

Optimal feedback control as a theory of motor coordination:

Supplementary Notes

Emanuel Todorov, Michael I. Jordan

1. Optimal control of modified Linear-Quadratic-Gaussian (LQG) systems

All simulations described in the main text are instances of the following general model:

$$\begin{aligned} \text{Dynamics} \quad \mathbf{x}_{t+1} &= A\mathbf{x}_t + B\mathbf{u}_t + \sum_{i=1}^k C_i \mathbf{u}_t \varepsilon_{i,t} \\ \text{Feedback} \quad \mathbf{y}_t &= H\mathbf{x}_t + \boldsymbol{\omega}_t \\ \text{Cost} \quad 0 &\leq \mathbf{x}_t^T Q_t \mathbf{x}_t + \mathbf{u}_t^T R \mathbf{u}_t \end{aligned} \tag{1}$$

where the $\varepsilon_{i,t}$ terms are independent standard normal random variables, and C_i are constant matrices. The sensory noise terms $\boldsymbol{\omega}_t$ are independent multivariate normal random variables with mean 0 and covariance matrix Ω^ω . The initial state \mathbf{x}_1 has multivariate normal distribution with mean $\hat{\mathbf{x}}_1$ and covariance Σ_1 . The optimal control problem is the following: given $A, B, C_1, \dots, C_k, \Sigma_1, H, \Omega^\omega, R, Q_1, \dots, Q_T$, find the control law $\mathbf{u}_t = \boldsymbol{\pi}(\hat{\mathbf{x}}_1, \mathbf{u}_1, \dots, \mathbf{u}_{t-1}, \mathbf{y}_1, \dots, \mathbf{y}_{t-1}, t)$ which minimizes the expected cumulative cost $E_{\varepsilon, \omega} \sum_{t=1}^T (\mathbf{x}_t^T Q_t \mathbf{x}_t + \mathbf{u}_t^T R \mathbf{u}_t)$ over the time interval [1; T]. Time is expressed in units of 10msec, which is the discrete time step we use.

When the system noise in Eq 1 is additive rather than multiplicative, the LQG problem has a well-known solution¹, which involves recursive linear state estimation (Kalman filtering) and linear mapping from estimated states $\hat{\mathbf{x}}_t$ to optimal control signals \mathbf{u}_t . In the case of multiplicative noise, we have derived² the following iterative algorithm for solving this problem. The state estimate is updated using a modified Kalman filter which takes into account the multiplicative noise. For a given control law L_t , the corresponding Kalman filter is:

$$\begin{aligned}
\hat{\mathbf{x}}_{t+1} &= A\hat{\mathbf{x}}_t + B\mathbf{u}_t + K_t(\mathbf{y}_t - H\hat{\mathbf{x}}_t) \\
K_t &= A\Sigma_t^e H^T (H\Sigma_t^e H^T + \Omega^\omega)^{-1} \\
\Sigma_{t+1}^e &= (A - K_t H)\Sigma_t^e A^T + \sum_n C_n L_t \Sigma_t^{\hat{\mathbf{x}}} L_t^T C_n^T; \quad \Sigma_1^e = \Sigma_1 \\
\Sigma_{t+1}^{\hat{\mathbf{x}}} &= K_t H \Sigma_t^e A^T + (A - BL_t)\Sigma_t^{\hat{\mathbf{x}}}(A - BL_t)^T; \quad \Sigma_1^{\hat{\mathbf{x}}} = \hat{\mathbf{x}}_1 \hat{\mathbf{x}}_1^T
\end{aligned} \tag{2}$$

The matrices $K_t, \Sigma_t^e, \Sigma_t^{\hat{\mathbf{x}}}$ correspond to the Kalman gain, the expected estimation error covariance, and the non-centered covariance of the state estimate. Note that computing the unknown matrices in Eq 2 requires a single forward pass through time.

For a given Kalman filter K_t , the optimal control law is:

$$\begin{aligned}
\mathbf{u}_t &= -L_t \hat{\mathbf{x}}_t \\
L_t &= \left(B^T S_{t+1}^{\mathbf{x}} B + R + \sum_n C_n^T (S_{t+1}^{\mathbf{x}} + S_{t+1}^e) C_n \right)^{-1} B^T S_{t+1}^{\mathbf{x}} A \\
S_t^{\mathbf{x}} &= Q_t + A^T S_{t+1}^{\mathbf{x}} (A - BL_t); \quad S_T^{\mathbf{x}} = Q_T \\
S_t^e &= A^T S_{t+1}^{\mathbf{x}} BL + (A - K_t H)^T S_{t+1}^e (A - K_t H); \quad S_T^e = 0
\end{aligned} \tag{3}$$

The matrix L_t is the time-varying feedback gain, and $S_t^e, S_t^{\mathbf{x}}$ are the parameters specifying the optimal cost-to-go function (see² for details). Computing the unknown matrices in Eq 2 requires a single backward pass through time.

To obtain the Kalman filter and control law optimal with respect to each other, we iterate Eq 2 and 3 until convergence. We have found numerically² that the iteration always converges exponentially, and to the same answer (regardless of initialization). If the multiplicative noise in Eq 1 is removed, the algorithm converges after one iteration and becomes identical to the classic LQG solution¹.

Note that the above formulation implies a sensory-motor delay of one time step, because the sensory feedback is received after control signal has been generated. It is straightforward to modify the problem specification so as to include an additional delay of d time steps. This was done by using an augmented state $\tilde{\mathbf{x}}_t \triangleq [\mathbf{x}_t; H\mathbf{x}_{t-d}; \dots H\mathbf{x}_{t-1}]$ and transforming all matrices accordingly. In particular, the new observation matrix \tilde{H} extracts the component $H\mathbf{x}_{t-d}$ of $\tilde{\mathbf{x}}_t$, and new dynamics matrix \tilde{A} removes $H\mathbf{x}_{t-d}$, shifts the remaining sensory readings, and includes $H\mathbf{x}_t$ in the next state $\tilde{\mathbf{x}}_{t+1}$.

2. Application to a 2D via-point task

We now illustrate how the above general framework can be specialized for a via-point task, and explain the parameters settings used in the simulations. Consider a 2D point mass $m = 1 \text{ kg}$ with position $p_x(t), p_y(t)$, driven by a pair of actuators that produce forces $f_x(t), f_y(t)$ along the x and y axes respectively (each actuator can both pull and push). The force output $f_{x/y}(t)$ of each actuator is obtained by applying a first-order linear filter ($\tau = 40 \text{ msec}$) to the corresponding neural control signal $u_{x/y}(t)$, polluted with multiplicative noise. In Sim 1-6 we actually used second-order linear muscle filters³, with time constants $\tau_1 = \tau_2 = 40 \text{ msec}$.

The task is to pass through a specified via-point $p_x^*(T/2), p_y^*(T/2)$ in the middle of the movement, and then end the movement at a specified end-point $p_x^*(T), p_y^*(T)$. Therefore the task error will be defined as:

$$\frac{1}{4} \left(\sum_{i=x,y} \sum_{t=T/2,T} \left(p_i^*(t) - p_i(t) \right)^2 + \sum_{i=x,y} \left(w_v \dot{p}_i(T) \right)^2 + \sum_{i=x,y} \left(w_f f_i(T) \right)^2 \right)$$

The first term enforces passing through the targets, while the last two terms enforce stopping (i.e. zero velocity and force) at time T. The scale factor 1/4 corresponds to the fact that we have 4 task constraints (two positional, one velocity, and one force). In simulations with P positional constraints, this scale factor becomes 1/(P+2). The weights $w_v = 0.1, w_f = 0.01$ define the relative importance of stopping; their magnitudes are constant in all simulations, and based on the fact that for the tasks of interest, velocities are an order of magnitude larger than displacements, and forces are an order of magnitude larger than velocities (expressed in compatible units of m, m/s, N).

The effort penalty is:

$$\frac{r}{T} \left(\sum_{t=1}^T u_x(t)^2 + u_y(t)^2 \right)$$

The scalar r sets the tradeoff between task error and effort. When r is made too large, the optimal strategy is not to move at all. Therefore we set r to a value that is not large enough to cause unrealistic negative biases, but still has some effect on the simulations. In Sim 1-6 we used $r = 0.002$; in the telescopic arm model (Sim 7-10) we had to

decrease that parameter to $r = 0.00002$ because the large mass, gravity, and actuator visco-elasticity required much larger control signals.

We discretize time at $\Delta t = 10 \text{ msec}$, and represent the system state with the 10-dimensional column vector:

$$\mathbf{x}_t = \left[p_x(t); p_y(t); \dot{p}_x(t); \dot{p}_y(t); f_x(t); f_y(t); p_x^*(T/2); p_y^*(T/2); p_x^*(T); p_y^*(T) \right]$$

Since we are dealing with an inertial system, the state has to include position and velocity; force is included because the linear filters describing the force actuators have their own state (for a second-order filter we need two state variables per actuator); the target positions are included (and propagated through time) so that the task error can be defined as a function of the state. As explained above, the initial state \mathbf{x}_1 is distributed as $N(\hat{\mathbf{x}}_1; \Sigma_1)$. The mean $\hat{\mathbf{x}}_1$ contains the average initial position, velocity, and force, as well as the target positions. The covariance Σ_1 encodes the uncertainty of the initial state. In all our simulations the target positions are known exactly (and therefore not included in the sensory feedback); however, one could model them as being uncertain, and include (noisy) sensory feedback that allows the controller to improve the initial estimate of target positions. The initial state was variable (and therefore uncertain) in Sim 5 and 6; everywhere else we used a constant initial state ($\Sigma_1 = 0$).

The noisy sensory feedback carries information about position, velocity, and force:

$$\mathbf{y}_t = \left[p_x(t); p_y(t); \dot{p}_x(t); \dot{p}_y(t); f_x(t); f_y(t) \right] + \boldsymbol{\omega}_t$$

In Sim 1-6, the feedback was delayed by 4 time steps (in addition to the one-step implicit delay – see Section 1) resulting in 50msec delay. In Sim 7-10 no extra delay was introduced.

The sensory noise terms in the vector $\boldsymbol{\omega}$ are independent 0-mean Gaussians, with standard deviations

$$\sigma_s [0.01 \text{ m}; 0.01 \text{ m}; 0.1 \text{ m/s}; 0.1 \text{ m/s}; 1 \text{ N}; 1 \text{ N}]$$

The relative magnitudes of the standard deviations are determined using the above order-of-magnitude reasoning.

The overall sensory noise magnitude is $\sigma_s = 0.4$ in Sim 1-6, and $\sigma_s = 0.5$ in Sim 7-10.

The control signal is:

$$\mathbf{u}_t = \left[u_x(t); u_y(t) \right]$$

and the multiplicative noise added to the control signal is:

$$\sigma_u \begin{bmatrix} \varepsilon_t^1 & \varepsilon_t^2 \\ -\varepsilon_t^2 & \varepsilon_t^1 \end{bmatrix} \mathbf{u}_t$$

Multiplying \mathbf{u}_t by the above stochastic matrix produces 2D Gaussian noise with circular covariance, whose standard deviation is equal to the length of the vector \mathbf{u}_t . In Sim 1-6 the scale factor was set to $\sigma_u = 0.4$, while in Sim 7-10 its value was $\sigma_u = 0.5$. As explained in the main text, the two parameters σ_s and σ_u were adjusted so that the overall variability generated by the optimal control law roughly matched all experimental observations we model. The noise magnitudes in Sim 1-6 were smaller, because in those simulations we included a sensory-motor delay which effectively increases the noise.

The discrete-time dynamics of the above system is given by:

$$\begin{aligned} p_{x/y}(t + \Delta t) &= p_{x/y}(t) + \dot{p}_{x/y}(t) \Delta t \\ \dot{p}_{x/y}(t + \Delta t) &= \dot{p}_{x/y}(t) + m^{-1} f_{x/y}(t) \Delta t \\ f_x(t + \Delta t) &= e^{-\Delta t/\tau} f_x(t) + u_x(t) + (u_x(t) \varepsilon_t^1 + u_y(t) \varepsilon_t^2) \sigma_u \\ f_y(t + \Delta t) &= e^{-\Delta t/\tau} f_y(t) + u_y(t) + (u_y(t) \varepsilon_t^2 - u_x(t) \varepsilon_t^1) \sigma_u \end{aligned}$$

which is transformed in the form of Eq 1 by the matrices:

$$A = \begin{bmatrix} 1 & . & \Delta t & . & . & . & 0_{6 \times 4} \\ . & 1 & . & \Delta t & . & . & \\ . & . & 1 & . & m^{-1} \Delta t & . & \\ . & . & . & 1 & . & m^{-1} \Delta t & \\ . & . & . & . & e^{-\Delta t/\tau} & . & \\ . & . & . & . & . & e^{-\Delta t/\tau} & \\ 0_{4 \times 6} & & & & & & I_{4 \times 4} \end{bmatrix}$$

$$B = \begin{bmatrix} 0_{4 \times 2} \\ I_{2 \times 2} \\ 0_{4 \times 2} \end{bmatrix} \quad C_1 = B \begin{bmatrix} \sigma_u & 0 \\ 0 & \sigma_u \end{bmatrix} \quad C_2 = B \begin{bmatrix} 0 & \sigma_u \\ -\sigma_u & 0 \end{bmatrix}$$

The sensory feedback matrix is $H = [I_{6 \times 6} \quad 0_{6 \times 4}]$.

The effort penalty matrix is $R = \frac{r}{T} I_{2 \times 2}$.

The matrices Q_t specifying the task constraints are 0 for all $t \neq T/2, T$. The task error at the via-point is encoded by:

$$Q_{T/2} = \frac{1}{4} D_{via}^T D_{via}; \quad D_{via} = \begin{bmatrix} -1 & . & . & . & . & . & . & 1 & . & . & . \\ . & -1 & . & . & . & . & . & . & 1 & . & . \end{bmatrix}$$

The task error at the end-point is encoded by:

$$Q_T = \frac{1}{4} D_{end}^T D_{end}; \quad D_{end} = \begin{bmatrix} -1 & . & . & . & . & . & . & . & . & 1 & . \\ . & -1 & . & . & . & . & . & . & . & . & 1 \\ . & . & w_v & . & . & . & . & . & . & . & . \\ . & . & . & w_v & . & . & . & . & . & . & . \\ . & . & . & . & w_f & . & . & . & . & . & . \\ . & . & . & . & . & w_f & . & . & . & . & . \end{bmatrix}$$

To encode a trajectory-tracking task we would specify targets at many points in time (e.g. P points). In that case, keeping all target positions in the state vector is inefficient. Instead, we append the constant 1 to the state vector, and enforce the spatial constraints using matrices of the form:

$$Q_t = \frac{1}{P+2} D_t^T D_t; \quad D_t = \begin{bmatrix} -1 & . & \dots & p_x^*(t) \\ . & -1 & \dots & p_y^*(t) \end{bmatrix}$$

Note that this approach makes it impossible to model target uncertainty (which we do not model here).

3. Simulations

We now describe each of the 10 simulations illustrated in the main text. The matrix notation will no longer be shown, but it is straightforward to adapt the above example to each specific model. Note that the parameters common to all models were already described; here we only list the task-specific parameters.

Sim 1. A 2D point mass (1kg) was initialized at position (0.2m; 0.2m), and required to make a movement that ends in 50 time steps (stopping as described above). The point mass was driven with two force actuators modelled as second-order linear filters. The task error term specified that the movement has to end on the line passing through the origin and oriented at -20° : $\left(\tan(-20^\circ) p_x(50) - p_y(50) \right)^2$.

Sim 2. Two 1D points masses (1kg each) were simulated, each driven with one second-order force actuator. Initial positions were $p_1(1) = -0.1\text{ m}$; $p_2(1) = 0.1\text{ m}$. The movement had to stop after 50 time steps (stopping enforced as before). The task error term specified that the two points have to end the movement at identical locations: $(p_1(50) - p_2(50))^2$.

Sim 3. This simulation was identical to the via-point task described in detail above, except that the number of via points was varied. Target locations are given in Fig 3A in the main text. In the 5 target condition A we set the movement duration to 1520msec as observed experimentally. Then we found numerically the intermediate-target passage times that minimized the total expected cost. The optimal passage times (460msec, 750msec, 1050msec) were close to the experimental measurements (400msec, 720msec, 1040msec). The passage times for the 21 target condition B were set to the times when the average trajectory from condition A passed nearest to each target (i.e. we modeled conditions A and B with identical timing).

Note that the time-window allowed in the experiment (1.2sec - 1.5sec) was measured from the time when the hand left a 2 cm diameter start region – at which point hand velocity was already substantial. In the data analysis, we defined movement onset as the point in time when hand velocity first exceeded 1cm/sec – and so the measured durations appear longer than allowed.

Sim 4. This simulation was also identical to the above via-point task, except that the spatial error at the smaller target was scaled by a factor of 2 – corresponding to the fact that the smaller target diameter was 50% of the diameter of the remaining targets. Target locations are given in Fig 3B in the main text. The predefined target passage times (550msec, 950msec, 1400msec) were in the observed range.

Sim 5. A 1kg 2D point mass (the “hand”) started moving from average position (1m, 0.3m), sampled from a circular 2D Gaussian with standard deviation 0.04m. The task error term specified a positional constraint (release region) at time 750msec and location (0.7m; 0m). The movement had to stop (stopping enforced as before) at time 900msec and unspecified location. Throwing was modelled by initializing the “ball” with the position and velocity of the hand observed at time 750msec. The task error term specified that the ball has to be at the target (2.2m, 0m) after flying with constant velocity for 500msec. The locations, times, and initial position variability roughly matched those observed experimentally.

Sim 6. The average trajectory of the task-optimal feedback controller from Sim 5 was used as a desired trajectory for an optimal trajectory-tracking controller. This was done by computing the average positions at 10 points equally spaced in time, and using them as spatial targets to form the task error term. Stopping was not enforced explicitly. The optimal feedback controller for the new task error was then computed using the above method.

Sim 7. The telescopic arm model used in Sim 7-10 is described in Figure 1. The 4-targets task required the end-effector to pass through targets ($P+0.3m$; $P+0.3m$; $P-0.3m$; P_m) at times (250msec; 500msec; 750msec; 1000msec), where P is the initial position of the end-effector ($P = M \times 0.3m$ as explained in the figure). Stopping at the final target was not required. This task was simulated for mechanical systems with different number (M) of point masses.

Sim 8. The task-optimal controller described in Sim 7 was constructed, and its average trajectory computed on several levels of description: end-effector, individual joint “angles”, individual actuator forces, and individual control signals. These average trajectories were then used to form optimal trajectory-tracking controllers. The control-signal tracking controller was simply an open-loop controller producing the average time-varying control signals of the task-optimal controller. For the remaining tracking controllers, the task error specified a target at each time step.

Sim 9. The end-effector of the $M=2$ system was required to track a specified sinusoid, with modulation $\pm 0.1m$, centered at the initial $0.6m$ position. An end-effector positional target was specified at each time step, for a total of 500 time steps. Stopping was not required. A different optimal controller was constructed for each oscillation frequency in the range $1.5Hz - 4Hz$, at $0.1Hz$ increments. In the perturbation experiment, an independent random number sampled from $\mathcal{N}(0; 30^2)$ was added to each signal, for 1 time step.

Sim 10. The postural task required the end-effector of the $M=2$ system to remain at the initial $0.6m$ position indefinitely. The stationary feedback control law was initialized to an open-loop control law, and gradually improved using the nonlinear simplex method in Matlab. The cost of each control law was evaluated using a Monte Carlo method (100 trials, 2 sec each, first 1 sec discarded). To speed up learning, the seed of the random number generator was reset before each evaluation⁴. Learning was interrupted after 5000 evaluations. Average results from 5 runs with different seeds are shown in the main text.

4. Additional analysis of Experiment 1

As stated in the main text, the average behavior in Experiment 1 was different between the 5-target condition A and the 21-target condition B. Here we test the possibility that the desired trajectory hypothesis can explain the observed difference in variability, given the difference in average behavior. For each condition, we built an optimal trajectory-tracking controller that reproduced the experimentally observed average path, speed profile, and duration. This was done by extracting from the average trajectory the locations and passage times of 21 equally spaced (in time) points, and building the optimal feedback controller for the resulting tracking task. Then we iteratively adjusted the specified target locations, until the average trajectory of the optimal controller matched the observed average trajectory. The latter was done iteratively, by adding to each (adjustable) target the vector connecting the data-extracted target with the nearest point on the average trajectory. The procedure converged in a couple of iterations; the resulting average trajectory of the optimal tracking controller was indistinguishable from the average experimental trajectory. The paths and speed profiles for each subject, the tracking controllers, and the 5-target optimal controller from the main text, are compared in Figure 2A,B.

In Figure 2C we plot the positional variance predicted by the two tracking controllers, and the variance predicted by the model in the main text. The variability predicted by the tracking controller for condition A is larger than the variability of the condition B controller – because the movement in condition A was faster, and therefore the multiplicative noise was larger. The difference, however, is a uniform offset rather than a change in modulation. In the main text we showed that the variability observed in conditions A and B differs in modulation, i.e. it increases at the intermediate targets and decreases at the midpoints between them. Thus the desired trajectory hypothesis cannot explain our results.

5. Analysis of Experiment 2

The change-in-modulation effect predicted by our model and observed in Experiment 1 was also confirmed by reanalyzing data from the previously published⁵ Experiment 2. In that experiment, 8 subjects were asked to move through sequences of 6 targets (condition A) or trace smooth curves projected on the table (condition B). Since our earlier experimental design pursued different goals, the stimuli were not adjusted so that the average trajectories in conditions A and B would match. Therefore the test here is less direct than in Experiment 1. The advantage of Experiment 2 is that we presented 6 different target configurations and 8 different smooth curves (in blocks of 10

consecutive trials each) – and so any effects due to the specific geometric shape of the movement trajectory should average out.

For each subject and block of trials, we computed the positional variance along the path as described in the main text. Then we defined a modulation index, which was the difference between the maximum and minimum variance, divided by the mean variance (all computed over the middle 60% of the path). This index of variance modulation was larger ($p < 0.01$) in the multiple target condition (2.10 ± 0.11) compared to the curve-tracing condition (1.43 ± 0.08). Thus, again, we see that moving through a small number of targets is accomplished by allowing increased variability between those targets.

6. Motor Adaptation

Optimal feedback control requires optimal state estimation, which in turn reposes on the ability to calibrate the internal models underlying the state estimation process. To account for motor adaptation we therefore consider the following natural extension of our theory: suppose that the internal model of the sensory-motor apparatus is continuously calibrated, and the feedback control law changes accordingly so as to remain optimal with respect to the current set of calibration parameters. What kinds of predictions does this extension make regarding adaptation? Unfortunately, in our framework it is difficult to answer such questions in the abstract – we need to specify a detailed quantitative model of both the dynamic and kinematic effects of perturbations, including the adaptive response of the nervous system to such perturbations. Indeed, in our approach: 1) the relationship between plant dynamics and optimal behavior is nonlinear and generally quite complex, such that changes to the dynamics can yield non-intuitive changes to optimal behavior; 2) the adapted behavior is optimal with respect to the peripheral changes that the nervous system *believes* to have taken place – which can be quite different from the experimental perturbations as conceived of by the experimenter, especially when the latter are ecologically implausible and experienced only briefly.

As an illustration of these issues, and the complexities that they introduce into the interpretation of experimental data, consider a nonlinear visual perturbation that makes straight reaching movements appear curved, but does not affect the perceived endpoint position⁶. In this setting subjects show partial (approximately 25%) adaptation of the trajectory. One might think that our theory predicts no adaptation, because our cost function only specifies a desired final state and not a desired trajectory. But in fact there are a variety of equally plausible

assumptions about the response of the nervous system to this visual perturbation that yield quite different predictions in this case, and the proper interpretation of the experiment is far from clear. Let us carefully consider the events that take place – not from the experimenter's point of view, but from the point of view of the subject's visuo-motor system. A systematic discrepancy is detected between the expected and received visual feedback, and therefore an internal model needs to be calibrated in order to account for this oddity. But which model? Is it the generative model of how sensory data reflects the system state, or is it the dynamic model of how the system state evolves as a function of the control signals? From the point of view of the experimenter, the correct interpretation is to adapt the generative model alone, in which case the nervous system should believe that the physical movement proceeds as normal, predicting a lack of adaptation under any hypothesis of motor control. But the visual distortion introduced in⁶ is extremely unlikely to occur in the real world. Indeed, the fact that there is partial adaptation of the trajectory shows that the discrepancy is at least partially interpreted as a dynamic change. That is, the nervous system believes that the same control signals now cause different physical movements.

Is the optimal control law for the inferred dynamics different from baseline? It probably is. Even if the inferred dynamical change does not cause endpoint bias under the baseline control law, it may lead to suboptimal endpoint variance as well as suboptimal energy consumption – both of which are penalized by our cost function. Indeed, in unpublished preliminary work we have designed two different force fields that (in conjunction with the baseline optimal feedback controller) can qualitatively explain the perceived curvature in⁶. Both of these force fields lead to changes in the optimal feedback controller, and both yield partial adaptation in accordance with the experimental results. Of course there may exist other force fields that are consistent with the visual perturbation and do not cause adaptation. Evaluating these possibilities will require the development of a detailed quantitative model, comparison to a range of experimental data, and great care in teasing apart the complex interactions between estimation, control and adaptation.

References

1. Davis,M.H.A. & R.B.,V. Stochastic Modelling and Control. Chapman and Hall, (1985).
2. Todorov,E. Optimal control under signal-dependent noise: Methodology for modeling biological movement. *Neural Computation*, under review.
3. Winter,D.A. Biomechanics and Motor Control of Human Movement. John Wiley and Sons, Inc., New York (1990).
4. Ng,A. & Jordan,M. PEGASUS: A policy search method for large MDPs and POMDPs. *Uncertainty in artificial intelligence* **16**, (2000).
5. Todorov,E. & Jordan,M.I. Smoothness maximization along a predefined path accurately predicts the speed profiles of complex arm movements. *Journal of Neurophysiology* **80**, 696-714 (1998).
6. Wolpert,D.M., Ghahramani,Z. & Jordan,M.I. Are Arm Trajectories Planned in Kinematic Or Dynamic Coordinates - An Adaptation Study. *Experimental Brain Research* **103**, 460-470 (1995).

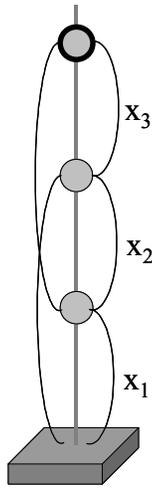


Figure 1. M point masses (1kg each) were sliding up and down a frictionless vertical pole in the presence of gravity (9.8 m/sec^2). Points $0:1, 1:2, \dots M-1:M$ (0 being the immovable base) were connected with “single joint” linear actuators that could both pull and push. The lengths $X_1, X_2, \dots X_M$ of the single-joint actuators correspond to joint “angles”. Points $0:2, \dots M-2:M$ are connected with “double-joint” actuators. The last point mass (heavy outline), at position $X_1 + X_2 + \dots + X_M$, was defined as the end-effector. The $2M - 1$ actuators had built-in viscosity (10 Nsec/m) and elasticity (50 N/m), with resting lengths of 0.3m for the single-joint actuators and 0.6m for the double-joint actuators. The system was always initialized at the resting lengths of all actuators; note however that the presence of gravity required control signals in order to maintain that configuration. All actuators were modelled as first-order linear filters (40msec time constant), each polluted with independent multiplicative noise (50% of the control signal). The noisy feedback included the length, velocity, and force output of each actuator. The effort penalty term was $r = 0.00002$ as explained above.

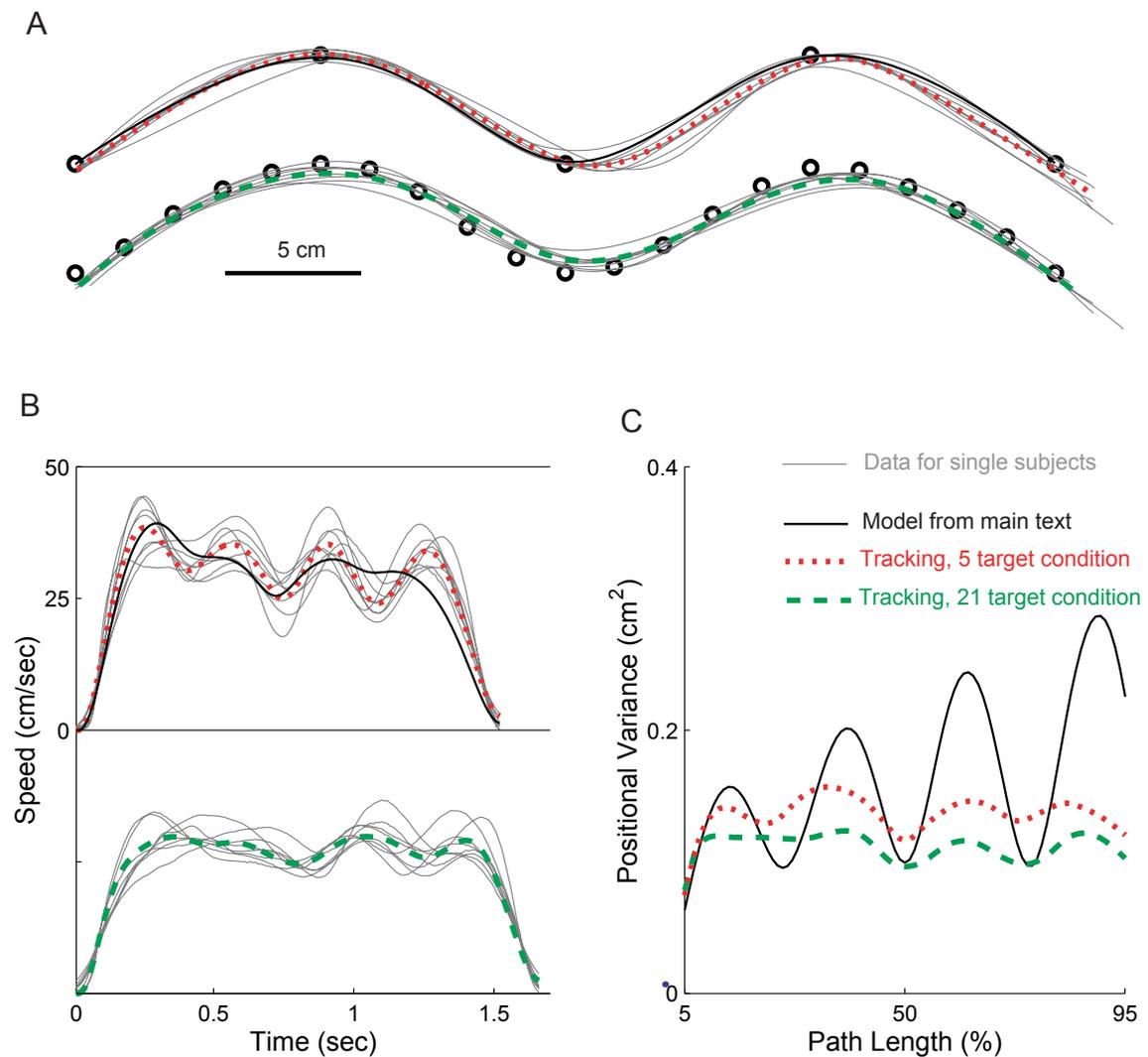


Figure 2.

A– The average trajectory for each subject in each condition (condition A – top, condition B – bottom) is shown in grey. Circles mark the target locations that the subjects saw during the experiment. Red-dotted and green-dashed lines are the average trajectories of the two trajectory-tracking controllers designed to reproduce the average behavior of the subjects (see text). Black is the average trajectory for the optimal controller in condition A described in the main text. **B**– Average speed profiles for each subject and each simulation. Average movement duration (determined using a 1cm/sec velocity threshold for start and stop) was 1520msec in condition A, and 1660msec in condition B; the movement duration for each subject was scaled to match the average duration in the corresponding condition. **C**– Positional variance along the path for each model (variance was computed as described in the main text).