# Stochastic optimal control methods for uncertain predictive reaching movements

Alex Simpkins, Dan Liu, and Emo Todorov

*Abstract*— **People learn from the varying environment, and adapt their control strategy accordingly. This paper explores two experiments and two methodologies for modeling human behavior on these tasks within the framework of stochastic optimal control. In addition, both methods are of interest to the broader control community because they can be used to solve interesting nonlinear stochastic optimal control problems where estimating parameters which allow predictive actions can minimize cost and improve performance in the system. The first method is based on a Markov Decision Process, which is a discrete solution, while the second method is based upon a continuous function approximation scheme. This second method augments the states with the Kalman filter estimates, making the problem fully observable. Control performance is compared to human subject behavior.**

## I. Introduction

To investigate how people learn from the varing environment and adapt both the estimator and controller accordingly, the target jump paradigm has been widely used in the biological motor control field. Such experimental paradigms consist of presenting a small visual target to be reached for and of changing the target position during the movement. Subjects are informed that target may be purturbed and are required to adjust their reaching movement towards the new target location within some time limit [4], [7]. Previous studies have shown that the hand path is smoothly corrected to reach the displaced target when the perturbation is introduced early, whereas such correction is incomplete when the perturbation happens late in the movement. In a recently study, an optimal feedback control model was developed to capture both phenomena. [7]. The authors suggested that the key is the fact that the optimal feedback gains are time varying. In another words, the controller is adapting to the constraints in these tasks. However, in all these studies the direction of target jump was unpredictable, which leaves the estimator unstudied. Apparently, people can learn from the statistics of the world and use such prediction in everyday movement. For a better comprehension of the characteristics of motor adaptation, especially how the estimator and controller work together, two experiments were conducted, where the target was displaced following certain distributions instead of being random. The first experiment was to study how the statistics of target jump may affect the uncertainty of the estimation and therefore the controller, and the second experiment aimed to study how perturbation time affects the way people integrate their estimation and the on-line visual feedback in controlling hand movement.

## II. Model development

The reaching task was modeled using a variety of control methods. Some are more optimal with respect to this task than others, but more complicated.

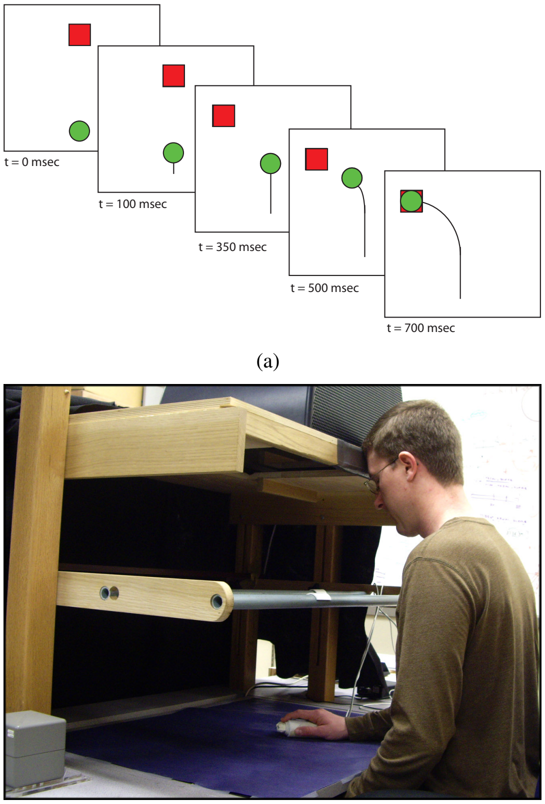
### A. Reaching task

Figure 1 (b) shows the experimental setup, the subject was making planar reaching movements on a table positioned at chest level. A 21 inch flatscreen monitor was mounted above the table facing down and was viewed in a see-through horizontal mirror. In this way computer-generated images could be physically aligned with the hand workspace. The subject held in his right hand a small pointer which was represented by a 2-cm square on the computer monitor, and the target was represented by a 2-cm diameter ball. The task was to move the pointer cursor to a starting position, wait for the target to appear, and move to the target when ready. During the movement, target was either stationary or displaced rapidly to the left 10cm or right 10cm, perpendicular to the main movement. Figure 1(a) shows an example where target jump to the left at 350msec and the allowed time is 700msec. Let $p_j$ represent the probability vector of jumping directions, where the first to the third element represnts the probability of jumping to the left, middle and right. The first experiment consisted of two blocks with different distributions of jump: $p_{j\_two\_peaks} = [0\ 0.5\ 0.5\ ]$, $p_{j\_three\_peaks} = [0.2\ 0.1\ 0.7]$. The two distributions were set this way so that their mean was the same but the variance differed. For both blocks, perturbation was introduced late during movement. In the second experiment, $p_j = [0.1\ 0.1\ 0.8]$ but the perturbation happened either at the onset of movement (early block) or late during the movement (late block).

### B. Modeling the task with stochastic optimal control with a single Gaussian (SOCG)

Let $h(t) \in \Re^{n_h}$ denote a fully observable state variable which in this case represents hand position. Let $s(t) \in \Re^{n_s}$ denote a target which needs to be tracked. Let $u(t) \in \Re^{n_h}$ be the control signal which modifies the hand position directly. Let $m(t) \in \Re^{n_s}$ be the estimated final target location. $\omega_s(t)$ and $\omega_m(t)$ represent Brownian motion processes with covariances $\Omega_s$ and $\Omega_m$, respectively.

But consider that, in this case, it may be more optimal (a target may be more readily tracked by learning the probable target destination and predictively moving toward that estimated state before the target jump occurs. However, initially there is no model of this probable destination, so a sensible cost function includes both feedback using the actual



(a)

(b)

Fig. 1. Experimental setup

target location and feedback using the estimated target final state.

The dynamics are assumed linear gaussian,

$$\begin{aligned} dh &= udt \qquad (1)\\ ds &= d\omega_s \\ dm &= d\omega_m. \end{aligned}$$

We assume that $h(t)$ and $s(t)$ are directly observable, but $m(t)$ is not directly observable and needs to be estimated.

The observation process is

$$dy = m(t)dt + d\omega_y \qquad (2)$$

where $y(t)$ corresponds to the integral of the noisy final target position. Assuming the prior over the initial estimate is Gaussian, with mean $\widehat{m}(0)$ and covariance $\Sigma(0)$, the posterior over $m(t)$ remains Gaussian $\forall t > 0$. The optimal estimate of the mean and error covariance of the map is propagated by the Kalman-Bucy filter [1], [11] given the additive Gaussian white noise model:

$$\begin{aligned} d\widehat{m} &= K(dy - \widehat{m}(t)dt), \qquad (3)\\ K &= \Sigma(t)\Omega_y^{-1}, \\ d\Sigma &= \Omega_m dt - K(t)\Sigma(t)dt. \end{aligned}$$

The mean and covariance of the state estimate is $\widehat{m}(t)$ and $\Sigma(t)$, respectively, and $d\omega_y$ is a white, zero-mean Gaussian random process as well, with covariance $\Omega_y$. $d\omega_m$ and $d\omega_y$ are assumed to be uncorrelated. The Kalman filter can be written in innovations form by expressing $\widehat{m}(t)$ as another stochastic process:

$$d\widehat{m} = K d\omega_{\widehat{m}}. \qquad (4)$$

Here again $\omega_{\widehat{m}}(t)$ is a standard Brownian motion process with unit covariance.

Now $\widehat{m}(t)$ and $\Sigma(t)$ act as state variables, and we are dealing with a fully observable system. $\Sigma(t)$ is a symmetric matrix, defined uniquely by its upper-triangular part. Let $\sigma(t) \in \Re^{n_s}$ represent the vector of upper-triangular elements of $\Sigma(t)$.

We now define a composite state vector of our system which captures the mean and covariance of our target jump location estimate.

$$x(t) = [h(t); s(t); \widehat{m}(t); \sigma(t)] \qquad (5)$$

and we write the stochastic dynamics [12] in control affine form

$$dx = (a(x) + Bu)dt + C(x)d\omega \qquad (6)$$

The uncontrolled dynamics $a(x)$ represent the evolution of the covariance matrix (note: explicit time dependence is temporarily dropped in the next three equations for clarity):

$$a(x) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ f\left[\Omega_m - \Sigma^{\mathsf{T}}\Omega_y^{-1}\Sigma\right] \end{bmatrix}. \qquad (7)$$

The controlled dynamics $Bu$ capture the evolution of the hand state:

$$B = \begin{bmatrix} I & 0 & 0 & 0 \end{bmatrix}^T. \qquad (8)$$

The noise-scaling matrix $C(x)$ captures the dependence of the innovation process on the filter gain matrix, as well as the covariance of the target movement:

$$C(x) = \begin{bmatrix} 0 & & & \\ & \sqrt{\Omega_s} & & \\ & & \Sigma^{\mathsf{T}}\Omega_y^{-1} & \\ & & & 0 \end{bmatrix}. \qquad (9)$$

Since we have what can be thought of as a tracking task (with large initial error), one obvious term for the cost rate is

$$||h(t) - s(t)||^2.$$

The quadratic function produces fair controllers, but improvements are provided by a nonlinear falloff of the cost (i.e. penalty increases steeply as the hand-target error is increased). Thus, a more appropriate cost relation is

$$\beta\left(1 - e^{-\frac{||h-s||^2}{\gamma}}\right) \qquad (10)$$

where $\beta$ and $\gamma$ It also stands to reason that prior knowledge of the probable final target location should be part of the control system, since some feedforward component can optimize task performance, assuming a good prediction of the final target location can be made. One such cost function includes a predictive target final location, as well as an uncertainty-weighting between terms,

$$\Big(\frac{||\Sigma||}{||\Sigma_{MAX}||}\Big)\beta\Big(1-e^{-\frac{||h-s||^2}{\gamma}}\Big)$$
$$+\Big(1-\frac{||\Sigma||}{||\Sigma_{MAX}||}\Big)\beta\Big(1-e^{-\frac{||h-\widehat{m}||^2}{\gamma}}\Big).$$

Define the weighting functions

$$W=\frac{||\Sigma||}{||\Sigma_{MAX}||} \tag{11}$$
$$\overline{W}=1-\frac{||\Sigma||}{||\Sigma_{MAX}||}$$

(where $\overline{(\cdot)}$ is used here to denote the complement). We can then add a quadratic control cost to make the cost rate

$$\ell\left(x,u\right)=W\beta\Big(1-e^{-\frac{||h-s||^2}{\gamma}}\Big) \tag{12}$$
$$+\overline{W}\beta\Big(1-e^{-\frac{||h-\widehat{m}||^2}{\gamma}}\Big)+\frac{1}{2}||u||^2.$$

### C. Markov Decision Process (MDP)

Since the LQG framework don't allow us to encode different distributions of target jump, here we tried to solve the optimal feedback control problem using a Markov Decision Process. First, the continuous state and action spaces were discretized [6] and the resulting discrete optimization problem is solved via dynamic programming [2]. To reduce the dimensionality, the arm is now modeled as a fully observable second-order plant with state vector containing hand position $p$ and velocity $v$ and control vector $u$ corresponding to hand acceleration. All quantities are expressed in units of centimeters and seconds. The initial state is$p(0)=v(0)=[0;0]$. The default target position is $p^*=[20;0]$ but can be perturbed to either $[20;-10]$or $[20;10]$. Instead of perturbing the target, we perturb the hand in the opposite direction (without changing hand velocity) and then correct the hand and target positions in the subsequent analysis. In this way, the target can be treated as constant and omitted from the state vector. Each trial ends when the horizontal hand position exceeds 20 cm (i.e., the hand reaches the target plane) or when the duration exceeds a maximum allowed duration of 0.7 s, whichever comes first. Let $t_f$denote the duration of a given trial. The total cost to be minimized, represented by $C$, is defined as follows:

$$C=c(t_f)+w_{energy}\int||u(t)||^2dt \tag{13}$$

The final cost $c(t_f)$, computed at the end of the movement, is defined as follows:

$$c(t_f)=\left\{\begin{array}{c} w_{time}t_f \text{ if } ||p^*-p(t_f)||<=1 \text{ and} \\ t_f<0.7 \text{ and } ||v(t_f)||<v_{max} \\ 100, \text{ otherwise} \end{array}\right\}$$

Noticed that the final cost is not in the quadratic form, as used in many models of reaching in sensorimotor control and learning [14],[13]. This is becuase subjects were rewarded when they hit the target, and punished when they missed the target. Also, recent studies suggest that people use a loss function in which the cost increases approximately quadratically with error for small errors and significantly less than quadratically for large errors [5]. Therefore, such hit-miss cost not only represents the experiments more precisely, but captures the robustness to outliers of our motor system.

## III. Solution methods

Two solution methods will be compared.

*1) **Markov Decision Process (MDP)**:*
The feedback control law is in the following general form:

$$u=\pi(p,v,t) \tag{14}$$

where $p$, $v$, and $t$ are constrained to the corresponding grids. Sincee we do not know in advance the form of $\pi$, we represent it as a lookup table that specifies the value of $u$ for every possible combination of $p$, $v$, and $t$. For each combination, $u$ is chosen to satisfy the finite horizon Bellman equation

$$v(x,k)=\min_u w_{energy}\|u\|^2+\underset{x'\tilde{}p(|x,u,k)}{E}[v(x',k+1)] \tag{15}$$

All the biologically related parameters were following [7] and the only two free parameters $w_{energy}$ and $w_{time}$ were adjusted to better fit the data.

*2) **SOCG**:*

Let us consider an infinite horizon discounted cost formulation with discount $\zeta>0$, for which the Hamilton-Jacobi-Bellman (HJB) equation is given by

$$\zeta v(x)=\min_u\Bigg\{q(x)+\frac{1}{2}\|u\|^2+(a(x)+Bu)^Tv_x \tag{16}$$
$$+\frac{1}{2}tr(C(x)C(x)^Tv_{xx})\Bigg\},$$

where the subscripts denote partial derivatives.

$$\pi(x)=-B(x)v_x(x). \tag{17}$$

The minimized HJB equation is found by dropping the *min* operator and substituting (17) into (16)

$$\zeta v(x)=q(x)+a(x)^Tv_x(x) \tag{18}$$
$$+\frac{1}{2}tr(C(x)C(x)^Tv_{xx}(x))-\frac{1}{2}\|\pi(x)\|^2.$$

Using the method of collocation [3] we will approximate a continuous time optimal control law. This method is similar to our recent paper[10]. This can be done with a general nonlinear (but linear in the to-be-determined parameters $w_i$) function approximator ( we will refer to this strategy as the function approximation scheme, or FAS)

$$v(x,w)=\sum_i w_i\phi^i(x)=\phi^T(x)w, \tag{19}$$

and its first and second derivatives,

$$v_x(x,w) = \sum_i w_i \phi_x^i(x) = \phi_x^T(x) w, \tag{20}$$

$$v_{xx}(x,w) = \sum_i w_i \phi_{xx}^i(x) = \phi_{xx}^T(x) w. \tag{21}$$

with $\{\phi^i\}$ a set of predefined features[9]. The features we choose [8] are a combination of Gaussians to fit fine details, and quadratics for the global approximation (vector to matrix conversions are implicit here):

$$\phi(x)_g^T w = \frac{1}{\sqrt{2\pi\sigma}} e^{-[(x-c)^T Q (x-c)]} w \tag{22}$$
$$\phi(x)_q^T w = x^T P x + Y^T x + S$$

where $Q = diag(1/var(x))$.

The nonlinear partial differential equation for $v(x,u)$ can be reduced to a linear least squares problem,

$$Rw = z \tag{23}$$

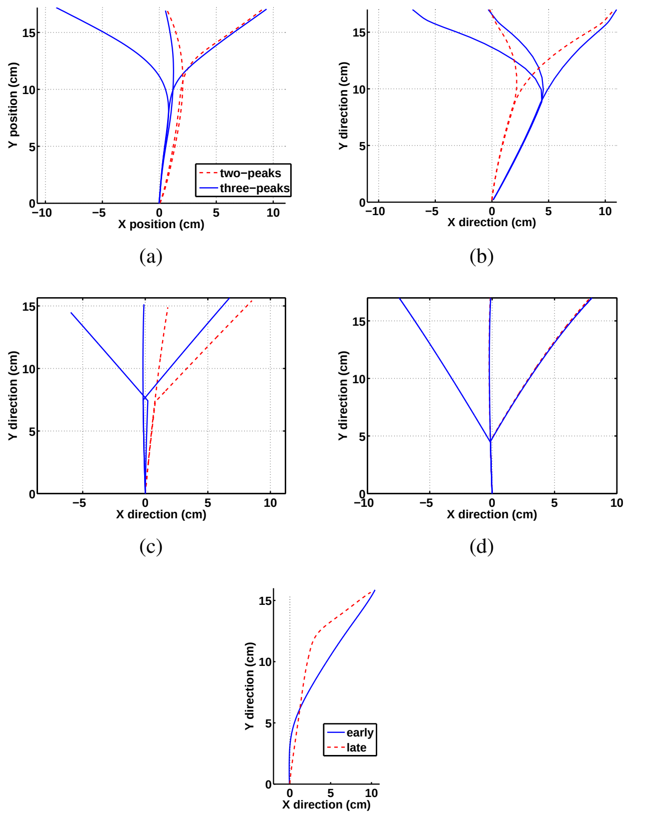
which can be solved to arbitrary precision. We get to (23) by starting with (18), collecting the cost and control terms on the right side, and those with weights on the left. Then we define

$$\mathbf{R} = \Big\{ R_{j,i} = \Big( \zeta \phi^i(x_j)^T - a(x_j)^T \phi_x^i(x_j)^T \tag{24}$$
$$- \frac{1}{2} tr\{C(x_j) C(x_j)^T \phi_{xx}^i(x_j)^T\} \Big), \forall i,j \Big\},$$

$$\mathbf{z} = \Big\{ z_j = q(x_j) - \frac{1}{2} \|\pi(x_j)\|^2, \forall j \Big\}. \tag{25}$$

Then computing the weights has been reduced to the least squares problem. Computing an approximately optimal controller consists of the following algorithm:

1) Randomly generate a set of states $\{x_j\}$ and centers for the Gaussians $\{c_{i_g}\}$ of appropriate range (with $n_j > n_i$).
2) Enforce that at all Gaussian centers have an associated state by
$$x_{1:length(c)} = c_{1:length(c)} \tag{26}$$
3) Compute and store $\phi^i(x_j)$, $\phi_x^i(x_j)$, $\phi_{xx}^i(x_j)$ $\forall i,j$
4) Initialize $w^0$ in one of several ways. One way that works well in practice is to set all $w$'s to 0
5) Initialize the policy :
$$\pi^0(x_j) = -B\phi_x(x_j)^T w^0. \tag{27}$$
6) Substitute the constraints $\phi^i(x_j)$, $\phi_x^i(x_j)$, and $\phi_{xx}^i(x_j)$ into (24), and $\pi^k(x_j)$ into (25) to obtain one constraint on $\mathbf{w}$ for every $j$.
7) Compute the least squares solution to (23) in one of several ways. For example, compute $R^\dagger$, where $()^\dagger$ represents the pseudo-inverse, then
$$w^{k+1} = R^\dagger z. \tag{28}$$
Store the $R^\dagger$ for all the rest of the computations, since it only needs to be computed once.
8) Compute $\pi^{k+1}(x_j)$ for every $j$ using (17)
9) Stop if the stopping criterion is met, otherwise go to step 6. Many criteria are possible, and the one used in the present results is
$$e^k = \frac{1}{n_j} \|Mw^k - d\|_2^2, \quad de^k = e^k - e^{k-1}, \tag{29}$$
$$if(\{e^k < \gamma\}|\{de^k < \beta\}) \rightarrow break, \tag{30}$$
where $\gamma = 10^{-3}$ and $\beta = 10^{-5}$ are tolerances . We also test for divergence:
$$if(\{(de^k) > \lambda\}|\{isnan(de^k) == true\}) \rightarrow break, \tag{31}$$
where $isnan$ is a test for invalid numbers, and $\lambda = 10^{-3}$ is a positive constant which is arbitrary, but can be on the order of one.



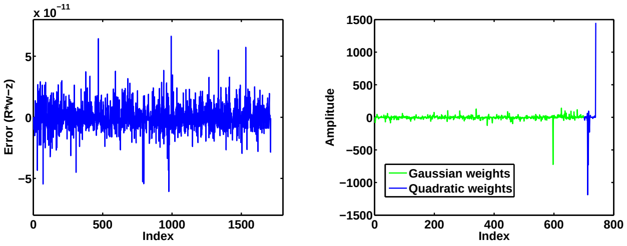
(a) (b) (c) (d) (e)

Fig. 2. Reaching data. (a) is human subjects averaged data for the second half of the two types of trials. (b) is the simulated trajectory created by the MDP solution, (c) and (d) are two forms of the FAS, the former without an uncertain mixing term, and the latter with an uncertain mixing term. (e) is human subject data due to the early-late jump paradigm.

## IV. Results

### A. Convergence of the FAS

The FAS algorithm converged to within 2e-11 in all cases, with random (small) weight initialization perturbations



(a) (b)

Fig. 3. (a) Bellman error, using a criterion of 1e-7. In all cases the FAS algorithm converged on the second iteration. (b) The FAS weights. The first 700 are the Gaussian feature weights, while the last 43 are the quadratic portion of the features. Note that, indeed, the quadratic terms dominate, fitting the general shape, while the Gaussians provide fine details for the fit.

about 0. In addition, the algorithm converged on the second iteration in each case. Similar to our previous paper, the selection of the number of Gaussians was determined by a combination of performance criteria (total cost at the end of each session), and minimization of Bellman Error [10].

### B. Comparing human and control system behavior

Fiugre 2 (a) shows the 2d hand trajectory averaged over all subjects for Experiment 1. Here, only trials in the second half of each block was included when the movement became stable and we assume that subjects have acquired the pattern of target jump. As we can see, initial lateral movement started right from the onset of movement, followed by a bigger lateral movement towards the target location after jumps. Interestingly, early lateral movement, presummably triggered by the estimation of end target location, is different for the two distriubtions even their mean is the same. More spicifically, when target either stayed in the middle or jumped to the right as in the two-peaks block, the initial movement was shooting roughly tarwards 5cm to the right, which is the mean of the jump distribution. However, when there was a small chance for the target to jump to the left as in the three-peaks block, subjects tended to not move as much as in the two-peaks condition. Intuitively, such stragagy is safe since a big divergence from the middle may cause an undershoot if target jumped to the opposite direction from estimation. Figure 2 (b) shows the prediction by the MDP model. Comparing with human movement, we can see that the model predicts the result for the two-peaks condition very well, but not the three-peaks condition. For the later distribution, rather than starying close to the middle, the model predicts a bigger initial movement to the right. A close look of how the value function evolves over time reveals the reason: since the probability of jumping to the right is much higher (0.7) than the other two directions (0.2 to the left, and 0.1 in the middle), the deepest part of the value function occured far to the right at jump time. If the only goal was to achieve the smallest value function at each time step, the controller would move further away from the center.

Figure 2 (c) shows the performance of the FAS without the uncertainty mixing terms. What is clear is that the prediction term affects the ability of the controller to reach the desired target. The issue is that the mean prediction is not at any of the points, and so a control which blends the predicted and measured points only by adding the associated trajectories together will not reach the target. Thus it is clear that a mixing term is necessary, and as we can see in figure 2 (d), the target can be reached more closely, while still anticipating the jump location and reacting before the jump occurs.

Figure 2 (e) shows the mean 2d trajectory for Experiment 2. Again, only the second half of the experiment was included when the learning effect became stable. Consistent with Experiment 1, subjects moved to the estimated direction even before target jumped when the jump happened late during movement. However, when such perturbation occured early, initial hand trajectory became straight. To understand whether this is due to the fact that people didn't learn at all in the early block, another experiment was done where subjects were told explicitly that the target had a much higher probability to go to the right. Yet still, they wouldn't use this knowledge in the movement, but rather fully depend on their visual feedback of target jump. Intuitively, if tracking the target is good enough to do the task as in early jump, there is no need to use estiamtion, which is not as reliable as the visual feedback of target jump. Therefore, the dependency of estimation may vary over different jump time, or the availability of the more reliable sources. Such effect can't be accounted for by the MDP (results not shown here).

## REFERENCES

[1] B. Anderson and J. Moore. *Optimal Filtering*. Prentice Hall, 1979.

[2] D. Bertsekas. *Dynamic programming and optimal control*. Athena Scientific, Bellmont, MA, 2nd edition, 2001.

[3] L. Collatz. *The Numerical Treatment of Differential Equations*. Springer-Verlag, 1966.

[4] D. Pelisson E. Komilis and C. Prablanc. Error processing in pointing at randomly feedback-induced double-step simuli. *Journal of Motor Behavior*, 25:299–308, 1993.

[5] K. P. Kording and D. M. Wolpert. The loss function of sensorimotor learning. *Proceedings of the National Academy of Sciences*, 101:9839–42, 2004.

[6] H. Kushner and P. Dupuis. *Numerical materials and methods for stochastic control problems in continuous time*. Springer-Verlag, New York, 2nd edition, 2001.

[7] D. Liu and E. Todorov. Evidence for the flexible sensorimotor strategies predicted by optimal feedback control. *Journal of Neuroscience*, 27(35):9354–9368, 2007.

[8] L. Ljung. *System Identification: Theory for the User*. Prentice Hall, PTR, Upper Saddle River, NJ, 2nd edition edition, 1999.

[9] J. Si, A. Barto, W. Powell, and D. Wunsch, editors. *Handbook of Learning and Approximate Dynamic Programming*. IEEE Press on Computational Intelligence, 2004.

[10] A. Simpkins and E. Todorov. Optimal tradeoff between exploration and exploitation. American Control Conference, IEEE Computer Society, 2008.

[11] H. W. Sorenson, editor. *Kalman Filtering: Theory and Application*. IEEE Press, 1985.

[12] R. Stengel. *Stochastic Optimal Control: Theory and Application*. John Wiley and Sons, 1986.

[13] E. Todorov. Optimality principles in sensorimotor control. *Nature Neuroscience*, 7:907–915, 2004.

[14] D. M. Wolpert, Z. Ghahramani, and M. Jordan. An internal forward model for sensorimotor integration. *Science*, (269):1880–1882, 1995.