MASSACHUSETTS INSTITUTE OF TECHNOLOGY
Department of Electrical Engineering and Computer Science

6.3100: Intro to Modeling and Control—Spring 2025

Fast and Spurious - Due 10:00am, 02/21/25

Estimating Speed

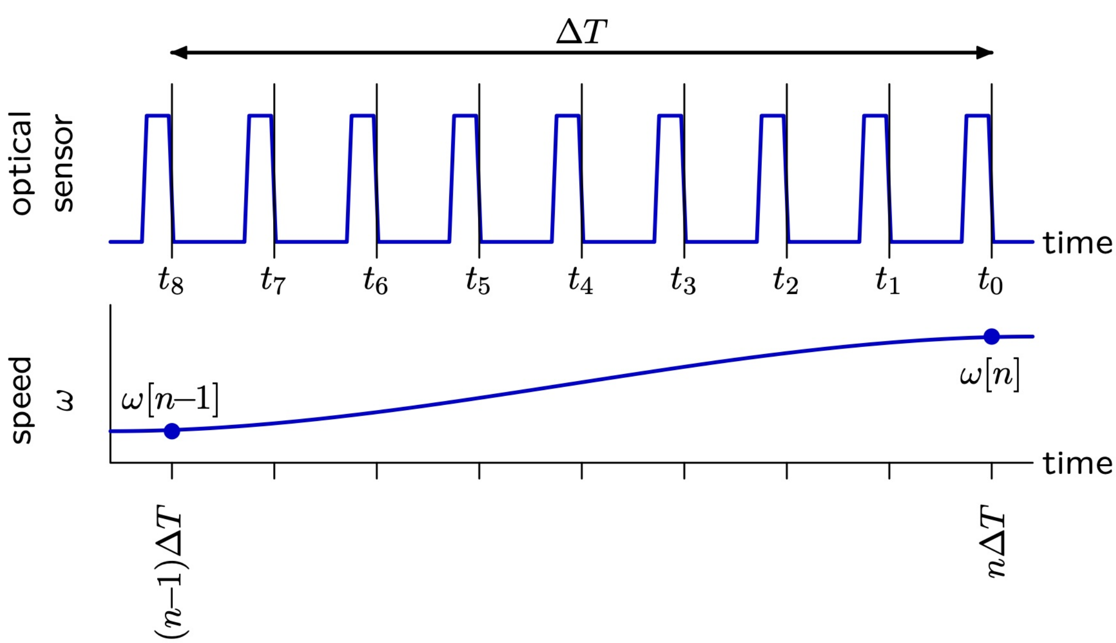
The goal of this Post-Lab is to analyze and understand the origin of the inconsistency you should have seen in lab. The major factor that contributes to this inconsistency stems from the way we measure rotor speed.

In Lab 1 (Fast and Spurious), we worked on motor speed control where the system is described by a first-order model:

$$\omega[n] = \omega[n-1] + \Delta T \left(\gamma c[n-1] - \beta\omega[n-1]\right)$$

In our model, ω represents the current value of the angular velocity of the rotor. However, our measurement hardware does not measure ω directly. Instead, ω is estimated by using an optical sensor to measure the time between pulses of light reflected from an eight-bladed rotor.

The following figure illustrates our method. When `ticksperupdate=8`, the control loop runs once after each set of 8 pulses. If we label the current time as t_0 and previous times as t_1, t_2, $\cdots$, then the sample time $\Delta T = t_0 - t_8$.



Let $\widetilde{\omega}$ represent our estimate of angular speed based on the experimentally determined times t_i. If `ticksperestimate=8`, the time for 8 pulses to occur is equal to the time for one turn of the rotor (since there are 8 blades on the rotor). Therefore the angular speed is approximately 1 revolution divided by the time for 8 pulses of light:

$$\widetilde{\omega}[n] \approx \frac{1}{t_0 - t_8} \text{ revolutions per second}$$

Notice however that this estimate matches the true speed ω best at a point midway between the sample times $t_0 = n\Delta T$ and $t_8 = (n{-}1)\Delta T$ – i.e., at t_4. If we make a piecewise linear approximation of speed, then

$$\widetilde{\omega}[n] \approx \frac{\omega[n] + \omega[n{-}1]}{2}$$

In our original formulation of the model, the control signal $c[n]$ was proportional to the difference between $\omega_d[n]$ and $\omega[n]$:

$$c[n] = K_p(\omega_d[n] - \omega[n])$$

However, the controller does not have direct access to $\omega[n]$, and a better model for our hardware is

$$c[n] = K_p(\omega_d[n] - \widetilde{\omega}[n]) = K_p\left(\omega_d[n] - \frac{\omega[n] + \omega[n-1]}{2}\right).$$

Problem One: The difference equations.

1A: Assuming $c[n] = K_p(\omega_d[n] - \omega[n])$, write the first-order difference equation for $\omega[n]$ in terms of $\Delta T, K_p, \beta, \gamma, \omega[n-1]$, and $\omega_d[n-1]$.

1B: Assuming $c[n] = K_p\left(\omega_d[n] - \frac{\omega[n]+\omega[n-1]}{2}\right)$, write the second-order difference equation for $\omega[n]$ in terms of $\Delta T, K_p, \beta, \gamma, \omega[n-1], \omega[n-2]$ and $\omega_d[n-1]$.

1C: You can rewrite your answer to 1B in matrix form,

$$\begin{bmatrix} \omega[n] \\ \omega[n-1] \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} \omega[n-1] \\ \omega[n-2] \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \omega_d[n-1].$$

What are the values of $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ and $B = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$ in terms of $\Delta T, K_p, \beta$, and γ.

Problem Two: Explaining the data.

Based on measurements, we calculated the system parameters β and γ. However, in Checkoff 5, we collected data and observed an inconsistency between the model in 1A above and the measurement. Demonstrate the inconsistency by plotting speed as a function of time under the conditions described below.

2A: For your setup, set `REPEATS`=3, `ticksperupdate`=8, `FREQ`=0.2, AMP=1, `ticksperestimate`=8, and `disturbAmp`=0. Then find a value for K_p so that the falling ω transition oscillates (or *rings*) three or four times before settling. Copy and past or redraw the result below.

2B: Use the natural frequencies, or the λ, of the model in 1A (using your measured β and γ from lab) to explain how your data is inconsistent with the first order model of 1A.

2C: Are the natural frequencies of the model in 1C, or the $\lambda's$ (which are the eigenvalues of the matrix A), more consistent with the data you collected. Please use your favorite software package to compute matrix eigenvalues (e.g. eig(A) in matlab), life is too short to compute eigenvalues by hand, even for 2x2 matrices.

Problem Three: Reducing the measurement delay.

In the previous section, we estimated the rotor speed by computing the time for the rotor to spin one full term, i.e., $t_0 - t_8$ in the above figure. That method results in an estimate of ω that is, sort of, delayed by half a rotation. More precisely, our system is better described by a second-order difference equation, as we noted in question 2C. In this problem, we investigate using an estimate based on a single pulse period, to see if restricting the estimate to use more recent information yeilds behavior that is more similar to our orginal first-order model.

If `tickesperestimate=1`, our hardware computes a different estimate of rotor speed:

$$\widehat{\omega}[n] = \frac{1/8}{t_0 - t_1}$$

where we scaled the measurement by 1/8 because only one-eighth of a rotation is completed between t_1 to t_0.

As with $\widetilde{\omega}[n]$ above, we can relate the estimated speed used by the controller to the actual speed $\omega[n]$ by linear interpolation:

$$\widehat{\omega}[n] = a\omega[n] + b\omega[n-1].$$

Note the the constants a and b are NOT equal to $\frac{1}{2}$.

Part 3A: What are the numerical values of a and b needed for $\widehat{\omega}$? Briefly explain your reasoning.

Part 3B: Using your values of a and b in the formula for $\widehat{\omega}[n]$, update the values of $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ and $B = \begin{bmatrix} b_1 \\ db_2 \end{bmatrix}$ in terms of $a, b, \Delta T, K_p, \beta$, and γ.

Part 3C: For your setup, set `REPEATS`=3, `ticksperupdate`=8, `FREQ`=0.2, AMP=1, and `tickesperestimate`=1, and `disturbAmp`=0 (note that we only `ticksperestimate` is different from the experiment in 2A). Then find a new value of K_p so that the falling ω transition oscillates (*rings*) three or four times before settling. Copy and past or redraw the result below.

Part 3D: Explain the behavior in 3C using natural frequencies (e.g. the eigenvalues of A). Be sure to explain why you needed a LARGER K_p to achieve the same amount of ringing. Why does the behavior appear to be "more first order"?

Problem Four: (challenging) What value of `ticksperestimate` gives you the best stability (requires the largest K_p to get three oscillations to settle after a falling transition). Does your second-order model agree with that prediction? Explain your answer (again, make use of some eigenvalue computing software!)