Last time, we studied the mass–spring–damper system with differential equation and state-space form:

\[ m\ddot{y} + c\dot{y} + ky = u \quad \iff \quad \begin{cases} 
  \dot{x} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{c}{m} \end{bmatrix} x + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u \\
  y = \begin{bmatrix} 1 & 0 \end{bmatrix} x + [0] u 
\end{cases} \]

In general, if we have derivatives \( y, \dot{y}, y^{(3)}, \ldots, y^{(n)} \) then the differential equation given by:

\[ y^{(n)} + a_{n-1} y^{(n-1)} + \cdots + a_2 \dot{y} + a_1 y = b_0 u \]

can be put into state-space form by defining the state \( x = [y, \dot{y}, \ldots, y^{(n-1)}]^T \in \mathbb{R}^n \). Then, we have:

\[
\begin{cases} 
  \dot{x} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \\ -a_n & -a_{n-1} & -a_{n-2} & \cdots & -1 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ b_0 \end{bmatrix} u \\
  y = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \end{bmatrix} x + [0] u 
\end{cases}
\]

So an \( n \)th order differential equation becomes a state-space system with \( n \) states.
We can also look at things in the Laplace domain. Taking the same differential equation:

\[ y^{(n)} + a_{n-1} y^{(n-1)} + \ldots + a_1 \dot{y} + a_0 y = b_0 u \]

The Laplace transform (assuming \( y^{(k)}(0) = 0 \) for \( k = 0, 1, \ldots, n-1 \)) is:

\[ (s^n + a_{n-1} s^{n-1} + \ldots + a_1 s + a_0) Y(s) = b_0 U(s). \]

so the map from \( u \) to \( y \) in the Laplace domain is:

\[ Y(s) = \frac{b_0}{s^n + a_{n-1} s^{n-1} + \ldots + a_1 s + a_0} U(s) \]

transfer function.

What about differential equations with \( u, \ddot{u}, \text{etc.} \) terms? Consider the equation:

\[ y^{(n)} + a_{n-1} y^{(n-1)} + \ldots + a_1 \dot{y} + a_0 y = b_{n-1} u^{(n-1)} + \ldots + b_1 \dot{u} + b_0 u \]

In the Laplace domain, we have:

\[ (s^n + a_{n-1} s^{n-1} + \ldots + a_1 s + a_0) Y(s) = (b_{n-1} s^{n-1} + \ldots + b_1 s + b_0) U(s) \]

so the transfer function is:

\[ Y(s) = \frac{b_{n-1} s^{n-1} + \ldots + b_1 s + b_0}{s^n + a_{n-1} s^{n-1} + \ldots + a_1 s + a_0} U(s). \]

But what is the state-space form?
To find the state-space form, we can't simply define $x_1 = y$, $x_2 = \dot{y}$, etc. as we did before because we will still have the terms $\ddot{u}$, $u$, etc. that won't go away. The solution is to define a signal $W(s)$ in the Laplace domain such that:

$$W(s) = \frac{1}{s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0} U(s)$$ \quad and \quad $Y(s) = (b_{n-1} s^{n-1} + \cdots + b_1 s + b_0)W(s)$.

Clearly we recover the original transfer function if we substitute and eliminate $W(s)$. But now convert back to the time domain and see what happens:

$$\begin{cases} 0 \ w^{(n)} + a_{n-1} w^{(n-1)} + \cdots + a_1 \dot{w} + a_0 w = u \\ y = b_{n-1} w^{(n-1)} + \cdots + b_1 \dot{w} + b_0 w \end{cases}$$

We can now convert to state-space by defining $x = [w, \dot{w}, \ldots, w^{(n-1)}]^T$:

$$\dot{x} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{n-1} \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u$$

$$y = \begin{bmatrix} b_0 & b_1 & b_2 & \cdots & b_{n-1} \end{bmatrix} x + \begin{bmatrix} 0 \end{bmatrix} u$$

So conversion to state-space form is essentially automatic, we just need to read off the coefficients.
The previous case considered a transfer function:

\[ Y(s) = \frac{b_{m-1}s^{m-1} + \cdots + b_1s + b_0}{s^m + a_{m-1}s^{m-1} + \cdots + a_1s + a_0} U(s) \]

where the degree of the numerator is strictly less than the degree of the denominator. This is called a strictly proper transfer function.

In state-space, strictly proper systems always have \( D = 0 \).

What about the proper case, where numerator and denominator have the same degree? We can use polynomial division to write: \((\text{proper}) = (\text{strictly proper}) + (\text{constant})\).

Example:

\[
\frac{b_2s^2 + b_1s + b_0}{s^2 + a_1s + a_0} = b_2 + \frac{(b_1 - b_2a_1)s + (b_0 - b_1a_0)}{s^2 + a_1s + a_0} \\
\text{constant} \quad \text{strictly proper}
\]

So the state-space realization is simply:

\[
\dot{x} = \begin{bmatrix} 0 & 1 \\ -a_0 & -a_1 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \\
\begin{bmatrix} \hat{b}_0 & \hat{b}_1 \end{bmatrix} x + \begin{bmatrix} b_2 \end{bmatrix} u
\]

So proper transfer functions have \( D \neq 0 \) in state-space.
Linearization of nonlinear systems.

A classic example: the pendulum! A simple force balance reveals that

\[ mI \ddot{\theta} + mg \sin(\theta) = 0 \]

or:

\[ \ddot{\theta} = -\frac{g}{l} \sin(\theta). \]

Define state variables \( x_1 = \theta \), \( x_2 = \dot{\theta} \). We have the nonlinear (autonomous) state-space equations:

\[
\begin{cases}
\dot{x}_1 = x_2 \\
\dot{x}_2 = -\frac{g}{l} \sin(x_1).
\end{cases}
\]

These equations tell us "where the state is going" as a function of "where it is now". One way to visualize this is by making a "phase portrait", which is a plot of trajectories of the system in \( \mathbb{R}^n \) (i.e., plot of the state itself).

For this pendulum, it amounts to a 2-D plot (\( x_1 \) vs \( x_2 \)).
Phase portrait for $\dot{\theta} = -\frac{g}{l} \sin \theta$.

Things to note:

- The picture is periodic (repeats every $2\pi$ in the $x_1$ direction).

- Near 0, $\pm 2\pi$, $\pm 4\pi$, etc, we have a **stable equilibrium** (more about this later!)

- And if we zoom in, it looks like a circle: (ellipse)

- Near $\pm \pi$, $\pm 3\pi$, etc, we have an **unstable equilibrium** and if we zoom in, it looks like a hyperbola!

- These cases correspond respectively to:

  - and **unstable**

  - stable
We can capture the behavior "near" a point of interest by using a Taylor expansion.

**A.** Near $x_1 = 0$, ($\theta = 0$), write $\theta = \Theta + \delta \theta$.

Then $\sin \theta = \sin \delta \theta \approx \delta \theta$, and $\ddot{\theta} = \delta \ddot{\theta}$

$$\Rightarrow \delta \ddot{\theta} = -\frac{g}{l} \delta \theta \Rightarrow \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -\frac{g}{l} x_1 \end{cases} \Rightarrow \dot{x} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} x$$

**A.** Near $x_1 = \pi$ ($\theta = \pi$), write $\theta = \pi + \delta \theta$.

Then $\sin \theta = \sin (\pi + \delta \theta) \approx -\delta \theta$, and $\ddot{\theta} = \delta \ddot{\theta}$

$$\Rightarrow \delta \ddot{\theta} = \frac{g}{l} \delta \theta \Rightarrow \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -\frac{g}{l} x_1 \end{cases} \Rightarrow \dot{x} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} x$$

**IMPORTANT:** when we make these approximations, $x \neq \begin{bmatrix} \theta \\ \delta \theta \end{bmatrix}$. We define $x$ as the increment or the change from normal.

So in the cases above, $x = \begin{bmatrix} \delta \theta \\ \delta \ddot{\theta} \end{bmatrix}$. 
In general, if we want to linearize a state-space equation of the form:

\[
\begin{align*}
\dot{x} &= f(x, u) \\
y &= g(x, u)
\end{align*}
\]

then we can expand this in terms of \(x_1, \ldots, x_n\) and \(u_1, \ldots, u_m\) and \(y_1, \ldots, y_p\). So the equations look like:

\[
\begin{align*}
\dot{x}_1 &= f_1(x_1, x_2, \ldots, x_n, u_1, \ldots, u_m) \\
\vdots \\
\dot{x}_n &= f_n(x_1, \ldots, x_n, u_1, \ldots, u_m) \\
y_1 &= g_1(x_1, \ldots, x_n, u_1, \ldots, u_m) \\
\vdots \\
y_p &= g_p(x_1, \ldots, x_n, u_1, \ldots, u_m).
\end{align*}
\]

Recall Taylor expansion in high dimension:

if we have \(z = h(x_1, \ldots, x_n)\) then we can expand about the nominal point \((\tilde{x}_1, \ldots, \tilde{x}_n)\) and write:

\[
z \approx \tilde{z} + \frac{\partial h}{\partial x_1}(\tilde{x}) (x_1 - \tilde{x}_1) + \cdots + \frac{\partial h}{\partial x_n}(\tilde{x}) (x_n - \tilde{x}_n),
\]

where \(\tilde{z} = h(\tilde{x})\).
Applying this to every equation, we have:

$$
\begin{bmatrix}
\delta x_1 \\
\vdots \\
\delta x_n
\end{bmatrix} = 
\begin{bmatrix}
\frac{\partial f_1(x,\tilde{u})}{\partial x_1} & \cdots & \frac{\partial f_1(x,\tilde{u})}{\partial x_n} \\
\vdots & \ddots & \vdots \\
\frac{\partial f_n(x,\tilde{u})}{\partial x_1} & \cdots & \frac{\partial f_n(x,\tilde{u})}{\partial x_n}
\end{bmatrix}
\begin{bmatrix}
\delta x_1 \\
\vdots \\
\delta x_n
\end{bmatrix} + 
\begin{bmatrix}
\frac{\partial f_1(x,\tilde{u})}{\partial u_1} & \cdots & \frac{\partial f_1(x,\tilde{u})}{\partial u_m} \\
\vdots & \ddots & \vdots \\
\frac{\partial f_n(x,\tilde{u})}{\partial u_1} & \cdots & \frac{\partial f_n(x,\tilde{u})}{\partial u_m}
\end{bmatrix}
\begin{bmatrix}
\delta u_1 \\
\vdots \\
\delta u_m
\end{bmatrix}
$$

$$
\begin{bmatrix}
\delta y_1 \\
\vdots \\
\delta y_p
\end{bmatrix} = 
\begin{bmatrix}
\frac{\partial g_1(x,\tilde{u})}{\partial x_1} & \cdots & \frac{\partial g_1(x,\tilde{u})}{\partial x_n} \\
\vdots & \ddots & \vdots \\
\frac{\partial g_p(x,\tilde{u})}{\partial x_1} & \cdots & \frac{\partial g_p(x,\tilde{u})}{\partial x_n}
\end{bmatrix}
\begin{bmatrix}
\delta x_1 \\
\vdots \\
\delta x_n
\end{bmatrix} + 
\begin{bmatrix}
\frac{\partial g_1(x,\tilde{u})}{\partial u_1} & \cdots & \frac{\partial g_1(x,\tilde{u})}{\partial u_m} \\
\vdots & \ddots & \vdots \\
\frac{\partial g_p(x,\tilde{u})}{\partial u_1} & \cdots & \frac{\partial g_p(x,\tilde{u})}{\partial u_m}
\end{bmatrix}
\begin{bmatrix}
\delta u_1 \\
\vdots \\
\delta u_m
\end{bmatrix}
$$

If we use the compact notation $\frac{\partial f}{\partial x} = \begin{bmatrix}
\frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\
\vdots & \ddots & \vdots \\
\frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n}
\end{bmatrix}$, then we simply have:

$$
\begin{align*}
\delta x &= \underbrace{\begin{bmatrix}
\frac{\partial f}{\partial x}(x,\tilde{u})
\end{bmatrix}}_{A} \delta x + \underbrace{\begin{bmatrix}
\frac{\partial f}{\partial u}(x,\tilde{u})
\end{bmatrix}}_{B} \delta u \\
\delta y &= \underbrace{\begin{bmatrix}
\frac{\partial g}{\partial x}(x,\tilde{u})
\end{bmatrix}}_{C} \delta x + \underbrace{\begin{bmatrix}
\frac{\partial g}{\partial u}(x,\tilde{u})
\end{bmatrix}}_{D} \delta u
\end{align*}
$$