EECS 16B Designing Information Systems and Devices II UC Berkeley Spring 2022

Homework 14

This homework is optional.

1. **Study Groups**

   If you are a student who participated in the study group survey we gave in the early weeks of the semester, we would really appreciate your feedback on the group you were matched with. If you did not participate in a study group, we would appreciate your input on what factors went into this decision. Please fill out this form to provide any feedback.

   This is optional, so to have something to write for the question, please tell us whether you filled out the survey or not.

2. Minimum Norm Variants

Given a wide matrix A (with m columns and n rows) and a wide matrix C (with m columns and r rows), we want to solve:

$$\min_{\vec{x} \text{ such that } A\vec{x}=\vec{y}} \|C\vec{x}\| \tag{1}$$

As mentioned above, the key new issue is to isolate the "free" directions in which we can vary $\vec{x}$ so that they might be properly exploited. Consider the full SVD of $C = U\Sigma_C V^\top = \sum_{i=1}^{\ell} \sigma_{c,i}\vec{u}_i\vec{v}_i^\top$. Here, we write:

$$V = \begin{bmatrix} V_C & | & V_F \end{bmatrix}, \quad V_C = \begin{bmatrix} | & | & & | \\ \vec{v}_1 & \vec{v}_2 & \cdots & \vec{v}_\ell \\ | & | & & | \end{bmatrix}, V_F = \begin{bmatrix} | & & | \\ \vec{v}_{\ell+1} & \cdots & \vec{v}_m \\ | & & | \end{bmatrix} \tag{2}$$

so that the columns of V_C all correspond to singular values $\sigma_{c,i} > 0$ of C, and the columns of V_F form an orthonormal basis for the nullspace of C.

Change variables in the problem to be in terms of $\vec{\tilde{x}} = \begin{bmatrix} \vec{\tilde{x}}_c \\ \vec{\tilde{x}}_f \end{bmatrix}$ where the ℓ-dimensional $\vec{\tilde{x}}_c$ has i-th entry $\tilde{x}_{c,i} = \alpha_i \vec{v}_i^\top \vec{x}$, and the $(m-\ell)$-dimensional $\vec{\tilde{x}}_f$ has i-th entry $\tilde{x}_{f,i} = \vec{v}_{\ell+i}^\top \vec{x}$. In vector/matrix form, $\vec{\tilde{x}}_f = V_F^\top \vec{x}$ and $\vec{\tilde{x}}_c = \begin{bmatrix} \alpha_1 & 0 & \cdots & 0 \\ 0 & \alpha_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \alpha_\ell \end{bmatrix} V_C^\top \vec{x}$. Or directly:

$$\vec{\tilde{x}} = \begin{bmatrix} \vec{\tilde{x}}_c \\ \vec{\tilde{x}}_f \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} \alpha_1 & 0 & \cdots & 0 \\ 0 & \alpha_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \alpha_\ell \end{bmatrix} V_C^\top \\ V_F^\top \end{bmatrix} \vec{x}, \quad \begin{bmatrix} \alpha_1 & 0 & \cdots & 0 \\ 0 & \alpha_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \alpha_\ell \end{bmatrix} V_C^\top \in \mathbb{R}^{\ell \times m}, \quad V_F^\top \in \mathbb{R}^{(m-\ell)\times m}. \tag{3}$$

(a) **Express $\vec{x}$ in terms of $\vec{\tilde{x}}_f$ and $\vec{\tilde{x}}_c$.** Assume the $\alpha_i \neq 0$ so the relevant matrix is invertible. *(HINT: If you get stuck on how to express $\vec{x}$ in terms of the new variables, think about the special case when $\ell = 1$ and $\alpha_1 = \frac{1}{2}$. How is this different from when $\alpha_1 = 1$?)*

Solution: Let us give a name to the matrix containing the α_i.

$$B = \begin{bmatrix} \alpha_1 & 0 & \cdots & 0 \\ 0 & \alpha_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \alpha_l \end{bmatrix} \tag{4}$$

Note that we can write $\vec{\tilde{x}}_c = BV_C^\top \vec{x}$ and $\vec{\tilde{x}}_f = V_F^\top \vec{x}$. Since the columns of V_C and V_F together form a basis for the space, it follows that we can write $\vec{x} = V_C\vec{w}_1 + V_F\vec{w}_2$, where $\vec{x}$ is expressed as a linear combination of the $\vec{v}_i$ with unknown weights $\vec{w}_1$ and $\vec{w}_2$. Then, by premultiplying $\vec{x}$ in by V_C and V_F, we have by orthonormality of the columns of V_C and V_F:

$$\begin{aligned} V_C^\top \vec{x} &= V_C^\top (V_C\vec{w}_1 + V_F\vec{w}_2) & V_F^\top \vec{x} &= V_F^\top (V_C\vec{w}_1 + V_F\vec{w}_2) \\ B^{-1}\vec{\tilde{x}}_c &= I\vec{w}_1 + 0\vec{w}_2 & \vec{\tilde{x}}_f &= 0\vec{w}_1 + I\vec{w}_2 \\ B^{-1}\vec{\tilde{x}}_c &= \vec{w}_1 & \vec{\tilde{x}}_f &= \vec{w}_2 \end{aligned} \tag{5}$$

We can thus write $\vec{x}$ in terms of $\vec{\tilde{x}}_c$ and $\vec{\tilde{x}}_f$.

$$\vec{x} = V_C B^{-1}\vec{\tilde{x}}_c + V_F\vec{\tilde{x}}_f = \begin{bmatrix} V_C B^{-1} & V_F \end{bmatrix} \begin{bmatrix} \vec{\tilde{x}}_c \\ \vec{\tilde{x}}_f \end{bmatrix} \tag{6}$$

Another approach can proceed from block matrix reasoning.

$$\vec{\tilde{x}} = \begin{bmatrix} \vec{\tilde{x}}_c \\ \vec{\tilde{x}}_f \end{bmatrix} = \begin{bmatrix} BV_C^\top \\ V_F^\top \end{bmatrix} \vec{x} = \begin{bmatrix} B & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} V_C^\top \\ V_F^\top \end{bmatrix} \vec{x} \tag{7}$$

$$\begin{bmatrix} B & 0 \\ 0 & I \end{bmatrix}^{-1} \begin{bmatrix} \vec{\tilde{x}}_c \\ \vec{\tilde{x}}_f \end{bmatrix} = V^\top \vec{x} \tag{8}$$

Since B is diagonal, the matrix containing B in the previous line is also diagonal, so its inverse has the reciprocal of all the diagonal entries which yields B^{-1} as a submatrix. We can also premultiply by V to cancel the $V^\top$.

$$\vec{x} = V \begin{bmatrix} B^{-1} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \vec{\tilde{x}}_c \\ \vec{\tilde{x}}_f \end{bmatrix} = V \begin{bmatrix} B^{-1}\vec{\tilde{x}}_c \\ I\vec{\tilde{x}}_f \end{bmatrix} = \begin{bmatrix} V_C & V_F \end{bmatrix} \begin{bmatrix} B^{-1}\vec{\tilde{x}}_c \\ \vec{\tilde{x}}_f \end{bmatrix} = V_C B^{-1}\vec{\tilde{x}}_c + V_F \vec{\tilde{x}}_f \tag{9}$$

The above level of detail and both approaches were not required and is for your understanding. Mark your approach appropriately for identifying the relationship between $\vec{x}$ and $\vec{\tilde{x}}$.

(b) Let us now focus on a simple case. Suppose that $A = \begin{bmatrix} A_C & | & A_F \end{bmatrix}$ where the columns of A_F are orthonormal, as well as orthogonal to the columns of A_C. The columns of A together span the entire n-dimensional space. We directly write $\vec{x} = \begin{bmatrix} \vec{x}_c \\ \vec{x}_f \end{bmatrix}$ so that $A\vec{x} = A_C\vec{x}_c + A_F\vec{x}_f$.

Now suppose that we want to solve $A\vec{x} = \vec{y}$ and only care about minimizing $\|\vec{x}_c\|$. We don't care about the length of $\vec{x}_f$ — it can be as big or small as necessary.

In other words, we want to solve:

$$\min_{\vec{x}=\begin{bmatrix} \vec{x}_c \\ \vec{x}_f \end{bmatrix} \text{such that} \begin{bmatrix} A_C & | & A_F \end{bmatrix} \begin{bmatrix} \vec{x}_c \\ \vec{x}_f \end{bmatrix} = \vec{y}} \|\vec{x}_c\| \tag{10}$$

Show that the optimal solution has $\vec{x}_f = A_F^\top \vec{y}$.

(HINT: Multiplying both sides of something by $A_F^\top$ might be helpful.))

Solution: The optimal solution of the minimization satisfies what all solutions, $\vec{x}_{\text{sol}}$, of the constraint equation must satisfy:

$$\vec{y} = A\vec{x}_{\text{sol}} = A_C\vec{x}_c + A_F\vec{x}_f$$

and hence we can multiply both sides by $A_F^\top$ on the left side to get

$$A_F^\top \vec{y} = A_F^\top A_C \vec{x}_c + A_F^\top A_F \vec{x}_f = \vec{0} + I \cdot \vec{x}_f = \vec{x}_f.$$

(c) Continuing the previous part, **compute the optimal $\vec{x}_c$.** Show your work.

(HINT: What is the work that $\vec{x}_c$ needs to do? $\vec{y} - A_F A_F^\top \vec{y}$ might play a useful role, as will the SVD of $A_C = \sum_i \sigma_i \vec{t}_i \vec{w}_i^\top$.))

Solution: From the above question, we have

$$\vec{y} = A_C\vec{x}_c + A_F\vec{x}_f = A_C\vec{x}_c + A_F A_F^\top \vec{y}$$

which can be rewritten as

$$A_C\vec{x}_c = (I - A_F A_F^\top)\vec{y}$$

and since we want to minimize $\|\vec{x}_c\|$, the solution will be

$$\widehat{\vec{x}}_c = A_C^\dagger (I - A_F A_F^\top)\vec{y},$$

where recall that $A_C^\dagger$ is the Moore-Penrose pseudoinverse of A_C. To calculate the pseudoinverse, let

$$A_C = \sum_i \sigma_i \vec{t}_i \vec{w}_i^\top = T\Sigma W^\top.$$

Here we are using the compact SVD of A_C where Σ is the square matrix of nonzero singular values, with T and W as non-square matrices with the corresponding left and right singular vectors. Then, the pseudoinverse of A_C is given by

$$A_C^\dagger = W\Sigma^{-1}T^\top.$$

Finally, the solution we want is

$$\vec{\hat{x}}_c = W\Sigma^{-1}T^\top(I - A_F A_F^\top)\vec{y} = \sum_i \vec{w}_i \left(\frac{1}{\sigma_i}\right)\vec{t}_i^\top (I - A_F A_F^\top)\vec{y}. \tag{11}$$

It is interesting to ask the question whether this special case can be further simplified. And indeed, it can be. Notice that the columns $\vec{t}_i$ span the columns of A_C and in this special case, we have said that the columns of A_C are orthogonal to the columns of A_F. This means that $\vec{t}_i^\top A_F = \vec{0}^\top$. Consequently,

$$\widehat{\vec{x}}_c = \sum_i \vec{w}_i \left(\frac{1}{\sigma_i}\right)\vec{t}_i^\top (I - A_F A_F^\top)\vec{y} = \sum_i \vec{w}_i \left(\frac{1}{\sigma_i}\right)\vec{t}_i^\top \vec{y} = W\Sigma^{-1}T^\top \vec{y} = A_C^\dagger \vec{y}$$

This is also a fully correct answer for this special case. The pseudo-inverse for a matrix that doesn't have a full column rank automatically contains an implicit projection-type aspect to it. This is something that you might have considered in the context of the MIMO HW problem's exploration of the connection between least-squares and minimum-norm solutions.

(d) Now suppose that A_C did not necessarily have its columns orthogonal to A_F. Continue to assume that A_F has orthonormal columns. (You can do this part even if you didn't get any of the previous parts.) Write the matrix $A_C = A_{C\perp} + A_{CF}$ where the columns of A_{CF} are all in the column span of A_F and the columns of $A_{C\perp}$ are all orthogonal to the columns of A_F. **Give an expression for A_{CF} in terms of A_C and A_F.**

(HINT: What does this have to do with projection and least squares?)

Solution: We can see that A_{CF} is the projection of A_C onto A_F, and hence we can directly get $A_{CF} = A_F A_F^\top A_C$ since the A_F has orthonormal columns. This is a projection onto a subspace (by using a whole matrix at a time) we used when we did system identification using least squares.

To expand the argument, we can also do it the following way. Note that

$$A_F^\top A_C = A_F^\top A_{C\perp} + A_F^\top A_{CF} = A_F^\top A_{CF}.$$

Since A_{CF} is in the span of A_F, we can write $A_{CF} = A_F W$ for some W. Then,

$$A_F^\top A_C = A_F^\top A_F W \implies W = (A_F^\top A_F)^{-1} A_F^\top A_C = A_F^\top A_C.$$

In this case, we didn't need to write out the SVD of A_F to get this inverse term. This is because we assumed A_F is orthonormal and hence $A_F^\top A_F = I$ is known.

Finally, recall that we defined $A_{CF} = A_F W$. Then, we conclude

$$A_{CF} = A_F A_F^\top A_C.$$

(e) Continuing the previous part, **compute the optimal $\vec{x}_c$** that solves (10): (copied below)

$$\min_{\vec{x}=\begin{bmatrix}\vec{x}_c\\ \vec{x}_f\end{bmatrix} \text{such that} \begin{bmatrix} A_C & | & A_F \end{bmatrix}\begin{bmatrix}\vec{x}_c\\ \vec{x}_f\end{bmatrix}=\vec{y}} \|\vec{x}_c\|$$

Show your work. Feel free to call the SVD as a black box as a part of your computation.
(HINT: What is the work that $\vec{x}_c$ needs to do? The SVD of $A_{C\perp}$ might be useful.))

Solution: Note that $\vec{y}$ can be written as $\vec{y}_{c\perp} + \vec{y}_f$ such that $\vec{y}_{c\perp}$ is in the span of $A_{C\perp}$, and $\vec{y}_f$ is in the span of A_F. By projection, we have $\vec{y}_f = A_F A_F^\top \vec{y}$ and $\vec{y}_{c\perp} = y - A_F A_F^\top \vec{y} = (I - A_F A_F^\top)\vec{y}$. On the other hand, the amount $\vec{x}_c$ contributes to $\vec{y}_{c\perp}$ is $A_{C\perp}\vec{x}_c$. Hence, all solutions must satisfy:

$$A_{C\perp}\vec{x}_c = \left(I - A_F A_F^\top\right)\vec{y}.$$

Notice that we couldn't have just used A_C here instead of $A_{C\perp}$ because we don't really know what $A_{C\perp}\vec{x}_c$ must be — it is allowed to have some components in the subspace spanned by A_F. However, $A_{C\perp}\vec{x}_c$ cannot have any components in that direction by construction, since every column in $A_{C\perp}$ is orthogonal to the entire subspace spanned by A_F.
Anyway, because we want to minimize $\|\vec{x}_c\|$, we solve this as

$$\vec{x}_c = A_{C\perp}^\dagger \left(I - A_F A_F^\top\right)\vec{y}.$$

To calculate this pseudoinverse, let the compact SVD of $A_{C\perp}$ be:

$$A_{C\perp} = U_{C\perp}\Sigma_{C\perp}V_{C\perp}^\top.$$

Then, the pseudoinverse of $A_{C\perp}$ is given by

$$A_{C\perp}^\dagger = V_{C\perp}\Sigma_{C\perp}^{-1}U_{C\perp}^\top.$$

Hence, the solution we want is

$$\vec{x}_c = V_{C\perp}\Sigma_{C\perp}^{-1}U_{C\perp}^\top\left(I - A_F A_F^\top\right)\vec{y}.$$

The insight that carries over from our simplification made in part (c) and (d) is the utilization of the pseudoinverse of non A_F directions we get from our minimized vector $\|\vec{x}_c\|$. The only difference is that we project A_C to $A_{C\perp} = A_C - A_{CF} = (I - A_F A_F^\top)A_C$ to only get directions that interact with $A_{C\perp}$ and cancel with A_{CF} to insure non-redundancy with what $\vec{x}_f$ gives us.
Once again, we can notice that because the columns of $A_{C\perp}$ are orthogonal to the columns of A_F by construction, that this also carries over to the relevant columns of $U_{C\perp}$ (the ones that correspond to nonzero singular values). Consequently, the above can be simplified further to:

$$\vec{x}_c = V_{C\perp}\Sigma_{C\perp}^{-1}U_{C\perp}^\top\vec{y}$$

by again implicitly leveraging the connection between minimum-norm solutions and least squares. We didn't expect any students to necessarily notice this fact that the pseudo-inverse of $A_{C\perp}$ effectively can do all the work itself.

(f) Continuing the previous part, **compute the optimal $\vec{x}_f$.** Show your work.
You can use the optimal $\vec{x}_c$ in your expression just assuming that you did the previous part correctly, even if you didn't. You can also assume a decomposition $A_C = A_{C\perp} + A_{CF}$ from further above in part (e) without having to write what these are, just assume that you did them correctly, even if you didn't do them at all.
(HINT: What is the work that $\vec{x}_f$ needs to do? How is A_{CF} relevant here?)
Solution: We have

$$A_C\vec{x}_c + A_F\vec{x}_f = \vec{y}. \tag{12}$$

The simplest aproach is just to collect the terms and solve for $\vec{x}_f$ by noticing $A_F\vec{x}_f = \vec{y} - A_C\vec{x}_c$ and so $\vec{x}_f = A_F^\top\vec{y} - A_F^\top A_C\vec{x}_c$. You could just stop here for full credit.

Replacing $A_C = A_{C\perp} + A_{CF}$ and then using $A_{CF} = A_F A_F^\top A_C$ as derived in part (e), we have

$$A_{C\perp}\vec{x}_c + A_F A_F^\top A_C \vec{x}_c + A_F \vec{x}_f = \vec{y}. \tag{13}$$

First, we multiply $A_F^\top$ on the left for both sides and recalling $A_F^\top A_{C\perp} = 0$, we get

$$A_F^\top A_C \vec{x}_c + \vec{x}_f = A_F^\top \vec{y}.$$

This again implies that

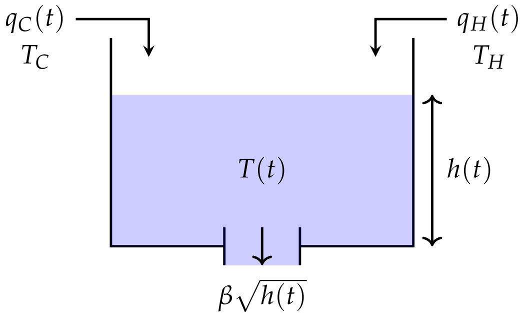
$$\vec{x}_f = A_F^\top \vec{y} - A_F^\top A_C \vec{x}_c. \tag{14}$$

So doing this substitution doesn't really buy us anything more.

Although you have worked out the problem here for the case of A_F having orthonormal columns, you should hopefully see that you actually could use what you know to solve the fully general case.

3. Linearization of Mixing Tank

Consider a mixing tank with cold and hot water supplies and constant supply temperatures $T_C < T_H$. We let $q_C(t)$ and $q_H(t)$ denote the input flow rate for each supply, and treat them as control inputs. We denote by $h(t)$ and $T(t)$ the height and temperature of the water in the tank, and treat them as state variables. The following picture may help in visualization:



The differential equations governing these variables are:

$$\frac{\mathrm{d}}{\mathrm{d}t}h(t) = \frac{1}{\alpha}\left(q_C(t) + q_H(t) - \beta\sqrt{h(t)}\right) \tag{15}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}T(t) = \frac{1}{\alpha h(t)}(q_C(t)[T_C - T(t)] + q_H(t)[T_H - T(t)]), \tag{16}$$

where α is the area of a cross-section of the tank and β is a constant, such that the term $\beta\sqrt{h(t)}$ dictates the rate at which water is drained. Using the standard state and input notation, we let $x_1(t) := h(t)$, $x_2(t) := T(t)$, $u_1(t) := q_C(t)$, $u_2(t) := q_H(t)$, and rewrite the equations above as

$$\frac{\mathrm{d}}{\mathrm{d}t}\underbrace{\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}}_{:=\vec{x}(t)} = \underbrace{\begin{bmatrix} f_1(x_1(t), x_2(t), u_1(t), u_2(t)) \\ f_2(x_1(t), x_2(t), u_1(t), u_2(t)) \end{bmatrix}}_{:=\vec{f}(\vec{x}(t),\vec{u}(t))}. \tag{17}$$

(a) **Write the functions $f_1(x_1, x_2, u_1, u_2)$ and $f_2(x_1, x_2, u_1, u_2)$ explicitly using Equations (15) and (16).**

Solution: Relabeling Equations (15) and (16), we have

$$\frac{\mathrm{d}}{\mathrm{d}t}x_1(t) = \frac{1}{\alpha}\left(u_1(t) + u_2(t) - \beta\sqrt{x_1(t)}\right) \tag{18}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}x_2(t) = \frac{1}{\alpha x_1(t)}(u_1(t)[T_C - x_2(t)] + u_2(t)[T_H - x_2(t)]) \tag{19}$$

Thus

$$f_1(x_1, x_2, u_1, u_2) = \frac{1}{\alpha}(u_1 + u_2 - \beta\sqrt{x_1}) \tag{20}$$

$$f_2(x_1, x_2, u_1, u_2) = \frac{1}{\alpha x_1}(u_1[T_C - x_2] + u_2[T_H - x_2]) \tag{21}$$

as desired.

(b) **Suppose we want the operating point to be $(h^\star, T^\star)$, where $h^\star > 0$ and $T_C \leq T^\star \leq T_H$. What are the corresponding input values, $(u_1^\star, u_2^\star)$?**

Solution: Since this is a continuous time model, the operating point concept is $\vec{f}(\vec{x}^\star, \vec{u}^\star) = \vec{0}$. In our notation, we would like $f_1(x_1^\star, x_2^\star, u_1^\star, u_2^\star) = 0$ and $f_2(x_1^\star, x_2^\star, u_1^\star, u_2^\star) = 0$. Since $x_1(t) = h(t)$ and $x_2(t) = T(t)$, we have that $x_1^\star = h^\star$ and $x_2^\star = T^\star$.

Plugging in, we have

$$\begin{aligned} 0 &= f_1(x_1^\star, x_2^\star, u_1^\star, u_2^\star) \tag{22}\\ &= f_1(h^\star, T^\star, u_1^\star, u_2^\star) \tag{23}\\ &= \frac{1}{\alpha}\left(u_1^\star + u_2^\star - \beta\sqrt{h^\star}\right) \tag{24}\\ \implies \beta\sqrt{h^\star} &= u_1^\star + u_2^\star \tag{25} \end{aligned}$$

and

$$\begin{aligned} 0 &= f_2(x_1^\star, x_2^\star, u_1^\star, u_2^\star) \tag{26}\\ &= f_2(h^\star, T^\star, u_1^\star, u_2^\star) \tag{27}\\ &= \frac{1}{\alpha h^\star}(u_1^\star[T_C - T^\star] + u_2^\star[T_H - T^\star]) \tag{28}\\ \implies 0 &= u_1^\star[T_C - T^\star] + u_2^\star[T_H - T^\star]. \tag{29} \end{aligned}$$

Putting these equations into a matrix, we have

$$\begin{bmatrix} 1 & 1 \\ T_C - T^\star & T_H - T^\star \end{bmatrix} \begin{bmatrix} \vec{u}_1^\star \\ \vec{u}_2^\star \end{bmatrix} = \begin{bmatrix} \beta\sqrt{h^\star} \\ 0 \end{bmatrix}. \tag{30}$$

Then

$$\begin{aligned} \begin{bmatrix} \vec{u}_1^\star \\ \vec{u}_2^\star \end{bmatrix} &= \begin{bmatrix} 1 & 1 \\ T_C - T^\star & T_H - T^\star \end{bmatrix}^{-1} \begin{bmatrix} \beta\sqrt{h^\star} \\ 0 \end{bmatrix} \tag{31}\\ &= \begin{bmatrix} \frac{T_H - T^\star}{T_H - T_C} & -\frac{1}{T_H - T_C} \\ \frac{T^\star - T_C}{T_H - T_C} & \frac{1}{T_H - T_C} \end{bmatrix} \begin{bmatrix} \beta\sqrt{h^\star} \\ 0 \end{bmatrix} \tag{32}\\ &= \begin{bmatrix} \beta\sqrt{h^\star}\frac{T_H - T^\star}{T_H - T_C} \\ \beta\sqrt{h^\star}\frac{T^\star - T_C}{T_H - T_C} \end{bmatrix}. \tag{33} \end{aligned}$$

Thus the equilibrium point is

$$(x_1^\star,\ x_2^\star,\ u_1^\star,\ u_2^\star) = \left(h^\star,\ T^\star,\ \beta\sqrt{h^\star}\frac{T^\star - T_C}{T_H - T_C},\ \beta\sqrt{h^\star}\frac{T_H - T^\star}{T_H - T_C}\right) \tag{34}$$

as desired.

(c) **Find the matrices A, B in the linearized model**

$$\frac{\mathrm{d}}{\mathrm{d}t}\delta\vec{x}(t) = A \cdot \delta\vec{x}(t) + B \cdot \delta\vec{u}(t) \tag{35}$$

where

$$\delta\vec{x}(t) := \vec{x}(t) - \begin{bmatrix} h^\star \\ T^\star \end{bmatrix}, \qquad \delta\vec{u}(t) := \vec{u}(t) - \begin{bmatrix} u_1^\star \\ u_2^\star \end{bmatrix}. \tag{36}$$

Solution: From linearization theory, we know

$$A = J_{\vec{x}}\vec{f}(\vec{x}^\star, \vec{u}^\star) \qquad \text{and} \qquad B = J_{\vec{u}}\vec{f}(\vec{x}^\star, \vec{u}^\star). \tag{37}$$

It remains to calculate these Jacobians. They are

$$J_{\vec{x}}\vec{f}(\vec{x}, \vec{u}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(\vec{x}, \vec{u}) & \frac{\partial f_1}{\partial x_2}(\vec{x}, \vec{u}) \\ \frac{\partial f_2}{\partial x_1}(\vec{x}, \vec{u}) & \frac{\partial f_2}{\partial x_2}(\vec{x}, \vec{u}) \end{bmatrix} \tag{38}$$

$$= \begin{bmatrix} -\frac{\beta}{2\alpha\sqrt{x_1}} & 0 \\ -\frac{u_1(T_C-x_2)+u_2(T_H-x_2)}{\alpha x_1^2} & -\frac{u_1+u_2}{\alpha x_1} \end{bmatrix} \tag{39}$$

$$J_{\vec{u}}\vec{f}(\vec{x},\vec{u}) = \begin{bmatrix} \frac{\partial f_1}{\partial u_1}(\vec{x},\vec{u}) & \frac{\partial f_1}{\partial u_2}(\vec{x},\vec{u}) \\ \frac{\partial f_2}{\partial u_1}(\vec{x},\vec{u}) & \frac{\partial f_2}{\partial u_2}(\vec{x},\vec{u}) \end{bmatrix} \tag{40}$$

$$= \begin{bmatrix} \frac{1}{\alpha} & \frac{1}{\alpha} \\ \frac{T_C-x_2}{\alpha x_1} & \frac{T_H-x_2}{\alpha x_1} \end{bmatrix} \tag{41}$$

Thus plugging in our definitions of $x_1^\star = h^\star$, $x_2^\star = T^\star$, $u_1^\star = \beta\sqrt{h^\star}\frac{T^\star-T_C}{T_H-T_C}$ and $u_2^\star = \beta\sqrt{h^\star}\frac{T_H-T^\star}{T_H-T_C}$, we have

$$J_{\vec{x}}\vec{f}(\vec{x}^\star,\vec{u}^\star) = \begin{bmatrix} -\frac{\beta}{2\alpha\sqrt{x_1^\star}} & 0 \\ -\frac{u_1^\star(T_C-x_2^\star)+u_2^\star(T_H-x_2^\star)}{\alpha(x_1^\star)^2} & -\frac{u_1^\star+u_2^\star}{\alpha x_1^\star} \end{bmatrix} \tag{42}$$

$$= \begin{bmatrix} -\frac{\beta}{2\alpha\sqrt{h^\star}} & 0 \\ 0 & -\frac{\beta}{\alpha\sqrt{h^\star}} \end{bmatrix} \tag{43}$$

$$J_{\vec{u}}\vec{f}(\vec{x}^\star,\vec{u}^\star) = \begin{bmatrix} \frac{1}{\alpha} & \frac{1}{\alpha} \\ \frac{T_C-x_2^\star}{\alpha x_1^\star} & \frac{T_H-x_2^\star}{\alpha x_1^\star} \end{bmatrix} \tag{44}$$

$$= \begin{bmatrix} \frac{1}{\alpha} & \frac{1}{\alpha} \\ \frac{T_C-T^\star}{\alpha h^\star} & \frac{T_H-T^\star}{\alpha h^\star} \end{bmatrix}. \tag{45}$$

Thus

$$A = \begin{bmatrix} -\frac{\beta}{2\alpha\sqrt{h^\star}} & 0 \\ 0 & -\frac{\beta}{\alpha\sqrt{h^\star}} \end{bmatrix} \qquad B = \begin{bmatrix} \frac{1}{\alpha} & \frac{1}{\alpha} \\ \frac{T_C-T^\star}{\alpha h^\star} & \frac{T_H-T^\star}{\alpha h^\star} \end{bmatrix} \tag{46}$$

as desired.

(d) **Determine whether the linearized model in part (c) is stable.**

Solution: The eigenvalues of A are

$$\lambda_1(A) = -\frac{\beta}{2\alpha\sqrt{h^\star}} \qquad \lambda_2(A) = -\frac{\beta}{\alpha\sqrt{h^\star}}. \tag{47}$$

Since this is a continuous-time model, the (asymptotic) stability criterion is $\text{Re}\{\lambda_i(A)\} < 0$ for all i. Since $\alpha, \beta, h^\star > 0$, we have that $\lambda_1(A)$ and λ_2 are real numbers; furthermore, we have that $\lambda_1(A) < 0$ and $\lambda_2(A) < 0$. Thus the linearized model is stable.

(e) Suppose $T_C = 10^\circ$, $T_H = 90^\circ$, $\alpha = 3$, $\beta = \frac{1}{6}$, and we choose $h^\star = 1\,\text{m}$, $T^\star = 25^\circ$. **Evaluate A and B in part (c) for these values. What are the eigenvalues of A?**

Solution: We have

$$A = \begin{bmatrix} -\frac{\beta}{2\alpha\sqrt{h^\star}} & 0 \\ 0 & -\frac{\beta}{\alpha\sqrt{h^\star}} \end{bmatrix} \tag{48}$$

$$= \begin{bmatrix} -\frac{1/6}{2\cdot3\cdot\sqrt{1}} & 0 \\ 0 & -\frac{1/6}{3\cdot\sqrt{1}} \end{bmatrix} \tag{49}$$

$$= \begin{bmatrix} -\frac{1}{36} & 0 \\ 0 & -\frac{1}{18} \end{bmatrix} \tag{50}$$

which has the eigenvalues $-\frac{1}{36}$ and $-\frac{1}{18}$, and

$$\begin{aligned} B &= \begin{bmatrix} \frac{1}{\alpha} & \frac{1}{\alpha} \\ \frac{T_C - T^\star}{\alpha h^\star} & \frac{T_H - T^\star}{\alpha h^\star} \end{bmatrix} && (51) \\ &= \begin{bmatrix} \frac{1}{3} & \frac{1}{3} \\ \frac{10-25}{3\cdot 1} & \frac{90-25}{3\cdot 1} \end{bmatrix} && (52) \\ &= \begin{bmatrix} \frac{1}{3} & \frac{1}{3} \\ -5 & \frac{65}{3} \end{bmatrix}. && (53) \end{aligned}$$

Thus the linearized system is

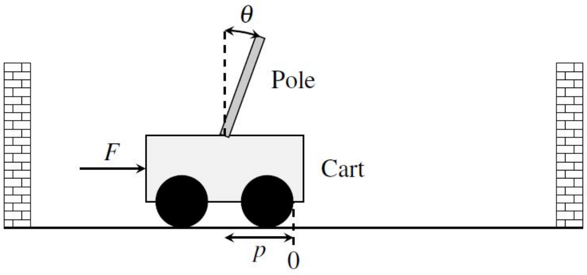
$$\frac{\mathrm{d}}{\mathrm{d}t}\delta\vec{x}(t) = \begin{bmatrix} -\frac{1}{36} & 0 \\ 0 & -\frac{1}{18} \end{bmatrix} \cdot \delta\vec{x}(t) + \begin{bmatrix} \frac{1}{3} & \frac{1}{3} \\ -5 & \frac{65}{3} \end{bmatrix} \cdot \delta\vec{u}(t). \tag{54}$$

4. Segway Tours

A segway is a stand on two wheels, and can be thought of as an inverted pendulum. The segway works by applying a force (through the spinning wheels) to the base of the segway, This controls both the position on the segway and the angle of the stand. As the driver pushes on the stand, the segway tries to bring itself back to the upright position, and it can only do this by moving the base.

Recall that you have analyzed a basic version of this segway question in Discussion 12B problem 2. You were given a linear discrete time representation of the segway dynamics, and were guided through the steps to find if it's possible to make the segway reach some desired states, essentially laying the foundation of controllability. Now, we will see how to derive the linear discrete time system from the equations of motion, and then do some further refined analysis based on our improved knowledge of controllabilty.

The main question we wish to answer is: Is it possible for the segway to be brought upright and to a stop from any initial configuration? There is only one input (force) used to control two outputs (position and angle). Let's model the segway as a cart-pole system and analyze.



A cart-pole system can be fully described by its position p, velocity $\frac{\mathrm{d}p}{\mathrm{d}t}$, angle θ, and angular velocity $\frac{\mathrm{d}\theta}{\mathrm{d}t}$. We can write this as the continuous time state vector $\vec{x}$ as follows:

$$\vec{x} = \begin{bmatrix} p \\ \frac{\mathrm{d}p}{\mathrm{d}t} \\ \theta \\ \frac{\mathrm{d}\theta}{\mathrm{d}t} \end{bmatrix} \tag{55}$$

The input to this system is a scalar quantity $u(t)$ at time t, which is the force F applied to the cart (or base of the segway). Let the coefficient of friction be k.

The equations of motion for this system are as follows:

$$\begin{aligned} \frac{\mathrm{d}^2 p}{\mathrm{d}t^2} &= \frac{1}{\frac{M}{m} + \sin^2\theta}\left(\frac{u}{m} + \left(\frac{\mathrm{d}\theta}{\mathrm{d}t}\right)^2 l \sin\theta - g\sin\theta\cos\theta - \frac{k}{m}\frac{\mathrm{d}p}{\mathrm{d}t}\right) \\ \frac{\mathrm{d}^2\theta}{\mathrm{d}t^2} &= \frac{1}{l\left(\frac{M}{m} + \sin^2\theta\right)}\left(-\frac{u}{m}\cos\theta - \left(\frac{\mathrm{d}\theta}{\mathrm{d}t}\right)^2 l\cos\theta\sin\theta + \frac{M+m}{m}g\sin\theta + \frac{k}{m}\frac{\mathrm{d}p}{\mathrm{d}t}\cos\theta\right) \end{aligned} \tag{56}$$

The derivation of these equations is a mechanics problem and not in 16B scope, but interested students can look up the details online.

(a) First let's linearize the system of equations in (56) about the upright position at rest, i.e. $\theta_* = 0$ and $\frac{\mathrm{d}\theta}{\mathrm{d}t}_* = 0$. **Show that the linearized system of equations is given by the following state**

space form:

$$\frac{\mathrm{d}\vec{x}(t)}{\mathrm{d}t} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{k}{M} & -\frac{m}{M}g & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{k}{Ml} & \frac{M+m}{Ml}g & 0 \end{bmatrix}}_{A} \vec{x}(t) + \underbrace{\begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ -\frac{1}{Ml} \end{bmatrix}}_{\vec{b}} u(t) \tag{57}$$

(HINT: Since we are linearizing around $\theta_* = 0$ and $\frac{d\theta}{dt}_* = 0$, you can use the following approximations for small values of θ:

$$\sin\theta \approx \theta$$
$$\sin^2\theta \approx 0$$
$$\cos\theta \approx 1$$
$$\left(\frac{\mathrm{d}\theta}{\mathrm{d}t}\right)^2 \approx 0.$$

You do not have to do the full linearization using Taylor series, you can just substitute the approximations above. You will get the same answer as doing the linear Taylor series approximation.)

Notice that for this particular choice of θ_* and $\frac{\mathrm{d}\theta}{\mathrm{d}t}_*$, the linearization does not depend on what p or $\frac{\mathrm{d}p}{\mathrm{d}t}$ is. This is partially a stroke of luck and partially a consequence of the fact that the position p doesn't appear in the dynamics equations.

Solution: Using the approximations from the hint, (56) is linearized as follows:

$$\begin{aligned} \frac{\mathrm{d}^2 p}{\mathrm{d}t^2} &= -\frac{k}{M}\frac{\mathrm{d}p}{\mathrm{d}t} - \frac{m}{M}g\theta + \frac{1}{M}u \\ \frac{\mathrm{d}^2\theta}{\mathrm{d}t^2} &= \frac{k}{Ml}\frac{\mathrm{d}p}{\mathrm{d}t} + \frac{M+m}{Ml}g\theta - \frac{1}{Ml}u \end{aligned} \tag{58}$$

Now (58) can be represented in state space form as

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} p \\ \frac{\mathrm{d}p}{\mathrm{d}t} \\ \theta \\ \frac{\mathrm{d}\theta}{\mathrm{d}t} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{k}{M} & -\frac{m}{M}g & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{k}{Ml} & \frac{M+m}{Ml}g & 0 \end{bmatrix}}_{A} \begin{bmatrix} p \\ \frac{\mathrm{d}p}{\mathrm{d}t} \\ \theta \\ \frac{\mathrm{d}\theta}{\mathrm{d}t} \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ -\frac{1}{Ml} \end{bmatrix}}_{\vec{b}} u(t) \tag{59}$$

(b) For all subsequent parts, assume that $m = 1$, $M = 10$, $g = 10$, $l = 1$ and $k = 0.1$. Let's consider the discrete time representation of the state space (57) at time $t = n\Delta$. For simplicity, assume $\Delta = 1$. The discrete time state $\vec{x}_d$ follows the following linear model:

$$\vec{x}_d[n+1] = A_d\vec{x}_d[n] + \vec{b}_d u_d[n] \tag{60}$$

where $A_d \in \mathbb{R}^{4\times 4}$ and $\vec{b}_d \in \mathbb{R}^{4\times 1}$. **Find A_d and $\vec{b}_d$ in terms of the eigenvalues and eigenvectors of A, and Δ. State numerical values for A_d and $\vec{b}_d$.**. Use the Jupyter notebook `segway.ipynb` for all numerical calculations, and approximate the results to 2 or 3 significant figures.

(HINT: Recall that the continuous time scalar differential equation $\frac{\mathrm{d}z(t)}{dt} = \lambda z(t) + cw(t)$ can be represented in discrete time ($n\Delta = t$) as follows:

$$z_d[n+1] = \begin{cases} (e^{\lambda\Delta}) \cdot z_d[n] + \left(\frac{e^{\lambda\Delta}-1}{\lambda}\right) \cdot cw_d[n] & \text{if } \lambda \neq 0 \\ (1) \cdot z_d[n] + (\Delta) \cdot cw_d[n] & \text{if } \lambda = 0 \end{cases}$$

Use the eigendecompostion of $A = V\Lambda V^{-1}$ to do change of basis variables, and you should finally reach

$$\vec{x}_d[n+1] = \underbrace{V\Lambda_d V^{-1}}_{A_d}\vec{x}_d[n] + \underbrace{VM_d V^{-1}\vec{b}}_{\vec{b}_d}u_d[n]$$

What are the elements of Λ_d and M_d in terms of the elements of Λ? You may find in later parts of the notebook that you have A_d and $\vec{b}_d$ which can serve as a sanity check for your derivation and numerical calculations.)

Solution: Using the eigendecompostion of $A = V\Lambda V^{-1}$ and a change of variable $\vec{y}(t) = V^{-1}\vec{x}(t)$, we can transform (57) into a system of decoupled equations

$$\frac{\mathrm{d}\vec{y}}{\mathrm{d}t} = \Lambda\vec{y}(t) + V^{-1}\vec{b}u(t) \tag{61}$$

This can be represented in discrete time as follows:

$$\vec{y}_d[n+1] = \Lambda_d\vec{y}_d[n] + M_d V^{-1}\vec{b}u_d[n] \tag{62}$$

where Λ_d is a diagonal matrix given by

$$\Lambda_{d\,ii} = \begin{cases} e^{\Lambda_{ii}\Delta} & \text{if } \Lambda_{ii} \neq 0 \\ 1 & \text{if } \Lambda_{ii} = 0 \end{cases} \tag{63}$$

and M_d is a diagonal matrix given by

$$M_{d\,ii} = \begin{cases} \frac{e^{\Lambda_{ii}\Delta}-1}{\Lambda_{ii}} & \text{if } \Lambda_{ii} \neq 0 \\ \Delta & \text{if } \Lambda_{ii} = 0 \end{cases} \tag{64}$$

Changing variables back to $\vec{x}_d[n] = V\vec{y}_d[n]$, we get

$$\vec{x}_d[n+1] = \underbrace{V\Lambda_d V^{-1}}_{A_d}\vec{x}_d[n] + \underbrace{VM_d V^{-1}\vec{b}}_{\vec{b}_d}u_d[n] \tag{65}$$

Plugging in the values of $m = 1$, $M = 10$, $g = 10$, $l = 1$, $k = 0.1$, $\Delta = 1$ in the Jupyter notebook, we get

$$\begin{aligned} A_d &\approx \begin{bmatrix} 1 & 0.994 & -1.161 & -0.286 \\ 0 & 0.987 & -4.138 & -1.161 \\ 0 & 0.012 & 13.797 & 4.150 \\ 0 & 0.041 & 45.634 & 13.797 \end{bmatrix} \\ \vec{b}_d &\approx \begin{bmatrix} 0.056 \\ 0.128 \\ -0.116 \\ -0.414 \end{bmatrix} \end{aligned} \tag{66}$$

(c) **Show that the linear-approximation discrete time system in (60) is controllable by using the appropriate matrix in the Jupyter notebook.**

(HINT: Is the controllability matrix full rank? You have to use numerical values of A_d and $\vec{b}_d$ from the previous part. Use the Jupyter notebook for all numerical calculations.)

Solution: Since $A_d \in \mathbb{R}^{4\times 4}$ and $\vec{b}_d \in \mathbb{R}^{4\times 1}$, the controllability matrix is given by

$$\mathcal{C} = \begin{bmatrix} \vec{b}_d & A_d\vec{b}_d & A_d^2\vec{b}_d & A_d^3\vec{b}_d \end{bmatrix}$$

Using the Jupyter notebook, we can see that $\mathcal{C}$ has rank $= 4$, hence the discrete time system is controllable.

(d) Since the linear-approximation discrete time system is controllable, it is possible to reach any final state $\vec{x}_{d,f}$ starting from any initial state $\vec{x}_{d,i}$ using an appropriate sequence of inputs in exactly 4 steps, provided that the deviations are small enough so that the linearization approximation is valid. **Set up a set of linear equations to solve for the $u_d[0]$, $u_d[1]$, $u_d[2]$, $u_d[3]$ given the initial and final states. Find the input sequence to reach the upright position $\vec{x}_{d,f} = \vec{x}_d[4] = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$ starting from an initial state $\vec{x}_{d,i} = \vec{x}_d[0] = \begin{bmatrix} -2 \\ 3.1 \\ 0.3 \\ -0.6 \end{bmatrix}$.** Use the Jupyter notebook for all numerical calculations and simulation. **Explain qualitatively what you observe from the segway simulation.**

(HINT: Use (60) and loop unrolling to express $\vec{x}_d[4]$ as a linear combination of $\vec{x}_d[0]$, $u_d[3]$, $u_d[2]$, $u_d[1]$, $u_d[0]$.)

Solution: In 4 time steps, the discrete time system in (60) reaches

$$\vec{x}_d[4] = A_d^4\vec{x}_d[0] + \vec{b}_d u_d[3] + A_d\vec{b}_d u_d[2] + A_d^2\vec{b}_d u_d[1] + A_d^3\vec{b}_d u_d[0]$$

$$\implies \vec{x}_d[4] - A_d^4\vec{x}_d[0] = \begin{bmatrix} \vec{b}_d & A_d\vec{b}_d & A_d^2\vec{b}_d & A_d^3\vec{b}_d \end{bmatrix} \begin{bmatrix} u_d[3] \\ u_d[2] \\ u_d[1] \\ u_d[0] \end{bmatrix}$$

$$= \mathcal{C} \begin{bmatrix} u_d[3] \\ u_d[2] \\ u_d[1] \\ u_d[0] \end{bmatrix}$$

$$\implies \begin{bmatrix} u_d[3] \\ u_d[2] \\ u_d[1] \\ u_d[0] \end{bmatrix} = \mathcal{C}^{-1}\left(\vec{x}_d[4] - A_d^4\vec{x}_d[0]\right)$$

Using the notebook, we can calculate $\begin{bmatrix} u_d[3] \\ u_d[2] \\ u_d[1] \\ u_d[0] \end{bmatrix} \approx \begin{bmatrix} -1.636 \\ 48.650 \\ -97.747 \\ 17.433 \end{bmatrix}$.

The simulation shows the inverted pendulum stabilizing to the upright position at rest from the initial position.

(e) Now suppose we try to use an initial state $\vec{x}_{d,i} = \vec{x}_d[0] = \begin{bmatrix} -2 \\ 3.1 \\ 3.3 \\ -0.6 \end{bmatrix}$ for which the approximation is poor since $\theta_i = 3.3$ is very far from the linearization point $\theta_* = 0$. **Using the equations derived in the previous part, use the Jupyter notebook to determine the input sequence to reach the same final upright position. Explain qualitatively what you observe from the segway simulation.** Use the Jupyter notebook for all numerical calculations and simulation.

Solution: Using the notebook, we can calculate $\begin{bmatrix} u_d[3] \\ u_d[2] \\ u_d[1] \\ u_d[0] \end{bmatrix} \approx \begin{bmatrix} -15.049 \\ 445.384 \\ -851.258 \\ 387.623 \end{bmatrix}$.

The simulation shows that the inverted pendulum again stabilizes to the upright position at rest from the initial position, but does some weird unexpected rotations before reaching there.

Compare the simulation results in parts (d) and (e). In both cases, the segway finally stabilizes to an upright position at rest. However, in part (d) the behavior of the segway looks more realistic whereas in part (e) it is doing some wild unexpected rotations.

This is because the linearization approximation is valid with the small initial values of θ and $\frac{\mathrm{d}\theta}{\mathrm{d}t}$ in part (d). So this discrete time linear model is a good representation of the original continuous time non-linear system. Hence the trajectory taken by the segway from the initial to the final position is similar to what we may expect from real life physics.

However in part (e), the linearization approximation is not really valid. The approximate model still converges to the final upright position because (60) is controllable as we proved in part (c). However, since the approximation is not valid anymore, this discrete time linear model is **not** a good representation of the original continuous time non-linear system. Hence the predicted trajectory is extremely weird with the segway undergoing a few full rotations, and does not match what we would expect from the real system.

We can still analyze the system in continuous time by directly solving the set of non-linear differential equations in (56) (out of 16B scope) or in discrete time using a finely discretized (and still nonlinear) version of (56). Note that there are two independent distinctions we are making, i.e. continuous vs discrete, and linear vs non-linear. Part (e) failed because it's beyond the scope of the linear model, not because we are using a discrete time system. A non-linear discrete time analysis would also give the correct solution.

(a) Let's analyze the behavior of the segway by comparing the continuous time linear model and continuous time non-linear model. Deriving the control input to bring the segway to the upright position at rest requires more care which is out of 16B scope, so we will just look at the simple case of the segway freely settling to steady state in the absence of any control input, i.e. $u(t) = 0$. **Toggle the `linearized` flag between `True` and `False` in the Jupyter notebook, and qualitatively explain the differences in the trajectory as the segway freely swings around.**

Solution: We notice that the non-linear system is a much more accurate representation of what a real segway looks like. The linearized system starts off looking realistic, but then behaves oddly as the simulation continues for longer. This is because we have approximated a non-linear system with a linear one, so there will be a discrepancy when we have a large deviation from the operating point.

There's actually much more that we could have you do with this problem with what is in scope in 16B. But the semester is drawing to a close and you need to study for other courses too. So we will stop here.

Contributors:

- Anant Sahai.
- Kuan-Yun Lee.
- Murat Arcak.
- Druv Pai.
- Ayan Biswas.
- Daniel Abraham.