

State-Space Decoding of Goal-Directed Movements

Deriving a nonrecursive computationally efficient method

Bayesian inference methods hold great promise for the prediction of hand-movement trajectories in neural prosthetic devices. The accuracy of such probabilistic methods can be improved by incorporating meaningful priors, thereby appropriately constraining the space of possible states that the system can attain. In this work we review and extend methods for constructing reach trajectories that incorporate prior information of the intended movement target. For computational tractability, we model arm motion as a linear dynamical system driven by Gaussian noise, conditioned on this end-point information. These assumptions, while biomechanically unrealistic, give rise to a priori model arm-paths that share many of the characteristics of natural arm trajectories. Moreover, in this model formulation we may compute the predicted arm position, given simultaneously observed neural data, using standard

Digital Object Identifier 10.1109/MSP.2007.909001

forward-backward computations familiar from the theory of the Kalman filter. Here we review an earlier recursive approach for computing such reach trajectories and present a new nonrecursive approach, with computations that may be performed analytically for the most part, leading to a significant gain in the accuracy of the inferred trajectory while imposing a very small computational burden. Finally, we discuss extensions of our approach, including the incorporation of multiple target observations at different times, and multiple possible target locations.

INTRODUCTION

In recent times there has been a significant effort, aided by the development of multielectrode arrays and sophisticated decoding algorithms, to design neural prostheses for patients with motor deficits [1], [2]. To derive the maximum clinical impact, research in the development of neural prostheses has focused on reaching movements, primarily because most day-to-day tasks are of this nature and also because it is a convenient framework in which a person in a locked-in state can interact with the world through a computer interface by choosing one of several possible options [3].

A number of studies have shown that firing rates of cortical neurons measured during reaching movements encode position [4], direction [5], velocity [6], acceleration [7], and intended reach goal [8], among other parameters of interest. This information encoded in the neural firing rates may be decoded to reconstruct the intended hand trajectory on a given behavioral trial. Algorithms that have been used include population vectors [5], optimal linear filters [3], [9], and Bayesian decoders [10]–[12]. Of these methods, the Bayesian approach is perhaps the most promising, since it allows for a principled incorporation of different sources of signal and noise, and prior information, in the decoding system. The standard Kalman filter is the most popular instance of a Bayesian decoder, with the well-known recursive nature of this filter potentially allowing for real-time implementation [13].

It has been recently emphasized that a key problem in the construction of an efficient decoder is in suitably constraining the hand trajectories. In Bayesian language, the challenge is to construct priors on hand trajectories which are accurate and computationally tractable. Accuracy in a probabilistic setting implies that samples drawn from this prior should resemble natural hand paths. Proper specification of these priors will clearly lead to improvements in the accuracy of Bayesian decoders. Such prior information includes knowledge about the smoothness of the hand path [14], [15] and also about the start and end state of the hand; for example, in a goal-directed reaching task the hand velocity at the beginning and end of the trial should be zero, and we may know the target location with some uncertainty [16], [17].

We will focus on the problem of incorporating this prior end-point information in a computationally efficient manner. We develop a model of the hand trajectory as a linear dynamical system driven by Gaussian noise conditioned on this end-point data and describe how to apply standard Kalman tools in this setting. We begin by reviewing techniques from the Kalman filter literature which may be used to incorporate this end-point informa-

tion optimally and briefly present the work in [17], where the authors develop a recursive algorithm to compute the conditioned trajectories. Subsequently, we present our approach, demonstrating that the necessary forward-backward computations of the Kalman filter in this model may be performed nonrecursively and, for the most part, analytically. This approach is more efficient than the recursive approach when the conditioned trajectories are to be computed at only a subset of times, a scenario that is likely when the neural signal is being sampled at a high-rate making the real-time implementation of the conditioned trajectories at all points in time computationally impractical. More importantly we show that our formulation lends itself easily to further extensions including to two clinically important problems that we consider: the moving target paradigm, in which the subject is required to reach multiple targets at different times, and the multiple choice paradigm, in which the subject reaches one of several targets at a given time. Throughout this article we will be neutral about the precise details of the neural observations available; the techniques developed here may be applied without great modification either to multielectrode single-unit [3], [18], local field potential [19], or electrocorticographic [20] recordings, or to any combination thereof.

THEORY

MODELING HAND DYNAMICS

We begin by specifying our model for the system dynamics. The state of the system at time t will be denoted as q_t . We will, for notational convenience, burden the interpretation of t to denote both running continuous time as well as a discrete time index; the interpretation implied should be readily clear from the context. The vector q_t may include various types of information about the state of the hand at time t : the hand position, the hand velocity, and higher time-derivatives like acceleration and jerk. To be concrete, for movement in a two-dimensional plane we may take the state vector q_t to be $q_t = [s_x(t), v_x(t), s_y(t), v_y(t)]'$, where $s(t)$ denotes the hand position at time t , $v(t)$ velocity, and x and y denote the horizontal and vertical axes, respectively.

We assume the following simple linear Gaussian dynamics for q_t :

$$\dot{q}_t = Rq_t + \rho + W_t, \quad (1)$$

where ρ is a constant-drift term, R is a fixed dynamics matrix, and W_t is zero-mean, temporally independent Gaussian noise with covariance matrix $E[W_s W_t'] = Q\delta(t-s)$ for some fixed matrix Q . This formulation of the arm dynamics implies q_t is a Gauss-Markov process and we have

$$p(q_t|q_{t-1}, q_{t-2}, \dots, q_0) = p(q_t|q_{t-1}) = G_{(\mu, \Sigma)}(q_t),$$

where we have introduced the notation $G_{(\mu, \Sigma)}(x)$ to denote that the random variable x has a Gaussian distribution whose mean is given by μ and covariance by Σ . For dynamical system given by (1)

$$\begin{aligned}\mu &= \Phi(t, t-1)q_{t-1} + \int_{t-1}^t \Phi(t, \tau)\rho d\tau \\ \Sigma &= \int_{t-1}^t \Phi(t, \tau)Q\Phi(t, \tau)'d\tau,\end{aligned}$$

where $\Phi(t, \tau) = \exp((t - \tau)R)$. We shall derive these terms later, but note for now that μ depends on the model parameters and the last state of the system, q_{t-1} , while Σ depends only on the model parameters. It is important to realize that this model formulation is an a priori assumption, which is quite classical in the control and estimation literature, primarily due to its analytical tractability. It is worth discussing the features and weaknesses of this model specifically in the neural prosthetic context.

We first discuss the individual terms of (1) in more depth. The deterministic terms R and ρ define the system's behavior in the absence of noise (i.e., as $Q \rightarrow 0$). The dynamics matrix R specifies how the different elements of the state vector q_t interact. For example, for the position-plus-velocity q_t specified above, a reasonable R might be of block diagonal form

$$R = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix};$$

this form specifies that the horizontal and vertical positions do not interact with each other and that the velocity signal is unaffected by the current position. Of course, by modifying this matrix it is easy to introduce dependencies between these variables and model, for example, anisotropic inertial effects due to the biomechanical properties of the arm. The constant drift term ρ , on the other hand, fixes the set point of the dynamics: assuming stable dynamics (i.e., the real parts of all the eigenvalues of R are less than zero), the system will evolve towards the fixed point $q_\infty = R^{-1}\rho$. Finally, the covariance matrix Q , which models neuromuscular noise, specifies the variability of the hand paths; large values of Q lead to hand paths whose position or velocity might vary quite noisily as a function of time, while smaller Q leads to smoother hand paths. We also note in passing that other interpretations of Q are possible; for example, it can reflect our uncertainty as external observers about the intended trajectory as time progresses [17].

Clearly, this linear-Gaussian model is at best an approximation. For example, if we include joint angle information in our state vector, where the elbow angle may take values only between 0 and π radians, the dynamics would be constrained and nonlinear. We will return to this issue later; for now, however, it is enough to note that this model may be considered a good *local* model, enforcing constraints such as the smoothness of the hand path on a local scale (i.e., for small time steps dt in $p(q_{t+1}|q_t)$).

REACHING FOR A SINGLE TARGET

In this section we describe how to use this dynamics model to efficiently incorporate the observed neural data, initial state, and end-point information to predict the hand state q_t at time t . We denote the observed neural data at time t by O_t and the entire set of observations over the interval $[0, t]$ by $\{O_t\}$. We first consider the case when we have a single end-point observation y_T where T is the duration of motion and is assumed to be known. We further assume that the observation y_T is some noise-contaminated version of the final hand state q_T , that is,

$$y_T = Kq_T + \nu, \quad (2)$$

where $\nu \sim \mathcal{G}_{(0, M)}(\nu)$. For now we assume that the Gaussian noise ν is independent of q_T (many of these assumptions may be relaxed somewhat). The observation matrix K is not assumed to be invertible, implying that we may have incomplete information of the reach goal. For example, if we know that the hand velocity must be zero at the end of the trial, but we don't know the end position, K would be of the form

$$K = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

with y_T being a two-dimensional zero vector. The noise ν here allows some error in this final velocity; that is, this final velocity might not always be exactly zero. Similarly, we may incorporate information about the final position by letting K be of full rank (with the noise covariance M chosen to emulate the size of the position and velocity errors observed during the reaching task). An identical model formulation was described in [17].

Now our goal is to compute $p(q_t|q_0, \{O_t\}, y_T)$. Here we make the key assumption that the neural observations $\{O_t\}$ depend on the state variable q_t in a simple Markov manner, that is,

$$p(O_t|q_t, q_{t-1}, \dots, q_0, q_{t+1}, q_T) = p(O_t|q_t).$$

Thus the neural data O_t only depends on the current hand state q_t . We again note in passing that this model is only approximately correct; for example, it is known that many neurons in the parietal and prefrontal cortex preferentially encode the *intended* hand position than the current position or velocity [19], [21]. We discuss methods for incorporating this *intentional* information below. Another situation where this assumption is violated is the history-dependent firing rate observed in the primary motor cortex [15], though this can be readily remedied by augmenting the state-vector to account for such history dependence.

Before we present our approach we first briefly review an earlier work [17] that considered the same problem of reaching a single fixed target and developed a recursive solution to obtain the target-constrained hand trajectories. The key *backwards* recursion that they used may be derived by expanding

$$\begin{aligned}
p(q_t|q_0, y_T, \{O_t\}) &= (1/Z)p(q_t, y_T, \{O_t\}|q_0) \\
&= (1/Z) \int \dots \int p(q_t, y_T, \{q_i\}, \\
&\quad \{O_t\}|q_0) d\{q_i\} \\
&= (1/Z) \int \dots \int p(q_t, \{O_t\}|q_0) p(q_{t+1}|q_t) \\
&\quad \times p(q_{t+2}|q_{t+1}) \dots p(q_T|q_{T-1}) \\
&\quad \times p(y_T|q_T) d\{q_i\} \\
&= (1/Z) p(q_t, \{O_t\}|q_0) \int p(q_{t+1}|q_t) \\
&\quad \times dq_{t+1} \dots \int p(q_T|q_{T-1}) dq_{T-1} \\
&\quad \times \int p(q_T|q_{T-1}) p(y_T|q_T) dq_T,
\end{aligned}$$

where Z is a normalization factor and the equalities follow using Bayes' rule and the Markov nature of the model.

We will subsequently show that we can compute the last integral on the right, corresponding to time T , using Gaussian integration formulas. And in particular we show that, $p(y_T|q_t)$ is an unnormalized Gaussian for all times t , and integrating $p(y_T|q_t)$ against a Gaussian term of the form $p(q_t|q_{t-1})$ preserves this Gaussianity. So we may simply recurse backward, from this last integral on the right, computing $p(y_T|q_u)$ for all times $t \leq u \leq T$, starting from T and moving backwards: $u = \{T-1, T-2, \dots, t\}$, until we obtain the desired $p(y_T|q_t)$ term. This is the standard backwards recursion, familiar from the forward-backward Baum-Welch algorithm in hidden Markov models [22]. In the language of hidden Markov models, the term $p(q_t, \{O_t\}|q_0)$ is the usual forward probability, while $\int p(y_T|q_T)p(q_T|q_t)dq_T$ is the backwards probability.

This recursive approach is much slower than the approach we develop if we only need to compute $p(q_t|q_0, y_T, \{O_t\})$ at a single time t , or more generally at a small subset of times. Such a scenario is likely when the neural observations are being sampled at a high rate, and the computations required to construct the reach-target constrained trajectories at all instants of time cannot be carried out in real time. Also, our approach allows us to efficiently incorporate multiple observations of a moving target. Finally, as we will describe below, our approach will allow us to

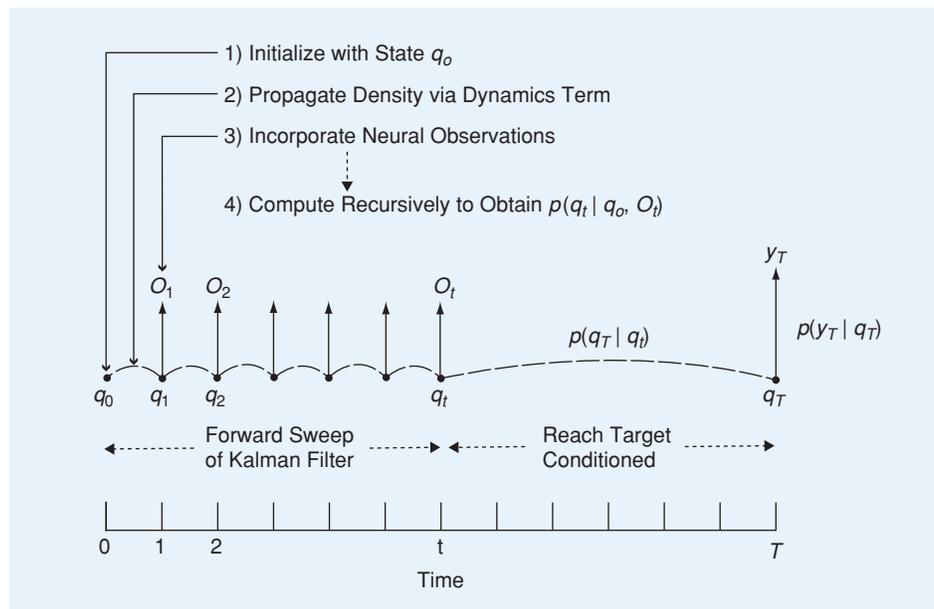
address the important situation when there are multiple reach targets with little increase in computational effort.

We are now in a position to develop our approach which circumvents the need for such a recursive approach we have that

$$\begin{aligned}
p(q_t|q_0, y_T, \{O_t\}) &= \frac{1}{Z} p(q_t, \{O_t\}, y_T|q_0) \\
&= (1/Z) p(q_t, \{O_t\}|q_0) p(y_T|q_t, q_0, \{O_t\}) \\
&= (1/Z) p(q_t, \{O_t\}|q_0) p(y_T|q_t) \\
&= (1/Z) p(q_t, \{O_t\}|q_0) \int p(y_T, q_T|q_t) dq_T \\
&= (1/Z) p(q_t, \{O_t\}|q_0) \\
&\quad \times \int p(y_T|q_T) p(q_T|q_t) dq_T,
\end{aligned} \tag{3}$$

where Z is a normalizing constant. The key fact now is that all of the terms in (3) may be computed quite efficiently. As mentioned earlier, $p(q_t, \{O_t\}|q_0)$ is the forward probability, while $\int p(y_T|q_T)p(q_T|q_t)dq_T$ is the backwards probability. In subsequent discussion we will refer to $p(q_t|q_0, y_T, \{O_t\})$ at the target-conditioned estimate, while we will refer to $p(q_t, \{O_t\}|q_0)$ as the forward-filter estimate (Figure 1).

We will not discuss computation of the forward filter term in depth because a number of good treatments are already available in the literature. In particular, in the case that the observed neural data may be treated as Gaussian given q_t , the computation of $p(q_t, \{O_t\}|q_0)$ corresponds exactly to the forward sweep of the



[FIG1] Schematic of algorithm. This figure demonstrates the implementation of the algorithm to compute the term $p(q_t|q_0, y_T, \{O_t\})$; that is, to compute at time t the target-conditioned trajectories. The algorithm is initialized with q_0 and this state is propagated forward via the stochastic differential equation (1) to the next time step. Neural observations O_1 are then incorporated into our estimate of the state to obtain $p(q_1|q_0, O_1)$. This process, known as the forward Kalman filter, is carried out recursively over the interval $[0, t]$ to obtain at t , $p(q_t|q_0, \{O_t\})$. Then the terms from (3) corresponding to the fact that we are conditioning on the reach-target (i.e., $p(y_T|q_t)$) are computed, and finally, the target-conditioned distribution, $p(q_t|q_0, y_T, \{O_t\})$, is obtained.

Kalman filter [13], [23], [24]. When we have point-process observations O_t on the Gauss-Markov process, we may employ approximate point-process filters, such as those that apply Gaussian approximations about the posterior density mean [14], or the prediction density mean [11]. Particle filters may be applied for more general representations of uncertainty [10], [25].

The remaining term $\int p(y_T|q_T)p(q_T|q_t)dq_T$ turns out to be similarly easy to deal with. Discrete-time recursions for this backwards probability (starting with $p(y_T|q_T)$, which we know explicitly, from (2), and recursing backwards for $t = T-1, T-2, \dots$) are well known; see [22] for the basic theory or [17] for an implementation in the Gaussian model under consideration here. Our insight here is that this term may be computed explicitly, with no need for a time-consuming recursion.

Both terms under the integral in (3) are Gaussian, due to the linear Gaussian nature of the model, and so computing the conditional distribution $p(y_T|q_t)$ just requires some simple Gaussian manipulations. Let's start by evaluating the mean and covariance $E[q_t|q_{t_0}]$ and $\text{Cov}[q_t|q_{t_0}]$, for some arbitrary time $t > t_0$. By applying linearity of expectation to our model (1), we have that

$$\frac{d}{dt}E[q_t|q_{t_0}] = \rho + RE[q_t|q_{t_0}].$$

This is a linear ordinary differential equation with initial condition $E[q_{t_0}|q_{t_0}] = q_{t_0}$, which we may solve explicitly to obtain

$$\begin{aligned} E[q_t|q_{t_0}] &= e^{(t-t_0)R}q_{t_0} + \int_{t_0}^t e^{(t-\tau)R}\rho d\tau \\ &= e^{(t-t_0)R}q_{t_0} + R^{-1}(I - e^{(t-t_0)R})\rho. \end{aligned}$$

Similarly, to compute the covariance, we have

$$\begin{aligned} \text{Cov}[q_t|q_{t_0}] &= E[q_t q_t'|q_{t_0}] - E[q_t|q_{t_0}]E[q_t|q_{t_0}]' \\ &= E \left[\left(\int_{t_0}^t \exp((t-u)R)W(u)du \right) \right. \\ &\quad \left. \times \left(\int_{t_0}^t \exp((t-v)R)W(v)dv \right)' \right] \\ &= \exp(tR) \left[\int_{t_0}^t \int_{t_0}^t \exp(-uR) \right. \\ &\quad \left. \times E[W(u)W(v)'] \exp(-vR)' du dv \right] \exp(tR)' \\ &= \exp(tR) \left[\int_{t_0}^t \exp(-uR)Q \right. \\ &\quad \left. \times \exp(-uR)' du \right] \exp(tR)' \\ &= GV(t-t_0)G' \end{aligned}$$

where we have used the fact that $E[W_t W_{t_0}'] = Q\delta(t-t_0)$ and we define

$$V(t)_{i,j} = \frac{(G^{-1}QG'^{-1})_{i,j}}{d_i + d_j} (\exp[(d_i + d_j)t] - 1).$$

We have assumed that $R = GDG^{-1}$ can be diagonalized, and d_i refers to the i th diagonal term of D . (In the case that the dynamics matrix R is not diagonalizable, we instead make use of the Jordan form, $R = SJS^{-1}$, where J is a simple block-diagonal matrix [26], [27]. The resulting expression for $\text{Cov}(q_t|q_{t_0})$ is more complicated but not significantly more computationally expensive than the form we have presented here in the diagonalizable case.)

We now use these expressions to compute the integral in (3). Let $\Phi_t = \exp(tR)$, and $C_{(t-t_0)} = \text{Cov}[q_t|q_{t_0}]$. Using the dynamics equation (1) we have $p(q_t|q_0) = \mathcal{G}_{(\Phi_t, q_0, C_t)}(q_t)$, $p(q_T|q_t) = \mathcal{G}_{(\Phi_{(T-t)}, q_t, C_{(T-t)})}(q_T)$ and $p(y_T|q_T) = \mathcal{G}_{(Kq_T, M)}(y_T)$.

We then have

$$\begin{aligned} \int p(y_T|q_T)p(q_T|q_t)dq_T &= \int \mathcal{G}_{(Kq_T, M)}(y_T) \\ &\quad \times \mathcal{G}_{(\Phi_{(T-t)}, q_t, C_{(T-t)})}(q_T) \\ &= w \exp \left\{ -\frac{1}{2}(q_t - m)'A(q_t - m) \right\}, \end{aligned}$$

where w is a normalizing constant and

$$\begin{aligned} A &= \left(\Phi'_{(T-t)} C_{(T-t)}^{-1} \Phi_{(T-t)} - \Phi'_{(T-t)} C_{(T-t)}^{-1} \hat{C} C_{(T-t)}^{-1} \Phi_{(T-t)} \right), \\ m &= A \left(\Phi'_{(T-t)} C_{(T-t)}^{-1} \hat{C} K' M^{-1} y_T \right), \\ \hat{C} &= \left(C_{(T-t)}^{-1} + K' M^{-1} K \right)^{-1}. \end{aligned}$$

If A is of full rank then this is a nonnormalized Gaussian in q_t with mean given by m and covariance by A^{-1} . More generally, q_t will be a Gaussian with infinite variance in some directions.

Now to complete our expansion of (3), we need to supply an expression for the forward probabilities $p(q_t, \{O_t\}|q_0)$. As emphasized earlier, these forward probabilities may be computed via a number of different algorithms, depending on the observed neural data. Common ways to compute these probabilities are via a Kalman filter [13] or approximate point-process filter [14], [15]. In this case, the forward probabilities are approximated as a weighted Gaussian form $p(q_t, \{O_t\}|q_0) \approx w_t^f \mathcal{G}_{(q_t^f, C_t^f)}(q_t)$, where w_t^f is a normalization factor. We have

$$\begin{aligned}
p(q_t | q_0, q_T, \{O_t\}) &= \frac{1}{p(\{O_t\}, y_T | q_0)} p(q_t, \{O_t\} | q_0) \\
&\times \int p(y_T | q_T) p(q_T | q_t) dq_T \\
&\approx \frac{1}{p(\{O_t\}, y_T | q_0)} w_t^f \mathcal{G}_{(q_t^f | C_t^f)}(q_t) \\
&\times \exp \left\{ -\frac{1}{2} (q_t - m)^T A (q_t - m) \right\} \\
&= \frac{1}{Z} G_{q_t^s, C_t^s}(q_t),
\end{aligned}$$

where Z is a normalizing constant and

$$\begin{aligned}
C_t^s &= \left[C_t^{f-1} + \Phi'_{(T-t)} C_{(T-t)}^{-1} \Phi_{(T-t)} \right. \\
&\quad \left. - \Phi'_{(T-t)} C_{(T-t)}^{-1} \hat{C} C_{(T-t)}^{-1} \Phi_{(T-t)} \right]^{-1}, \quad (4)
\end{aligned}$$

$$q_t^s = C_t^s \left[C_t^{f-1} q_t^f + \Phi'_{(T-t)} C_{(T-t)}^{-1} \hat{C} K' M^{-1} y_T \right]. \quad (5)$$

The superscript s here implies the smoothed estimate, given all available information. We have thus obtained, in closed form, the estimates for q_t given the initial state q_0 , the forward probabilities $p(q_t, \{O_t\} | q_0)$, and the end-point information y_T . Computing these quantities requires just some simple low-dimensional linear-algebraical manipulations, and therefore this decoding algorithm may be tractably implemented in real time; note in particular that many of the quantities that appear in the above equations may either be precomputed (e.g., G, D , and $K' M^{-1}$) or computed just once per time-step and *recycled* for efficiency (e.g., the term $C_{(T-t)}^{-1} \Phi_{(T-t)}$).

INCORPORATING MULTIPLE OBSERVATIONS: TRACKING A MOVING TARGET

Now we are ready to develop an important generalization of the above single-observation case. Imagine that we have a sequence of target observations y_{T_i} to be reached at times T_i for $\{i = 1, 2, \dots, N\}$, with $T_1 < T_2 < \dots < T_N$, or y_{T_i} could correspond to observations of a single moving target whose state changes as a function of time. Performing inference in this setting may be done by a version of the standard backwards recursion described above; in particular, we may use our analytical solution to work our way back from the observation at time T_N to the current time t . We have

$$\begin{aligned}
p(q_t | q_0, \{y_{T_i}\}) &= (1/Z) p(q_t, \{y_{T_i}\} | q_0) \\
&= (1/Z) \int \dots \int p(q_t, \{y_{T_i}\}, \{q_{T_i}\} | q_0) d\{q_{T_i}\} \\
&= (1/Z) \int \dots \int p(q_t | q_0) p(q_{T_1} | q_t) p(y_{T_1} | q_{T_1}) \\
&\quad \times \prod_{i=2}^N p(q_{T_i} | q_{T_{i-1}}) p(y_{T_i} | q_{T_i}) d\{q_{T_i}\} \\
&= (1/Z) p(q_t | q_0) \int p(q_{T_1} | q_t) p(y_{T_1} | q_{T_1}) dq_{T_1} \\
&\quad \times \int p(q_{T_2} | q_{T_1}) p(y_{T_2} | q_{T_2}) dq_{T_2} \dots \\
&\quad \dots \int p(q_{T_{N-1}} | q_{T_{N-2}}) p(y_{T_{N-1}} | q_{T_{N-1}}) dq_{T_{N-1}} \\
&\quad \times \int p(q_{T_N} | q_{T_{N-1}}) p(y_{T_N} | q_{T_N}) dq_{T_N}.
\end{aligned}$$

So, as before, we apply our analytic solution to compute each of the above integrals, in the sequence $\{T_N, T_{N-1}, T_{N-2}, \dots, T_1, t\}$. Note that this solution is typically much more efficient than the full backwards recursion, since we only perform computations at the subset of observation times T_i , instead of at all times $t < u < T_N^2$. (See [17] for a different approach to this problem.)

MULTIPLE TARGETS

In this section we consider the multiple target setting, which is of key importance in the neural prosthetic context [3], [28], [29]. Let the random variable X denote one of the possible K targets, with the probability that reach to target denoted by X_k is given by $p(X = X_k)$. We assume that our prior information about these targets is given by a Gaussian distribution where $p(q_T | X = X_k) \sim \mathcal{G}_{(\mu_k, \Pi)}(q_T)$. Note that this is a special instance of the case considered earlier where we did not assume that the variance in our prior information of the state vector was finite for all its elements (K was allowed to be noninvertible). Though the more general case can be considered, in this section, we stick to the simpler case to keep the formulas derived straightforward while still demonstrating the flexibility of our modeling approach.

Now to compute the marginal distribution $p(q_t | q_0)$ we average over all possible end states

$$\begin{aligned}
p(q_t | q_0) &= \sum_k p(X = X_k) \int p(q_T | X = X_k) \\
&\quad \times p(q_t | q_0, q_T, X = X_k) dq_T.
\end{aligned}$$

The integral can be computed using the same methods developed earlier, and we are left with a mixture-of-Gaussians model

$$p(q_t | q_o) = \frac{1}{\sum_k w_{k,t}^s} \left(\sum_k p(X = X_k) w_{k,t}^s \mathcal{G}_{(q_{k,t}^s, C_t^s)}(q_t) \right)$$

where

$$A_t = \left(\Pi^{-1} + C_{(T-t)}^{-1} \right)^{-1},$$

$$B_t = \left(C_t^{f-1} + \Phi'_{(T-t)} C_{(T-t)}^{-1} \Phi_{(T-t)} - \Phi'_{(T-t)} C_{(T-t)}^{-1} A_t C_{(T-t)}^{-1} \Phi_{(T-t)} \right)^{-1}.$$

with the smoothed covariance C_t^s defined as above (4)

$$q_{k,t}^s = C_t^s \left[C_t^{f-1} q_t^f + \Phi'_{(T-t)} C_{(T-t)}^{-1} \right. \\ \left. \times \left(\Pi^{-1} + C_{(T-t)} \right)^{-1} \Pi^{-1} \mu_k \right],$$

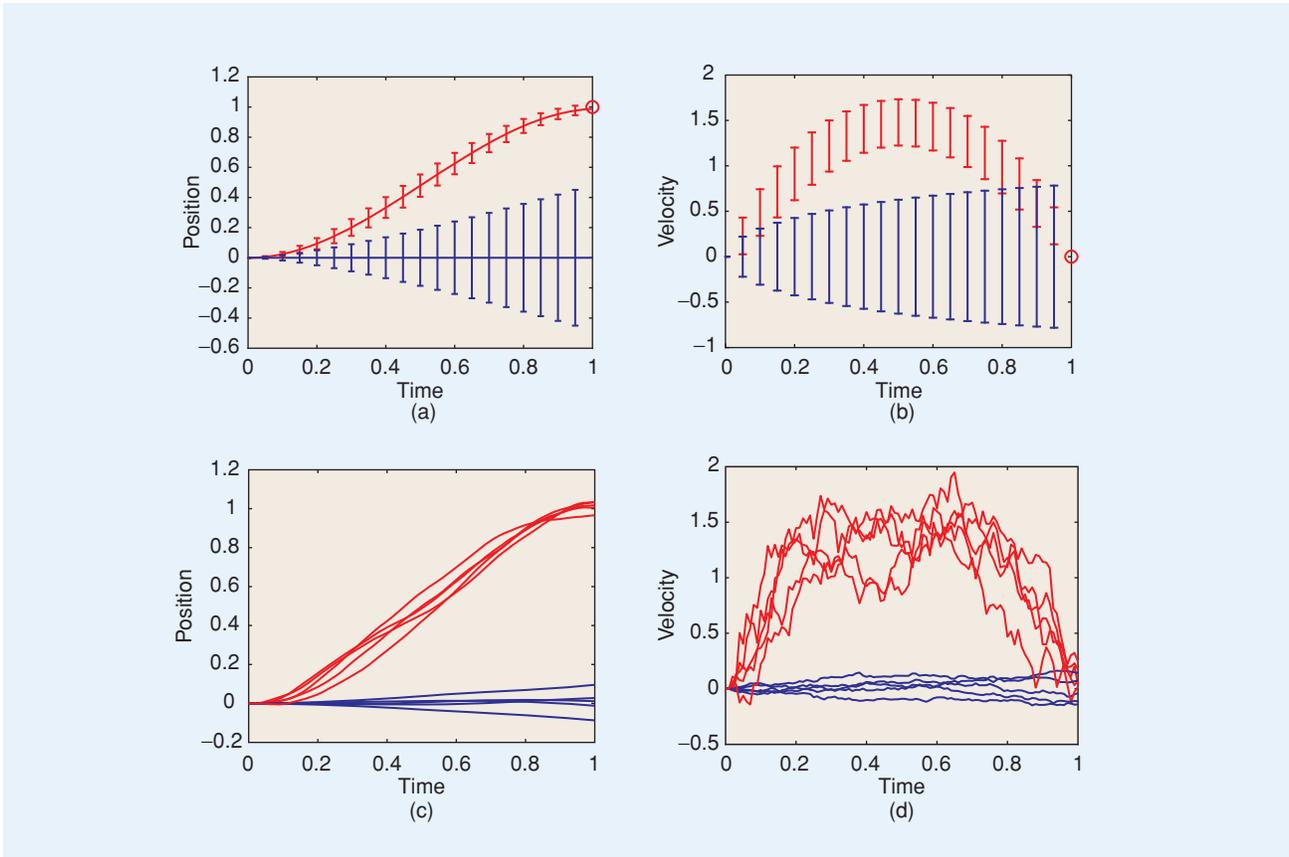
and

$$w_{k,t}^s = \exp \left[-\frac{1}{2} \mu_k' \left(\Pi^{-1} - \Pi^{-1} A_t^{-1} \Pi^{-1} - \Pi^{-1} A_t^{-1} C_{(T-t)}^{-1} \right. \right. \\ \left. \left. \times \Phi'_{(T-t)} B_t \Phi_{(T-t)} B_t \Phi_{(T-t)} C_{(T-t)}^{-1} A_t^{-1} \Pi^{-1} \right) \mu_k \right. \\ \left. + \mu_k' \Pi^{-1} A_t C_{(T-t)}^{-1} \Phi_{(T-t)} B_t C_t^{f-1} q_t^f \right],$$

Thus, our analytic solution allows for efficient implementation of this more complex and realistic setting. Extensions to mixtures of target times T (not just target locations q_T) are also straightforward, although we lack the space to pursue these applications here.

RESULTS

In this section we present some simulation results to demonstrate the utility of our approach. We start with the simplest



[FIG2] Reach trajectory conditioned on terminal state without neural observations. In these figures we depict a simulation in which the state vector q_t is two dimensional, capturing the position and velocity of the hand, starting from the origin, and reaching for a noisy observation of the terminal point depicted by the red 'o' sign. In (a) the thick red line depicts the expected value of the position conditioned on the target, while the thick blue line depicts its expected value in the absence of any end-point information. The standard deviation is also shown at a few points along the trajectories. Similar plots for $v_x(t)$ are shown in (b). Plots (c) and (d) show sample paths drawn from the target-conditioned and the forward filter distributions. The values of the various parameters are presented in the accompanying text. The key observation is that the target-conditioned trajectory has a very natural bell-shaped profile, as seen in the plot of velocity against time (b), while the unconditioned paths wander aimlessly, with unrealistic velocity profiles.

case, in which the reach movement is constrained to be along the horizontal (x) axis. We consider the case where no neural observations have been made, that is we want to compute $p(q_t | q_0, y_T)$. Figure 2 shows the mean target-conditioned trajectory $E(q_t | q_0, y_T)$, along with the standard deviation. For our simulations, we consider a state vector $q_t = [s_x(t), v_x(t)]'$, with

$$R = \begin{bmatrix} 0 & 1 \\ -0.01 & -0.5 \end{bmatrix}.$$

The -0.5 term here corresponds to frictional forces (which are proportional to the velocity), while the -0.01 term models a weak restorative (spring-like) force towards the origin. (None of the results presented here depend strongly on these friction and spring tension coefficients.) The noise driving the system enters through Q , which is taken to be

$$Q = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

This models the realistic scenario that while acceleration can be discontinuous, the velocity and position of the hand will have continuous trajectories. We choose the reach target to be $q_T = [1 \ 0]'$, specifying that we would like to reach the target at position $s_x(T) = 1$ with zero velocity. The observation matrix K is taken to be the identity matrix, while the observation noise covariance M is a diagonal matrix, $M = 10^{-3} \cdot \mathcal{I}_2$, indicating that uncertainty in the knowledge of $s_x(T)$ is not correlated with $v_x(T)$, and that both uncertainties are fairly modest. Finally, the duration of the reach motion is taken to be $T = 1$.

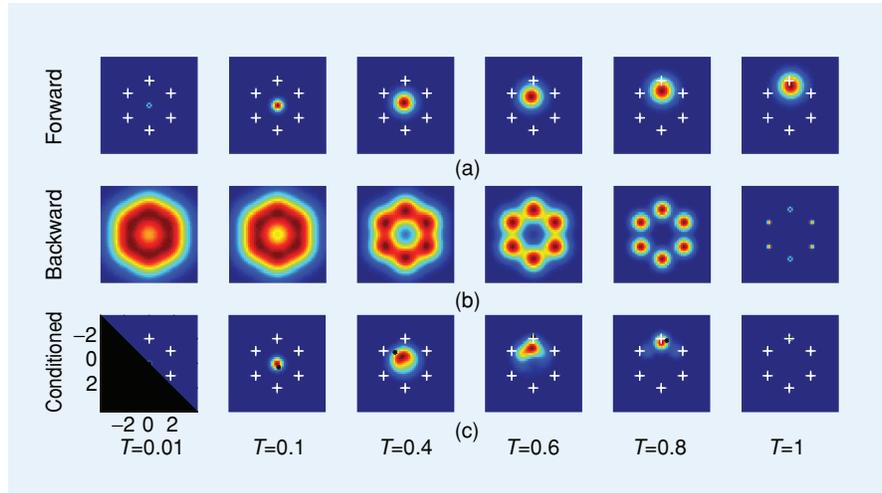
The first thing to note from the Figure 2 is that the target-conditioned trajectories have a natural profile, as exemplified in Figure 2(b) by the bell-shaped form of the expected velocity $E(v_x(t) | q_0, q_T)$. We also note that the uncertainty in our estimate is the least at the terminal points and is greatest in the middle of the motion (since we have conditioned on both the start and end points of the trajectory). In the absence of target information (or more generally, when the target uncertainty becomes very large), the trajectory wanders about unsystematically, as shown by the blue lines in Figure 2(c) and (d). Thus, while the basic linear Gaussian model (1) does not produce realistic priors, conditioning on the target produces trajectories which are similar to nat-

ural reaching trajectories. See [17] for further discussion and examples.

In the next set of simulations (Figure 3) we consider the mixture-model setting described earlier, where reach is to be made to one of six targets. The state vector is taken to be four dimensional, $q_t = [s_x(t), v_x(t), s_y(t), v_y(t)]'$. The targets are situated symmetrically in the x, y plane at a radial distance of two units, and the velocity components of all the targets are zero. For the simulation one of the six targets is chosen at random. Following our notation

$$R = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -0.01 & -0.5 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -0.01 & -0.5 \end{bmatrix}, \quad \Pi = 0.01 \cdot \mathcal{I}_4$$

and $W = \mathcal{I}_4$. To simulate the neural data used in the forward-filter, $p(q_t | q_0, \mu_k)$ is first computed following our results to the chosen target with mean μ_k for each t . The mean of this distribution is taken to be the true trajectory, and noisy Gaussian observations of this true trajectory are used as neural data in the forward filter. The observational noise is assumed to be univariate Gaussian with zero mean and covariance of $20 \cdot \mathcal{I}_4$. Finally, the duration of the motion is taken to be $T = 1$.



[FIG3] Sample simulation for reach trajectory conditioned on multiple targets with neural observations. In this figure we consider reaching motion towards one of several terminal states which are known a priori. The figure shows the x, y plane of (a) the forward distribution $p(q_t | \{O_t\}, q_0)$ conditioned on the initial state and neural observations up to that point, (b) the backward distribution $p(q_t | q_t)$, and (c) the reach-conditioned distribution $p(q_t | \{O_t\}, q_0, y_T)$. In this case the forward distribution is a single Gaussian, the backward distribution is an unweighted mixture of Gaussians, and the smoothed distribution is a weighted mixture of Gaussians (where the weights w_k^s on each component reflect the probability that the hand will end up at target location k). The white plus signs in the top and bottom panel show the known end-targets, while the white disc in the bottom panel shows the noisy observation made at that time. Note, in particular, the forward and smoothed estimates at $T = 0.6$. At this time, the smoothed estimate, conditioned on the targets, shows peaks in its distribution indicative of this a priori knowledge. Note also that by $T = 0.8$ the smoothed estimate has zeroed in on the correct reach target while the forward-filter estimate is still at a distance from it. The values of the various parameters are presented in the accompanying text.

As can be seen from the figure, the mixture model is a useful framework in which to model reach to one of multiple targets. While the forward filter (by construction) does not take into account the location of the targets, the target-conditioned estimate of q_t is better at minimizing the error on average between the estimated state and the true state.

CONCLUSIONS

We have discussed the problem of constructing reaching trajectories conditioned on a known initial state, a noisy observation of the terminal state, and neural data. We made the assumption that arm motion can be reasonably modeled by a linear dynamical system driven by Gaussian noise, and we also assumed that measurement noise over the terminal state is Gaussian. While the noise model for the observational process is not required to be Gaussian, at each step of the forward filter the posterior distribution after incorporating the neural observations is approximated by a Gaussian. Under these simplifying assumptions we have presented a method which can compute the target-conditioned trajectories, in different cases, in a computationally efficient manner allowing for its real-time implementation.

The first and foremost requirement of any algorithm proposed for a neural prosthetic setting is its real-time applicability. The most important advantage that our method offers is its computational speed. Our modeling approach is also very flexible. Using our approach, we have considered cases where reach is to a target over which noisy observations are made, to different targets at different times, and to one of several targets.

The major limitations of our method arise from our assumptions of linearity for the arm dynamics and assumptions about the Gaussian nature of the various sources of noise. These assumptions were made to allow for a real-time implementation. The most limiting of these assumptions is that regarding linear dynamics for arm motion. We note that recent approaches which extend the Kalman filter to nonlinear systems without significantly increasing computational overhead [30], [31] might enable us to circumvent this limitation.

ACKNOWLEDGMENTS

Jayant Kulkarni would like to acknowledge the Swartz Foundation for a post-doctoral fellowship which supported this research. Liam Paninski is supported by an Alfred P. Sloan Research Fellowship and an NSF Faculty Early Career Development (CAREER) award.

AUTHORS

Jayant E. Kulkarni (jk2619@columbia.edu) received his doctoral degree from the Department of Theoretical and Applied Mechanics at Cornell University, and he is currently a Swartz Fellow at the Center for Theoretical Neuroscience at Columbia University.

Liam Paninski (liam@stat.columbia.edu) is an assistant professor in the Department of Statistics and affiliated with the Center for Theoretical Neuroscience at Columbia University.

REFERENCES

- [1] J. Donoghue, "Connecting cortex to machines: Recent advances in brain interfaces," *Nature Neurosci.*, vol. 5, pp. 1085–1088, Nov. 2002.
- [2] A.B. Schwartz, "Cortical neural prosthetics," *Annu. Rev. Neurosci.*, vol. 27, pp. 487–507, 2004.
- [3] L.R. Hochberg, M.D. Serruya, G.M. Friehs, J.A. Mukand, M. Saleh, A.H. Caplan, A. Branner, D. Chen, R.D. Penn, and J.P. Donoghue, "Neuronal ensemble control of prosthetic devices by a human with tetraplegia," *Nature*, vol. 442, no. 7099, pp. 164–171, 2006.
- [4] R. Kettner, A. Schwartz, and A. Georgopoulos, "Primate motor cortex and free arm movements to visual targets in three-dimensional space. III. Positional gradients and population coding of movement direction from various movement origins," *J. Neurosci.*, vol. 8, no. 8, pp. 2938–2947, 1988.
- [5] A. Georgopoulos, R. Kettner, and A. Schwartz, "Neuronal population coding of movement direction," *Science*, vol. 233, pp. 1416–1419, Sept. 1986.
- [6] D. Moran and A. Schwartz, "Motor cortical representation of speed and direction during reaching," *J. Neurophys.*, vol. 82, no. 5, pp. 2676–2692, 1999.
- [7] D. Flament and J. Hore, "Relations of motor cortex neural discharge to kinematics of passive and active elbow movements in the monkey," *J. Neurophys.*, vol. 60, no. 4, pp. 1268–1284, 1988.
- [8] J. Messier and J.F. Kalaska, "Covariation of primate dorsal premotor cell activity with direction and amplitude during a memorized-delay reaching task," *J. Neurophys.*, vol. 84, no. 1, pp. 152–165, 2000.
- [9] L. Paninski, M. Fellows, S. Shoham, N. Hatsopoulos, and J. Donoghue, "Superlinear population encoding of dynamic hand trajectory in primary motor cortex," *J. Neurosci.*, vol. 24, no. 39, pp. 8551–8561, 2004.
- [10] A. Brockwell, A. Rojas, and R. Kass, "Recursive Bayesian decoding of motor cortical signals by particle filtering," *J. Neurophys.*, vol. 91, no. 4, pp. 1899–1907, 2004.
- [11] U. Eden, L. Frank, R. Barbieri, V. Solo, and E. Brown, "Dynamic analysis of neural encoding by point process adaptive filtering," *Neural Comput.*, vol. 16, no. 5, pp. 971–978, May 2004.
- [12] G. Santhanam, S.I. Ryu, B.M. Yu, A. Afshar, and K.V. Shenoy, "A high-performance brain-computer interface," *Nature*, vol. 442, no. 7099, pp. 195–198, 2006.
- [13] W. Wu, E. Bienenstock, J.P. Donoghue, and M.J. Black, "Bayesian population coding of motor cortical activity using a Kalman filter," *Neural Comput.*, vol. 18, no. 80–118, 2006.
- [14] E. Brown, L. Frank, D. Tang, M. Quirk, and M. Wilson, "A statistical paradigm for neural spike train decoding applied to position prediction from ensemble firing patterns of rat hippocampal place cells," *J. Neurosci.*, vol. 18, pp. 7411–7425, Sept. 1998.
- [15] W. Truccolo, U. Eden, M. Fellows, J. Donoghue, and E. Brown, "A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects," *J. Neurophys.*, vol. 93, no. 2, pp. 1074–1089, 2005.
- [16] C. Kemere, G. Santhanam, B. Yu, K.V. Shenoy, and T. Meng, "Decoding of plan and peri-movement neural signals in prosthetic systems," in *Proc. IEEE Workshop Signal Processing Systems*, 2002, pp. 276–283.
- [17] L. Srinivasan, U.T. Eden, A.S. Willsky, and E.N. Brown, "A state-space analysis for reconstruction of goal-directed movements using neural signals," *Neural Comput.*, vol. 18, no. 10, pp. 2465–2494, 2006.
- [18] D. Taylor, S. Tillery, and A. Schwartz, "Direct cortical control of 3-D neuroprosthetic devices," *Science*, vol. 296, no. 5574, pp. 1829–1832, 2002.
- [19] R. Andersen, S. Musallam, and B. Pesaran, "Selecting the signals for a brain-machine interface," *Current Opinion in Neurobiol.*, vol. 14, no. 6, pp. 720–726, 2004.
- [20] E.C. Leuthardt, G. Schalk, J.R. Wolpaw, J.G. Ojemann, and D.W. Moran, "A brain-computer interface using electrocorticographic signals in humans," *J. Neural Eng.*, vol. 1, no. 2, pp. 63–71, 2004.
- [21] B.M. Yu, G. Santhanam, S.I. Ryu, and S.K. V., "Feedback-directed state transition for recursive bayesian estimation of goal-directed trajectories," in *Proc. Computational and Systems Neuroscience Meeting*, Salt Lake City, UT, 2005, p. 305.
- [22] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, 1989.
- [23] R. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME- J. Basic Eng.*, pp. 35–40, Mar. 1960.
- [24] R. Kalman and R. Bucy, "New results linear filtering and prediction problems," *Trans. ASME- J. Basic Eng.*, vol. 83, p. 83, 1961.
- [25] S. Shoham, L. Paninski, M. Fellows, N. Hatsopoulos, J. Donoghue, and R. Normann, "Optimal decoding for a primary motor cortical brain-computer interface," *IEEE Trans. Biomed. Eng.*, vol. 52, no. 7, pp. 1312–1322, 2005.
- [26] C. Moler and C. Van Loan, "Nineteen dubious ways to compute the exponential of a matrix," *SIAM Rev.*, vol. 20, no. 4, pp. 801–836, 1978.
- [27] R.A. Horn and C.R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1986.
- [28] B.M. Yu, C. Kemere, G. Santhanam, A. Afshar, S.I. Ryu, T.H. Meng, M. Sahani, and K.V. Shenoy, "Mixture of trajectory models for neural decoding of goal-directed movements," *J. Neurophys.*, vol. 97, no. 5, pp. 3763–3780, 2007.
- [29] L. Srinivasan, U. Eden, S. Mitter, and E. Brown, "General purpose filter design for neural prosthetic devices," *J. Neurophys.*, vol. 98, no. 4, pp. 2456–2475, 2007.
- [30] S.J. Julier and J.K. Uhlmann, "A new extension of the kalman filter to nonlinear systems," in *Proc. AeroSense: The 11th Int. Symp. Aerospace/Defence Sensing, Simulation and Controls*, 1997.
- [31] E. Wan and R. Merwe, "The unscented Kalman filter," in *Kalman Filtering and Neural Networks*, S. Haykin, Ed. Hoboken, NJ: Wiley, 2004, pp. 131–158. 