Stochastic Optimal Control and Kalman Filtering: Practical Example

F. Crevecoeur

August 6, 2018

This document will help you through the basic steps towards building a simple control model. Then, you will be encouraged to add complexity in the model in order to take several features of the sensorimotor system into account. The suggested starting point is a model of a rigid segment rotating around an axis aligned with one extremity, which can be used to model a single joint reaching movement. We will first consider the fully observable case (i.e. perfect state information) for simplicity. It will be a starting point to elaborate more complex models.

1 Model Formulation

The very first step is to define the model. Remember that we need a model of a control system in the following form:

$$x_{k+1} = Ax_k + Bu_k. \tag{1}$$

We must define the state vector and the model parameters. To begin, we must identify the differential equation that governs the motion of the system. For mechanical systems, Newton's laws usually allows us deriving such differential equation. The second low reads (sum of forces equals mass $\times$ acceleration):

$$\sum \vec{F} = m\vec{a}. \tag{2}$$

$$\tag{3}$$

For a rotating segment the analog in rotation is (sum of torque loads equals inertia $\times$ angular acceleration, noted $\vec{a}_R$):

$$\sum \vec{T} = I\vec{a}_R. \tag{4}$$

We call θ the angle as a function of time. There is one degree of freedom so we will drop the vector notation and work with a scalar ordinary differential equation (ODE). Thus, the angular acceleration is the second derivative of θ. We will consider two sources of torque loads (modelling choice, to be discussed !): a dissipative torque opposite and proportional to velocity with constant $G > 0$, plus the actuator torque that is generated by the controller (T). Equation 4 becomes:

$$I\ddot{\theta} = -G\dot{\theta} + T. \tag{5}$$

We will add a muscle model which transforms the command (representing the neural signal) into joint torque. In agreement with physiological studies (Li, 2007), we use a first order filter with time constant $\tau = 60$ms. The linear differential equation for the muscle model is then:

$$\tau\dot{T} = u - T, \tag{6}$$

where u represents the input to the system. Equations 5 and 6 capture the system dynamics. Since the system model involves the derivative of T and the second derivative of θ, we can define the following state vector: $x = [\theta, \dot{\theta}, T]^T$. We rewrite the two equations as follows:

$$\begin{aligned}
\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{T} \end{bmatrix} &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & -G/I & 1/I \\ 0 & 0 & -1/\tau \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ T \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1/\tau \end{bmatrix} u & (7) \\
&\Leftrightarrow & \\
\dot{x} &= A_c x + B_c u, & (8)
\end{aligned}$$

where the subscript c stands for continuous time dynamics.

This representation is an algebraic form for the system of ODEs. We are almost done. Remember we would like a discrete time system in order to include stochastic disturbances. This can be done in various ways. A popular technique is to use explicit Euler integration over one time step (note that an exact form is given with matrix exponentials). This is based on first order Taylor expansion over time ($\mathcal{I}$ is the identity matrix):

$$\begin{aligned}
x(t+\delta t) &= x(t) + \delta t \dot{x}(t), & (9) \\
x(t+\delta t) &= x(t) + \delta t (A_c x(t) + B_c u(t)), & (10) \\
x(t+\delta t) &= (\mathcal{I} + \delta t A_c) x(t) + \delta t B_c u(t). & (11)
\end{aligned}$$

If we equate t with index k and consider that δt is one time step, we can define $A := \mathcal{I} + \delta t A_c$, and $B := \delta t B_c$, and we are done !

2 Cost Function

The framework of LQG allows minimising a cost of the form:

$$J(x,u) = x_N^T Q_N x_N + \sum_{k=1}^{N-1} \left(x_k^T Q_k x_k + u_k^T R u_k \right) \tag{12}$$

We could either map all target to 0, or if we want to do target jumps we can include a target vector consisting of desired (fixed) coordinates (x^*) and augment the state vector as follows: $z_k^T = [x_k^T, x_k^{*T}]$. Note that the state spaces matrices must be augmented to include the target state with dynamics $\dot{x}^* = 0$ for a static target.

Now observe that defining e_n as the n-th row of the identity matrix of the same size of x, and $Q = w[e_n; -e_n][e_n^T, -e_n^T]$, we have:

$$z_k^T Q z_k = w(x[n]_k - x[n]_k^*)^2, \tag{13}$$

where the notation ".$[n]$" designates the n-th entry of the corresponding vector. This technique can be used to define a cost function fo the general form:

$$J(x) = w_1(\theta_k - \theta_k^*)^2 + w_2(\dot{\theta}_k - \dot{\theta}_k^*)^2 + w_3(T_k - T_k^*)^2. \tag{14}$$

In the example we use $Q_N = w(x_N - x_N^*)^2$, $Q_k = 0$, $k < N$ and R a constant parameter (see script for numerical values).

3 Optimal Control

Now that we have a state space representation and a cost function, we can implement the backward recursion that gives the series of optimal control gains. Recall that the recurrence is as follows:

$$S_N = Q_N, \qquad s_N = 0, \tag{15}$$

$$L_k = \left(R + B^T S_{k+1} B \right)^{-1} B^T S_{k+1} A, \tag{16}$$

$$S_k = Q_k + A^T S_{k+1}(A - BL_k), \tag{17}$$

$$s_k = s_{k+1} + tr(S_{k+1} + \Omega_\xi). \tag{18}$$

We ca then simulate the system forward. Since the control signal is defined as $u_k = -L_k x_k$, the dynamics becomes:

$$x_{k+1} = (A - BL_k)x_k + \text{noise}. \tag{19}$$

The script generates a plot of $L[1]_k$ and $L[2]_k$. How can you interpret these functions?

4 State Estimation

We now consider that we have imperfect state information. The linear model for this is a noisy mixture of state measurements:

$$y_k = Hx_k + \omega_k,$$

with $\omega_k \sim \mathcal{N}(0, \Omega_\omega)$. There are variants of optimal estimation that fall under minimum-variance linear estimation (Bayesian or predictive cases are very similar). In the predictive case we derived the following recurrence:

$$\begin{aligned} \Sigma_1, \hat{x}_1 & \quad \text{given} & (20) \\ K_k &= A\Sigma_k H^T \left(H\Sigma_K H^T + \Omega_\omega \right), & (21) \\ \Sigma_{k+1} &= \Omega_\xi + (A - K_k H)\Sigma_k A^T & (22) \\ \hat{x}_{k+1} &= A\hat{x}_k + Bu_k + K_k \left(y_k - H\hat{x}_k \right). & (23) \end{aligned}$$

The script implements the recurrences above with numerical values. It is generally very informative to look at all the signals that you can imagine. What is the estimation error? What is the posterior variance? These signals can be easily extracted.

5 Further Assignments

1. Add signal dependent noise for fully observable case (*)
2. Implement signal dependent noise with Kalman filtering (numerical approximation based on iteration: Todorov, 2005) (***)
3. Augment the state vector to include a feedback delay (**)
4. Think about how to simulate perturbation paradigms (**)
5. Implement an optimal control model of saccadic eye movement in closed form with 100ms delay, or with a state estimator corresponding to $\hat{x}_{k+1} = A\hat{x}_k + Bu_k$. What do you observe? (***)