Time Domain **Analysis** of Continuous Time Systems

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

Lecture 7

**elec3004@itee.uq.edu.au**  
**http://robotics.itee.uq.edu.au/~elec3004/**  
March 20, 2013

**Announcements:**

- Assignment 1 is up on Platypus!  
  - Due at the end of next week!  
  - Next week’s tutorial sessions are Q&A meetings

Lab 1:  
- Don’t forget the Pre-Lab  
- Extra Session Needed ????

- Space Audit this **Friday** and next **Wednesday**  
  - Sounds like an ideal time for a **pop-quiz**
### Today:

<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>2</td>
<td>3-Mar</td>
<td>Systems Overview</td>
</tr>
<tr>
<td>3</td>
<td>8-Mar</td>
<td>Linear Dynamical Systems</td>
</tr>
<tr>
<td>4</td>
<td>13-Mar</td>
<td>Signals &amp; Signal Models</td>
</tr>
<tr>
<td>5</td>
<td>15-Mar</td>
<td>System Models</td>
</tr>
<tr>
<td>6</td>
<td>20-Mar</td>
<td>Time Domain Analysis of Continuous Time Systems</td>
</tr>
<tr>
<td>7</td>
<td>22-Mar</td>
<td>System Behaviour &amp; Stability</td>
</tr>
<tr>
<td>8</td>
<td>27-Mar</td>
<td>Signal Representation</td>
</tr>
<tr>
<td>9</td>
<td>29-Mar</td>
<td>Holiday</td>
</tr>
<tr>
<td>10</td>
<td>6-Apr</td>
<td>Frequency Response &amp; Fourier Transform</td>
</tr>
<tr>
<td>11</td>
<td>10-Apr</td>
<td>Analog Filters</td>
</tr>
<tr>
<td>12</td>
<td>17-Apr</td>
<td>LTI Systems</td>
</tr>
<tr>
<td>13</td>
<td>19-Apr</td>
<td>Analog Filters</td>
</tr>
<tr>
<td>14</td>
<td>24-Apr</td>
<td>2-Dimensional Signals</td>
</tr>
<tr>
<td>15</td>
<td>1-May</td>
<td>Discrete-Time Signals</td>
</tr>
<tr>
<td>16</td>
<td>3-May</td>
<td>Discrete-Time Systems</td>
</tr>
<tr>
<td>17</td>
<td>10-May</td>
<td>State-Space</td>
</tr>
<tr>
<td>18</td>
<td>12-May</td>
<td>Controllability &amp; Observability</td>
</tr>
<tr>
<td>19</td>
<td>17-May</td>
<td>Introduction to Digital Control</td>
</tr>
<tr>
<td>20</td>
<td>19-May</td>
<td>Stability of Digital Systems</td>
</tr>
<tr>
<td>21</td>
<td>22-May</td>
<td>PID &amp; Computer Control</td>
</tr>
<tr>
<td>22</td>
<td>24-May</td>
<td>Information Theory &amp; Communications</td>
</tr>
<tr>
<td>23</td>
<td>29-May</td>
<td>Applications In Industry</td>
</tr>
<tr>
<td>24</td>
<td>31-May</td>
<td>Summary and Course Review</td>
</tr>
</tbody>
</table>

### Convolution & Properties

**Convolution:**

\[
f_1(t) * f_2(t) \equiv \int_{-\infty}^{\infty} f_1(\tau)f_2(t-\tau) \, d\tau
\]

**Properties:**

- **Commutative:** \( f_1(t) * f_2(t) = f_2(t) * f_1(t) \)
- **Distributive:** \( f_1(t) * [f_2(t) + f_3(t)] = f_1(t) * f_2(t) + f_1(t) * f_3(t) \)
- **Associative:** \( f_1(t) * [f_2(t) * f_3(t)] = [f_1(t) * f_2(t)] * f_3(t) \)
- **Shift:**
  
  if \( f_1(t) * f_2(t) = c(t) \), then \( f_1(t-T) * f_2(t) = f_1(t-T) * f_2(t) = c(t-T) \)
- **Identity (Convolution with an Impulse):**
  
  \( f(t) * \delta(t) = f(t) \)
- **Total Width:**
  
  ![Impulse Response Diagram](image)

Based on Lathi, SPS, Sec 2.4-1
Convolution & Properties [II]

- Convolution systems are \textbf{linear}:
  \[ h \ast (\alpha u_1 + \beta u_2) = \alpha(h \ast u_1) + \beta(h \ast u_2) \]

- Convolution systems are \textbf{causal}: the output \( y(t) \) at time \( t \) depends only on past inputs

- Convolution systems are \textbf{time-invariant}
  (if we shift the signal, the output similarly shifts)

\[ \tilde{u}(t) = \begin{cases} 0 & t < T \\ \omega(t-T) & t \geq 0 \end{cases} \]
\[ \tilde{y}(t) = \begin{cases} 0 & t < T \\ y(t-T) & t \geq 0 \end{cases} \]

Convolution & Properties [III]

- Composition of convolution systems corresponds to:
  - multiplication of transfer functions
  - convolution of impulse responses

\begin{center}
\begin{tikzpicture}
  \node[draw] (A) at (0,0) {$A$};
  \node[draw] (B) at (2,0) {$B$};
  \node (u) at (-1,0) {$u$};
  \node (y) at (3,0) {$y$};
  \draw[->] (u) -- (A);
  \draw[->] (A) -- (B);
  \draw[->] (B) -- (y);
  \node[draw] (BA) at (1,0) {$BA$};
  \node (u2) at (-1,0) {$u$};
  \node (y2) at (3,0) {$y$};
  \draw[->] (u2) -- (BA);
  \draw[->] (BA) -- (y2);
\end{tikzpicture}
\end{center}

- Thus:
  - We can manipulate block diagrams with transfer functions as if they were simple gains
  - convolution systems commute with each other
Convolution & Systems

• Convolution system with input $u$ ($u(t) = 0$, $t < 0$) and output $y$:
  
  $$ y(t) = \int_0^t h(\tau)u(t-\tau) \, d\tau = \int_0^t h(t-\tau)u(\tau) \, d\tau $$

• abbreviated:
  
  $$ y = h * u $$

• in the frequency domain:
  
  $$ Y(s) = H(s)U(s) $$

Convolution & Feedback

• In the time domain:
  
  $$ y(t) = \int_0^t g(t-%C5%AE-%C3%89\tau)(u(\tau) - y(\tau)) \, d\tau $$

• In the frequency domain:
  
  $$ Y(s) = G(s)(U-Y) $$
  
  $$ \Rightarrow Y(s) = H(s)U(s) $$
  
  $$ H(s) = \frac{G(s)}{1 + G(s)} $$
Graphical Understanding of Convolution

For $c(\tau) = f * g(t) = \int_{-\infty}^{\infty} f(\tau) g(t-\tau) \, d\tau$:

1. Keep the function $f(\tau)$ fixed
2. Flip (invert) the function $g(\tau)$ about the vertical axis ($\tau=0$)
   = this is $g(-\tau)$
3. Shift this frame ($g(-\tau)$) along $\tau$ (horizontal axis) by $t_0$.
   = this is $g(t_0-\tau)$

For $c(t_0)$:
4. $c(t_0) =$ the area under the product of $f(\tau)$ and $g(t_0-\tau)$

5. Repeat this procedure, shifting the frame by different values (positive and negative) to obtain $c(t)$ for all values of $t$.

Graphical Understanding of Convolution (Ex)
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 Locus Diagram]

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

The Root Locus

- We often care about the effect of increasing gain of a control compensator design:

\[
\frac{y}{r} = \frac{kCH}{1 + kCH}
\]

Multiplying by denominator:

\[
\frac{y}{r} = \frac{kC_nH_n}{C_dH_d + kCnHn}
\]

![Root Locus Diagram with Control System]
The root locus

- Pole positions change with increasing gain
  - The trajectory of poles on the pole-zero plot with changing $k$ is called the “root locus”
  - This is sometimes quite complex

- (In practice you’d plot these with computers)

Root Locus Drawing Rules

1. The root locus is symmetric with respect to the real axis.
2. The root loci start from $n$ poles $p_i$ (when $K = 0$) and approach the $n$ zeros ($m$ finite zeros $z_j$ and $n - m$ infinite zeros when $K \to \infty$).
3. The root locus includes all points on the real axis to the left of an odd number of open-loop real poles and zeros.
4. As $K \to \infty$, $n - m$ branches of the root-locus approach asymptotically $n - m$ straight lines (called asymptotes) with angles

$$\theta = \frac{(2k + 1)180^\circ}{n - m}, \quad k = 0, \pm 1, \pm 2, \ldots$$

and the starting point of all asymptotes is on the real axis at

$$\kappa = \frac{\sum_{i=1}^{n} p_i - \sum_{j=1}^{m} z_j}{n - m} = \frac{\sum \text{poles} - \sum \text{zeros}}{n - m}.$$
5. The **breakaway points** (where the root loci meet and split away, usually on real axis) and the **breakin points** (where the root loci meet and enter the real axis) are among the roots of the equation: $\frac{dK(s)}{ds} = 0$. (On the real axis, only those roots that satisfy Rule 3 are breakaway or breakin points.)

6. The **departure angle** $\phi_k$ (from a pole, $p_k$) is given by

$$\phi_k = \sum_{i=1}^{m} \angle(p_k - z_i) - \sum_{j=1, j \neq k}^{n} \angle(p_k - p_j) \pm 180^\circ.$$ 

(In the case $p_k$ is $l$ repeated poles, the departure angle becomes $\phi_k/l$.)

The **arrival angle** $\psi_k$ (at a zero, $z_k$) is given by

$$\psi_k = - \sum_{i=1, i \neq k}^{m} \angle(z_k - z_i) + \sum_{j=1}^{n} \angle(z_k - p_j) \pm 180^\circ.$$ 

(In the case $z_k$ is $l$ repeated zeros, the arrival angle becomes $\psi_k/l$.)

---

**TABLE 5.1: Root locus rules: $0 \leq K \leq \infty$.**

---

**Recall (Lecture 5, Slide 10):**

**Linear Dynamic [Differential] System**

$\equiv$ LTI systems for which the input & output are linear ODEs

$$a_0 y + a_1 \frac{dy}{dt} + \cdots + a_n \frac{d^n y}{dt^n} = b_0 x + b_1 \frac{dx}{dt} + \cdots + b_m \frac{d^m x}{dt^m}$$

Laplace:

$$a_0 Y(s) + a_1 s Y(s) + \cdots + a_n s^n Y(s) = b_0 X(s) + b_1 s X(s) + \cdots + b_m s^m X(s)$$

$$A(s) Y(s) = B(s) X(s)$$

- **Total response** $=$ **Zero-input response** $+$ **Zero-state response**

**Initial conditions**

**External Input**
Recall: Second Order Systems

Second order systems

\[ ay'' + by' + cy = 0 \]

assume \( a > 0 \) (otherwise multiply equation by \(-1\))

solution by Laplace transform:

\[ a(s^2Y(s) - ay(0) - y'(0)) + b(sY(s) - y(0)) + cY(s) = 0 \]

solve for \( Y \) (just algebra)

\[ Y(s) = \frac{asy(0) + ay'(0) + by(0)}{as^2 + bs + c} = \frac{\alpha s + \beta}{as^2 + bs + c} \]

where \( \alpha = ay(0) \) and \( \beta = ay'(0) + by(0) \)

Second Order Systems

so solution of \( ay'' + by' + cy = 0 \) is

\[ y(t) = \mathcal{L}^{-1} \left( \frac{\alpha s + \beta}{as^2 + bs + c} \right) \]

- \( \chi(s) = as^2 + bs + c \) is called characteristic polynomial of the system
- form of \( y = \mathcal{L}^{-1}(Y) \) depends on roots of characteristic polynomial \( \chi \)
- coefficients of numerator \( \alpha s + \beta \) come from initial conditions
Second Order Response

Three Types:
• I: Underdamped: \((0 < \xi < 1)\):

\[
C(t) = \frac{R(s)}{R(s)} = \frac{s^2 + \xi \omega_n s + \omega_n^2}{s(s + \xi \omega_n + j \omega_c)} = s + \omega_n \sqrt{1 - \xi^2}, \]

\[
X(t) = c(t) = e^{-\xi \omega_n t} \left[ \cos(\omega_d t) + \frac{\xi}{\sqrt{1 - \xi^2}} \sin(\omega_d t) \right],
\]

\[
= 1 - e^{-\omega_c t} \sin\left(\omega_d t + \tan^{-1}\left(\frac{\xi}{\sqrt{1 - \xi^2}}\right)\right).
\]

Second Order Response

Three Types:
• II: Critically Damped: \((\xi = 1)\):

For a unit-step input, \(R(s) = 1/s\) and \(C(s)\) can be written

\[
C(s) = \frac{\omega_d^2}{s(s + \omega_d)^2}
\]

\[
|ln\left(\frac{\sin(\omega_d t)}{\sqrt{1 - \xi^2}}\right)| = \left|ln\left(\frac{\sin(\omega_d \sqrt{1 - \xi^2})}{\sqrt{1 - \xi^2}}\right)\right| = \omega_c t.
\]
Three Types:
- III: Over Damped: \((\zeta > 1)\)

For a unit-step input, \(R(s) = 1/s\) and \(C(s)\) can be written

\[
C(s) = \frac{\omega_n}{(s + \zeta \omega_n + \omega_n^2 - 1)(s + (\omega_n - \omega_n \sqrt{\zeta^2 - 1}))}
\]
Second Order Response
Envelope Curves

• Delay time, \( t_d \): The time required for the response to reach half the final value
• Rise time, \( t_r \): The time required for the response to rise from 10% to 90%
• Peak time, \( t_p \): The time required for the response to reach the first peak of the overshoot
• Maximum (percent) overshoot, \( M_p \):
  \[
  M_p = \frac{c(t_f) - c(\infty)}{c(\infty)} \times 100\%
  \]
• Settling time, \( t_s \): The time to be within 2-5% of the final value

Second Order Response
Unit Step Response Terms
Second Order Response
Seeing this on the S-plane

- The addition of a zero (an \( s \) term) gives a system with a shorter rise time, a shorter peak time, and a larger overshoot.

Second Order Response
The Case of **Adding a Zero**

- Increasing \( \tau \)
Second Order Response
The Case of Adding a Zero

- The addition of a pole (a $1/s$ term) slows down the system response and reduces the overshoot.

Example: Quarter-Car Model
Example: Quarter-Car Model (2)

\[
\begin{align*}
\ddot{x} + \frac{b}{m_1}(\dot{x} - \dot{y}) + \frac{k_1}{m_1}(x - y) + \frac{k_2}{m_1}x &= \frac{k_o}{m_1}r_s, \\
\ddot{y} + \frac{b}{m_2}(\dot{y} - \dot{x}) + \frac{k_1}{m_1}(y - x) &= 0.
\end{align*}
\]

\[
\begin{align*}
x^2X(s) + \frac{b}{m_1}(X(s) - Y(s)) + \frac{k_1}{m_1}(X(s) - Y(s)) + \frac{k_2}{m_1}X(s) &= \frac{k_o}{m_1}R(s), \\
x^2Y(s) + \frac{b}{m_2}(Y(s) - K(s)) + \frac{k_1}{m_1}(Y(s) - X(s)) &= 0.
\end{align*}
\]

\[
\begin{align*}
Y(s) &= \frac{\frac{k_o b}{m_1 m_2} \left( s^2 + \frac{k_o}{m_1} \right)}{s^4 + \left( \frac{k_o}{m_1} + \frac{k_o}{m_2} \right) s^2 + \left( \frac{k_o}{m_1} + \frac{k_o}{m_2} \right) s + \frac{k_o k_2}{m_1 m_2}},
\end{align*}
\]

Next Time…

- Stability
  - A performance measure which informs the extent to which all the poles of the transfer function have negative real parts
  - Aka:
  - Attempts to spontaneously disassemble itself

or

- Review:
  - Section 3.10 of Lathi
"Back to the Future": Laplace Review!

Recall dynamic responses

- Moving pole positions change system response characteristics

Faster

More Oscillatory

More damped

Pure integrator

“More unstable”
Dynamic compensation

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

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

\[
\begin{align*}
\text{combined system} & \\
\begin{array}{c}
 s+2 \\
\end{array} & \quad y \\
\begin{array}{c}
 s(s+1) \\
\end{array} & \quad y \\
\end{align*}
\]

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)