SFWR ENG 3DX4 Summary

Instructor: Dr. Lawford
Course: SFWR ENG 3DX4

Math objects made using MathType; graphs made using Winplot.

Table of Contents

Note: the following summaries may be useful:

- SFWR ENG 2MX3
- ENGINEER 3N03
- TRON 3TA4

I may review to clarify or correct, but mostly I will omit those things.

Introduction to Systems

Systems can be represented by **block diagrams** to make it easier to marginalize the different parts of the systems.

Laplace

Useful for…

Time begins when your signal begins

$$h(t) = \begin{cases} 0, & t < 0 \\ 1, & t \geq 0 \end{cases}$$

Initial conditions:

- $c(0)$

Time domain (t): variables are lower case, e.g. $f(t)$
Frequency domain (s): variables are upper case, e.g. $F(s)$

Transfer function:

When doing the inverse Laplace, it's useful to break your fractions up so that you can

Strictly Stable: it will eventually get back to the initial position
Marginally Stable:
Unstable: it will progressively get worse

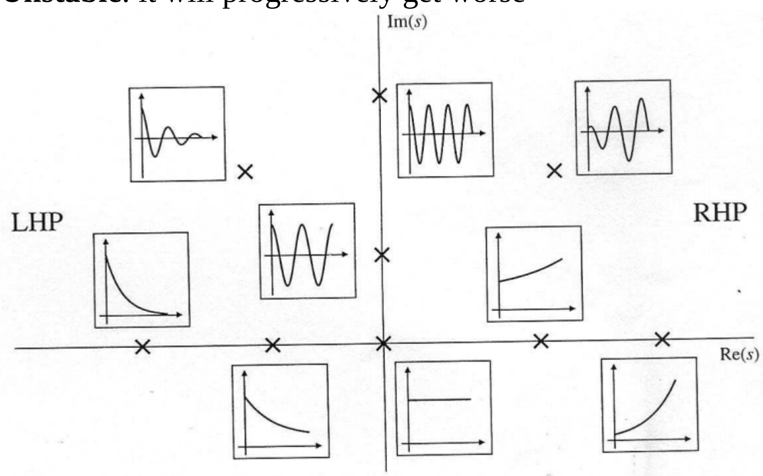


Figure 2.5 from Dorf and Bishop, *Modern Control Systems (10th Edition)*, Prentice-Hall, 2004.

Transfer Functions

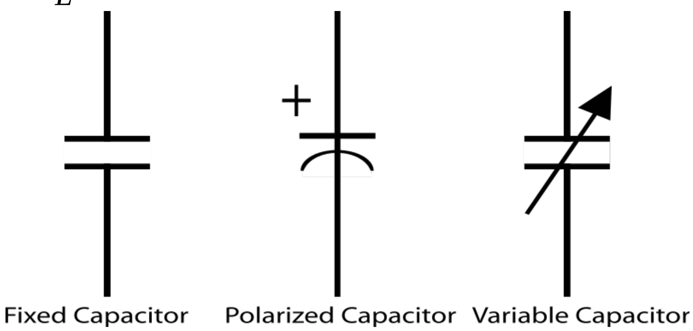
Electrical

Component stuff

$$i = VR$$

$$i = C\frac{\mathrm{d}v}{\mathrm{d}t}$$

$$i = \frac{1}{L}\int_0^t v\mathrm{d}t$$

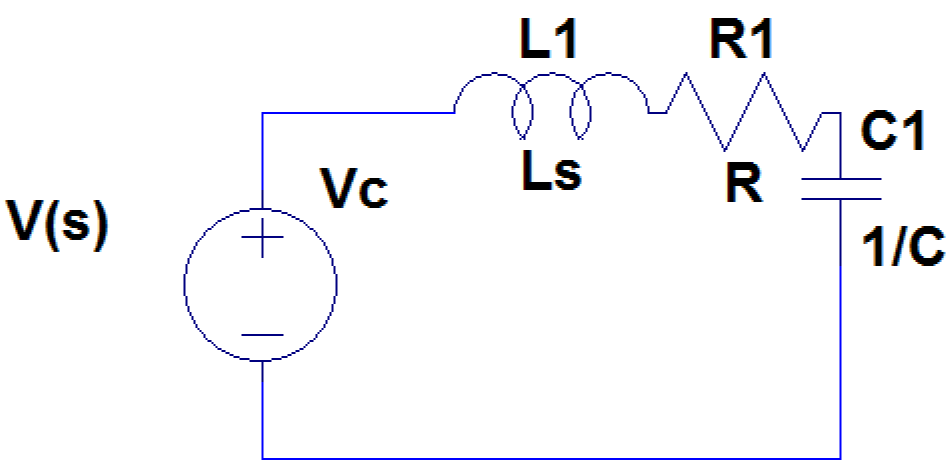


admittance:

$$Y(s) = \frac{I(s)}{V(s)} = \frac{1}{R} = G$$

$$V_c(s) = I(s)\frac{1}{Cs}$$

$$I(s) = \frac{V(s)}{L_s + R + \frac{1}{Cs}}$$



Mesh Analysis

You cannot use Ohm's law to find the current through a voltage source, so represent the current by $i_{something}$, like i_x.

Cramer's Rule

$$x_1 = \frac{\det(A_1)}{\det(A)}, x_2 = \frac{\det(A_2)}{\det(A)}, \ldots, x_n = \frac{\det(A_n)}{\det(A)}$$

$$V_C(s) = \overbrace{H(s)}^{\substack{\text{transfer}\\\text{function}}} \frac{1}{Cs}$$

OP-Amps

Mechanical

Translational systems:
Rotational Systems:

Newton's Second Law of Motion: $\Sigma f = Ma$

$$Z_m(s) = \frac{F(s)}{X(x)}$$

$$\begin{aligned} f(t) &= Ma(t) \\ &= M\frac{\mathrm{d}^2x}{\mathrm{d}t^2} \end{aligned}$$

Translational Systems

Spring

Spring is like a capacitor

Force displacement: $f(t) = Kx(t)$

Viscous Damper

Using viscous fluid to slow something down

Viscous Damper is like a resistor

Force displacement: $f(t) = f_v \frac{\mathrm{d}x(t)}{\mathrm{d}t}$

Mass

Mass is like a inductor

Force displacement: $f(t) = M\frac{\mathrm{d}^2x(t)}{\mathrm{d}t^2}$

Rotational Systems

Transducer: anything that converts energy to electrical energy
Transmitter: long distances

Unstable systems have ∞ steady state error
Steady-state error [e_∞]:

$$e_\infty = \lim_{t\to\infty} e(t)$$

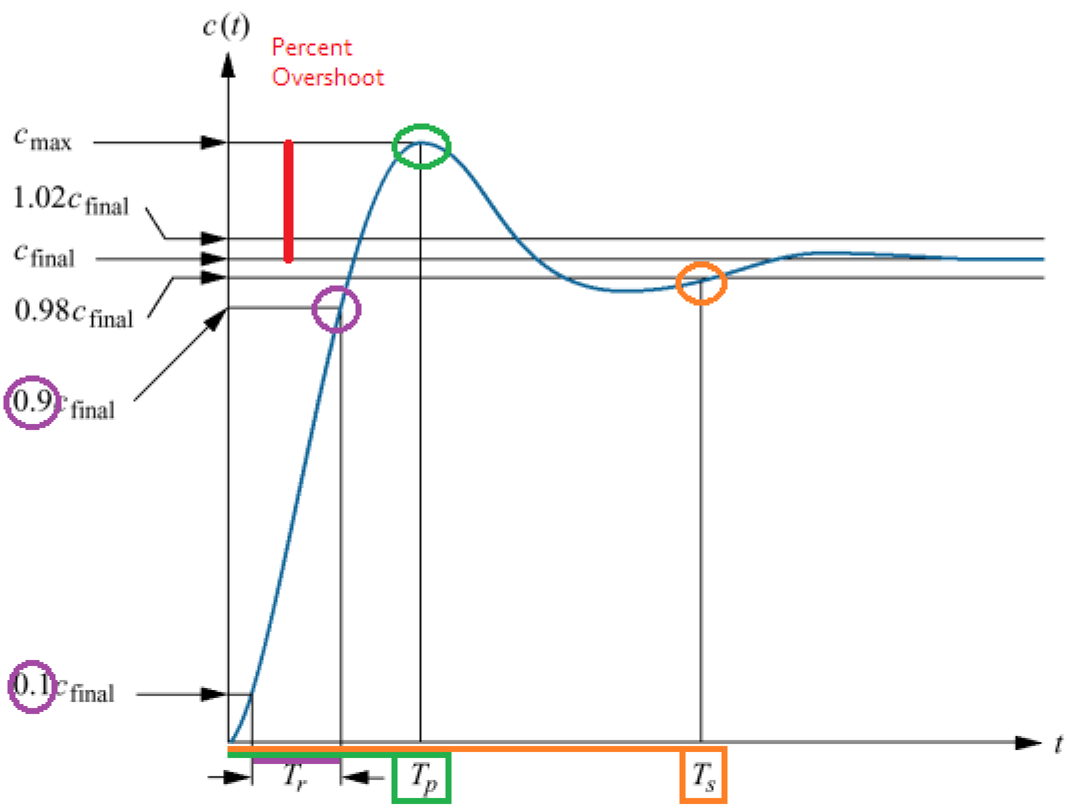
Final value theorem: finds steady state error

$$\lim_{x\to\infty} f(t) = \lim_{x\to 0} sF(s)$$

So $e_\infty = \lim_{x\to 0} sF(s)$ and you're given F(s), so just multiply by s and find the limit.

Rise time [T_r]: time between 10% and 90% of final value (c_{final})
Peak time [T_p]: time it takes to get to highest peak ($c_{\max}$)
Settling time [T_s]: how long it takes to get to the steady state within ±2%
Percent overshoot [%OS]: how much further is the peak from the final

$$\%\text{OS} = \frac{c_{\max} - c_{\text{final}}}{c_{\text{final}}} \times 100\%$$



Non-/Linear Systems

- Op Amps are linear

- If you don't have enough voltage, your motor magnets won't have enough power to switch poles, so they require a minimum voltage

You can't model non-linear systems, until you linearize it. To do this, we find the slope and approximate the equation of the line, using y=mx+b

Proportional-Integral-Derivative (PID):
If your gears are vibrating, your PID is probably too high

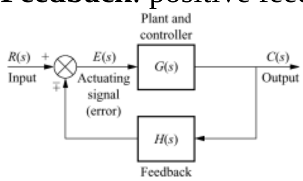
Block Diagrams

A way of representing a system

Summing junction: could be an X or +, but usually an X in this course

Cascade: subsystems in series are multiplied
Parallel: parallel subsystems have a *summing junction* at the end, so you just add everything together

Feedback: positive feedback is bad



Positive: $\dfrac{G(s)}{1-G(s)H(s)}$

Negative: $\dfrac{G(s)}{1+G(s)H(s)}$

Simplification:

State Space Equations

Yeah, you think you know them from 2MX3, but you don't really know them. Apparently the ABCD variables actually have names.

- **System Matrix [A]**:
- **Input Matrix [B]**:
- **Output Matrix [C]**:
- **Feedforward Matrix [D]**:

Transfer Function -> State Space

Phase Variable Approach:

The *n* state variables will consist of:

- *y*
- the derivatives of *y*