# Fast and Spurious: Post-Lab – Solutions

## Problems

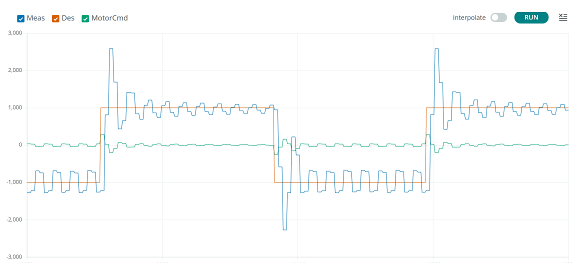
### 1. Latent Inconsistency

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

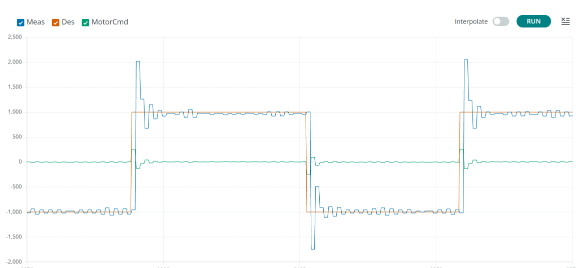
$$\omega[n] = \omega[n-1] + \Delta T \Big( \beta\omega[n-1] + \gamma c[n-1] \Big)$$

Based on measurements, we calculated the system parameters $\beta$ and $\gamma$. However, in Checkoff 5, we collected data and observed an inconsistency between the model and the measurement. Demonstrate the inconsistency by plotting speed as a function of time under the following two conditions:

**1a.** $K_p$ is set to its largest stable value, `REPEATS`=3, `ticksperupdate`=8, `FREQ`=0.05, AMP=1, and `ticksperestimate`=8, `disturbA`=0.




**1b.** $K_p$ is set to its largest stable value, `REPEATS`=3, `ticksperupdate`=8, `FREQ`=0.05, `AMP`=1, and `ticksperestimate`=1, `disturbA`=0.



**1c.** Describe the inconsistency that is demonstrated by the previous plots.
**Hint:** You should have discussed this point with the TAs during Checkoff 5.

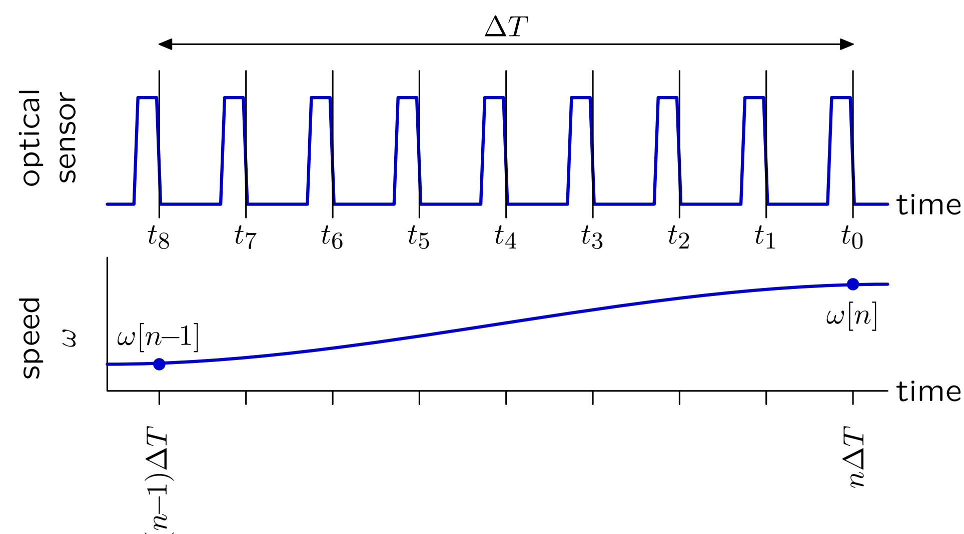
When `tickssperestimate=8`, the system does not show the pointwise oscillation that we expect of a first-order system. Instead, the oscillation involves 2 sample points on each side. So the frequency is approximately 3.75 Hz. This type of oscillation can occur for second-order (and higher-order) systems, but not for first-order systems.

When we change `tickssperestimate` to 1, we see pointwise oscillation, and the system behaves more like a first-order system. The oscillation also decays faster, suggesting a change in the natural frequency $\lambda$.

## 2. Estimating Speed

The goal of this Post-Lab is to analyze and understand the origin of the inconsistency that was seen in the previous part. The major factor that contributes to this inconsistency stems from the way we measure rotor speed. In the model, $\omega$ represents the current value of the angular velocity of the rotor. However, our experiment hardware does not provide a direct measurement of $\omega$. Instead, we must estimate the angular velocity from the time between pulses of light detected by the optical sensor.

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 `tickssperestimate=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 $\omega$ 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 $\omega[n]$:

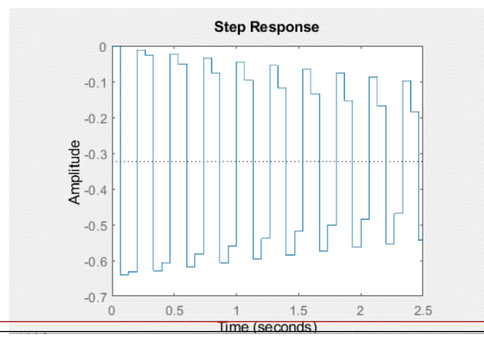
$$c[n] = K_p(\omega_d[n] - \omega[n])$$

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

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

Simulate a new model for the experiment hardware using this controller instead of our original controller. Use the measured parameters $\beta$ and $\gamma$ you obtained from Lab 1, and suppose $\omega_d[n] = 1$ and $\omega[0] = 0$.

**2a.** Provide a plot of $\omega[n]$ as a function of $n$ using the programming language of your choice (e.g., MATLAB, Python, ...).




**2b.** Briefly describe the important features of your plot, and how they relate to the inconsistency that was described in part 1 of this post-lab.

The important result from this analysis is that the new model correctly predicts that $\omega[n]$ does not exhibit pointwise oscillation. Because of the way we estimate angular speed of the rotor, we have introduced additional delay: $\widetilde{\omega}[n]$ depends on both $\omega[n]$ and $\omega[n-1]$, so the system is actually second-order not first-order.

## 3. Estimating Speed With Less 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 previous figure. This method introduces significant delay in the feedback loop, and makes the system behave more like a second-order system than as a first-order system. In this section, we investigate an estimate based on a single pulse period to see if restricting the estimate to more recent times can better preserve the first-order nature of the hardware that we expected.

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

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

The 1/8 in the numerator results because the rotor rotates just one-eighth of a turn in the time from $t_1$ to $t_0$. As with $\widetilde{\omega}[n]$ we can relate the estimated speed to the model speed $\omega[n]$ by linear interpolation:

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

However, the constants $a$ and $b$ are no longer $\frac{1}{2}$.

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

$a = \frac{15}{16}$. $b = \frac{1}{16}$.

The new estimate $\widehat{\omega}[n]$ best estimates the speed at a point that is midway between $t_0$ and $t_1$. By linear interpolation, this estimate is approximately

$$\widehat{\omega}[n] \approx \frac{7.5\omega[n] + 0.5\omega[n-1]}{8} = \frac{15\omega[n] + \omega[n-1]}{16}$$

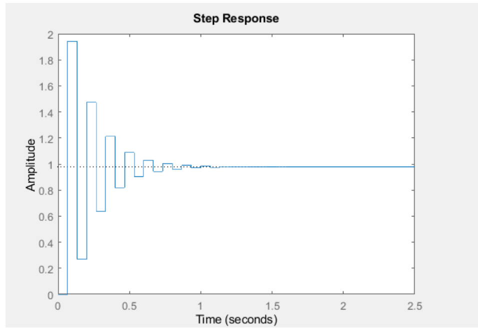
Determine the control system equation that results from these new values of $a$ and $b$.

**3b.** Enter the new system equation. Briefly show how you derived this equation by providing a few intermediate steps.

$$\begin{aligned}
\omega[n] &= \omega[n-1] + \Delta T(\beta\omega[n-1] + \gamma c[n-1]) \\
&= \omega[n-1] + \Delta T(\beta\omega[n-1] + \gamma K_p(\omega_d[n-1] - \omega[n-1])) \\
&= \omega[n-1] + \Delta T\beta\omega[n-1] + \Delta T\gamma K_p(\omega_d[n-1] - a\omega[n-1] - b\omega[n-2]) \\
&= \omega[n-1] + \Delta T\beta\omega[n-1] + \Delta T\gamma K_p\omega_d[n-1] - \Delta T\gamma K_p a\omega[n-1] - \Delta T\gamma K_p b\omega[n-2] \\
&= \omega[n-1](1 + \Delta T\beta - \Delta T\gamma K_p a) + \Delta T\gamma K_p\omega_d[n-1] - \Delta T\gamma K_p b\omega[n-2] \\
&= \omega[n-1](1 + \Delta T\beta - \Delta T\gamma K_p a) - \omega[n-2]\Delta T\gamma K_p b + \Delta T\gamma K_p\omega_d[n-1]
\end{aligned}$$

Write a program to compute $\omega[n]$ as a function of $n$ when $\omega_d[n] = 1$ and $\omega[0] = \omega[1] = 0$.

**3c.** Provide a plot of $\omega[n]$ as a function of $n$.




**3d.** Briefly describe the important features of your plot, and how they relate to the inconsistency that was described in part 1 of this post-lab.

When the velocity estimate is based on just one previous tick of the optical sensor, the response is much more consistent with the pointwise oscillation that we expect of a first-order system.

Comparing the two simulated results, which case shows faster convergence? More broadly, if we want our control system to convert the fastest, is there an optimal way for measuring the velocity $\widetilde{\omega}$? If we are free to choose $a$ and $b$, what are the optimal values?

Hint: you need to solve this problem numerically (i.e., sweep through different combinations of $a$ and $b$ and identify the optimal pair.)

Please show your analysis (possibly with graphs) below.

**3e.** What values of $a$ and $b$ provide the fastest convergence? Briefly explain.

In both simulation and experiments, we found faster sampling leads to faster convergence. However, the fastest convergence is NOT given by $a = 1$ and $b = 0$. The optimal value is given by the case where the pair of natural frequencies have the smallest magnitude. This can be calculated computationally. We sweep through different pairs of a and b (under the constraint $a + b = 1$), and plot the corresponding eigenvalues.