Chapter 7 HW 2 Hints

Problem 1. In this problem you should (a) find the system equations like (7.45), but also (b) find the state-space (*i.e.* state and output) equations for the *controller*. Assume that matrix $\mathbf{C}$ produces output y, and that matrix $\mathbf{C}_m$ produces the measurement y_m used by the estimator.

(a) To find the combined system state equation of plant + estimator + control law, use the plant state equation (7.33) and the prediction estimator equation (7.36) (with $y = y_m$ and $\mathbf{C} = \mathbf{C}_m$) along with the control law based on the estimated state:

$$u(k) = -\mathbf{K}\hat{\mathbf{x}}(k).$$

A partial result is

$$\begin{bmatrix} \mathbf{x}(k+1) \\ \hat{\mathbf{x}}(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi} & ?? \\ \mathbf{L}\mathbf{C}_m & ? - \mathbf{\Gamma}? - ?\mathbf{C}_m \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \hat{\mathbf{x}}(k) \end{bmatrix}$$

where each "?" represents one matrix term. This state equation can be used to model the controlled system's behavior from an initial condition.

(b) To find a state-variable model of just the controller (control law + estimator) just use the estimator update equation (7.36) and the control law using the estimated state (don't need the plant model). Now you will have a state equation *and* an output equation. This is what you need to develop a Simulink model using the state-space controller. The state and output equations will be of the form

$$\begin{aligned} \hat{\mathbf{x}}(k+1) &= \left[\mathbf{\Phi} - \mathbf{\Gamma}? - ?\mathbf{C}_m\right] \hat{\mathbf{x}}(k) + \mathbf{L} y_m(k) \\ u(k) &= \left[?\right] \hat{\mathbf{x}}(k) + \left[?\right] y_m(k) \end{aligned}$$

where each "?" is one term. The controller is of order n.

Problem 2. I have expanded this problem into three parts.

(a) The problem statement said the measurement was $y = \theta_l$, the load position. However, the units were not stated, but are *degrees*. This should be reflected in your choice for the $\mathbf{C}_m$ matrix for the estimator design.

I suggest picking the poles for the estimator error to be two or three times as fast as the "control law" poles from HW 1. That is, if you had picked $\omega_n = 100$ rad/s for the control law, perhaps pick $\omega_n = 250$ rad/s for the estimator.

Solving for estimator gain matrix $\mathbf{L}$ should be easy using MATLAB `place` (remember to transpose the appropriate matrices in accordance with the notes). The estimator is generating both position and velocity from a measurement of position.

To demonstrate the effectiveness of your estimator you should combine the control law and estimator in a form like (7.45) or Problem 1 above, and start the system with some initial state $\mathbf{x}(0)$ and *different* initial estimated state $\hat{\mathbf{x}}(0)$ (or a nonzero state error $\tilde{\mathbf{x}}(0)$ if using (7.45)) and find the time history of the state and the estimated state (or the state error). The estimator error should go to zero fairly quickly.

(b) Can you design an estimator in which the measurement is load velocity (rpm or deg/s—units don't matter)? What if the measurement is motor current(A)? Now the estimator is estimating current, position, and velocity from a measurement of velocity or current. *Hint:* Check the rank of the observability matrix (done with MATLAB `ctrb(Phi',C')`).

(c) (Extra Credit) Add some noise to the measurement y_m and see how well the estimator of part (a) does. This is probably easiest using a Simulink model of the system, although it *could* be done within MATLAB. If you do this in Simulink, use a continuous-time block for the plant model (motor/load) since it really is a continuous system. Use a "state-space" Simulink block for the control law + estimator. You will need to use Simulink Zero-order hold blocks before the plant and before the estimator to model the sampling.