Goals:
- Describe how control laws are implemented in engineering systems
- Briefly summarize the main principles and tools for the course

Reading:
- Åström and Murray, *Analysis and Design of Feedback Systems*, Ch 10

Modern Control System Components

- **Process**: Physical system, actuation, sensing
- **Controller**: Microprocessor plus conversion hardware (single chip)
- **Feedback**: Interconnection between plant output, controller input
Frequency Response of Control Components

Components limit performance

- Often ignore sensor, actuator, computation dynamics when designing controllers
- Each adds dynamics that limit achievable bandwidth
- OK to ignore if components are sufficiently high performance

General guidelines

- Sensor: 5-10X BW
- Actuation: 2-5X BW
- Computing: 10-20X BW

Other issues

- Sensor aliasing – filter response should be small at ½ sampling rate

Plant: second order, tracking to 20 rad/sec
Sensing: 1000 rad/s bandwidth (1st order lag)
Actuation: 100 rad/s bandwidth (1st order lag)
Computation: 500 Hz (3000 rad/sec) update

Modern Implementation Environments

Graphical control environments

- Define controller using traditional block diagrams
- Program generates C code corresponding to digital implementation
  - Create state space realization of transfer functions
  - Convert to discrete time equivalent
- Compile code on target processor environment (PC or embedded proc)
- Real-time interface allows monitoring and debugging

Example: dSPACE real-time control system

- Compiles Simulink diagrams for DSP processors
Example: Multi-Vehicle Wireless Testbed (MVWT)

Testbed features:
- Distributed computation on vehicles + command and control console
- Point to point networking (bluetooth) + local area networking (802.11)
- Overhead vision system provides global position data (LPS)

Caltech Hovercraft

- Interface board
- Gyro/accel (sense)
- 802.11 wireless
- Zaurus PDA (compute)
- "Hat" (sense)
- Thrust fan (actuate)
- Lift fan

Cremean et al
CDC 2002

Jin, Waydo, Wildanger, Lammers, Scholze, Foley, Held, Murray
ACC 2004
Dynamics

Vehicle dynamics:
\[
\begin{align*}
\dot{x} &= f_1 \cos \theta - f_2 \sin \theta - \dot{x} \\
\dot{y} &= f_1 \sin \theta + f_2 \cos \theta - \dot{y} \\
J \dot{\theta} &= rf_1
\end{align*}
\]

Fan dynamics
\[
\begin{align*}
\dot{\omega}_1 &= -\tau \alpha + \alpha(i) \\
\dot{\omega}_2 &= -\tau \alpha + \alpha(i)
\end{align*}
\]

Vision System
- Time delay + noise (pixelation)
- Variable time delay

Wireless network
- Nonlinear map

Linearization

Linearize around a constant velocity along \( y \)
\[
\begin{align*}
\dot{v} &= v_0 \\
x &= 0, \dot{x} = 0 \\
\theta &= 0, \dot{\theta} = 0
\end{align*}
\]

Shift coordinates to the origin and write in state space form:
\[
\begin{align*}
\dot{\tilde{v}} &= \tilde{v} - v_0 \\
\dot{u}_1 &= f_2 \\
\dot{u}_2 &= f_1 - dv_0
\end{align*}
\]

Parameters (Kelly II)
\[
\begin{align*}
m &= 5.5 \\
b &= 0.1 \\
J &= 0.047 \\
r &= 0.123 \\
d &= 0.5
\end{align*}
\]

Remarks
- Ignores actuator dynamics (assume fast)
- Ignores time delays (for now)
Control Design: State Space

\[ \begin{bmatrix} \dot{v} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} -\frac{d}{m} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{bJ} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{d\theta}{m} & -\frac{d}{m} \end{bmatrix} \begin{bmatrix} v \\ \theta \\ y \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 1/m \\ 0 \\ 0 \\ 0 \end{bmatrix} u \]

Choose control to stabilize error, \( e = x-x_d \rightarrow u = K(x-x_d)+u_d \)

- \( x_d \) = desired state
- \( u_d \) = nominal force

Control Design: Frequency Domain

\[ P_{\text{req}} = \frac{1}{s+d} \quad P_{\text{req}} = \frac{J\omega^2 + bs + Jd\omega/m}{(ms^2 + ds)(J\omega^2 + bs)} \]

Compute transfer functions using \( H = C(sI-A)^{-1}B \)

Use loop shaping to design compensator

- Forward velocity: simple proportional gain
- Lateral position: use lead compensator
Control Design: Frequency Domain

\[ P_{eq} = \frac{1}{s + d} \]
\[ C_i = K_p \]
\[ K_p = 100 \]
\[ a = 0.1 \]
\[ b = 10 \]

\[ P_{eq} = \frac{Js^2 + bs + Jd\nu_i/m}{(ms^2 + ds)(Js^2 + bs)} \]
\[ C_i = \frac{s + a}{s + b} \]

Frequency (rad/sec)

Phase (deg); Magnitude (dB)

Time Delay Estimates

- Camera takes picture: 8 ms
- LPS Broadcasts data: 33 ms
- Laptop receives data: 6 ms

- PIC sends new PWM signal and lights the LEDs: 11 ms
- PIC receives data: 6 ms
- Controller receives data: 1 ms

EVENT

- No actuator delay included in estimates or measurement
- Total delay estimate: 65 ms
- Measured delay: 65.3 ms average
Unmodeled Dynamics: Sensor Delay

Control algorithm  →  Fan dynamics  →  Vehicle dynamics

Wireless network  →  Vision system  →  Plant

Bode Diagrams

Frequency (rad/sec)

Phase (deg)

Magnitude (dB)

Loop xfer fcn

From: U(1)

To: Y(1)

Amplitude

Time (sec.)

Step Response

From: U(1)

To: Y(1)

Amplitude

Time (sec.)

Step Response

Summary: Control Implementation

Control = sense, compute, actuate
- Computation can be done in analog or digital form
- Sensing should be high bandwidth, low noise for best performance
- Modern environments allow direct implementation through Simulink

Actuator  →  System  →  Sensor  →  Process

Stability   Performance   Robustness

Operator input  →  A/D  →  Computer  →  D/A  →  Controller

29 November 2004