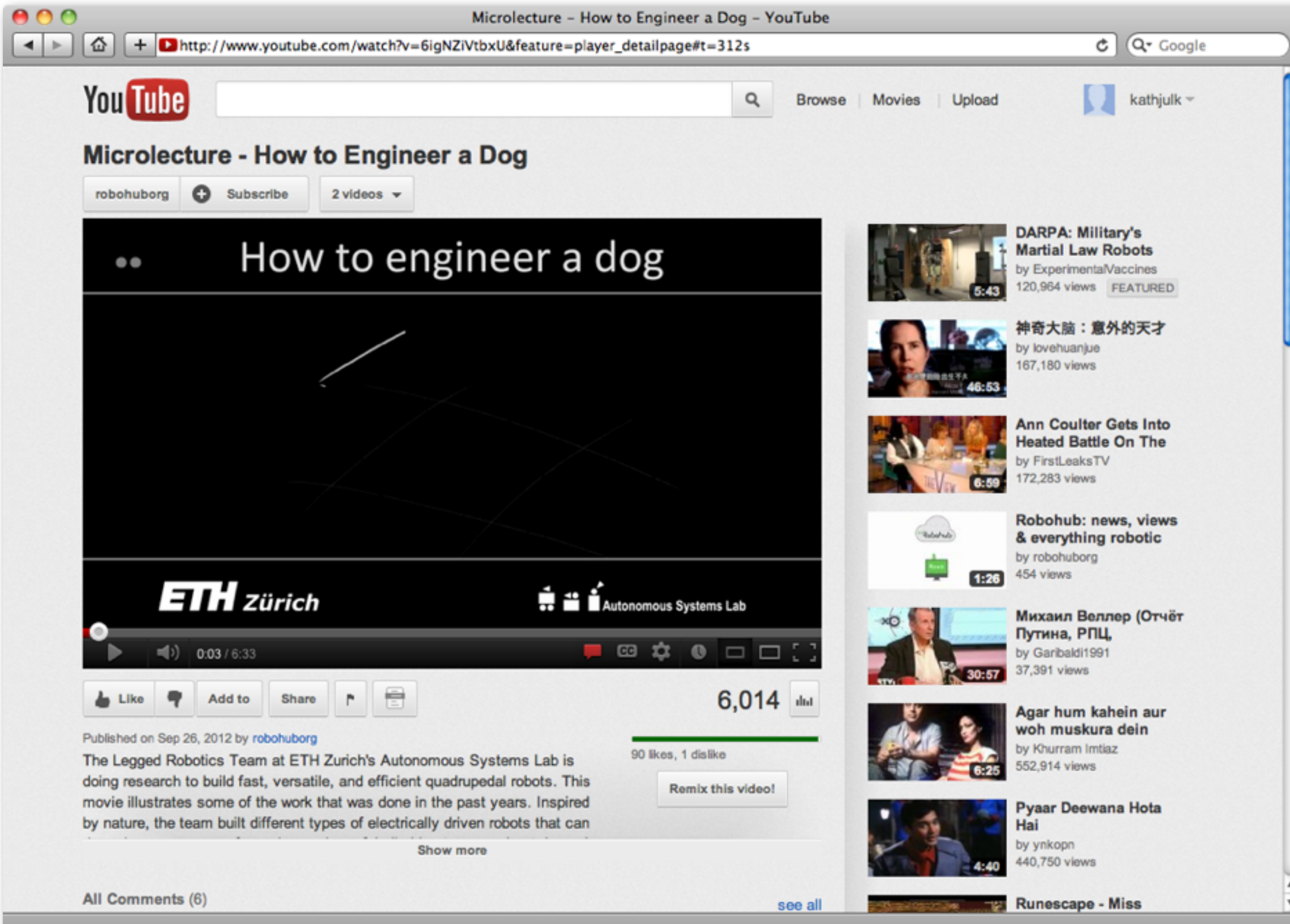
MEAM 520

Haptic Interface Hardware

Katherine J. Kuchenbecker, Ph.D.
General Robotics, Automation, Sensing, and Perception Lab (GRASP)
MEAM Department, SEAS, University of Pennsylvania

Lecture 16: November 13, 2012


Microlecture – How to Engineer a Dog – YouTube
http://www.youtube.com/watch?v=6igNZiVtbxU&feature=player_detailpage#t=312s
Google
YouTube
Browse
Movies
Upload
kathjulk
Microlecture - How to Engineer a Dog
robohuborg
Subscribe
2 videos
How to engineer a dog
ETH Zürich
Autonomous Systems Lab
0:03 / 6:33
Like
Add to
Share
6,014
Published on Sep 26, 2012 by robohuborg
The Legged Robotics Team at ETH Zurich's Autonomous Systems Lab is doing research to build fast, versatile, and efficient quadrupedal robots. This movie illustrates some of the work that was done in the past years. Inspired by nature, the team built different types of electrically driven robots that can
90 likes, 1 dislike
Remix this video!
Show more
All Comments (6)
see all
DARPA: Military's Martial Law Robots
by ExperimentalVaccines
120,964 views
FEATURED
5:43
神奇大脑：意外的天才
by lovehuanjue
167,180 views
46:53
Ann Coulter Gets Into Heated Battle On The
by FirstLeaksTV
172,283 views
6:59
Robohub: news, views & everything robotic
by robohuborg
454 views
1:26
Михаил Веллер (Отчёт Путина, РПЦ,
by Garibaldi1991
37,391 views
30:57
Agar hum kahein aur woh muskura dein
by Khurram Imtiaz
552,914 views
6:25
Pyaar Deewana Hota Hai
by ynkopn
440,750 views
4:40
Runescape - Miss

Name ____________________

Midterm Exam

MEAM 520, Introduction to Robotics
University of Pennsylvania
Katherine J. Kuchenbecker, Ph.D.

November 8, 2012

You must take this exam independently, without assistance from anyone else. You may bring in a calculator and two 8.5"×11" sheets of notes for reference. Aside from these two pages of notes, you may not consult any outside references, such as the textbook or the Internet. Any suspected violations of Penn's Code of Academic Integrity will be reported to the Office of Student Conduct for investigation.

This exam consists of several problems. We recommend you look at all of the problems before starting to work. If you need clarification on any question, please ask a member of the teaching team. When you work out each problem, please show all steps and $\boxed{\text{box your answer}}$. On problems involving actual numbers, please keep your solution symbolic for as long as possible; this will make your work easier to follow and easier to grade. The exam is worth a total of 100 points, and partial credit will be awarded for the correct approach even when you do not arrive at the correct answer.

	Points	Score
Problem 1	20	
Problem 2	20	
Problem 3	15	
Problem 4	20	
Problem 5	25	
Total	100	

I agree to abide by the University of Pennsylvania Code of Academic Integrity during this exam. I pledge that all work is my own and has been completed without the use of unauthorized aid or materials.

Signature ____________________

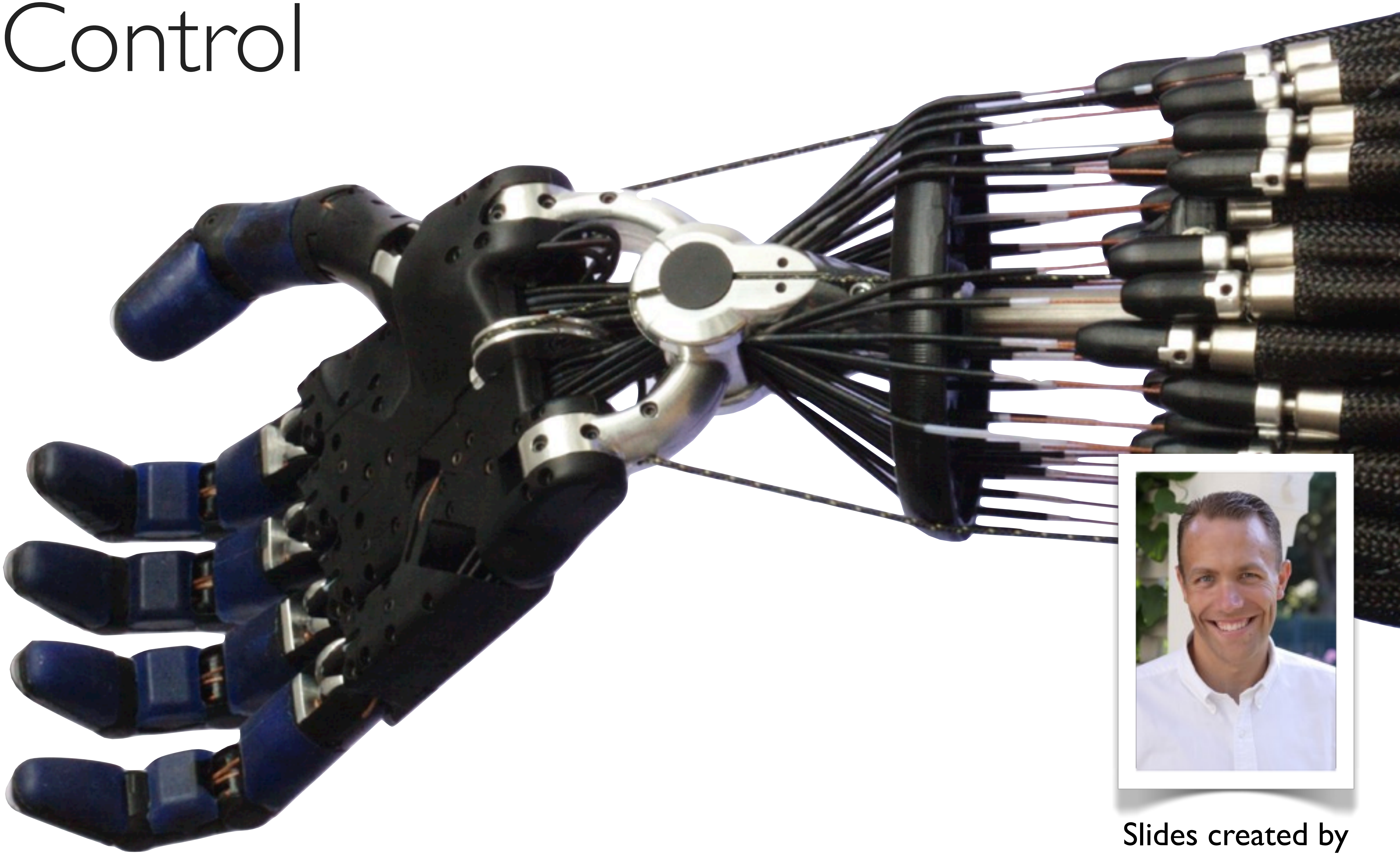
Date ____________________

Probably graded by next Tuesday

Update?

Project 1 : PUMA Light Painting

Manipulator Hardware and Control



Slides created by Jonathan Fiene

A Biological Inspiration

Mechanical Structure

Bones	Frame / Links
Joints	Joints

Actuators

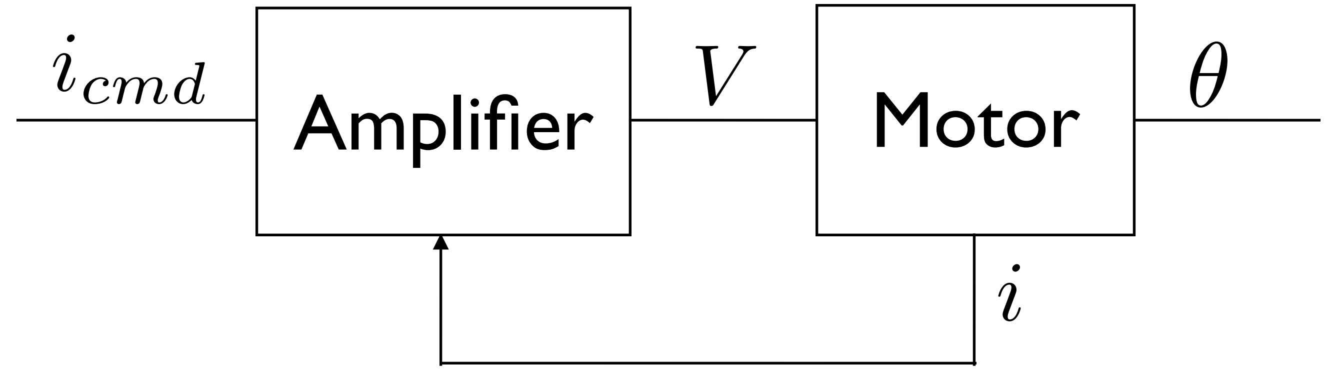
Muscles	Electric Motors
	Hydraulics
	Pneumatics
	SMA, etc.

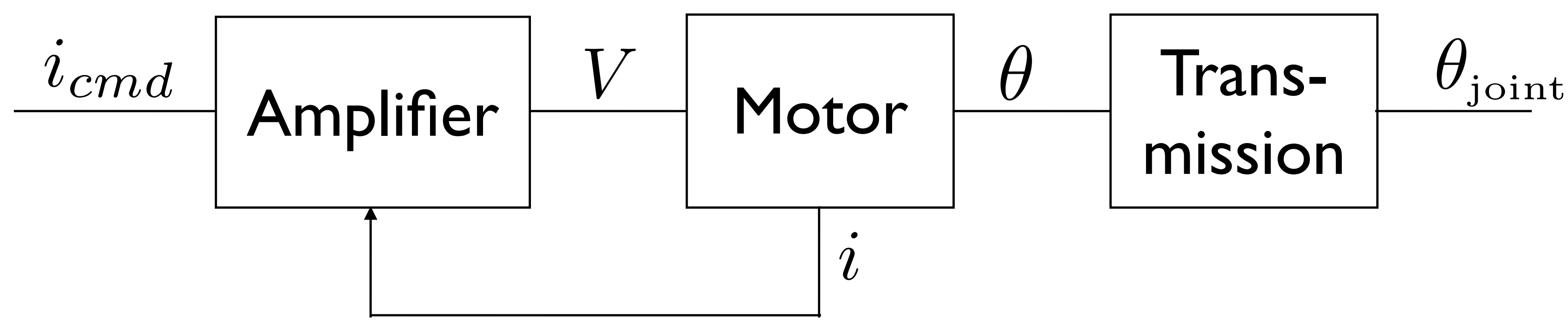
Sensors

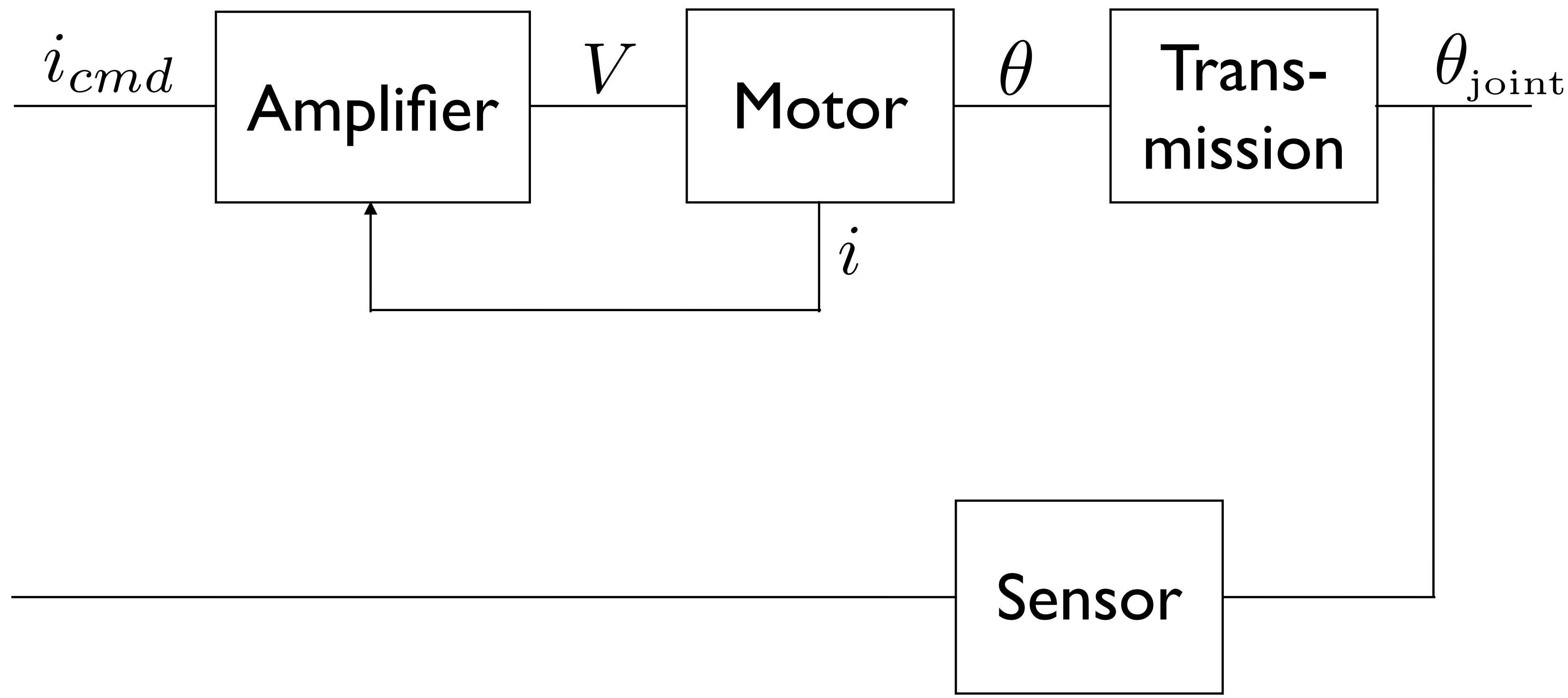
Kinesthetic	Encoders
Tactile	Load Cells
Vision	Vision
Vestibular	Accelerometers

Controller

Brain	Computer
Spinal Cord Reflex	Local feedback


i_{cmd}
Amplifier
V
Motor
θ
i


i_{cmd}
Amplifier
V
Motor
θ
Trans-
mission
θ_{joint}
i



i_{cmd}
Amplifier
V
Motor
θ
Trans-
mission
θ_{joint}
i
Sensor

A Biological Inspiration

Mechanical Structure

Bones	Frame / Links
Joints	Joints

Actuators

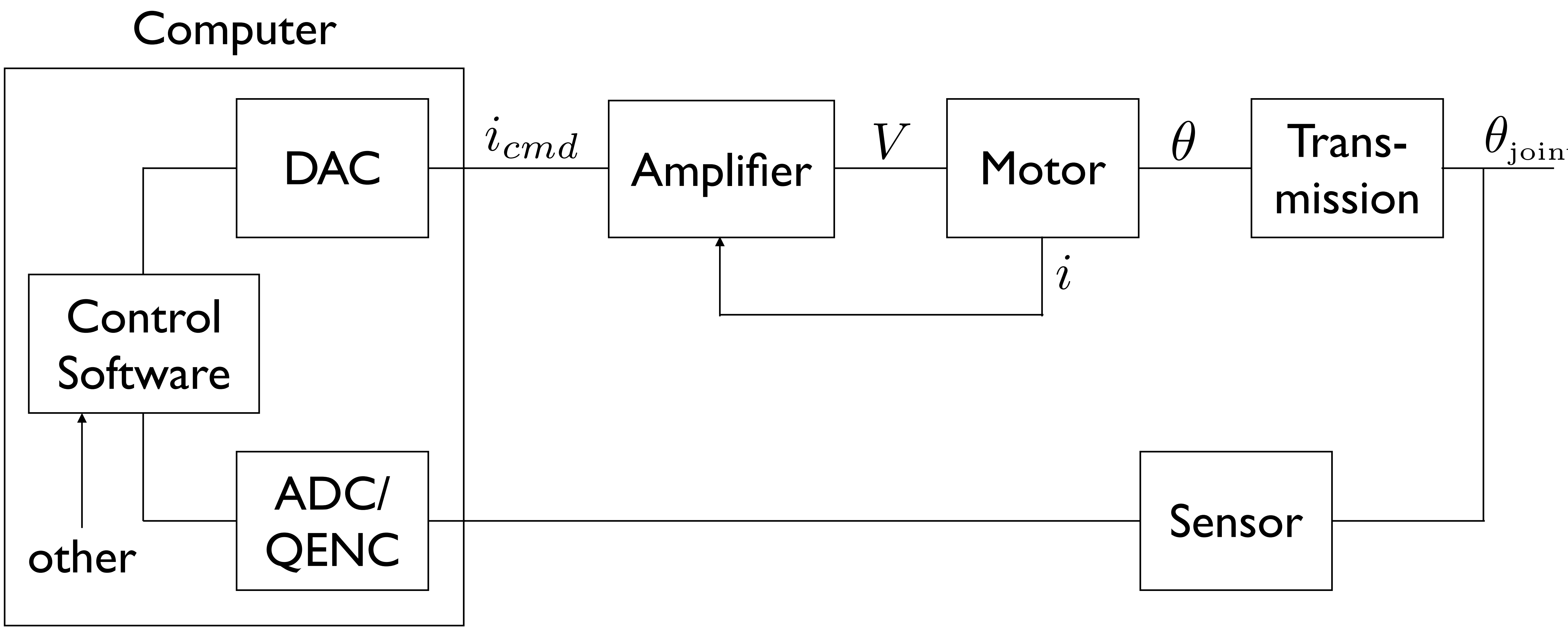
Muscles	Electric Motors
	Hydraulics
	Pneumatics
	SMA, etc.

Sensors

Kinesthetic	Encoders
Tactile	Load Cells
Vision	Vision
Vestibular	Accelerometers

Controller

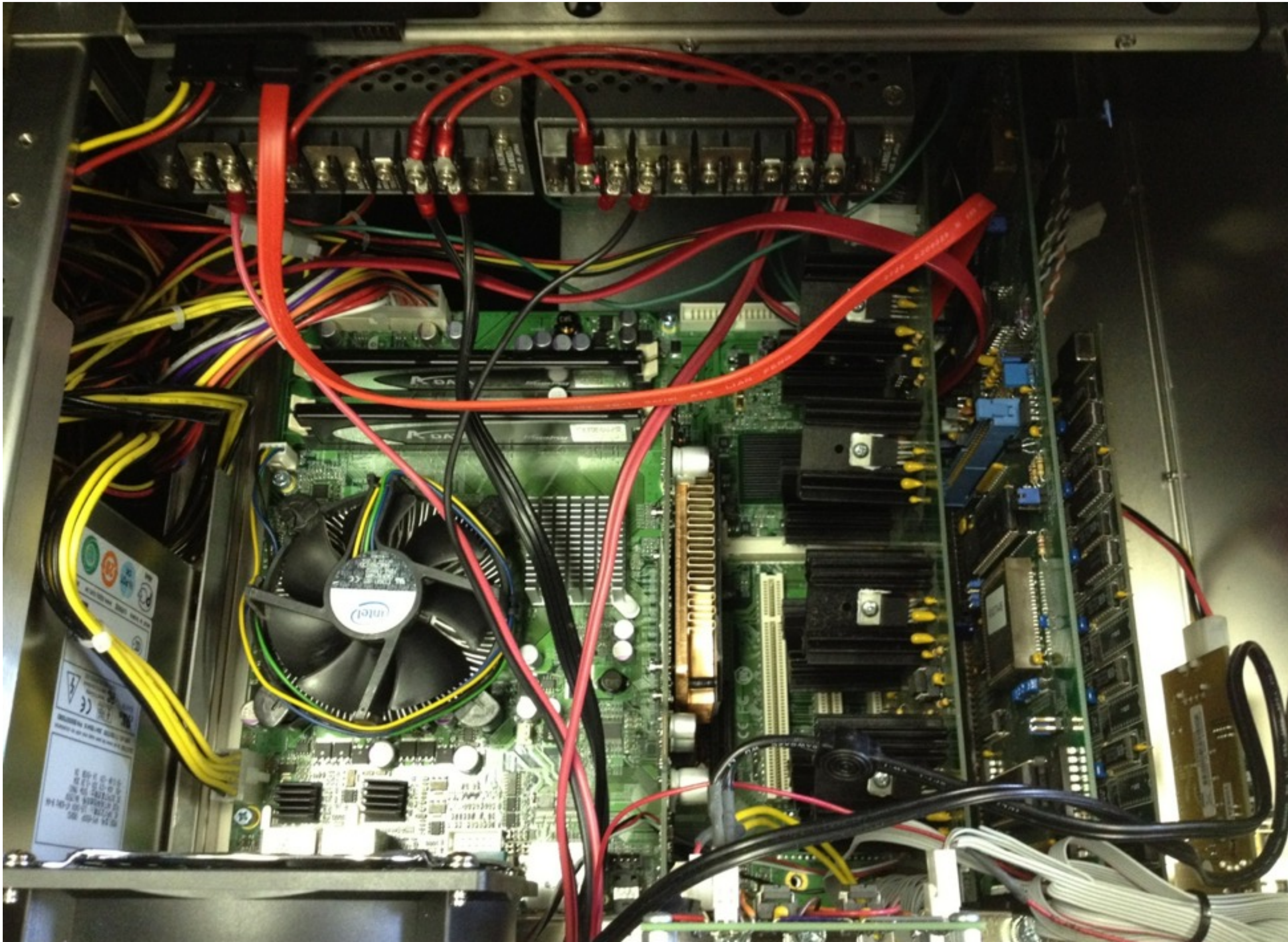
Brain	Computer
Spinal Cord Reflex	Local feedback

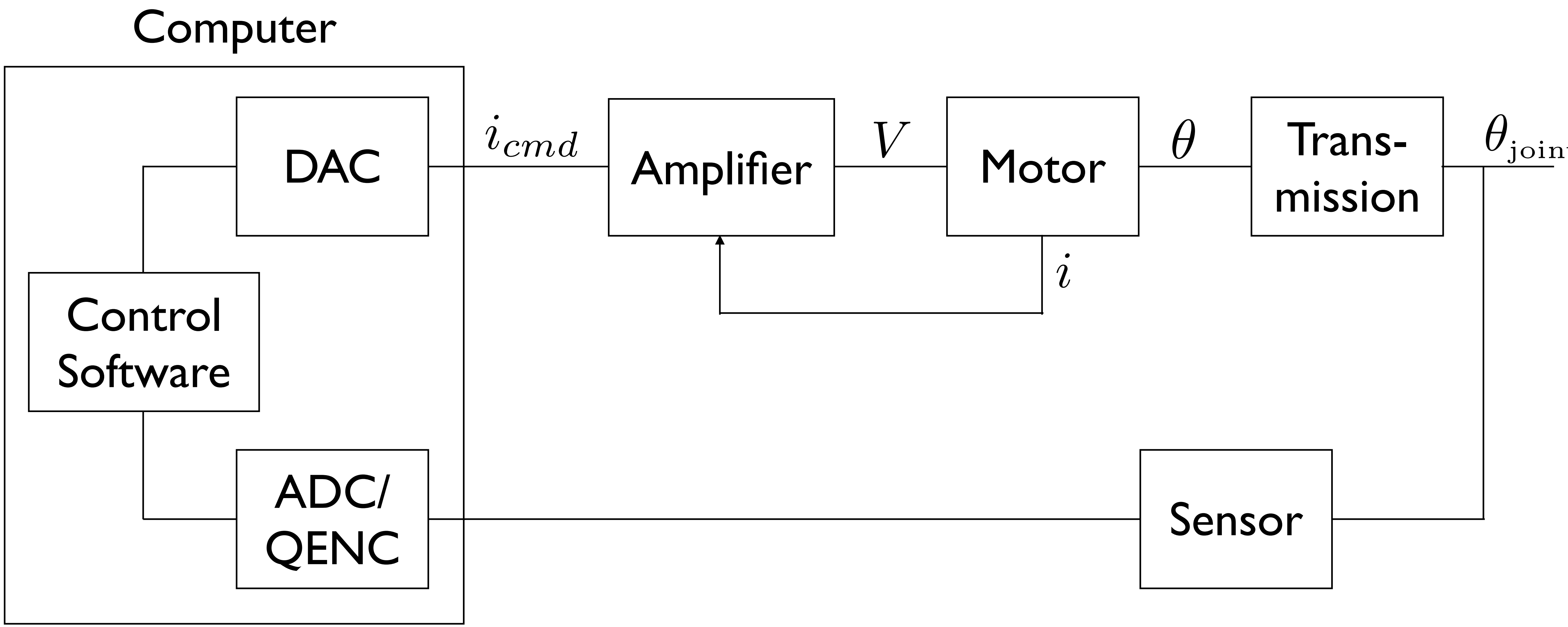

Computer
DAC
Control Software
ADC/ QENC
other
i_{cmd}
Amplifier
V
Motor
i
θ
Trans-mission
θ_{joint}
Sensor

Computer

- Runs a servo loop at a fixed rate, often 1000 Hz
- Electrically connects to your hardware through an ISA card, PCI card, Firewire cable, USB cable, or other.
- At each iteration, samples all of the sensors, computes the location of the robot, determines the forces and torques that should be exerted in response, and sends appropriate current commands to all of the system's actuators.

inside the Puma260 controller





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

Adequate for applications that don't involve very fast motions, especially if the transmission has a large gear reduction, which helps decouple the links from one another.

This is what we do on the PUMA. :)

Common Controllers

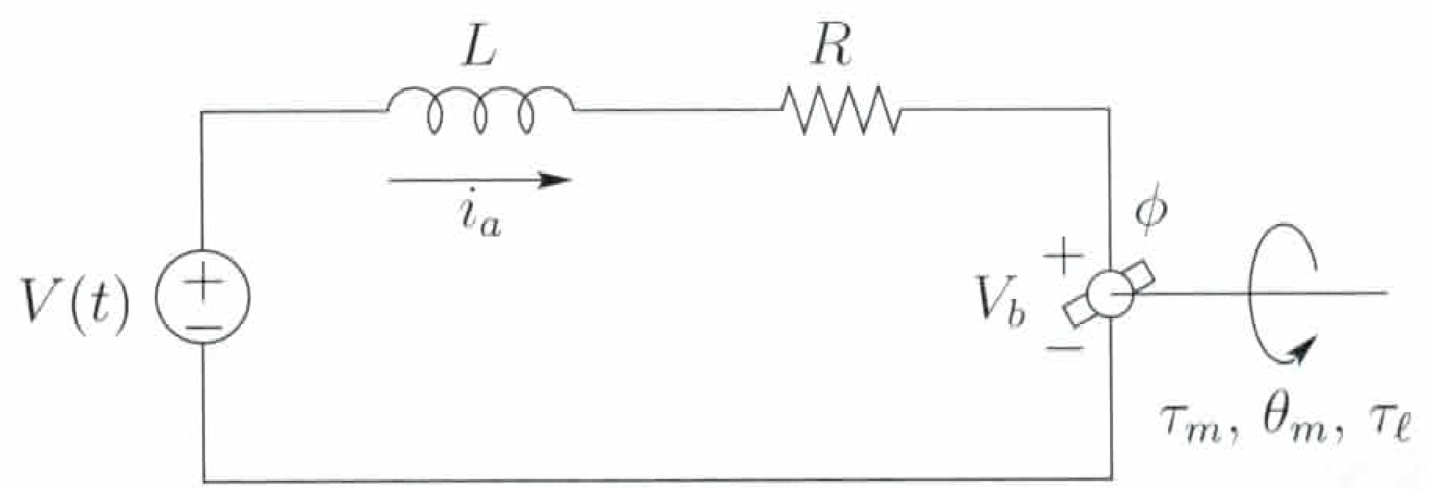
$$\tau_{\text{joint}} = K_P(\theta_{\text{desired}} - \theta_{\text{measured}})$$

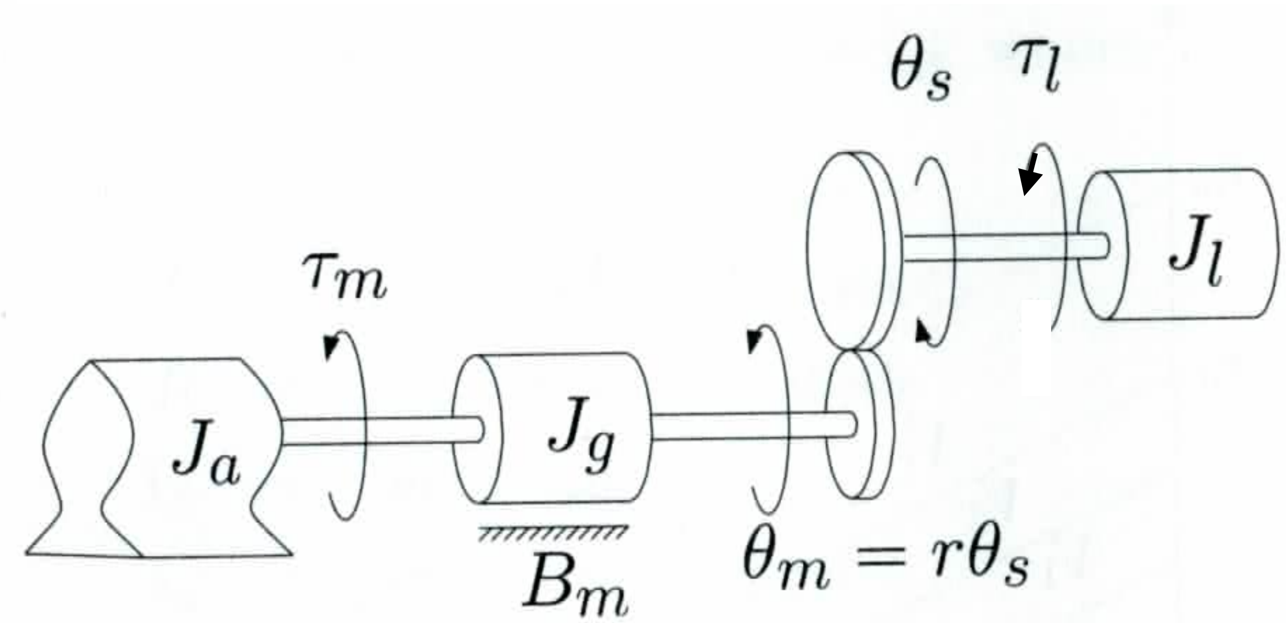
$$\tau_{\text{joint}} = K_P(\theta_{\text{desired}} - \theta_{\text{measured}}) - K_D\,\dot{\theta}_{\text{measured}}$$

$$\tau_{\text{joint}} = K_P(\theta_{\text{desired}} - \theta_{\text{measured}}) + K_D(\dot{\theta}_{\text{desired}} - \dot{\theta}_{\text{measured}})$$

$$\tau_{\text{joint}} = K_P(\theta_{\text{desired}} - \theta_{\text{measured}}) - K_D\,\dot{\theta}_{\text{measured}} + K_G \cdot fn(\theta_{\text{meas}})$$

Joint Dynamics (SHV 6.2)





Linear Model of Electrical and Mechanical Dynamics

$$\sum \tau \text{ on motor} = \tau_m - \tau_l/r - B_m\dot{\theta}_m = (J_a + J_g)\ddot{\theta}_m$$

$$L\dot{i}_a + Ri_a = V - V_b$$

$$V_b = k_v\dot{\theta}_m$$

A Biological Inspiration

Mechanical Structure

Bones	Frame / Links
Joints	Joints

Actuators

Muscles	Electric Motors
	Hydraulics
	Pneumatics
	SMA, etc.

Sensors

Kinesthetic	Encoders
Tactile	Load Cells
Vision	Vision
Vestibular	Accelerometers

Controller

Brain	Computer
Spinal Cord Reflex	Local feedback


PHANToM

Homework 5: Input/Output Calculations for a Real Robot

MEAM 520, University of Pennsylvania
Katherine J. Kuchenbecker, Ph.D.

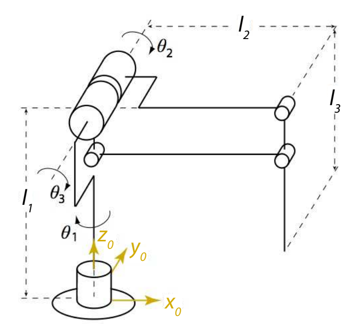
November 13, 2012

This assignment is due on **Tuesday, November 20**, by 5:00 p.m. sharp. You should aim to turn the paper part in during class that day. If you don't finish until later in the day, you can turn it in to Professor Kuchenbecker's office, Towne 224. Late submissions will be accepted until 5:00 p.m. on Wednesday, November 21, but they will be penalized by 25%. After that deadline, no further assignments may be submitted.

You may talk with other students about this assignment, ask the teaching team questions, use a calculator and other tools, and consult outside sources such as the Internet. To help you actually learn the material, what you write down should be your own work, not copied from a peer or a solution manual.

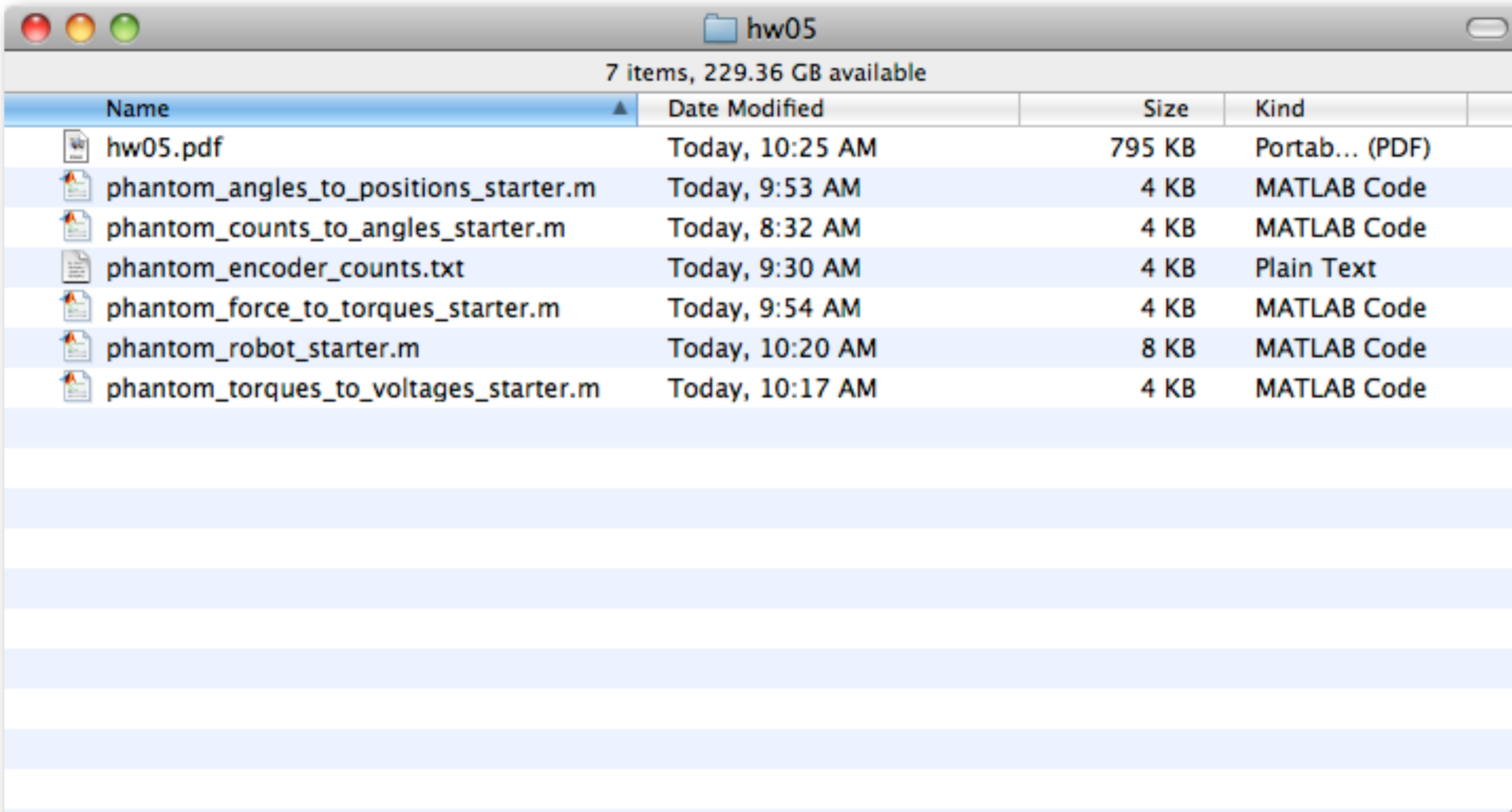
SensAble Phantom Premium 1.0 *(60 points)*

This entire assignment is focused on a particular robot – the SensAble Phantom Premium 1.0. As shown in the photo below left, the Phantom is an impedance-type haptic interface with three actuated rotational joints. Designed to be lightweight, stiff, smooth, and easily backdrivable, this type of haptic interface enables a human user to interact with a virtual environment or control the movement of a remote robot through the movement of their fingertip while simultaneously feeling force feedback.



A thimble is attached to the tip of the robot via a passive non-encoded three-axis gimbal to allow the user to move the robot around while freely rotating their fingertip. As shown in the diagram above right, the Phantom haptic device looks similar to the standard RRR articulated manipulator base, but it uses a unique four-bar mechanism to co-locate the shoulder and elbow joints while also keeping the upper arm and forearm in the plane that intersects the axis of the waist joint.

Each of the four questions below includes both a written explanation and the programming of a specific Matlab function. For the paper parts, write in pencil, show your work clearly, $\boxed{\text{box your answers}}$, and staple your pages together. For the programming, download the starter code from this assignment's page on the class wiki, change all function and file names to include your PennKey, comment your code, and follow the instructions at the end of this document to submit all of your Matlab files.


hw05

7 items, 229.36 GB available

Name	Date Modified	Size	Kind
hw05.pdf	Today, 10:25 AM	795 KB	Portab... (PDF)
phantom_angles_to_positions_starter.m	Today, 9:53 AM	4 KB	MATLAB Code
phantom_counts_to_angles_starter.m	Today, 8:32 AM	4 KB	MATLAB Code
phantom_encoder_counts.txt	Today, 9:30 AM	4 KB	Plain Text
phantom_force_to_torques_starter.m	Today, 9:54 AM	4 KB	MATLAB Code
phantom_robot_starter.m	Today, 10:20 AM	8 KB	MATLAB Code
phantom_torques_to_voltages_starter.m	Today, 10:17 AM	4 KB	MATLAB Code

My definition of a haptic interface

Senses a physical quantity from the user,
such as motion or force

Physically acts on the user via a variable actuator

Connects sensing to acting with fast processing

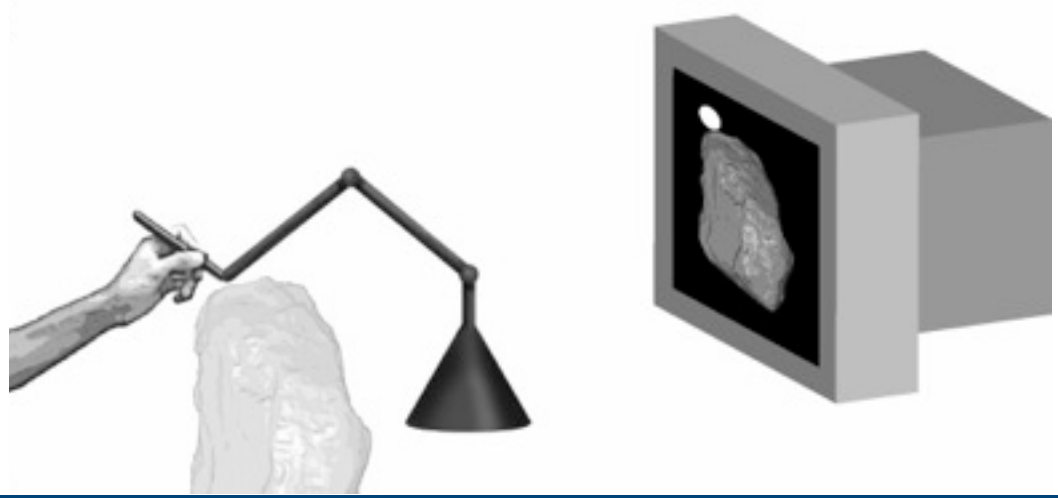
Environment

Real

Assistive Interaction: augments human sensing and/or motion capabilities in real physical environments

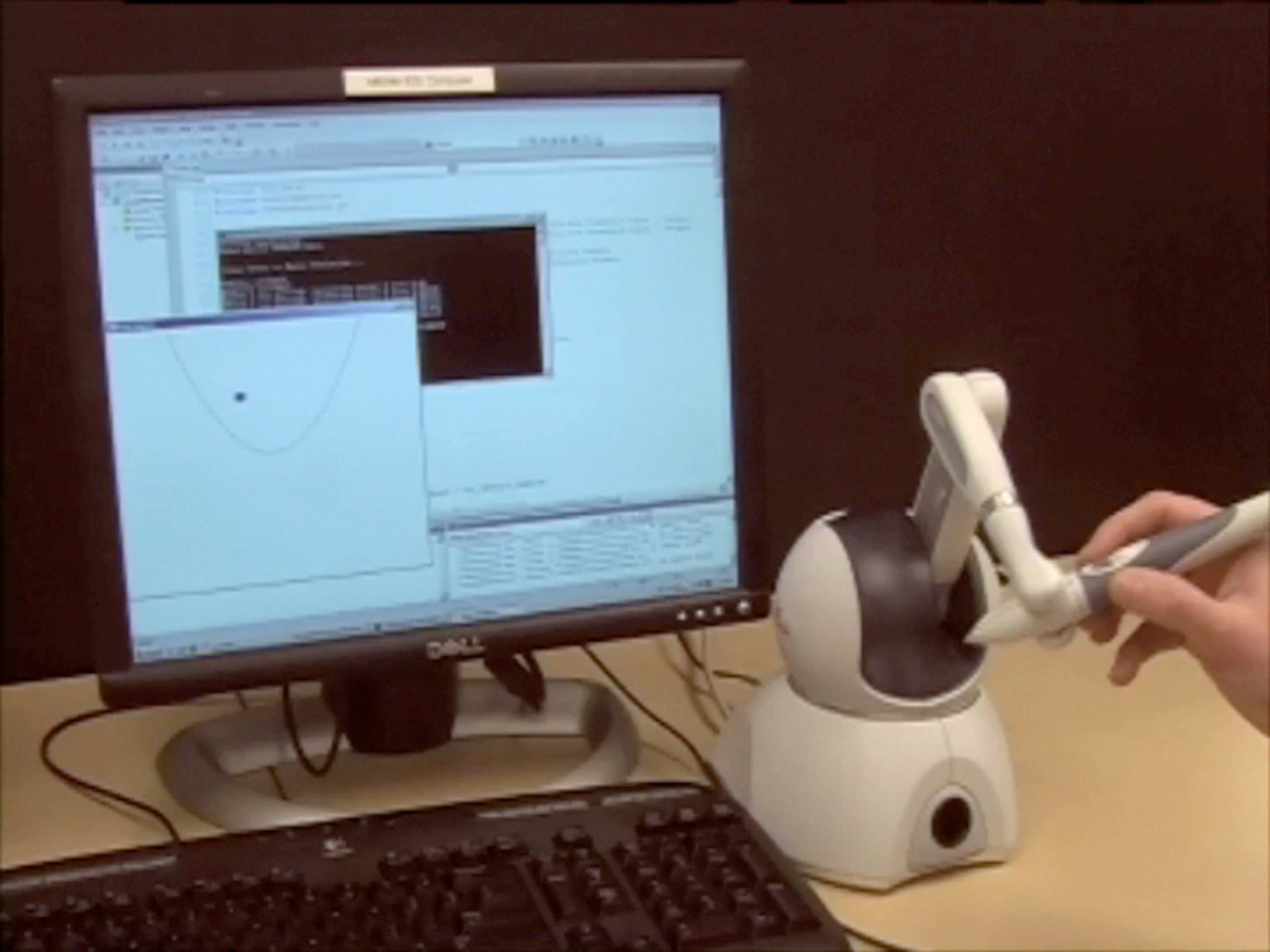
Remote

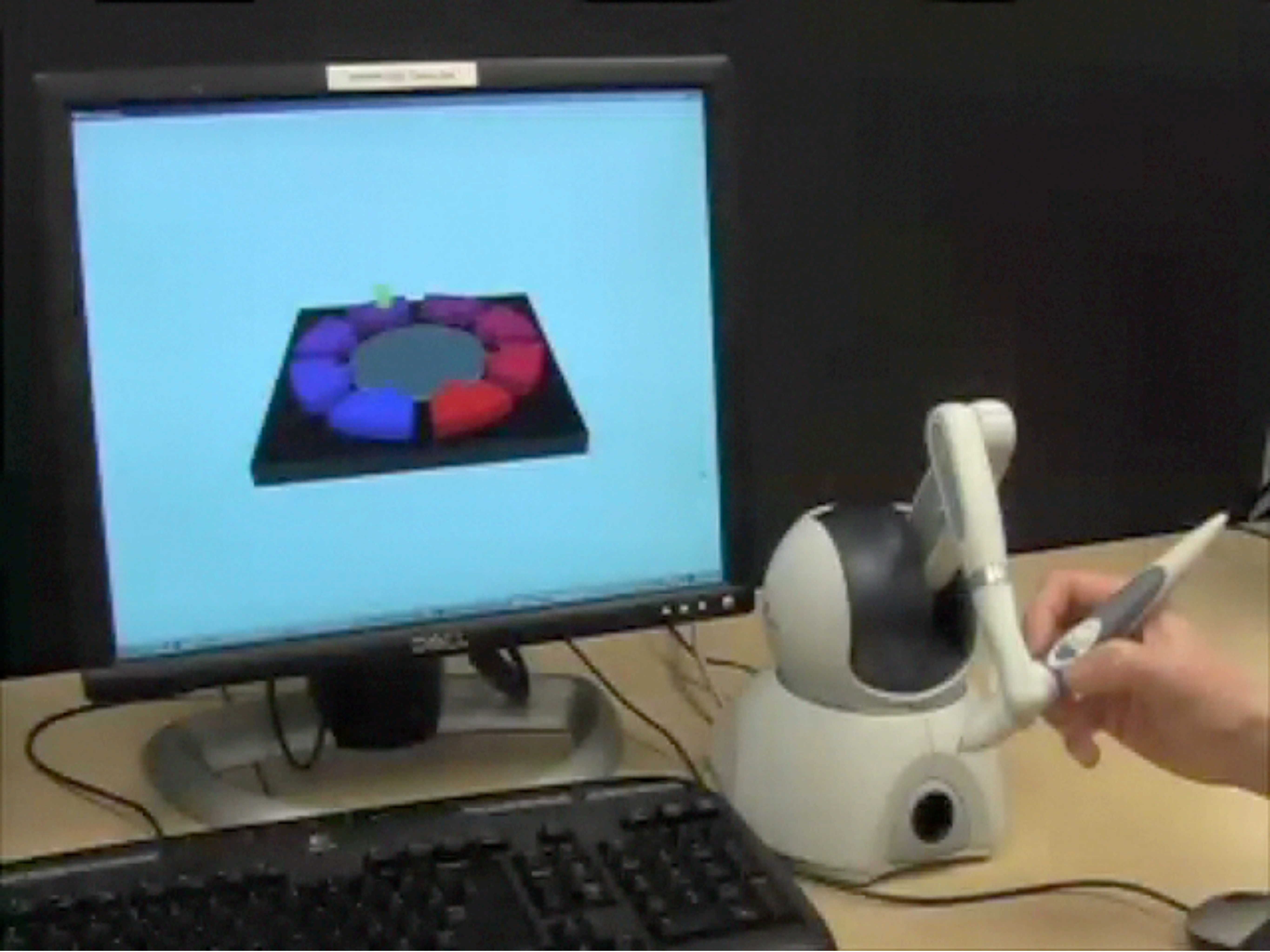
Teleoperation: extends the reach of the human hand to remote, hazardous, unreachable environments



Virtual

Simulation: enables humans to touch geometric and dynamic computer-based data and models


DELL

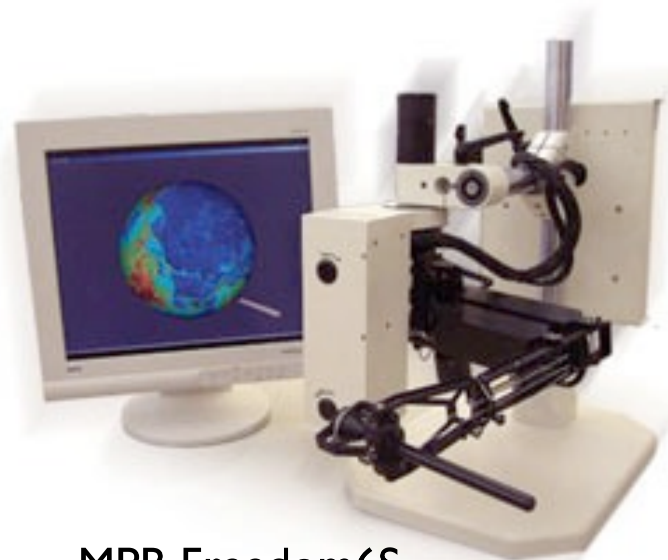



Without Acceleration Matching
With Acceleration Matching
world HAPTICS 2009

Commercial Kinesthetic Haptic Interfaces

SensAble Phantom Premiums

Force Dimension Omega

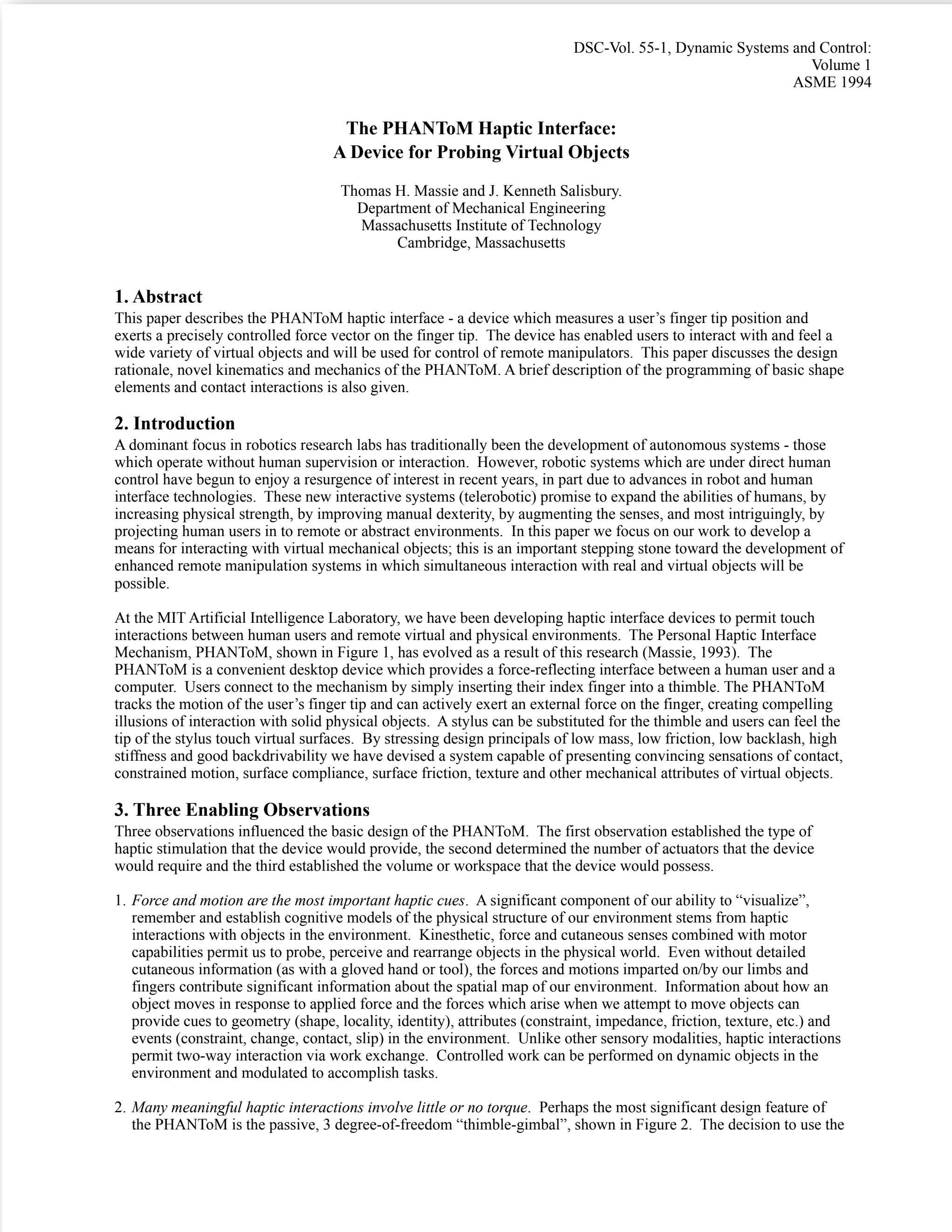


MPB Freedom6S

Immersion Impulse Engine

SensAble Omni

Novint Falcon



DSC-Vol. 55-1, Dynamic Systems and Control:
Volume 1
ASME 1994

The PHANToM Haptic Interface:
A Device for Probing Virtual Objects

Thomas H. Massie and J. Kenneth Salisbury.
Department of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, Massachusetts

1. Abstract
This paper describes the PHANToM haptic interface - a device which measures a user's finger tip position and exerts a precisely controlled force vector on the finger tip. The device has enabled users to interact with and feel a wide variety of virtual objects and will be used for control of remote manipulators. This paper discusses the design rationale, novel kinematics and mechanics of the PHANToM. A brief description of the programming of basic shape elements and contact interactions is also given.

2. Introduction
A dominant focus in robotics research labs has traditionally been the development of autonomous systems - those which operate without human supervision or interaction. However, robotic systems which are under direct human control have begun to enjoy a resurgence of interest in recent years, in part due to advances in robot and human interface technologies. These new interactive systems (telerobotic) promise to expand the abilities of humans, by increasing physical strength, by improving manual dexterity, by augmenting the senses, and most intriguingly, by projecting human users in to remote or abstract environments. In this paper we focus on our work to develop a means for interacting with virtual mechanical objects; this is an important stepping stone toward the development of enhanced remote manipulation systems in which simultaneous interaction with real and virtual objects will be possible.

At the MIT Artificial Intelligence Laboratory, we have been developing haptic interface devices to permit touch interactions between human users and remote virtual and physical environments. The Personal Haptic Interface Mechanism, PHANToM, shown in Figure 1, has evolved as a result of this research (Massie, 1993). The PHANToM is a convenient desktop device which provides a force-reflecting interface between a human user and a computer. Users connect to the mechanism by simply inserting their index finger into a thimble. The PHANToM tracks the motion of the user's finger tip and can actively exert an external force on the finger, creating compelling illusions of interaction with solid physical objects. A stylus can be substituted for the thimble and users can feel the tip of the stylus touch virtual surfaces. By stressing design principals of low mass, low friction, low backlash, high stiffness and good backdrivability we have devised a system capable of presenting convincing sensations of contact, constrained motion, surface compliance, surface friction, texture and other mechanical attributes of virtual objects.

3. Three Enabling Observations
Three observations influenced the basic design of the PHANToM. The first observation established the type of haptic stimulation that the device would provide, the second determined the number of actuators that the device would require and the third established the volume or workspace that the device would possess.

1. *Force and motion are the most important haptic cues*. A significant component of our ability to "visualize", remember and establish cognitive models of the physical structure of our environment stems from haptic interactions with objects in the environment. Kinesthetic, force and cutaneous senses combined with motor capabilities permit us to probe, perceive and rearrange objects in the physical world. Even without detailed cutaneous information (as with a gloved hand or tool), the forces and motions imparted on/by our limbs and fingers contribute significant information about the spatial map of our environment. Information about how an object moves in response to applied force and the forces which arise when we attempt to move objects can provide cues to geometry (shape, locality, identity), attributes (constraint, impedance, friction, texture, etc.) and events (constraint, change, contact, slip) in the environment. Unlike other sensory modalities, haptic interactions permit two-way interaction via work exchange. Controlled work can be performed on dynamic objects in the environment and modulated to accomplish tasks.

2. *Many meaningful haptic interactions involve little or no torque*. Perhaps the most significant design feature of the PHANToM is the passive, 3 degree-of-freedom "thimble-gimbal", shown in Figure 2. The decision to use the

T. H. Massie and J. K. Salisbury. *The PHANToM haptic interface: A device for probing virtual objects*. In Proceedings of the Third International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, ASME Dynamic Systems and Control, Volume 55(1), pages 295-300. American Society of Mechanical Engineers, 1994.

This is the most highly cited paper in the field of haptics. It describes the design objectives of the Phantom haptic interface, which was developed in Ken Salisbury's lab at MIT. The paper also anecdotally describes the reactions of users of the device. The described device is very similar to the SensAble Phantom Premium 1.0 you are using in this class.

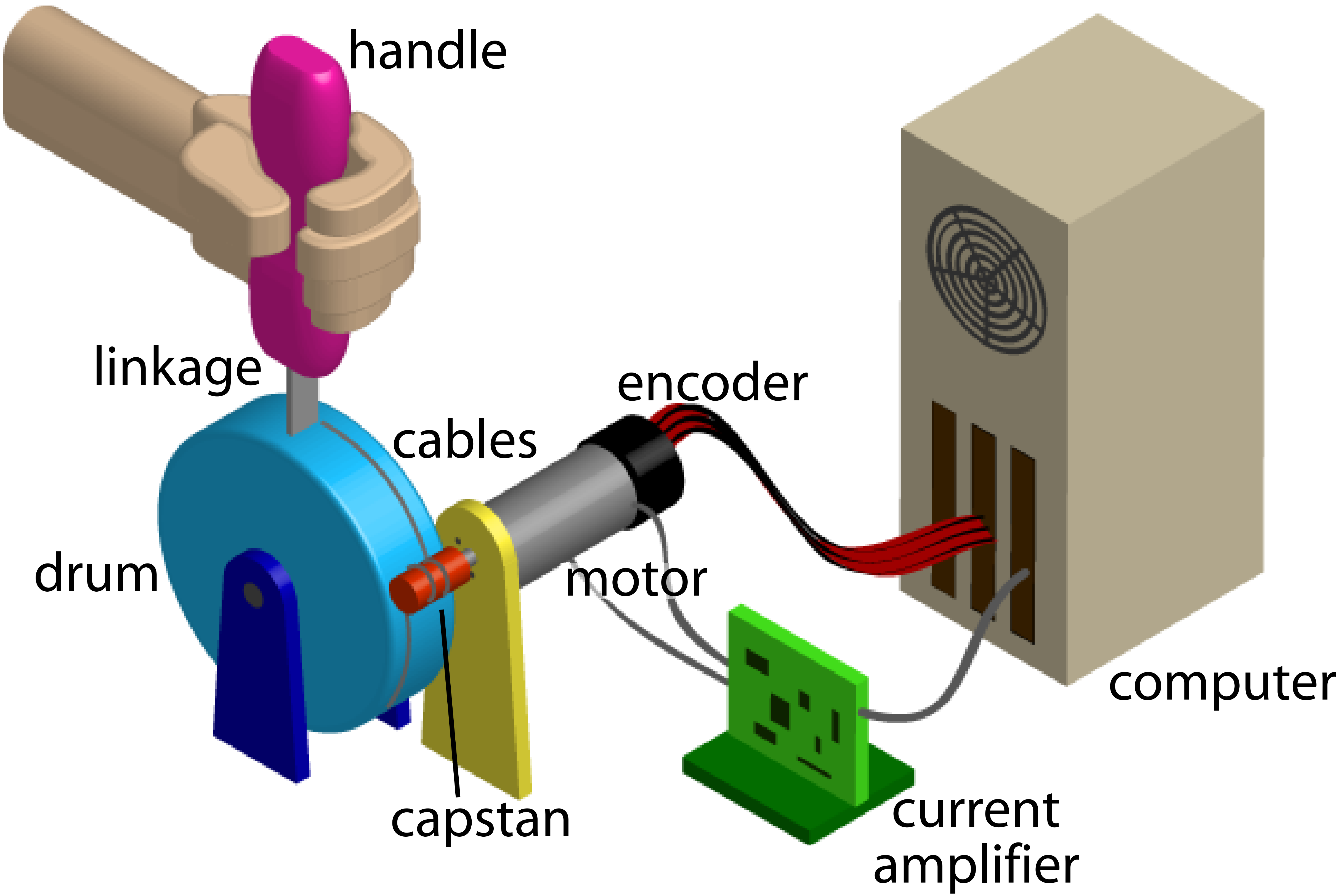
Three Necessary Criteria

1. Free space must feel free.

2. Solid virtual objects must feel stiff.

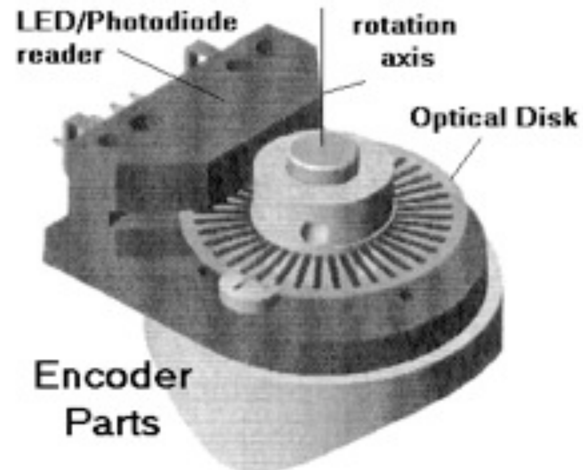
3. Virtual constraints must not be easily saturated.

We want a design that can achieve all of these objectives simultaneously.


handle
linkage
encoder
cables
drum
motor
capstan
computer
current
amplifier

Common Haptic Interface Components

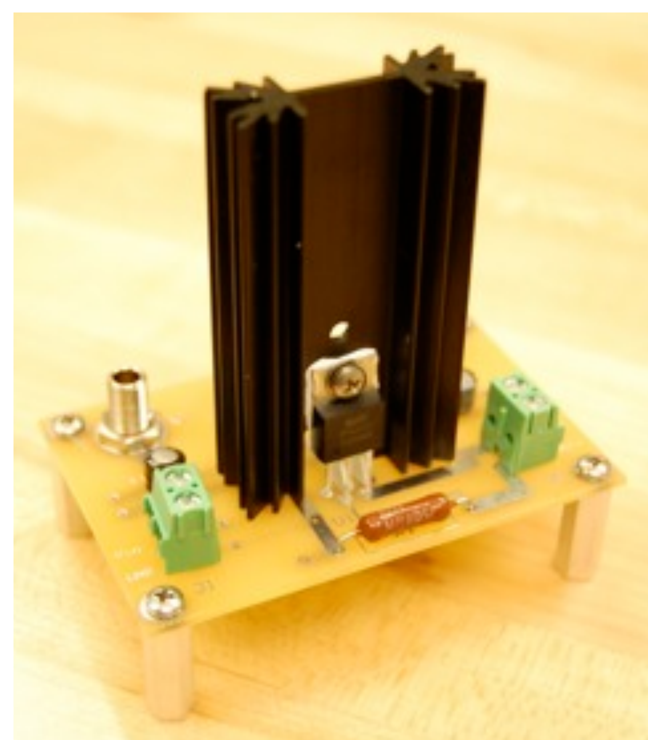
Capstan & Cable Drive
Stiff Metal Linkages



Incremental
Optical Encoder

Computer Interface Card

Brushed Permanent Magnet
Direct Current Motor



Current Amplifier

Elements of Haptic Interfaces

Katherine J. Kuchenbecker
Department of Mechanical Engineering and Applied Mechanics
University of Pennsylvania
kuchenbe@seas.upenn.edu

Course Notes for MEAM 625, University of Pennsylvania
Adapted from Section 3.1 of Professor Kuchenbecker's Ph.D. thesis [3].

A haptic interface plays the important role of connecting the user to the controller during interactions with remote and virtual objects. Such systems incorporate mechanical, electrical, and computational elements, which all interact to create the touch-based sensations experienced by the user. This document is concerned specifically with actuated impedance-type interfaces, which currently dominate the field due to their excellent free-space characteristics and their widespread use in a variety of applications. During an interaction, the controller of an impedance-type device must measure the user's hand motion and apply an appropriate force in response. Impedance-type haptic interfaces vary in design, but they usually include a series of electrical and mechanical elements between the handle and the computer, as described below.

Overview

Haptic interfaces typically provide two or three degrees of freedom in position, sensing the user's motion and applying feedback forces within this workspace. Many devices also permit changes in the orientation of the end effector; these rotational degrees of freedom can be unsensed, sensed but not actuated, or sensed and actuated. The remainder of this discussion will focus on translation rather than orientation, though the described design features can be applied to either.

Figure 1 illustrates the chain of elements typically present in each axis of a haptic interface. For clarity, the illustration depicts a device with a single degree of freedom, but typical systems combine several degrees of freedom in parallel or series to allow unrestricted translation and/or orientation. Although differences exist, individual position axes of most mechanisms can be represented by such an arrangement. The terms "haptic interface" and "master" are often used interchangeably to represent all electrical and mechanical elements depicted in Figure 1, extending from the amplifier and encoder to the handle.

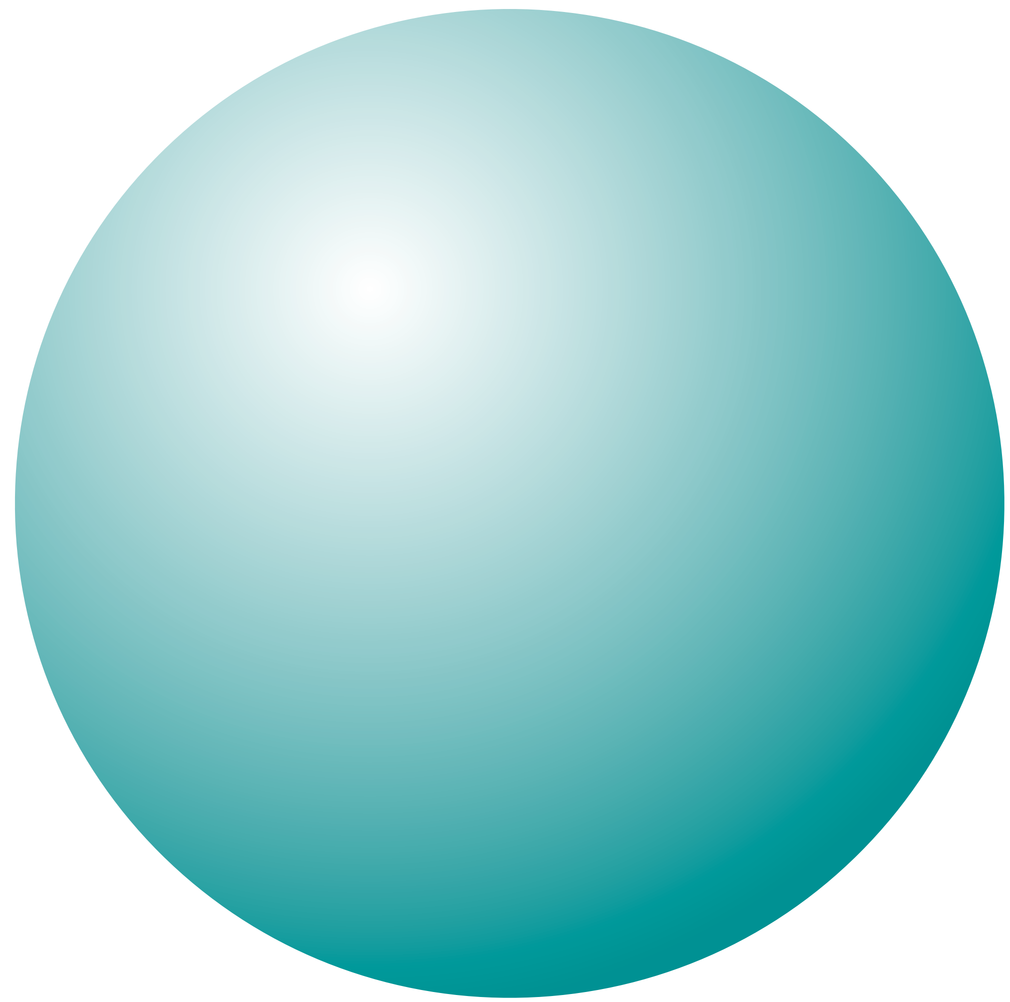
Your General Haptics Programming Task

Specify:
Force vector

As a function of:
Everything that is known, including
position vector, velocity vector, time,
model geometry, model properties,
keyboard inputs, mouse inputs, button inputs,
and randomness

In order to:
Fool the user into thinking they are touching something

How would you make the user believe they are touching a suspended sphere?



High-Level Approach For Rendering Shapes

Used by Massie and Salisbury in 1994

Now known as the *Vector-Field Approach*

- Force is a function of only the present position

$$\vec{F}_h = f(\vec{x}_h)$$

- Divide space into volumes that represent objects
- The rest of the volume is free space.
- Object contact is rendered by keeping track of a geometric proxy. It stays on the surface of the object, and we use a virtual spring to pull the user toward its location, always perpendicular to the surface.

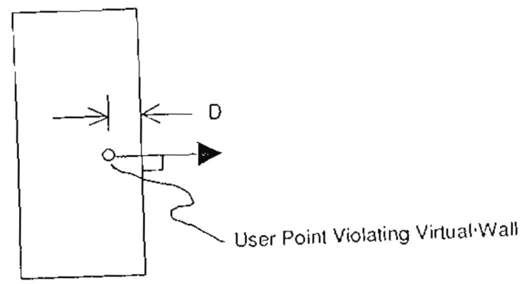


Figure 4: Simulation of virtual planes. Force, $F = Kx$, is exerted normal to plane when the user pushes into virtual surface.

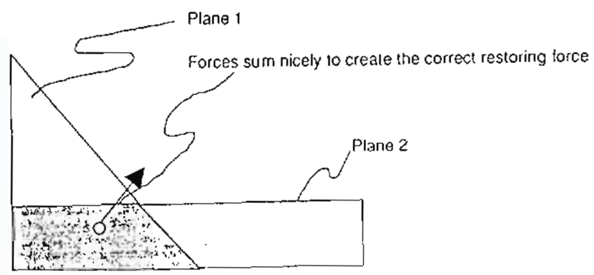


Figure 5: Forces at interior corner defined by 2 planes. Forces sum properly to create correct restoring force when planes meet at obtuse angle.

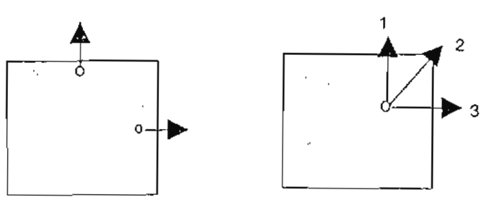


Figure 6: Virtual Cubes. Figure at left illustrates reasonable force vectors for cube with slightly compliant surface. For the point shown in figure at right, it is not clear which of forces 1, 2 and 3 should be exerted. This is path dependent.

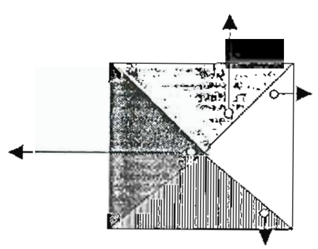


Figure 7: Solution to corner problem with cubes. Dividing cube into regions shown provides simple solution to path ambiguity problem. In 3-D, regions are pyramid in shape and permit stable behavior at edges and corners. If large forces are exerted at corners, probe point may move into the adjacent region and be pushed off object giving the sensation of "plucking" the corner.

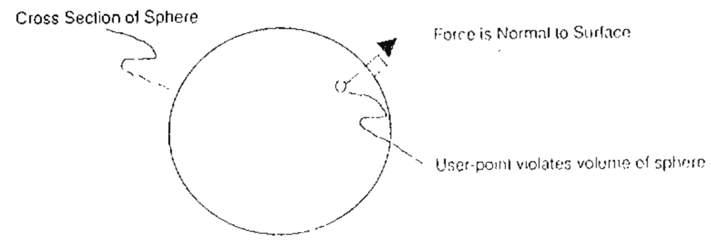


Figure 8: Forces generated by contact with sphere. As with planes, force is proportional to penetration depth and in direction normal to surface. Because force is always normal, spheres feel very slippery.

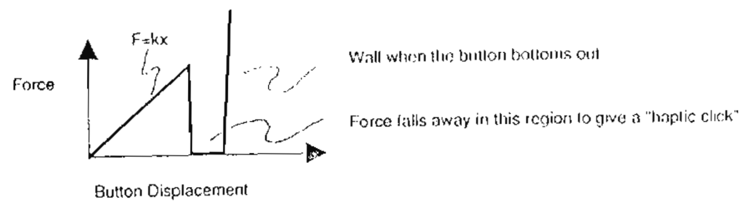
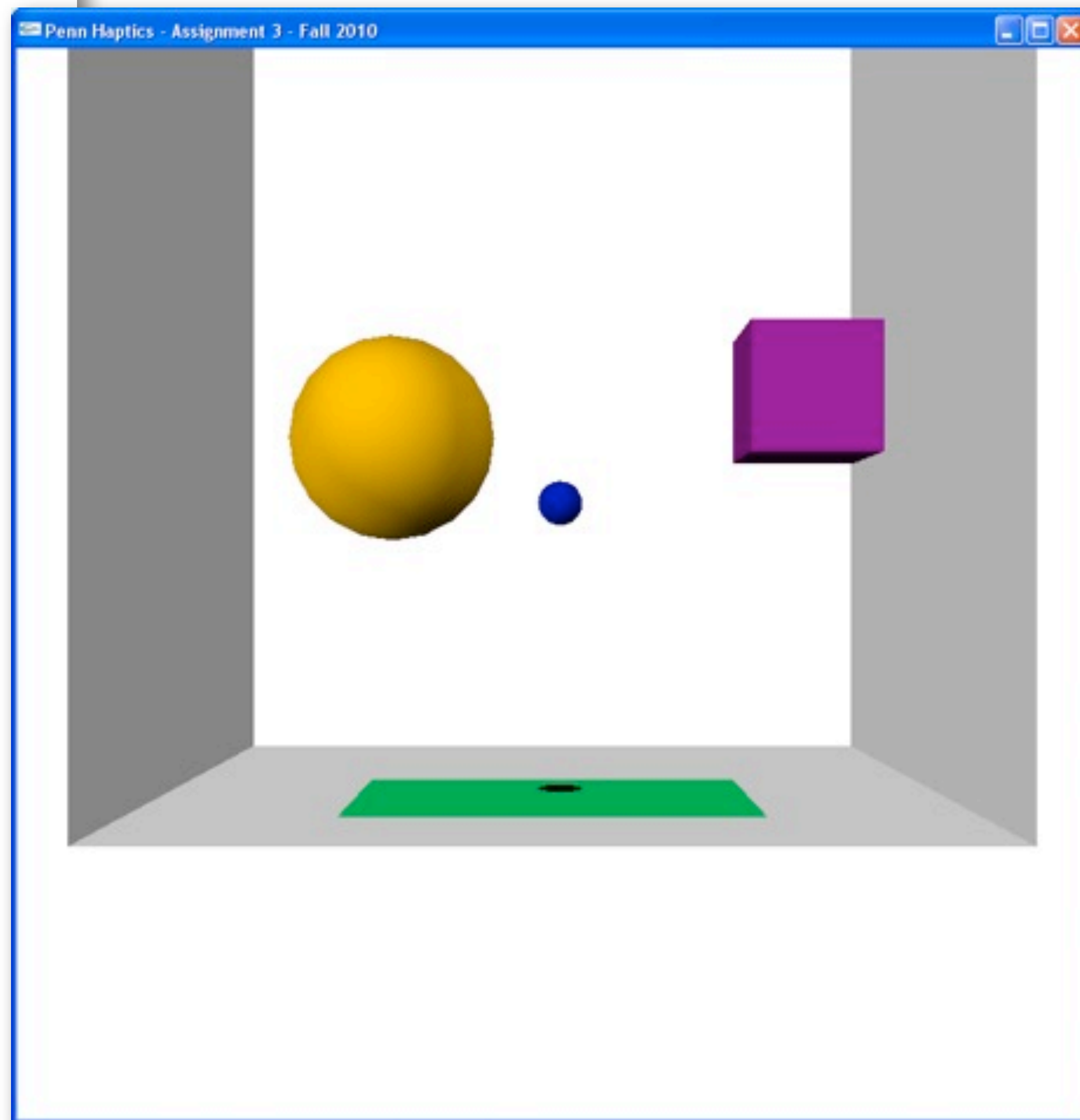


Figure 9: Virtual buttons. Nonlinear force-deflection curve for button demonstration showing "fall-away" force characteristic.

11 Appendix 1: PHANToM Specifications

Force resolution:	12 bit
Nominal spatial resolution:	400 dpi
Peak force:	10 Nt
Continuous force:	1.5 Nt
Backdrive friction:	0.1 Nt
Max/min force (Dyn. range):	100 : 1
Inertia at tip:	100 gm
Workspace:	8 x 17 x 25 cm
Max. object stiffness	35 Nt/cm



Typical Software Configuration

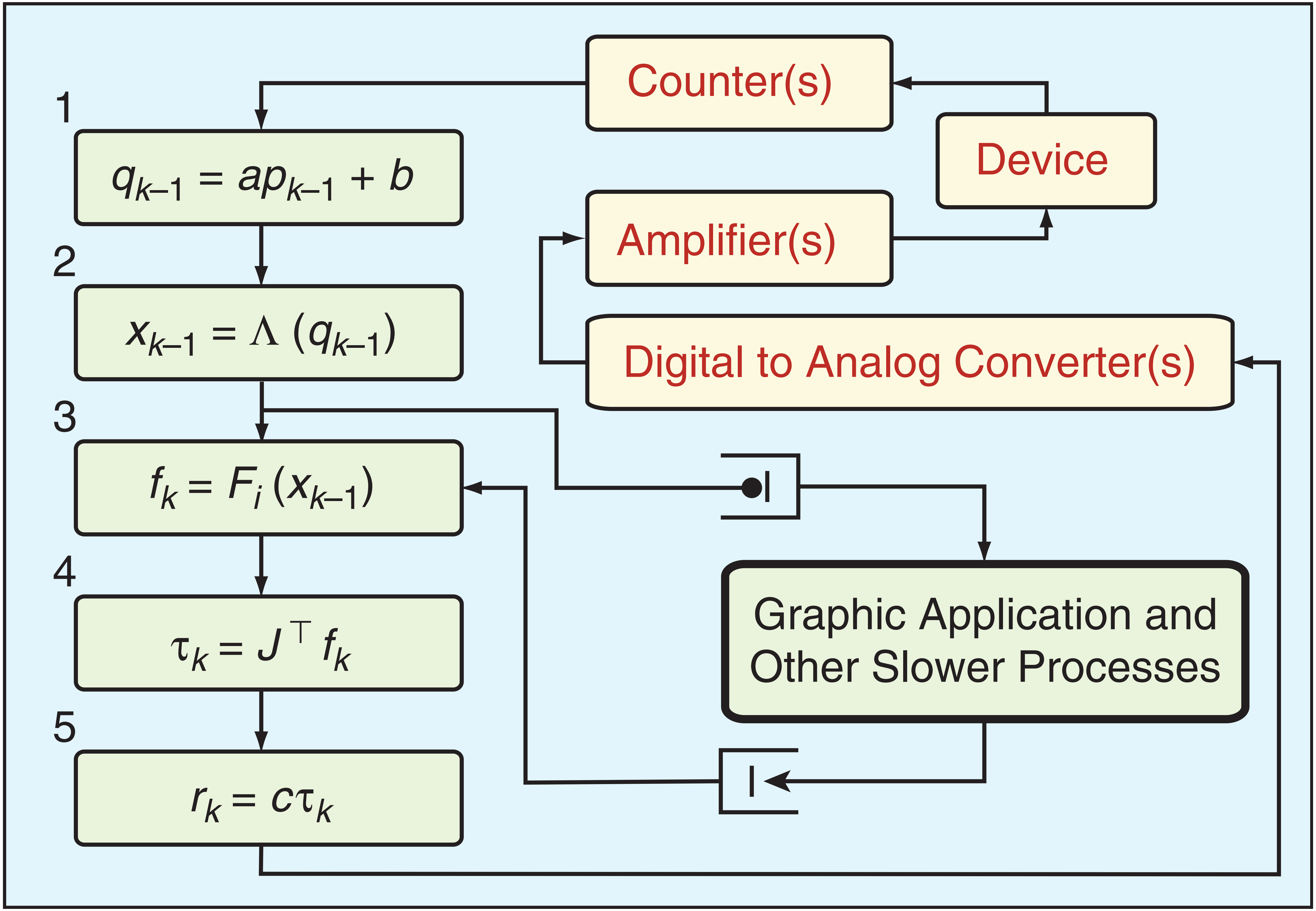
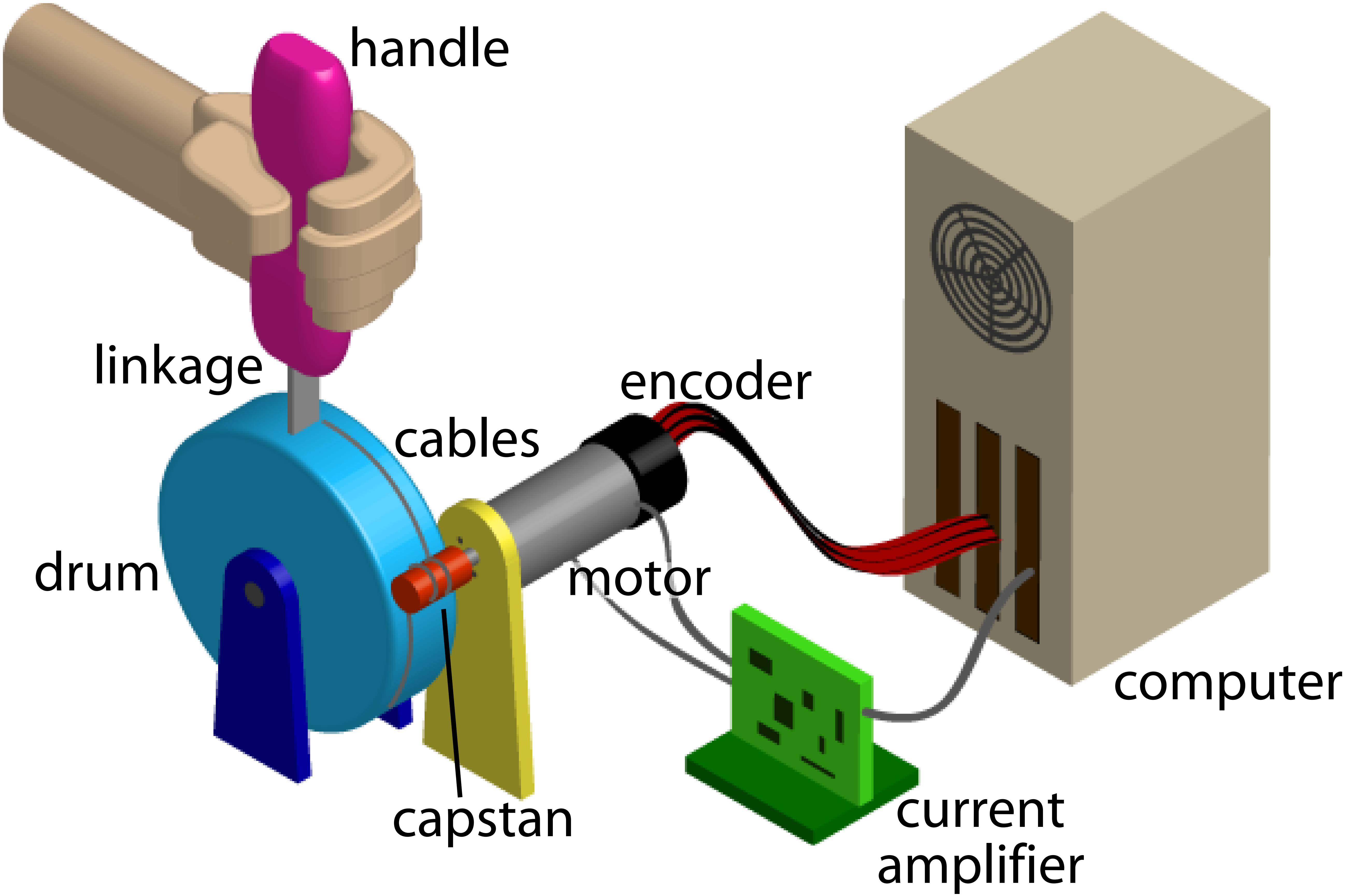


Illustration from V. Hayward and K. MacLean. "Do It Yourself Haptics: Part I."
IEEE Robotics and Automation Magazine, December, 2007, pp. 88-104.


handle
linkage
encoder
cables
drum
motor
capstan
computer
current
amplifier

Typical Software Configuration

Input: from sensor signals to counts

Counter or Serial or A2D
(Analog-to-Digital Converter)

Input Processing: from counts to joint values

$$\vec{q}_k = \vec{a} * \vec{Q}_k + \vec{b}$$

Forward Kinematics: from joint values to tip position

$$\vec{x}_k = \Lambda(\vec{q}_k)$$

Force Computation: from tip position to tip force

$$\vec{f}_k = F_i(\vec{x}_k)$$

Graphics, Remote Robot,
and Other Slower Processes

Torque Computation: from tip force to joint torques

$$\vec{\tau}_k = [J(\vec{q}_k)]^T \vec{f}_k$$

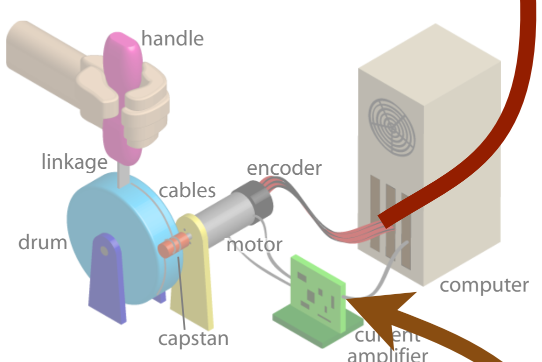
Output Processing: from joint torques to counts

$$\vec{r}_k = \vec{c} * \vec{\tau}_k + \vec{d}$$

Output: from counts to command signals

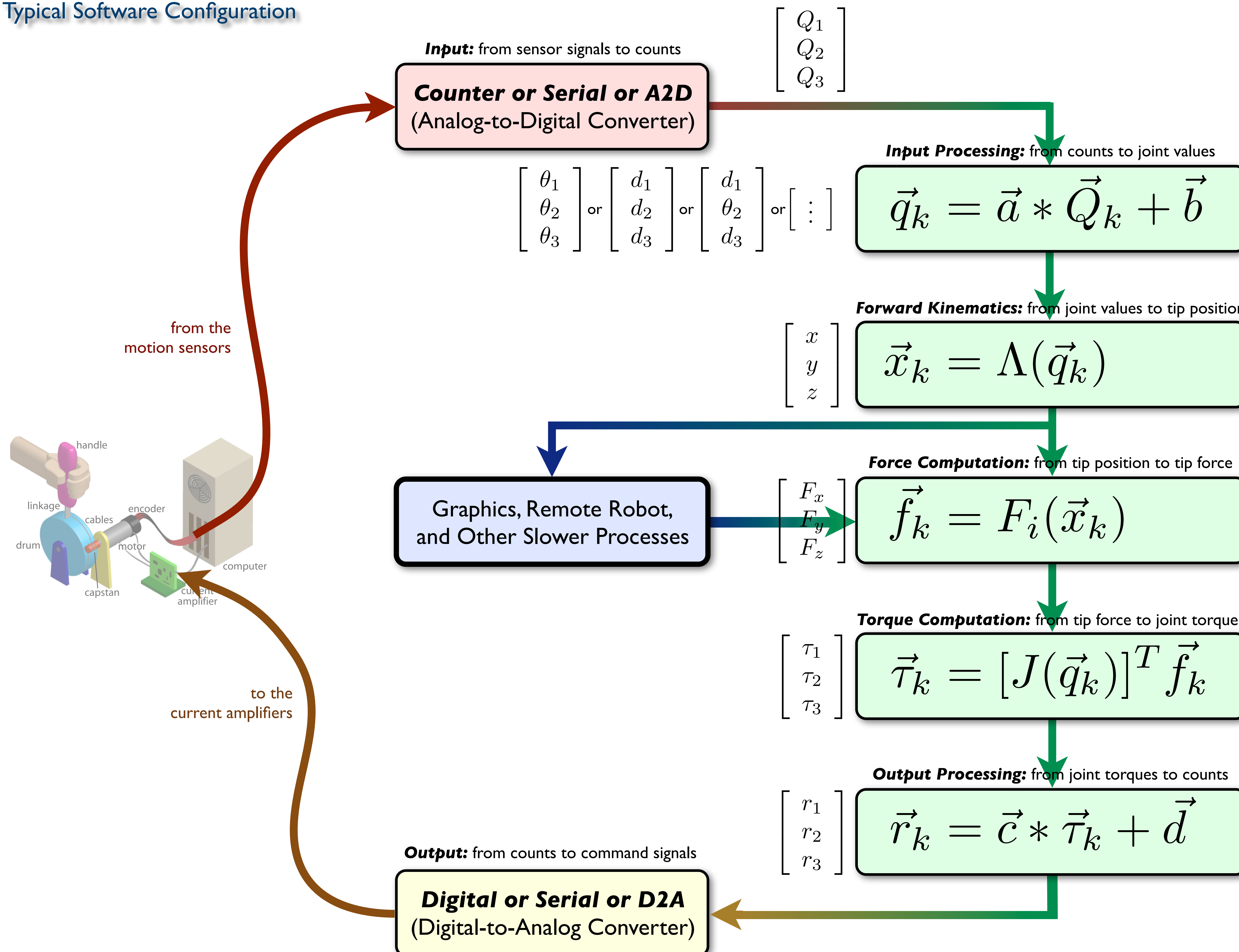
Digital or Serial or D2A
(Digital-to-Analog Converter)

from the
motion sensors



to the
current amplifiers

Typical Software Configuration



Input Processing Steps

Input Processing: from counts to joint values

$$\vec{q}_k = \vec{a} * \vec{Q}_k + \vec{b}$$

- Get counts $\boldsymbol{Q_j}$ from encoder counters, serial communications, or analog-to-digital conversions.
- Convert counts to sensor shaft angles θ_{sj} or sensor displacements $\boldsymbol{d_{sj}}$ using knowledge of the sensor's characteristics.
- Convert sensor angles to joint coordinates $\boldsymbol{q_j}$ (joint angles θ_j or joint displacements $\boldsymbol{d_j}$) using the gear ratio. In this process, use a negative sign to flip the joint angle direction if desired.

Input Processing Steps

Input Processing: from counts to joint values

$$\boxed{\vec{q}_k = \vec{a} * \vec{Q}_k + \vec{b}}$$

- Check your work along the way.
- How?
  - Units
  - Known configurations
  - Ranges
  - Record and graph
- Check *before* you use the movement information to output forces.

Typical Software Configuration

Input: from sensor signals to counts

Counter or Serial or A2D
(Analog-to-Digital Converter)

$$\begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix}$$

Input Processing: from counts to joint values

$$\vec{q}_k = \vec{a} * \vec{Q}_k + \vec{b}$$

$$\begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} \text{ or } \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} \text{ or } \begin{bmatrix} d_1 \\ \theta_2 \\ d_3 \end{bmatrix} \text{ or } \begin{bmatrix} \vdots \end{bmatrix}$$

Forward Kinematics: from joint values to tip position

$$\vec{x}_k = \Lambda(\vec{q}_k)$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Force Computation: from tip position to tip force

$$\vec{f}_k = F_i(\vec{x}_k)$$

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix}$$

Graphics, Remote Robot, and Other Slower Processes

Torque Computation: from tip force to joint torques

$$\vec{\tau}_k = [J(\vec{q}_k)]^T \vec{f}_k$$

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix}$$

Output Processing: from joint torques to counts

$$\vec{r}_k = \vec{c} * \vec{\tau}_k + \vec{d}$$

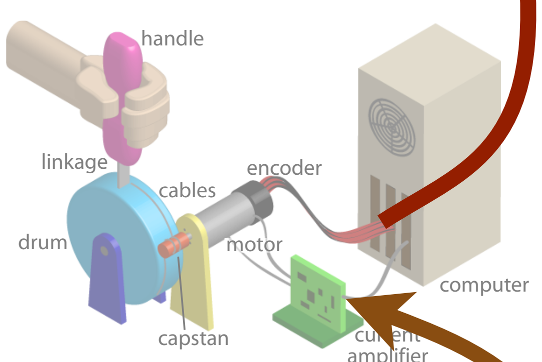
$$\begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix}$$

Output: from counts to command signals

Digital or Serial or D2A
(Digital-to-Analog Converter)

from the motion sensors

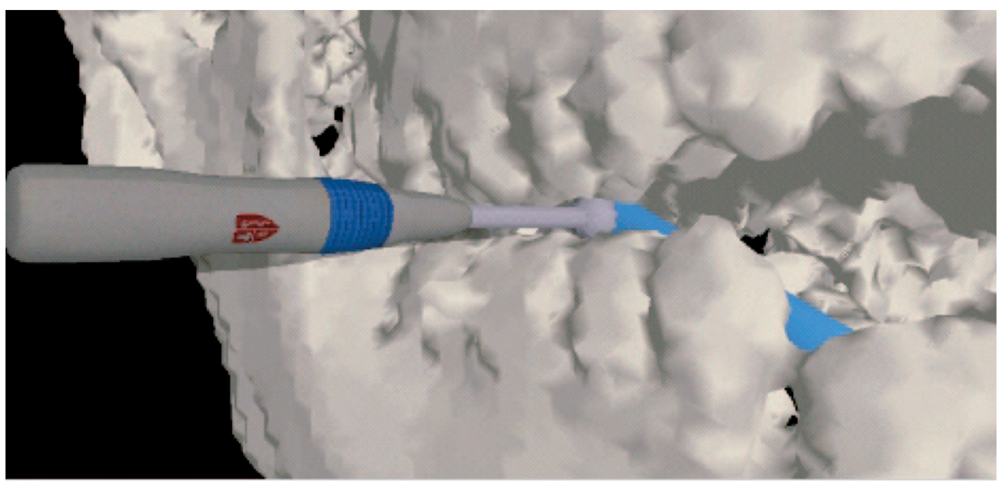
to the current amplifiers

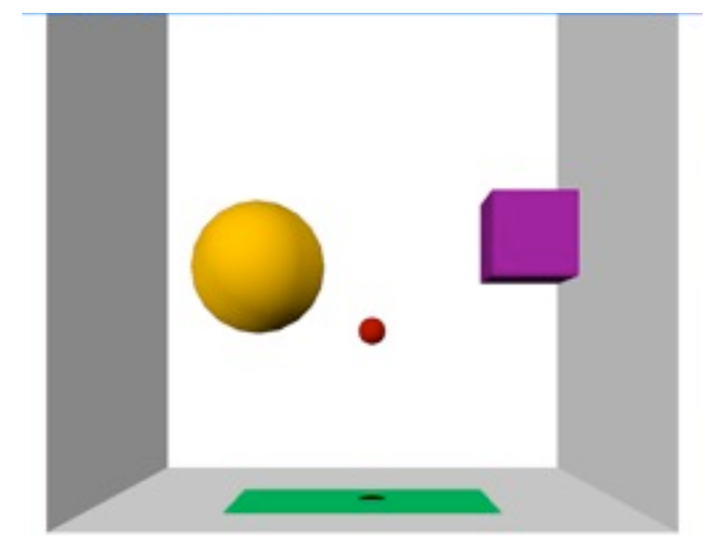


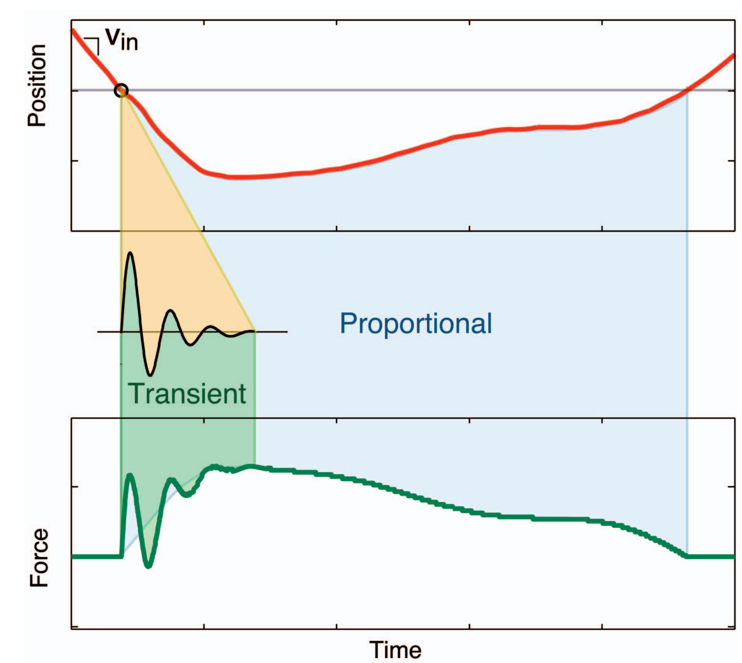
Force Computation

Force Computation: from tip position to tip force

$$\boxed{\vec{f}_k = F_i(\vec{x}_k)}$$







Computing Joint Effort

Torque Computation: from tip force to joint torques

$$\vec{\tau}_k = [J(\vec{q}_k)]^T \vec{f}_k$$

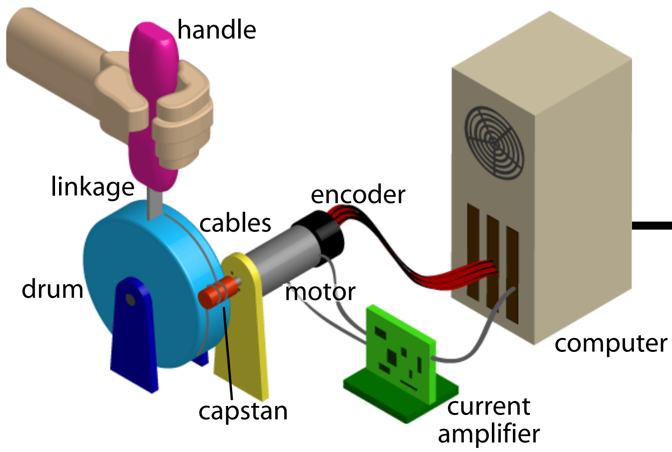
- Calculate the device's Jacobian matrix, ***J***, which depends on the present configuration.
- Multiply Cartesian tip force by ***J^T***, the transpose of the Jacobian, to transform it to joint efforts (torques and forces).
- Note that you need a different combination of joint torques to create the same force at different locations in the workspace.
- This approach stems from virtual work.

Output Processing Steps

Output Processing: from joint torques to counts

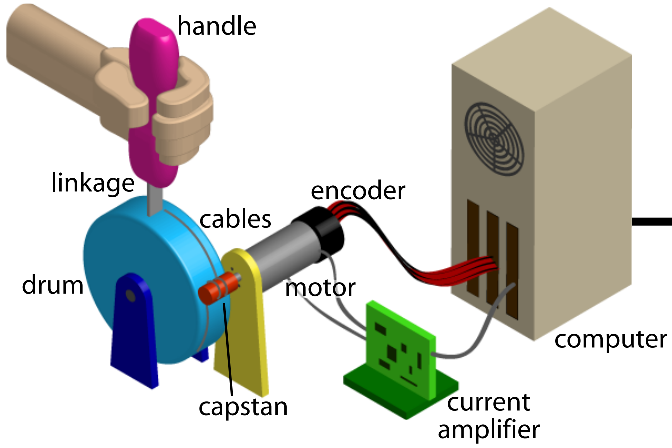
$$\vec{r}_k = \vec{c} * \vec{\tau}_k + \vec{d}$$

- Know joint efforts (torques τ_i and/or forces F_i) required to create the desired Cartesian tip force.
- Calculate motor torques τ_{mi} and/or linear motor forces F_{mi} that will create those joint efforts, using gear ratios. Remember to flip signs if needed.
- Compute the required actuator currents i_i using the torque constant of the motor.
- Compute appropriate command voltages V_i using knowledge of the current amplifier.
- Work out DAC counts r_i that will generate voltages.

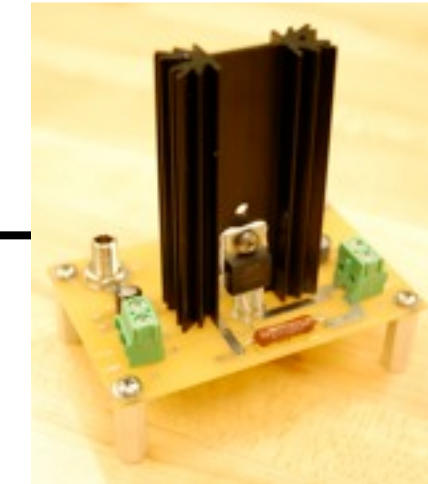


Computer

- Runs a servo loop at a fixed rate, often 1000 Hz
- Electrically connects to your hardware through an ISA card, PCI card, Firewire cable, USB cable, or other.
  - Sensoray 626, Acromag 341, 482, 732, others
- At each iteration, samples all of the sensors, computes the location of the user's hand, determines the force that should be exerted in response, and sends appropriate current commands to all of the system's actuators.
  - Need to read manual to understand how it works.

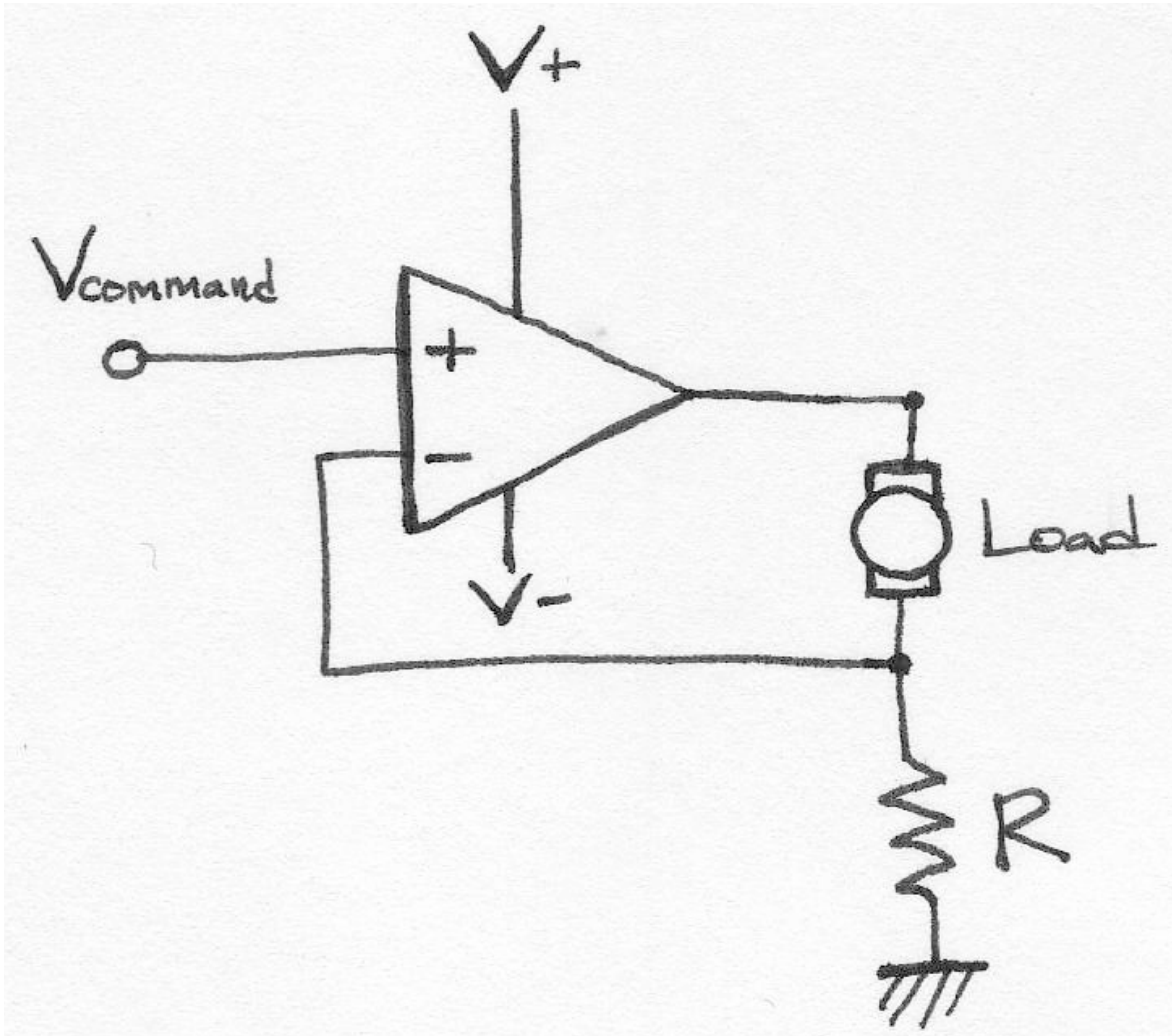


Current Amplifier



- Takes an information signal (usually an analog voltage) from the computer and drives the requested amount of current through the actuator.
- Note that this is a *current drive* scenario, not a voltage drive. Motor torque is proportional to current, regardless of speed, so we can essentially ignore the motor's electrical dynamics.
- Two common types are Pulse Width Modulation (PWM) and Linear. KJK prefers linear amplifiers for their high bandwidth and reduced electrical noise.

Current Amplifier Circuit



Typical Software Configuration

Input: from sensor signals to counts

Counter or Serial or A2D
(Analog-to-Digital Converter)

$$\begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix}$$

Input Processing: from counts to joint values

$$\vec{q}_k = \vec{a} * \vec{Q}_k + \vec{b}$$

$$\begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} \text{ or } \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} \text{ or } \begin{bmatrix} d_1 \\ \theta_2 \\ d_3 \end{bmatrix} \text{ or } \begin{bmatrix} \vdots \end{bmatrix}$$

Forward Kinematics: from joint values to tip position

$$\vec{x}_k = \Lambda(\vec{q}_k)$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Graphics, Remote Robot, and Other Slower Processes

Force Computation: from tip position to tip force

$$\vec{f}_k = F_i(\vec{x}_k)$$

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix}$$

Torque Computation: from tip force to joint torques

$$\vec{\tau}_k = [J(\vec{q}_k)]^T \vec{f}_k$$

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix}$$

Output Processing: from joint torques to counts

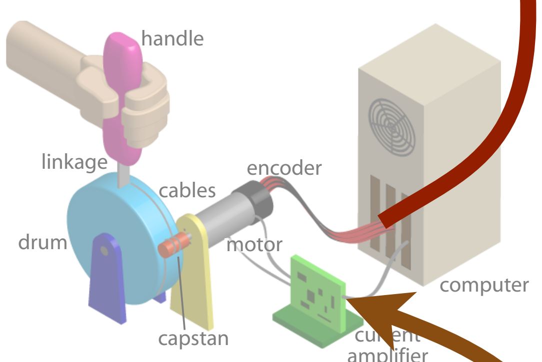
$$\vec{r}_k = \vec{c} * \vec{\tau}_k + \vec{d}$$

$$\begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix}$$

Output: from counts to command signals

Digital or Serial or D2A
(Digital-to-Analog Converter)

from the motion sensors



to the current amplifiers