MEAM 520
Lecture 23: Actuation and Control

Cynthia Sung, Ph.D.

Mechanical Engineering & Applied Mechanics

University of Pennsylvania

Previously: Manipulator Equation

We can write this as a matrix equation

$$\tau = \underbrace{D(q)}\ddot{q} + C(q, \dot{q})\dot{q} + g(q)$$

SHV uses a bit of strange notation.
Most people call this matrix *H* or *M*.

where

$D(q)$ is the *nxn* mass matrix (inertia terms)

$C(q, \dot{q})$ is the *nxn* matrix of centrifugal (square of joint velocities) and Coriolis (product of two different joint velocities) terms

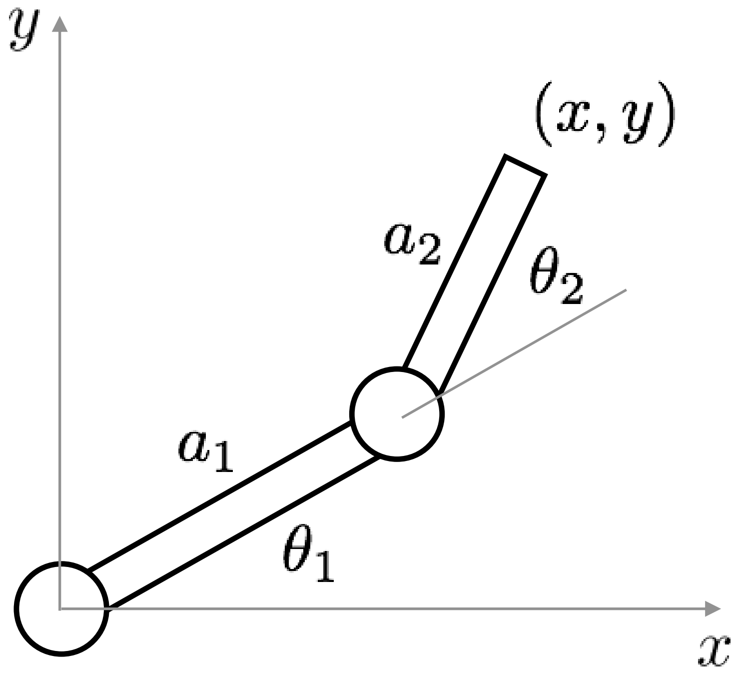
$g(q)$ is a *nx1* vector of gravitational terms

Previously: Euler-Lagrange Method

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q)$$

$$D = \sum_{i=1}^{N} \left(m_i J_{vci}^{\top} J_{vci} + J_{\omega i}^{\top} R_i I_i R_i^{\top} J_{\omega i} \right)$$

$$g = \frac{\partial}{\partial q} \sum_{i=1}^{N} m_i \vec{g} \cdot \vec{r}_i$$



$$(C\dot{q})_k = \sum_{i,j} \frac{1}{2} \left(\frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k} \right) \dot{q}_i \dot{q}_j$$

or

$$c_{kj} = \sum_{i} \frac{1}{2} \left(\frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k} \right) \dot{q}_i$$

Previously: Newton-Euler (for revolute joints)

Start with $\omega_0 = 0, \alpha_0 = 0, a_{c,0} = 0, a_{e,0} = 0$

Solve kinematic constraints for i from 1 to n

Start with $f_{n+1} = 0, \tau_{n+1} = 0$

Solve force/moments for i from n to 1

Kinematic constraints

i terms on the left

No forces/moments!

$$\omega_i = R_{i-1}^i \omega_{i-1} + z_{i-1}^i \dot{q}_i$$

$$\alpha_i = R_{i-1}^i \alpha_{i-1} + z_{i-1}^i \ddot{q}_i + \omega_i \times z_{i-1}^i \dot{q}_i$$

$$a_{e,i} = R_{i-1}^i a_{e,i-1} + \dot{\omega}_i \times r_{i,i+1} + \omega_i \times (\omega_i \times r_{i,i+1})$$

$$a_{c,i} = R_{i-1}^i a_{e,i-1} + \dot{\omega}_i \times r_{i,c_i} + \omega_i \times (\omega_i \times r_{i,c_i})$$

i-1 terms on the right

Forces/Moments

$$f_i - R_{i+1}^i f_{i+1} + m_i g_i = m_i a_{c,i}$$

$$\tau_i - R_{i+1}^i \tau_{i+1} + f_i \times r_{i,c_i} - (R_{i+1}^i f_{i+1}) \times r_{i+1,c_i} = I_i \dot{\omega}_i + \omega_i \times (I_i \omega_i)$$

MATLAB code is posted online

Previously: Method Comparisons

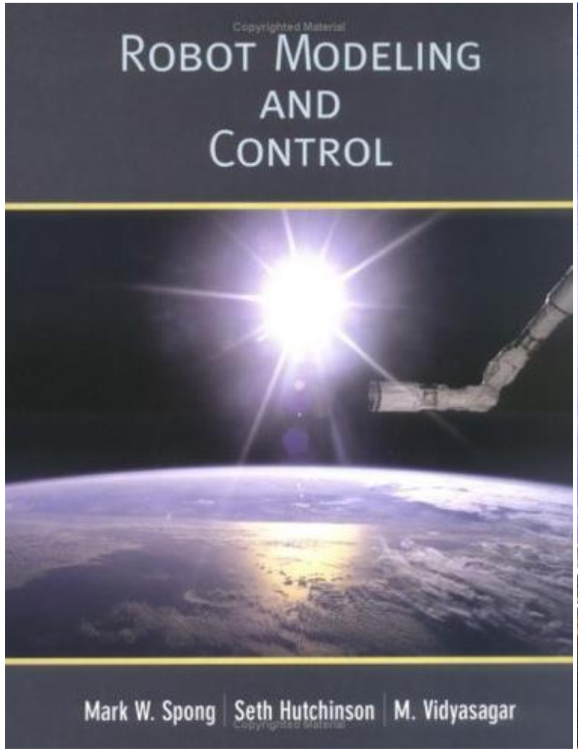
Newton-Euler

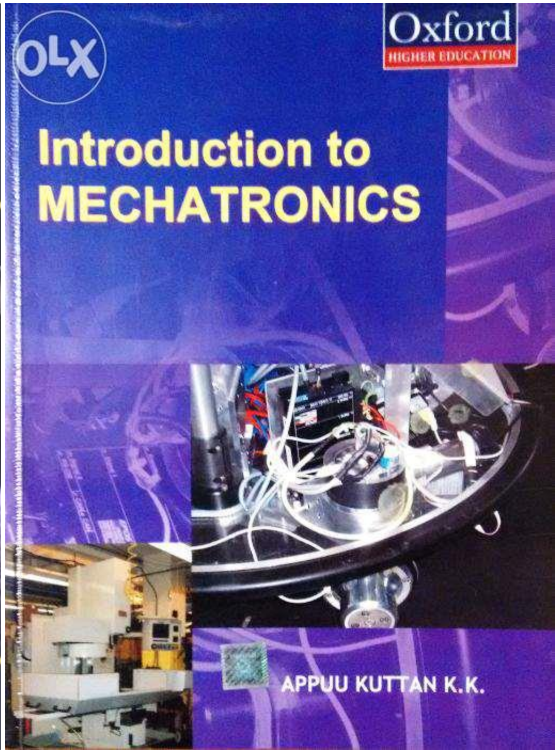
- Complete solution for all forces and kinematic variables
- Inefficient when only a few of the system’s forces need to be solved for

Euler-Lagrange

- Disregard all interactive and constraint forces that do not perform work
- Need to differentiate scalar energy functions
- Inefficient for large multi-body systems

How do we make a robot move to a particular location?





Chapter 6: Independent Joint Control

- Read 6.intro-6.4

AKKK: 7.2 – 7.3

More details in

ESE 500 Linear Systems

MEAM 513 Control Systems

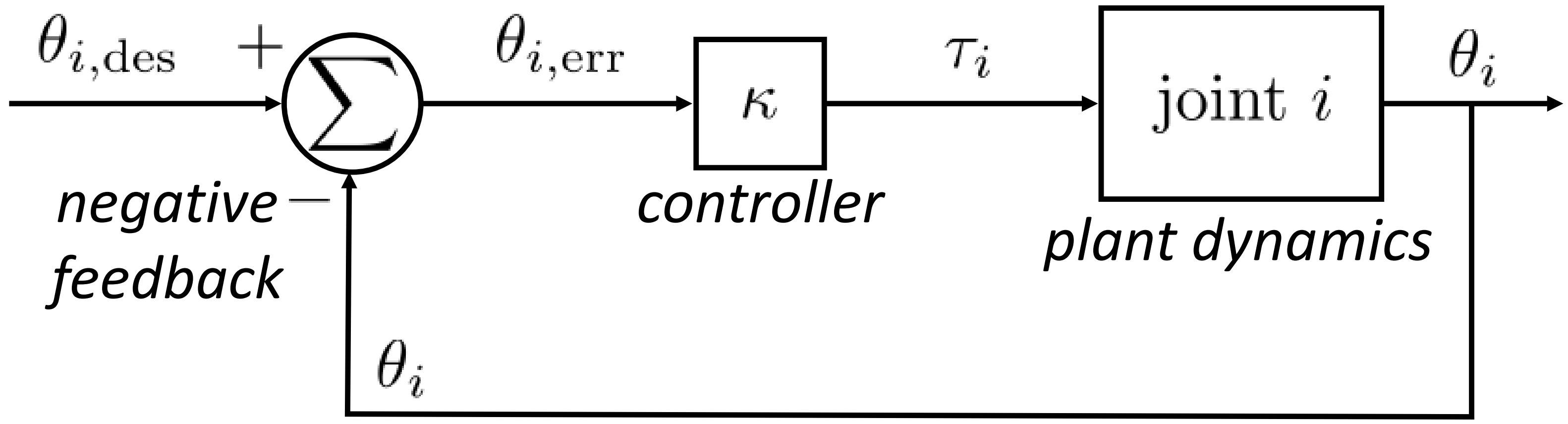
Desired Joint Angles

$\theta_{1,\mathrm{des}}, \theta_{2,\mathrm{des}}, \theta_{3,\mathrm{des}}$ ...

Actual Joint Angles

$\theta_1, \theta_2, \theta_3$ ...

Proportional Feedback Controller



Desired Joint Angles

$\theta_{1,\mathrm{des}}, \theta_{2,\mathrm{des}}, \theta_{3,\mathrm{des}}$ ...

Actual Joint Angles

$\theta_1, \theta_2, \theta_3$ ...

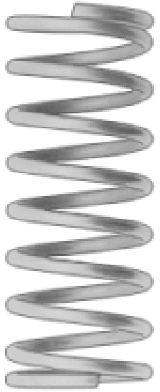
Proportional Feedback Controller

joint torques

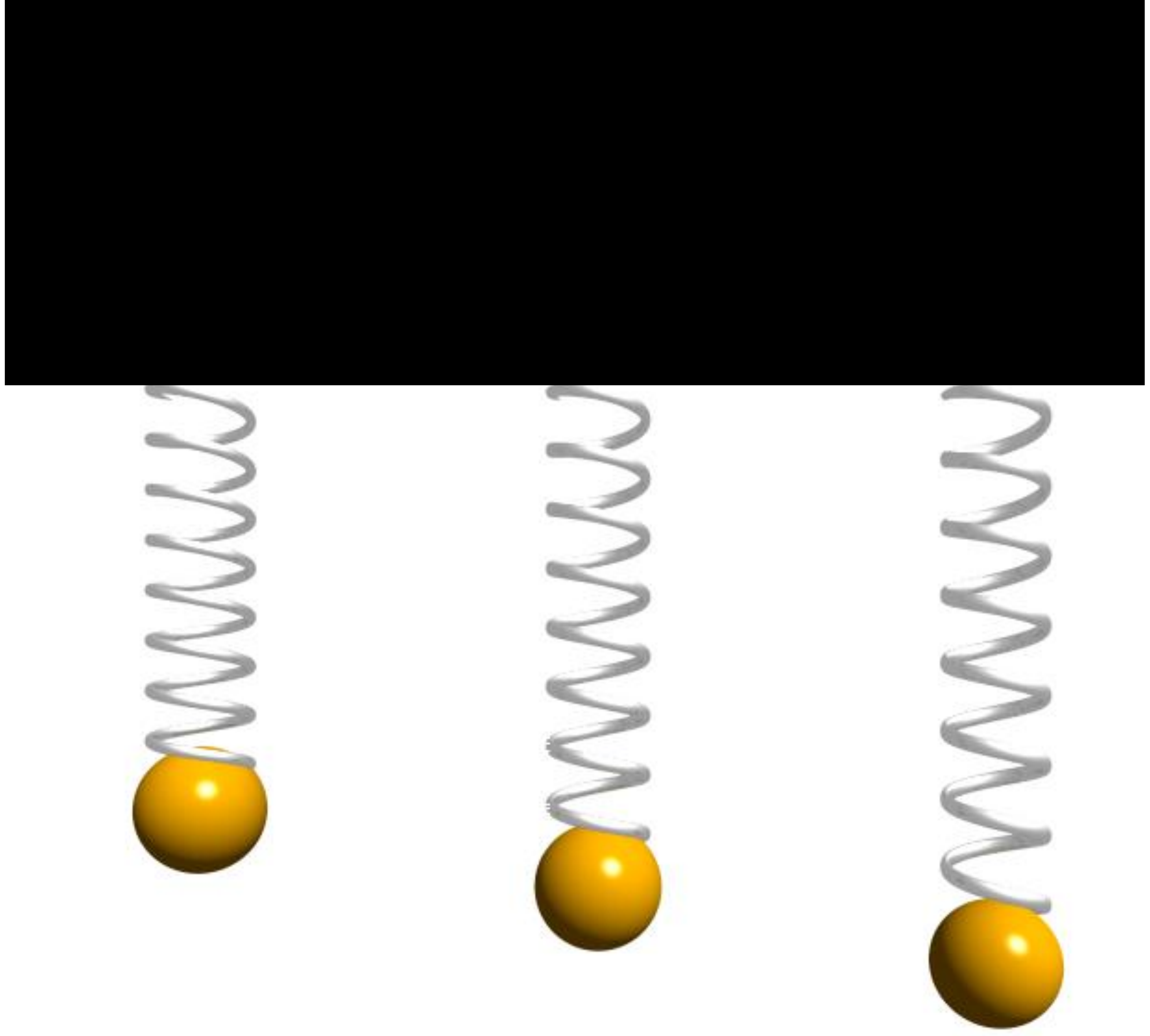
$$\tau_1 = \kappa(\theta_{1,\mathrm{des}} - \theta_1)$$
$$\tau_2 = \kappa(\theta_{2,\mathrm{des}} - \theta_2)$$
$$\tau_3 = \kappa(\theta_{3,\mathrm{des}} - \theta_3)$$

joint angle errors

proportional gain in Nm / rad



Proportional feedback acts like a torsional spring with linear stiffness, pulling each joint to the desired angle.



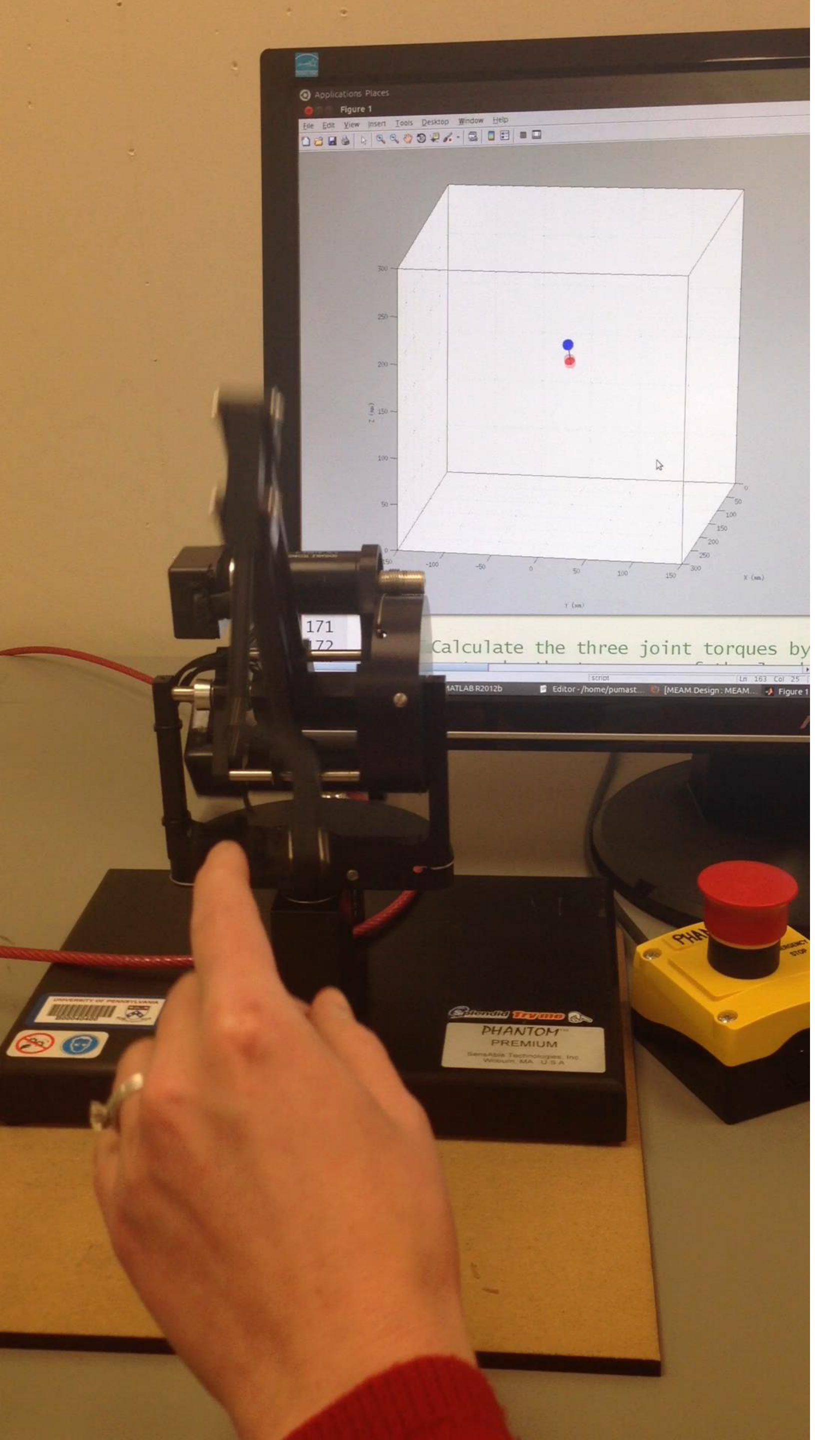
Mass on a spring: simple harmonic oscillator

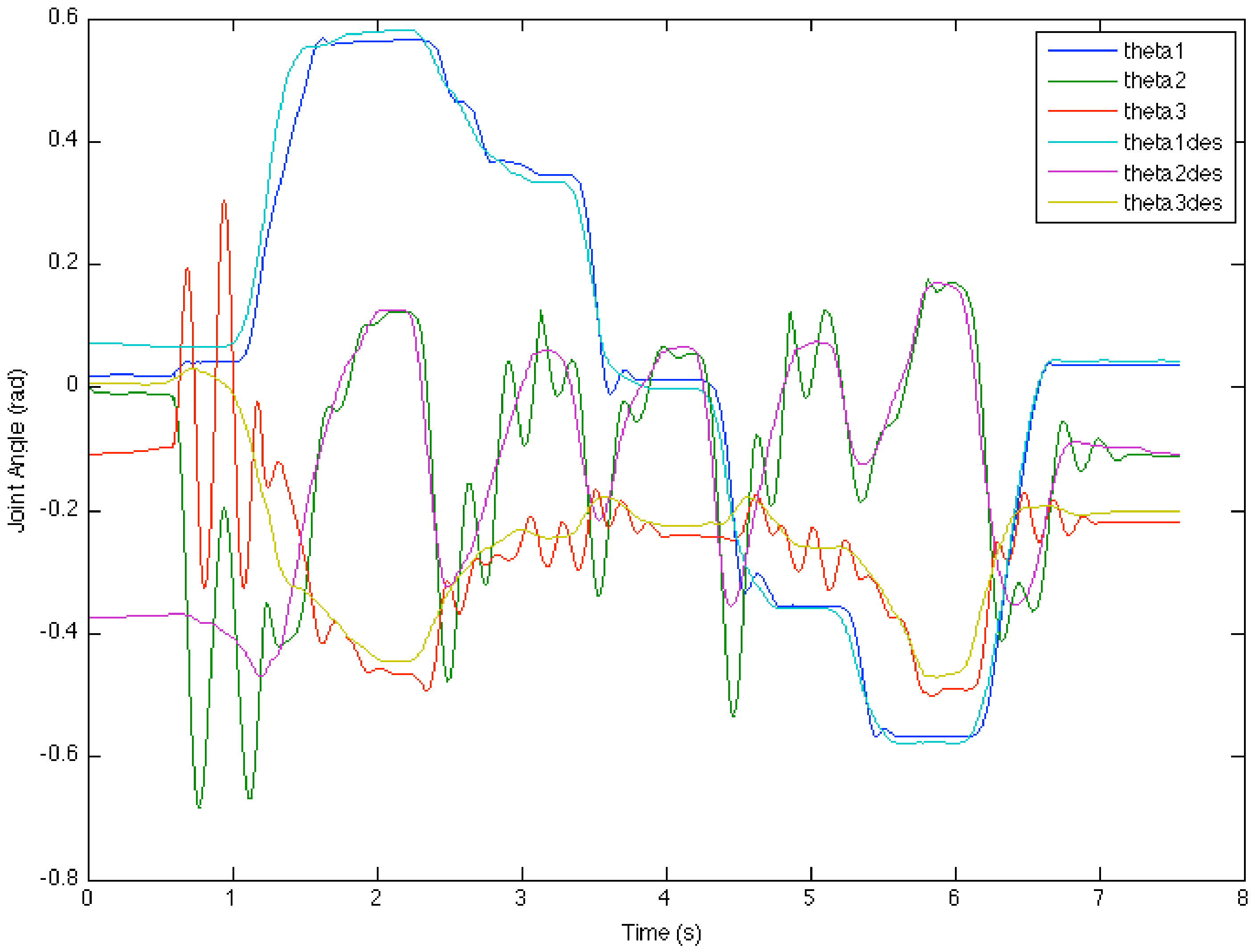
$$\tau_i = \kappa\left(\theta_{i,des} - \theta_i\right)$$

$$f_1(q)\ddot{q} + f_2(q,\dot{q}) = \kappa\left(\theta_{i,des} - \theta_i\right)$$

$$f_1(q)\ddot{q} = \left[\kappa\left(\theta_{i,des} - \theta_i\right) - f_2(q,\dot{q})\right]$$

Image from
http://lifshitz.ucdavis.edu/~dmartin/phy7/7C.html





It's pretty oscillatory.

How can we improve the controller’s tracking?

Add derivative feedback – virtual damping.

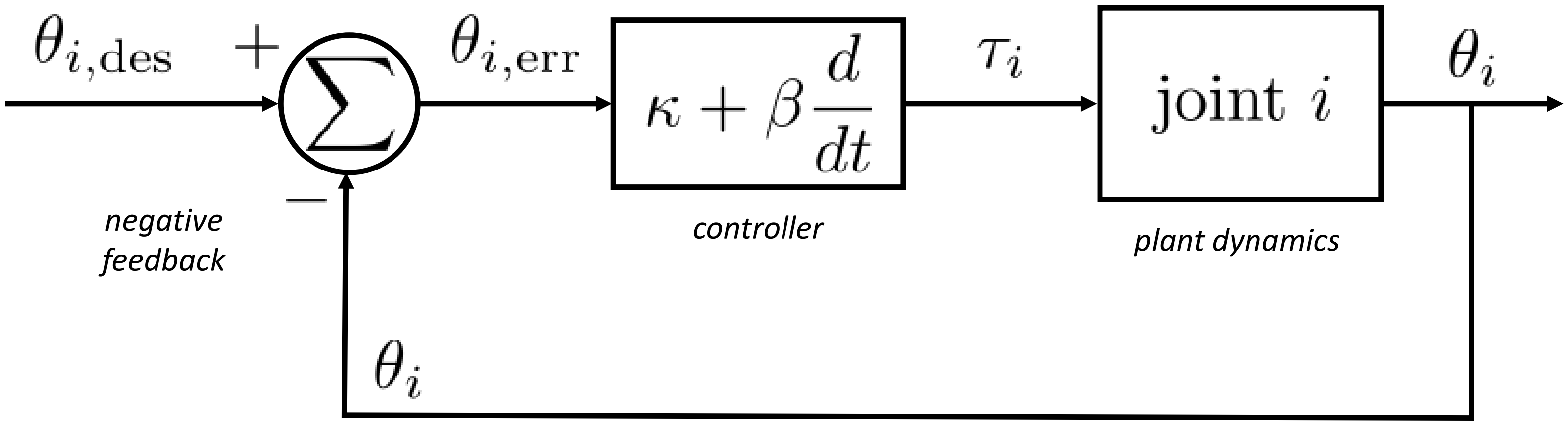
Desired Joint Angles

$\theta_{1,\mathrm{des}}, \theta_{2,\mathrm{des}}, \theta_{3,\mathrm{des}}$ ...

Actual Joint Angles

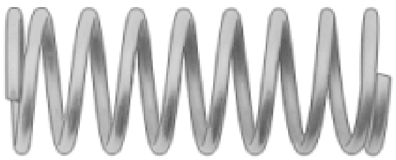
$\theta_1, \theta_2, \theta_3$ ...

Proportional Derivative Feedback Controller

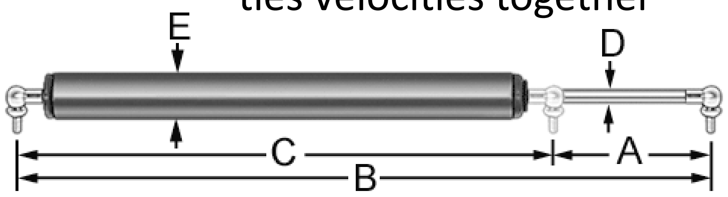


Add a derivative term to our position feedback controller, making it a Proportional Derivative (PD) controller.

virtual spring,
ties positions together



virtual damper,
ties velocities together



$$\tau_1 = \kappa(\theta_{1,\text{des}} - \theta_1) + \beta(\omega_{1,\text{des}} - \omega_1)$$

$$\tau_2 = \kappa(\theta_{2,\text{des}} - \theta_2) + \beta(\omega_{2,\text{des}} - \omega_2)$$

$$\tau_3 = \kappa(\theta_{3,\text{des}} - \theta_3) + \beta(\omega_{3,\text{des}} - \omega_3)$$

proportional gain in Nm / rad

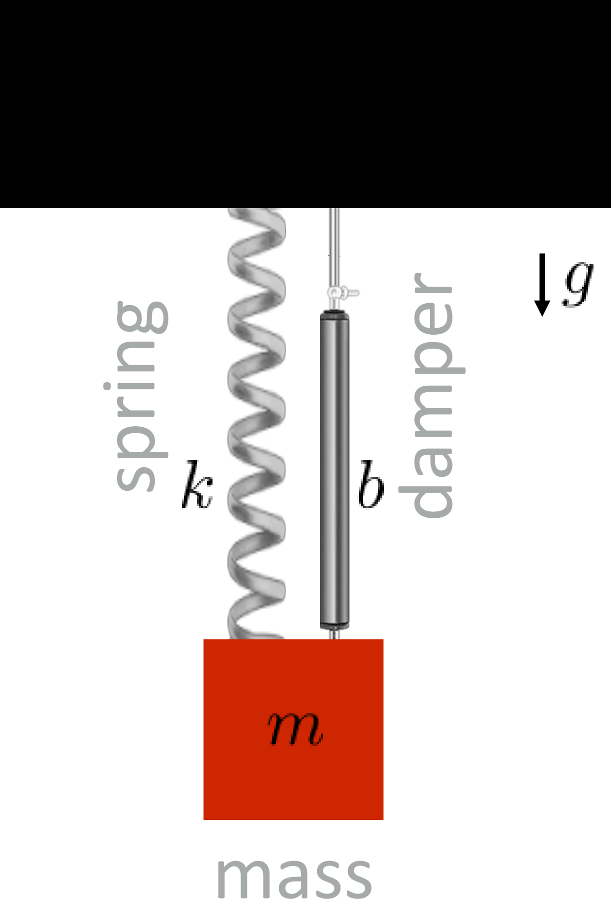
derivative gain in Nm / (rad/s)

The gains are typically tuned separately for each joint.

$$\theta_{i,\text{err}} = \theta_{i,\text{des}} - \theta_i$$

$$\dot{\theta}_{i,\text{err}} = \omega_{i,\text{des}} - \omega_i$$

$$\tau_i = \kappa\,\theta_{i,\text{err}} + \beta\,\dot{\theta}_{i,\text{err}}$$



$$\Sigma F_y = m\ddot{y}$$

$$-mg - ky - b\dot{y} = m\ddot{y}$$

$$-mg = m\ddot{y} + b\dot{y} + ky$$

$$-g = \ddot{y} + \frac{b}{m}\dot{y} + \frac{k}{m}y$$

Second-order system

$$\frac{k}{m} = \omega_n^2$$

natural frequency

$$-g = \ddot{y} + 2\zeta\omega_n\dot{y} + \omega_n^2 y$$

$$\frac{b}{m} = 2\zeta\omega_n$$

$$k_{\text{controller}} = m\omega_{n,\text{desired}}^2$$

$$b_{\text{controller}} = 2m\zeta_{\text{desired}}\omega_n - b_{\text{robot}}$$

usually, $\zeta_{\text{desired}} = 1$ *damping ratio*

What are the effects of the gains?

- The system goes unstable if either k_p or k_d are negative
- The system is critically damped if $\zeta = \frac{b}{2m\omega_n} = 1$
- For a fast response, k_p should be as high as possible, subject to saturation, chattering, etc.
- With a constant disturbance D (e.g., gravity), the constant offset with PD control is $-\frac{D}{k_p}$

https://www.youtube.com/watch?v=C3jDRbwDzsM

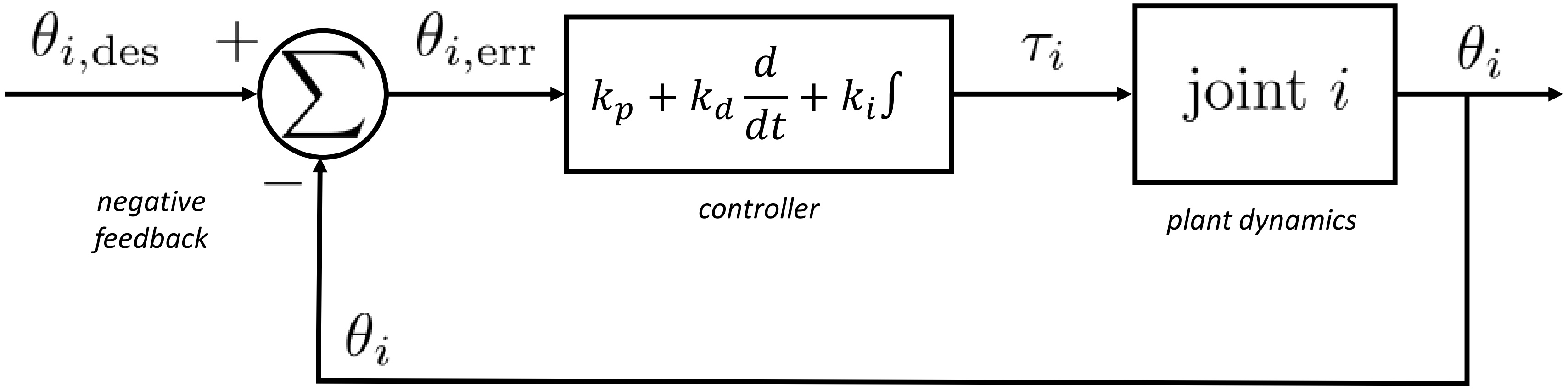
Desired Joint Angles

$\theta_{1,\mathrm{des}}, \theta_{2,\mathrm{des}}, \theta_{3,\mathrm{des}}$ ...

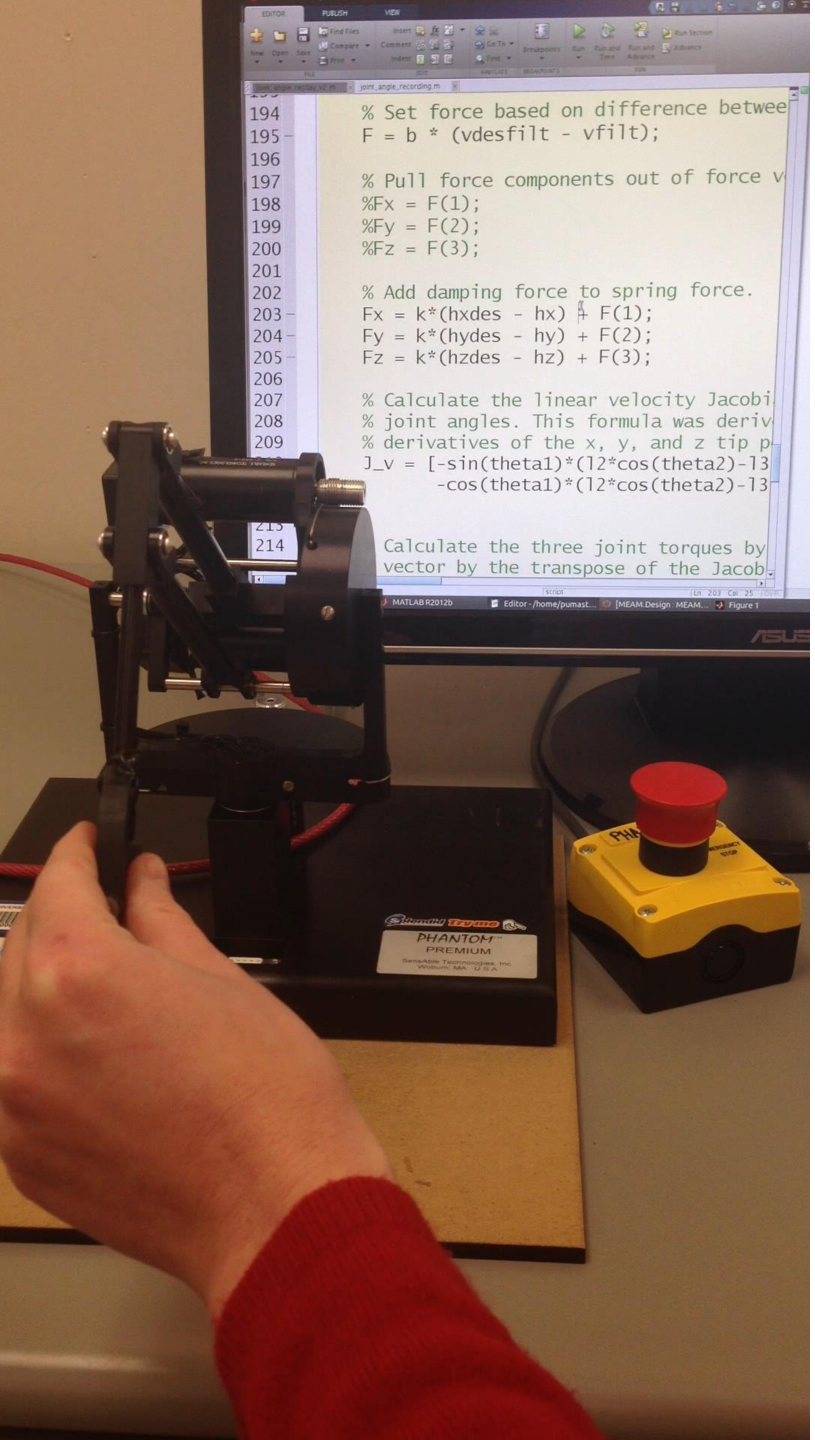
Actual Joint Angles

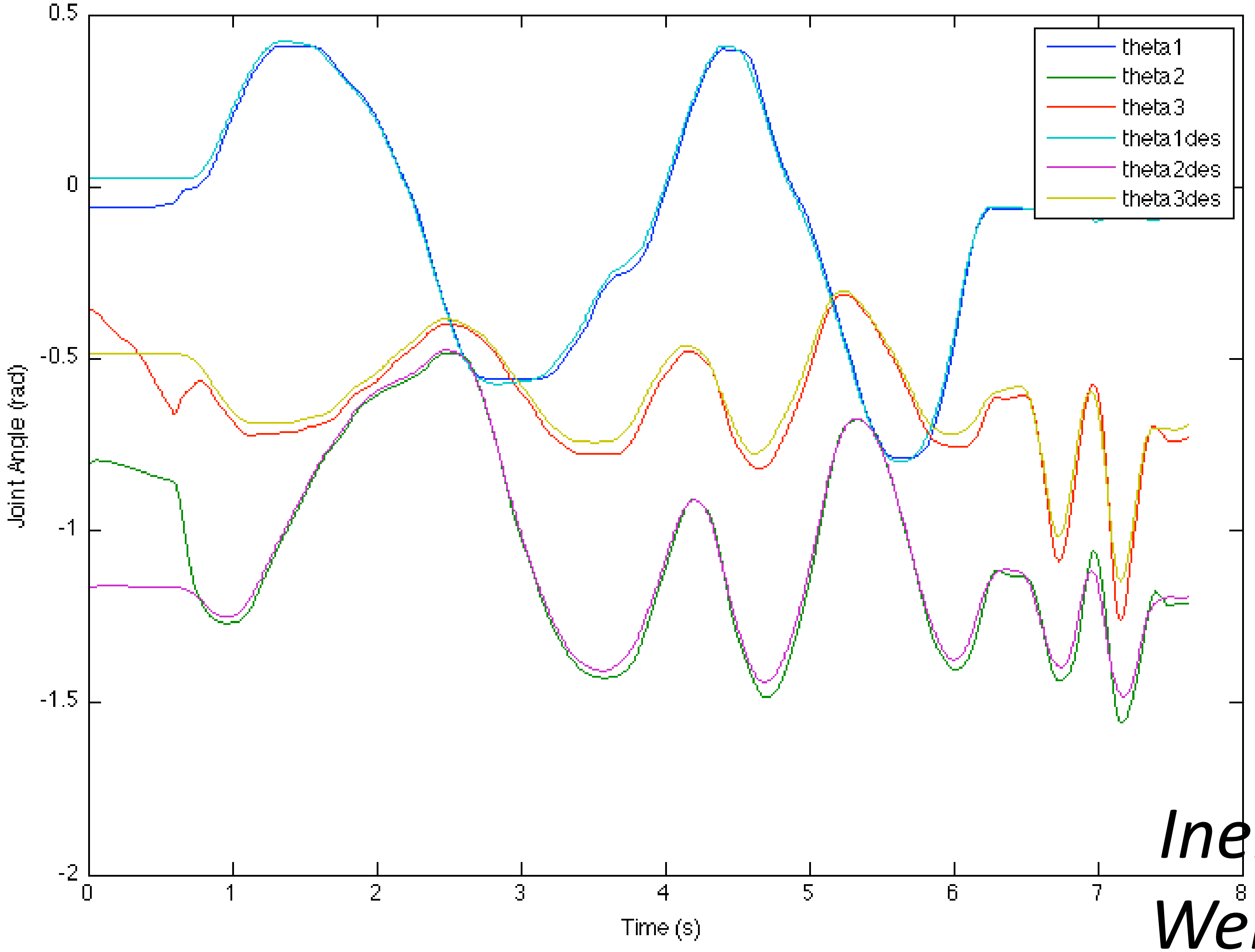
$\theta_1, \theta_2, \theta_3$ ...

Proportional Integral Derivative Feedback Controller



$$\tau_i = k_p(\theta_{i,des} - \theta_i) + k_d(\omega_{i,des} - \omega_i) + k_i \int \theta_{i,des} - \theta_i$$



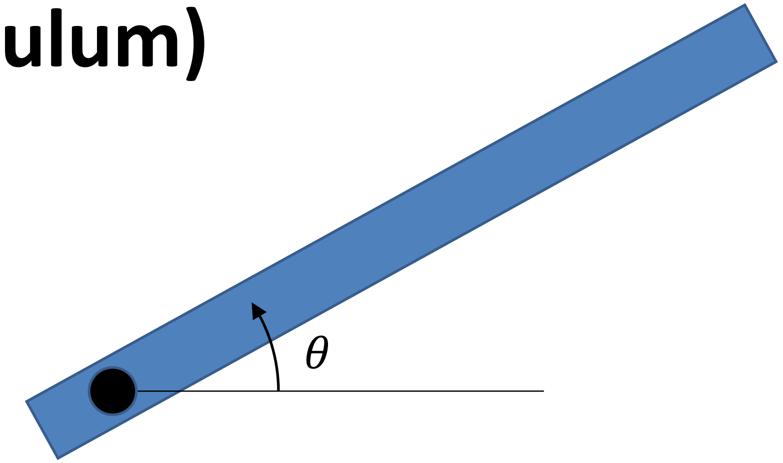


Inertia
Weight
Friction

Why isn’t it perfect?

The robot’s dynamics interfere with tracking.

Example: Single Link Manipulator (i.e., Pendulum)



EOM: $m\ddot{\theta} + b_m\dot{\theta} = \tau + mgl\cos\theta$

Controller: $\tau = k_p\theta_e + k_i\int \theta_e dt + k_d\dot{\theta}_e$

System dynamics:

$$m\ddot{\theta}_e + (b_m + k_d)\dot{\theta}_e + k_p\theta_e + k_i\int \theta_e dt = mgl\cos\theta$$

The inertia, weight, and friction of the robot all interfere with tracking.

Robot designers generally try to minimize the **inertia (mass & mass distribution)** of the robot so it can **accelerate** more quickly and be less affected by gravity.

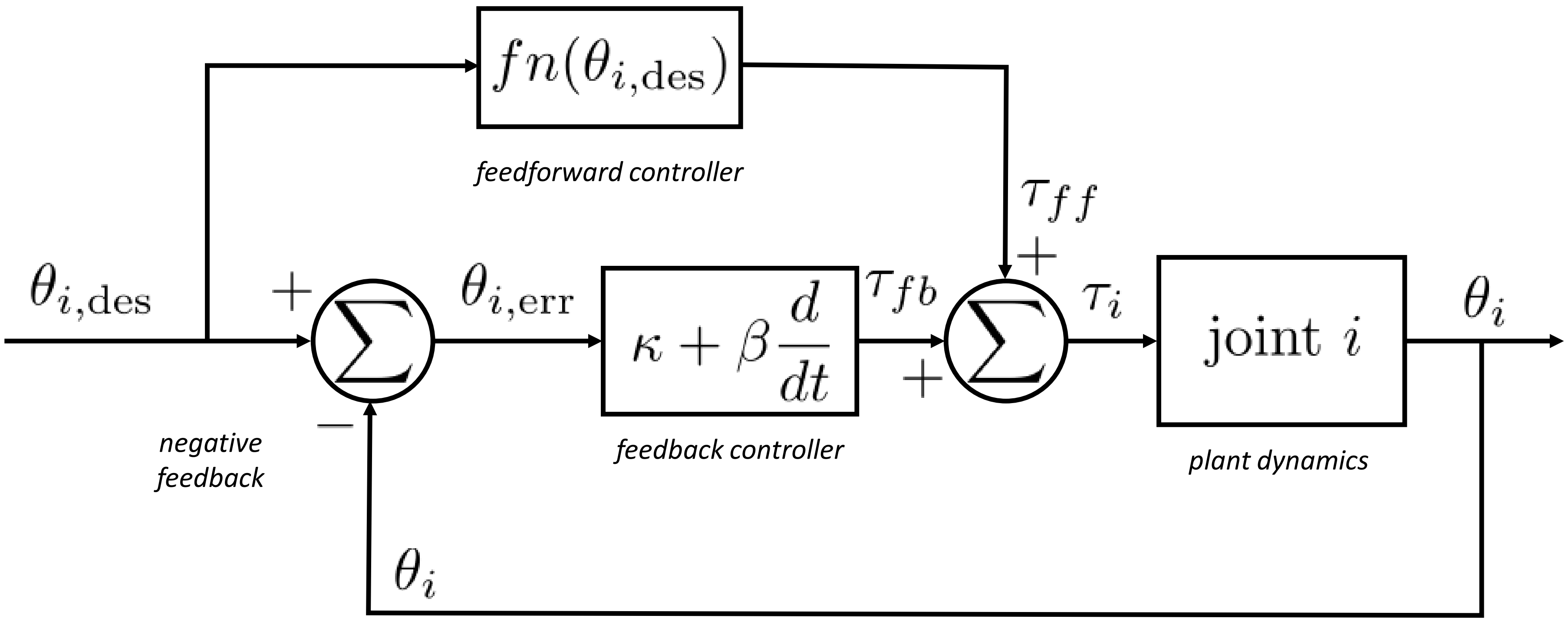
Similarly, robot designers try to minimize the **friction** of the robot so that the start of motion is **smooth** and sustained motion doesn’t require much torque.

How can we improve the controller’s tracking?

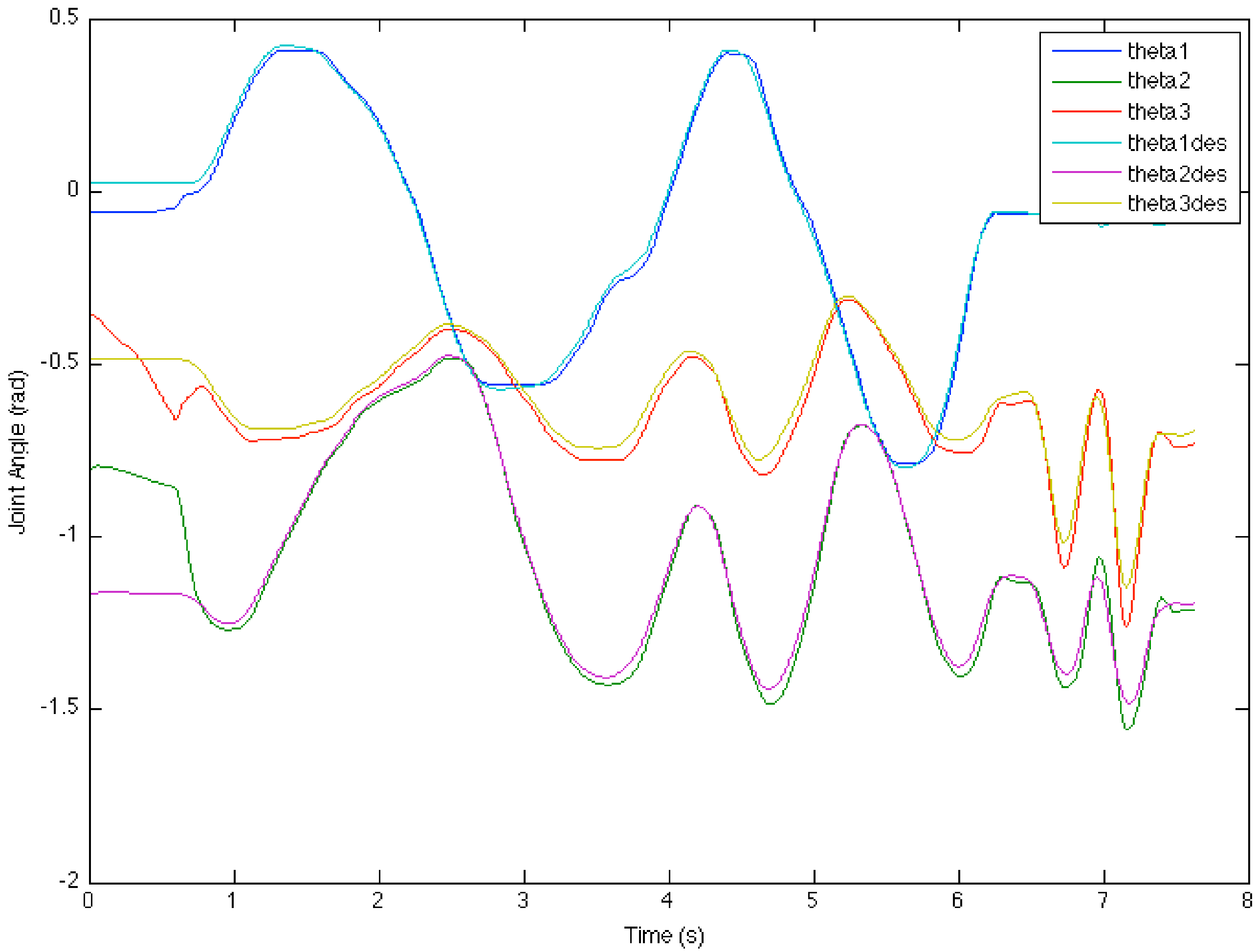
Try to compensate for the robot's dynamics in advance, instead of just reacting to errors when they occur.

This approach is called **feedforward control**, and it's very powerful for tracking time-varying trajectories.

Adding a Feedforward Term to the PD Controller



Which aspect of the robot dynamics can we feedforward?



Inertia

Weight

Friction

We could try any of them, but robot weight is generally the easiest and most useful.

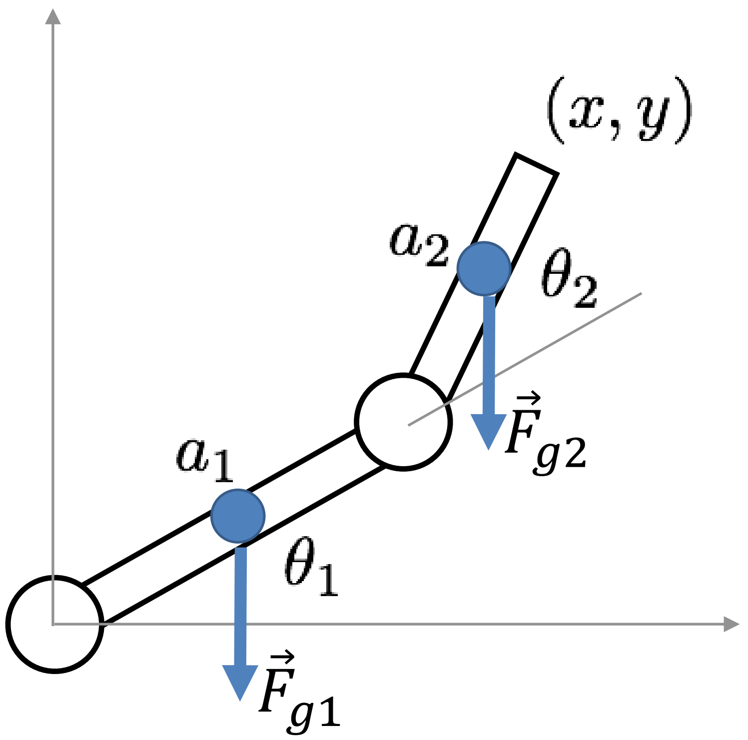
Previously: Gravitational Force/Torque

$$\vec{\tau}^\top d\vec{q} = \vec{F}^\top d\vec{x}$$

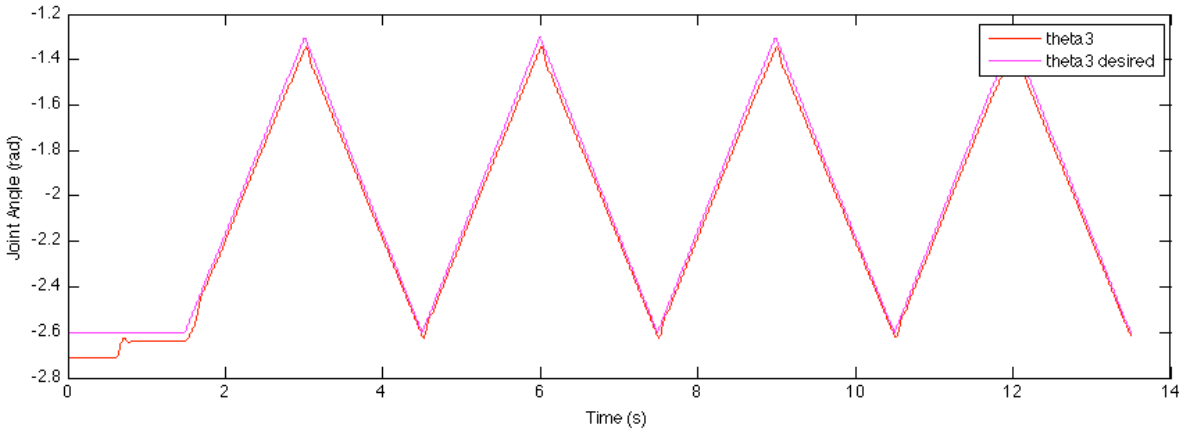
$$\vec{\tau}^\top d\vec{q} = \sum_{i=1}^{n} \vec{F}_{gi}^\top d\vec{x}_i$$

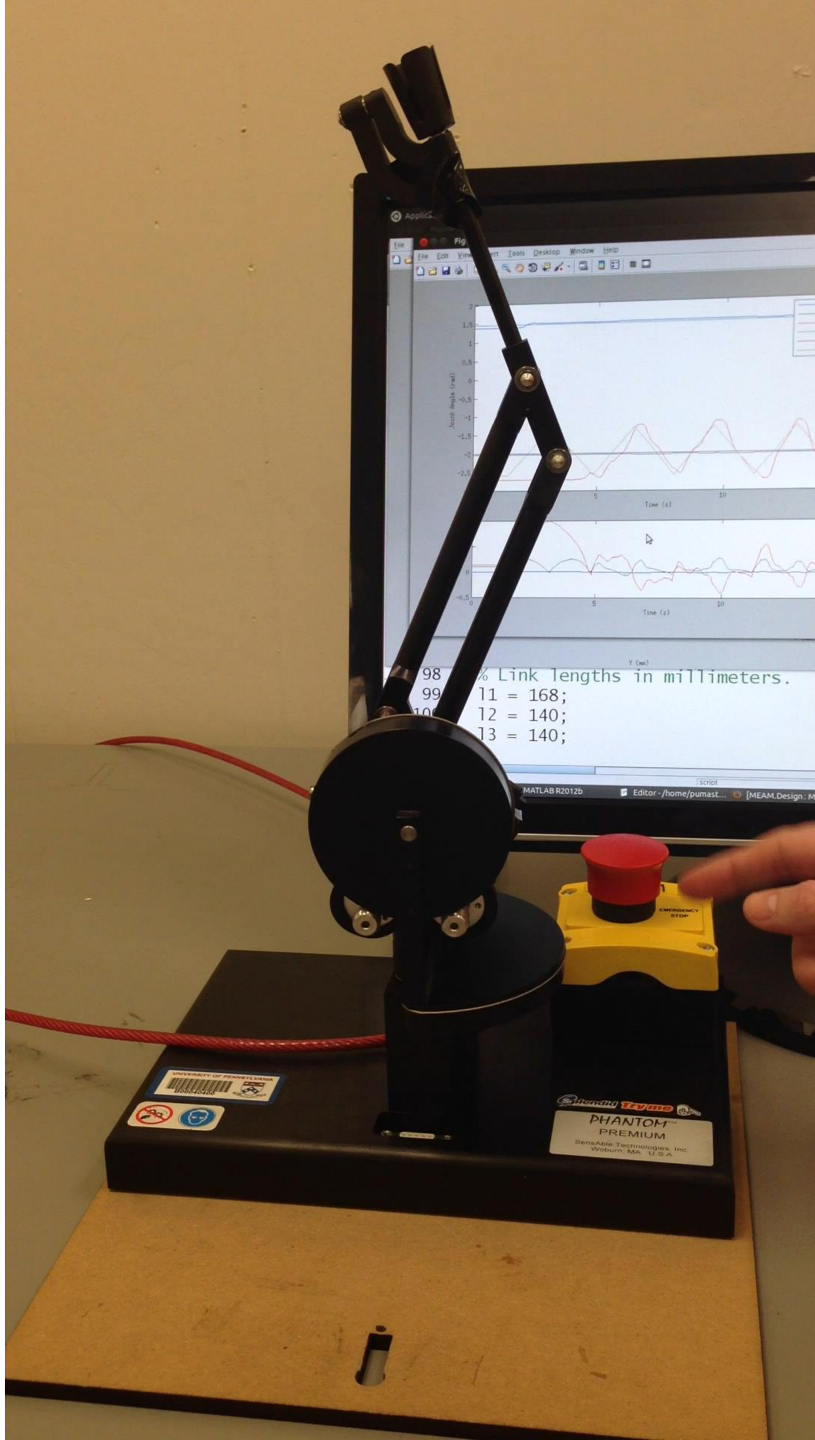
$$\vec{\tau}^\top d\vec{q} = \sum_{i=1}^{n} \vec{F}_{gi}^\top J_i d\vec{q}$$

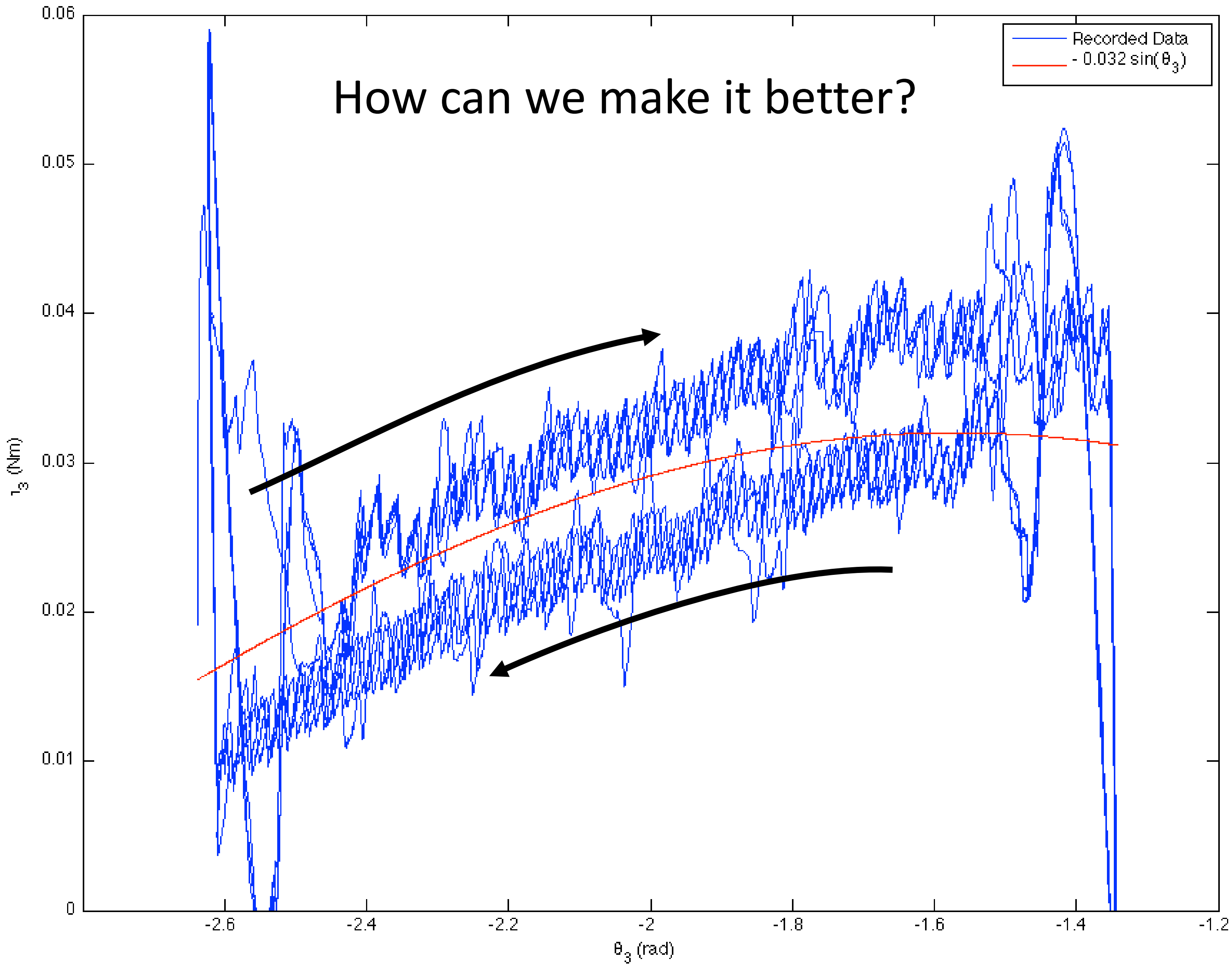
$$\vec{\tau} = \sum_{i=1}^{n} J_i^\top \vec{F}_{gi}$$



Another Option: Record torque on robot through a trajectory







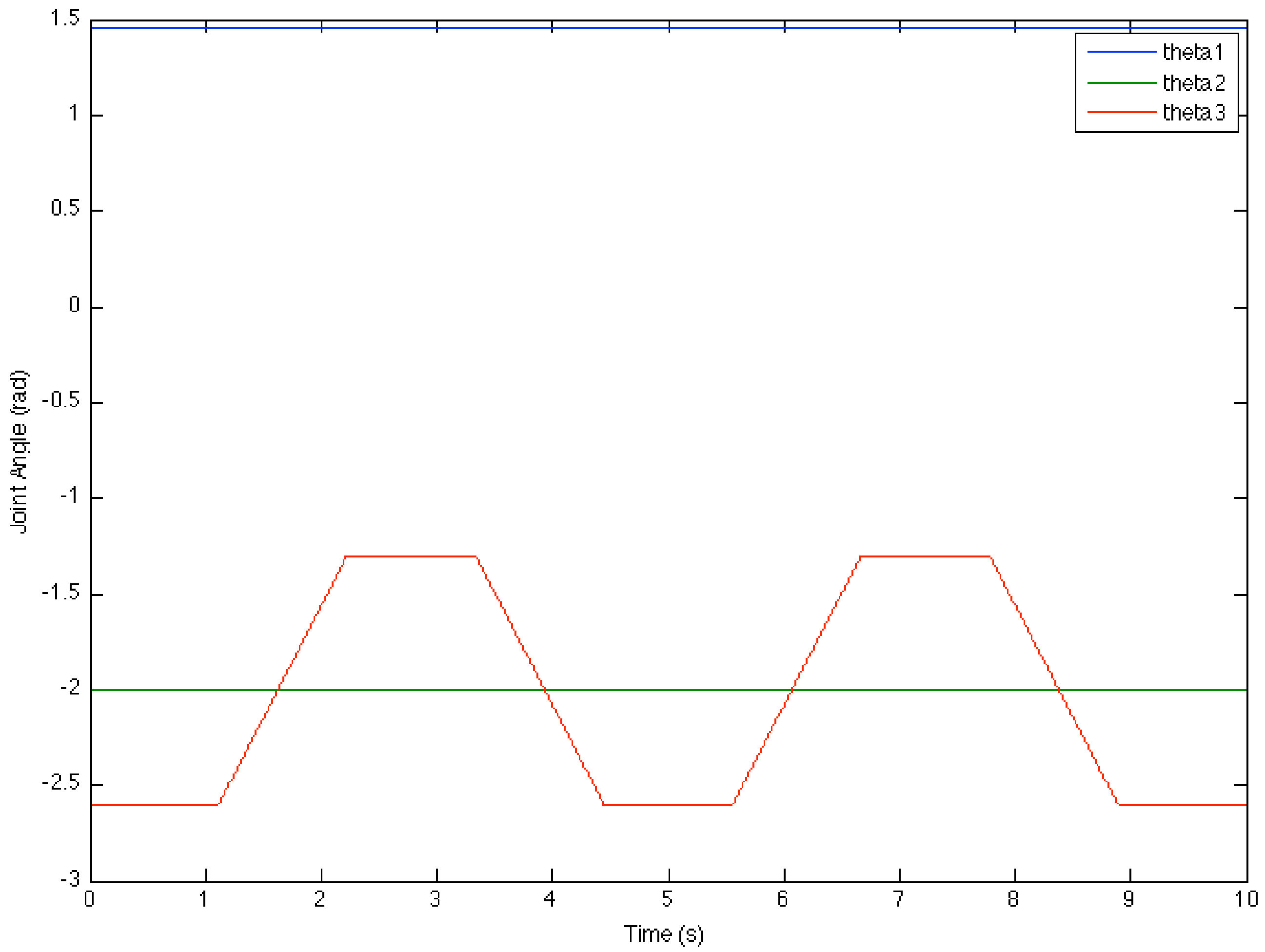
Inertia

Weight

Friction

How does each of these dynamic properties affect this test?

Flatten the sharp corners



Trajectory Generation

start

end

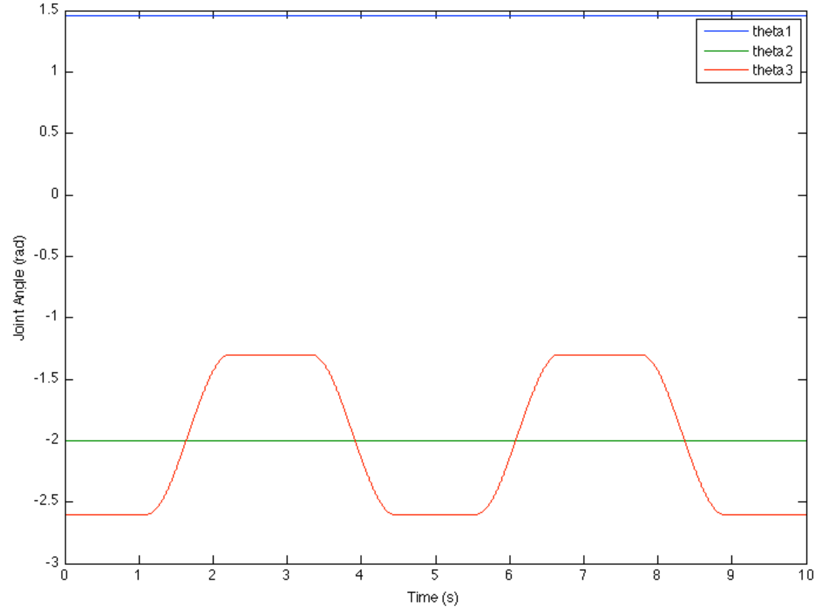
$$q(t_0) = q_0 \longrightarrow q(t_f) = q_f$$

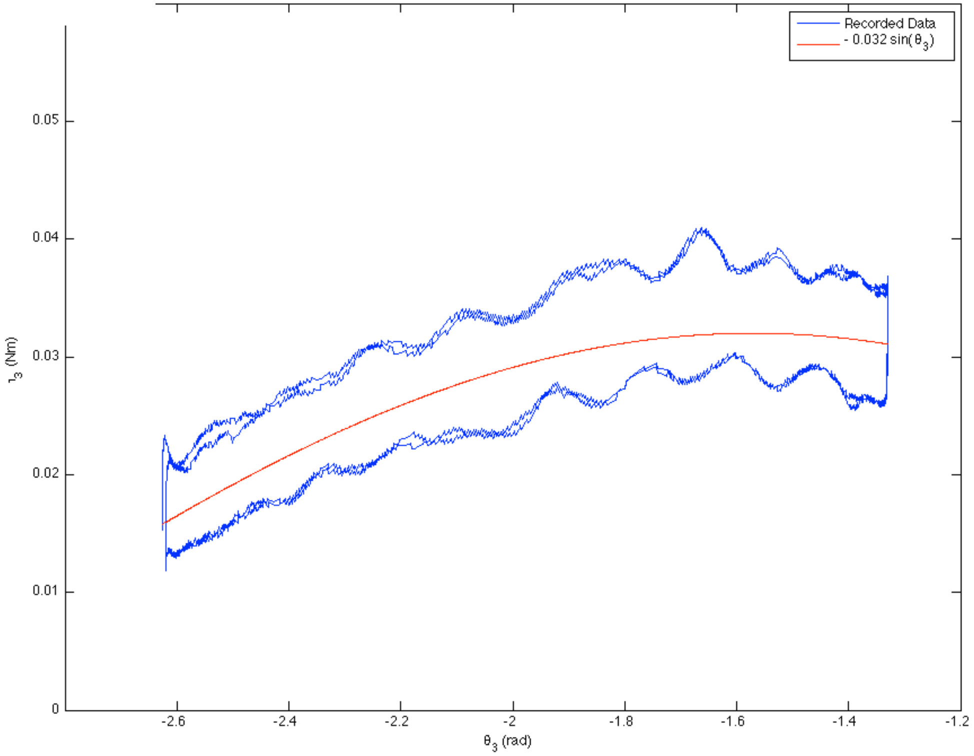
$$\dot{q}(t_0) = v_0 \longrightarrow \dot{q}(t_f) = v_f$$

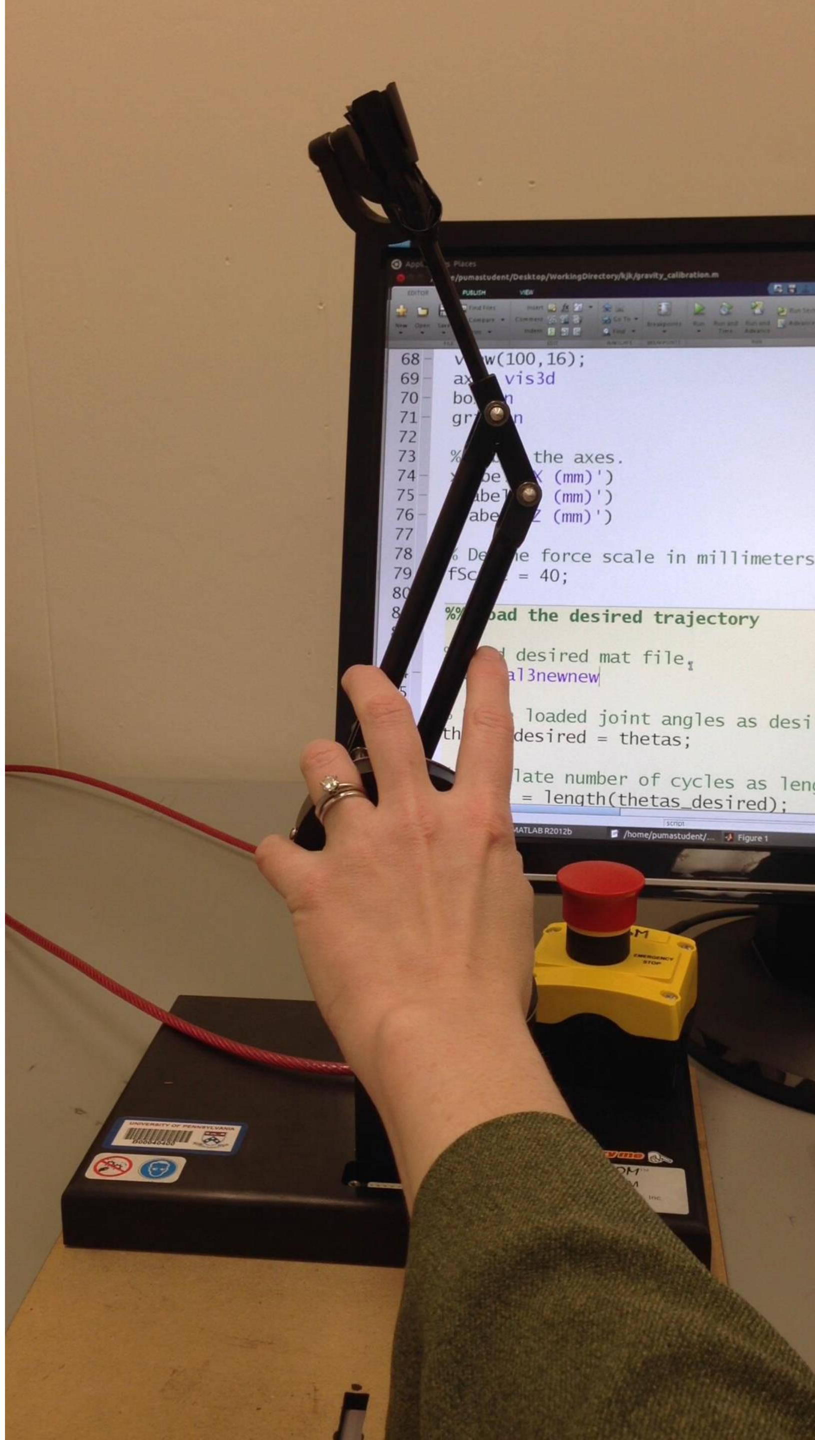
cubic polynomial

$$q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

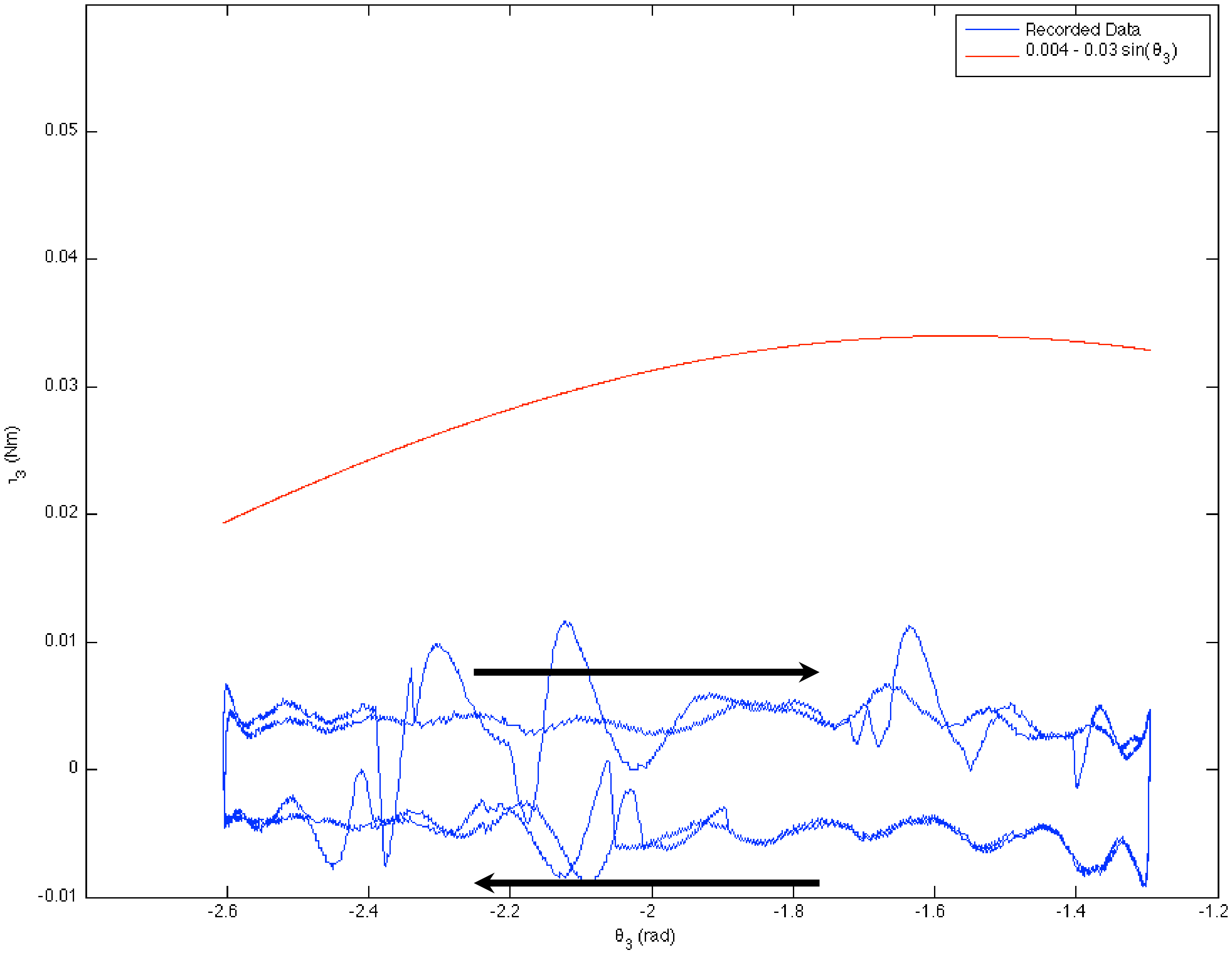
$$\begin{bmatrix} q_0 \\ v_0 \\ q_f \\ v_f \end{bmatrix} = \begin{bmatrix} 1 & t_0 & {t_0}^2 & {t_0}^3 \\ 0 & 1 & 2t_0 & 3{t_0}^2 \\ 1 & t_f & {t_f}^2 & {t_f}^3 \\ 0 & 1 & 2t_f & 3{t_f}^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

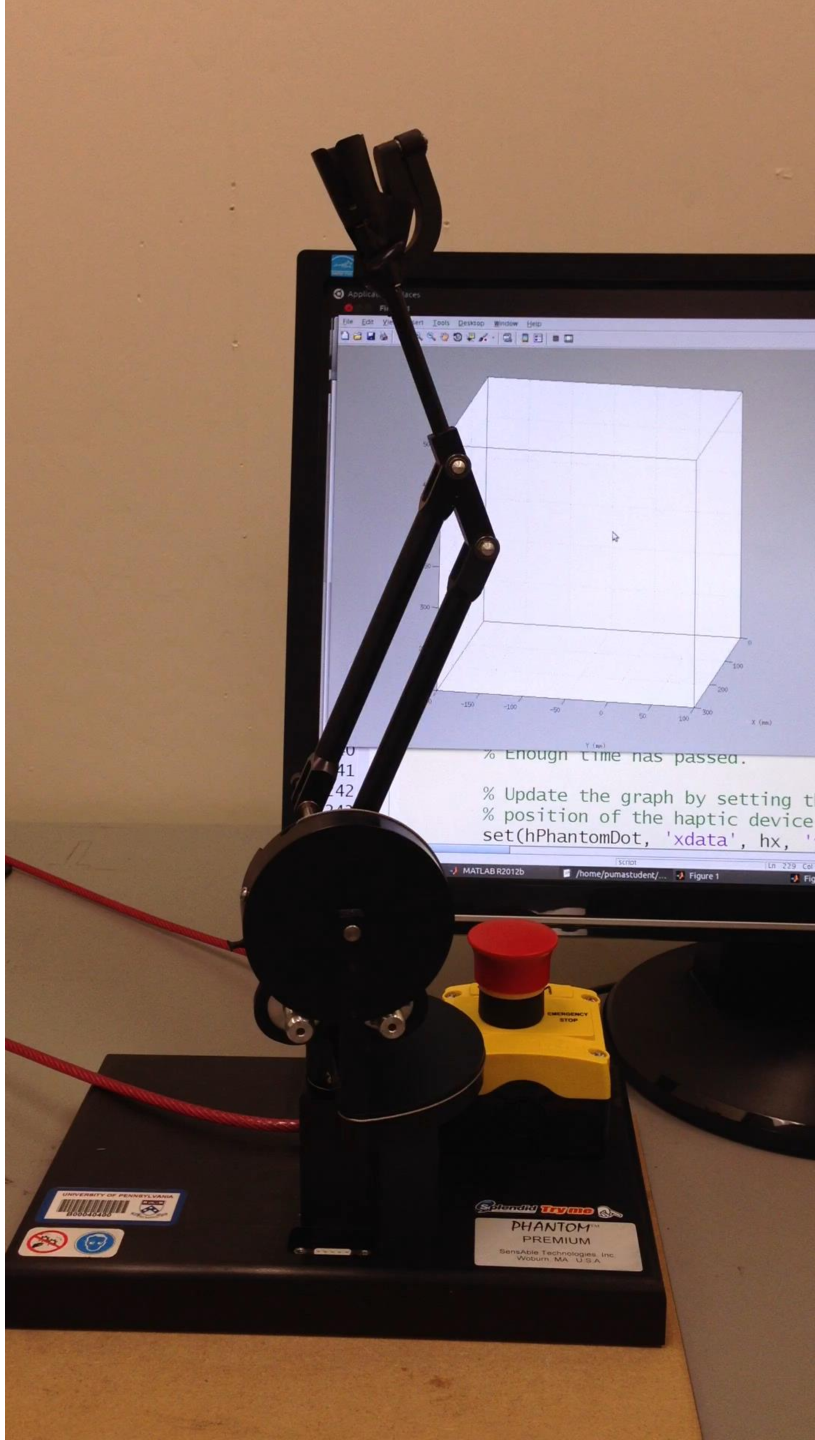

theta1
theta2
theta3
Joint Angle (rad)
Time (s)
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
-3
0
1
2
3
4
5
6
7
8
9
10


Recorded Data
$-0.032\sin(\theta_3)$
τ_3 (Nm)
θ_3 (rad)
0.05
0.04
0.03
0.02
0.01
0
-2.6
-2.4
-2.2
-2
-1.8
-1.6
-1.4
-1.2

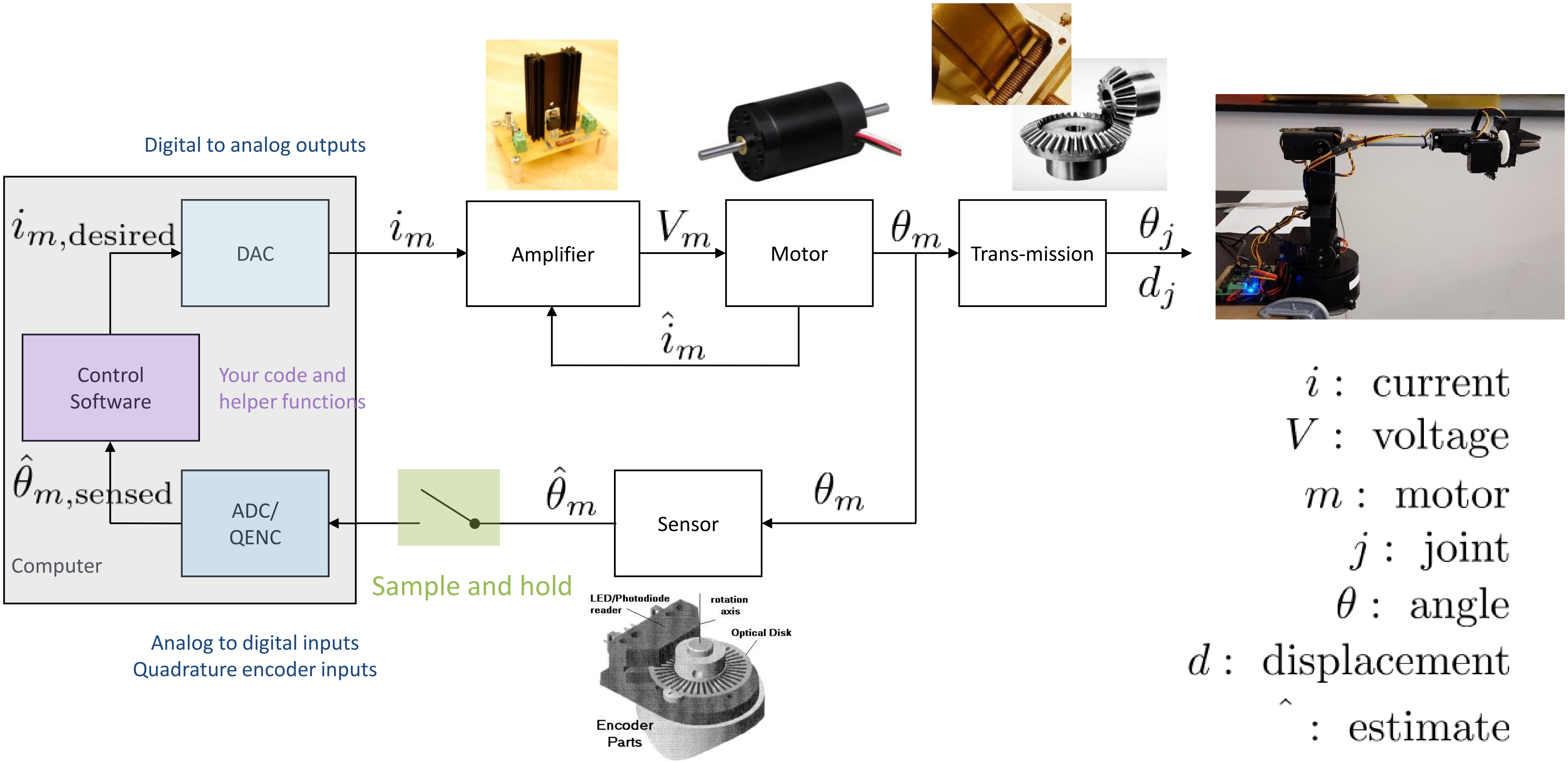


You can also use this strategy to compensate for friction.



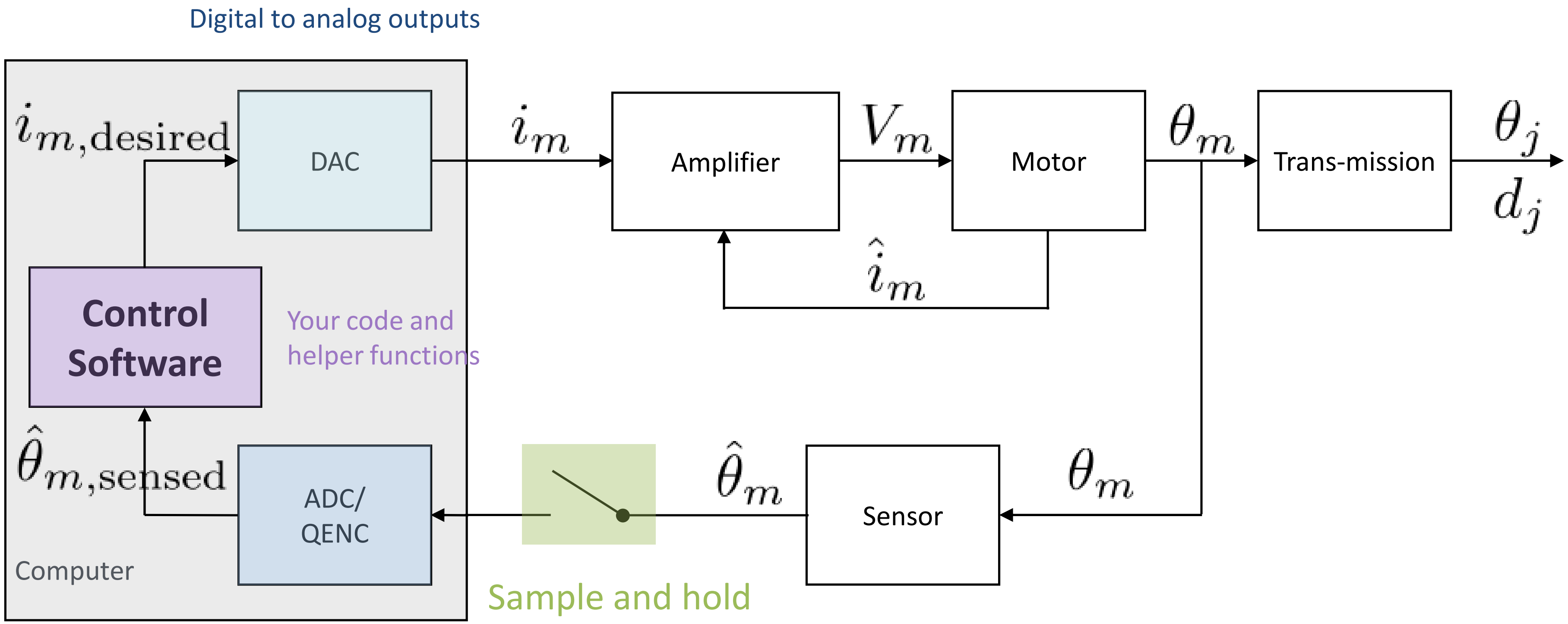


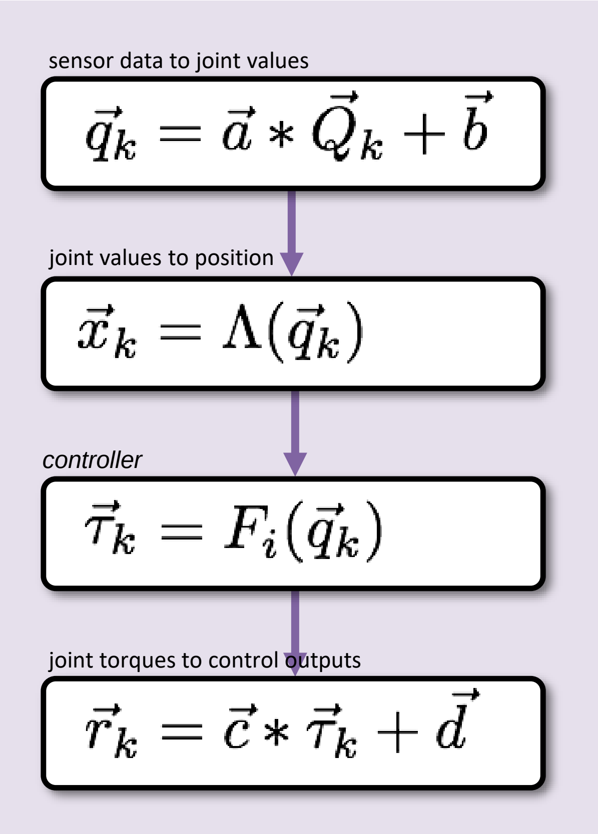
How most real robots work



i : current
V : voltage
m : motor
j : joint
θ : angle
d : displacement
$\hat{}$: estimate

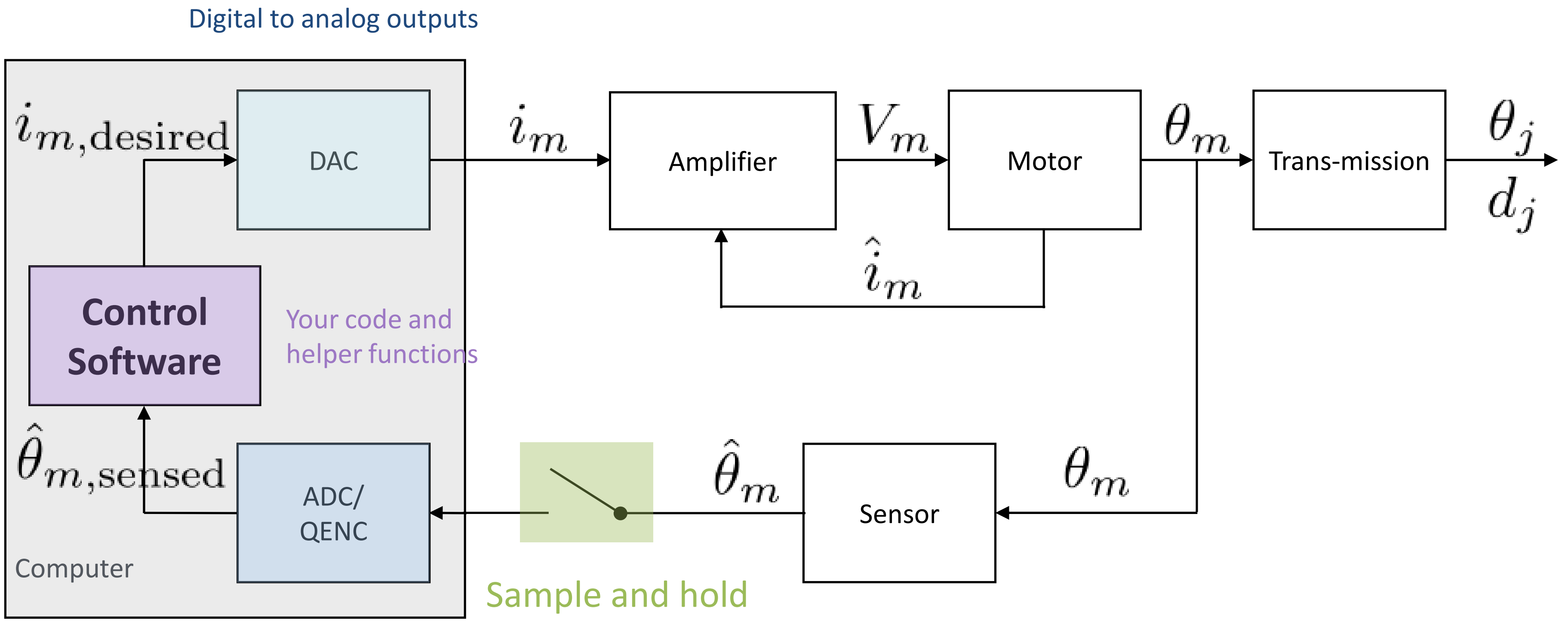
Control Software

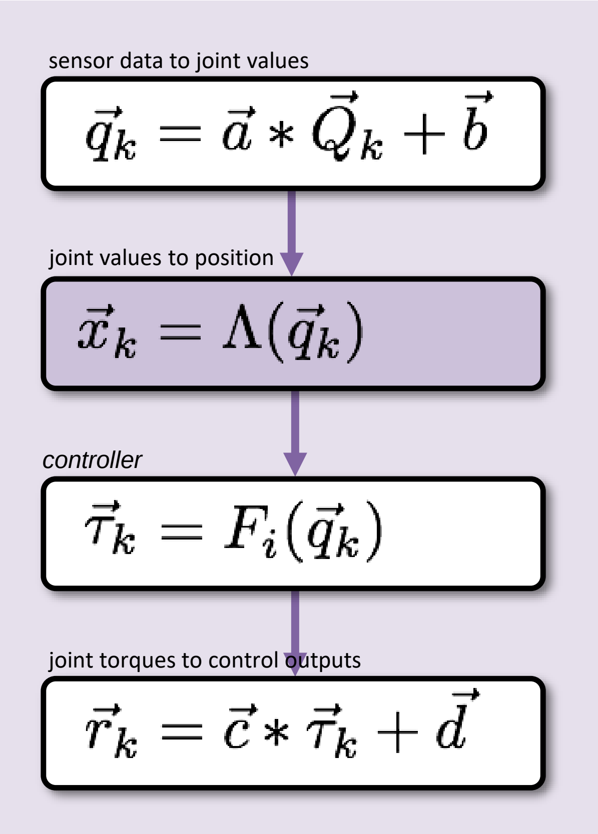




Joint Angles to Robot Position

How can we estimate robot pose from $\hat{\theta}_j$? Forward kinematics!



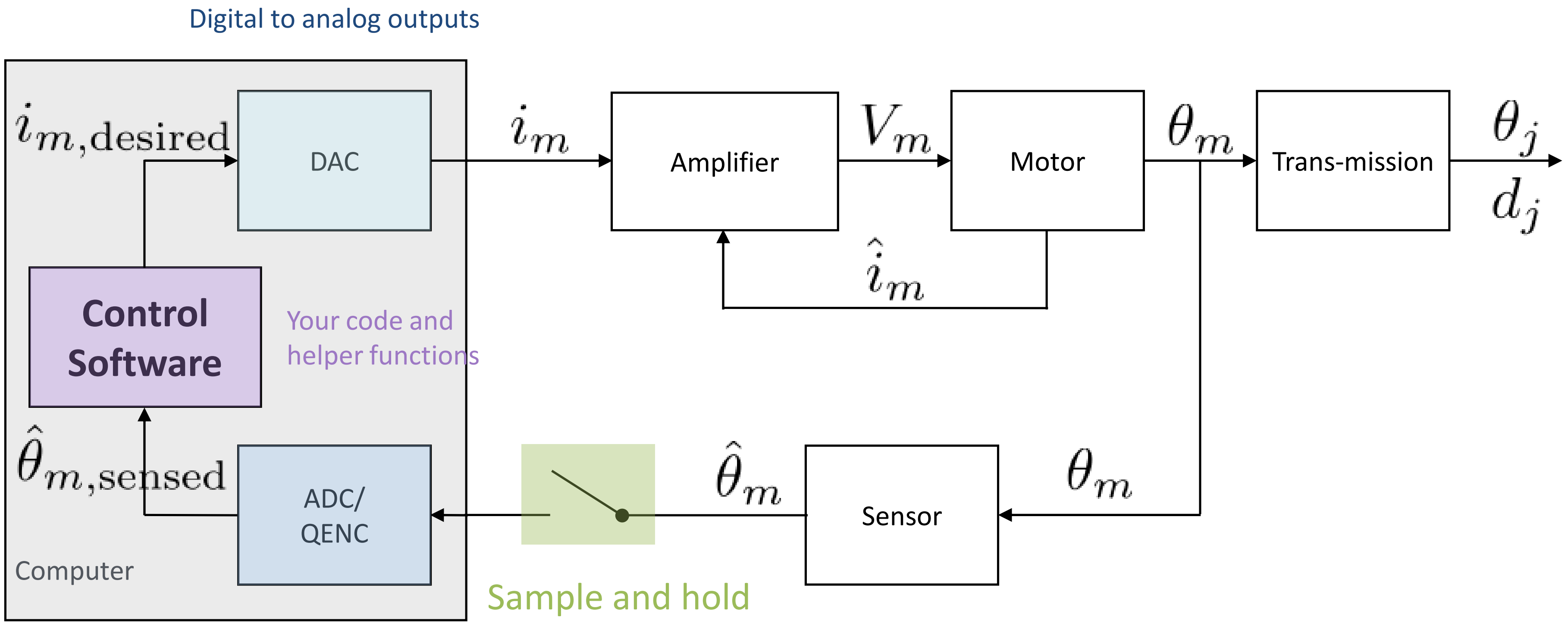


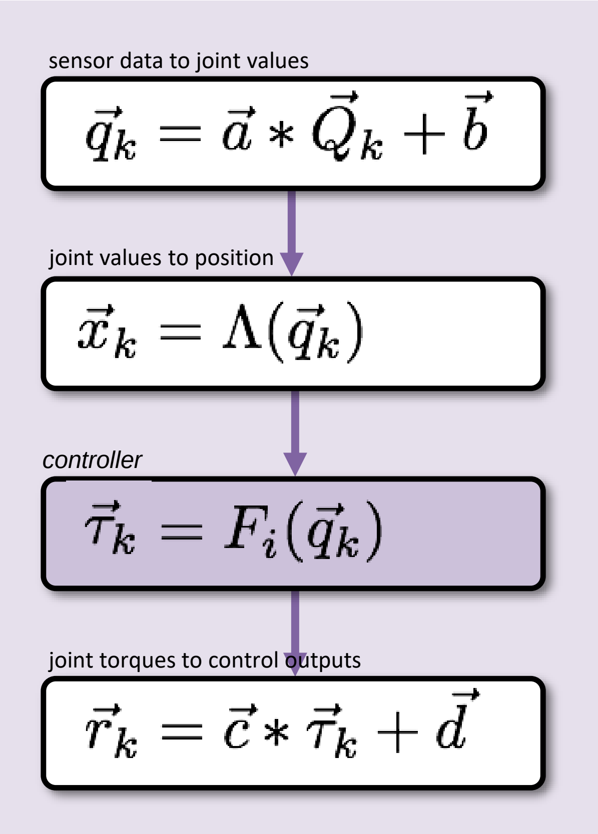
Derive using DH or geometry.

zero pose, link lengths, and link offsets matter!

Robot Positions to Torques

How can we calculate joint torques from positions? PID control!



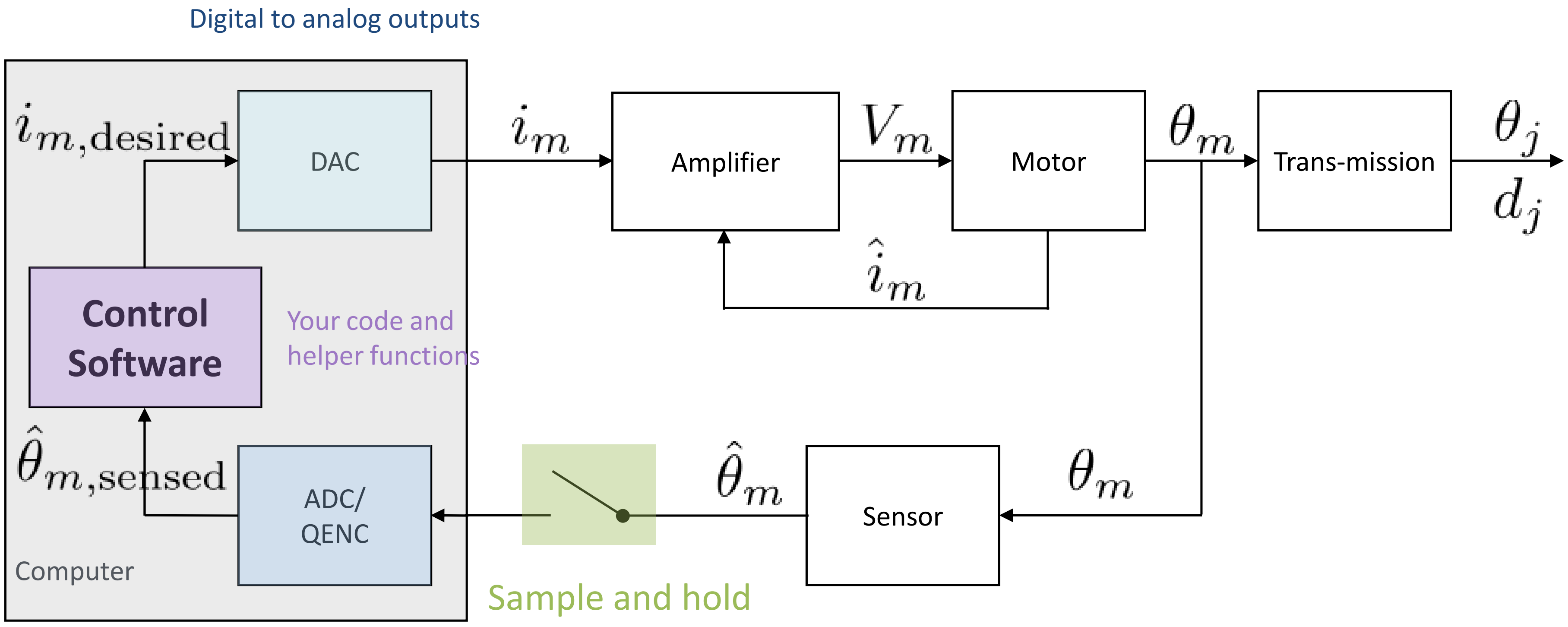


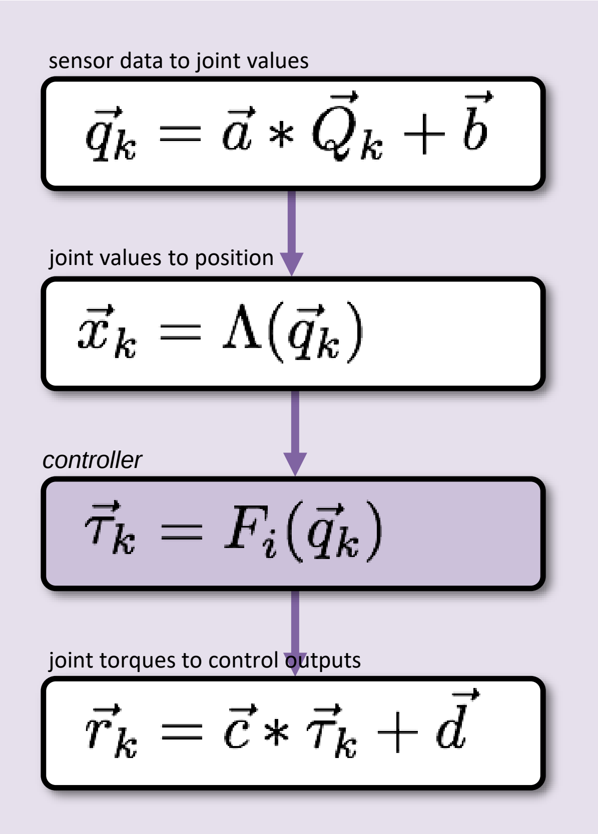
$$\tau_i = k_p(\theta_{i,des} - \theta_i) + k_d(\omega_{i,des} - \omega_i) + k_i \int \theta_{i,des} - \theta_i$$

Alternatively, Robot Positions to Forces to Torques

How can we calculate joint torques from desired forces?

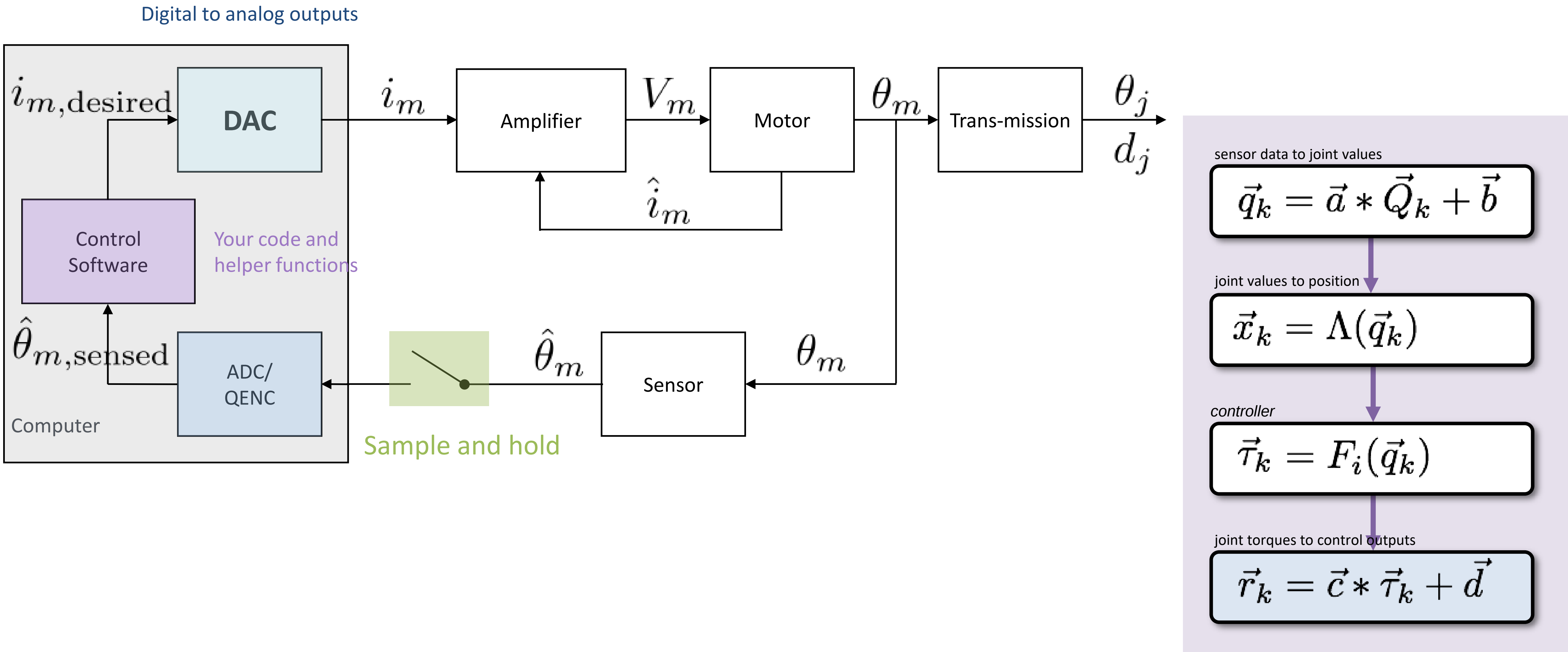
Manipulator equation!



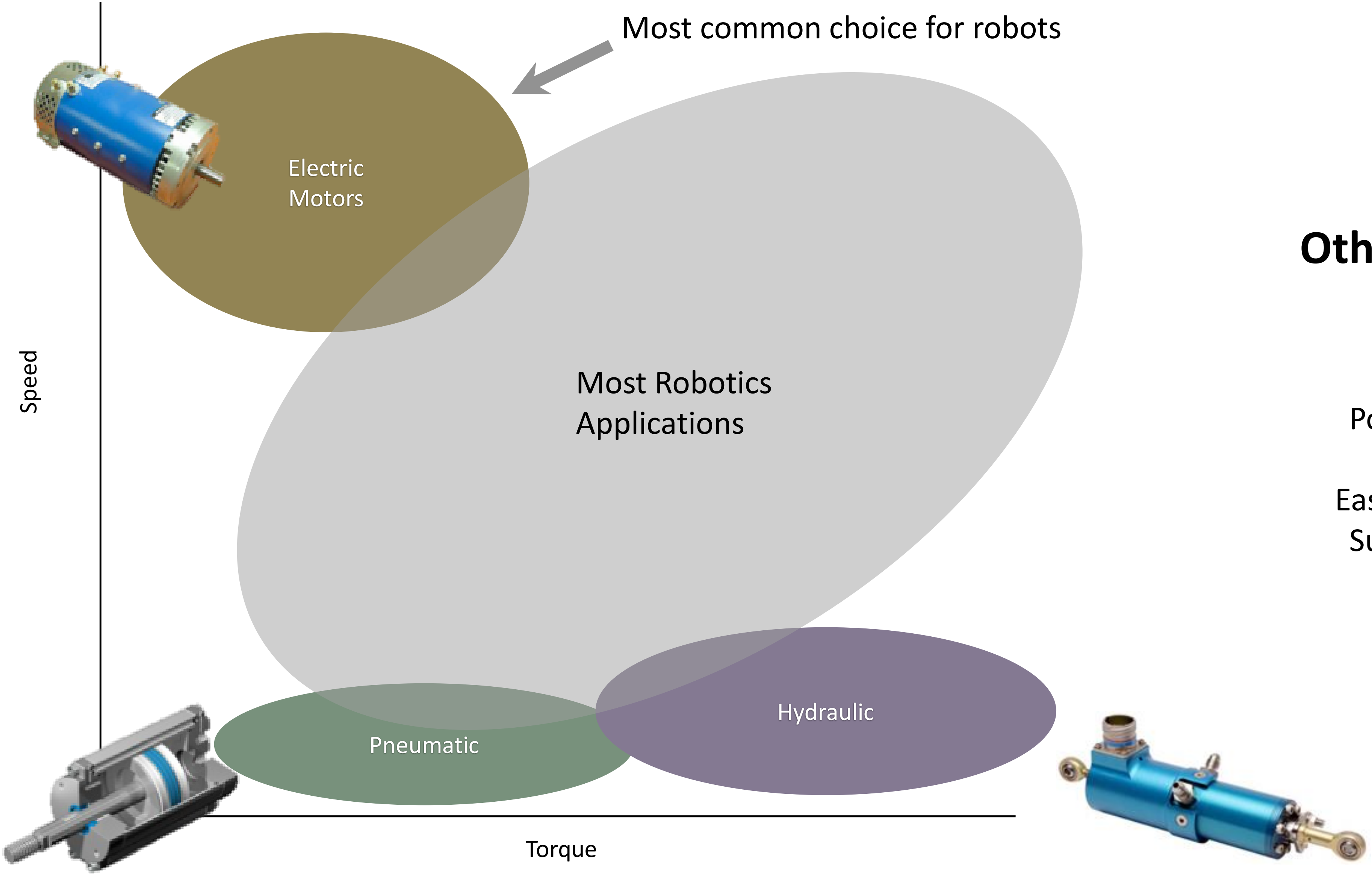


$$\tau = D(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q)$$

What goes in here?



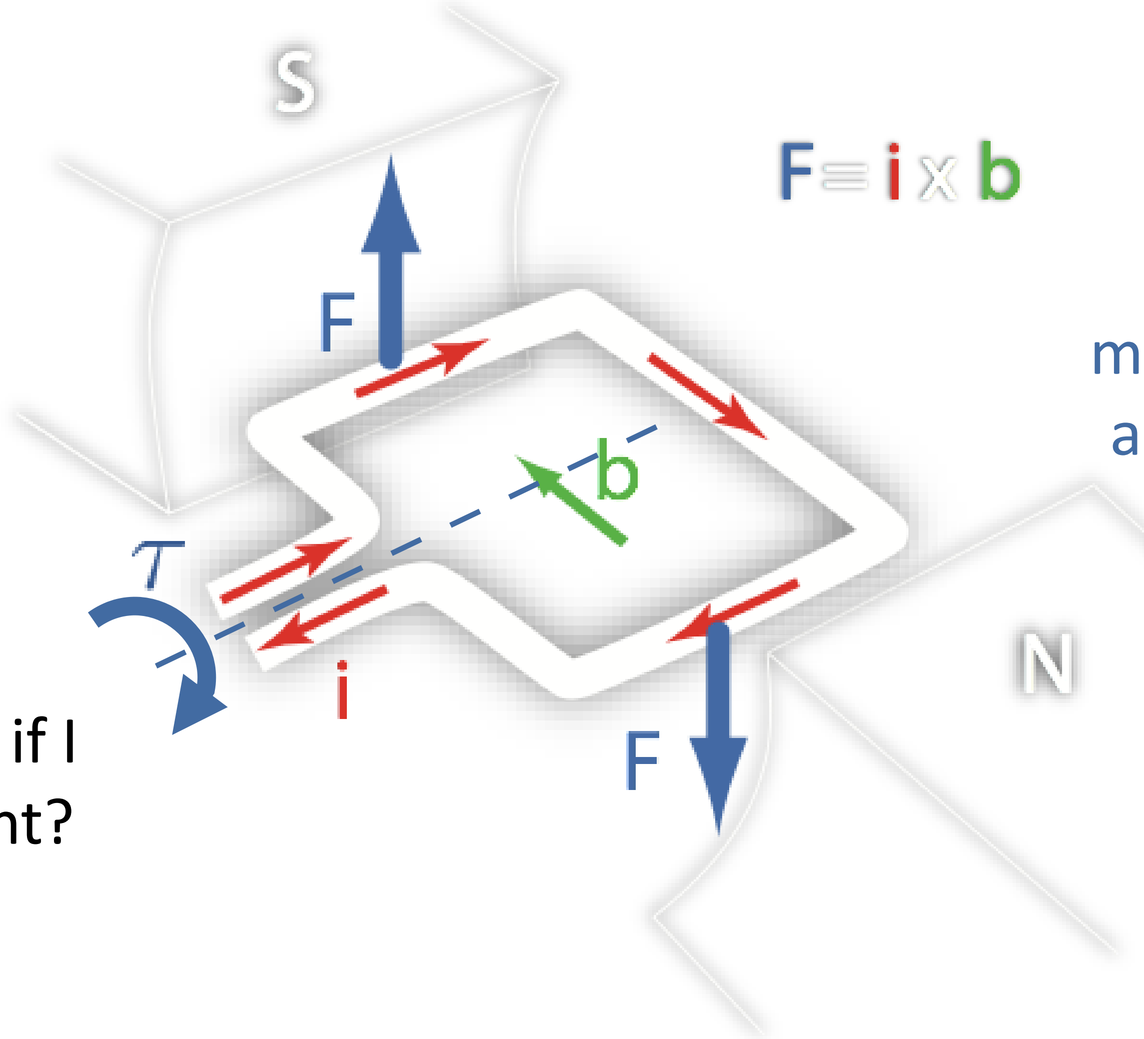
Types of Actuators



Other concerns?

Cost
Availability
Size
Power to Weight
Stiffness
Ease of Integration
Support Systems

Electric Motors

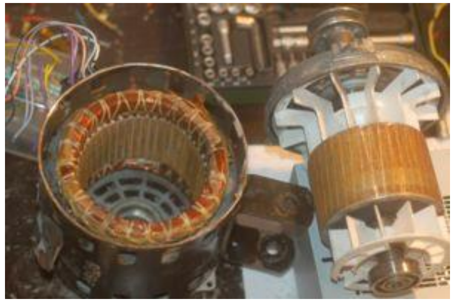


Pair of offset equal-magnitude forces causes a **torque** (a.k.a. couple, moment) around the axis of rotation

What would happen if I turned off the current?

What would happen if I flipped the sign of the current?

What if I kept the current but rotated coil to another position?



AC

Magnetic Rotor
Coil Stator
Output speed is a sub-multiple of voltage supply frequency

DC Brushed

Most common!

Coil Rotor
Magnetic Stator
Brushes carry current to the rotor

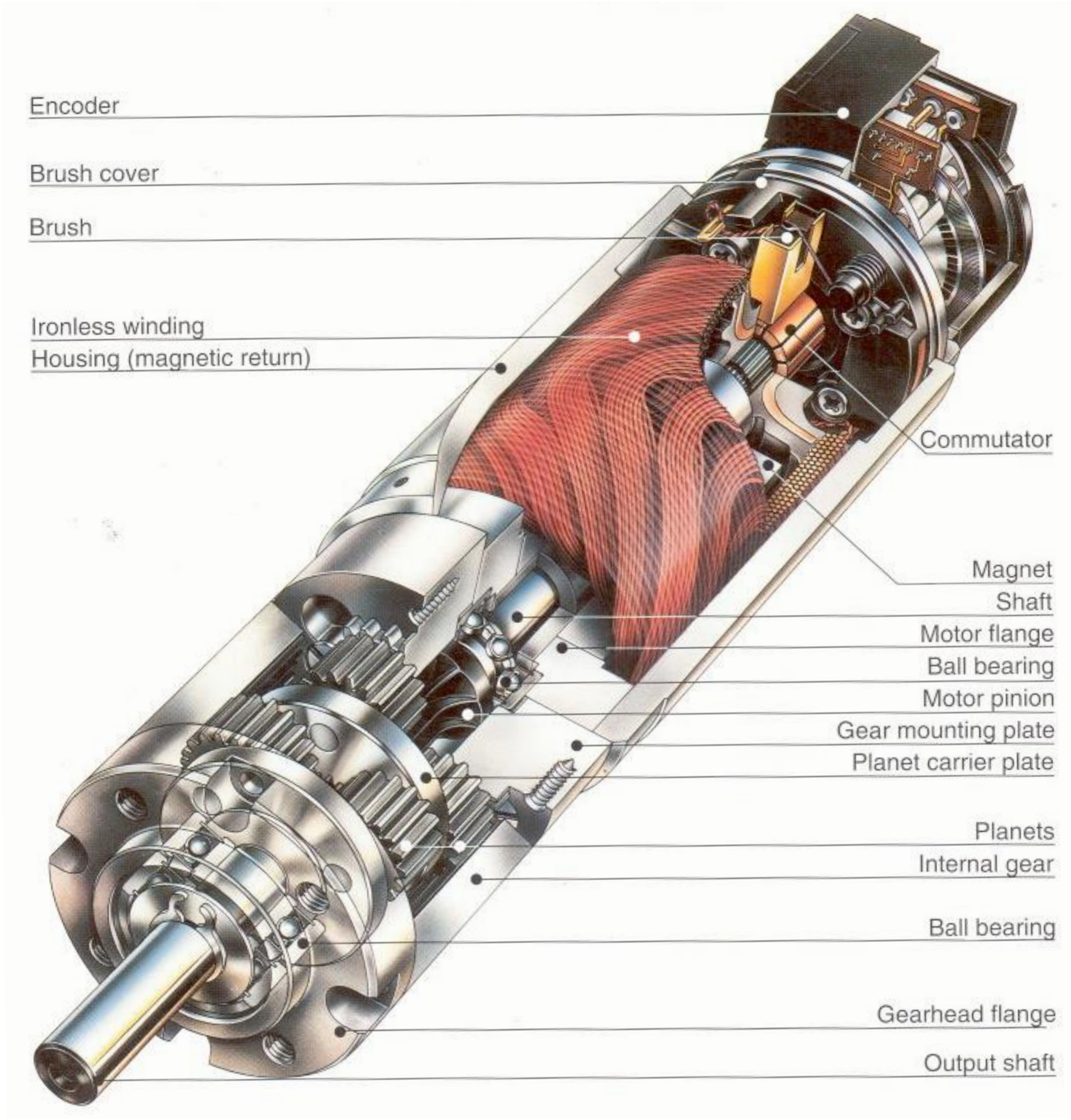
DC Brushless

Magnetic Rotor
Coil Stator
Similar in construction to AC, but electrically commutated
Requires a position sensor (commonly built in)

Stepper

Toothed Magnetic Rotor
Multi-Coil Stator
Capable of open-loop positioning
Requires a controller

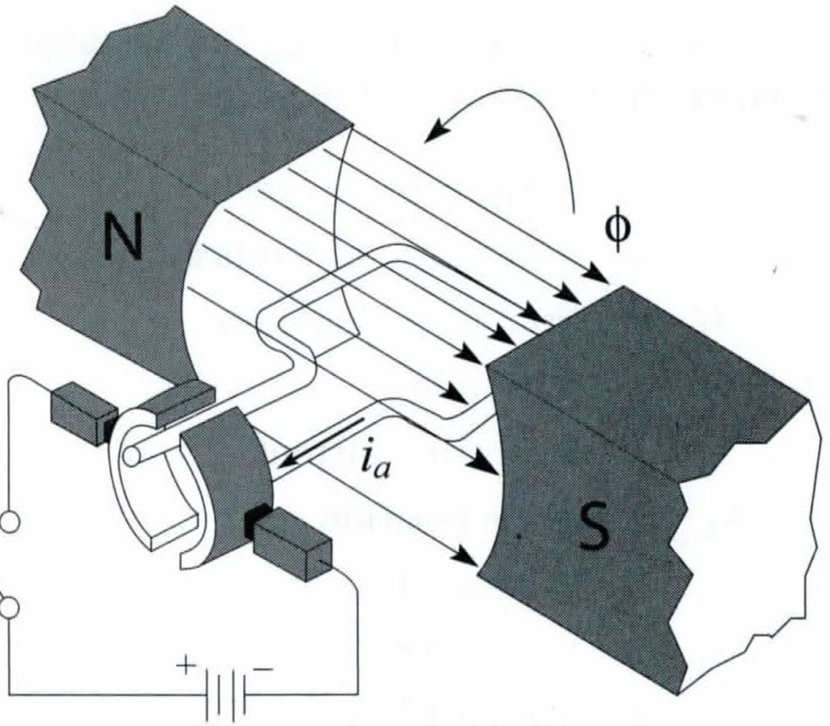
DC Motor

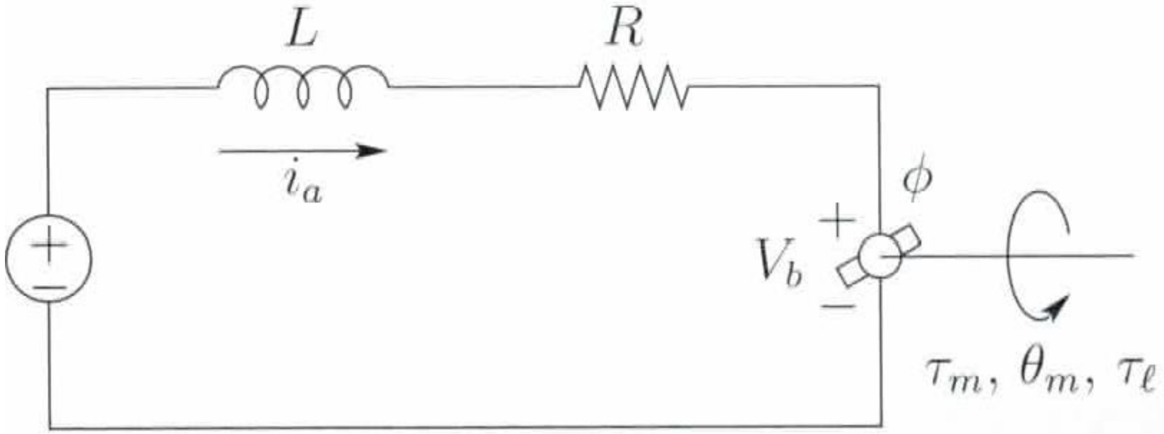


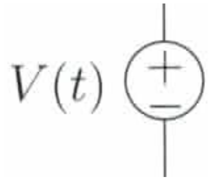
Encoder
Brush cover
Brush
Ironless winding
Housing (magnetic return)
Commutator
Magnet
Shaft
Motor flange
Ball bearing
Motor pinion
Gear mounting plate
Planet carrier plate
Planets
Internal gear
Ball bearing
Gearhead flange
Output shaft

DC Motor

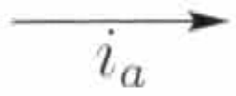
Commutation ensures that the coil with the best mechanical advantage is receiving the current.



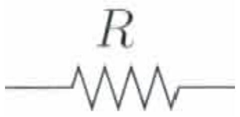




Time-varying **voltage** supply.
Compare **voltage** to water pressure.

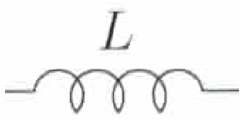


Current through the motor armature.
Compare **current** to water flow.



Electrical **resistance** of the armature.
Compare **resistance** to water flow resistance in pipes (small diameter)
Follows Ohm's Law.

$$V = iR$$

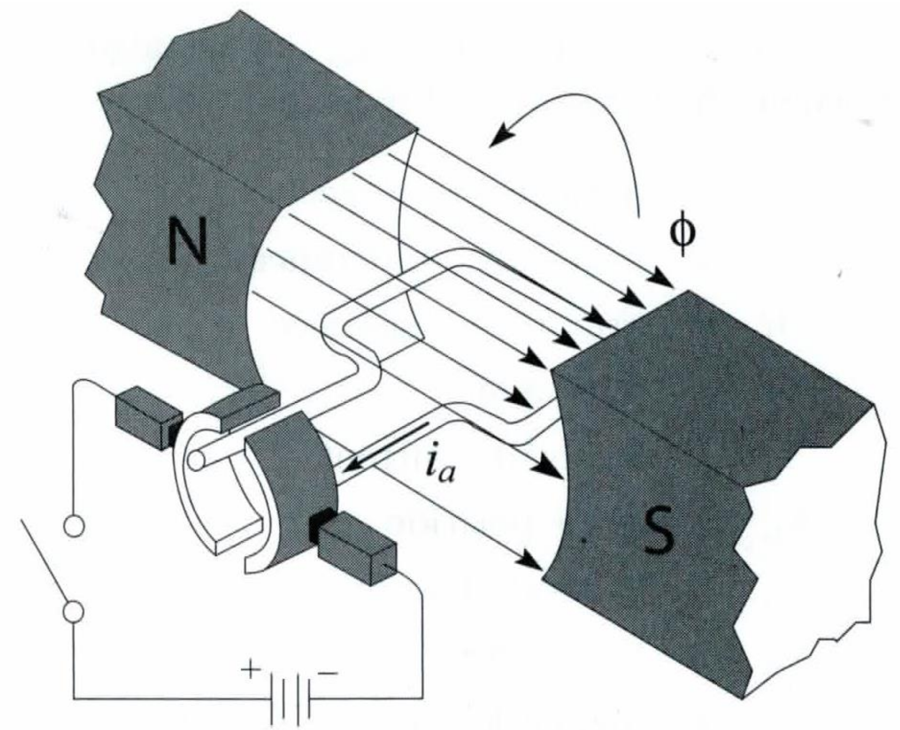


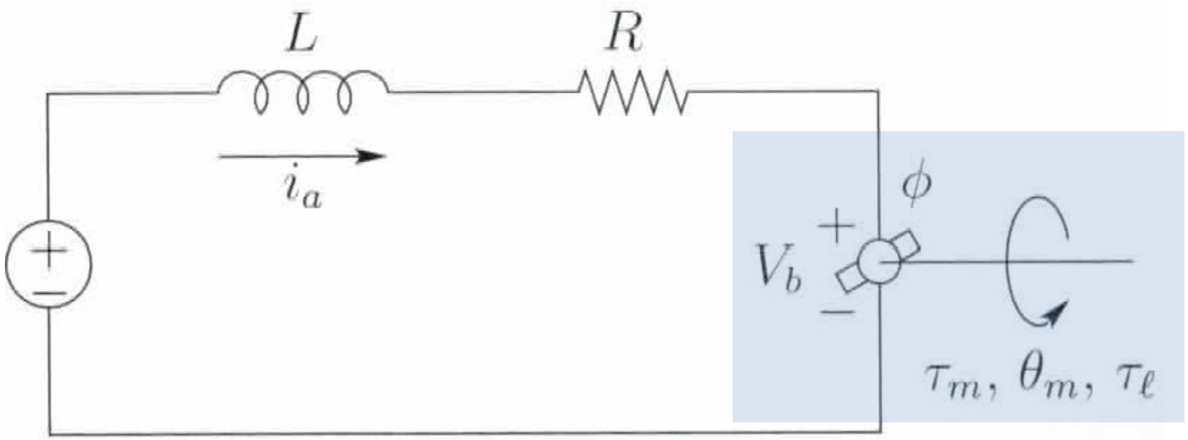
Electrical **inductance** of armature.
Compare **inductance** to the momentum of the water flowing in a pipe.
Follows constitutive equation.

$$V = L\frac{di}{dt}$$

Circuit representation for a DC motor driven by a time-varying voltage.

DC Motor





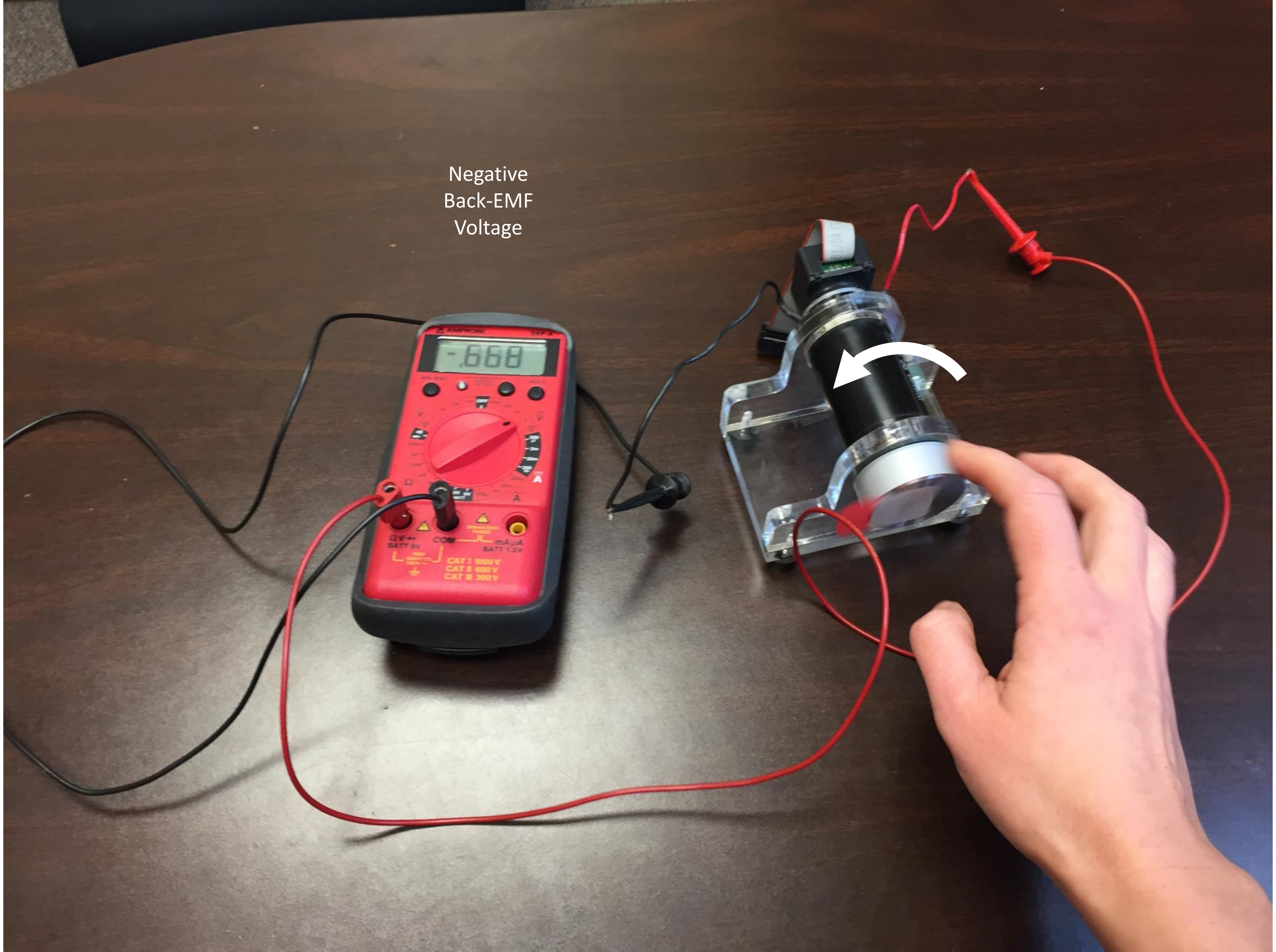
Circuit representation for a DC motor driven by a time-varying voltage.

Time-varying voltage caused by the back electro-motive force (**back-emf**).

Back-emf voltage is proportional to the rotational speed of the motor and opposes the voltage that would drive the motor in the direction in which it is rotating.

$$V_b = k_v\,\omega_m$$


Positive
Back-EMF
Voltage
.798
COM
CAT I 1000 V
CAT II 600 V
CAT III 300 V

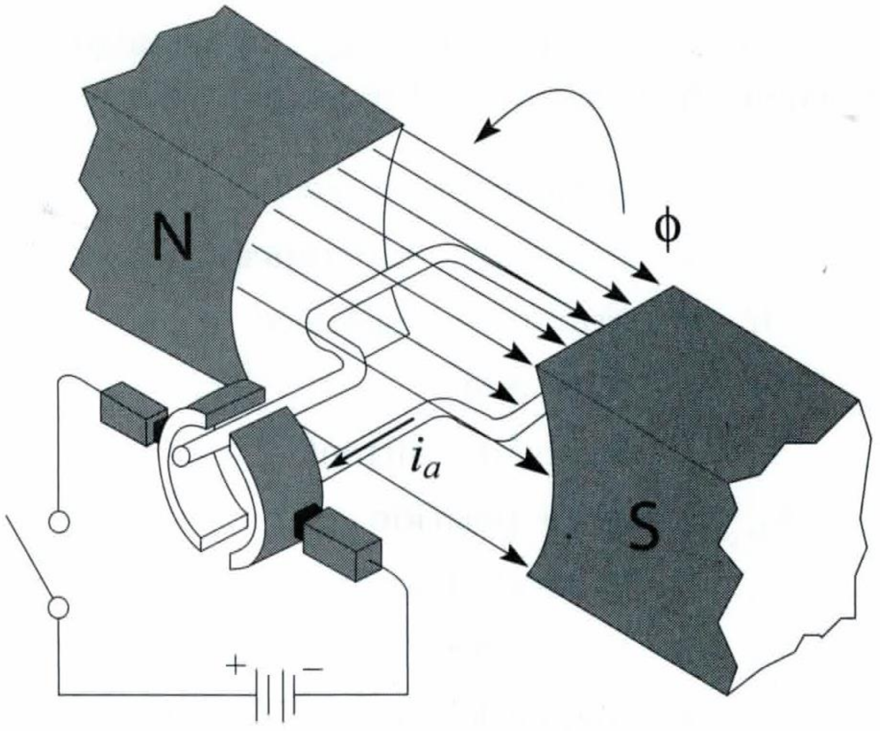

Negative
Back-EMF
Voltage
-.668
COM

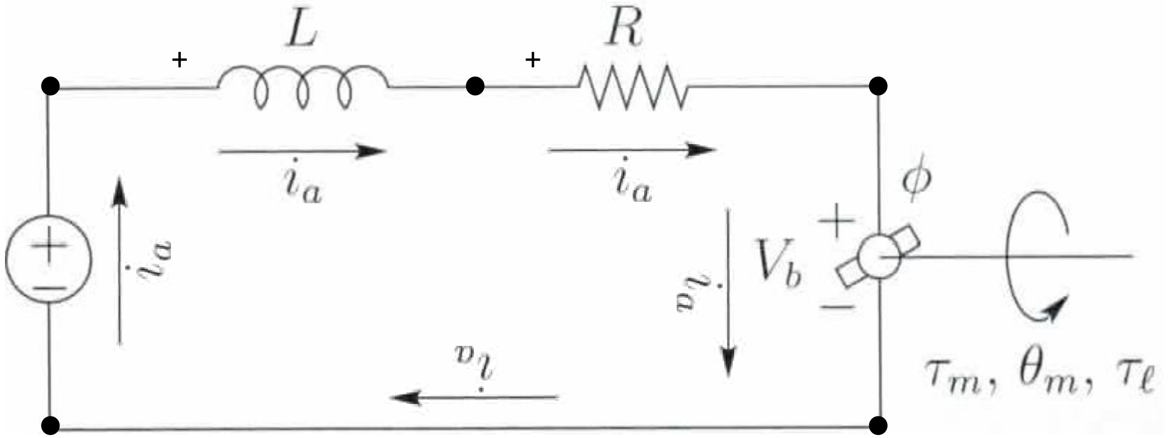
Turning one motor creates back-emf (a voltage across that motor's terminals).

If two motors are connected terminal to terminal,
turning the first motor
makes the other motor turn.

https://youtu.be/6kgzrXFSDwA

DC Motor





How do we analyze this circuit?

Kirchoff's Current Law (KCL)

The sum of the currents flowing into (positive) and out of (negative) a node in the circuit is zero.

The current flowing through all elements of our circuit is the same.

Kirchoff's Voltage Law (KVL)

The sum of the voltage drops around any loop in the circuit is zero.

Draw a + sign where current enters each element other than voltage sources.

$$V(t) - V_L - V_R - V_b = 0$$

$$V = L\frac{di}{dt}$$

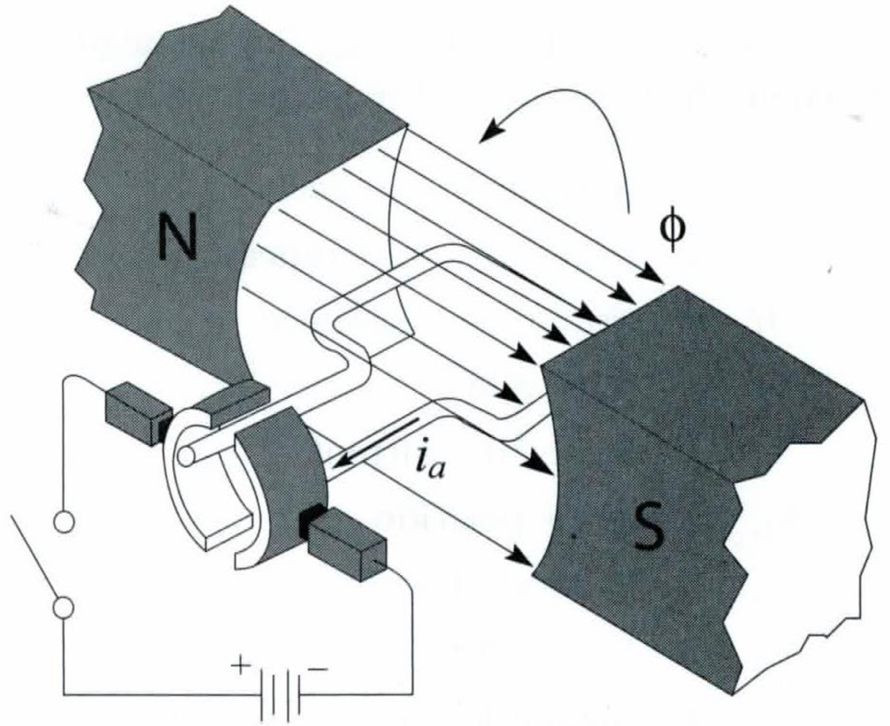
$$V = iR$$

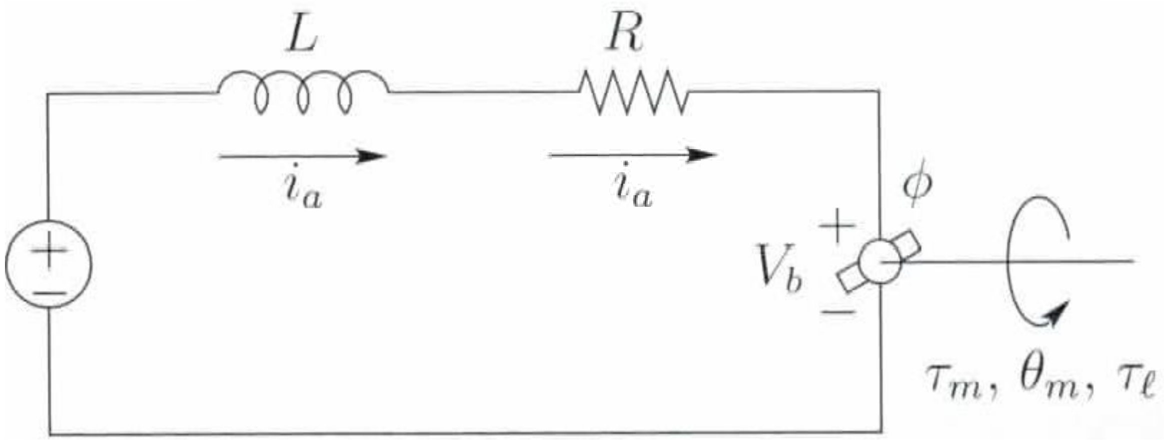
$$V_b = k_v\,\omega_m$$

$$\boxed{V(t) = L\frac{di_a}{d_t} + R\,i_a + k_v\,\omega_m}$$

DC Motor

(SHV Section 6.1)





magnetic flux (webers)

torque constant (N•m/A)

$$\tau_m = K_1\,\phi\,i_a = k_t\,i_a$$

generated torque (N•m)

physical constant

armature current (A)

armature current (A)

$$k_t = k_v$$

if using meters, kilograms and seconds

back emf (V)

magnetic flux (webers)

back-emf constant (V•s)

$$V_b = K_2\,\phi\,\omega_m = k_v\,\omega_m$$

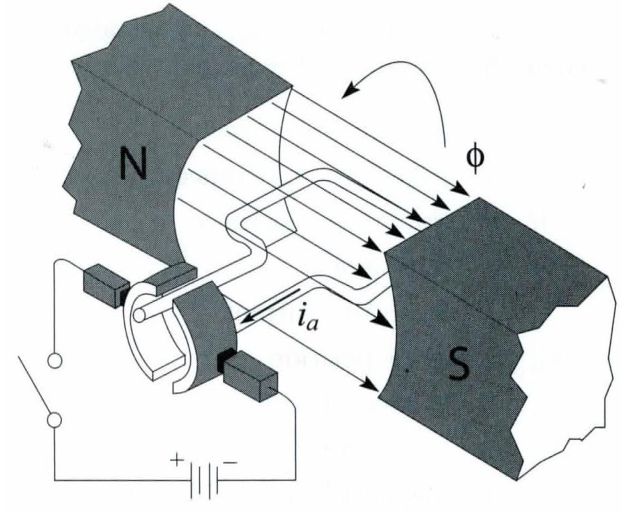
physical constant

motor velocity (rad/s)

motor velocity (rad/s)

(SHV Section 6.1)

DC Motor



Electrical Dynamics

$$V(t) = L\frac{di_a}{dt} + Ri_a + k_v\frac{d\theta_m}{dt}$$

Physical Dynamics

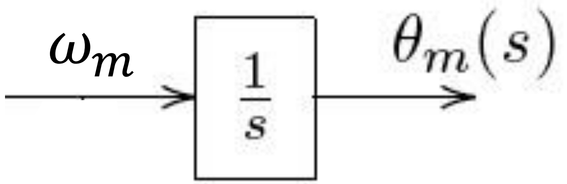
SHV shows the load torque in the wrong direction and confusingly calls gear ratio “r”

$$J_m\frac{d^2\theta_m}{dt^2} + B_m\frac{d\theta_m}{dt} = \tau_m + \tau_{ext} = k_t i_a + \tau_{ext}$$

external disturbances from connections

input

torque constant



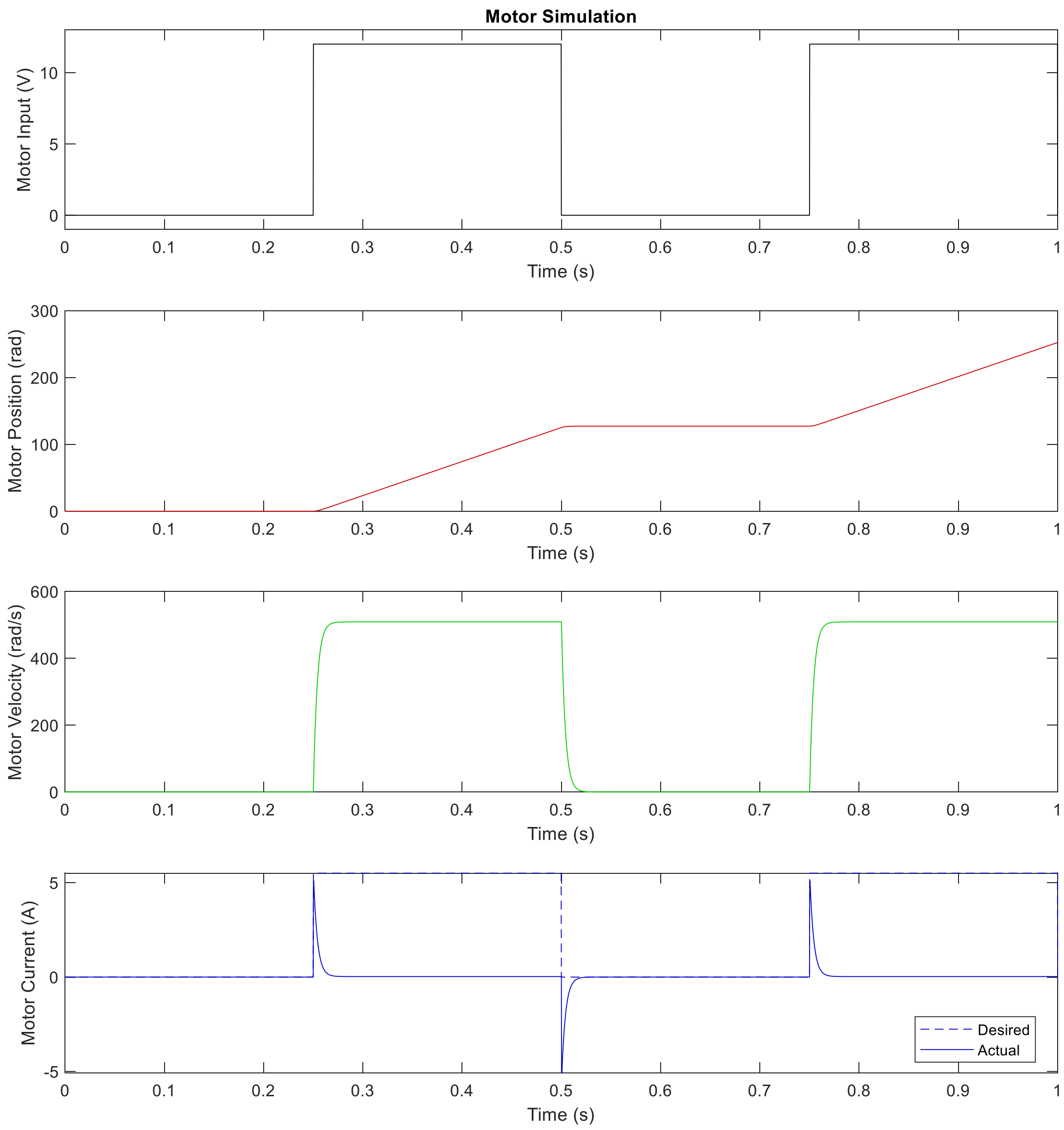
electrical dynamics

motor torque

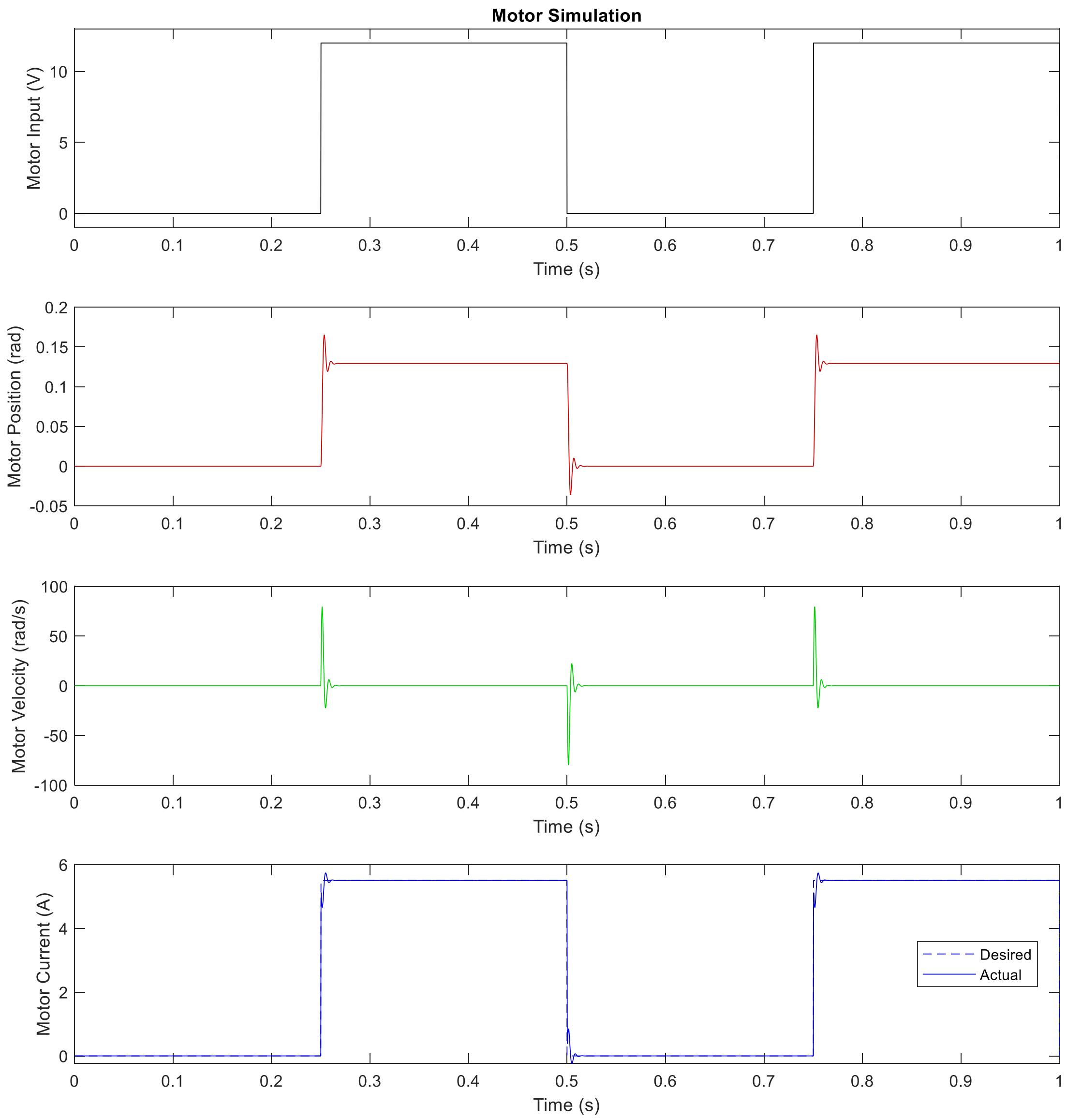
motor physics

back emf constant

No Load



Stall

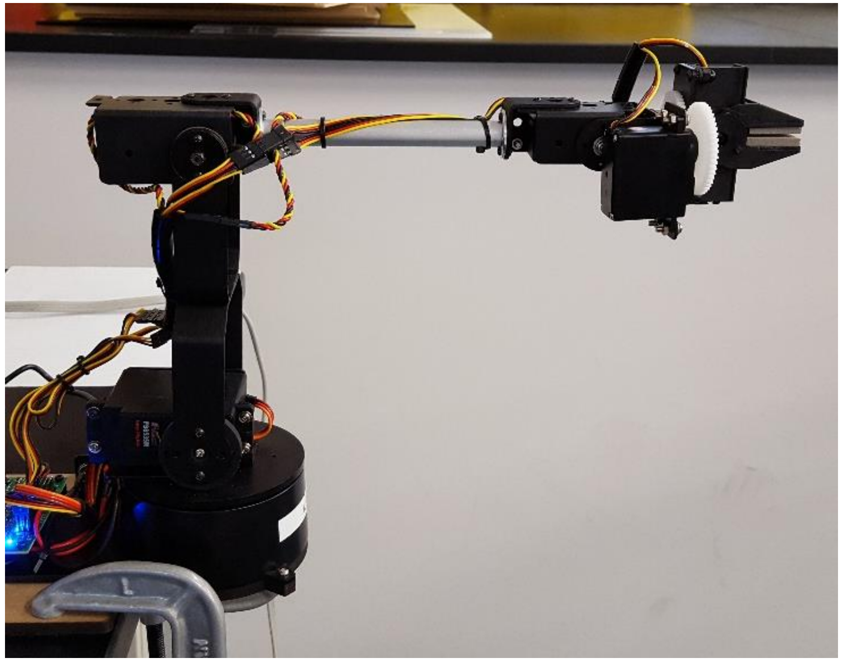


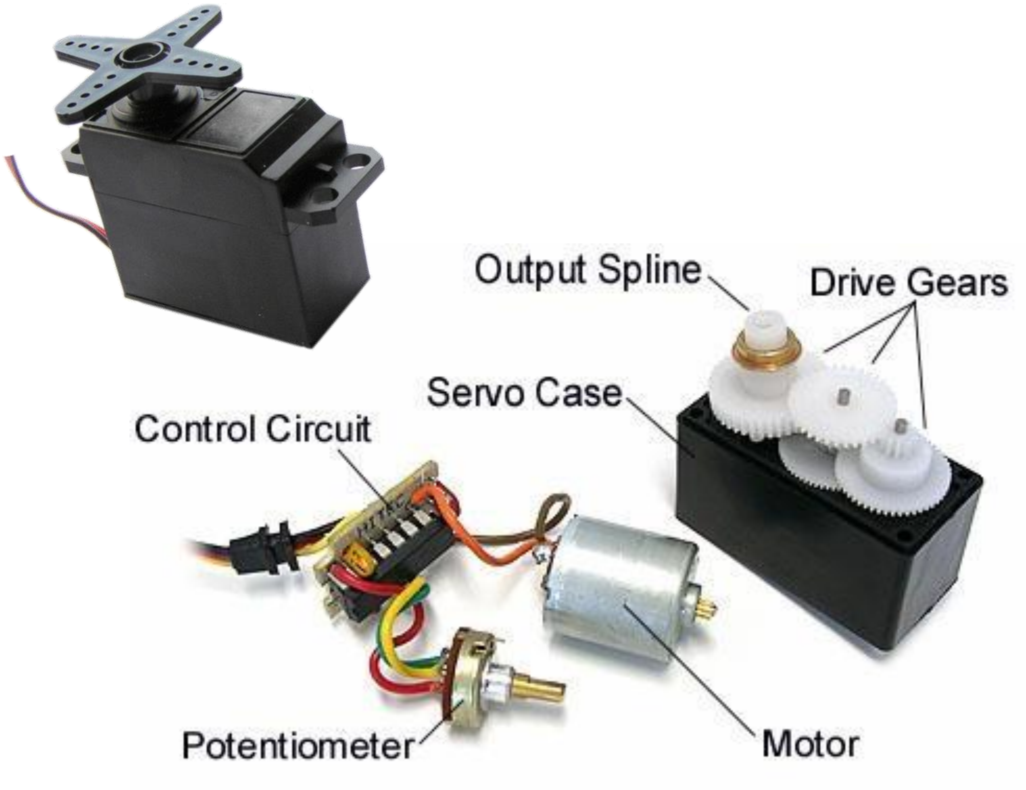
DC Motors

The best brushed DC motors are made by Maxon. They are rather expensive, but they work quite well.

- **Smooth torque output**, independent of motor angle. In other words, very low cogging and torque ripple.
- **Low friction**, both at low and high speeds, due to high quality bearings and low eddy currents.
- **Relatively high stall torque**, which is the torque the motor can deliver when it is not rotating.
- **Larger motors** create higher torques, but they also have higher inertia, higher friction, and higher cost.

Position Control: Servomotors





DC brushed motor
with integrated sensor
and controller

Typically, roboticists treat each joint independently, as a single-input/single-output (SISO) model.

This is adequate for applications that don't involve very fast motions, which helps decouple the links from one another.

Transmission

Most DC motors are designed for high-speed low-torque output.

In order to create robots that can bear load, we need a transmission system.

Common Transmissions

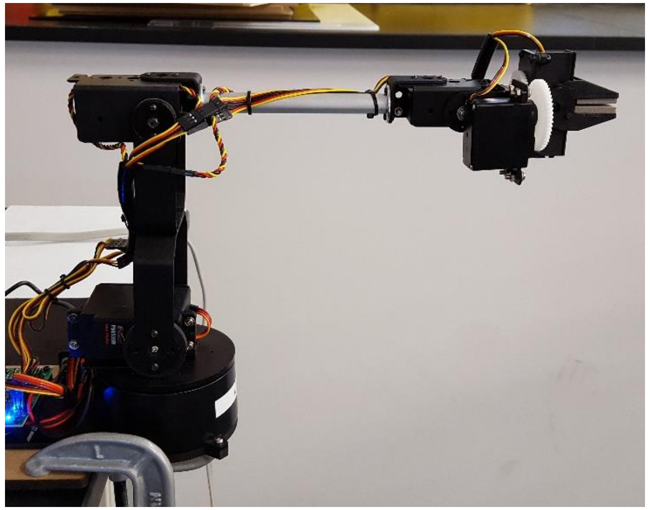
Direct Drive: simplest implementation

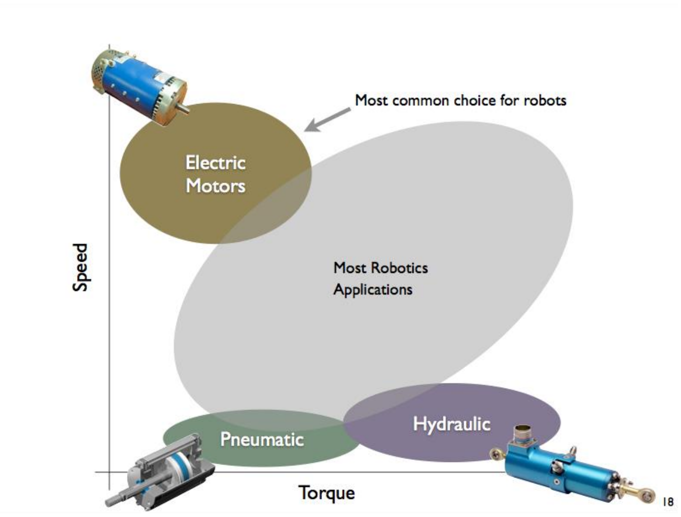
Band/Belt Drive:

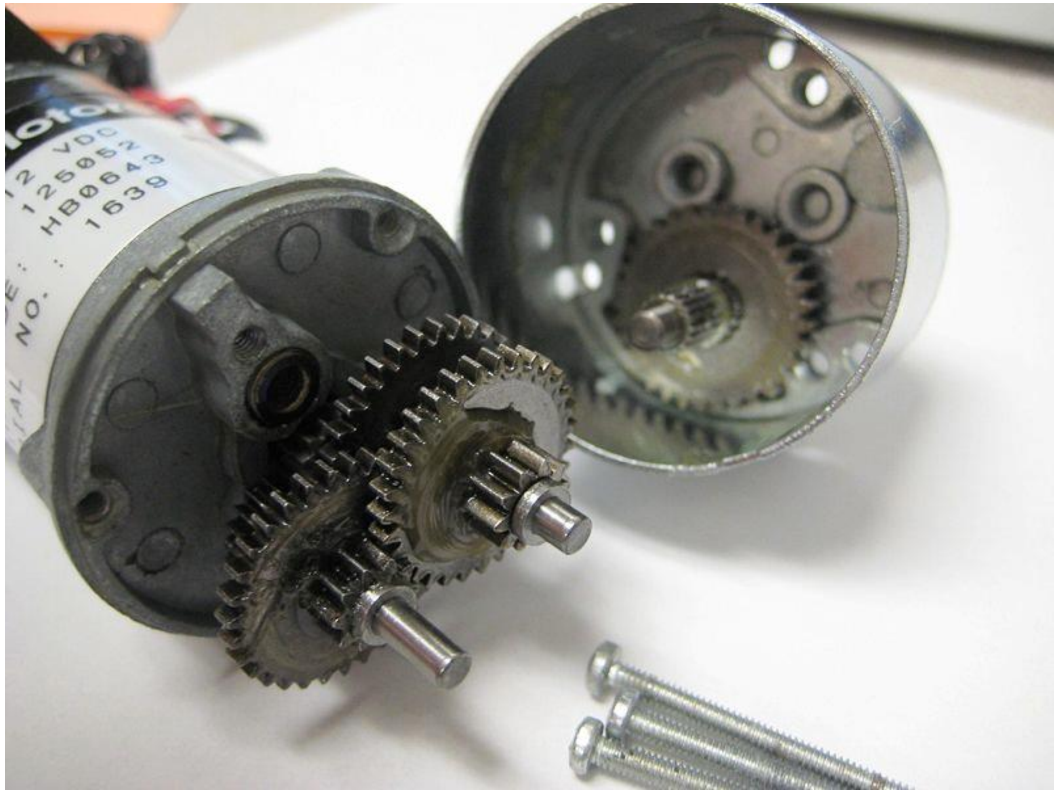
- move actuator mass away from joint
- smoothest drive but unstable when belts are long

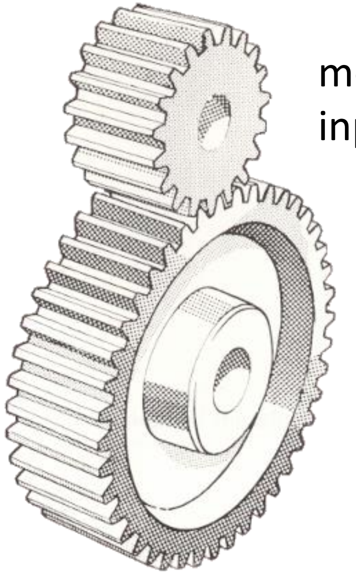
Gear Drive:

- high torque low speed
- backlash









motor input

load output

gear ratio

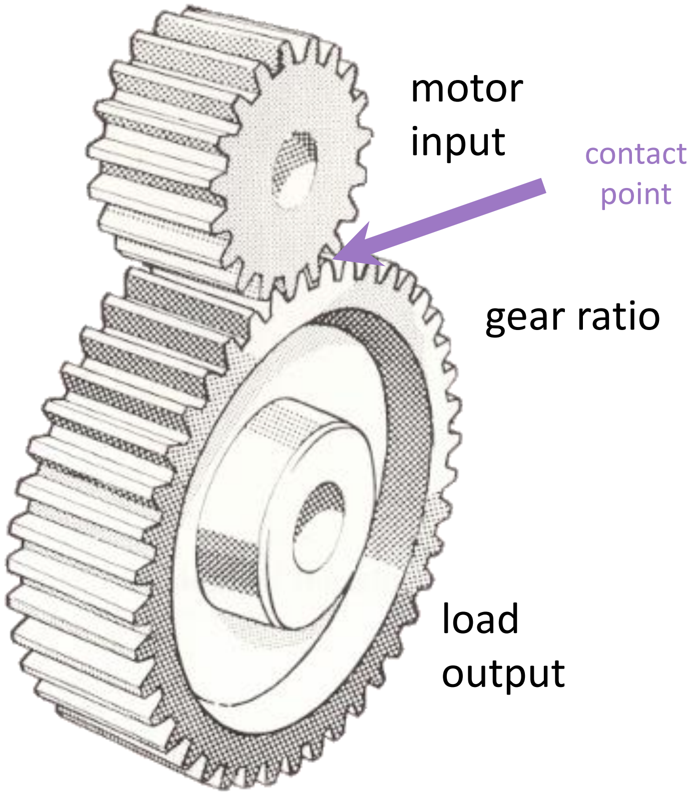
$$N = \frac{n_{\text{out}}}{n_{\text{in}}} = \frac{r_{\text{out}}}{r_{\text{in}}} = \frac{\omega_{\text{in}}}{\omega_{\text{out}}} = \frac{\tau_{\text{out}}}{\tau_{\text{in}}}$$

Teeth | Radius | Speed | Torque

Same equations apply for belts, pulleys, and friction drive

$$N = \frac{n_{\text{out}}}{n_{\text{in}}} = \frac{r_{\text{out}}}{r_{\text{in}}} = \frac{\omega_{\text{in}}}{\omega_{\text{out}}} = \frac{\tau_{\text{out}}}{\tau_{\text{in}}}$$

Teeth Radius Speed Torque



Imagine the small gear rotates an angle equivalent to 5 gear teeth.

How far does the large gear rotate?

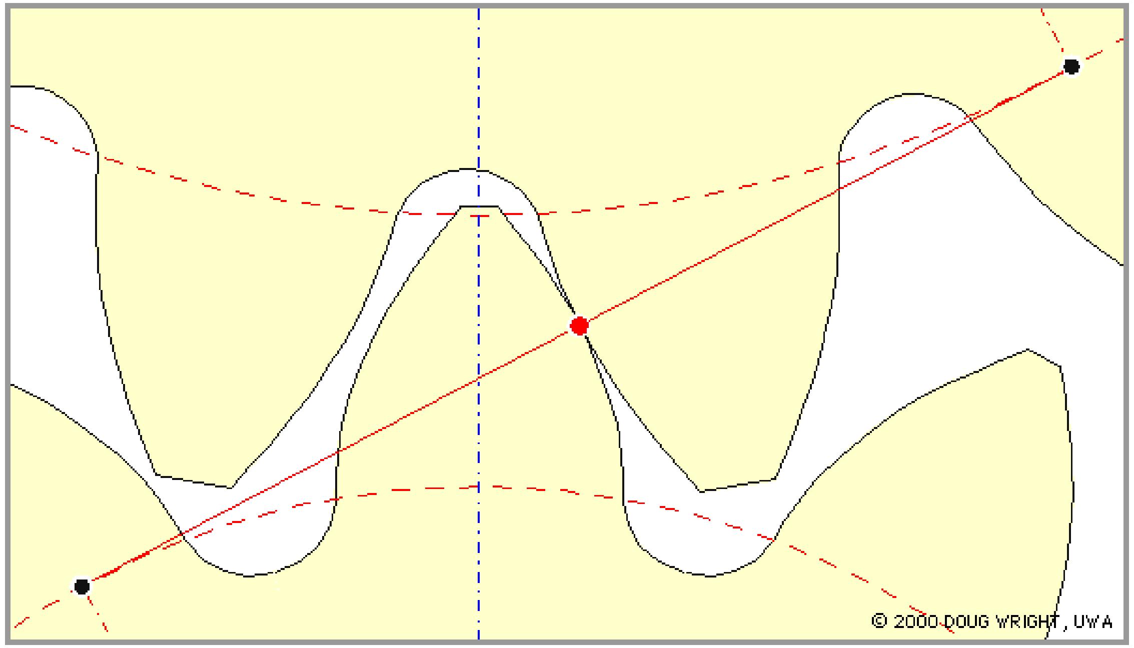
Imagine the you apply a torque of 1 Nm to the small gear while holding the large gear still.

What torque do you feel on the other gear?

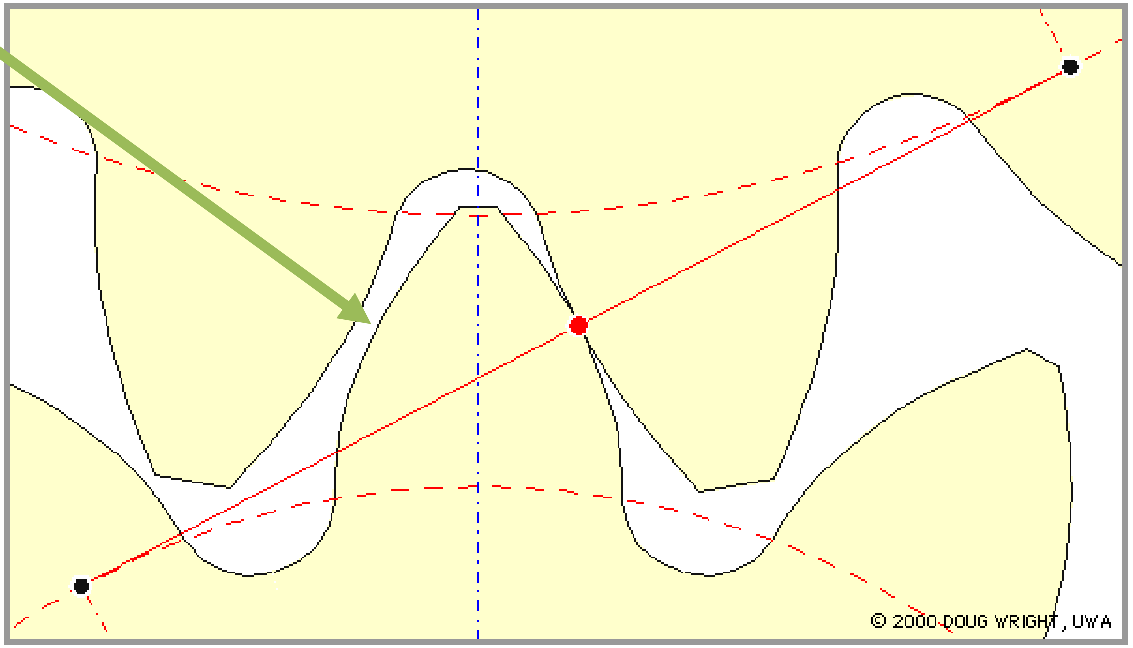
Types of Gears

FIGURE 6.14 Hypoid gears. (Courtesy of Gleason Works.)

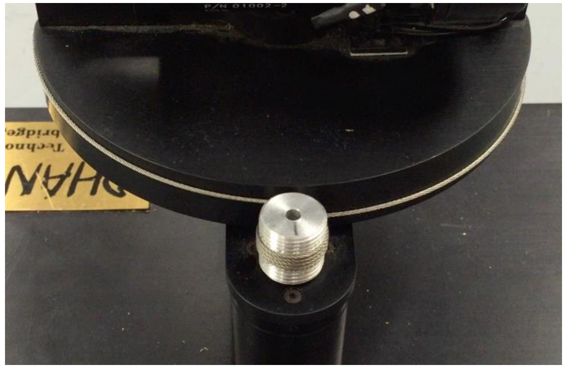
Backlash



Gap between teeth means that if the rotation changes direction, one gear can move a small amount without making the other move.

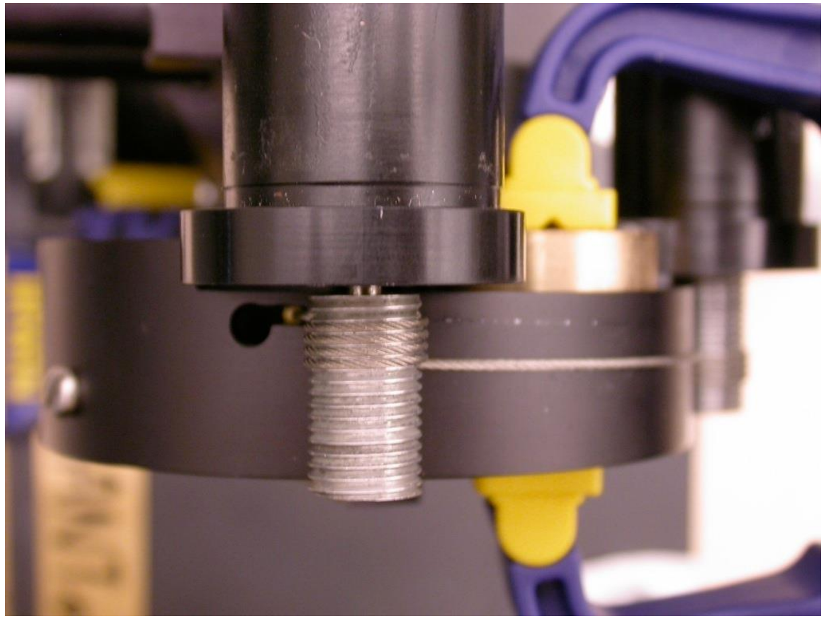


Capstan Drives

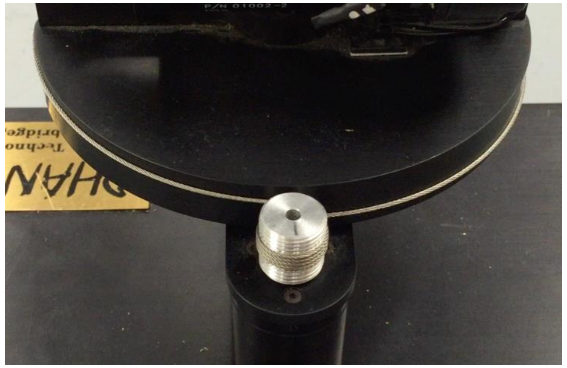


Many high-performance robots use a capstan drive with thin stranded cables.

- The rotation of the motor shaft is coupled to the rotation of a larger drum or the motion of a linear stage by wrapping cables around a capstan.
- When pre-tensioned, cables provide a very stiff connection with zero backlash.
- The motion is smooth and efficient, without vibrations.



Capstan Drives



The gear ratio is the ratio of the diameters (or equivalently the ratio of the radii).

$$\rho = \frac{d_d}{d_c}$$

The drum is almost always larger than the capstan, so rho is greater than one. $\tau_d = \rho\, \tau_m$ $\omega_m = \rho\, \omega_d$

The drum torque is greater than the motor torque.

The motor speed is greater than the drum speed.

A drawback is that the output feels amplified versions of the motor's inertia and friction. This amplification goes with the gear ratio squared.

Next time:

2017 IEEE International Conference on Robotics and Automation (ICRA)
Singapore, May 29 - June 3, 2017

Autonomous Robotic Stone Stacking with Online next Best Object Target Pose Planning

Fadri Furrer*,[1], Martin Wermelinger*,[2], Hironori Yoshida*
Fabio Gramazio[3], Matthias Kohler[3], Roland Siegwart[1], Marco Hutter[2]

Abstract—**Predominately, robotic construction is applied as prefabrication in structured indoor environments with standard building materials. Our work, on the other hand, focuses on utilizing irregular materials found on-site, such as rubble and rocks, for autonomous construction. We present a pipeline that detects randomly placed objects in a scene that are used by our next best stacking pose searching method employing gradient descent with a random initial orientation, exploiting a physics engine. This approach is validated in an experimental setup using a robotic manipulator by constructing balancing vertical stacks without mortars and adhesives. We show the results of eleven consecutive trials to form such towers autonomously using four arbitrarily in front of the robot placed rocks.**

I. Introduction

Over the last decade, robotics has been introduced to architectural construction not only for safer and more efficient construction, but also for exploring diverse forms [1]. However, there are still intensive manual labor works involved for on-site assembly of these components [2].

Digital fabrication has explored applications of autonomous robots in on-site operation scenarios [3], but are restricted to build with regular materials. Building structures with irregular shaped objects was presented in [4], however they apply glue to increase stability. To reduce the environmental impact we aim to use such material without additional adhesives, to build dry-stack compositions. Therefore, our work focuses on developing an automated fabrication process using irregular objects, which are not processed but found on-site.

As a case study, discrete rigid elements, such as stones or concrete rubble, are targeted as a building material. Our goal is to construct a balancing vertical tower with found objects, while maintaining the structure in static equilibrium using a robotic manipulator. To achieve this, we developed a holistic work-flow including precise object detection, motion control, and planning the next target pose (see Figure 1). As part of this work-flow, we describe an algorithm suggesting stable poses for stacking, validated by an implementation of this autonomous stacking work-flow in a real-world experiment. Due to the instability of vertical tower, it is natural to observe errors between a desired target pose and an actual stacked pose. Thus, the work-flow puts emphasis on the resultant pose evaluation and a dynamic re-planning of the target pose.

This paper makes the following contributions regarding handling irregularly shaped objects:

– a pose searching algorithm considering structural stability using a physics engine
– an object detection pipeline
– an autonomous system for constructing balancing vertical towers using a manipulator

II. Related Work

Research in architecture and digital fabrication investigates novel production techniques in which material behavior is linked to fabrication and assembly tasks [5]. Recently, it has

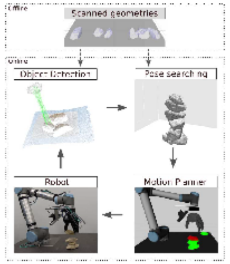


Fig. 1: In an offline step we scan a set of objects (top). These objects, or a subset of it, can be distributed arbitrarily on the work-space and get detected by our object detection pipeline (middle-left). From the detected objects the presented pose searching algorithm proposes the next stable stack (middle-right). A motion planner (bottom-right) is used to generate the trajectories to replicate the proposed stack with the robot arm (bottom-left). After placing the object, its pose is measured and used as base for the subsequent pose searching step.

[1]Autonomous Systems Lab (ASL), ETH Zurich, Switzerland
[2]Robotic Systems Lab (RSL), ETH Zurich, Switzerland
[3]Gramazio Kohler Research (GKR), ETH Zurich, Switzerland
*The authors contributed equally to this work. F.F. was responsible for the object detection, M.W. for the manipulation tasks, H.Y. for the pose searching algorithm.

This work was supported in part by the Swiss National Science Foundation (SNF), the National Centre of Competence in Research Digital Fabrication.

978-1-5090-4633-1/17/$31.00 ©2017 IEEE

“Autonomous Robot Stone Stacking…”
(ICRA 2017)

Final Project

MEAM 520, University of Pennsylvania

November 4, 2018

Project Guidelines

The final project will be an open-ended project. Intro to Robotics is a fast-paced overview of many different topics in robotics. The purpose of the final project is for you to explore one of these topics in more depth.

What to do: The final project can take many forms depending on your interest. Possible final projects include:

- Implementation and evaluation of a method or procedure we learned in class but did not use in lab. This option should have a substantial evaluation and analysis leading to conclusions of when the method works or does not work.
- Literature review on recent results in an area discussed in class. I would expect you to read and synthesis at least 10 papers.
- Research project on a open-ended robotics problem. This is ambitious. I suggest you couple this with a small lit review so you have something to report.

Topics: Possible topics include any topics on the course schedule. Sample topics are listed at the end of this document.

Scope: The project should be equivalent in effort to 1/2 lab per person.

Partners: The project can be done in groups of up to four people. Although not required, I encourage you to do the project in a group so that you can get outside feedback.

Overlap with projects outside of class: This project does NOT have to be an isolated effort. You are strongly encouraged to do a project that you are actually interested in. I support doing a project related to your work in another course (such as MEAM 510 or senior design) or in a research group.

Project Proposal (due 11/16)

Your project proposal should have the following information:

- **Goal:** Summarize what you want to accomplish
- **Approach:** Describe what techniques you plan to use
- **Relation to the course:** Explain the relevance to the course material

The project proposal is a completion grade worth 5 pts on Canvas. Make sure to submit the proposal as a group, following the Submission Instructions below. After the due date, I will provide you with feedback.

Final project proposal due 11/16