Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Rapid Computation of Time-Optimal, Open-Loop Forearm Movement

Jeffrey N. Shelton [*] **Oh-Sang Kwon** [**] **George T.-C. Chiu** [***]

[*] *School of Mechanical Engineering, Purdue University, West Lafayette, IN 47907, USA (shelton@purdue.edu)*
[**] *Department of Psychological Sciences, Purdue University, West Lafayette, IN 47907, USA (landarzt@psych.purdue.edu)*
[***] *School of Mechanical Engineering, Purdue University, West Lafayette, IN 47907, USA (gchiu@purdue.edu)*

Abstract: Minimal movement time for open-loop rotation of the human forearm, along with the associated input signal to the forearm muscles, is calculated using matrix multiplication. This permits rapid evaluation of movement times across a four-dimensional mesh of initial conditions, each moving to a common terminal state. The described discrete-time solution is based on the continuous-time solution of Tanaka et al., and the minimum-variance theory of Harris and Wolpert. Underlying algorithm concepts are discussed, and proofs of solution existence are provided.

Keywords: human movement; time-optimal control; minimum-variance control.

1. INTRODUCTION

An infinite number of trajectories are available for any reaching task. Yet humans easily select and execute one particular trajectory each time they reach for a drinking glass, or pick up a pen. These aimed movements display stereotypical kinematic properties; the hand moves generally in a straight line toward the target, exhibiting a bell-shaped velocity profile, with an interval of positive acceleration followed immediately thereafter by a period of deceleration (Khan et al. [2006]). This commonality in human reaching suggests the utilization of optimization to select a “best” path from among all the available options.

Finding time-optimal rotation of the human forearm is the focus of this manuscript, although the methods described herein have also been used to model rapid eye movements. Experimental data for rapid forearm rotation can be seen in Fig. 1. The smooth, but slightly skewed velocity profile is typical of aimed movement in humans. Waviness in the acceleration curve is believed due to corrective submovements initiated en route to the target.

Harris and Wolpert described a minimal variance solution for single-joint movement of the forearm, in which movement time is fixed and positional variance is minimized over some post-movement interval (Harris and Wolpert [1998]). This method produces a trajectory that possesses many of the characteristics of aimed human movement. Fundamental to this approach is an assumption that the neural control signal, which directs muscle movement, is corrupted by a gaussian noise exhibiting a variance proportional to the control signal magnitude. The resulting trajectory agrees with with Fitts’ Law (Fitts [1954]), a widely accepted expression of the tradeoff between speed and accuracy in rapid point-to-point motion. Further work by Feng et al. [2004] has demonstrated how the firing patterns of neurons could accomplish minimal variance movement.

One difficulty in applying the Harris method is that movement duration must be assumed *a priori*. A potential means for overcoming this shortcoming is to repeatedly apply the convex programming solution utilized by Harris and Wolpert, and vary movement time until the resulting variance falls below a desired bound. However, this approach is relatively slow. Seeking a solution for optimal movement duration, Tanaka et al. relate a time-optimal solution in which movement time is minimized, while variance is bounded (Tanaka et al. [2006]). This results in an equation that can be iteratively solved to determine the duration of a time-optimal input, without having to solve the optimal input problem as an intermediate step.

The Tanaka method solves the open-loop problem; a time-optimal closed-loop solution is likewise available using a linear quadratic gaussian (LQG) formulation. However, human motion appears to use irregular, intermittent feedback, typically resulting in two or three discrete submovements that comprise most aimed motions (Fishbach et al. [2007]). To examine intermittent feedback, it was desired to find the time-optimal combination of two open-loop submovements, with the profile of the second submovement based on the terminal state of the first. This article describes computation of durations for the second submovement, thus allowing for rapid evaluation of total movement time once initial submovement duration and terminal state are known.

A four-dimensional mesh (with axes for position, velocity, acceleration, and jerk) is constructed to span a wide range of second submovement initial states, with the optimal movement duration to a common target associated with each point. As the Tanaka approach requires continuous-

978-1-1234-7890-2/08/$20.00 © 2008 IFAC 10.3182/20080706-5-KR-1001.2651

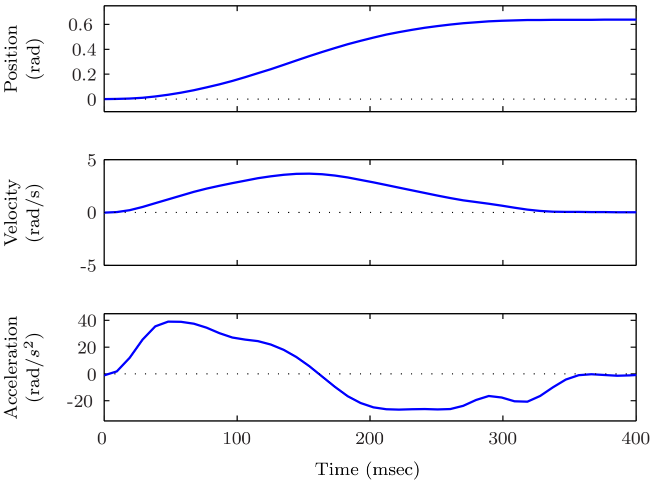

Fig. 1. Forearm movement profile showing mean values of 30 trials for a single individual. Forearm rotation of approximately 36° to a target of 2° width.

time integration, an alternative, computationally efficient, algorithm was sought. In contrast to the Tanaka method, the solution described herein calculates optimal movement time using matrix multiplication; this is accomplished by moving the computation to discrete-time. This paper also provides proofs of applicability and details the underlying theory. Nonetheless, Tanaka et al. must be credited for doing much of the heavy lifting.

2. STATE-SPACE MODEL

To simulate rotation of the human forearm about the elbow joint in a single plane, both Harris and Tanaka adopt the "standard" model taken from van der Helm and Rozendaal [2000]. A second-order linear model describing muscle dynamics,

$$\tau + (t_a + t_e)\,\dot{\tau} + (t_a t_e)\,\ddot{\tau} = u \tag{1}$$

is appended to the second-order linear skeletal model,

$$J\ddot{\theta}(t) + B\dot{\theta}(t) = \tau(t) \tag{2}$$

In (1), parameters t_a and t_e represent the time constants of muscle activation and excitation, J is the moment of inertia, and B is intrinsic viscosity (damping). Forearm position with respect to the upper arm is denoted by θ, while τ represents the net muscle torque applied at the joint, and u is the neural input signal to the muscle. Values for the time constants are $t_a = 30$ ms and $t_e = 40$ ms, while J and B are set to 0.25 $\mathrm{kg}\cdot\mathrm{m}^2$ and 0.20 $\mathrm{kg}\cdot\mathrm{m}^2/\mathrm{s}$, respectively.

Repeatedly differentiating (2) with respect to time and substituting into (1), it is possible to express the system dynamics as

$$\ddddot{\theta} + \alpha_3\dddot{\theta} + \alpha_2\ddot{\theta} + \alpha_1\dot{\theta} + \alpha_0\theta = \beta u \tag{3}$$

where

$$\begin{aligned}\alpha_3 &= \frac{B}{J} + \left(\frac{1}{t_a} + \frac{1}{t_e}\right)\\ \alpha_2 &= \frac{1}{t_a t_e} + \left(\frac{1}{t_a} + \frac{1}{t_e}\right)\frac{B}{J}\\ \alpha_1 = \frac{B}{J t_a t_e};\qquad \alpha_0 &= 0;\qquad \beta = \frac{1}{J t_a t_e}\end{aligned} \tag{4}$$

The linear system of (3) can be expressed in continuous-time state-space form as

$$\begin{bmatrix}\dot{\theta}\\ \ddot{\theta}\\ \dddot{\theta}\\ \ddddot{\theta}\end{bmatrix} = \underbrace{\begin{bmatrix}0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ -\alpha_0 & -\alpha_1 & -\alpha_2 & -\alpha_3\end{bmatrix}}_{\Phi}\begin{bmatrix}\theta\\ \dot{\theta}\\ \ddot{\theta}\\ \dddot{\theta}\end{bmatrix} + \underbrace{\begin{bmatrix}0\\ 0\\ 0\\ \beta\end{bmatrix}}_{\Psi} u \tag{5}$$

Letting the state vector be $\mathbf{x} = \left[\theta\ \dot{\theta}\ \ddot{\theta}\ \dddot{\theta}\right]^T$, it is possible to represent this relationship as

$$\dot{\mathbf{x}} = \Phi\mathbf{x} + \Psi u \tag{6}$$

where Φ and Ψ are the matrix and vector so denoted in (5). It is desired, however, to express the system dynamics in discrete-time notation; that is, in the form

$$\mathbf{x}(n+1) = A\mathbf{x}(n) + Bu(n); \qquad n = 0, 1, 2, \ldots \tag{7}$$

where n is a non-negative integer representing the number of uniform time periods, each of length T, that have passed since $n = 0$. For a linear time-invariant system, discretization of the continuous-time system is well-known (see Chapter 4 of Franklin et al. [1997]):

$$A = e^{\Phi T}; \qquad B = \left[\int_0^T e^{\Phi\xi} d\xi\right]\Psi \tag{8}$$

This conversion can become quite cumbersome when performed by hand, so the Matlab command 'c2d' is utilized to convert Φ and Ψ to their discrete-time equivalents, A and B. Hereafter, the initial state, $\mathbf{x}(0)$, is denoted as $\mathbf{x}_0$, and it is assumed that $u(n) = 0$ for all $n < 0$.

3. VARIANCE CALCULATION

Assume that the input signal $u(n)$ is corrupted by a white, zero-mean, gaussian noise, $w(n)$. The system equation thus becomes

$$\mathbf{x}(n+1) = A\mathbf{x}(n) + B\left[u(n) + w(n)\right] \tag{9}$$

It can be shown that the solution to (9) is

$$\mathbf{x}(n) = A^n\mathbf{x}_0 + \sum_{i=0}^{n-1} A^{n-1-i}B\left[u(i) + w(i)\right] \tag{10}$$

With random noise added to the input signal, state variable $\mathbf{x}(n)$ becomes a random vector. The covariance matrix for $\mathbf{x}$ is expressed as

$$\Sigma = \mathrm{cov}\left(\mathbf{x}, \mathbf{x}\right) = \mathbf{E}\Big[\left(\mathbf{x} - \mathbf{E}\left[\mathbf{x}\right]\right)\left(\mathbf{x} - \mathbf{E}\left[\mathbf{x}\right]\right)^T\Big] \tag{11}$$

Following the lead of Harris and Wolpert [1998], the white noise variance, σ_w^2, is assumed to be proportional to the square of the input magnitude, such that

$$\sigma_w^2 = \mathbf{E}\left[w(i)^2\right] = k|u|^2 \tag{12}$$

Substituting (10) and (12) into (11) produces a covariance matrix of

$$\Sigma(n) = k\sum_{i=0}^{n-1}\left(A^{n-1-i}B\right)\left(A^{n-1-i}B\right)^T u(i)^2 \tag{13}$$

Since the first element of state vector $\mathbf{x}$ is the angular position θ, the $(1,1)$ element of Σ contains the position variance, σ_θ^2. Let V represent the position variance held in the $(1,1)$ element of the 4×4 covariance matrix,

$$V(n) = \Sigma(n)_{[1,1]} \tag{14}$$

To allow for subsequent minimization, it is desired to express the variance in quadratic matrix (convex) form at each sample instant; that is,

$$V(n) = \mathbf{u}^T \; Q(n) \; \mathbf{u} \tag{15}$$

Suppose that

$$\chi(n) = \left[A^{n-1}B\right] \tag{16}$$

$$\rho(n) = \chi(n) \cdot \chi(n)^T \tag{17}$$

Then $\rho(i)$ is the (4×4) matrix multiplied by $u(i)^2$ in the summation of (13), and the (1,1) element of that matrix, once multiplied by the variability coefficient, k, is

$$\Gamma(n) = k\rho(n)_{[1,1]} \tag{18}$$

If $\Gamma(n)$ values are placed into diagonal matrix $\Omega(n) = \text{diag}\{\Gamma(n), \Gamma(n-1), \ldots, \Gamma(1)\}$, then the Q matrix of (15) can be expressed as

$$Q(n) = \left[\begin{array}{c|c} \Omega(n) & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} \end{array}\right] \tag{19}$$

Both Harris and Tanaka define two distinct intervals: a period of movement from the initial to final state, t_f, and a post-movement period, t_p, over which the post-movement variance is either bounded (Tanaka), or minimized (Harris). For a sample period of length T, the number of movement periods is $n_f = [t_f/T]$, and the number of post-movement periods is $n_p = [t_p/T]$. Thus, the length of the input vector u is $(n_f + n_p)$, and Q must accordingly be a square matrix of size $(n_f + n_p)$.

4. COST FUNCTION

Since a time-optimal response is desired, define a cost function that is equivalent to the number of movement periods:

$$J = n_f \tag{20}$$

Given the desired point-to-point motion, there are two relevant constraints. First, the expected state must reach the target state, $\mathbf{x}_f$, by the conclusion of the movement period, and remain there throughout the post-movement interval. Thus,

$$\mathbf{x}_f = \mathbf{E}\left[\mathbf{x}(n)\right]; \qquad n_f \leq n \leq (n_f + n_p) \tag{21}$$

Second, a bound must be placed on post-movement variance. To simplify computation, Tanaka et al. chose to enforce a bound on *average* variance during the post-movement interval. The resulting mean post-movement variance is defined as

$$\overline{V}_{pm}(n_f) = \frac{1}{n_p} \sum_{i=n_f+1}^{n_f+n_p} V(i) = \frac{1}{n_p}\, \mathbf{u}^T H(n_f)\mathbf{u} \tag{22}$$

where

$$H(n_f) = \sum_{i=n_f+1}^{n_f+n_p} Q(i) \tag{23}$$

Let $\widetilde{V}$ represent the variance bound to be enforced. Then, using the method of Lagrangian multipliers (see Chapter 3 of Stengel [1994]), augment the cost function with the aforementioned variance and position constraints,

$$J = n_f + \lambda \left[\widetilde{V} - \frac{\mathbf{u}^T H(n_f)\mathbf{u}}{n_p}\right] + \sum_{i=n_f}^{n_f+n_p} \mu^T(i)\left\{\mathbf{x}_f - \mathbf{E}\left[\mathbf{x}(i)\right]\right\} \tag{24}$$

The reasoning of Tanaka in evaluating the average post-movement variance is now evident; just a single multiplier, λ, is required to enforce the variance constraint after the movement is completed. To simplify the cost function so that, likewise, only a single (1×4) costate vector μ^T is needed, let the input $u(n)$ be constant throughout the post-movement interval, so that

$$u(n) = u_f = \frac{\alpha_0 \theta_f}{\beta}; \qquad (n_f \leq n \leq n_f + n_p)$$

with α_0 and β as defined in (3). This input serves to counteract any "springback" present in the system.

Given the constant post-movement input, u_f, it is necessary to enforce the target position only at time n_f, when the motion comes to a stop. As a result, the cost function can be simplified to

$$J = n_f + \lambda \left[\widetilde{V} - \frac{\mathbf{u}^T H(n_f)\mathbf{u}}{n_p}\right] + \mu^T \left\{\mathbf{x}_f - \mathbf{E}\left[\mathbf{x}(n_f)\right]\right\} \tag{25}$$

This produces the desired single vector for μ^T. In addition to reducing the number of multipliers required, a constant post-movement input can also simplify numeric computation of a time-optimal input — there is no need to calculate the post-movement input, as it is already known. However, a bit of equation manipulation is required to make use of this computational advantage.

With a fixed value for u_f, the input vector can be separated into movement and post-movement portions, so that $\mathbf{u} = \left[\mathbf{u}_m^T \; \mathbf{u}_{pm}^T\right]^T$, where $\mathbf{u}_m = \left[u_1 \; u_2 \; \ldots \; u_{n_f}\right]^T$ is an n_f-vector, and $\mathbf{u}_{pm} = \left[u_f \; u_f \; \ldots \; u_f\right]^T$ is an n_p-vector. Using this notation, the $Q(n)$ matrix of (15) and (19) can be segmented such that

$$V(n) = \left[\mathbf{u}_m^T \; \mathbf{u}_{pm}^T\right] \left[\begin{array}{c|c} F(n) & \mathbf{0} \\ \hline \mathbf{0} & P(n) \end{array}\right] \left[\mathbf{u}_m^T \; \mathbf{u}_{pm}^T\right]^T \tag{26}$$

where $F(n)$ is the $(n_f \times n_f)$ upper left hand submatrix of $Q(n)$, and $P(n)$ is the $(n_p \times n_p)$ lower right hand submatrix of the same. Expanding (26),

$$V(n) = \mathbf{u}_m^T \, F(n) \, \mathbf{u}_m + \mathbf{u}_{pm}^T \, P(n) \, \mathbf{u}_{pm} \tag{27}$$

When $n \geq (n_f + 1)$, then $P(n) = \text{diag}\{\Gamma(n - n_f) \ldots \Gamma(1) \; \mathbf{0}\}$. Thus,

$$V(n) = \mathbf{u}_m^T \, F(n) \, \mathbf{u}_m + \sum_{i=1}^{n-n_f} \Gamma(i) {u_f}^2 \tag{28}$$

Substituting this result into the mean variance of (22),

$$\overline{V}_{pm}(n_f) = \frac{1}{n_p}\left[\mathbf{u}_m^T \, H_m(n_f) \, \mathbf{u}_m + H_{pm}(n_f)\, u_f^2\right] \tag{29}$$

where

$$H_m(n_f) = \sum_{i=n_f+1}^{n_f+n_p} F(i) \tag{30}$$

and

$$H_{pm}(n_f) = \sum_{i=n_f+1}^{n_f+n_p} \left\{ \sum_{j=1}^{i-n_f} \Gamma(j) \right\} \tag{31}$$

Inserting this result into the cost function of (25),

$$\begin{aligned} J = n_f + \lambda \left\{ \widetilde{V} - \frac{1}{n_p}\left[\mathbf{u}_m^T \, H_m(n_f) \, \mathbf{u}_m + H_{pm}(n_f)\, u_f^2\right] \right\} \\ + \mu^T \left\{\mathbf{x}_f - \mathbf{E}\left[\mathbf{x}(n_f)\right]\right\} \end{aligned} \tag{32}$$

An optimal value for this cost function can now be determined using the calculus of variations.

5. OPTIMAL SOLUTION

Two parameters influence cost J; the movement period input vector, $\mathbf{u}_m$, and the length of that vector, n_f. To determine the optimal values for these parameters, the first variation of cost is examined:

$$\Delta J = \frac{\partial J}{\partial n_f}\Delta n_f + \frac{\partial J}{\partial \mathbf{u}_m}\Delta \mathbf{u}_m \tag{33}$$

It is desired to have $\frac{\partial J}{\partial n_f}$ and $\frac{\partial J}{\partial \mathbf{u}_m}$ go to zero, so that any changes in n_f or $\mathbf{u}_m$ will produce no change in cost J.

5.1 Movement Length Variation

The partial derivative of cost with respect to movement duration is

$$\begin{aligned}\frac{\partial J}{\partial n_f} = 1 - \frac{\lambda}{n_p}\left(\mathbf{u}_m^T \frac{\partial H_m(n_f)}{\partial n_f}\mathbf{u}_m + \frac{\partial H_{pm}(n_f)}{\partial n_f}u_f^2\right)\\ - \mu^T\left(\frac{\partial \mathbf{E}\left[\mathbf{x}(n_f)\right]}{\partial n_f}\right)\end{aligned} \tag{34}$$

If integer value n_f increments upward by one, then the value of $H_m(n_f)$ increases to $H_m(n_f+1)$. Thus,

$$\frac{\partial H_m(n_f)}{\partial n_f} \approx H_m(n_f+1) - H_m(n_f) \tag{35}$$

Substituting (30) into the prior result,

$$\begin{aligned}\frac{\partial H_m(n_f)}{\partial n_f} &\approx \sum_{i=n_f+2}^{n_f+n_p+1} F(i) - \sum_{i=n_f+1}^{n_f+n_p} F(i)\\ &= F(n_f+n_p+1) - F(n_f+1)\end{aligned} \tag{36}$$

Taking a similar tack with the post-movement mean variance matrix of (31),

$$\frac{\partial H_{pm}(n_f)}{\partial n_f} \approx H_{pm}(n_f+1) - H_{pm}(n_f) = 0 \tag{37}$$

Since the expected terminal state is fixed at $\mathbf{E}\left[\mathbf{x}(n_f)\right] = \mathbf{x}_f$, the partial derivative $\frac{\partial \mathbf{E}[\mathbf{x}(n_f)]}{\partial n_f}$ is zero. Thus, (34) simplifies to

$$\frac{\partial J}{\partial n_f} = 1 - \frac{\lambda}{n_p}\left(\mathbf{u}_m^T\ F_{pm}(n_f)\ \mathbf{u}_m\right) \tag{38}$$

where

$$F_{pm}(n_f) = F(n_f{+}n_p{+}1) - F(n_f{+}1) \tag{39}$$

For an optimal solution, $\frac{\partial J}{\partial n_f}$, must be zero. Making this substitution in (38), it is possible to rearrange the equation to express a relationship, albeit indirect, between the movement duration, n_f, and optimal input, $\mathbf{u}_m$:

$$\frac{n_p}{\lambda} = \mathbf{u}_m^T\ F_{pm}(n_f)\ \mathbf{u}_m \tag{40}$$

Next, the influence of input vector variation on cost is examined.

5.2 Input Vector Variation

Utilizing the expected value of state vector $\mathbf{x}(n_f)$, the cost equation of (32) can be expanded as

$$\begin{aligned}J = n_f{+}\lambda\left\{\widetilde{V}{-}\frac{1}{n_p}\left[\mathbf{u}_m^T H_m(n_f)\,\mathbf{u}_m{+}H_{pm}(n_f)\,u_f^2\right]\right\}\\ + \mu^T\left\{\mathbf{x}_f - A^{n_f}\mathbf{x}_0 - \sum_{i=0}^{n_f-1} A^{n_f-1-i}Bu(i)\right\}\end{aligned} \tag{41}$$

Taking the partial derivative of cost with respect to the input vector,

$$\begin{aligned}\frac{\partial J}{\partial \mathbf{u}_m} = \frac{-\lambda}{n_p}\left(\mathbf{u}_m^T\left\{H_m(n_f) + H_m(n_f)^T\right\}\right)\\ - \mu^T\left(\frac{\partial}{\partial \mathbf{u}_m}\left\{\sum_{i=0}^{n_f-1} A^{n_f-1-i}Bu(i)\right\}\right)\end{aligned} \tag{42}$$

Relying on (16), create a $(4 \times n_f)$ matrix

$$S(n_f) = [\ \chi(n_f)\ \chi(n_f-1)\ \ldots\ \chi(1)\ ] \tag{43}$$

Then

$$\sum_{i=0}^{n_f-1} A^{n_f-1-i}Bu(i) = S(n_f)\mathbf{u}_m \tag{44}$$

and

$$\frac{\partial}{\partial \mathbf{u}_m}\left\{\sum_{i=0}^{n_f-1} A^{n_f-1-i}Bu(i)\right\} = \frac{\partial}{\partial \mathbf{u}_m}\left\{S(n_f)\mathbf{u}_m\right\} = S(n_f) \tag{45}$$

Since $H_m(n_f)$ is a diagonal matrix, $H_m(n_f) = H_m(n_f)^T$, and (42) becomes

$$\frac{\partial J}{\partial \mathbf{u}_m} = \frac{-\lambda}{n_p}\left\{2\mathbf{u}_m^T H_m(n_f)\right\} - \mu^T S(n_f) \tag{46}$$

To achieve optimality, set $\frac{\partial J}{\partial \mathbf{u}_m} = \mathbf{0}$. Then

$$\mathbf{u}_m^T H_m(n_f) = -\frac{1}{2}\left(\frac{n_p}{\lambda}\right)\mu^T S(n_f) \tag{47}$$

This optimal condition, along with that of (40), allows a solution to be developed. In preparation for that step, however, the invertibility of matrix $H_m(n_f)$ is first examined.

5.3 Invertibility of H_m

To guarantee invertibility of matrix $H_m(n_f)$, each of its diagonal elements must be non-zero. It can be seen from (30) that $H_m(n_f)$ is a summation of matrices $F(n_f+1)$ through $F(n_f{+}n_p)$. Each diagonal element of $F(n)$ is $\Gamma(j)$, where $j = n, n{-}1, \ldots, 1$. These values, in turn, comes from the (1,1) element of matrix product $\left[A^{j-1}B\right]\left[A^{j-1}B\right]^T$, as evident from (18). From (8), it can be seen that the B vector will have non-zero elements in the same locations as the Ψ vector. In other words, only the fourth element of B will be non-zero; let that value be represented as b_4. Allowing the (1,4) element of matrix A^{j-1} to be denoted by a_{14}, it can be shown that the (1,1) element of product $\left[A^{j-1}B\right]\left[A^{j-1}B\right]^T$ is $(a_{14}b_4)^2$. Thus, $\Gamma(j)$ is never negative. For it to be non-zero, the (1,4) element of A^{j-1} must be likewise non-zero. Unfortunately, discrete-time matrix A is not directly available; it must be evaluated using continuous-time dynamic matrix Φ and the conversion of (8). Using the Cayley-Hamilton theorem, however, is it possible to expand the definition for A^{j-1} into a linear combination of Φ^n, where $n \in \{0, 1, 2, 3\}$.

According to Cayley-Hamilton, any power $j \geq n$ of $(n \times n)$ matrix M can be expressed as a summation of lower powers of M, such that

$$M^j = \sum_{i=0}^{n-1} c_j M^n \tag{48}$$

Relying on the textbook equation for an exponential matrix, it is possible to expand the definition for A given in (8) as

$$A = e^{\Phi T} = I + T\Phi + \frac{T^2}{2!}\Phi^2 + \frac{T^3}{3!}\Phi^3 + \ldots \tag{49}$$

Since any power of Φ greater than three can be expressed as in (48), the $(j-1)$ power of A can be stated as

$$\begin{aligned} A^{j-1} &= \left(e^{\Phi T}\right)^{j-1} = e^{(j-1)\Phi T} \\ &= \psi_0 I_4 + \psi_1\Phi + \psi_2\Phi^2 + \psi_3\Phi^3 \end{aligned} \tag{50}$$

where each ψ_x value is a constant scalar coefficient. Examining the (1,4) element in each right hand side matrix,

$$\begin{aligned} I_{[1,4]} &= 0; & \Phi_{[1,4]} &= 0 \\ \Phi^2_{[1,4]} &= 0; & \Phi^3_{[1,4]} &= 1 \end{aligned} \tag{51}$$

Thus, for A^{j-1} to have a non-zero (1,4) element, coefficient ψ_3 must be non-zero. To determine specific coefficient values, diagonalize (50), such that

$$e^{(j-1)\Lambda T} = \psi_0 I_4 + \psi_1\Lambda + \psi_2\Lambda^2 + \psi_3\Lambda^3 \tag{52}$$

where $\Lambda = \operatorname{diag}(\lambda_1,\ \lambda_2,\ \lambda_3,\ \lambda_4)$. Note that diagonalization requires distinct eigenvalues. If $\psi_3 = 0$, then (52) is equivalent to

$$\underbrace{\begin{bmatrix} 1 & \lambda_1 & \lambda_1^2 \\ 1 & \lambda_2 & \lambda_2^2 \\ 1 & \lambda_3 & \lambda_3^2 \\ 1 & \lambda_4 & \lambda_4^2 \end{bmatrix}}_{L'} \underbrace{\begin{bmatrix} \psi_0 \\ \psi_1 \\ \psi_2 \end{bmatrix}}_{C'} = \underbrace{\begin{bmatrix} e^{(j-1)\lambda_1 T} \\ e^{(j-1)\lambda_2 T} \\ e^{(j-1)\lambda_3 T} \\ e^{(j-1)\lambda_4 T} \end{bmatrix}}_{E} \tag{53}$$

Given distinct eigenvalues, matrix L' will have full column rank, and there can be only one solution for C' — a solution that must exist to satisfy (50). That solution, of course, is that two rows of L' are identical, resulting in two rows of E being likewise identical. This, however, contradicts the assumption of distinct eigenvalues. Thus, ψ_3 cannot be zero if matrix A has distinct eigenvalues, and matrix H_m is therefore invertible. What if Φ has repeated eigenvalues? Using the Jordan form of Φ, it can be shown that H_m is always invertible when n_p is greater than the number of repeated eigenvalues.

5.4 Optimal Input

Having discussed the invertibility of $H_m(n_f)$, rearrange (47) to solve for $\mathbf{u}_m$:

$$\mathbf{u}_m^T = \left(\frac{n_p}{2}\right)\left(\frac{-\mu^T}{\lambda}\right) S(n_f) H_m(n_f)^{-1} \tag{54}$$

Joining (21) and (44) with the expected value of (10), the final position constraint is

$$\mathbf{x}_f = \mathbf{E}\left[\mathbf{x}(n_f)\right] = A^{n_f}\mathbf{x}_0 + S(n_f)\mathbf{u}_m \tag{55}$$

Taking the transpose,

$$\mathbf{x}_f^T = \mathbf{x}_0^T A^{n_f T} + \mathbf{u}_m^T S(n_f)^T \tag{56}$$

Substituting (54) into the previous result,

$$\mathbf{x}_f^T = \mathbf{x}_0^T A^{n_f T} - \left[\left(\frac{n_p\mu^T}{2\lambda}\right) S(n_f) H_m(n_f)^{-1}\right] S(n_f)^T \tag{57}$$

Rearranging to isolate the ratio between Lagrangian multipliers,

$$\left(\frac{\mu^T}{\lambda}\right) = \left(\frac{-2}{n_p}\right)\left[\mathbf{x}_f - A^{n_f}\mathbf{x}_0\right]^T \left[S(n_f) H_m(n_f)^{-1} S(n_f)^T\right] \tag{58}$$

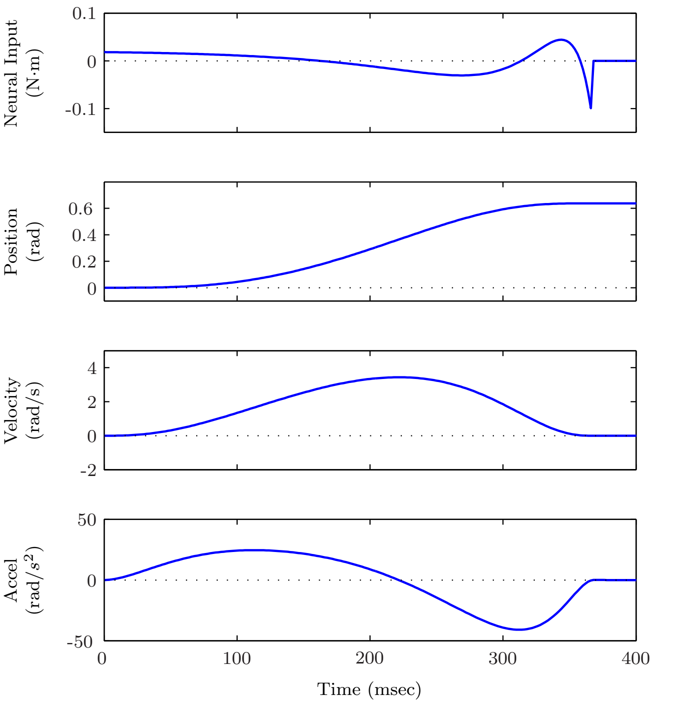

Fig. 2. Calculated open-loop, time-optimal input and associated forearm response. Rotation angle matches that of the experimental data shown in Fig. 1.

Dropping this result into (54), an optimal input is produced:

$$\mathbf{u}_m^T = \left[\mathbf{x}_f - A^{n_f}\mathbf{x}_0\right]^T \gamma(n_f)^{-1} S(n_f) H_m(n_f)^{-1} \tag{59}$$

where

$$\gamma(n_f) = S(n_f) H_m(n_f)^{-1} S(n_f)^T \tag{60}$$

A complete optimal input vector, $\mathbf{u}$, is produced by appending the terminal input, u_f, to movement vector $\mathbf{u}_m$ a total of n_p times, producing an optimal input vector of length $(n_f + n_p)$. Before continuing, though, the invertibility of $\gamma(n_f)$ needs to be considered.

5.5 Invertibility of $\gamma(n_f)$

Inversion of $\gamma(n_f)$ is possible when the matrix possesses full rank. To examine the conditions under which this occurs, take a singular value decomposition of real matrix $S(n_f)$, such that $S(n_f) = UDV^T$, where U and V are orthogonal matrices, and D is a $(4 \times n_f)$ diagonal matrix containing the singular values of $S(n_f)$. Then

$$\gamma(n_f) = UDV^T H_m(n_f)^{-1} V D^T U^T \tag{61}$$

Orthogonal matrices U and V are full rank, as is $H_m(n_f)$. Thus, by virtue of the matrix property that $\operatorname{rank}(AB) \le \operatorname{rank}(A) \cdot \operatorname{rank}(B)$, it is necessary that rectangular matrix D have a full row rank of $r = 4$ for $\gamma(n_f)$ to be invertible. As a result, all the singular values of $S(n_f)$ must be nonzero. Recalling that the number of non-zero singular values is indicated by matrix rank, it is necessary that $S(n_f)$ display full row rank. Since $S(n_f)$ has four rows, this is equivalent to requiring that the matrix possess at least four independent columns.

By the definition of (43), the four rightmost columns of $S(n_f)$, where $n_f \ge 4$, are $\chi(4), \chi(3), \chi(2)$, and $\chi(1)$.

Each χ value, as set forth in (16), can be expressed in terms of the four ψ_x values from (50), as well as the dynamic system coefficients from (4). Converting the four indicated χ vectors to their reduced-row form reveals that they are linearly independent when ψ_3, α_1, α_2, α_3, and β are non-zero, as is necessary for the system dynamics to be relevant. When $S(n_f)$ has fewer than four columns, it retains full rank. Hence, $\gamma(n_f)$ can always be inverted, as long as $H_m(n_f)$ remains invertible (as discussed previously).

5.6 Optimal Movement Duration

Substituting (59) and (60) into the average variance of (29), while dropping the (n_f) notation momentarily,

$$\begin{aligned}\overline{V}_{pm} n_p = &\\ &\left\{[\mathbf{x}_f - A^{n_f}\mathbf{x}_0]^T \left(SH_m{}^{-1}S^T\right)^{-1} SH_m^{-1}\right\} H_m \\ &\left\{H_m^{-1}S^T \left\{\left(SH_m^{-1}S^T\right)^{-1}\right\}^T [\mathbf{x}_f - A^{n_f}\mathbf{x}_0]\right\} \\ &+ H_{pm}\, u_f^2\end{aligned} \tag{62}$$

Using the definition for $\gamma(n_f)$ given in (60), simplify as

$$\overline{V}_{pm}(n_f) n_p = [\mathbf{x}_f - A^{n_f}\mathbf{x}_0]^T \gamma(n_f)\, [\mathbf{x}_f - A^{n_f}\mathbf{x}_0] + H_{pm}(n_f)\, u_f^2 \tag{63}$$

As noted in Appendix B of Tanaka et al. [2006], the initial state is $x_0 = [\theta_0\ 0\ 0\ 0]^T$ for point-to-point movement, so

$$\mathbf{x}_f - A^{n_f}\mathbf{x}_0 = \theta_f - \theta_i = \Theta \tag{64}$$

When $\alpha_0 = 0$, the terminal input is zero, and

$$\overline{V}_{pm}(n_f) n_p = \gamma(n_f)_{[1,1]}\Theta^2 \tag{65}$$

This equation is slightly easier to solve than (63), but is only valid for point-to-point motion. The more general form of (63) permits the calculation of remaining movement duration for a motion already in progress.

6. IMPLEMENTATION

A shortcoming in the application of (63) or (65) is the inability to solve directly for movement duration. It is necessary to iteratively solve for $\overline{V}_{pm}(n_f)$, first assuming some value for n_f, then comparing the resultant average variance to the established variance bound. Mathematically,

$$n_f = \arg\min_{n_f} |\widetilde{V} - \overline{V}_{pm}(n_f)| \tag{66}$$

An appropriate value for $\widetilde{V}$ can be determined by setting an acceptable probability that the final position will fall inside a desired target. Referencing a standard normal z-table, a value of $r = 1.96$ corresponds with a 95% chance that the forearm will land inside the desired target width, W. Therefore, the desired variance bound is set as

$$\widetilde{V} = W/r^2 \tag{67}$$

All matrices needed to compute an optimal duration can be generated and stored as Matlab variables prior to mesh generation, as all matrices are functions of movement time, n_f, and not a particular initial state. Computation of (66) is then rapidly executed for each movement configuration by selecting the proper matrices for each evaluation of a potentially optimal n_f. Since memory and performance concerns require an upper limit on the number of stored matrices, it may be necessary to adjust T to cover the entire range of movement durations. Point-to-point forearm rotation resulting from a calculated input is shown in Figure 2. In comparing these results with Fig. 1, it is evident that the velocity and acceleration profiles are characteristic of human motion.

Using a personal computer capable of 550 MFLOPs, the algorithm described above made six passes through nearly a half-million different movement configurations, increasing T by a factor of four with each pass, and calculated movement durations ranging from 2 ms to 512 s, with an upper limit of 250 on n_f. This computation was accomplished in 52 minutes, at an overall average of 146.6 solutions per second. Although larger matrices are associated with increasing values of n_f, individual evaluations of any particular movement duration consistently took less than one millisecond, will little change due to matrix dimension.

7. CONCLUSION

Rapid computation of second submovement durations allows a fine mesh to be searched for minimal response time. Work is underway to describe the computation of an initial submovement, as well as the timing of a single feedback instant to produce an overall time-optimal movement.

REFERENCES

J. F. Feng, G. Tartaglia, and B. Tirozzi. A note on minimum-variance theory and beyond. *Journal Of Physics A-Mathematical And General*, 37(17):4685–4699, April 2004.

Alon Fishbach, Stephane Roy, Christina Bastianen, Lee Miller, and James Houk. Deciding when and how to correct a movement: discrete submovements as a decision making process. *Experimental Brain Research*, 177(1):45–63, February 2007.

Paul M Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47:381–391, 1954.

Gene F. Franklin, Michael L. Workman, and Dave Powell. *Digital Control of Dynamic Systems*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1997.

Christopher M Harris and Daniel M. Wolpert. Signal-dependent noise determines motor planning. *Nature*, 394:780–784, 1998.

Michael A. Khan, Ian M. Franks, Digby Elliott, Gavin P. Lawrence, Romeo Chua, Pierre-Michel Bernier, Steve Hansen, and Daniel J. Weeks. Inferring online and offline processing of visual feedback in target-directed movements from kinematic data. *Neuroscience & Biobehavioral Reviews*, 30(8):1106–1121, 2006.

Robert F. Stengel. *Optimal Control and Estimation*. Dover, 1994.

Hirokazu Tanaka, John W. Krakauer, and Ning Qian. An Optimization Principle for Determining Movement Duration. *J Neurophysiol*, 95(6):3875–3886, 2006.

Frans C.T. van der Helm and Leonard A. Rozendaal. *Biomechanics and Neural Control of Posture and Movement*, chapter Musculoskeletal Systems with Intrinsic and Proprioceptive Feedback, pages 164 – 176. Springer-Verlang, 2000.