RBT350
GATEWAY TO ROBOTICS

Introduction to Control

Mitch Pryor

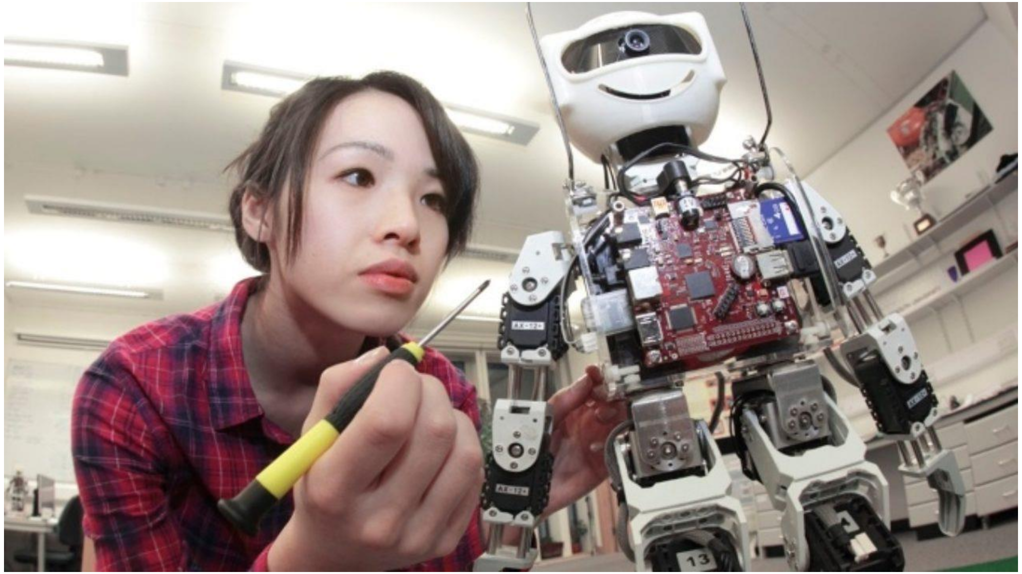
Research Professor in Mechanical Engineering

Big Recap – How to Build our Robots?

- Statics, Friction, Grasping
  - Newton, FBD, wrenches, stiction, kinetic friction, viscous friction
- Physics of Materials
  - Physics of deformable bodies, strain-stress diagrams
- Articulations
  - Types of joints, Grueber's Formula
- Analog Electronics
  - Current, Voltage, R, C, L, Kirchoff's laws, power
- Digital Electronics
  - Transistors, gates, truth tables, AND, OR, … Boolean algebra
- State Machines
  - DFA, Deterministic/Stochastic DFAs, Petri Nets, Hybrid Automata
- Mechatronics
  - Types of actuators, motors, torque/speed curve, encoders

Everything to build a robot

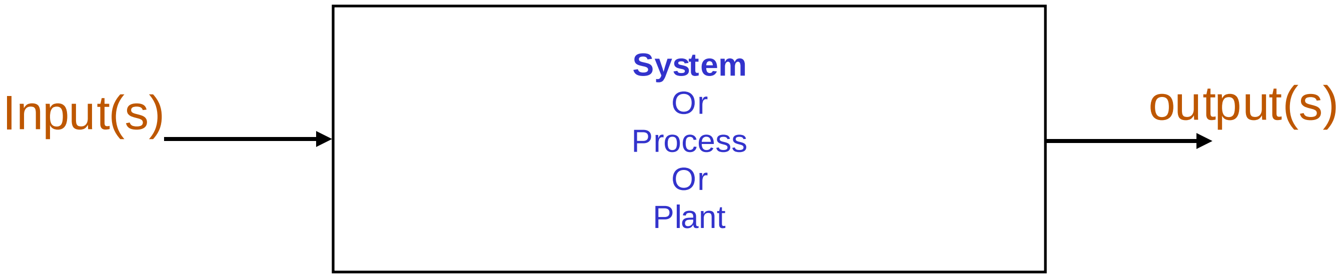
But how do we move it?



What will you learn today?

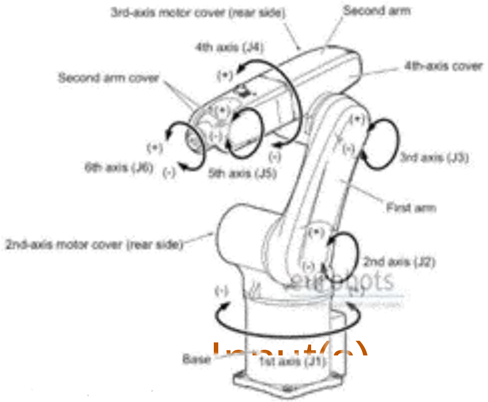
- What is control and what is it for?
- Controller Strategies
  - Bang-Bang control
  - Proportional control (P)
  - Integral control (I)
  - Derivative control (D)
- State-Space Representation

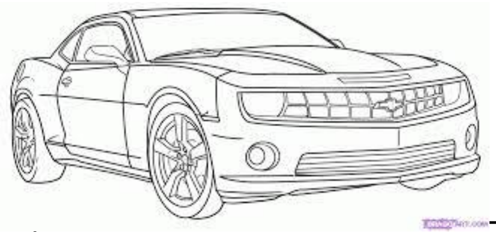
It all starts with…

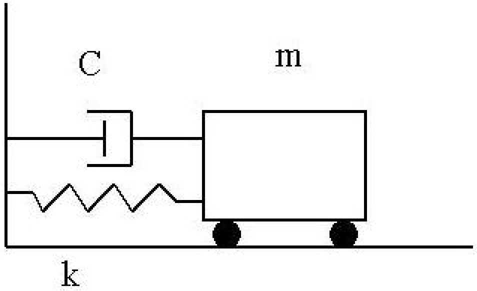


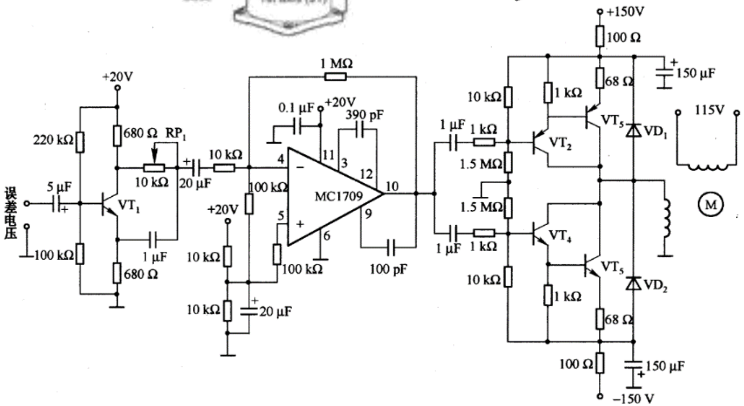
Filling in this box for circuits, mechanical systems, etc. is something you study in many other courses.

So many examples…

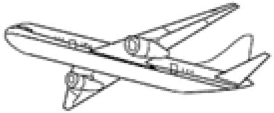


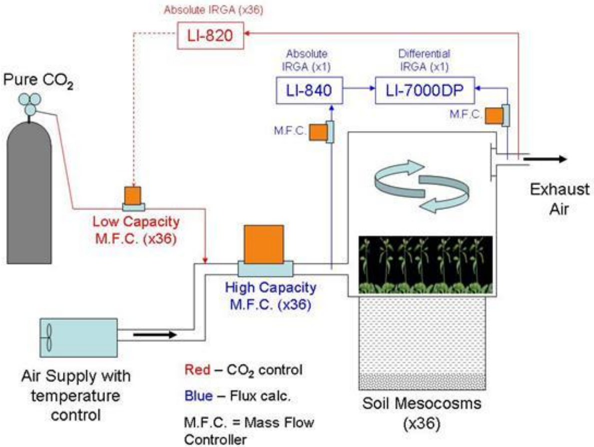






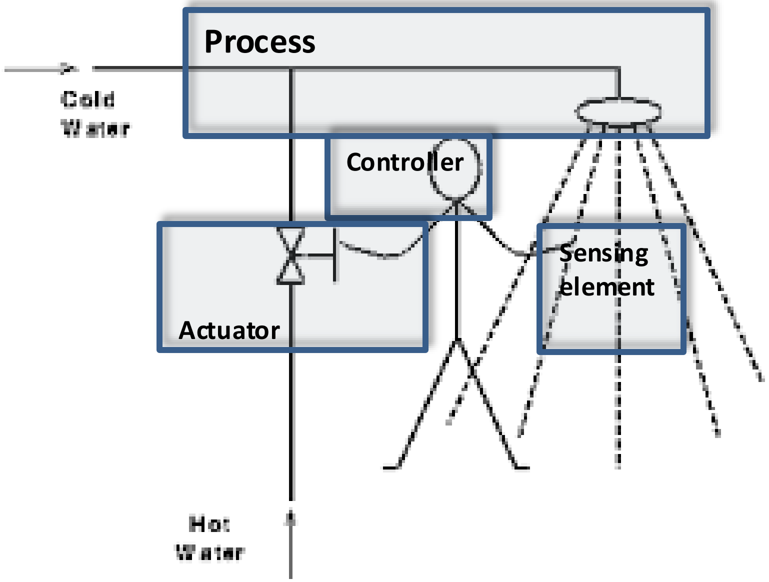
System
Or
Process
Or
Plant

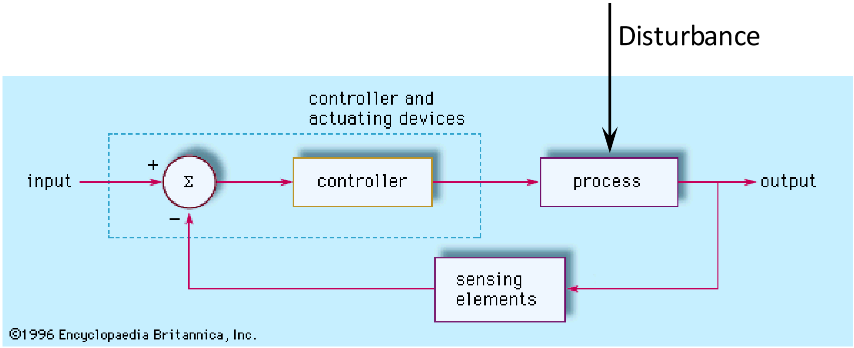




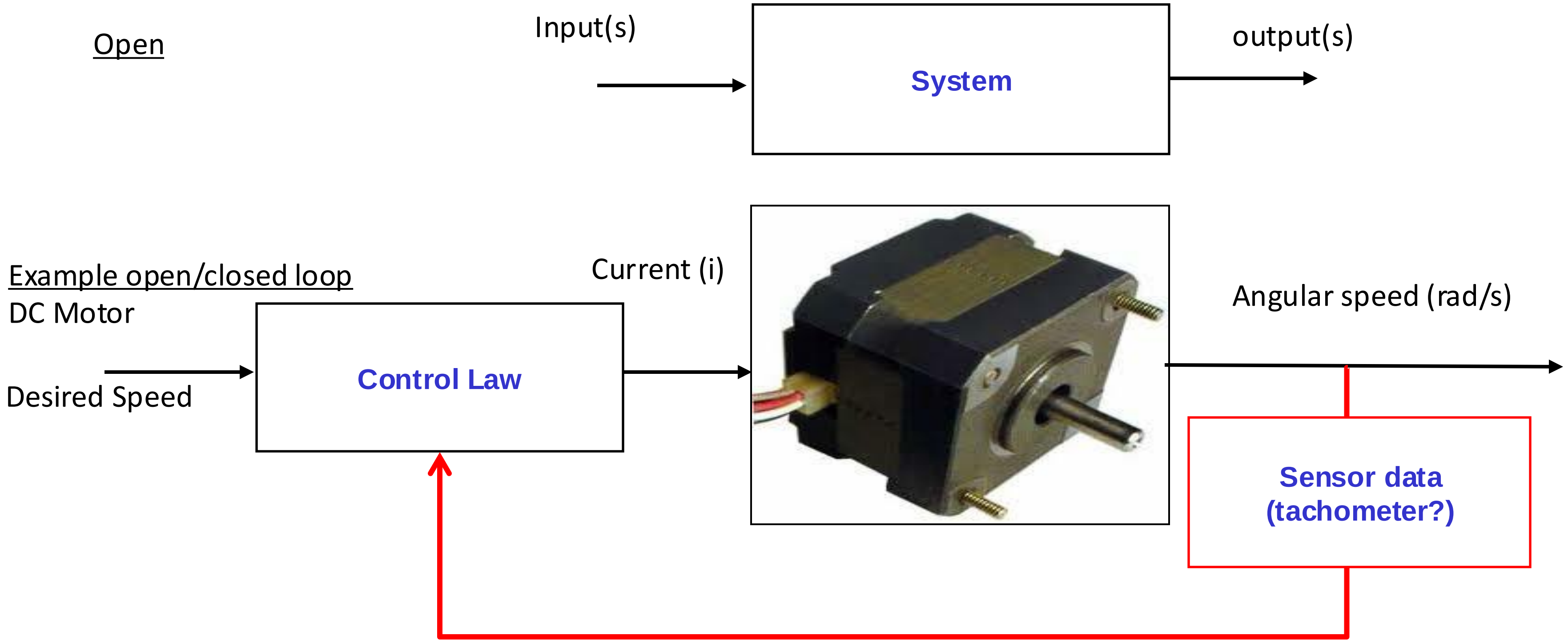
So now….

Our goal: Understand something you do (almost) everyday.





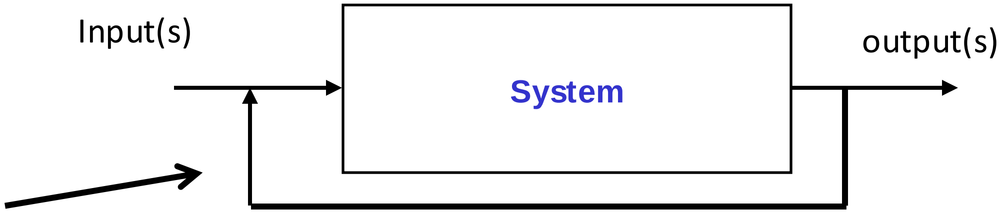
Open vs. Closed loop systems…



Close the loop

Open vs. Closed loop systems…

<u>Closed</u>

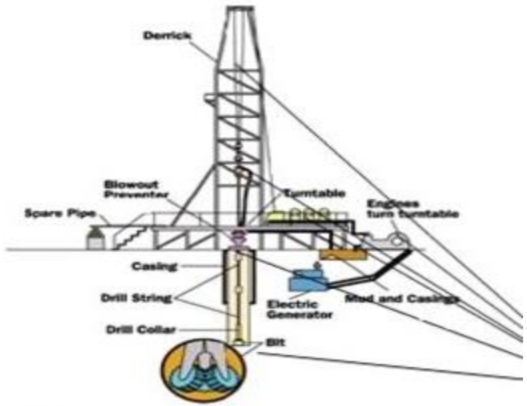


“Feedback”

SISO, MISO, MIMO Systems…

Single Input Single Output (SISO)

gas → speed

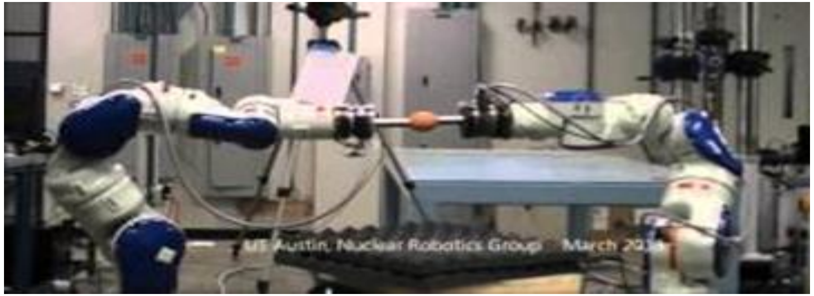


Multi-Input Single Output (MISO)

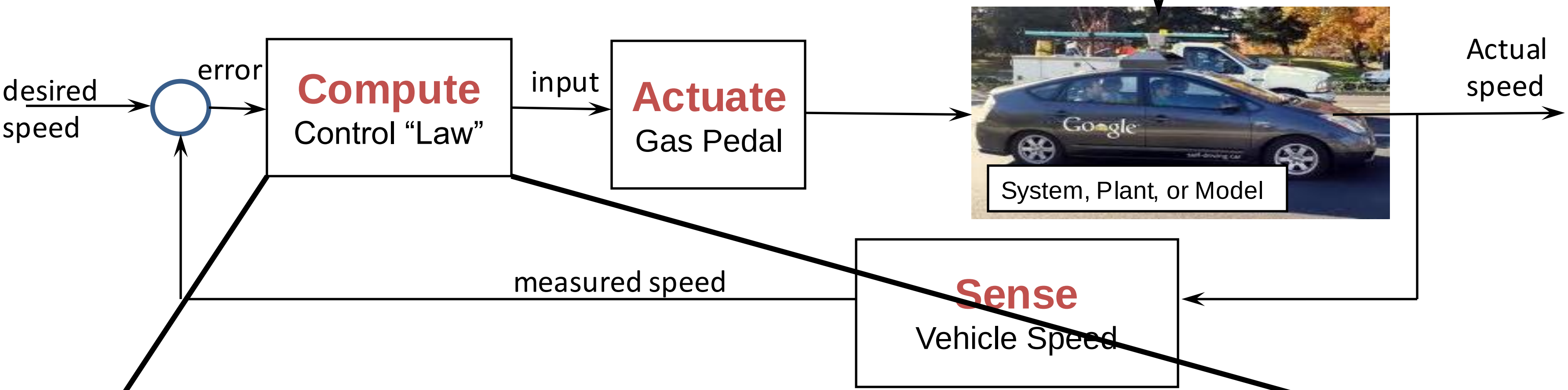
Weight on Bit (WOB), RPM → ROP (Rate of Penetration)

Multi-Input Multi-Output (MIMO)

7 joint currents → wrench (3 translation forces and 3 moments)



Our Controller's Objective

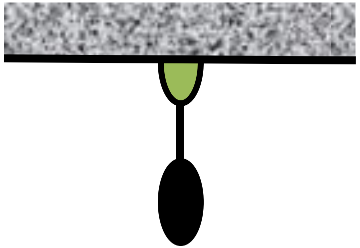


Goals

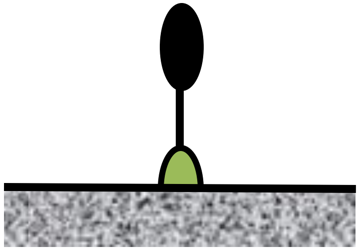
- ❑Stability: system maintains desired operating point (hold steady speed, tracks a particular trajectory)
- ❑Robustness: system tolerates perturbations in dynamics (mass, drag, etc)
- ❑Performance: system responds rapidly to changes (accelerate to 6 m/sec)

Stable vs unstable systems

In a stable systems, small perturbations will maintain the status quo



In unstable systems, small perturbations can disrupt the status quo



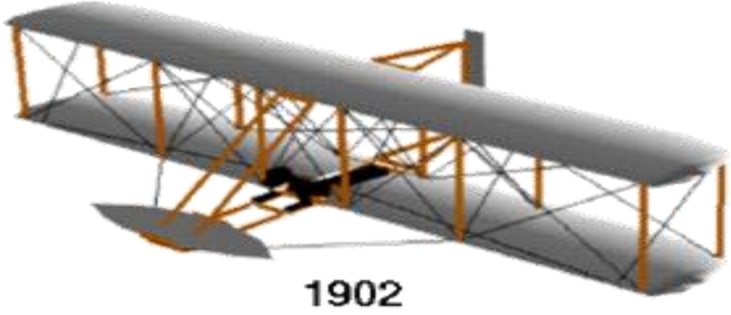
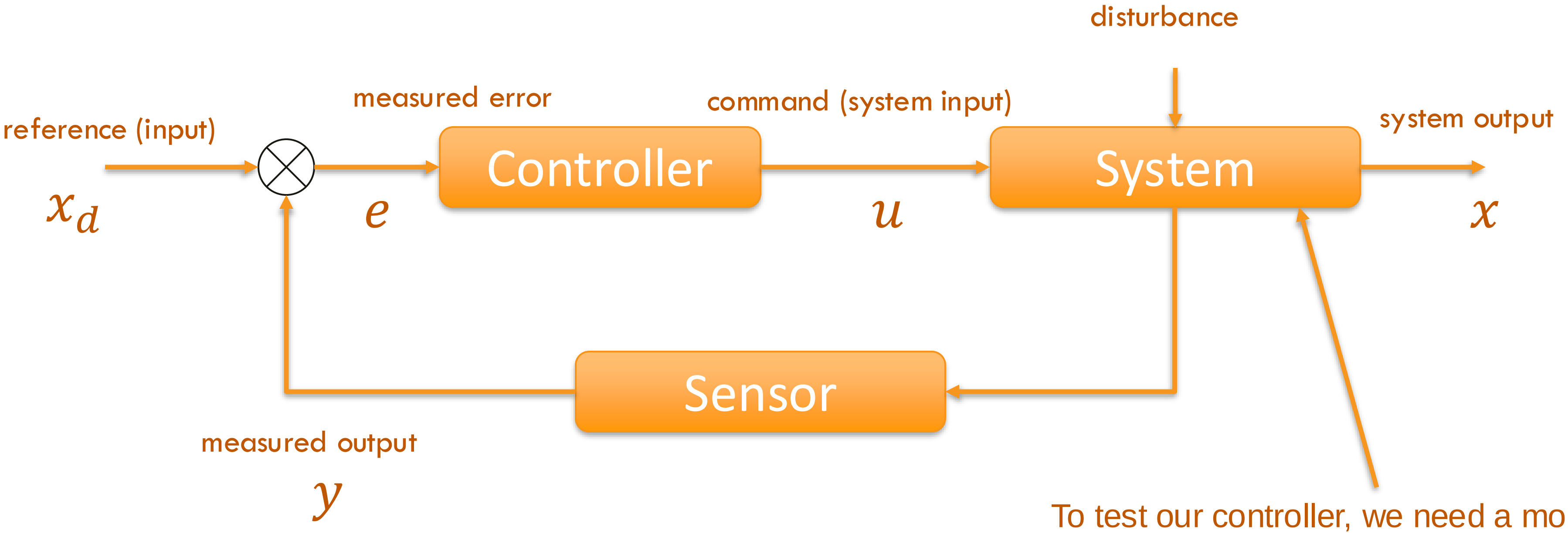


Diagram of the System and Controller



To test our controller, we need a model.

What is control and what is it for?

$$s\dot{m}th = \frac{\delta smth}{\delta t}$$

- Control deals with commanding a dynamical system, a system that changes its state over time
- With control, we influence those changes, ideally towards our desires
- Control is a mechanism to produce **inputs** to the dynamical system to try to guide its **state** towards a desired state
- Of course, we are assuming that the control has an effect in the state: $\frac{\delta F}{\delta u} \neq 0$

$$\dot{x} = F(x, u)$$

$$y = G(x)$$

We define this!

$$u = H_i(y)$$

$$\dot{x} = F(x, H_i(G(x)))$$

A very special case: Linear Dynamics

- Many systems in robotics can be assumed to have linear dynamics
- Many others can be assumed to be ”linearizable”
- If the system is linear, we can use matrix algebra:

$$\dot{x} = F(x, u) = Ax + Bu$$

- If the relationship between the state and the observation is also linear, we can do the same here:

$$y = G(x) = Gx$$

State-space model

- mathematical model of a system's inputs, outputs, and states represented as a set of 1st order ODEs.

Let,

$x(t) \in \mathbb{R}^n$ State vector

$u(t) \in \mathbb{R}^p$ Input vector

$y(t) \in \mathbb{R}^q$ Output (or measured) vector

State-space model

- So, for a LTI SISO system…

nxn matrix

*n*x1 state vector

*n*x1 vector

scalar input

$$\dot{x} = Ax + Bu$$

How the states change due to the current values of the states and due to any inputs.

1xn row vector

*n*x1 state vector

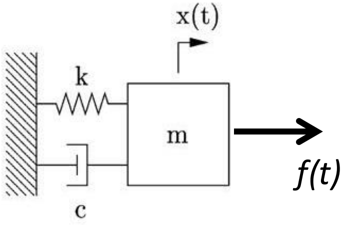
$$y = Gx$$

Provides a measured value(s) in terms of the states

$$x(t) \in \mathbb{R}^n$$
$$u(t) \in \mathbb{R}^p$$
$$y(t) \in \mathbb{R}^q$$

Mass Spring Damper Example

<u>Given</u>: Convert the EOM (equations of motion) model for a mass-spring-damper (MSD) system to a state-space model where the position is the measured output.



$$ma = \overset{+}{\rightarrow}\sum F_x$$

$$m\ddot{x} = -c\dot{x} - kx + f(t)$$

We control this (more or less)!

<u>Solve</u>:

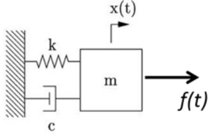
<u>Step 1</u>: Write the ODE(s) in the form:

$$\frac{d^n x}{dt^n} + a_1 \frac{d^{n-1}x}{dt^{n-1}} + a_2 \frac{d^{n-2}x}{dt^{n-2}} + \cdots + a_{n-1}\frac{dx}{dt} + a_n x = u$$

which in this case is…

$$\ddot{x} + \frac{c}{m}\dot{x} + \frac{k}{m}x = \frac{f(t)}{m} = u$$

Mass Spring Damper Example



Result from step 1.

$$\ddot{x} + \frac{c}{m}\dot{x} + \frac{k}{m}x = \frac{F(t)}{m} = u$$

Step 2: Define the state variables

Let,

$$z_1 = x \qquad \dot{z}_1 = z_2$$

$$z_2 = \dot{x} \qquad \dot{z}_2 = -\frac{k}{m}z_1 - \frac{c}{m}z_2 + u$$

Step 3: Rewrite in matrix form

$$\dot{z} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{c}{m} \end{bmatrix} z + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \qquad y = \begin{bmatrix} 1 & 0 \end{bmatrix}\mathbf{z}$$

Works for any linear system!

- We can put any linear model configured as a set of ordinary differential equations (ODEs) into state-space form.
- While most of the systems we will see in this class will be like the examples given, the s-s form can be found for any set of linear ODEs. Try the following:

$$\begin{aligned}\ddot{x}_1 - 3x_2 + 2x_1 + \dot{x}_2 &= 0\\ 2\ddot{x}_2 - 3x_1 + 2\dot{x}_2 + u &= 0\end{aligned}$$

$$\dddot{x} + 3\ddot{x} + 4x - 4u = 0$$

$$\begin{aligned}\dddot{x} + 3\ddot{x} + 4x + y &= 0\\ \dot{y} - 4\ddot{x} - 4u &= 0\end{aligned}$$

- Why State-space form?
  - Utilization of linear algebra for system analysis.
  - Examination of canonical systems represented in state-space form.
  - Can focus on control of systems in a particular form instead of modeling.
  - Application of numerical algorithms to solve systems in s-s form.

Important: Discrete time system

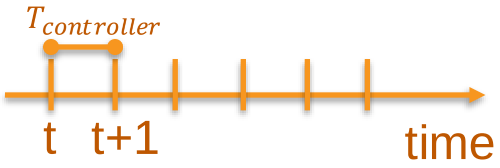
- Our controller is going to act <u>at discrete time steps</u>
- In most cases in robotics we assume a discrete system because:
  - We have a sensor that provides signals at discrete time steps
    - Camera providing images at N fps
  - We compute the controller response in a computer (discrete time!)
- Controller frequency: $f_{controller} = \frac{1}{T_{controller}}$

$$x_{t+1} = F(x_t, u_t)$$

$$y_t = G(x_t)$$

$$u_t = H_i(y_t)$$

$$x_{t+1} = F(x_t, H_i(G(u_t)))$$



VIP: Linear Dynamics in Discrete-Time Systems

$$\boxed{\begin{gathered} x_{t+1} = F(x_t, u_t) = Ax_t + Bu_t \\ \mathrm{y_t} = G(x_t) = Gx_t \end{gathered}}$$

Systems to Control

Why do we care about control?

- You are roboticists!
  - Your goal is to command the robot to achieve some motion or apply some force
  - Once you have found the desired value for the robot, how do you ensure that the robot executes it?
  - In other words, what command do we send to each motor?
  - Control!!!!

What is our goal?

- With control, we want to bring the state of the system do a desired value called *set point:* x_d
- The (feedback) controller is going to "respond" based on the <u>control error</u>, the difference between the set point and the current state of the system:

$$e = x - x_d$$

- The goal is to set u such that the error is minimized, i.e., $e = 0$

$$Find\ u\ such\ that$$
$$e(t) = x(t) - x_d$$
$$for\ t\ small$$

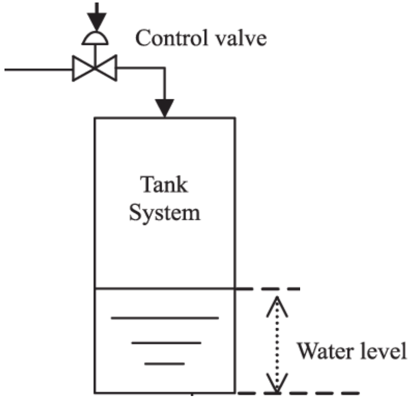
The intuition behind control

- Use action u to “push back” toward error e = 0
  - error e depends on state x (via sensors y)
- What does pushing back do?
  - Depends on the structure of the system: what can we control
  - For example, position vs. velocity vs. acceleration vs. force/torque control
- How much should we “push back”?
  - What does the magnitude of u depend on?
  - This defines different types of control implementation

Example: Velocity vs. Acceleration Control

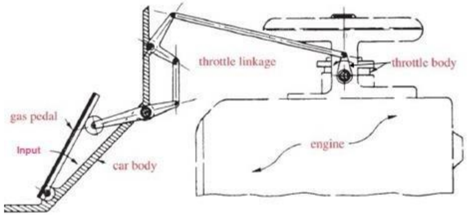
- We assume the state is (at least) position
- In our system, is u affecting velocity or acceleration?

- Example 1: Controlling a valve that regulates how fast a tank fills
  - $x = (tank\ level) = (l)$
  - $\dot{x} = (velocity\ the\ tank\ fills) = (\dot{l}) = F(x, u) = u$



Example: Velocity vs. Acceleration Control

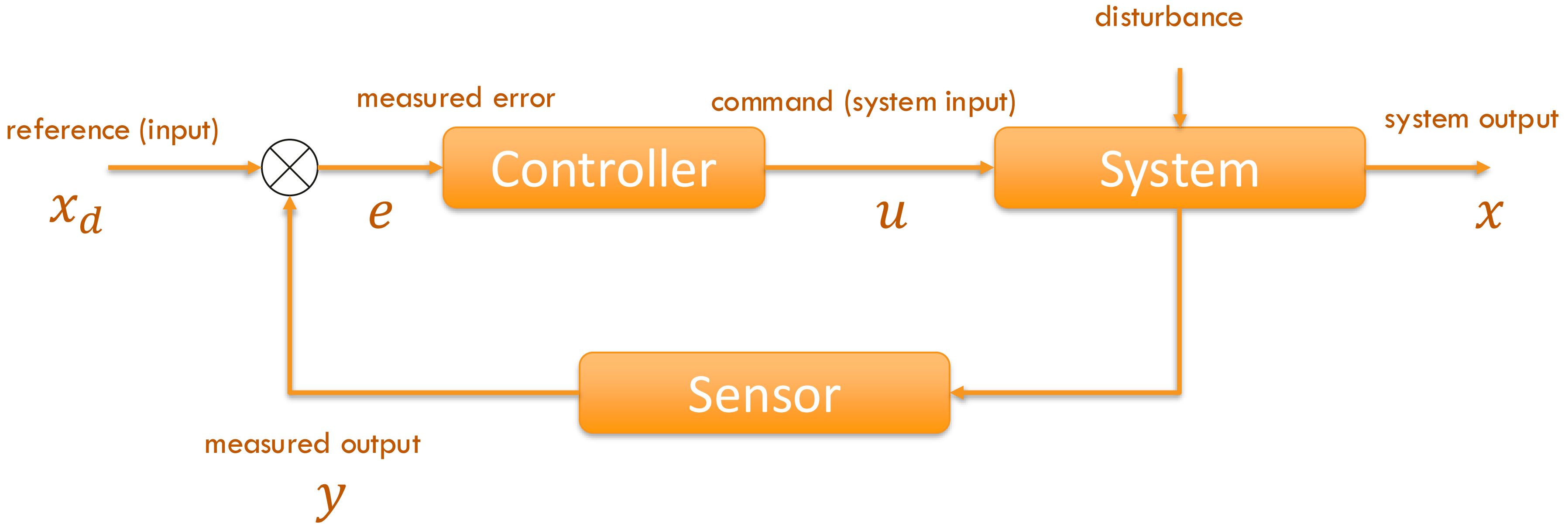
- We assume the state is (at least) position
- In our system, is u affecting velocity or acceleration?



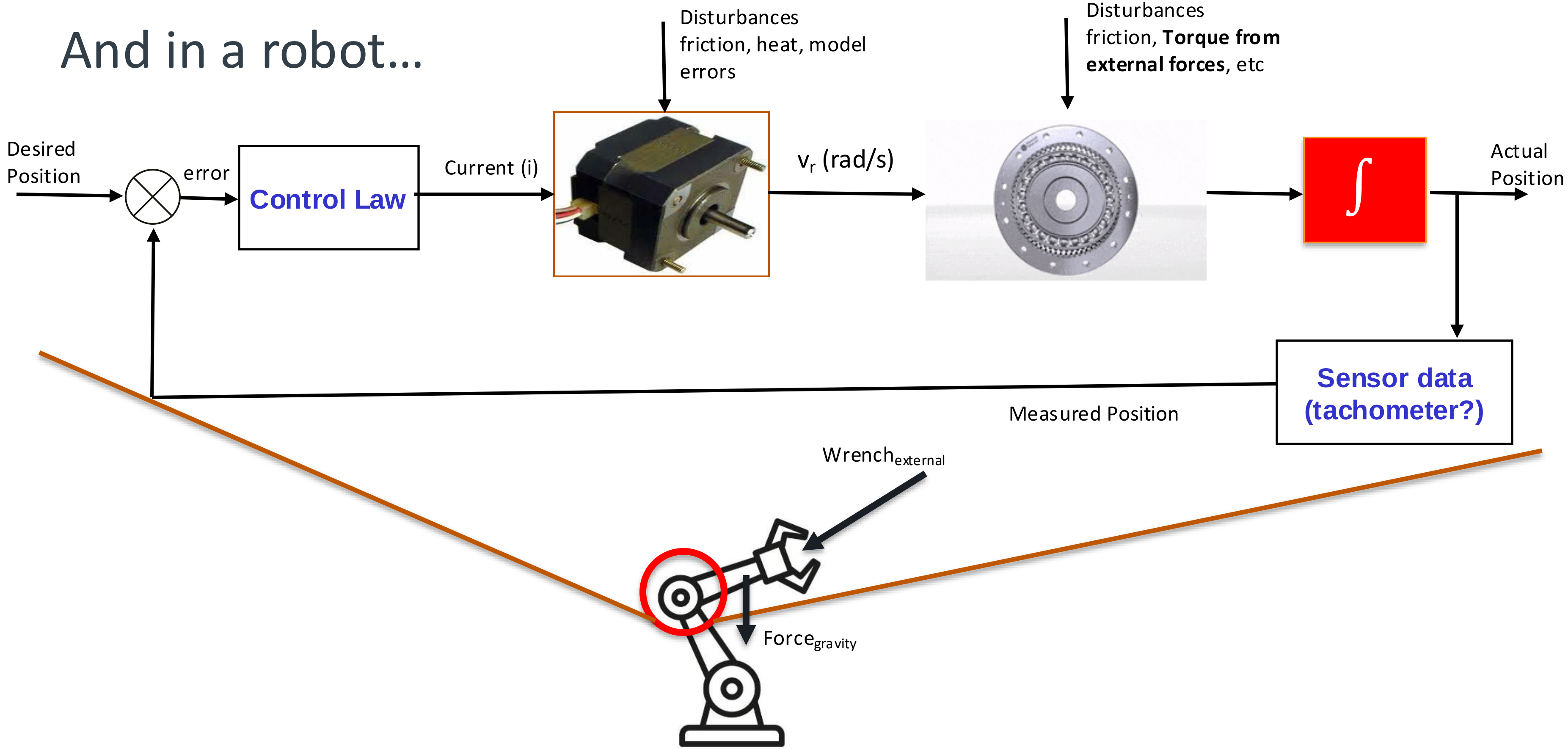
- Example 2: Controlling the motion of a car
  - $x = \begin{pmatrix} position\ of\ the\ car \\ velocity\ of\ the\ car \end{pmatrix} = \begin{pmatrix} p \\ v \end{pmatrix}$
  - $\dot{x} = \begin{pmatrix} velocity\ of\ the\ car \\ acceleration\ of\ the\ car \end{pmatrix} = \begin{pmatrix} \dot{p} \\ \dot{v} \end{pmatrix} = \begin{pmatrix} v \\ a \end{pmatrix} =$

$$= F(x,u) = \begin{pmatrix} v \\ u \end{pmatrix}$$

Diagram of the System and Controller



And in a robot...



One control loop for each joint and/or actuator

Summary

- We know how to build robots/actuators
- Now we need to control them.
  - Use action u to "push back" toward error e = 0
    - error e depends on state x (via sensors y)
- State-Space Models
  - Can be generated for work for any Linear, Time Invariant System
  - Even nonlinear systems can often be *linearized* for control.
- For robots
  - System controllers do exist (mainly in academia)
  - But system level coordination of joint/actuator controllers is more common.
  - That coordination (FK/IK) is coming soon.
- Next: Control strategies (mainly PID Control)