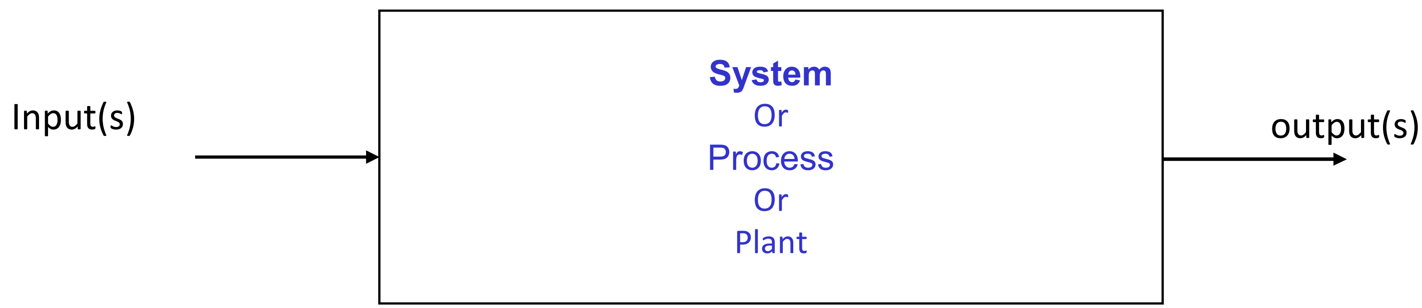
Why Feedback Control?

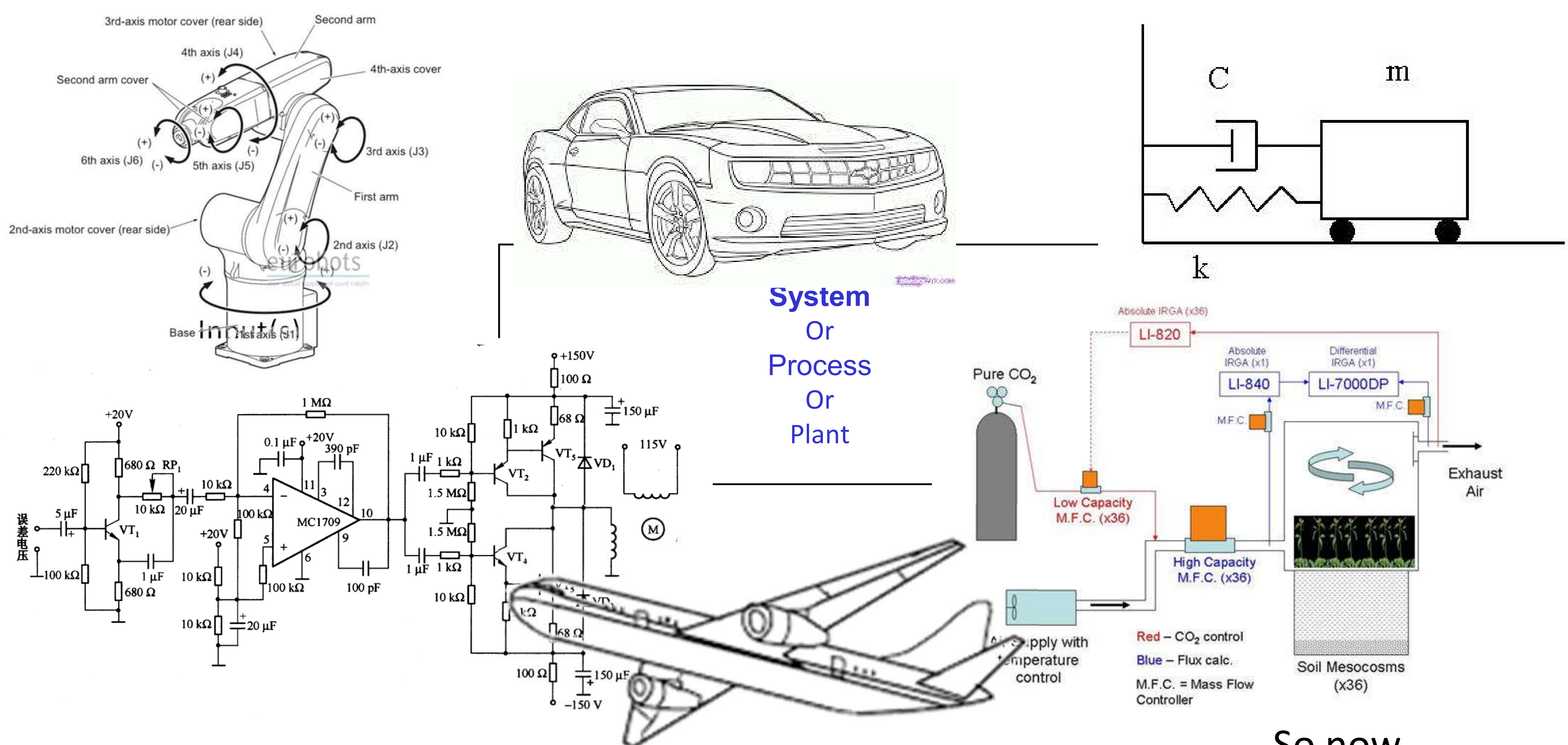
Dr. Mitch Pryor

It all starts with…



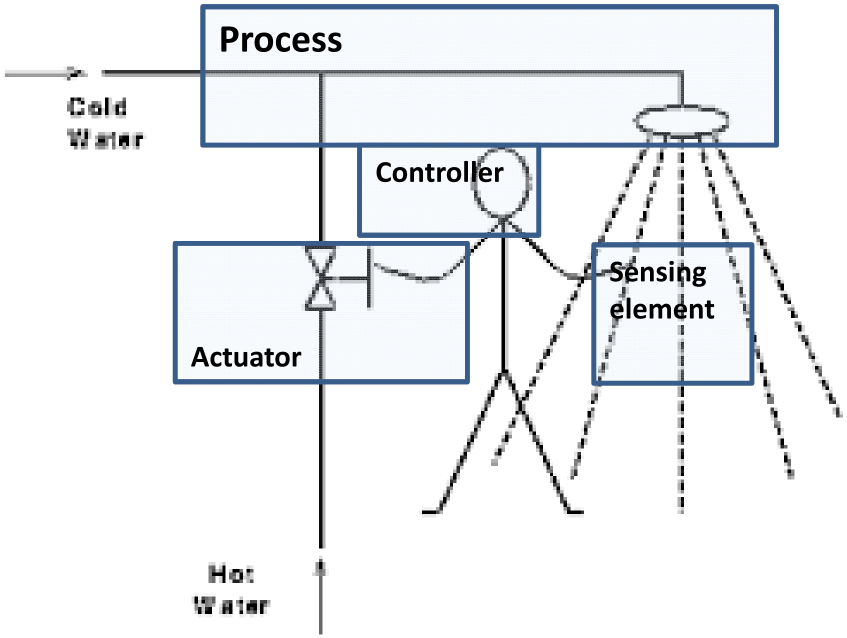
Filling in this box for dynamic mechanical systems is something you have studied in other courses.

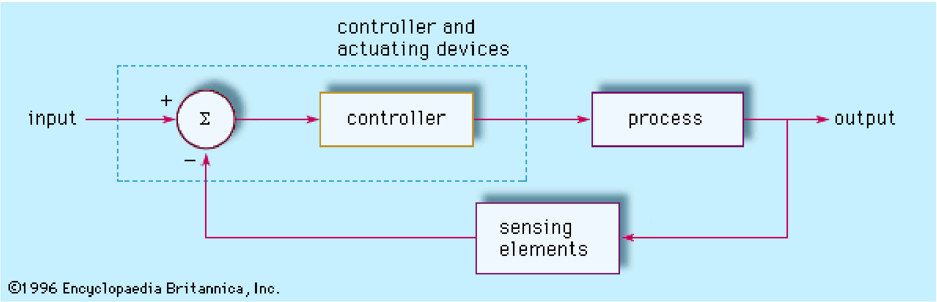
So many examples…



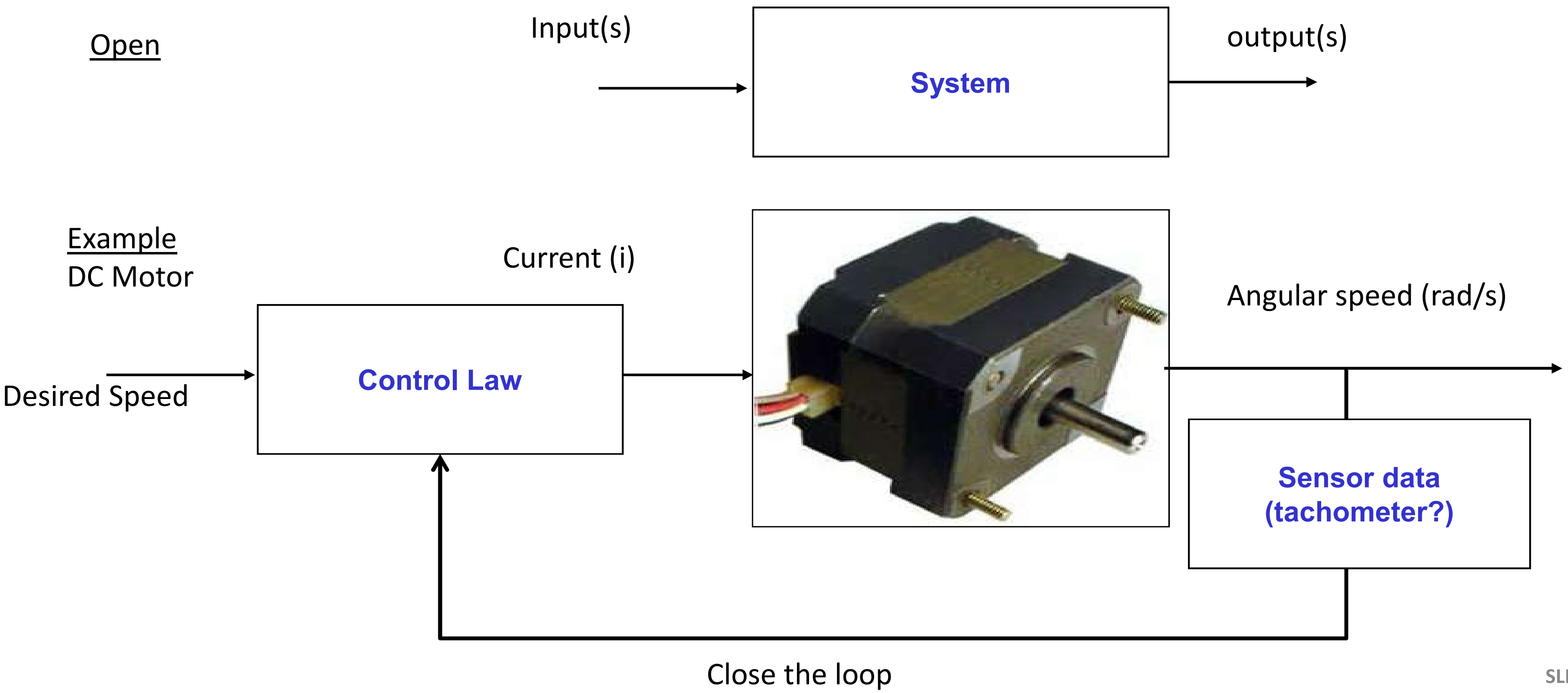
So now….

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





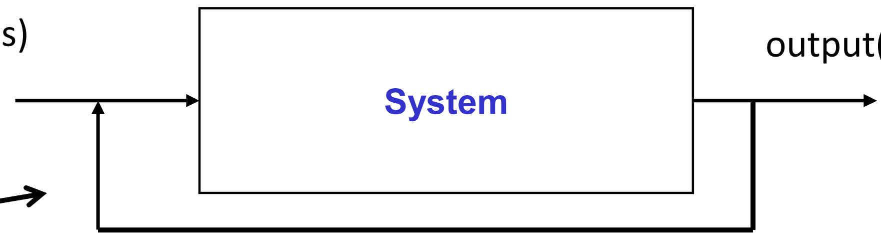
Open vs. Closed loop systems…



Open vs. Closed loop systems…

Closed

Input(s)

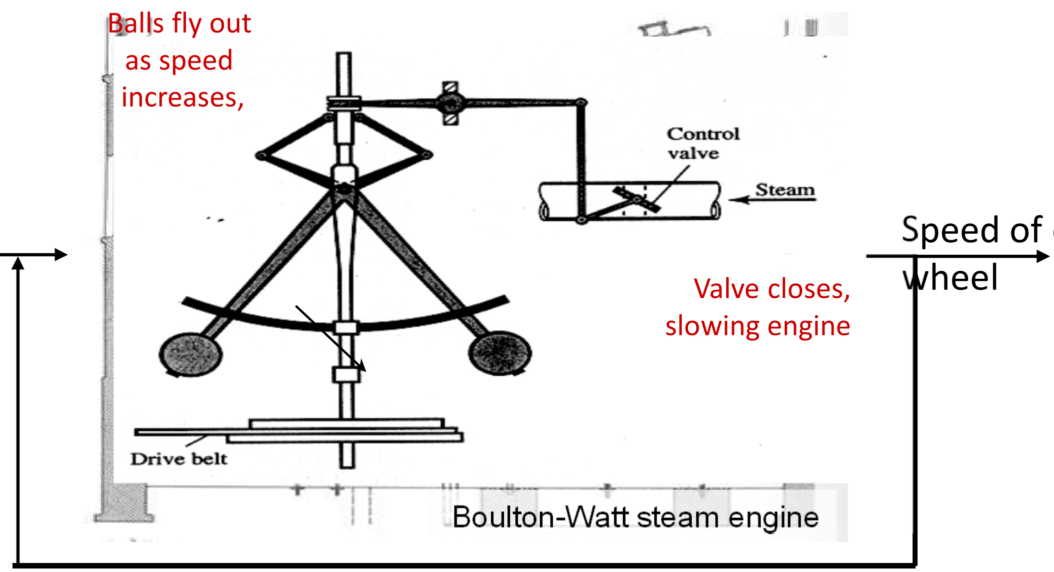


output(s)

"Feedback"

Example
"Flyball" Governor (1788)

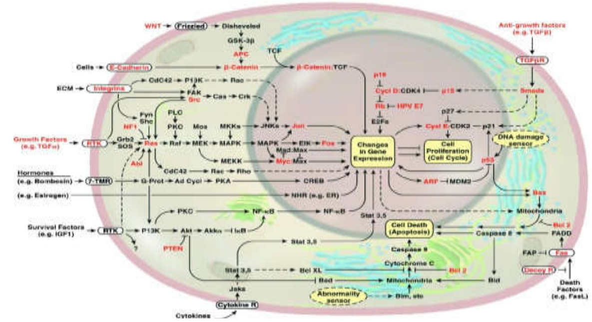
Amount of Steam

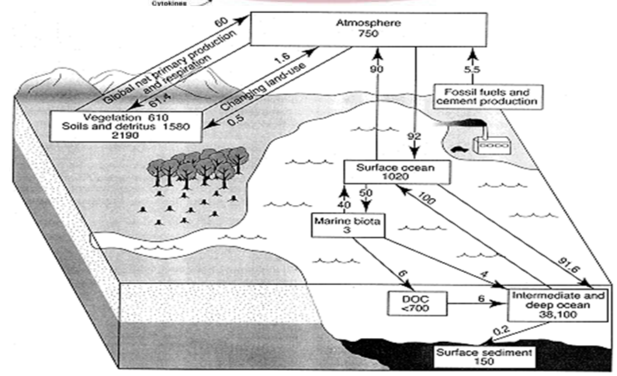


Speed of engine wheel

Other Examples of Feedback

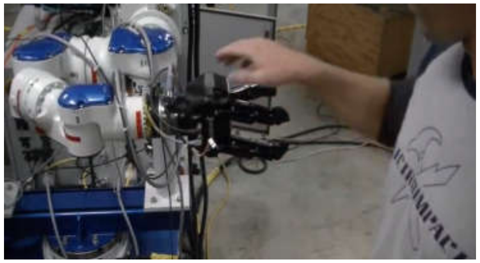
- Biological Systems
  – Physiological regulation (homeostasis)
  – Bio-molecular regulatory networks
- Environmental Systems
  – Microbial ecosystems
  – Global carbon cycle
- Financial Systems
  – Markets and exchanges
  – Supply and service chains
- Cool Toys!
  – https://youtu.be/j4OmVLc_oDw
- ECE Class Assignments
  – https://youtu.be/ocOtLYbkpjw





Even More examples…

Position & force control (path follow)
www.youtube.com/watch?v=1oZ5R1EcTe4

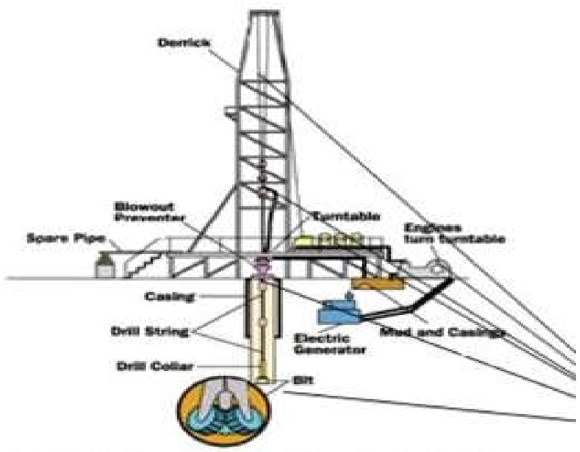


Compliance Control Demo (error response)
www.youtube.com/watch?v=w9as_hzPf5g

High level decision making
http://www.youtube.com/watch?v=FEv7b5iqoPk

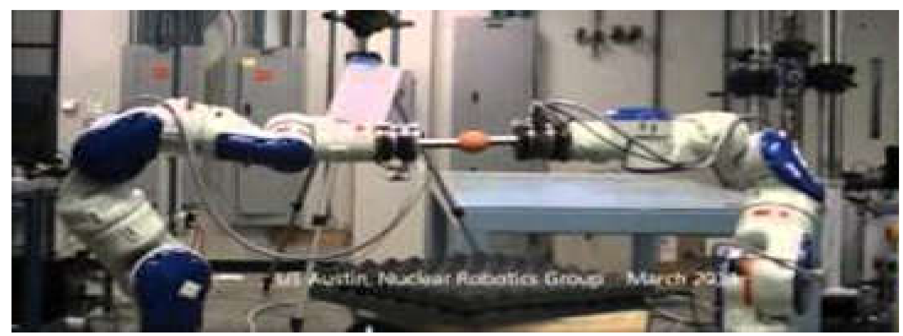
SISO, MISO, MIMO Systems…

Single Input Single Output (SISO)
gas → speed



Multi-Input Single Output (MISO)
Weight on Bit (WOB), RPM → ROP (Rate of Penetration)

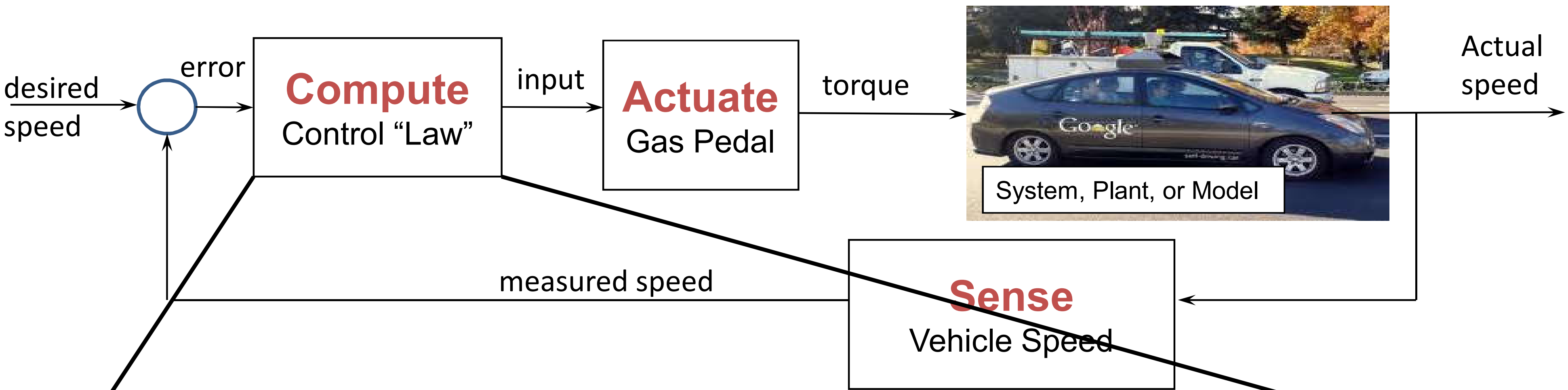
Multi-Input Multi-Output (MIMO)
7 joint currents → wrench (3 translation forces and 3 moments)



Our Objective

- To use feedback from the system or additional sensors to modify the inputs and produce a desired output. **To control the system**.

Our Controller’s Objective

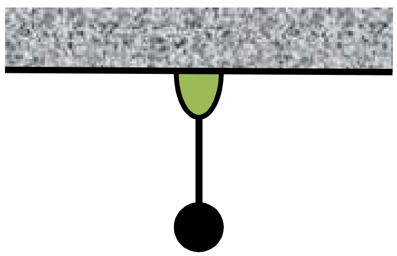


Goals

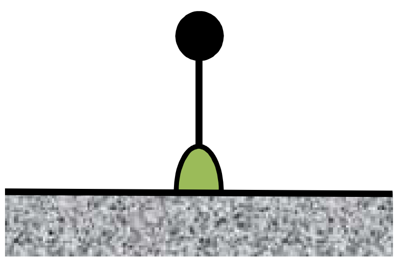
- ❑Stability: system maintains desired operating point (hold steady speed, tracks a particular trajectory)
- ❑Robustness: system tolerates perturbations in dynamics (mass, drag, etc)
- ❑Performance: system responds rapidly to changes (accelerate to 6 m/sec)

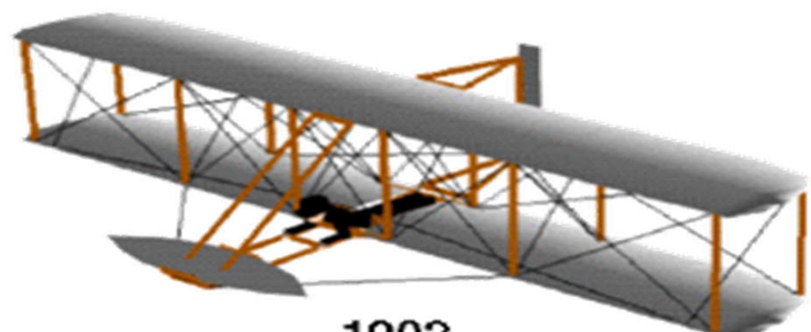
Stable vs unstable systems

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



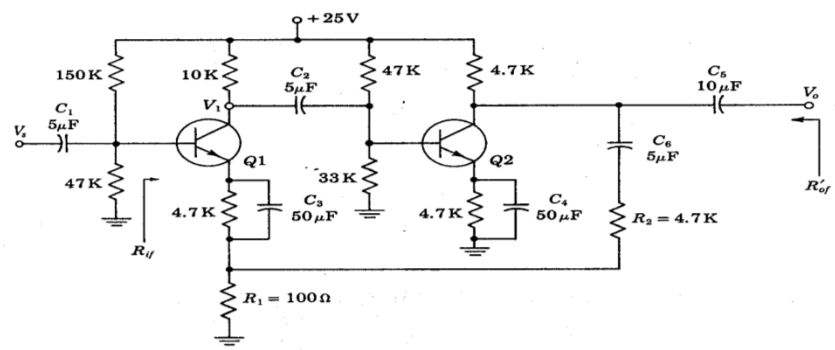
In unstable systems, small perturbations can disrupt the status quo





Robustness & Performance

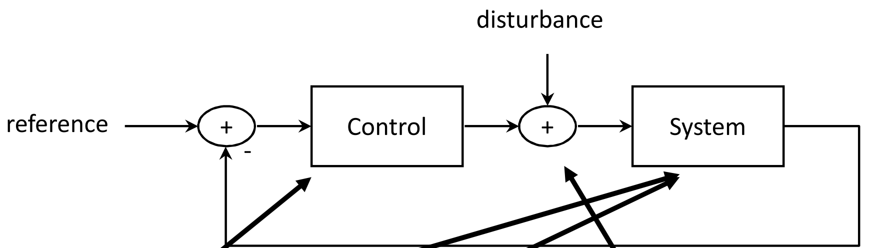
- **Robustness** to Uncertainty through Feedback
  - Feedback allows high performance in the presence of uncertainty
  - Example: **repeatable performance** of amplifiers with ±15% variation in component properties.
  - accurate *sensing* to compare actual to desired
  - correction through *computation* and *actuation*



- Better **Performance** through Feedback
  - Feedback allows the dynamics (behavior) of a system to be modified
  - Example: stability augmentation for highly agile, unstable aircraft
  - ***Find a closed loop controller that modifies the <u>natural</u> behavior***

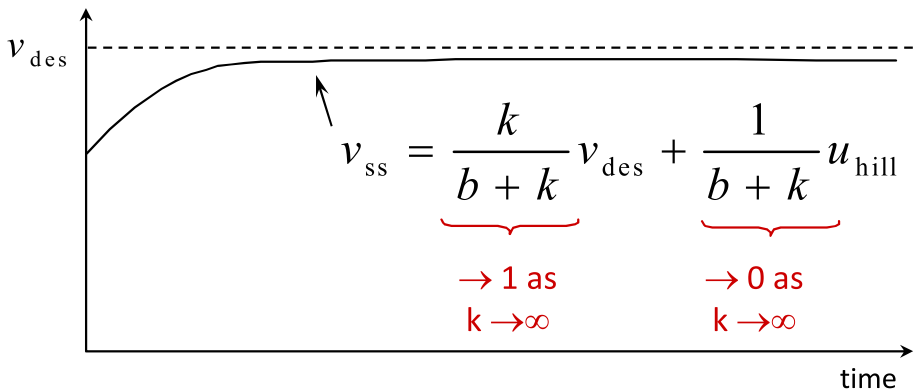
X-29 experimental aircraft

Simple Example: Speed Control



$$m\dot{v} = -bv + f_{\text{hill}} + f_{\text{engine}}$$

$$f_{\text{engine}} = k(v_{\text{desired}} - v)$$



$$v_{ss} = \underbrace{\frac{k}{b+k}}_{\to 1 \text{ as } k \to \infty} v_{des} + \underbrace{\frac{1}{b+k}}_{\to 0 \text{ as } k \to \infty} u_{\text{hill}}$$

- **Stability**
  - Stable.
- **Performance**
  - Steady state velocity *approaches* (but does not reach) desired velocity as $k \to \infty$.
  - Smooth response; no overshoot or oscillations
- **Robustness**
  - Effect of disturbances (eg, hills) approaches zero as $k \to \infty$
  - ***But k can only be so large!***

Another common control approach…

- **Bang-Bang Control**
- Something is turned on or turned off in response to sensor:
  - If the error signal is greater than e_1, turn system on
  - If error is signal is less than e_2, turn system off
  - e_1 and e_2 are *usually* not be the same value
  - The control action could be reversed
- Example: Sump pump
  - if water level is above 20 ft., turn pump on
  - if water level is below 5 ft., turn pump off
- Example: Air conditioning thermostat
  - If temp is > 85 degrees, turn on
  - If temp is < 83 degrees, turn off

Other common system characterizations

- **Number of inputs / Number of outputs**
  - Single Input/Single Output, Multi Input/Multi Output, etc. ((SISO, MISO, MIMO…)
  - We will generally focus on SISO systems.
- **Linear vs. Nonlinear Systems**
  - Can the system be modelled as a set of linear differential equations?
  - We will generally focus on linear systems.
  - *A common way to address nonlinear systems is to linearize them anyway….*
- **Time Invariant Systems**
  - Systems where the model is not a function of time
- **Continuous vs. Discrete Time Systems**
  - Analog vs Digital Systems
  - We will primarily focus on continuous systems
- **The system type most considered in this course is a SISO LTI Systems**
  - Single Input / Single Output, Linear, Continuous , Time Invariant Systems

So, what do we need to know?

- **Modeling**
  - Input/output representations for subsystems + interconnection rules
  - System identification theory and algorithms
  - Converting models to one or more ODEs
- **Analysis**
  - Methods to solve systems represented as ODEs.
  - Determine stability of feedback systems
  - Quantify the performance of input/output systems (response time, error margins, etc.)
  - Evaluate the robustness (regions of stability, disturbance rejection, etc.)
- **Synthesis**
  - Design of feedback systems
  - Testing of feedback systems
  - Visualizing feedback systems
  - Tools for signal processing and estimation (Kalman filters)

So let's get started!!!!

Course Outline

- **Today - What is feedback control?**
- **Part I – Modelling/Simulation**
  - Modeling with Differential and Difference Equations
  - Solving systems modelled as a set of ODEs
  - Analyzing Linear Systems, Convolution, and Canonical 2^{nd} order Systems
- **Part II – State/Output Feedback in time domain**
  - Stability in time domain
  - State feedback designs
  - Output feedback designs
- **Part III – Control system analysis in the frequency domain**
  - Transfer functions and block diagrams
  - Visual representation: Bode plots, root locus, and polar plots
  - Frequency response
  - Design of Proportional, Integral and Differential (PID) and other controllers.

The Deets, etc… (to the syllabus!)

MATLAB

- MATLAB (MATrix LABoratory) is a “numerical computing environment and programming language developed by Mathworks.”
- MATLAB has
  - Built-in matrix manipulation functions
  - Efficient plotting capabilities of vectors and matrices
  - Multiple tool-boxes and built in capabilities supporting the control algorithms discussed in this class.
  - Writing and storing user developed scripts
  - Interface with a variety of other programming languages (C++, Python, etc.) and programming environments (i.e. ROS, embedded systems, etc.)

MATLAB

- MATLAB (MATrix LABoratory) is a “numerical computing environment and programming language developed by Mathworks.”
- MATLAB is
  - a power graphing calculator
  - an interpreted language (commands executed line by line)
  - Highly customizable (Files → Preferences)
  - easy to find help for off and online (>> help help)
  - Available for all UT students

Final advice…

- The concept of a controller is easy to understand…
  – But…..
- You will not learn controls without creating controllers
  – (i.e. the homework, practice)

Learning Method	Retention By Learner
What They Read	10%
What They Hear	26%
What They See	30%
What They See and Hear	50%
What They Say	70%
What They Say As They Do Something	90%

(Styce, 1987)

- Homework is the key to success in this course.
  - Procrastination, programming, and comprehension are not good bedfellows.
- So……
  - Any questions?