1. SVD I

Find the singular value decomposition of the following matrix (leave all work in exact form, not decimal):

\[ A = \begin{bmatrix} 2 & 2 \\ 3 & -3 \end{bmatrix} \]

(a) Find the eigenvalues of \( AA^\top \) and order them from largest to smallest, \( \lambda_1 > \lambda_2 \).

**Solution:**

\[ \lambda_1 = 18 \quad , \quad \lambda_2 = 8 \]

(b) Find orthonormal eigenvectors \( \vec{u}_i \) of \( AA^\top \) (all eigenvectors are mutually orthogonal and unit length).

**Solution:**

\[ \vec{u}_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad , \quad \vec{u}_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \]

(c) Find the singular values \( \sigma_i = \sqrt{\lambda_i} \). Find the \( \vec{v}_i \) vectors from:

\[ A^\top \vec{u}_i = \sigma_i \vec{v}_i \]

**Solution:**

\[ \sigma_1 = 3\sqrt{2} \quad , \quad \sigma_2 = 2\sqrt{2} \]

\[ \vec{v}_1 = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix} \quad , \quad \vec{v}_2 = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \]

(d) Write out \( A \) as a weighted sum of rank-1 matrices:

\[ A = \sigma_1 \vec{u}_1 \vec{v}_1^\top + \sigma_2 \vec{u}_2 \vec{v}_2^\top \]

**Solution:**

\[ A = 3\sqrt{2} \begin{bmatrix} 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 0 & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} + 2\sqrt{2} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 0 \end{bmatrix} \]
(e) Use \texttt{numpy.linalg.svd} to compute the SVD of $A$ and compare it to your results for the previous part. Will calculating the SVD by hand and calculating it using \texttt{numpy} always return identical results (up to floating point error)? Why or why not?

Note: Be sure to carefully read the documentation for the \texttt{svd} function. In particular, pay attention to the format of the returned values.

Solution: See ipython notebook

2. SVD II

Find the singular value decomposition of the following matrix (leave all work in exact form, not decimal):

$$A = \begin{bmatrix}
1 & 0 & -\sqrt{3} \\
\sqrt{3} & 0 & 1 \\
0 & 3 & 0
\end{bmatrix}$$

(a) Find the eigenvalues of $A^\top A$ and order them from largest to smallest, $\lambda_1 > \lambda_2$.

Solution:

$$\lambda_1 = 9 \quad , \quad \lambda_2 = 4 \quad , \quad \lambda_3 = 4$$

(b) Find orthonormal eigenvectors $\vec{v}_i$ of $A^\top A$ (all eigenvectors are mutually orthogonal and unit length).

Solution:

$$\vec{v}_1 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

Since $\lambda_2 = \lambda_3$, any two mutually orthogonal unit vectors that are also orthogonal to $\vec{v}_1 = [0, 1, 0]^\top$ will work. For example:

$$\vec{v}_2 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \vec{v}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

(c) Find the singular values $\sigma_i = \sqrt{\lambda_i}$. Find the $\vec{u}_i$ vectors from:

$$A\vec{v}_i = \sigma_i\vec{u}_i$$

Solution:

$$\sigma_1 = 3 \quad , \quad \sigma_2 = 2 \quad , \quad \sigma_3 = 2$$

$$\vec{u}_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad \vec{u}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ \sqrt{3} \\ 0 \end{bmatrix}, \quad \vec{u}_3 = \frac{1}{\sqrt{2}} \begin{bmatrix} -\sqrt{3} \\ 0 \\ 1 \end{bmatrix}$$
(d) Write out $A$ as a weighted sum of rank-1 matrices:

$$A = \sigma_1 \vec{u}_1 \vec{v}_1^\top + \sigma_2 \vec{u}_2 \vec{v}_2^\top + \sigma_3 \vec{u}_3 \vec{v}_3^\top$$

**Solution:**

$$A = 3 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} + 2 \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + 2 \begin{bmatrix} \frac{-\sqrt{3}}{2} & 0 & 0 \\ \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

3. Balance

Justin is working on a small jumping robot named Salto. Salto can bounce around on the ground, but Justin would like Salto to balance on its toe and stand still. In this problem, we’ll work on systems that could help Salto balance on its toe using its reaction wheel tail.

![Figure 1: Picture of Salto and the x-z physics model. You can watch a video of Salto here:](http://www.youtube.com/watch?v=2dJmArHRn0U)

Standing on the ground, Salto’s dynamics in the x-z plane (called the sagittal plane in biology) look like an inverted pendulum with a flywheel on the end:

$$(I_1 + (m_1 + m_2)l^2)\ddot{\theta}_1 = -K_t u + (m_1 + m_2)l g \sin(\theta_1)$$

$$I_2 \ddot{\theta}_2 = K_t u$$

Where $\theta_1$ is the angle of the robot’s body relative to the ground (0 is straight up), $\dot{\theta}_1$ is its angular velocity, $\dot{\theta}_2$ is the angular velocity of the reaction wheel tail, and $u$ is the current input to the tail motor. $m_1, m_2, I_1, I_2, l, K_t$ are positive constants representing system parameters (masses and angular momentums of the body and tail, leg length, and motor torque constant respectively) and $g = 9.81 \text{m/s}^2$ is the acceleration due to gravity.

Numerically substituting Salto’s physical parameters, the differential equations become approximately:

$$0.001 \ddot{\theta}_1 = -0.025 u + 0.1 \sin(\theta_1)$$

$$5(10^{-5}) \ddot{\theta}_2 = 0.025 u$$
For this problem, we’ll look at a reduced suite of sensors on Salto. Our only output will be the tail encoder that measures the angular velocity of the tail relative to the body:

\[ y = \dot{\theta}_2 - \dot{\theta}_1 \]

(a) Using the state vector \([\theta_1, \dot{\theta}_1, \dot{\theta}_2]^\top\), input \(u\), and output \(y\) linearize the system about the point \([0, 0, 0]^\top\). Write the linearized equations as \(\frac{d}{dt}\vec{x} = A\vec{x} + Bu\) and \(y = C\vec{x}\). Write the matrices with the physical numerical values, not symbolically.

Note: since the tail is like a wheel, we care only about its angular velocity \(\dot{\theta}_2\) and not its angle \(\theta_2\).

**Solution:** Numerically, the dynamics are:

\[
\begin{bmatrix}
\dot{\theta}_1 \\
\dot{\theta}_1 \\
\dot{\theta}_2
\end{bmatrix} = 
\begin{bmatrix}
0 & 1 & 0 \\
100 & 0 & 0 \\
0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\theta_1 \\
\dot{\theta}_1 \\
\dot{\theta}_2
\end{bmatrix} + 
\begin{bmatrix}
0 \\
-25 \\
500
\end{bmatrix} u
\]

\[ y = \begin{bmatrix} 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \]

For those interested, the symbolic dynamics are:

\[
\begin{bmatrix}
\dot{\theta}_1 \\
\dot{\theta}_1 \\
\dot{\theta}_2
\end{bmatrix} = 
\begin{bmatrix}
\frac{(m_1 + m_2)g \sin(\theta)}{I_1 + (m_1 + m_2)l^2} & 1 & 0 \\
\frac{10^5 l}{I_1 + (m_1 + m_2)l^2} & 0 & 0 \\
0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\theta_1 \\
\dot{\theta}_1 \\
\dot{\theta}_2
\end{bmatrix} + 
\begin{bmatrix}
0 \\
-\frac{K_t}{I_1 + (m_1 + m_2)l^2} \\
K_t
\end{bmatrix} u
\]

\[ y = \begin{bmatrix} 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \]

Note: text in red are solutions for if you solved before a typo was fixed

If you solved before the typo fix using:

\[ 0.001\dot{\theta}_1 = -0.025u + 100\sin(\theta_1) \]

Then the dynamics are:

\[
\begin{bmatrix}
\dot{\theta}_1 \\
\dot{\theta}_1 \\
\dot{\theta}_2
\end{bmatrix} = 
\begin{bmatrix}
0 & 1 & 0 \\
10^5 & 0 & 0 \\
0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\theta_1 \\
\dot{\theta}_1 \\
\dot{\theta}_2
\end{bmatrix} + 
\begin{bmatrix}
0 \\
-25 \\
500
\end{bmatrix} u
\]

\[ y = \begin{bmatrix} 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \]

(b) Is the system fully controllable? Is the system fully observable?

**Solution:**

\[
C = \begin{bmatrix} B & AB & A^2B \end{bmatrix} = 
\begin{bmatrix}
0 & -25 & 0 \\
-25 & 0 & -2500 \\
500 & 0 & 0
\end{bmatrix}
\]
which is full rank so the system is fully controllable.

\[ \mathcal{O} = \begin{bmatrix} C \\ CA \\ CA^2 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 1 \\ -100 & 0 & 0 \\ 0 & -100 & 0 \end{bmatrix} \]

which is rank 3 so the system is fully observable.

If you solved before the typo fix and used:

\[ 0.001 \dot{\theta}_1 = -0.025u + 100\sin(\theta_1) \]

Then you get:

\[ \mathcal{O} = \begin{bmatrix} B & AB & A^2B \end{bmatrix} = \begin{bmatrix} 0 & -25 & 0 \\ -25 & 0 & -25 \times 10^5 \\ 500 & 0 & 0 \end{bmatrix} \]

\[ \mathcal{O} = \begin{bmatrix} C \\ CA \\ CA^2 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 1 \\ -10^5 & 0 & 0 \\ 0 & -10^5 & 0 \end{bmatrix} \]

Both observability and controllability matrices are rank 3, so the system is controllable and observable.

(c) Design an observer of the form \( \frac{d\hat{x}}{dt} = (A + LC)\hat{e} \) and solve for the gains in L that make the observer dynamics converge with all eigenvalues \( \lambda_1 = \lambda_2 = \lambda_3 = -10 \).

**Solution:** The observer dynamics are dictated by

\[ \dot{\hat{e}} = (A + LC)\hat{e} \]

where \( \hat{e} \) is the error between the estimated state \( \hat{x} \) and the true state \( \bar{x} \). The characteristic polynomial is:

\[ \begin{vmatrix} A + LC \end{vmatrix} = 0 \]

\[ \begin{vmatrix} 0 & 1 + l_1 & -l_1 \\ 100 & l_2 & -l_2 \\ 0 & l_3 & -l_3 \end{vmatrix} = 0 \]

\[ \lambda^3 + (l_3 - l_2)\lambda^2 + (-100l_1 - 100)\lambda - 100l_3 = 0 \]

The desired characteristic polynomial is:

\[ (\lambda + 10)^3 = 0 \]

\[ \lambda^3 + 30\lambda^2 + 300\lambda + 1000 = 0 \]

which we can achieve by matching the coefficients of matching powers:

\[ l_3 - l_2 = 30 \]

\[ -100l_1 - 100 = 300 \]

\[ -100l_3 = 1000 \]
These equations are solved by the gains: $l_1 = -4, l_2 = -40$, and $l_3 = -10$. Written as a matrix,
\[
L = \begin{bmatrix} -4 & -40 & -10 \end{bmatrix}
\]
If you solved before the typo fix and set $\lambda_1 = \lambda_2 = \lambda_3 = 10$, then the characteristic polynomial is:
\[
(\lambda - 10)^3 = 0
\]
\[
\lambda^3 - 30\lambda^2 + 300\lambda - 1000 = 0
\]
which we can achieve by matching the coefficients of matching powers:
\[
l_3 - l_2 = -30
\]
\[
-100l_1 - 100 = 300
\]
\[
-100l_3 = -1000
\]
Solving the system of equations gives you:
\[
L = \begin{bmatrix} -4 & 40 & 10 \end{bmatrix}
\]
If you solved using:
\[
0.001\ddot{\theta}_1 = -0.025u + 100\sin(\theta_1)
\]
And set $\lambda_1 = \lambda_2 = \lambda_3 = -10$, then the characteristic polynomial for the error is:
\[
\begin{vmatrix} 0 & 1 + l_1 & -l_1 \\ 10^5 & l_2 & -l_2 \\ 0 & l_3 & -l_3 \end{vmatrix} = 0
\]
\[
\lambda^3 + (l_3 - l_2)\lambda^2 + (-10^5l_1 - 10^5)\lambda - 10^5l_3 = 0
\]
Matching coefficients:
\[
l_3 - l_2 = 30
\]
\[
-10^5l_1 - 10^5 = 300
\]
\[
-10^5l_3 = 1000
\]
Solving the system of equations gives you:
\[
L = \begin{bmatrix} -1.003 & -30.01 & -0.01 \end{bmatrix}
\]
If you solved using:
\[
0.001\ddot{\theta}_1 = -0.025u + 100\sin(\theta_1)
\]
And set $\lambda_1 = \lambda_2 = \lambda_3 = 10$, then you get:
\[
L = \begin{bmatrix} -1.003 & 30.01 & 0.01 \end{bmatrix}
\]
(d) Let’s implement a controller for our system using an analog electrical circuit! You can use the following circuit components in Figure 2:

Using state feedback, Justin has selected the control gains $\tilde{K} = [20 \ 5 \ 0.01]$. Draw a circuit in the box in Figure 3 that implements this controller. Use relatively reasonable component values.

*Optional bonus: what are the eigenvalues of the closed loop dynamics for the given $K$?*

**Solution:**

The original diagram for the summer circuit was incorrect. Circuits which had a resistor value of $n \times R_1$ for the feedback of the first amplifier in the summing block will not be deducted points.
4. Closed-loop control of SIXT33N

Last time, we discovered that open-loop control was not enough to ensure that our car goes straight in the event of model mismatch. In this problem, we will introduce closed-loop control which will hopefully make SIXT33N finally go straight.

Previously, we introduced $\delta(t) = d_L(t) - d_R(t)$ as the difference in positions between the two wheels. If both wheels of the car are going at the same velocity, then this difference $\delta$ should remain constant, since no wheel will advance by more ticks than the other. In our closed loop control scheme, we will consider a control scheme which will apply a simple proportional control $k_L$ and $k_R$ against $\delta(t)$ in order to try to prevent $|\delta(t)|$ from growing without bound.

$$v_L(t) = d_L(t + 1) - d_L(t) = \theta_L u_L(t) - \beta_L$$
$$v_R(t) = d_R(t + 1) - d_R(t) = \theta_R u_R(t) - \beta_R$$

We want to achieve the following equations:
v_L(t) = d_L(t+1) - d_L(t) = v^* - k_L \delta(t)

v_R(t) = d_R(t+1) - d_R(t) = v^* + k_R \delta(t)

We can put the equations in the following form to figure out how we should change our control inputs.

v_L(t) = d_L(t+1) - d_L(t) = \theta_L(v^* + \beta_L \delta(t)) - \beta_L

v_R(t) = d_R(t+1) - d_R(t) = \theta_R(v^* + \beta_R \delta(t)) + k_R \delta(t)

These are our new closed-loop control inputs - the new closed-loop proportional control is the $k_L/k_R$ term.

u_L(t) = \frac{v^* + \beta_L}{\theta_L} \delta(t) - k_L \frac{\delta(t)}{\theta_L}

u_R(t) = \frac{v^* + \beta_R}{\theta_R} \delta(t) + k_R \frac{\delta(t)}{\theta_R}

(a) Let’s examine the feedback proportions $k_L$ and $k_R$ more closely. Should they be positive or negative? What do they mean? Think about how they interact with $\delta(t)$.

**Solution:** If $\delta(t) > 0$, it means that $d_L(t) > d_R(t)$, so the left wheel is ahead of the right one. In order to correct for this, we should help the right wheel catch up, and we should do this by making $k_L > 0$ in order to apply less power on the left wheel and $k_R > 0$ in order to apply more power to the right wheel. Likewise, if $\delta(t) < 0$, it means that $d_L(t) < d_R(t)$, so the right wheel is ahead of the left one. In this case, $k_L > 0$ is still valid, since $k_L \delta(t) > 0$ and so the left wheel speeds up, and likewise $k_R > 0$ is still correct since $k_R \delta(t) < 0$ so the right wheel slows down.

(b) Let’s look a bit more closely at picking $k_L$ and $k_R$. Firstly, we need to figure out what happens to $\delta(t)$ over time. Find $\delta(t+1)$ in terms of $\delta(t)$.

**Solution:**

\[
\delta(t + 1) = d_L(t + 1) - d_R(t + 1)
= v^* - k_L \delta(t) + d_L(t) - (v^* + k_R \delta(t) + d_R(t))
= v^* - k_L \delta(t) + d_L(t) - v^* + k_R \delta(t) - d_R(t)
= -k_L \delta(t) - k_R \delta(t) + (d_L(t) - d_R(t))
= -k_L \delta(t) - k_R \delta(t) + \delta(t)
= \delta(t)(1 - k_L - k_R)
\]
(c) Given your work above, what is the eigenvalue of the system defined by \( \delta(t) \)? For discrete-time systems like our system, \( \lambda \in [-1, 1] \) is considered stable. Are \( \lambda \in [0, 1] \) and \( \lambda \in [-1, 0] \) identical in function for our system? Which one is "better"? (Hint: preventing oscillation is a desired benefit.)

Based on your choice for the range of \( \lambda \) above, how should we set \( k_L \) and \( k_R \) in the end?

**Solution:** The eigenvalue is \( \lambda = 1 - k_L - k_R \).

As a discrete system, both are stable, but \( \lambda \in [-1, 0] \) will cause the car to oscillate due to overly high gain. Therefore, we should choose \( \lambda \in [0, 1] \).

As a result, \( 1 - k_L - k_R \in [0, 1] \rightarrow (k_L + k_R) \in [0, 1] \) means that we should set the gains such that \( (k_L + k_R) \in [0, 1] \).

(d) Let’s re-introduce the model mismatch from last week in order to model environmental discrepancies, disturbances, etc. How does closed-loop control fare under model mismatch? Find \( \delta_s = \delta(t \rightarrow \infty) \), assuming that \( \delta(0) = \delta_0 \). What is \( \delta_s \)? (To make this easier, you may leave your answer in terms of appropriately defined \( c \) and \( \lambda \) obtained from an equation in the form of \( \delta(t + 1) = \delta(t) \lambda + c \).

Check your work by verifying that you reproduce the equation in part (c) if all model mismatch terms are zero. Is it better than the open-loop model mismatch case from last week?

\[
\begin{align*}
v_L(t) &= d_L(t+1) - d_L(t) = (\theta_L + \Delta \theta_L)u_L(t) - (\beta_L + \Delta \beta_L) \\
v_R(t) &= d_R(t+1) - d_R(t) = (\theta_R + \Delta \theta_R)u_R(t) - (\beta_R + \Delta \beta_R)
\end{align*}
\]

\[
\begin{align*}
\frac{d}{dt}u_L(t) &= \frac{v^* + \beta_L}{\theta_L} - k_L \frac{\delta(t)}{\theta_L} \\
\frac{d}{dt}u_R(t) &= \frac{v^* + \beta_R}{\theta_R} + k_R \frac{\delta(t)}{\theta_R}
\end{align*}
\]

**Solution:**

\[
\begin{align*}
\delta(t + 1) &= d_L(t + 1) - d_R(t + 1) \\
&= (\theta_L + \Delta \theta_L)u_L(t) - (\beta_L + \Delta \beta_L) + d_L(t) - ((\theta_R + \Delta \theta_R)u_R(t) - (\beta_R + \Delta \beta_R) + d_R(t)) \\
&= \theta_Lu_L(t) - \beta_L + \Delta \theta_Lu_L(t) - \Delta \beta_L + d_L(t) - (\theta_Ru_R(t) - \beta_R + \Delta \theta_Ru_R(t) - \Delta \beta_R + d_R(t)) \\
&= v^* - k_L \delta(t) + \Delta \theta_Lu_L(t) - \Delta \beta_L + d_L(t) - (v^* + k_R \delta(t) + \Delta \theta_Ru_R(t) - \Delta \beta_R + d_R(t)) \\
&= v^* - k_L \delta(t) + \Delta \theta_Lu_L(t) - \Delta \beta_L + d_L(t) - v^* + k_R \delta(t) - \Delta \theta_Ru_R(t) + \Delta \beta_R - d_R(t) \\
&= v^* - v^* + (d_L(t) - d_R(t)) - k_L \delta(t) - k_R \delta(t) + \Delta \theta_Lu_L(t) - \Delta \beta_L - \Delta \theta_Ru_R(t) + \Delta \beta_R \\
&= \delta(t)(1 - k_L - k_R) + \Delta \theta_Lu_L(t) - \Delta \beta_L - \Delta \theta_Ru_R(t) + \Delta \beta_R \\
&= \delta(t)(1 - k_L - k_R) + \Delta \theta_L\left(\frac{v^* + \beta_L}{\theta_L} - k_L \frac{\delta(t)}{\theta_L}\right) - \Delta \theta_R\left(\frac{v^* + \beta_R}{\theta_R} + k_R \frac{\delta(t)}{\theta_R}\right) - \frac{\Delta \theta_L}{\theta_L} - \frac{\Delta \beta_L}{\theta_L} + \frac{\Delta \theta_R}{\theta_R} - \Delta \beta_R + \Delta \beta_R \\
&= \delta(t)(1 - k_L - k_R) - \Delta \theta_L\left(\frac{v^* + \beta_L}{\theta_L} - k_L \frac{\delta(t)}{\theta_L}\right) - \Delta \theta_R\left(\frac{v^* + \beta_R}{\theta_R} + k_R \frac{\delta(t)}{\theta_R}\right) + \frac{\Delta \theta_L}{\theta_L} - \frac{\Delta \beta_L}{\theta_L} + \frac{\Delta \theta_R}{\theta_R} - \Delta \beta_R + \Delta \beta_R
\end{align*}
\]
Let us define \( c = \left( \frac{\Delta \theta_L}{\theta_L} (v^* + \beta_L) - \Delta \beta_L \right) - \left( \frac{\Delta \theta_R}{\theta_R} (v^* + \beta_R) - \Delta \beta_R \right) \), and our new eigenvalue \( \lambda = 1 - k_L - k_R - k_L \frac{\Delta \theta_L}{\theta_L} - k_R \frac{\Delta \theta_R}{\theta_R} \). In this case,

\[
\delta(1) = \delta_0 \lambda + c \\
\delta(2) = \lambda (\delta_0 \lambda + c) + c = \delta_0 \lambda^2 + c \lambda + c \\
\delta(3) = \lambda (\delta_0 \lambda^2 + c \lambda + c) + c = \delta_0 \lambda^3 + c \lambda^2 + c \lambda + c \\
\delta(4) = \lambda (\delta_0 \lambda^3 + c \lambda^2 + c \lambda + c) + c = \delta_0 \lambda^4 + c \lambda^3 + c \lambda^2 + c \lambda + c \\
\delta(5) = \delta_0 \lambda^5 + c (\lambda^4 + \lambda^3 + \lambda^2 + \lambda + 1) \\
\delta(n) = \delta_0 \lambda^n + c (1 + \lambda + \lambda^2 + \lambda^3 + \lambda^4 + \ldots + \lambda^n) \\
\delta(n) = \delta_0 \lambda^n + c \left( \sum_{k=0}^{n} \lambda^k \right) \text{(rewriting in sum notation)} \\
\delta(n) = \delta_0 \lambda^n + c \left( \frac{1 - \lambda^{n+1}}{1 - \lambda} \right) \text{(sum of a geometric series)}
\]

If \( \lambda < 1 \), then \( \lambda^\infty = 0 \), so those terms drop out:

\[
\delta(n = t \to \infty) = \delta_0 \lambda^\infty + c \left( \frac{1 - \lambda^\infty}{1 - \lambda} \right) \\
\delta(n = t \to \infty) = c \frac{1}{1 - \lambda}
\]

\[
\delta_{ss} = c \frac{1}{1 - \lambda}
\]

For your entertainment only: \( \delta_{ss} \) is fully-expanded form (not required) is

\[
\left( \frac{\Delta \theta_L}{\theta_L} (v^* + \beta_L) - \Delta \beta_L \right) - \left( \frac{\Delta \theta_R}{\theta_R} (v^* + \beta_R) - \Delta \beta_R \right) \\
\frac{k_L + k_R + k_L \frac{\Delta \theta_L}{\theta_L} + k_R \frac{\Delta \theta_R}{\theta_R}}{1 - \lambda}
\]

The answer is correct, since plugging in zero into all the model mismatch terms into \( c \) causes \( c = 0 \), so \( \delta_{ss} = 0 \) if there is no model mismatch. Compared to the open-loop result of \( \delta_{ss} = \pm \infty \), the closed loop \( \delta_{ss} = c \frac{1}{1 - \lambda} \) is a much-desired improvement.

What does this mean for the car? It means that the car will turn initially for a bit but eventually converge to a fixed heading and keep going straight from there.

**Contributors:**

- Justin Yim.
- Tianrui Guo.
- Edward Wang.