Dynamic-goal state equations for tracking reaching movements using neural signals

Lakshminarayan Srinivasan[*,†,‡] and Emery N. Brown[†,‡,**]

[*]Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology
[†]Division of Health Sciences & Technology, Harvard / MIT
[‡]Department of Anesthesia and Critical Care, Massachusetts General Hospital
[**]Department of Brain and Cognitive Sciences, MIT

***Abstract* – Motor prosthetic algorithms were recently proposed to combine target and path information to drive reaching arm movements to a static goal. In this paper, we extend two approaches to support goals that may themselves evolve over the duration of the reaching movement. The resulting probabilistic and control-based dynamic-goal reach state equations represent an intermediate level of user flexibility between static-goal reach state equations and unconstrained movement.**

***Index Terms* – motor, neural prosthetics, goal-directed, estimation, state equation.**

I. Introduction

STUDIES within the past five years have demonstrated the feasibility of controlling a device from electrodes that monitor neural activity. Various anatomical regions have been investigated in this endeavour, including motor, premotor, and parietal cortical regions. Two principle interfaces that have been prototyped include continuous movements of an on-screen cursor [1,2,3], and selection from a discrete set of icons [4,5]. Called motor and cognitive prosthetics respectively, both approaches must provide a protocol by which neural signals specify the state of the device being controlled. This problem is conveniently described as the inference of a user's intent from observations of their neural signals.

Recently, several papers have investigated estimation procedures designed for goal-directed movements, where neural activity might be available to specify the intended target state as well as the path towards the target state [6,7,8,9]. These methods enable an intermediate level of flexibility between discrete target selection and unconstrained free motion in the control of a computer icon, robot arm, or other assistive device. Two of these proposed methods, one derived probabilistically and another based on stochastic optimal control, employ the state-space paradigm, meaning that a difference or differential equation is specified to generate a prior on the trajectory as it evolves in time towards the target. A state equation forms the basis for neural interfaces that use recursively computed estimates, including the Kalman filter [3], point process filter [10], and particle filter [11].

Both probabilistic and control-based static-goal reach state equations share the same linear structure. The state vector includes the current desired device state x_t and a static target state z:

$$\begin{pmatrix} x_{t+1} \\ z \end{pmatrix} = M_t \begin{pmatrix} x_t \\ z \end{pmatrix} + \begin{pmatrix} \varepsilon_t \\ 0 \end{pmatrix} \tag{1.1}$$

where

$$M_t = \begin{pmatrix} \Psi & \Gamma \\ 0 & I \end{pmatrix}, \tag{1.2}$$

I denotes the identity matrix, and ε_t is the increment, a random variable that represents uncertainty in the propagation of the state variables forward in time.

While the probabilistic and control-based reach state equations differ in specific parameter choices, they both describe reaching movements where the desired target state z remains unchanged over the duration of the reach, and the increment random variable adding to the target component is identically zero.

In practice, we may want to support the possibility of a target state that evolves in time, such as with a fieldsman that must adjust their reach during an unanticipated bounce of a baseball. Even when the intended target is constant, estimation on state variables that evolve forward in time without incurring uncertainty (directly or through relations with other variables) is generally susceptible to bias when the initial estimate is inaccurate and calculated to have an inappropriately low variance. For dimensions of the state equation to which stochastic increments do not propagate,

* This work was supported in part by the NIH Medical Scientist Training Program Fellowship to LS and NIH grant R01 DA015644 to ENB. Correspondence can be addressed to ls2@neurostat.mgh.harvard.edu

the estimation procedure then down-weights new observations on those dimensions, and estimates are slow to settle around the correct value. In this paper, we apply both probabilistic and control frameworks to derive two alternate state equations for describing reaching movements with an intended target that may itself evolve over the duration of the reach.

II. Dynamic Goal in Probabilistic Framework

We seek a probabilistically consistent description for a trajectory x_t that evolves towards the current target z_t in the same way the trajectory evolved towards a stationary target z in (1.1). The stationary target state equation (1.1) is discussed extensively in [6]. It is a specification for the state transition probability density

$$p(x_{t+1} \mid x_t, z) \tag{1.3}$$

based on the single assumption that x_t evolves as a standard linear state equation to describe the evolution of the arm without target information over the discrete time interval [0,T]:

$$x_{t+1} = A_t x_t + w_t \tag{1.4}$$

The stochastic increment w_t is a zero-mean Gaussian random variable with $E[w_t w_\tau'] = Q_t \delta_{t-\tau}$. Note the indexing used in equation (1.4) follows general convention [12] and is different from that used previously in [6]. The state transition matrix and increment are now marked A_t and w_t rather than A_{t+1} and w_{t+1}.

We desire additionally to describe the temporal evolution of the target as another linear state equation

$$z_{t+1} = B_t z_t + \eta_t \tag{1.5}$$

with zero-mean Gaussian increment η that has covariance $E[\eta\eta'] = R_t \delta_{t-\tau}$.

A graphical model of the state space provides a convenient tool to visualize the relationship between the random variables that comprise the full trajectory of arm states and time-varying targets. In this representation, the x_t and z_t are nodes in a directed acyclic graph. A random variable at a given node is independent of random variables on other nodes in the space conditioned on its parent node random variables. To generate a probabilistically consistent prior on the trajectory, the conditional density must be specified for each node conditioned on its parents.

Several possible graphs could be drawn that relate target variables z_t to path variables x_t, each of which would require that we specify different conditional densities to define a consistent prior on the entire trajectory.

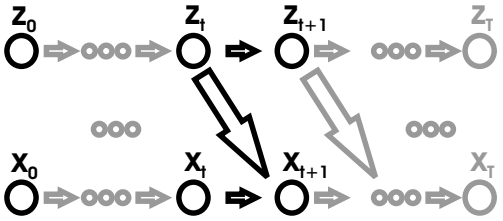

Figure 1. One of many possible graphical model descriptions of a dynamic-goal reach state equation. Nodes of this directed acyclic graph represent random variables. To generate a consistent prior across all nodes, one conditional probability density is specified for each node conditioned on its parents. This particular model is a convenient choice in that it explicitly specifies conditional densities $p(x_{t+1}|x_t, z_t)$ analogous to those that were previously described for the static goal reach state equation.

The graph depicted in Figure 1 requires precisely the conditional densities we described above from equations (1.3) and (1.5). Specifically, each target node z_{t+1} has only one parent z_t, and each arm state node x_{t+1} has parent nodes x_t, and z_t. Nodes x_0 and z_0 are parentless. Accordingly, we must define $p(x_{t+1} \mid x_t, z_t)$, $p(z_{t+1} \mid z_t)$, $p(x_0)$, and $p(z_0)$.

By setting $p(x_{t+1} \mid x_t, z_t)$ as given in equation (1.3), we capture the notion that the arm state x_t evolves toward the current target z_t as if it were the stationary target in the original reach state equation. By setting $p(z_{t+1} \mid z_t)$ according to equation (1.5), we introduce the constraint that the target itself evolves as a linear state equation with Gaussian increments, independent of the arm state x_t.

Because the graphical model we chose stipulates that x_{t+1} and z_{t+1} are independent when conditioned on x_t and z_t, we have for this model that

$$p(x_{t+1}, z_{t+1} \mid x_t, z_t) = p(x_{t+1} \mid x_t, z_t)\, p(z_{t+1} \mid z_t) \tag{1.6}$$

The final probabilistic dynamic-goal reach state equation is a variant of the original reach state equation:

$$\begin{pmatrix} x_t \\ z_t \end{pmatrix} = R_t \begin{pmatrix} x_{t-1} \\ z_{t-1} \end{pmatrix} + \begin{pmatrix} \varepsilon_t \\ \eta_t \end{pmatrix} \tag{1.7}$$

where

$$R_t = \begin{pmatrix} \Psi & \Gamma \\ 0 & B \end{pmatrix} \tag{1.8}$$

The symbols are identical to those defined for equations (1.1), (1.2), and (1.5), with Ψ and Γ defined from the probabilistic static-goal reach state equation, as detailed in [6]. Moreover, the solution reduces to the static case with $B_t = I$ and $E[\eta\eta'] = 0$.

III. Dynamic Goal in Stochastic Optimal Control Framework

The control-based static-target reach state equation also affords a convenient extension to support a dynamic target. The static target case is addressed in [9], where the authors propose a reach state equation that corresponds to a linear plant under a control optimized to a cost that is a quadratic form on the end state and control inputs. A concise and accessible introduction to this classic linear quadratic (LQ) optimal control solution is available [12].

To derive the closed-loop state equation for optimized control to a dynamic target, we adapt the same general approach as in [9] and [12]. We write the problem in a standard form and directly apply the classical result for the optimal controller in linear systems with quadratic cost discussed in [12].

First, analogous to the static case in [9], we augment the state vector $s_t = [x_t \quad z_t]$ to include both x_t and z_t. We then write the state equation that describes the dynamics of the plant:

$$\begin{pmatrix} x_{t+1} \\ z_{t+1} \end{pmatrix} = \begin{pmatrix} A_t & 0 \\ 0 & B_t \end{pmatrix} \begin{pmatrix} x_t \\ z_t \end{pmatrix} + Cu_t + \begin{pmatrix} w_t \\ \eta_t \end{pmatrix} \tag{1.9}$$

where A_t, B_t, w_t, and η_t are defined in section II. For an mx1 control vector u_t, the nxm matrix C specifies how control inputs affect plant dynamics. This matrix C can be set to specify control along certain dimensions, such as with the assignment in (1.20) to specify velocity control.

The cost function is now a standard quadratic form, penalizing for deviations between x_T and z_T as well as for energy expenditure associated with the control u_t:

$$J(s_0) = \underset{\substack{w_t \\ k=0,1,\ldots,T-1}}{E} \left\{ s_T' Q_T s_T + \sum_{t=0}^{T-1} u_t' R u_t \right\} \tag{1.10}$$

where Q_T is a 2N x 2N block matrix and x_T and z_T are each Nx1 vectors. The following assignment for Q_T weighs deviations between all dimensions of x_T and z_T equally, where q is a scalar:

$$Q_T = q \times \begin{pmatrix} I & -I \\ -I & I \end{pmatrix} \tag{1.11}$$

R_T is chosen to penalize the cost according to the square of the deviation from zero in each control dimension:

$$R_T = r \times I \tag{1.12}$$

where R is a $m \times m$ matrix given a $m \times 1$ control vector u_t, and r is a scalar.

Stated in this form, the optimal controller for a moving target is simply given by the classical optimal controller for linear systems with quadratic cost, reproduced below from [12]:

$$u_t = L_t s_t \qquad t = 0,1,\ldots,T-1 \tag{1.13}$$

where

$$L_t = -(B_t' K_{t+1} B_t + R)^{-1} B_t' K_{t+1} A_t \tag{1.14}$$

and K_{k+1} is calculated recursively from the following two equations:

$$K_T = Q_T \tag{1.15}$$

$$K_t = A_t'(K_{t+1} - K_{t+1} B_t (B_t' K_{t+1} B_t + R_t)^{-1} B_t' K_{t+1}) A_t + Q_t \qquad t = 0,1,\ldots,T-1 \tag{1.16}$$

The resulting closed loop state equation is simply (1.9) with u_t as defined in (1.13). This is the dynamic-goal control-based reaching movement state equation.

A simple derivation of the well-studied equations (1.13)-(1.16) is given in [12]. Note that if B_t=I, the deterministic component of the control-based dynamic-goal reach state equation is identical to the static goal case. If furthermore $E[\eta\eta']$= 0, then the dynamic and static equations are identical. The fact that a target increment appears in the dynamic-goal plant but not the static-goal plant does not affect the choice of optimal control policy, a result known as certainty equivalence [12] by which deterministic and stochastic closed-loop control strategies are equivalent in the subclass of LQ problems where plant dynamics are known.

IV. 2D Sample Trajectories of Dynamic Goal State Equations in Both Frameworks

We now generate sample trajectories from the probabilistic (1.7) and optimal control (1.9) dynamic goal state equations in the x-y plane. First, the matrices in those equations are precomputed, and increment values are drawn from independent Gaussian distributions. Next, arm states are iteratively computed using the respective state equations. Both the current arm state x_t and current target z_t are each 4x1 vectors composed of the following elements: [x coord , y coord, x velocity, y velocity]'. The 4x4 matrices A_t and B_t that describe the evolution of x_t and z_t respectively are chosen for both state equations as follows:

$$A = B = \begin{pmatrix} 1 & 0 & \Delta & 0 \\ 0 & 1 & 0 & \Delta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{1.17}$$

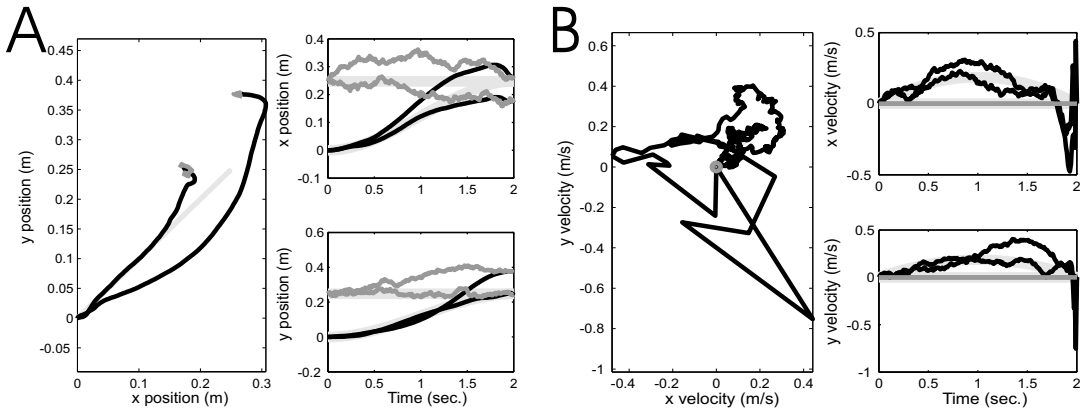

Figure 2. Sample position (A) and velocity (B) traces from the probabilistic dynamic-goal reach state equation. Trajectories are simulated in a 2 dimensional workspace according to state equation (1.7) with parameters described in section IV. In both panels, black lines indicate the current arm state from the x_t vector, dark grey lines indicate the current target state from the z_t vector, and light grey lines indicate the average trajectory obtained analytically from the state equation by setting stochastic increments identically to zero.

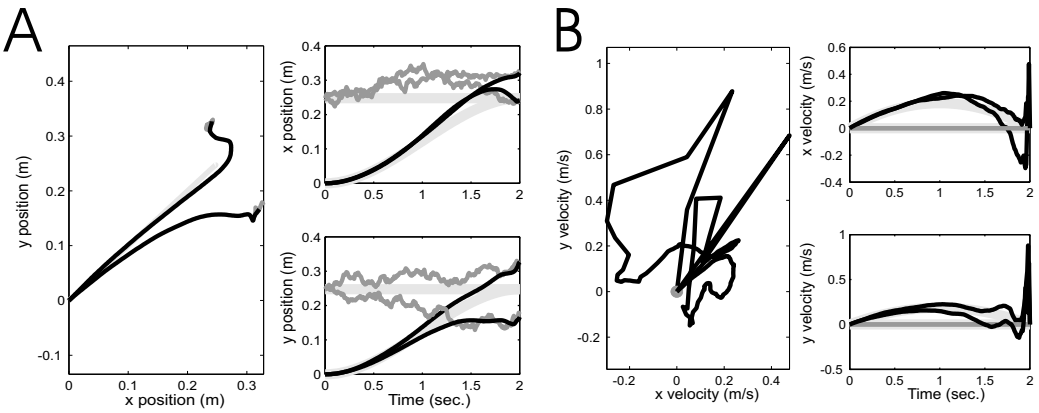

Figure 3. Sample trajectories from the control-based approach to the dynamic-goal reach state equation. Trajectories are simulated in a 2 dimensional workspace according to state equation (1.9) with parameters described in section IV. In both panels, black lines indicate the current arm state from the x_t vector, dark grey lines indicate the current target state from the z_t vector, and light grey lines indicate the average trajectory obtained analytically from the state equation by setting stochastic increments identically to zero.

The 4x1 increment matrices w_t and η_t have the following 4x4 covariance matrices in both state equations:

$$E[w_t w_t] = \omega \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{1.18}$$

$$E[\eta_t \eta_t] = n \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{1.19}$$

where ω = 1e-4 and n = 5e-3.

For the control-based state equation, q = 440 in (1.11), and r = 1e-7 in (1.12). The control u_t is a 2x1, with C defined as a 8x2 matrix to specify velocity control:

$$C = Transpose \begin{bmatrix} 00100000 \\ 00010000 \end{bmatrix} \tag{1.20}$$

Figures 2 and 3 show sample trajectories resulting from the two state equations. The A panels in both figures show trajectories that move towards the current notion of target that itself evolves independently as linear stochastic difference equation. A close examination of the x position versus y position graphs in the A panels reveals that the probabilistic trajectory (Figure 2A) converges to z_{T-1}, whereas the velocity-control-based trajectory (Figure 3A) converges to z_{T-2}. The z_{T-1} convergence occurs in the probabilistic trajectory because B=I and $E[\eta_{T-1}]$=0 implies

the distribution of z_T given z_{T-1} is centered around the value of z_{T-1}. This is also true in the control-based situation, but in addition, velocity control inputs applied at time t take one time step to propagate into position. This results in a convergence to z_{T-2} in the control-based state equation with B=I and $E[\eta_{T-1}]=0$.

The B panels in Figures 2 and 3 show mean velocity trajectories, obtained analytically by setting the stochastic increments identically to zero, that appear bell-shaped as with the static case. Unlike the static case, the velocity profiles of Figures 2 and 3 deviate sharply (but only momentarily) with high velocity in the last four time steps in an effort to close the remaining distance to the stochastic target locations. These last minute high velocity corrections are presumably not needed in the static case because the target remains constant, so that velocity corrections throughout the reach provide the desired accuracy in achieving the final target.

V. Conclusion

Recent advances in the design of motor prosthetic algorithms allow for the coordinated estimation of arm trajectory and a static goal in a reaching movement [6,7,8,9]. In this paper, we have extended this work to describe the estimation of reaching movements towards dynamic goals. This extension represents an alternative for the neural prosthetic interface that may provide increased flexibility over a static-goal reach state equation and better convergence of target estimates, while still maintaining the ability to integrate target and path-related neural activity.

Two state equations were presented, each of which represents the extension of probabilistic [6] and control-based [9] approaches to the static target case. Both state equations reduce to their static counterparts for appropriately chosen parameters. Simulated trajectories of the two evolving-goal state equations both appear qualitatively similar for the specific parameter choices made in Section IV, although they differ in derivation and mathematical specification. Both state equations allow for an independent, stochastically evolving target. Simulation of sample neural activity and estimation performed with these state equations will be critical to evaluating the advantages and limitations of the dynamic-goal approach compared to the static-goal reach state equations.

Acknowledgment

L.S. thanks Jason Johnson, Dimitry Malioutov, Uri T. Eden, Aman Chawla, Lav Varshney, and Ali Shoeb for discussions.

References

[1] Carmena, J. M., Lebedev, M. A., Crist, R. E., O'Doherty, J. E., Santucci, D. M., Dimitrov, D. F., Patil, P. G., Henriquez, C. S., & Nicolelis, M. A. L. (2003). Learning to control a brain-machine interface for reaching and grasping by primates. *Public Library of Science Biology, 1(2),* 193-208.

[2] Taylor, D. M., Tillery, S. I. H., & Schwartz, A. B. (2002). Direct cortical control of 3D neuroprosthetic devices. *Science, 296*, 1829-1832.

[3] Wu, W., Shaikhouni, A., Donoghue, J. P., & Black, M. J. (2004). Closed-loop neural control of cursor motion using a Kalman filter. *The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, San Francisco, CA.

[4] Mussallam, S., Corneil, B. D., Greger, B., Scherberger, H., & Andersen, R. A. (2004). Cognitive control signals for neural prosthetics. Science, 305, 258-262.

[5] Santhanam, G., Ryu, S. I., Yu, B. M., Afshar, A., Shenoy, K. V. (In Press). A high performance neurally-controlled cursor positioning system. IEEE Engineering in Medicine and Biology (EMBS) 2nd International Conference on Neural Engineering.

[6] Srinivasan, L., Eden, U. T., Willsky, A. S., & Brown, E. N. (2005). Goal-directed state equation for tracking reaching movements using neural signals. Proceedings of the 2nd International IEEE EMBS Conference on Neural Engineering, Arlington, VA, p. 352.

[7] Cowan, T. M., Taylor, D. M. (2005). Predicting reach goal in a continuous workspace for command of a brain-controlled upper-limb neuroprosthesis. Proceedings of the 2nd International IEEE EMBS Conference on Neural Engineering, Arlington VA, p. 74.

[8] Yu, B. M., Santhanam, G., Ryu, S. I., & Shenoy, K. V. (2005). Feedback-directed state transition for recursive Bayesian estimation of goal-directed trajectories. Computational and Systems Neuroscience (COSYNE) meeting, Salt Lake City, UT.

[9] Kemere, C., & Meng, T. H. (2005). Optimal estimation of feed-forward-controlled linear systems. IEEE International Conference on Acoustics, Speech and Signal Processing, Philadelphia, PA.

[10] Eden, U. T., Frank, L. M., Barbieri, R., Solo, V., & Brown, E. N. (2004). Dynamic analyses of neural encoding by point process adaptive filtering. *Neural Computation, 16(5),* 971-998.

[11] Brockwell, A.E., Rojas, A.L., & Kass, R.E. (2004). Recursive bayesian decoding of motor cortical signals by particle filtering. J. Neurophysiology, 91(4):1899-1907.

[12] Bertsekas, Dimitri P. (2005) *Dynamic Programming and Optimal Control*, 3rd Edition, Vol. 1, Athena Scientific.