Digital Control

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

Lecture 9

elec3004@itee.uq.edu.au

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

May 6, 2014

Lecture Schedule:

<table>
<thead>
<tr>
<th>Week</th>
<th>Date</th>
<th>Lecture Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4-Mar</td>
<td>Introduction &amp; Systems Overview</td>
</tr>
<tr>
<td></td>
<td>6-Mar</td>
<td>Linear Dynamical Systems</td>
</tr>
<tr>
<td>2</td>
<td>11-Mar</td>
<td>Signals as Vectors &amp; Systems as Maps</td>
</tr>
<tr>
<td></td>
<td>13-Mar</td>
<td>Signals</td>
</tr>
<tr>
<td>3</td>
<td>18-Mar</td>
<td>Sampling &amp; Data Acquisition &amp; Anti-aliasing Filters</td>
</tr>
<tr>
<td></td>
<td>20-Mar</td>
<td>Sampling</td>
</tr>
<tr>
<td>4</td>
<td>25-Mar</td>
<td>System Analysis &amp; Convolution</td>
</tr>
<tr>
<td></td>
<td>27-Mar</td>
<td>Convolution &amp; FT</td>
</tr>
<tr>
<td>5</td>
<td>1-Apr</td>
<td>Frequency Response &amp; Filter Analysis</td>
</tr>
<tr>
<td></td>
<td>3-Apr</td>
<td>Filters</td>
</tr>
<tr>
<td>6</td>
<td>8-Apr</td>
<td>Discrete Systems &amp; Z-Transforms</td>
</tr>
<tr>
<td></td>
<td>10-Apr</td>
<td>Z-Transforms</td>
</tr>
<tr>
<td>7</td>
<td>15-Apr</td>
<td>Introduction to Digital Control</td>
</tr>
<tr>
<td></td>
<td>17-Apr</td>
<td>Feedback</td>
</tr>
<tr>
<td>8</td>
<td>29-Apr</td>
<td>Digital Filters</td>
</tr>
<tr>
<td></td>
<td>1-May</td>
<td>Digital Filters</td>
</tr>
<tr>
<td>9</td>
<td>6-May</td>
<td>Digital Control Design</td>
</tr>
<tr>
<td>10</td>
<td>8-May</td>
<td>Digital Control</td>
</tr>
<tr>
<td></td>
<td>13-May</td>
<td>Stability of Digital Systems</td>
</tr>
<tr>
<td></td>
<td>15-May</td>
<td>Stability</td>
</tr>
<tr>
<td>11</td>
<td>20-May</td>
<td>State-Space</td>
</tr>
<tr>
<td></td>
<td>22-May</td>
<td>Controllability &amp; Observability</td>
</tr>
<tr>
<td>12</td>
<td>27-May</td>
<td>PID Control &amp; System Identification</td>
</tr>
<tr>
<td></td>
<td>29-May</td>
<td>Digital Control System Hardware</td>
</tr>
<tr>
<td>13</td>
<td>3-Jun</td>
<td>Applications in Industry &amp; Information Theory &amp; Communications</td>
</tr>
<tr>
<td></td>
<td>5-Jun</td>
<td>Summary and Course Review</td>
</tr>
</tbody>
</table>
Digital control

Once upon a time…

• Electromechanical systems were controlled by electromechanical compensators
  – Mechanical flywheel governors, capacitors, inductors, resistors, relays, valves, solenoids (fun!)
  – But also complex and sensitive!

• Humans developed sophisticated tools for designing reliable analog controllers

\[ \rightarrow \text{Idea: Digital computers in real-time control} \]

  – Transform approach (classical control)
    • Root-locus methods (pretty much the same as METR 3200)
    • Bode’s frequency response methods (these change compared to METR 3200)
  – State-space approach (modern control)

\[ \rightarrow \text{Model Making: Control of frequency response as well as Least Squares Parameter Estimation} \]
Many advantages

- Practical improvement over analog control:
  - **Flexible**: reprogrammable to implement different control laws for different systems
  - **Adaptable**: control algorithms can be changed on-line, during operation
  - **Insensitive** to environmental conditions;
    (heat, EMI, vibration, etc)
  - **Compact**: handful of components on a PCB
  - **Cheap**

Feedback Control

(Simple) control systems have three parts:

- The plant is the system to be controlled (e.g. the robot).
- The sensor measures the output of the plant.
- The controller sends an input command to the plant based on the difference from the actual output and the desired output.
Archetypical control system

- Consider a continuous control system:

\[
\begin{align*}
r(t) & \rightarrow e(t) \rightarrow C(s) \rightarrow u(t) \rightarrow H(s) \rightarrow y(t) \\
& \text{controller} \quad \text{plant}
\end{align*}
\]

- The functions of the controller can be entirely represented by a discretised computer system.

Simple Controller Goes Digital

\[
\begin{align*}
d_i & \leftarrow e(t) \rightarrow u(t) \rightarrow y(t) \rightarrow d_o \\
& \text{controller} \quad \text{plant} \\
\text{sensor: } y[n] & = u[n-1] \\
\text{controller: } y[n] & = Ku[n] \\
\text{plant: } y[n] & = y[n-1] - Tu[n-1] \\
\end{align*}
\]

Complex system behaviors, depending on \( K \)
Digital Control

\[ L = \frac{1}{2} M \left( \dot{x}^2 + \frac{1}{4} \dot{y}^2 \right) - mgx \cos \theta \]

where \( x \) is the position of the cart and \( y \) is the offset of the pole from the cart. \( \dot{x} \) and \( \dot{y} \) can be expressed in terms of \( \dot{\theta} \) and \( \ddot{\theta} \) by solving the equations of motion for the cart-pole system.

\[ \dot{\theta} = \left( \frac{1}{2} I \right) \left( \ddot{\theta} - \frac{1}{2} \frac{\tau}{I} \right) \]

Solving the equation for \( \tau \) leads to

\[ \tau = 2 I \dot{\theta} \cos \theta \]

The Lagrange equation is now given by

\[ L = \frac{1}{2} (I + M I) \dot{\theta}^2 - 2I M \dot{x} \dot{\theta} \cos \theta - \frac{1}{2} M \dot{y}^2 - mgx \cos \theta \]

and the equations of motion are:

\[
\begin{align*}
\dot{x} &= \frac{1}{2} I \ddot{\theta} + M I \dot{\theta} \cos \theta \\
\dot{y} &= \frac{1}{2} I \ddot{\theta} + M I \dot{\theta} \cos \theta \\
\dot{\theta} &= \frac{\tau}{2 I} - \frac{1}{2} \frac{\tau}{I} \cos \theta
\end{align*}
\]

Substituting this into the equations and solving leads to the equations that describe the motion:

\[
\begin{align*}
\dot{x} &= \frac{1}{2} I \ddot{\theta} + M I \dot{\theta} \cos \theta \\
\dot{y} &= \frac{1}{2} I \ddot{\theta} + M I \dot{\theta} \cos \theta \\
\dot{\theta} &= \frac{\tau}{2 I} - \frac{1}{2} \frac{\tau}{I} \cos \theta
\end{align*}
\]

Complex system behaviors, depending on \( K \)

Simple Controller Goes Digital

The diagram shows a simple control system where \( d_i \) is the desired input, \( d_o \) is the output, and the controller adjusts the input to the plant based on the sensor feedback.

\[ y[n] = y[n-1] - Tu[n-1] \]

\[ sensor: \quad y[n] = u[n-1] \]

\[ controller: \quad y[n] = Ku[n] \]

\[ d_i = \text{desiredFront} \]

\[ d_o = \text{distanceFront} \]
Return to the discrete domain

- Recall that continuous signals can be represented by a series of samples with period $T$

![Graph showing discrete samples of a continuous signal]

Zero Order Hold

- An output value of a synthesised signal is held constant until the next value is ready
  - This introduces an effective delay of $T/2$
How to Handle the Digitization?

(z-Transforms)

The z-transform

• In practice, you’ll use look-up tables or computer tools (ie. Matlab) to find the z-transform of your functions

<table>
<thead>
<tr>
<th>$F(s)$</th>
<th>$F(kt)$</th>
<th>$F(z)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\frac{1}{s}$</td>
<td>1</td>
<td>$\frac{z}{z-1}$</td>
</tr>
<tr>
<td>$\frac{1}{s^2}$</td>
<td>$kT$</td>
<td>$\frac{Tz}{(z-1)^2}$</td>
</tr>
<tr>
<td>$\frac{1}{s+a}$</td>
<td>$e^{-akT}$</td>
<td>$\frac{z}{z-e^{-aT}}$</td>
</tr>
<tr>
<td>$\frac{1}{(s+a)^2}$</td>
<td>$kTe^{-akT}$</td>
<td>$\frac{zTe^{-aT}}{(z-e^{-aT})^2}$</td>
</tr>
<tr>
<td>$\frac{1}{s^2+a^2}$</td>
<td>$\sin(akT)$</td>
<td>$\frac{z \sin aT}{z^2 - (2 \cos aT)z + 1}$</td>
</tr>
</tbody>
</table>
Zero-order-hold (ZOH)

• Assume that the signal \( x(t) \) is zero for \( t < 0 \), then the output \( h(t) \) is related to \( x(t) \) as follows:

\[
h(t) = x(0)[1(t) - 1(t - T)] + x(T)[1(t - T) - 1(t - 2T)] + \cdots
\]

\[
= \sum_{k=0}^{\infty} x(kT)[1(t - kT) - 1(t - (k + 1)T)]
\]

Transfer function of Zero-order-hold (ZOH)

• Recall the Laplace Transforms (\( \mathcal{L} \)) of:

\[
\mathcal{L}[\delta(t)] = 1 \quad \mathcal{L}[f(t - kT)] = F(s)e^{-kTs}
\]

\[
\mathcal{L}[\delta(t - kT)] = e^{-kTs} \quad \mathcal{L}[1(t - kT)] = \frac{e^{-kTs}}{s}
\]

• Thus the \( \mathcal{L} \) of \( h(t) \) becomes:

\[
\mathcal{L}[h(t)] = \mathcal{L}\left[ \sum_{k=0}^{\infty} x(kT)[1(t - kT) - 1(t - (k + 1)T)] \right]
\]

\[
= \sum_{k=0}^{\infty} x(kT)e^{-kTs} - e^{-(k+1)Ts} = \sum_{k=0}^{\infty} x(kT)\frac{e^{-kTs} - e^{-(k+1)Ts}}{s}
\]

\[
= \sum_{k=0}^{\infty} x(kT)e^{-kTs} - \frac{e^{-(k+1)Ts}}{s} = \sum_{k=0}^{\infty} x(kT)e^{-kTs} = \frac{1 - e^{-Ts}}{s} \sum_{k=0}^{\infty} x(kT)e^{-kTs}
\]
Transfer function of Zero-order-hold (ZOH)

... Continuing the $\mathcal{L}$ of $h(t)$ ...

$$\mathcal{L}[h(t)] = \mathcal{L}\left[\sum_{k=0}^{\infty} x(kT)[1(t - kT) - 1(t - (k + 1)T)]\right]$$

$$= \sum_{k=0}^{\infty} x(kT)\frac{e^{-kTs} - e^{-(k+1)Ts}}{s} = \sum_{k=0}^{\infty} x(kT)\frac{1 - e^{-Ts}}{s}e^{-kTs} = \sum_{k=0}^{\infty} x(kT)e^{-kTs}$$

$$\rightarrow X(s) = \mathcal{L}\left[\sum_{k=0}^{\infty} x(kT)\delta(t - kT)\right] = \sum_{k=0}^{\infty} x(kT)e^{-kTs}$$

$$\therefore H(s) = \mathcal{L}[h(t)] = \frac{1 - e^{-Ts}}{s} \sum_{k=0}^{\infty} x(kT)e^{-kTs} = \frac{1 - e^{-Ts}}{s}X(s)$$

$\Rightarrow$ Thus, giving the transfer function as:

$$G_{ZOH}(s) = \frac{H(s)}{X(s)} = \frac{1 - e^{-Ts}}{s} \rightarrow G_{ZOH}(z) = \frac{1 - e^{-aT}}{z - e^{-aT}}$$

$L(ZOH)=??? :$ What is it?

- Wikipedia
- Lathi
- Franklin, Powell, Workman
- Franklin, Powell, Emani-Naeini
- Dorf & Bishop
- Oxford Discrete Systems: (Mark Cannon)
- MIT 6.002 (Russ Tedrake)
- Matlab
- Proof!
Coping with complexity

- Transfer functions help control complexity
  - Recall the Laplace transform:
    \[ \mathcal{L}\{f(t)\} = \int_0^\infty f(t)e^{-st} \, dt = F(s) \]
    where
    \[ \mathcal{L}\{f'(t)\} = sF(s) \]

\[ x(t) \rightarrow H(s) \rightarrow y(t) \]

Is there a something similar for sampled systems?

The \( z \)-transform

- The discrete equivalent is the \( z \)-Transform\(^\dagger\):
  \[ \mathcal{Z}\{f(k)\} = \sum_{k=0}^\infty f(k)z^{-k} = F(z) \]
  and
  \[ \mathcal{Z}\{f(k-1)\} = z^{-1}F(z) \]

\[ x(k) \rightarrow F(z) \rightarrow y(k) \]

Convenient!

\(^\dagger\)This is not an approximation, but approximations are easier to derive
The z-transform

- Some useful properties
  - **Delay by** \( n \) **samples**: \( \mathcal{Z}\{f(k-n)\} = z^{-n}F(z) \)
  - **Linear**: \( \mathcal{Z}\{af(k) + bg(k)\} = aF(z) + bG(z) \)
  - **Convolution**: \( \mathcal{Z}\{f(k) * g(k)\} = F(z)G(z) \)

So, all those block diagram manipulation tools you know and love will work just the same!

---

The z-transform

- In practice, you’ll use look-up tables or computer tools (ie. Matlab) to find the z-transform of your functions

<table>
<thead>
<tr>
<th>( F(s) )</th>
<th>( F(kt) )</th>
<th>( F(z) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \frac{1}{s} )</td>
<td>1</td>
<td>( \frac{z}{z-1} )</td>
</tr>
<tr>
<td>( \frac{1}{s^2} )</td>
<td>( kT )</td>
<td>( \frac{Tz}{(z-1)^2} )</td>
</tr>
<tr>
<td>( \frac{1}{s+a} )</td>
<td>( e^{-akT} )</td>
<td>( \frac{z}{z-e^{-aT}} )</td>
</tr>
<tr>
<td>( \frac{1}{(s+a)^2} )</td>
<td>( kTe^{-akT} )</td>
<td>( \frac{zTe^{-aT}}{(z-e^{-aT})^2} )</td>
</tr>
<tr>
<td>( \frac{1}{s^2+a^2} )</td>
<td>( \sin(akT) )</td>
<td>( \frac{z \sin aT}{z^2 - (2 \cos aT)z + 1} )</td>
</tr>
</tbody>
</table>
Why $z$-Transform

- Makes it easy to analyse feedback systems governed by difference equations
- For any complex number $z = re^{j\omega}$, $y[n] \leftrightarrow Y(z)$

- Forward Analysis: $Y(z) = \sum_{k=-\infty}^{\infty} h[k]z^{-k}$

- Backward Synthesis
  (for any fixed $r > 0$ on which the Z-transform converges):

$$y[n] = \frac{1}{2\pi} \int_{2\pi} \frac{Y(re^{j\omega})}{(re^{j\omega})^n} d\omega$$

---

$z$-Transforms for Difference Equations

- First-order linear constant coefficient difference equation:

First-order linear constant coefficient difference equation:

$$y[n] = ay[n-1] + bu[n]$$

$h[n]$:

$$h[n] = \begin{cases} 
  b a^n & \text{n} \geq 0, \\
  0 & \text{otherwise.}
\end{cases}$$

$$H(z) = \sum_{k=0}^{\infty} b a^k z^{-k} = b \sum_{k=0}^{\infty} \left(\frac{a}{z}\right)^k = \frac{b}{1 - a z^{-1}}, \quad \text{when } |z| > |a|.$$
z-Transforms for Difference Equations

First-order linear constant coefficient difference equation:

\[ y[n] = ay[n-1] + bu[n] \]

\[ y[n] - ay[n-1] = bu[n] \]

\[ Y(z) - az^{-1}Y(z) = bU(z) \]

\[ H(z) = \frac{Y(z)}{U(z)} = \frac{b}{1 - az^{-1}} \]

when does it converge?

Digitisation

- Continuous signals sampled with period \( T \)
- \( k \)th control value computed at \( t_k = kT \)
Digitisation

- Continuous signals sampled with period $T$
- $k$th control value computed at $t_k = kT$

![Diagram of digitisation process]

Difference equations

- How to represent differential equations in a computer? Difference equations!
- The output of a difference equation system is a function of current and previous values of the input and output:

$$y(t_k) = D(x(t_k), x(t_{k-1}), ..., x(t_{k-n}), y(t_{k-1}), ..., y(t_{k-n}))$$

  - We can think of $x$ and $y$ as parameterised in $k$
  
  Useful shorthand: $x(t_{k+i}) \equiv x(k + i)$
Euler’s method*

- Dynamic systems can be approximated† by recognising that:

\[
\dot{x} \approx \frac{x(k + 1) - x(k)}{T}
\]

- As \( T \rightarrow 0 \), approximation error approaches 0

*Also known as the forward rectangle rule
†Just an approximation – more on this later

An example!

Convert the system \( \frac{Y(s)}{X(s)} = \frac{s+2}{s+1} \) into a difference equation with period \( T \), using Euler’s method.

1. Rewrite the function as a dynamic system:

\[
sY(s) + Y(s) = sX(s) + 2X(s)
\]

Apply inverse Laplace transform:

\[
\dot{y}(t) + y(t) = \dot{x}(t) + 2x(t)
\]

2. Replace continuous signals with their sampled domain equivalents, and differentials with the approximating function

\[
\frac{y(k + 1) - y(k)}{T} + y(k) = \frac{x(k + 1) - x(k)}{T} + 2x(k)
\]
An example!

Simplify:

\[ y(k + 1) - y(k) + Ty(k) = x(k + 1) - x(k) + 2Tx(k) \]
\[ y(k + 1) + (T - 1)y(k) = x(k + 1) + (2T - 1)x(k) \]
\[ y(k + 1) = x(k + 1) + (2T - 1)x(k) - (T - 1)y(k) \]

We can implement this in a computer.

Cool, let’s try it!

Back to the future

A quick note on causality:

- Calculating the “\((k+1)th\)” value of a signal using
  \[ y(k + 1) = x(k + 1) + Ax(k) - By(k) \]
  relies on also knowing the next (future) value of \(x(t)\). (this requires very advanced technology!)

- Real systems always run with a delay:
  \[ y(k) = x(k) + Ax(k - 1) - By(k - 1) \]
Region of Convergence (ROC) Plots

\[ H(z) = \frac{Y(z)}{U(z)} = \frac{b}{1 - az^{-1}}, \quad |z| > |a| \]

\[ a = 0.5 \quad \text{and} \quad a = 1.2 \]
Properties of the ROC

- The ROC is always defined by circles centered around the origin.
  \[ h[k]r^{-k} \text{ is absolutely summable, where } r = |z|. \]

- Right-sided signals have “outsided” ROCs.

  if \( \exists n_0 \) such that \( h[n] = 0 \forall n < n_0 \), then if \( r_0 \in \text{ROC} \), then \( \forall r \) with \( r_0 < r < \infty \) are also in the ROC.

- Left-sided signals have “insided” ROCs.

  (with \( \forall r \) within \( 0 < r < r_0 \))

Combinations of Signals

\[ y_1[n] = \begin{cases} ba^n & n \geq 0 \\ 0 & n < 0 \end{cases} \quad y_2[n] = \begin{cases} 0 & n \geq 0 \\ -ba^n & n < 0 \end{cases} \]

- \( a = 0.5 \)
- \( a = 2 \)

ROC for \( \alpha_1 y_1[n] + \alpha_2 y_2[n] \)
Higher-order difference equations

\[ y[n] = a_1 y[n-1] + a_2 y[n-2] + a_3 y[n-3] + b_0 u[n] + b_1 u[n-1] + \ldots \]

Easy to take the Z-transform

\[ Y(z) = a_1 z^{-1} Y(z) + a_2 z^{-2} Y(z) + a_3 z^{-3} Y(z) + b_0 U(z) + \ldots \]

\[ H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \ldots}{1 - a_1 z^{-1} - a_2 z^{-2} - a_3 z^{-3} + \ldots} \]

---

Final value theorem

- An important question: what is the steady-state output a stable system at \( t = \infty \)?
  - For continuous systems, this is found by:
    \[ \lim_{t \to \infty} x(t) = \lim_{s \to 0} sX(s) \]
  - The discrete equivalent is:
    \[ \lim_{k \to \infty} x(k) = \lim_{z \to 1} (1 - z^{-1}) X(z) \]
    (Provided the system is stable)
An example!

- Back to our difference equation:
  \[ y(k) = x(k) + Ax(k - 1) - By(k - 1) \]

  becomes
  \[ Y(z) = X(z) + Az^{-1}X(z) - Bz^{-1}Y(z) \]
  \[ (z + B)Y(z) = (z + A)X(z) \]

  which yields the transfer function:
  \[ \frac{Y(z)}{X(z)} = \frac{z + A}{z + B} \]

Note: It is also not uncommon to see systems expressed as polynomials in \( z^{-n} \)

This looks familiar…

- Compare:
  \[ \frac{Y(s)}{X(s)} = \frac{s + 2}{s + 1} \]
  \[ \frac{Y(z)}{X(z)} = \frac{z + A}{z + B} \]

  How are the Laplace and \( z \) domain representations related?
Consider the simplest system

- Take a first-order response:
  \[ f(t) = e^{-at} \Rightarrow \mathcal{L}\{f(t)\} = \frac{1}{s + a} \]
- The discrete version is:
  \[ f(kT) = e^{-akT} \Rightarrow Z\{f(k)\} = \frac{z}{z - e^{-aT}} \]

The equivalent system poles are related by
\[ z = e^{sT} \]

That sounds somewhat profound… but what does it mean?

---

The \( z \)-Plane

- \( z \)-domain poles and zeros can be plotted just like \( s \)-domain poles and zeros:

![Diagram of \( s \)-Plane and \( z \)-Plane](image@example.com)
Deep insight #1

The mapping between continuous and discrete poles and zeros acts like a distortion of the plane.

\[ \text{max frequency} \]

The \( z \)-plane

- We can understand system response by pole location in the \( z \)-plane.

[Adapted from Franklin, Powell and Emami-Naeini]
Effect of pole positions

- We can understand system response by pole location in the $z$-plane

Increasing frequency

Most like the s-plane
Effect of pole positions

- We can understand system response by pole location in the $z$-plane

\[
z = e^{sT} \text{ where } s = -\zeta \omega_n \pm j \omega_n \sqrt{1 - \zeta^2}
\]
z-plane stability

- In the z-domain, the unit circle is the system stability bound.
z-plane stability

- The z-plane root-locus in closed loop feedback behaves just like the s-plane:

Deep insight #2

Gains that stabilise continuous systems can actually destabilise digital systems!
Example:

- Is this system stable?
  \[ u(k) = 0.9u(k-1) - 0.2u(k-2) \]

- Time-shift it:
  \[ u(k+2) = 0.9u(k+1) - 0.2u(k) \]

- z-Transform:
  \[ (1)z^2 - 0.9z + 0.2 = 0 \]

- Characteristic Roots:
  \[ z=0.5, z=0.4 \Rightarrow \text{STABLE!} \]