction k was -1 if there was a fielded trial in direction k between movements n and m. The value was 1 if there was a catch trial in direction k between movements n and m. The value was 0 if there were no movements in direction k between movements n and m. If there was more than one movement in direction k between movements n and m, only the last one was considered. This algorithm is quite different from the direct role played by the force in the new model.

S.4 Summary of assumptions and limitations

Our theoretical development is driven by a number of assumptions, some of which are innacurate or simply wrong. Its application is also limited by a number of approximation that go into the development of Eq. S7. While some of these assumptions and limitations are addressed in the discussion, we felt it was important to provide a more complete treatment.

1. We have assumed that the internal model is evaluated in a feed forward manner and depends on a desired trajectory. However, evidence suggests that sensory feedback does influence input to the internal model. This influence appears significant beyond 350 ms into a reaching movement (Bhushan and Shadmehr, 1999). We have limited our analysis to the early part of the movement where the role of feedback is more limited. Along the same lines, while evidence indicates that the motor system may use a desired trajectory (specified using kinematic variables) (Wolpert, Ghahramani, and Jordan, 1995), this evidence is not strong and other possibilities have been considered (Polit and Bizzi,

1978; Haruno, Wolpert, and Kawato, 2001; Todorov and Jordan, 2002). We consider the desired trajectory a convenient and simple assumption, but no more than a guess.

2. In our simulations, the learning rate, as well as the stiffness and viscosity of the arm, are fixed. This assumption greatly simplified our derivation of a dynamical model, leading to a constant generalization function and a constant compliance matrix. However, it is possible that some or all of these variables change as people adapt. This would lead to a poor fit of Eq. 3 to the human data. As we show, the fit to the data is good, suggesting that, in our task, these changes are relatively minor. However, our experience with this task does suggest that there are small changes in stiffness of the arm during training and those changes occur over the very first 32 movements (Thoroughman and Shadmehr, 1999).

3. We fix the basis elements (g_i) and only adapt the force vectors associated with each basis ($w_{xy,i}$). We did this for two reasons. The first is that it simplified our derivation. The second is that adapting the basis elements produces uneven groupings of similar force vectors and segregation of basis elements with different force vectors. More sophisticated approaches would address this problem, and there are at least two possibilities. One would adapt all three parameters of the basis elements (the center, the width, and the force vector). Another would start with a small number of basis elements and allow for limited movement of their centers, adding new basis elements as needed (Schaal and Atkeson, 1998). These possibilities will be considered in future research.

S.5 Fitting the model to data

In order to characterize the parameters in Eq. S7 (the generalization function B and the compliance matrix D), we record a sequence of N movements towards target directions $k^{(n)}$, $n = 1, \ldots, N$ in a random order. Each movement is associated with a certain force $\mathbf{F}^{(n)}$ and a certain error $\mathbf{y}^{(n)}$ measured at maximum velocity for that movement. We then try to find the best set of parameters B and, D and initial conditions ($\mathbf{z}_k^{(1)}$, $k = 1, \ldots, 8$) that when inserted into the dynamics described in Eq. S7, reproduces the recorded data. That is, when we find these parameters and run the dynamical system in Eq. S7, we want the sequence of outputs, $\hat{\mathbf{y}}^{(n)}$, to match the actual sequence of errors, $\mathbf{y}^{(n)}$, that we measured. This can be framed as a least-squares minimization problem with B, D and $\mathbf{z}_k^{(1)}$ as the unknown parameters minimizing:

$$\sum_{n=1}^{N} \|\hat{\mathbf{y}}^{(n)} - \mathbf{y}^{(n)}\|^2.$$

This is a non-linear optimization problem, and to solve it we used the optimization toolbox of Matlab Release 12.1 (Mathworks, Natick, MA). However, optimizing these parameters requires repeated iterated evaluations of Eq. 3 in order to produce sequences of estimated errors, $\hat{\mathbf{y}}^{(n)}$, that could be compared with $\mathbf{y}^{(n)}$ and could be quite slow, especially when we evaluated bootstrap statistics (see below). We were also concerned with the possibility of local minima, given the highly non-linear nature of the data. We thought that if we could find an approximate solution to seed the optimization routines, we would solve both of these problems. We found a way of using linear techniques to reach an approximate solution to the problem and we present it below.

Let us write the sequence of changes that take place in the hidden state, $\mathbf{z}_{k^{(n)}}$, from the

beginning of the set until we reach the errors made on movement n (in direction $k^{(n)}$):

$$\begin{aligned}
\mathbf{z}^{(2)}_{k^{(n)}} - \mathbf{z}^{(1)}_{k^{(n)}} &= B_{k^{(n)},k^{(1)}}\mathbf{y}^{(1)} \\
\mathbf{z}^{(3)}_{k^{(n)}} - \mathbf{z}^{(2)}_{k^{(n)}} &= B_{k^{(n)},k^{(2)}}\mathbf{y}^{(2)} \\
&\vdots \\
\mathbf{z}^{(n)}_{k^{(n)}} - \mathbf{z}^{(n-1)}_{k^{(n)}} &= B_{k^{(n)},k^{(n-1)}}\mathbf{y}^{(n-1)}
\end{aligned}$$

When we add all of the above equations we have:

$$\mathbf{z}^{(n)}_{k^{(n)}} - \mathbf{z}^{(1)}_{k^{(n)}} = \sum_{p=1}^{n-1} B_{k^{(n)},k^{(p)}}\mathbf{y}^{(p)}$$

Which can be rearranged to be a linear equation for the hidden internal state

$$\mathbf{z}^{(n)}_{k^{(n)}} = \sum_{p=1}^{n-1} B_{k^{(n)},k^{(p)}}\mathbf{y}^{(p)} + \mathbf{z}^{(1)}_{k^{(n)}}$$

Substituting $\mathbf{z}$ from the first equation in Eq. S7 into above we have:

$$D\mathbf{F}^{(n)} - \mathbf{y}^{(n)} = \sum_{p=1}^{n+m-1} B_{k^{(n)},k^{(p)}}\mathbf{y}^{(p)} + \mathbf{z}^{(1)}_{k^{(n)}}$$

We can combine two equations with this form: one describing movement n and one describing movement $n+m$. After rearranging terms, we get

$$\mathbf{y}^{(n)} - \mathbf{y}^{(n+m)} = \sum_{p=1}^{n+m-1} B_{k^{(n+m)},k^{(p)}}\mathbf{y}^{(p)} - \sum_{p=1}^{n-1} B_{k^{(n)},k^{(p)}}\mathbf{y}^{(p)} + \mathbf{z}^{(1)}_{k^{(n+m)}} - \mathbf{z}^{(1)}_{k^{(n)}} + D(\mathbf{F}^{(n)} - \mathbf{F}^{(n+m)}) \tag{S8}$$

Eq. S8 is useful to us because it provides us with a linear relationship between our behavioral measurements $k, \mathbf{y}, \mathbf{F}$ and the unknown parameters B, D, and $\mathbf{z}^{(1)}$. Each pair of movements can provide one vector equation as shown in Eq. S8 (or equivalently, two scalar equations). Thus, if there are N total movements in a target set, we could get $N(N-1)$ equations. However, practically, we found that we get equivalent results keeping only $8N$ linear equations. We kept only the equations for movements n and $n+m$ for which there was no movement in direction $k^{(n+m)}$ between them. That is, we combined each movement with the next movement in every other direction (including the direction of that movement). Thus, we had a linear system with $8N$ equations and 28 parameters. We used the solution of this system as the seed for our non-linear optimization routines.

S.6 Using a joint-angle representation

In our simulation, the internal model mapped hand velocity to force. One alternative is a transformation from joint angular velocities to joint torques. We built a simulation that had an internal model that used this transformation. As shown in Fig. S3, this simulation produced results that are quite similar to the ones produced by the Cartesian representation, suggesting that, within the context of our small workspace, the coordinate system of representation does not make a large difference. This is a result of the fact that for small movements, the transformation from joint velocity to hand velocity is well described by a constant Jacobian matrix.

S.7 Changing the desired trajectory

Our estimate of error assumes that subjects are trying to bring their trajectories in line with a 'desired trajectory.' This is consistent with much current thinking in motor control. However, we were interested in testing the consequences for our model if this assumption is violated. When we assumed that the desired trajectory was fixed, we used the null field training sets to assess the desired trajectory. If the desired trajectory has changed, this would no longer be appropriate. Instead, we estimated the desired trajectory from the movements at the end of each set. This allows for development of the desired trajectory over the course of three sets, but assumes that the movements near the end of each set reflect the intended trajectory by that time. That is, we assumed that by the end of a set, the time average of the hidden state changes has become constant:

$$\langle \Delta \mathbf{z}_l \rangle = 0 = \langle B_{l,l} \mathbf{y} \rangle$$

where the change in hidden state of the system is:

$$\Delta \mathbf{z}_l^{(n)} = \mathbf{z}_l^{(n+1)} - \mathbf{z}_l^{(n)}$$

and for the sake of this discussion we only consider the central value of the generalization function. The time-averaged error $\mathbf{y}$ can be separated into catch trials and field trials (with p_{field} and p_{catch} being the probability of a field trial and catch trial):

$$\langle B_{l,l} \mathbf{y} \rangle = 0 = B_{l,l} \left(p_{\text{field}} \langle \mathbf{y}_{\text{field}} \rangle + p_{\text{catch}} \langle \mathbf{y}_{\text{catch}} \rangle \right)$$

Since we are trying to find the appropriate reference point for the error, let us rewrite the error in terms of the actual location of the hand at peak velocity, $\mathbf{y}_a$, and the desired trajectory at peak velocity, $\mathbf{y}_d$: $\mathbf{y} = \mathbf{y}_a - \mathbf{y}_d$. Then the above equation becomes

$$0 = p_{\text{field}} \left(\langle \mathbf{y}_{a,\text{field}} \rangle - \mathbf{y}_d \right) + p_{\text{catch}} \left(\langle \mathbf{y}_{a,\text{catch}} \rangle - \mathbf{y}_d \right)$$

Which can be simplified to

$$\mathbf{y}_d = p_{\text{field}} \langle \mathbf{y}_{a,\text{field}} \rangle + p_{\text{catch}} \langle \mathbf{y}_{a,\text{catch}} \rangle \tag{S9}$$

p_{field} and p_{catch} were estimated simply by counting the total number of field and catch trials performed by the subjects. $\mathbf{y}_a$ was estimated by averaging the movements in the last 1/3 of the data set. Catch trials and field trials were averaged separately. Thus, in estimating the desired trajectory, we did not increase the number of parameters in the model.

S.8 Bootstrapping the parameters

We used standard bootstrap techniques (Efron and Tibshirani, 1993) to determine confidence limits for our parameters. In any particular experiment, we used 200 resamplings (called bootstrap samples) of the subjects. Each bootstrap sample included a random collection of the subjects selected with replacement, so each subject could appear once, more than once, or no times at all in a any given bootstrap sample. The number of subjects in each bootstrap sample was equal to the number of subjects in the original sample. For each bootstrap sample, we now performed our analysis just as we had on the original data. Perpendicular displacements were averaged across the subjects (appropriately weighted if they appeared more than once) in the bootstrap sample, and the dynamical model of Eq. S7 was fit to this sequence of perpendicular displacements. The parameters generated by making this fit are called bootstrap estimates of

the original parameter, and we had 200 bootstrap estimates for both B and D. The distribution of these parameters was used to generate confidence limits by sorting the values and taking the 5th and 195th values to be the low and high bounds for our 95% confidence limits. In presentation, we also used the standard deviation of this distribution as an estimate of the standard error of the parameter (S.E.).

S.9 Randomization test for r^2

We used a test similar to the bootstrap called the randomization test (Manly, 1997) to test the significance of our r^2 parameters. First, we generated a null distribution of r^2. A null distribution obeys the null hypothesis that the parameters have no explanatory power, and can be used to assess the statistical significance for rejecting the null hypothesis. Under the null hypothesis that B and D have no explanatory power, the sequence $\mathbf{y}^{(n)}$ for movements in a given direction is just as likely as any other permutation of that sequence. Thus, to generate the null distribution we generated random permutations (called randomizations) of $\mathbf{y}^{(n)}$ associated with each direction of movement separately, leaving the sequence of target directions and forces fixed. For each randomization, we again fit the dynamical model of Eq. S7 and used the fit to generate values for r^2, r_B^2 and r_D^2. Thus, we had a distribution of values for each of these statistics in a situation very similar to our experimental situation but where we knew that the sequence of errors was indeed random. Thus, our statistical hypothesis is that the r^2, r_B^2, and r_D^2 in our actual data was greater than 95% or 99% of the r^2 generated by randomization. We tested the significance of our r^2 by comparing them to the 190th or 198th value in the distribution of randomized r^2 that we produced.

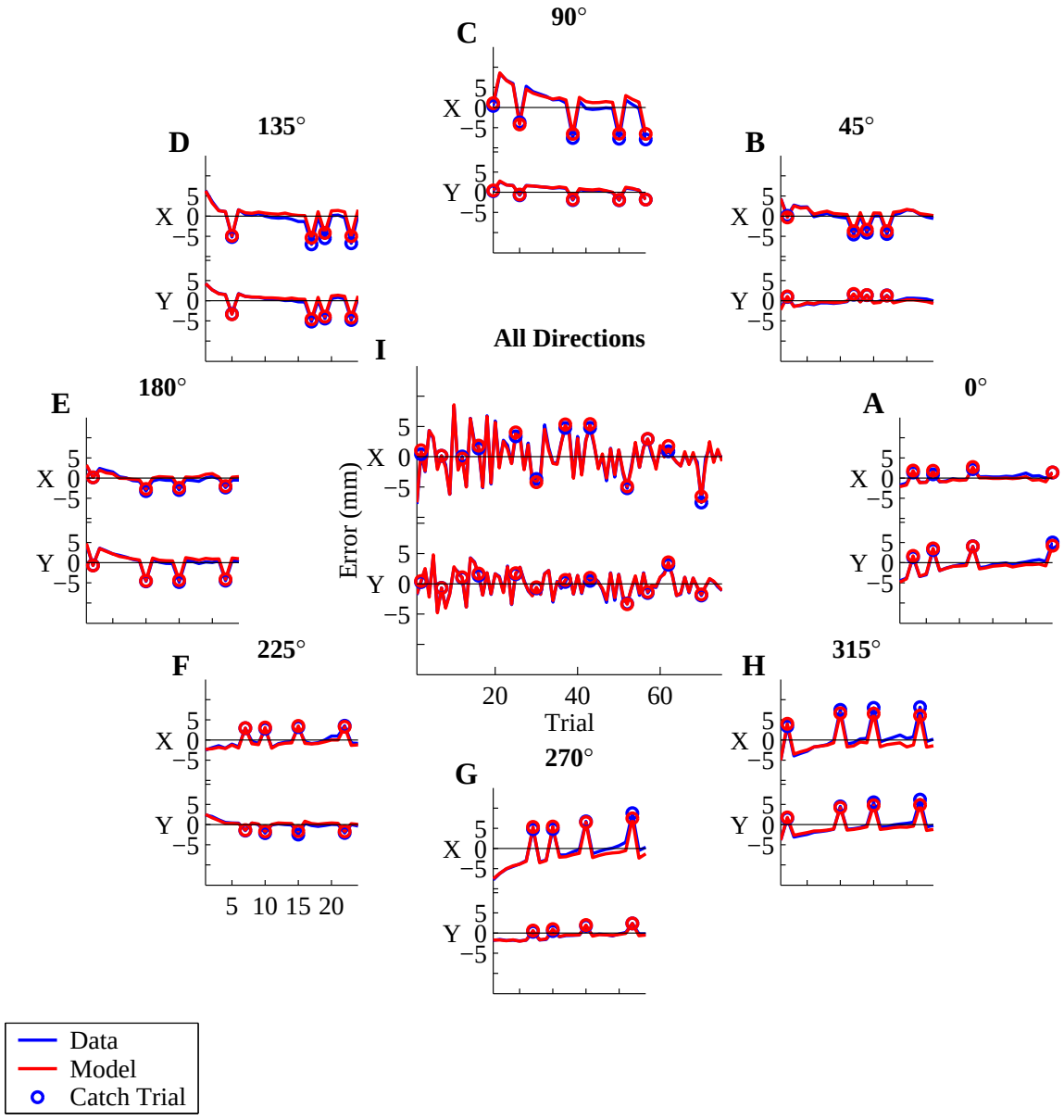

Figure S1: We simulated movement in a clockwise curl-field governed by an adaptive controller. 192 movements were performed in a random sequence of directions. The controller learned an internal model of the field using Gaussian bases of width $\sigma = 0.2$. X and Y of the movement error vector (blue lines) was computed as the displacement from the unperturbed trajectory at peak velocity. Catch trials are indicated by circles. The sequence of errors were fit to Eq. S7 as described in the Supplementary Material. The resulting parameters of the fit, B and D, were used to reconstruct the error sequence (red lines). The r^2 of the fit between model and data for the entire 192 target sequence was 0.995.

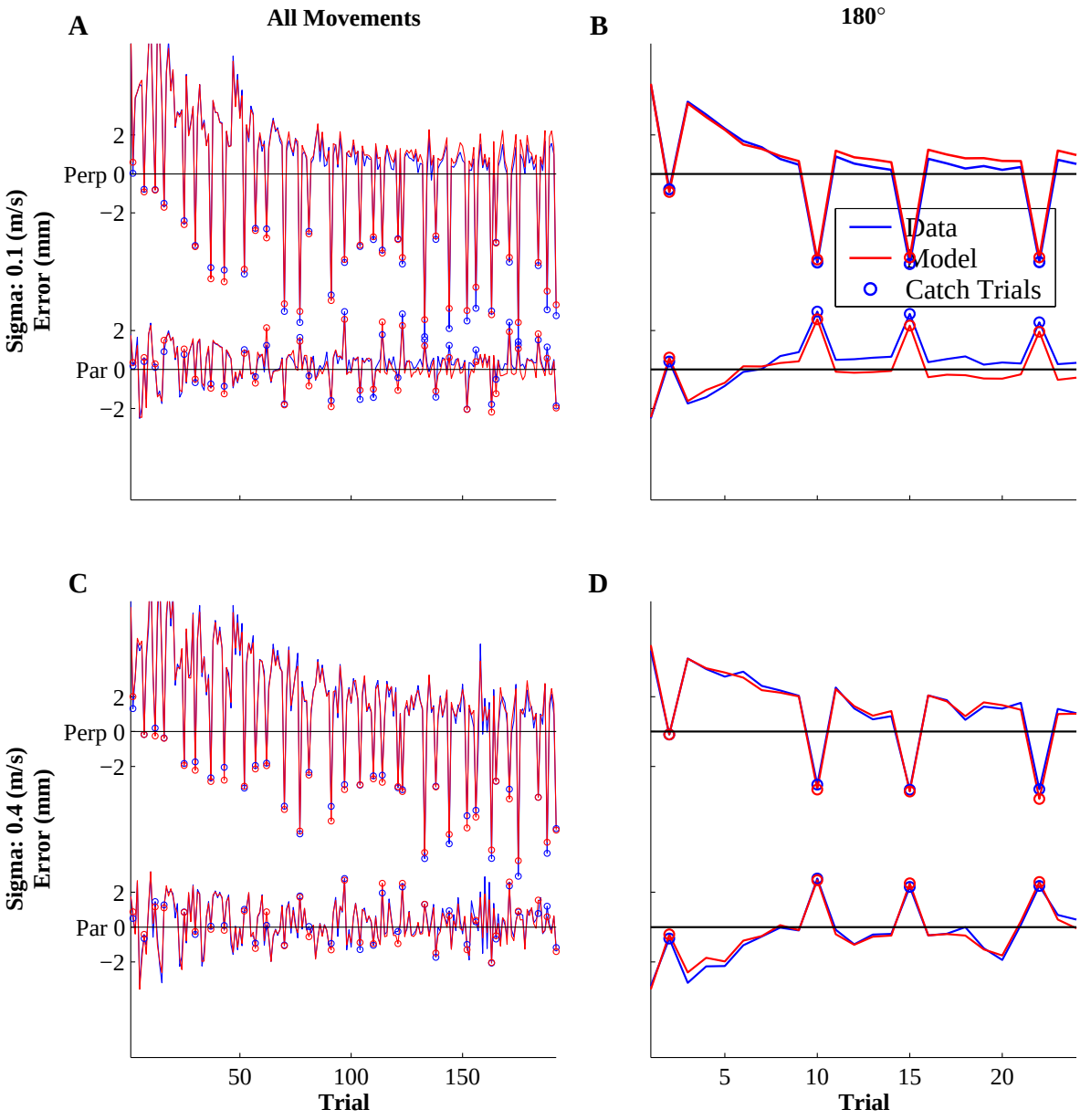

Figure S2: A comparison of movement errors for an adaptive controller that learned with narrow bases $\sigma = 0.1$ (**A** and **B**), or wide bases $\sigma = 0.3$ m/s (**C** and **D**). The controller trained on 192 movements in a random sequence of directions in the clockwise curl field. The error vector plotted here is decomposed into perpendicular and parallel displacements. In all plots, the blue lines are the errors generated by the adaptive controller and the red lines are the estimates of Eq. S7. Catch trials are indicated by circles. **A** and **C**: Errors for the entire sequence of movements. **B** and **D**: Errors for movements to 180°. Note the monotonic character of the data and fit in **B** and the non-monotonic character of the data and fit in **D**.

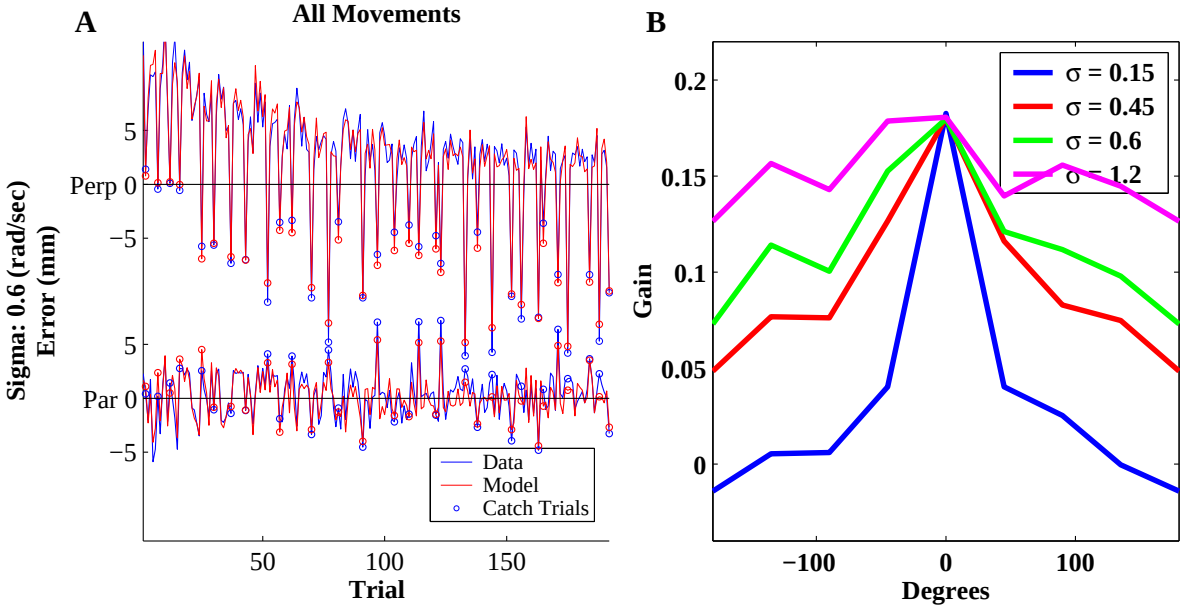

Figure S3: Results from a simulation in which the internal model represented the field as a transformation from joint angles to torques, rather than from Cartesian velocity to forces. A) The behavior of the simulation, and the fit, seem similar to those generated by the Cartesian representation. B) It is possible by varying the width of the representation to generate generalization functions that are very similar to the ones generated by the Cartesian representation. σ is in rad / sec.

References

Bhushan, N. and R. Shadmehr (1999). Computational nature of human adaptive control during learning of reaching movements in force fields. *Biol. Cybern.* 81: 39–60.

Donchin, O. and R. Shadmehr (2002). *Linking Motor Learning to Function Approximation: Learning in an Unlearnable Force Field*, pp. 197–203. MIT Press, Cambridge, MA.

Efron, B. and R. J. Tibshirani (1993). *An Introduction to the Bootstrap*. Monographs on statistics and applied probability. Chapman and Hall, New York. These pages are for bootstrapping residuals in a linear regression.

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

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

Manly, B. F. J. (1997). *Randomization, Bootstrap and Monte Carlo Methods in Biology*. Texts in Statistical Science Series. Chapman and Hall, London, 2 edition.

Poggio, T., M. Fahle, and S. Edelman (1992). Fast perceptual learning in visual hyperacuity. *Science* 256: 1018–1021.

Polit, A. and E. Bizzi (1978). Processes controlling arm movements in monkeys. *Science* 201: 1235–1237.

Sanner, R. M. and M. Kosha (1999). A mathematical model of the adaptive control of human arm motions. *Biol. Cybern.* 80: 369–382.

Schaal, S. and C. G. Atkeson (1998). Constructive incremental learning from only local information. *Neural Comput.* 10: 2047–2084.

Shadmehr, R. and T. Brashers-Krug (1997). Functional stages in the formation of human long-term motor memory. *J. Neurosci.* 17: 409–419.

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

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

Todorov, E. and M. Jordan (2002). Optimal feedback control as a theory of motor coordination. unpublished work.

Wolpert, D. M., Z. Ghahramani, and M. I. Jordan (1995). Are arm trajectories planned in kinematic or dynamic coordinates? an adaptation study. *Exp. Brain Res.* 103: 460–470.