Next Time in Linear Systems ….

<table>
<thead>
<tr>
<th>Week</th>
<th>Date</th>
<th>Lecture Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>15-May</td>
<td><strong>PID &amp; Computer Control</strong></td>
</tr>
<tr>
<td>12</td>
<td>22-May</td>
<td>10.5: Applications in Industry</td>
</tr>
<tr>
<td>13</td>
<td>29-May</td>
<td>10.6: Information Theory/Communications &amp; Review</td>
</tr>
<tr>
<td>14</td>
<td>31-May</td>
<td>10.7: Summary and Course Review</td>
</tr>
</tbody>
</table>

ELEC 3004: Systems: Signals & Controls
Dr. Surya Singh

Lecture 20

elec3004@itee.uq.edu.au
http://robotics.itee.uq.edu.au/~elec3004/

© 2013 School of Information Technology and Electrical Engineering at The University of Queensland
Goals for the Week

- Stability of Digital Systems
- Lead/Lag Compensators
- ZOH and Discretisation

Two classes of control design

The system...
- Isn’t fast enough
- Isn’t damped enough
- Overshoots too much
- Requires too much control action
  (“Performance”)

- Attempts to spontaneously disassemble itself
  (“Stability”)

ELEC 3004: Systems
Recall dynamic responses
• Moving pole positions change system response characteristics

Recall dynamic responses
• Ditto the z-plane:
The fundamental control problem

The poles are in the wrong place

How do we get them where we want them to be?

Recall the root locus

• We know that under feedback gain, the poles of the closed-loop system move
  – The root locus tells us where they go!
  – We can solve for this analytically*

Root loci can be plotted for all sorts of parameters, not just gain!
Dynamic compensation

- We can do more than just apply gain!
  - We can add dynamics into the controller that alter the open-loop response

\[
\begin{align*}
\text{compensator} & \quad \text{plant} \\
\frac{s + 2}{s(s + 1)} & \quad \frac{1}{s(s + 1)} \\
& \quad y
\end{align*}
\]

\[\text{combined system} \quad \frac{s + 2}{s(s + 1)}
\]

- Recognise the following:
  - A root locus starts at poles, terminates at zeros
  - “Holes eat poles”
  - Closely matched pole and zero dynamics cancel
  - The locus is on the real axis to the left of an odd number of poles (treat zeros as ‘negative’ poles)

But what dynamics to add?

- Recognise the following:
  - A root locus starts at poles, terminates at zeros
  - “Holes eat poles”
  - Closely matched pole and zero dynamics cancel
  - The locus is on the real axis to the left of an odd number of poles (treat zeros as ‘negative’ poles)
Some standard approaches

- Control engineers have developed time-tested strategies for building compensators
- Three classical control structures:
  - Lead
  - Lag
  - Proportional-Integral-Derivative (PID)
    (and its variations: P, I, PI, PD)

How do they work?

Lead/lag compensation

- Serve different purposes, but have a similar dynamic structure:
  \[ D(s) = \frac{s + a}{s + b} \]

Note:
Lead-lag compensators come from the days when control engineers cared about constructing controllers from networks of op amps using frequency-phase methods. These days pretty much everybody uses PID, but you should at least know what the heck they are in case someone asks.
Lead compensation: $a < b$

- Acts to decrease rise-time and overshoot
  - Zero draws poles to the left; adds phase-lead
  - Pole decreases noise
- Set $a$ near desired $\omega_n$; set $b$ at $\sim 3$ to $20 \times a$

Lag compensation: $a > b$

- Improves steady-state tracking
  - Near pole-zero cancellation; adds phase-lag
  - Doesn’t break dynamic response (too much)
- Set $b$ near origin; set $a$ at $\sim 3$ to $10 \times b$
PID – the Good Stuff

• Proportional-Integral-Derivative control is the control engineer’s hammer*
  – For P, PI, PD, etc. just remove one or more terms

\[ C(s) = k \left( 1 + \frac{1}{\tau I s} + \tau D s \right) \]

*Everything is a nail. That’s why it’s called “Bang-Bang” Control 😄

PID – the Good Stuff

• PID control performance is driven by three parameters:
  – \( k \): system gain
  – \( \tau I \): integral time-constant
  – \( \tau D \): derivative time-constant

You’re already familiar with the effect of gain. What about the other two?
Integral

- Integral applies control action based on accumulated output error
  - Almost always found with P control
- Increase dynamic order of signal tracking
  - Step disturbance steady-state error goes to zero
  - Ramp disturbance steady-state error goes to a constant offset

Let’s try it!

**Integral**

- Consider a first order system with a constant load disturbance, \( w \); (recall as \( t \to \infty, s \to 0 \))

\[
\begin{align*}
  y &= k \frac{1}{s + a} (r - y) + w \\
  (s + a)y &= k (r - y) + (s + a)w \\
  (s + k + a)y &= kr + (s + a)w \\
  y &= \frac{k}{s + k + a} r + \frac{(s + a)}{s + k + a} w
\end{align*}
\]

Steady state gain = \( a/(k+a) \)

(never truly goes away)
Now with added integral action

\[
y = k \left( 1 + \frac{1}{\tau_s} \right) \frac{1}{s + a} (r - y) + w
\]

\[
y = k \left( s + \tau_i^{-1} \right) \frac{1}{s + a} (r - y) + w
\]

\[
s(s + a)y = k(s + \tau_i^{-1})(r - y) + s(s + a)w
\]

\[
(s^2 + (k + a)s + \tau_i^{-1})y = k(s + \tau_i^{-1})r + s(s + a)w
\]

\[
y = \frac{k(s + \tau_i^{-1})}{(s^2 + (k + a)s + \tau_i^{-1})} r + \frac{s(s + a)}{k(s + \tau_i^{-1})} w
\]

Must go to zero for constant w!

Derivative

- Derivative uses the rate of change of the error signal to anticipate control action
  - Increases system damping (when done right)
  - Can be thought of as ‘leading’ the output error, applying correction predictively
  - Almost always found with P control*

*What kind of system do you have if you use D, but don’t care about position? Is it the same as P control in velocity space?
Derivative

- It is easy to see that PD control simply adds a zero at $s = -\frac{1}{r_d}$ with expected results
  - Decreases dynamic order of the system by 1
  - Absorbs a pole as $k \to \infty$
- Not all roses, though: derivative operators are sensitive to high-frequency noise

![Bode plot of a zero](image)

PID

- Collectively, PID provides two zeros plus a pole at the origin
  - Zeros provide phase lead
  - Pole provides steady-state tracking
  - Easy to implement in microprocessors
- Many tools exist for optimally tuning PID
  - Zeigler-Nichols
  - Cohen-Coon
  - Automatic software processes
Be alert

• If gains and time-constants are chosen poorly, all of these compensators can induce oscillation or instability.

• However, when used properly, PID can stabilise even very complex unstable third-order systems.

Now in discrete

• Naturally, there are discrete analogs for each of these controller types:

  Lead/lag: \( \frac{1 - \alpha z^{-1}}{1 - \beta z^{-1}} \)

  PID: \( k \left( 1 + \frac{1}{\tau_i (1 - z^{-1})} + \tau_d (1 - z^{-1}) \right) \)

  But, where do we get the control design parameters from? The s-domain?
**Emulation vs Discrete Design**

- Remember: polynomial algebra is the same, whatever symbol you are manipulating:
  
  \[ s^2 + 2s + 1 = (s + 1)^2 \]
  
  \[ z^2 + 2z + 1 = (z + 1)^2 \]

  Root loci behave the same on both planes!

- Therefore, we have two choices:
  - Design in the s-domain and digitise (emulation)
  - Design only in the z-domain (discrete design)

---

**Emulation design process**

1. Derive the dynamic system model ODE
2. Convert it to a continuous transfer function
3. Design a continuous controller
4. Convert the controller to the z-domain
5. Implement difference equations in software
Emulation design process

- Handy rules of thumb:
  - Use a sampling period of 20 to 30 times faster than the closed-loop system bandwidth
  - Remember that the sampling ZOH induces an effective $T/2$ delay
  - There are several approximation techniques:
    - Euler’s method
    - Tustin’s method
    - Matched pole-zero
    - Modified matched pole-zero

### Tustin’s method

- Tustin uses a trapezoidal integration approximation (compare Euler’s rectangles)
- Integral between two samples treated as a straight line:
  $$u(kT) = \frac{T}{2} \left[ x(k-1) + x(k) \right]$$

Taking the derivative, then z-transform yields:
$$s = \frac{2z^{-1}}{Tz+1}$$

which can be substituted into continuous models

![Tustin's method diagram](attachment://tustin_diagram.png)
### Matched pole-zero

- If $z = e^{sT}$, why can’t we just make a direct substitution and go home?

\[
\frac{Y(s)}{X(s)} = \frac{s+a}{s+b} \quad \rightarrow \quad \frac{Y(z)}{X(z)} = \frac{z-e^{-aT}}{z-e^{-bT}}
\]

- Kind of!
  - Still an approximation
  - Produces quasi-causal system (hard to compute)
  - Fortunately, also very easy to calculate.

### The process:

1. Replace continuous poles and zeros with discrete equivalents:
   \[
   (s + a) \rightarrow (z - e^{-aT})
   \]

2. Scale the discrete system DC gain to match the continuous system DC gain

3. If the order of the denominator is higher than the numerator, multiply the numerator by $(1 - z^{-1})$ until they are of equal order*

   * This introduces an averaging effect like Tustin’s method
Modified matched pole-zero

- We’re prefer it if we didn’t require instant calculations to produce timely outputs
- Modify step 2 to leave the dynamic order of the numerator one less than the denominator
  - Can work with slower sample times, and at higher frequencies

Discrete design process

1. Derive the dynamic system model ODE
2. Convert it to a discrete transfer function
3. Design a digital compensator
4. Implement difference equations in software
5. Platypus Is Divine!
Discrete design process

• Handy rules of thumb:
  – Sample rates can be as low as twice the system bandwidth
    • but 5 to 10× for “stability”
    • 20 to 30 × for better performance
  
  – A zero at $z = -1$ makes the discrete root locus pole behaviour
    more closely match the $s$-plane
  
  – Beware “dirty derivatives”
    • $dy/dt$ terms derived from sequential digital values are called ‘dirty
      derivatives’ – these are especially sensitive to noise!
    • Employ actual velocity measurements when possible

Announcements:

• Final Exam:
  – 15 Questions (60% Short Answer, 40% Regular Problems)
  – 3 Hours
  – Closed-book
  – Took tutor ~90min to complete
  – Equation sheet will be provided (in addition to your own)
    [See Prac Final – Coming out next week]
  – Yes, it has an unexpected twist at the end, but you’ll like it. 😊

=> Saturday, June 15 at 9:30 AM (sorry!)
### Next Time in Linear Systems …. 

<table>
<thead>
<tr>
<th>Week</th>
<th>Date</th>
<th>Lecture Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>27-Feb</td>
<td>Introduction</td>
</tr>
<tr>
<td></td>
<td>1-Mar</td>
<td>Systems Overview</td>
</tr>
<tr>
<td>2</td>
<td>6-Mar</td>
<td>Signals &amp; Signal Models</td>
</tr>
<tr>
<td></td>
<td>8-Mar</td>
<td>System Models</td>
</tr>
<tr>
<td>3</td>
<td>13-Mar</td>
<td>Linear Dynamical Systems</td>
</tr>
<tr>
<td></td>
<td>15-Mar</td>
<td>Sampling &amp; Data Acquisition</td>
</tr>
<tr>
<td>4</td>
<td>20-Mar</td>
<td>Time Domain Analysis of Continuous Time Systems</td>
</tr>
<tr>
<td></td>
<td>22-Mar</td>
<td>System Behaviour &amp; Stability</td>
</tr>
<tr>
<td>5</td>
<td>27-Mar</td>
<td>Signal Representation</td>
</tr>
<tr>
<td></td>
<td>29-Mar</td>
<td>Holiday</td>
</tr>
<tr>
<td>6</td>
<td>10-Apr</td>
<td>Frequency Response</td>
</tr>
<tr>
<td></td>
<td>12-Apr</td>
<td>z-Transform</td>
</tr>
<tr>
<td></td>
<td>13-Apr</td>
<td>Noise &amp; Filtering</td>
</tr>
<tr>
<td></td>
<td>19-Apr</td>
<td>Analog Filters</td>
</tr>
<tr>
<td>7</td>
<td>24-Apr</td>
<td>Discrete-Time Signals</td>
</tr>
<tr>
<td></td>
<td>26-Apr</td>
<td>Discrete-Time Systems</td>
</tr>
<tr>
<td>8</td>
<td>3-May</td>
<td>Digital Filters &amp; IIR/FIR Systems</td>
</tr>
<tr>
<td></td>
<td>3-May</td>
<td>Fourier Transform &amp; DTFT</td>
</tr>
<tr>
<td>9</td>
<td>8-May</td>
<td>Introduction to Digital Control</td>
</tr>
<tr>
<td></td>
<td>10-May</td>
<td>Stability of Digital Systems</td>
</tr>
<tr>
<td>10</td>
<td>15-May</td>
<td>PID &amp; Computer Control</td>
</tr>
<tr>
<td>11</td>
<td>17-May</td>
<td>Applications in Industry</td>
</tr>
<tr>
<td>12</td>
<td>22-May</td>
<td>State Space</td>
</tr>
<tr>
<td></td>
<td>24-May</td>
<td>Controllability &amp; Observability</td>
</tr>
<tr>
<td>13</td>
<td>29-May</td>
<td>Information Theory/Communications &amp; Review</td>
</tr>
<tr>
<td></td>
<td>31-May</td>
<td>Summary and Course Review</td>
</tr>
</tbody>
</table>
→ **feedback**

\[ y = G(e) \]
\[ \frac{y}{r} = \frac{G}{1 + GH} \]

**What's bang-bang control**

\[ u(t) = \begin{cases} 1 & \text{for certain conditions} \\ 0 & \text{else} \end{cases} \]

**Fridge**

\[ u_F(t) = \begin{cases} 1 & \text{if } T > T^* \\ 0 & \text{if } T \leq T^* \end{cases} \]
\[ y(t) = \sin(\omega t) + 0.01 \cdot \sin(100\omega t) \]

\[ y'(t) = 100 \left( \cos(\omega t) + 100 \cdot 0.01 \cdot \cos(100\omega t) \right) \]

\[ y''(t) = -100^2 \left[ \sin(\omega t) + 100^2 \cdot 0.01 \cdot \sin(100\omega t) \right] \]

\[
\begin{array}{c}
\frac{G(s)}{1+G(s)} \\
\frac{H(s)}{1+kH} \\
1+kH = 0
\end{array}
\]

\[
\begin{array}{c}
1 \times 0.1052 \\
\frac{2}{2-1.1052}
\end{array}
\]
\[
\frac{y(s)}{x(s)} = H(s) = \frac{a}{s + a}
\]

\[\dot{y} + ay = ax\]

**Trapezoidal**

\[
HT = \frac{a}{\left(\frac{2}{T}\right) \left[ \frac{z-1}{z+1} \right] + a}
\]

**Forward**

\[
\frac{z-1}{T}
\]

**Backward**

\[
\frac{z-1}{T^2}
\]

**TRAP**

\[
\frac{2}{T} \left( \frac{z-1}{z+1} \right)
\]