SLIDES 9b – System Modelling for Control and Estimation

In the past few sessions, we have looked in detail at designing controllers and observers. The core equation of state-space control is

$$\boldsymbol{u} = -\boldsymbol{K}\mathbf{x}$$

where we used full state feedback to control the system, including regulator and tracking problems, and using an observer to estimate states.

We have focused attention on SISO systems, but state-space control is equally comfortable with MIMO systems. A controller with multiple inputs has greater freedom to give stable and high-performance control, while a multi-input observer can give improved estimation of the states (and cross-check between sensors to detect faults).

State-Space Control

For MIMO systems it is more appropriate to use optimal control methods mentioned before: the Kalman filter (optimal observer) and LQR (optimal controller).

The structure of the controller and observer is exactly the same when using LQR and Kalman filtering, but the method of finding $\boldsymbol{K}$ and $\boldsymbol{K_e}$ is different.

Once understood, these methods are easily extended to MIMO systems.

LQR is covered in a very nice webinar from MathWorks

Kalman filters are covered in a series of videos which also go over the 'normal' state observer.

Now we go 'back to basics' – modelling electrical and mechanical systems … to find A, B, C, D and allow us to apply the knowledge of state-space control (pole placement, LQR, Kalman filters etc.)

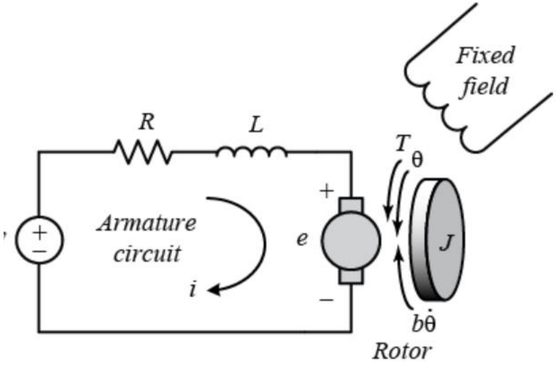
EXAMPLE 1 – POSITION CONTROL USING A DC ELECTRIC MOTOR

A DC motor generates a magnetic field in the rotor windings (armature) which then interacts with a fixed magnetic field surrounding it (stator)

The active part is the armature circuit which switches as the contacts (brushes) touch different parts of the commutator.

The motor is controlled by the applied voltage $\boldsymbol{V(t)}$, with torque proportional to the current in the armature

$$\boldsymbol{T = K_t i}$$



Other equations based on electromagnetic principles:

back-emf

$$\boldsymbol{e = K_b\,\dot{\theta}}$$

rotor dynamics

$$\boldsymbol{J\ddot{\theta} + b\dot{\theta} = K_t i}$$

voltage balance

$$\boldsymbol{L\,\frac{di}{dt} + Ri = V - K_b\dot{\theta}}$$

and the two motor constants are equal, $\boldsymbol{K_b = K_t = K}$.

Define state variables: $\boldsymbol{x_1 = \theta, x_2 = \dot{\theta}, x_3 = i}$. Then the state equations are as follows

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = -bJ^{-1}x_2 - KJ^{-1}x_3$$

$$\dot{x}_3 = -KL^{-1}x_2 - RL^{-1}x_3 + L^{-1}V(t)$$

Hence, with $u = V$ (input voltage)

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & -bJ^{-1} & KJ^{-1} \\ 0 & -KL^{-1} & -RL^{-1} \end{pmatrix}, \qquad B = \begin{pmatrix} 0 \\ 0 \\ L^{-1} \end{pmatrix}$$

For position control the output is

$$y = \theta$$

so

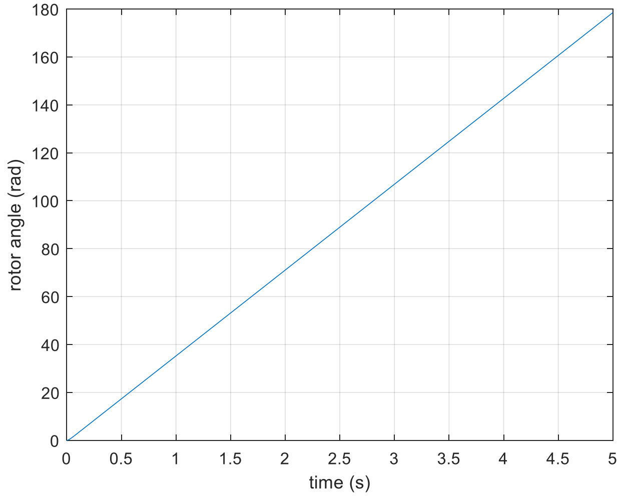
$$C = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}, \qquad D = 0$$

The following shows some particular data and computes the transfer function and other basics [see example_9b1.m]

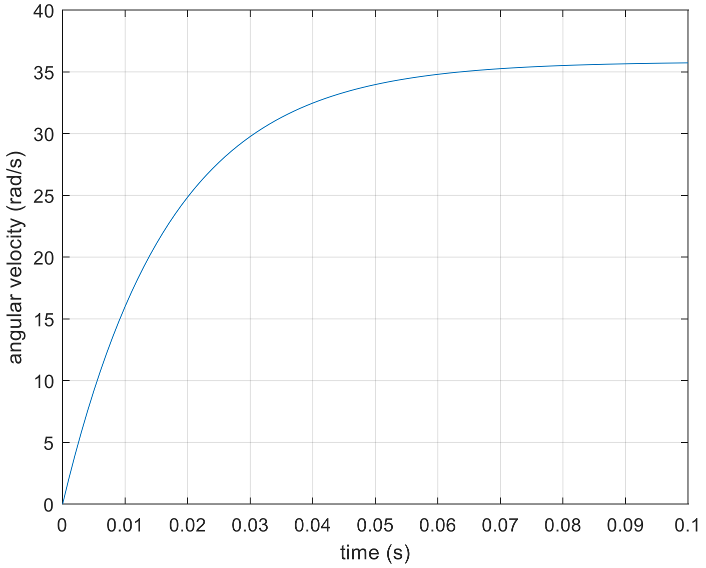
```
J = 3.2284E-6;
b = 3.5077E-6;
K = 0.0274;
R = 4;
L = 2.75E-6;

A=[ 0    1    0
    0 -b/J  K/J
    0 -K/L  -R/L];
B=[0 0 1/L]';
C=[1 0 0];
D=0;
```

We find the eigenvalues [0, -59.226, -1.454 x 10^6]. We could have anticipated the zero value (why) and the others are stable … one VERY fast!

To test the model we can use an open-loop step response



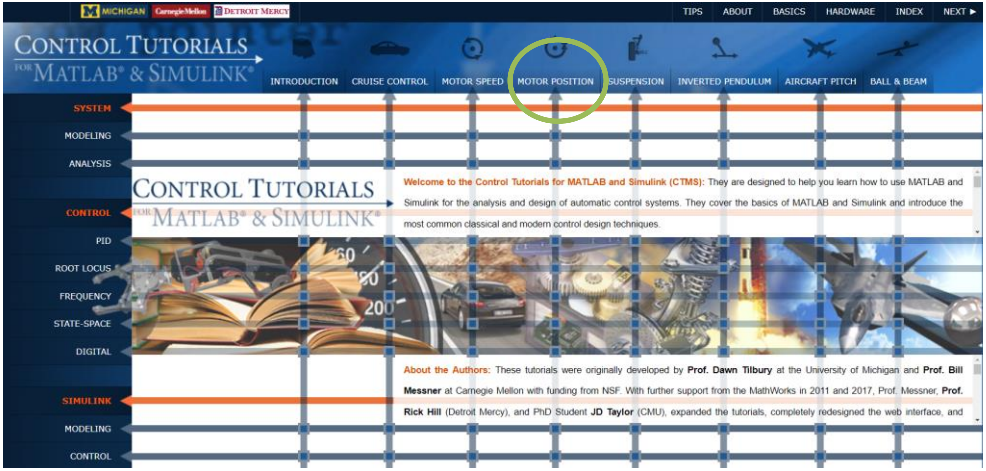
not very informative! Rotor speed is more relevant (change the C matrix to [0 1 0] …



The motor reaches its steady-state velocity in around 0.1 sec.

The above is based on the control tutorial website

https://ctms.engin.umich.edu/CTMS/index.php?aux=Home



Welcome to the Control Tutorials for MATLAB and Simulink (CTMS): They are designed to help you learn how to use MATLAB and Simulink for the analysis and design of automatic control systems. They cover the basics of MATLAB and Simulink and introduce the most common classical and modern control design techniques.

About the Authors: These tutorials were originally developed by **Prof. Dawn Tilbury** at the University of Michigan and **Prof. Bill Messner** at Carnegie Mellon with funding from NSF. With further support from the MathWorks in 2011 and 2017, Prof. Messner, **Prof. Rick Hill** (Detroit Mercy), and PhD Student **JD Taylor** (CMU), expanded the tutorials, completely redesigned the web interface, and

State-Space Control

Here we have covered the modelling section. The state-space control is also of interest (use the menu or the following link)

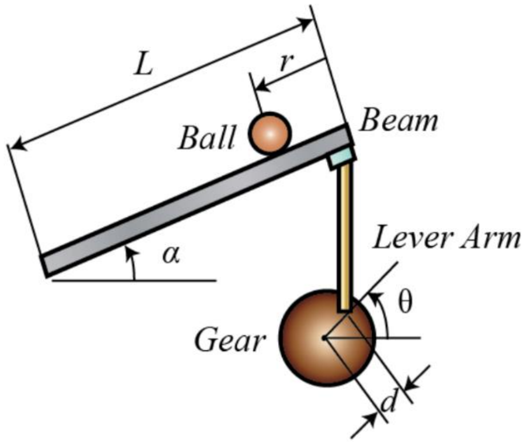
https://ctms.engin.umich.edu/CTMS/index.php?example=MotorPosition&section=ControlStateSpace

We have previously covered all the material, but it is worth taking a quick look to see what's the same and what is slightly different.

EXAMPLE 2 – BALL & BEAM POSITION CONTROL

This example is also taken from the University of Michigan tutorials. Here we give more details about the modelling.

The smaller gear wheel tilts the beak which causes the ball to roll. The aim is to move the ball to any desired location on the beam.



We assume θ is the control variable, we can command it directly. *What happens in reality? When is this assumption OK?*

In the setup, the beam is horizontal when $\boldsymbol{\theta} = \mathbf{0}$. For small angles it's clear that $\boldsymbol{\theta}$ and $\boldsymbol{\alpha}$ are proportional. Equating the change in height of the lever arm

$$d\theta = L\alpha$$

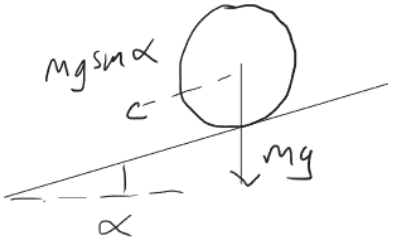
It's quite common to make simplifying assumptions during the modelling. The detailed geometry of the gear and lever arm can be a little complicated.

Even if we assume the lever arm is long enough to remain vertical during the motion, the correct equation for vertical motion is

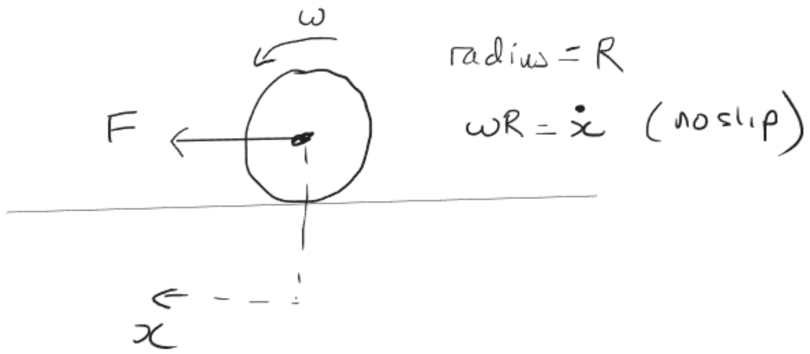
$$d \sin\theta = \boldsymbol{L} \sin \boldsymbol{\alpha}$$

The weight of the ball provides a force down the slope

$$F = mg\,\mathbf{sin}\,\alpha \approx mg\alpha$$

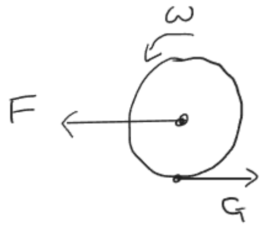


Detailed nonlinear modelling of the system is rather complex (but see below). Here we simplify the problem by treating motion relative to the beam and ignoring the dynamics of the beam itself



Although there is just one degree of freedom (x) the mechanics is complicated by the coupled translation and rotation

There is a reaction force G at the contact point to keep the wheel rotating at the correct angular velocity ω.



$$F - G = m\ddot{x}$$
$$GR = J\dot{\omega}$$
$$\dot{\omega}R = \ddot{x}$$

The Newton-Euler equations are given, together with the derivative of the 'no-slip' condition.

We can then eliminate G and $\dot{\omega}$ to find

$$F = (m + J/R^2)\ddot{x}$$

Hence the moment of inertia creates an additional 'effective mass' for the ball

$$m' = m + J/R^2$$

Hence the (approximate) equation of motion for the beam and ball is

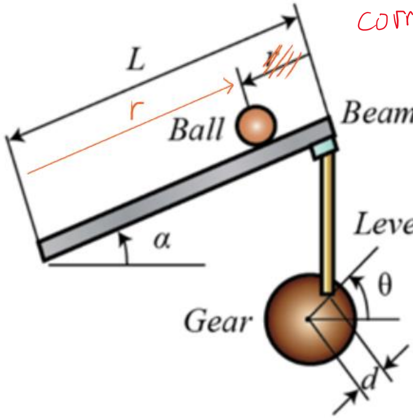
$$-mg\alpha = (m + J/R^2)\ddot{r}$$

or in terms of the control input θ

$$(m + J/R^2)\ddot{r} = -\frac{mgd}{L}\theta$$

[There is a slight error in online,
r should be measured up the slope]

corrected



$$\left(\frac{J}{R^2} + m\right)\ddot{r} = -mg\frac{d}{L}\theta$$

online version

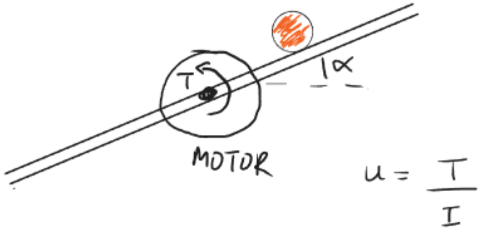
The state-space equations are easily found ($x_1 = r, x_2 = \dot{r}$)

$$\begin{bmatrix} \dot{r} \\ \ddot{r} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} r \\ \dot{r} \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{mgd}{L\left(\frac{J}{R^2}+m\right)} \end{bmatrix} \theta$$

[online version]

If the control input is from controlling the motor torque we will have a 4th order state-space system (see online)

Here u is the motor torque divided by the moment of inertia of beam plus motor.



What additional physical effect is being ignored?

Summing Up

- In this module we tend to ‘grab’ transfer functions $G(s)$ and state-space matrices ‘out of the blue’. Here we have gone through a reminder of where these objects come from.
- Modelling for control commonly involves simplifying assumptions so the equations of motion can be put in linear (A, B, C, D) format
- Modelling mechanical systems, such as the ball and beam, often uses more advanced techniques – especially the Lagrangian equations of motion: $\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} = 0.$
- For the Lagrange method, the effort is concentrated on finding the functions for kinetic energy $T(q, \dot{q})$ and potential energy $V(q)$, then substituting $L = T - V$ then performing the differentiation in a routine manner.