We would like to acknowledge the many colleagues whose course materials were borrowed and adapted in putting together this course, namely: Drs. Allison Okamura (JHU), Katherine Kuchenbecker (U Penn), Francois Conti, Federico Barbagli, and Kenneth Salisbury (Stanford), Ed Colgate (Northwestern), Hong Tan (Purdue), Blake Hannaford and Ganesh Sankaranarayanan (U. Washington), and Karon MacLean (UBC).

Today’s Class

- Physical Haptic Systems
  - Actuators for haptic/tactile systems
    - Actuator types
    - Actuator examples
  - Sensors
  - Amplifier
  - Control Card
  - Overview of Control Architecture
    - Overall system view
    - Kinematics
- Readings

Physical Haptic Systems

- Basic Elements of a haptic system
  - Physical device (the plant of a control system)
    - Actuators and transmission
    - Sensors
  - High power (current) amplifier + power supply (what magnitude of currents?)
  - Controller
    - Computer Controller
      - Control (DAQ) Card
      - Servo control loop (typically at $\geq 1$kHz) Why???
    - Analog Controller (quite rare)


Physical Haptic Systems

- Actuators and transmission

Haptic/Tactile Actuator Types

- Electromagnetic Actuators
  - Electric motors
    - DC Brushed PM Motors (most commonly used in haptics)
    - DC Brushless
    - Stepper Motors
    - Voice coils and linear motors
    - Lawrence force
- Pneumatic Actuators
- Hydraulic
- SMA
- Piezoelectric
- Electro-rheological
- Magneto-rheological (and magnetic particle brakes)
- Not typically used in haptic systems. Why??

Brushed DC (direct current) PM (permanent magnet) motors

- Most commonly used actuator in haptic devices
- How do they work?
  - Rotating armature with coil windings is caused to rotate relative to a permanent magnet
  - current is transmitted through brushes to armature, and is constantly switched so that the armature magnetic field remains fixed

DC Motors: Two-pole Example

- 3-pole most common (Maxon motors have more)
- Power supplied through brushes generates a magnetic field around the armature.
- The north pole of the armature is pushed away from the north magnet and pulled toward the south, causing rotation.
- When the armature becomes horizontal, the connections between the commutator and the brushes reverse, and the process repeats.

DC motor terms

- Cogging / Torque Ripple
  - Tendency for torque output to ripple as the brushes transfer power (on bad motors this can be felt)
- Friction/damping
  - Caused by bearings and eddy currents (mostly, it's coulomb friction that you feel, but at high speeds, especially through a geared transmission, viscous friction can begin to dominate)
- Stall torque
  - Max torque delivered by motor when operated continuously without cooling (in most applications, you design motors to stay away from stall for efficiency. However, haptic devices are mostly standing still)
Motor Equations

- Torque constant, $K_T$
  \[ T = K_T I \]
  \[ K_t = K_e \rightarrow \text{for } K_t \equiv N^*m/A,\; K_e \equiv V/(\text{Rad/s}) \]
  \[ K_t = 9.55 \times 10^{-3} \text{ Ke } \rightarrow \text{for } K_t \equiv N^*m/A,\; K_e \equiv V/\text{KRPM} \]
  \[ K_t = 1.3524 \text{ Ke oz*in/A} \rightarrow \text{for } K_t \equiv \text{oz*in/A},\; K_e \equiv V/\text{KRPM} \]

- Dynamic equation
  \[ V = L \frac{dI}{dt} + RI + E \]

Brushed DC motors

- Pros
  - High bandwidth linear actuator
  - Easy to obtain with a wide variety of sizes
  - Easy to control

- Cons
  - To get large torques, the armature inertia gets too high to make a good impedance device
  - Generates lots of heat
  - Relatively expensive for good quality motors (e.g., Maxon motors)

DC Motors with leadscrews or ball screws for linear motion

- Pros
  - High mechanical advantage and high forces
    - Can render stiff interactions

- Cons
  - Most not backdrivable
  - Requires (expensive) force sensor
  - Difficult controller design
  - Possible eccentric vibration

DC Motors with capstan drives for rotary motion

- Pros
  - Cog-less “gear-ratio” transmission

- Cons
  - Issues with friction transmission of cable on capstan pulley
  - High cable tension $\rightarrow$ high drag
  - Possibly high reflected $N^2$ inertia
DC Motors with capstan drives for linear motion

- **Pros**
  - Cog-less linear transmission

- **Cons**
  - Relatively low forces
  - Issues with friction transmission of cable on capstan pulley

---

Voice Coil Actuators

- **Pros**
  - Very high bandwidth linear actuator
  - Relatively high forces

- **Cons**
  - Large relative size
  - Short stroke
  - Generates lots of heat

---

Lawrentz force magnetic levitation devices

- **Pros**
  - 6-DOF force feedback

- **Cons**
  - Complexity
    - hardware design
    - Controller
  - Limited workspace

---

Pneumatic actuators

- **How do they work?**
  - Compressed air pressure is used to transfer energy from the power source to haptic interface.

- **Many different types** (piston, McKibben, …)

- **Concerns are friction and bandwidth**

---
Pneumatic actuators, cont.

- McKibben Artificial Muscle, pneumatically actuated with similar properties to biological muscles
  - Spring-like, flexible, for use within robotic arms and for prosthesis
  - High force to weight ratio
  - Constructed from internal air bladder and outer braided weave shell
  - As bladder inflates, shell allows for expansion while maintaining cylindrical shape, causing the muscle to shorten

Examples of pneumatic actuators for tactile displays

- Shadowrobot hand
- Shadowrobot, McKibben Actuator
- Deflated – extended
- Inflated – contracted
- http://www.shadowrobot.com/airmuscles/

Hydraulic actuators

- Pros
  - High forces
  - Small actuator size on arm or end effector
- Cons
  - Dangerous: Can kill you!!
  - Need big hydraulic pump
  - Hydraulic seals need maintenance

Shape Memory Alloy (SMA) (Linear Actuators)

- Most commonly in wire form: contracts when heated
  - Repeated strains can be up to 3%
- Based upon a solid state phase change (molecular realignment)
  - Two phases: Martensite and Austenite
  - At low temperatures, the material is Martensite
  - If the Martensite material is loaded, this creates a deformation (typically seen as a shortening wire)
  - ‘Shape memory’ occurs when heated: the material will ‘remember’ its undeformed shape by heating to the Austenite phase
  - Since the Austenite and undeformed Martensite phases have the same shape
Shape Memory Alloy (SMA)

- Pros
  - Relatively robust
  - High force / High power density
  - Small size
  - Inexpensive and readily available (especially in wire form)

- Cons
  - Low efficiency (bad for portable applications)
  - Gets hot (bad for direct contact with skin typically… however, see [Scheibe et al. 2007] on next slide).

Examples of SMA Tactile Displays


© R. Wood (Harvard ES159/259) 2008 + W. Provancher 2009

Piezoelectric Actuators

- Piezoelectric crystal strains when voltage is applied
  - Strains can be up to 0.1%

- Details
  - Direct piezoelectric effect: a piezoelectric material will generate charges (electric field) upon the application of stress along a properly poled direction
  - Converse piezoelectric effect: a piezoelectric material will change its shape upon the application of an electric field (along a poled direction)

Examples of Piezoelectric Tactile Displays

Winfield & Colgate (2007)

TPaD: Tactile Pattern Display.

01 Vibration mode of bending element

Winfield & Colgate (2007)

Piezoelectric tactile pin array

[Summers & Chanter 2002]

© W. Provancher 2009
Physical Haptic Systems

- **Sensors**

  - **Position Sensor**
  - **Force Sensor**


We'll discuss the most common of these

Force Sensors

- **How do they work?**
  - Typically a flexure + a strain gage (sometimes also piezoelectric sensors, but these tend to drift)
    - Good quality 6-axis force-torque sensor is ~$6-7K
  - **Force Sensing Resistor (FSR)**
    - Piezoresistive ink
    - Tons of sensor drift
    - Cheap ~$10

Tekscan Flexiforce FSR

www.tekscan.com/flexiforce

www.interlinkelectronics.com

We'll discuss the most common of these

Tracking Sensor Applications

- **Eye tracking**
- **Head tracking**
- **Body tracking**
- **Hand tracking**
  - Most important for typical haptic interfaces
**Degrees of Freedom**

- Number of independent position variables needed to in order to locate all parts of a mechanism
- DOF of motion
- DOF of sensing
- DOF of actuation
- DOF does not always correspond to number of joints!

**Sensor types**

- Magnetic
- Optical
- Acoustic
- Inertial
- Mechanical
  - Most important for typical haptic interfaces

**Mechanical Trackers**

- Ground-based linkages most commonly used
- Position Sensors
  - Digital: encoders (optical or MR)
  - Analog: Hall-effect (magnetic)

**Optical Encoders**

- How do they work?
  - Focused beam of light aimed at a paired photodetector is interrupted by a coded patterned disk
  - Produces a number of pulses per revolution (↑ pulses = ↑ cost)
  - Quantization problems at low speeds
- Absolute vs. differential encoders
  - Why choose one vs. other?
  - Extra considerations if choosing incremental?
Optical Encoders

- Phase-quadrature encoder
  - 2 channels, 90° out of phase
  - allows sensing of direction of rotation
  - 4-fold increase in resolution
  - Schmitt trigger electronics create square waveform from analog detector signal
- MR encoders work similarly, but with magnetic sensors and metal disk

Hall-Effect Sensors

- How do they work?
  - A small transverse voltage is generated across a current-carrying conductor in the presence of a magnetic field

Hall-Effect Sensors

\[ V_h = \frac{R_h IB}{t} \]

- Amount of voltage output related to the strength of magnetic field passing through.
- Linear over small range of motion
  - Need to be calibrated
- Affected by temperature, other magnetic objects in the environments
Hall-Effect Sensors

From Stanford Haptic Paddle

- The voltage varies sinusoidally with rotation angle

\[ V_h = \frac{R_h IB}{t} \]

Measuring Velocity

- Differentiate position
  - advantage: use same sensor as position sensor
  - disadvantage: get noise signal
- Alternative
  - for encoders, measure time between ticks

Digital differentiation

- Many different methods
- Simple Example:
  \[ v = \frac{P2 - P1}{t} \]
  - Position reading at time 1 = P1
  - Position reading at time 2 = P2
  - t is the period of the servo loop (in sec. or counts)
  - The position is typically sampled on a fixed interval
- Differentiation increases noise

Noisy Velocity readings

- Noise on velocity signal can create jitter, especially at high \( K_v \) gains (many haptic controllers don't even include damping)
- Common solutions
  - Tach/Generator
    - Voltage a speed (same source as back EMF)
    - Resolution only as great as your A/D converter
    - Not great at low speeds
  - Integrate the signal from an accelerometer
    - Time per tick rather than ticks per time
    - use a special chip that measures time between ticks
    - Especially good to do at slow speeds
    - Fares poorly at high velocities
  - Filtering (conventional to smooth or Kalman filtering to combine sensor signals)
- Almost always FILTER your velocity signal!!
Digital differentiation
Dealing with noise by filtering

- How do you deal with noise
  - Filtering
  - Example 1: Pre-filter positions
    - Average 20 readings = P1
    - Average next 20 readings = P2
  - Example 2: Post-filter Velocities (more common)
    - Position reading at time 1 = P1
    - Position reading at time 2 = P2
    - Calculate instantaneous velocity
    - Set velocity equal to filtered version of instantaneous velocity
      - Filtered velocity = Filtered instantaneous velocity
        used in controller

Digital Filtering

- Common filter types
  - FIR filters
  - Median (with defined sample window)
  - Moving average (with defined sample window)
  - IIR filters
    - Fading memory filter
      - Filtered value = old_value * filter_ratio
        + current_value * (1 – filter_ratio)
      - Typical filter ratio 0.8 to 0.9
      - Can simulate in Matlab with a step input to figure out equivalent cutoff frequency (function of filter_ratio and sample frequency)
        - Filter output = 0.667 at $\tau$ sec and 10%-90% rise time ~2.2 $\tau$
        - Time constant $\tau$ = RC = 1/$\omega_0$, Corner frequency $f_c = 1/(2 \pi \tau)$
        - E.g., 90:10 fading memory filter (i.e., filter ratio = 0.9) has $f_c = 15.2$ Hz when running a 1 KHz servo loop.
      - Set $f_c$ near (typically above) the mechanical time constant of your system
      - Why not just set filter_ratio = .999 ???

Physical Haptic Systems

- Power Amplifier

Power (Current) Amplifiers

- PWM (Pulse Width Modulation)
  - More efficient
  - Copley, AMC, etc. make PWM amps with closed-loop current control capability (~$350)
    - Voltage in \rightarrow proportional current out
  - Other low end H-Bridge PWM amps also available (e.g., LMD18200, ~$10)
- Linear
  - Higher bandwidth than PWM, but not typically needed
  - Haptic paddle uses a LM675 high current op. amp
  - Many commercial linear amps also available
Example of PWM Control Signal

- Varying the duty cycle of the PWM signal is analogous to changing the voltage.

**50% Duty Cycle**

**~90% Duty Cycle**

**Physical Haptic Systems**

- Computer Controller
- Control (DAQ) Card


**DAQ/Controller Computer Card**

- Usually a PCI bus card

**Some Terms**

- **AD/DA**
  - analog to digital
  - digital to analog
- **Interrupt service routine**
- **Servo Loop**
- **Servo rate**
  - Usually needs to be >500 Hz, and most often is 1,000 Hz

**Common control cards**

- Sensoray 626 (~$700)
  - Four 14-bit D/A outputs, up to 6 encoder inputs,…
- dSpace 1104 (ACE1104CLP, ~$5K)
- Quanser Q4 and Q8 (≥ $4K)
- IOTech (e.g., DaqBoard/3001 ~$1K))
- Servo2go (ISA bus only, but 8 ch. for ~$1K)

**Computer Boards**

- DAC and Encoder inputs on separate cards

**National Instruments**

- DAC and Encoder inputs on separate cards

© W. Provancher 2009

© A. Okamura 2006
D/A and A/D

- Converts between voltages and counts
- Computer stores information digitally, and communicates with the outside world using +/-10V signals
- E.g., for 8-bit 0-5V ADC
  \[ 2.5V = 10000000 \]
  MSB  LSB

<table>
<thead>
<tr>
<th>Decimal</th>
<th>Binary</th>
<th>Hexadecimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0000</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0001</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0010</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>0011</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>0100</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>0101</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>0110</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>0111</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>1000</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>1001</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>1010</td>
<td>A</td>
</tr>
<tr>
<td>11</td>
<td>1011</td>
<td>B</td>
</tr>
<tr>
<td>12</td>
<td>1100</td>
<td>C</td>
</tr>
<tr>
<td>13</td>
<td>1101</td>
<td>D</td>
</tr>
<tr>
<td>14</td>
<td>1110</td>
<td>E</td>
</tr>
<tr>
<td>15</td>
<td>1111</td>
<td>F</td>
</tr>
</tbody>
</table>

Physical Haptic Systems

- Computer Controller
  - Controller code/software

Typical Software Configuration

The Prototypical Closed Loop Control System

Example controller
Implementing the simplest possible PID Controller

Control Effort = \( K_p \cdot e + K_i \cdot \sum e - K_D \cdot v \)

- \( K_p \): Proportional Gain
- \( K_i \): Integral Gain
- \( K_D \): Derivative Gain
- \( e \): Error = Desired Position – Actual Position
- \( v \): Velocity

Other things your controller will need
- Grounded interfaces (e.g., force feedback devices)
  - Very similar to robots
  - Need Kinematics
    - Determine endpoint position
    - Calculate velocities
    - Calculate force-torque relationships
  - Sometimes need Dynamics
- If you are weak on these topics, you may want to check out Introduction to robotics: mechanics and control by Craig (1989) or notes from your robotics course.

Kinematics
- Think of a manipulator/ interface as a set of bodies connected by a chain of joints
- Revolute most common for haptic interfaces
**Kinematics**

- Moving between **Joint Space** and **Cartesian Space**

- Forward kinematics: based on joint angles, calculate end-effector position

\[
x = L_1 \cos(\theta_1) + L_2 \cos(\theta_2)
\]
\[
y = L_1 \sin(\theta_1) + L_2 \sin(\theta_2)
\]

- Sometimes done this way with haptic devices (e.g., can be used when controlling elbow and shoulder joints of a Phantom Premium since its joint motions cause motion independently at the end effector)

**Joint variables**

- Be careful how you define joint positions

\[
x = L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2)
\]
\[
y = L_1 \sin(\theta_1) + L_2 \sin(\theta_1 + \theta_2)
\]

- Usually done this way with robots, sometimes with haptic devices
- Typically used if you have actuators at joints (i.e., relative actuation)
Velocity

- Recall that the forward kinematics tells us the end-effector position based on joint positions
- How do we calculate velocity?
- Use a matrix called the Jacobian

\[
\dot{x} = J \dot{\theta}
\]

Formulating joint torques

- The Jacobian can be used to relate joint torques to end-effector forces:

\[
T = J^T F
\]

- Why is this important for haptic displays?

Reading for next time


- Next lecture will be on “Overview of dynamics/controls & terminology”

Supplemental Information on Kinematics and Jacobians

- Although “Introduction to Robotics” is a prerequisite for this class, we recognize that some of you may be a little rusty with kinematics, hence the pages that follow will provide a refresher as a reference to look at on your own
Joint variables

- Be careful how you define joint positions

![Diagram of joint variables](image)

Absolute forward kinematics

\[ x = L_1 \cos(\theta_1) + L_2 \cos(\theta_2) \]
\[ y = L_1 \sin(\theta_1) + L_2 \sin(\theta_2) \]

- Sometimes done this way with haptic devices (e.g., can be used when controlling elbow and shoulder joints of a Phantom Premium since its joint motions cause motion independently at the end effector)

![Diagram of absolute forward kinematics](image)

Relative forward kinematics

\[ x = L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) \]
\[ y = L_1 \sin(\theta_1) + L_2 \sin(\theta_1 + \theta_2) \]

- Usually done this way with robots, sometimes with haptic devices
- Typically used if you have actuators at joints (i.e., relative actuation)

![Diagram of relative forward kinematics](image)

Inverse Kinematics

- Using the end-effector position, calculate the joint angles
- Not used often in haptics
  - But useful for calibration
- There can be:
  - No solution (workspace issue)
  - 1 solution
  - More than 1 solution

![Diagram of inverse kinematics](image)
Example

- Two possible solutions
  - Two approaches:
    - algebraic method
    - geometric method

Velocity

- Recall that the forward kinematics tells us the end-effector position based on joint positions
- How do we calculate velocity?
- Use a matrix called the Jacobian

\[ \dot{x} = J \dot{\theta} \]

Formulating the Jacobian

- Use the chain rule:
  \[ \dot{x} = \frac{\partial x}{\partial \theta_1} \dot{\theta}_1 + \frac{\partial x}{\partial \theta_2} \dot{\theta}_2 \]
  \[ \dot{y} = \frac{\partial y}{\partial \theta_1} \dot{\theta}_1 + \frac{\partial y}{\partial \theta_2} \dot{\theta}_2 \]

- Substitute in expressions for \( x \) & \( y \) from before and take partial derivatives:
  \[ \frac{\partial x}{\partial \theta_1} = -L_1 \sin(\theta_1) - L_2 \sin(\theta_1 + \theta_2) \]
  \[ \frac{\partial x}{\partial \theta_2} = -L_2 \sin(\theta_1 + \theta_2) \]
  \[ \frac{\partial y}{\partial \theta_1} = L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) \]
  \[ \frac{\partial y}{\partial \theta_2} = L_2 \cos(\theta_1 + \theta_2) \]

Assemble in a Matrix

\[
\begin{bmatrix}
\dot{x} \\
\dot{y}
\end{bmatrix} = \begin{bmatrix}
\frac{\partial x}{\partial \theta_1} & \frac{\partial x}{\partial \theta_2} \\
\frac{\partial y}{\partial \theta_1} & \frac{\partial y}{\partial \theta_2}
\end{bmatrix}
\begin{bmatrix}
\dot{\theta}_1 \\
\dot{\theta}_2
\end{bmatrix}
\]

\[
\begin{bmatrix}
\dot{x} \\
\dot{y}
\end{bmatrix} = \begin{bmatrix}
-L_1 \sin(\theta_1) - L_2 \sin(\theta_1 + \theta_2) & -L_2 \sin(\theta_1 + \theta_2) \\
L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) & L_2 \cos(\theta_1 + \theta_2)
\end{bmatrix}
\begin{bmatrix}
\dot{\theta}_1 \\
\dot{\theta}_2
\end{bmatrix}
\]
**Singularities**

- Most devices will have values for which the Jacobian is singular
- This means that the device has lost one or more degrees of freedom in Cartesian Space
- Two kinds:
  - Workspace boundary
  - Workspace interior

**Singularity Math**

- If the matrix is invertible, then it is non-singular
  \[ \dot{\theta} = J^{-1} \dot{x} \]
- Can check invertibility if \( J \) by taking the determinant of \( J \). If it is equal to 0, then it is singular
- Can use this method to check which values of \( \theta \) will cause singularities

**Calculating Singularities**

- Simplify nomenclature: \( \sin(\theta_1+\theta_2) = s_{12} \)
  \[
  \det(J(\theta)) = \begin{vmatrix} -L_1s_1 - L_2s_{12} & -L_2s_{12} \\ L_1c_1 + L_2c_{12} & L_2c_{12} \end{vmatrix} = (-L_1s_1 - L_2s_{12})L_2c_{12} + (L_1c_1 + L_2c_{12})L_2s_{12}
  \]
- For what values of \( \theta_1 \) and \( \theta_2 \) does this equal zero?

**Even more useful....**

- The Jacobian can be used to relate joint torques to end-effector forces:
  \[ T = J^T F \]
- Why is this important for haptic displays?
How do you get this equation?

- **Principle of virtual work**
  - States that changing the coordinate frame does not change the total work of a system

\[
\begin{align*}
    F^T \delta x &= \tau^T \delta \theta \\
    F^T J \delta \theta &= \tau^T \delta \theta \\
    F^T J &= \tau^T \\
    J^T F &= \tau
\end{align*}
\]