An Overview of Linear Systems

The content from this course was hosted on TechOnline.com from 1999 - 2004. TechOnline.com is now targeting commercial clients, so the content, (without animation and voice) is now being hosted here.

Description

This course provides an introduction to linear systems with a view towards modeling, simulation, filtering, and control system design. The material introduces linear, time-invariant systems that can be modeled with ordinary, constant coefficient, differential equations. The Laplace transform and transfer functions are used to simply the analysis. Bode and Nyquist plots are used to present the system frequency response.

Module List:

1) Modeling of continuous time invariant linear systems
2) Analysis of linear systems, state space representation, numerical simulation
3) Analysis of linear systems, control system design and synthesis
4) Implementation of control systems, discretization, z-transforms.

Author: Duane Mattern

Background:

Duane Mattern is an independent contractor specializing in modeling, simulation, control system design and implementation. He is experienced with rapid prototyping software tools like the Mathwork's Matlab/Simulink/Controls/RTW and Integrated System's Xmath/MatrixX/ SystemBuild/Autocode. As a mechanical engineer specializing in instrumentation and controls with more than 10 years of experience, he has a broad range of practical knowledge, including automatic testing machines, turbofan engine control, integrated flight and propulsion control, servo-systems including voice coil and electromagnetic actuation, diagnostics, and neural networks. His current interests are in modeling, real-time simulation, control system design and embedded system programming for control system implementation,

Prerequisites:

Familiarity with the following concepts: (i) phasor notation and the fundamentals of complex variables; (ii) integration and differentiation; (iii) superposition, the Laplace transform and transfer functions; (iv) frequency response using Bode and Nyquist plots, (v) basic linear algebra.

Intended Audience:

(1) Engineer or practitioner who would like to renew their knowledge of linear systems;
(2) Engineer or practitioners who would like a fast introduction to linear systems;
(3) College student who desires an alternative presentation to linear systems, separate from what they receive in their normal courses.

Estimated Total Learning Time:

2 hours

Module 11: Discrete Implementation of linear controllers

- Purpose
  - To discuss the implementation of a linear control design on an example problem.
  - To discuss the use of fixed point integer mathematics and scaling.

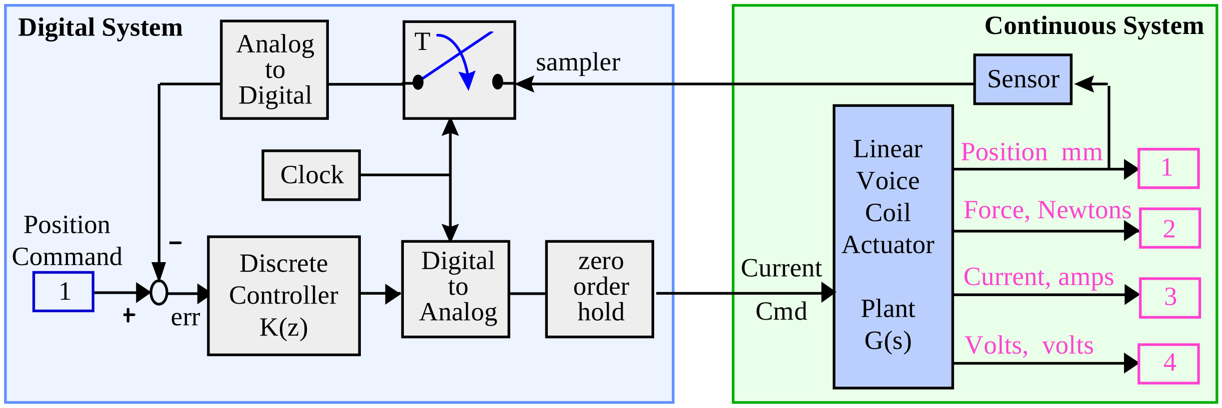
  Objective:
  - Become comfortable with discrete systems and difference equations.
  - Understand discrete time systems and z-domain transfer functions.
  - Gain an understanding of the need for fixed point integer mathematics and scaling.
  - Gain an appreciation for process of implementing a control system.

- Contents: 9 pages
  1 test questions

- Learning Time: 20 minutes

This module covers the implementation of the discrete control system obtained from a the continuous control design. The controller is in the form of a SISO transfer function. Upon completion, you will be able to accomplish the objective listed here. Click the Forward arrow when you're ready to continue.

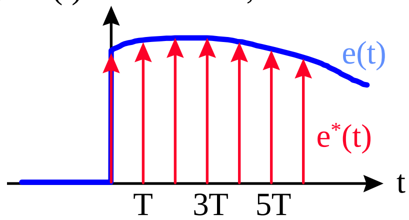
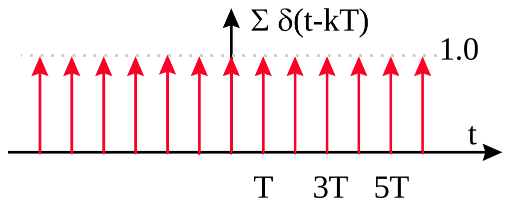
Content Slide 1: Discrete controller, continuous plant



[E2]

In the previous module we used the Tustin transformation to approximate the continuous controller, K(s), with a discrete controller, K(z) and we introduced the z-transform. In this module we will first provide a mathematical background to the z-transform to show its relationship to the Laplace transform. We will use this background to show that the z-transform provides a mapping from the complex s-domain to the complex z-domain. We will also show how the Tustin transformation approximates this mapping.

We will then complete the discussion of this control system implementation by showing the system closed-loop response. We will compare the system performance using different sampling rates. In this comparison, we will assume that aliasing is not an issue so that we can eliminate the effects of the anti-aliasing filter from the comparison.

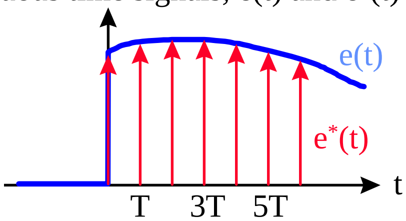
<table>
<tr><td colspan="2">Content Slide 2: Pulse train</td></tr>
<tr><td colspan="2">$$K(s)=\frac{u(s)}{e(s)}=\frac{b_2 s^2+b_1 s+b_0}{s^2+a_1 s}\qquad s\approx\frac{2}{T}\left(\frac{z-1}{z+1}\right)\ \Rightarrow\qquad K(z)=\frac{u(z)}{e(z)}=\frac{\beta_2 z^2+\beta_1 z+\beta_0}{z^2+\alpha_1 z+\alpha_0}$$</td></tr>
<tr><td>Continuous time signals, e(t) and $e^*(t)$
$e^*(t)$ = e(t) when t = kT, and is 0 when t ≠ kT

</td>
<td>Train of unit pulses, magnitude 1.
Σδ(t-kT) = 1, for t = kT and is 0 for t ≠ kT.

</td></tr>
<tr><td>e*(t) equals product of e(t) and pulse train
$$e^*(t)=e(t)\left(\sum_{k=0}^{\infty}\delta\,(t-kT)\right)$$</td>
<td>Equation for pulse-train of unit impulse functions
$$\text{pulse train}=\sum_{k=-\infty}^{\infty}\delta\,(t-kT)$$
where δ(t) = 1, for t = 0 and, 0 for t ≠ 0.</td></tr>
<tr><td colspan="2">and since δ(t) = 0 for t ≠ 0, we can write this as:
$$e^*(t)=e^*(kT)=\sum_{k=0}^{\infty}e(kT)\,\delta\,(t-kT)$$</td></tr>
</table>

[E3 highlight row 1]
In the previous module we showed how trapezoidal integration could be used to numerically integrate the differential equation associated with the continuous time transfer function. By introducing the z-transform and the unit time delay, z^{-1}, we arrived at a discrete time, z-domain transfer function. Now we'll provide a mathematical background for the z-transform.

[E3b highlight row 2]
Consider the continuous time function, e(t). Let's sample this function at a fixed sampling period, T. We can define a new variable, $e^*(t)$, [e-star] that is the sampled version of e(t). We can write this new variable as the product of e(t) and a train of unit pulses. The pulse train has value of one when the time corresponds to the sample time and zero elsewhere.

[E3c highlight row 3]
The pulse train can be represented as a summation of "delta" functions. Since the pulse train is zero when time is not equal to the sample time, we only have to evaluate the function when t = kT.

[E3d highlight row 4
The result is a continuous function of time that is a summation of values at discrete time points. Previously, we have taken the Laplace transform of continuous time function to avoid having to work with differential equations. This converts the function from the time domain to the s-domain and allows us to work with algebraic equations. Let's take the Laplace transform of the sampled data function, $e^*(t)$, to see if we can work with it in the "s-domain", as we have with previous functions..

<table>
<tr><td colspan="2">Content Slide 3: z-transforms</td></tr>
<tr><td>Continuous time signals, e(t) and $e^*(t)$

</td><td>Continuous time, sampled data function, $e^*(t)$
$$e^*(t) = e^*(kT) = \sum_{k=0}^{\infty} e(kT)\,\delta\,(t - kT)$$</td></tr>
<tr><td>Laplace Transform of e(t)
$$e(s) = \mathcal{L}(e(t)) = \int_0^{\infty} e(t)\exp(-st)dt$$</td><td>Laplace Transform of $e^*(t)$
$$e^*(s) = \mathcal{L}(e^*(t)) = \int_0^{\infty} e^*(t)\exp(-st)dt$$
$$e^*(s) = \sum_{k=0}^{\infty} e(kT)\exp(-kTs)$$</td></tr>
<tr><td colspan="2">Define exp(Ts) = z and we can definition of the z-transform of e(t) as:
$$e(z) = \mathcal{Z}(\,e(t)\,) = \sum_{k=0}^{\infty} e(kT)\,z^{-k}$$</td></tr>
</table>

[E4 high light row 2]
Recall that the Laplace transform of e(t) is defined as the integral with respect to time from zero to infinity of e(t) times the exponential of negative "st". We can take the Laplace transform of $e^*(t)$ [e-star], since it is a function of time. Moving the integral inside of the summation and recalling that the integral of the unit impulse is equal to one, results in a summation of the product of e(kT) and the exponential of (-kTs).

[E4b high light row 3]
Instead of working with exponential of "s", we can define a new variable z = exp(Ts). Substituting z into Y(s) for exp(Ts) yields the definition of the "z-transform" as shown above. So the z-transform results from the Laplace transform of a periodically sampled signal and a substitution of variables such that $z = e^{sT}$. The term "z^{-k}" is a k-step, time delay resulting from the pulse-train used to describe the sampled variable.

Content Slide 4: Pulse transfer function	
The z-Transform of x(t): $x(z) = \mathcal{Z}(\,x(t)\,) = \sum_{k=0}^{\infty} x(kT)\,z^{-k}$	
1) Unit Step Input x(t) = 1 for $t \geq 0$ and x(t) = 0 for $t < 0$ Laplace transform, $X(s) = \frac{1}{s}$	1) series converges $Z\{x(t)\} = \sum_{k=0}^{\infty} x(kT)\,z^{-k} = 1 + 1z^{-1} + 1z^{-2} + \ldots.$ $= \frac{z}{z-1} = \frac{1}{1-z^{-1}} = \frac{output(z)}{input(z)}$
2) Exponential $x(t) = e^{-at}$ = t for $t \geq 0$ and x(t) = 0 for $t < 0$. Laplace Transform, $X(s) = \frac{1}{s+a}$	2) series converges $Z\{x(t)\} = \sum_{k=0}^{\infty} e^{-akT}\,z^{-k} = 1 + e^{-aT}z^{-1} + e^{-2aT}z^{-2} + ..$ $= \frac{z}{z-e^{-aT}} = \frac{1}{1-z^{-1}e^{-aT}} = \frac{output(z)}{input(z)}$

[E4]
We can show a few examples of how the z-transforms are obtained.

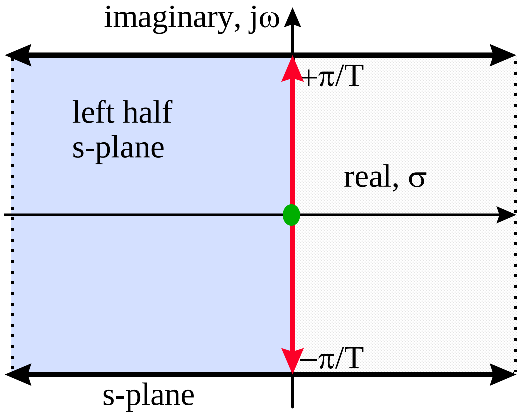
[E4 high light row 2]
Consider the time response of the unit step function (1). If we sample this function every T seconds, the resulting continuous function is a pulse-train of values equal to one. The z-transform of this time history results in a summation of powers of the delay operator z^{-1}. This infinite series converges to the function z over z minus one. Note that this equations also defines a pulse transfer function, from a pulse input, to a pulse output.

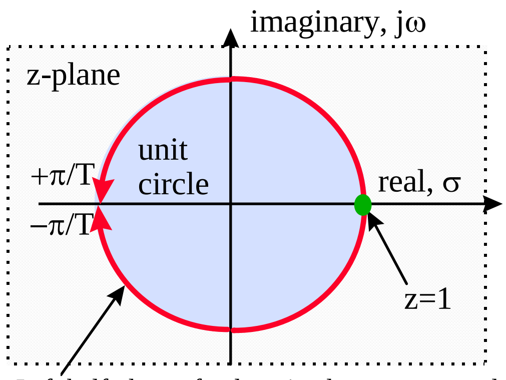
[E4b high light row 3]
Consider the z-transform of an exponentially decaying function of time (2). If we sample this function every T seconds, the resulting continuous function is a pulse train of values that have an exponential envelope. The z-transform of this time history results in an infinite series which converges to the function z over z minus the exponent of negative “aT”. This also defines a pulse or discrete transfer function.

These transformations can get complex. That is why there are tables of z-transforms that have already been calculated. If you want to use the z-transform to convert a continuous transfer function to a discrete transfer function, you need to include the effects of the zero-order hold. Computer-aided control system design package have automated the convert between the s-domain and z-domain, so you many not need to use the z-transform tables. In the previous module, we demonstrated the bilinear or Tustin’s transformation method. Let’s look at the mapping from the s-domain provided by both the z-transform and Tustin transformation.

Content Slide 5: Mapping

$z = e^{Ts}$; $z = e^{T(\sigma+j\omega)}$ $z = e^{\sigma T} e^{j\omega T}$; z = magnitude $e^{\sigma T}$, phase angle ωT





Left half plane of s-domain also gets mapped into the unit circle by the bilinear transformation.

[E7]

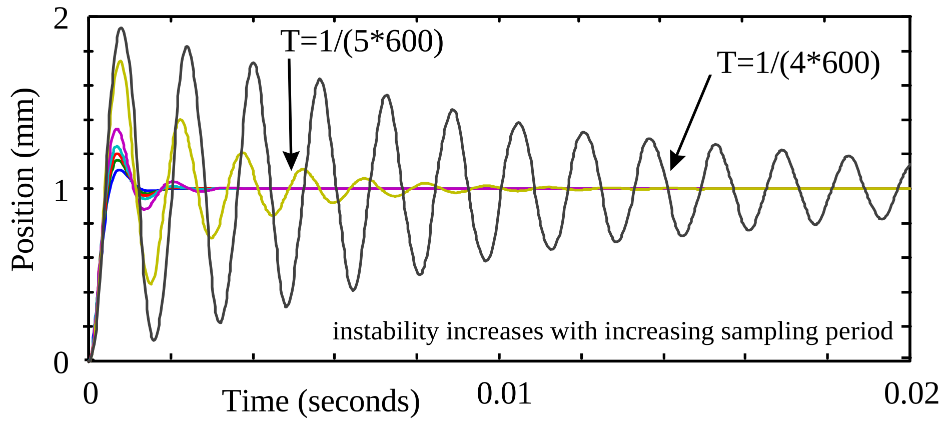
$z = e^{Ts}$ defines a nonlinear mapping from the "s" domain to the "z" domain. A point moving towards the left in the s-plane, would move towards the origin in the z-plane. Similarly, a point moving vertically in the s-plane, would move in a counter clockwise direction in the z-plane. The mapping can be reduced to a set of horizontal strips from the "s" domain to the entire "z" domain. These horizontal strips are $2\pi/T$ wide, but we'll only consider the first strip that is symmetric about the real axis.

From the study of continuous transfer functions, we know that transfer functions with poles in the left half of the complex plane are unstable. This means the real part of the pole must be negative for stability of a continuous transfer function. These stable poles get mapped from the left half s-plane to the unit circle in the z-plane. The unstable s-plane poles are on the right of the "j-omega" axis. These unstable s-plane poles get mapped outside the unit circle in the z-plane. Any poles on the "j-omega" axis in the s-plane get mapped to the unit circle in the z-plane.

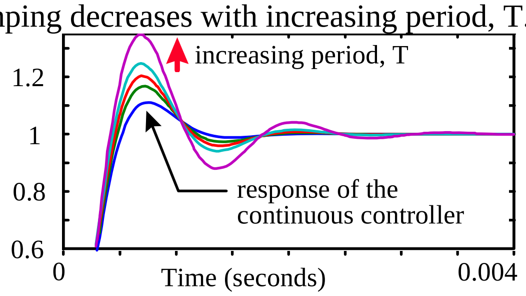
The bilinear transformation also performs a mapping from the s-plane to the z-plane. This mapping is difference than the z-transform, but similar in that the left halve of the s-plane (stable poles), get mapped into the unit circle in the z-domain. A more detailed discussion of this type of mapping relationship can be obtained from the topic of conformal mapping in complex variable theory.

Content Slide 6: The controller step response

Closed-loop step response with discrete controller, K(z)



Damping decreases with increasing period, T.



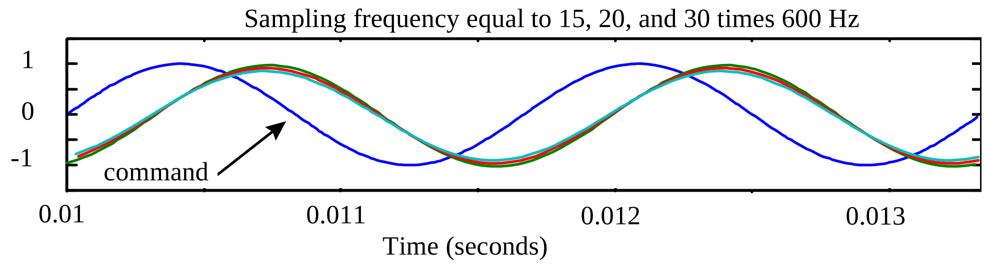
OverShoot (OS) indicative of damping

continuous	OS 10%	damping 0.6
T=1(30*600) = 5.6 μs	16%	ζ = 0.5
T=1/(20*600) = 8.3 μs	20%	ζ = 0.46
T=1/(15*600) = 11 μs	23%	ζ = 0.43
T=1/(10*600) = 17 μs	40%	ζ = 0.29
T= 1/(5*600) = 33 μs	75%	ζ = 0.09
T = 1/(4*600) = 41 μs	90%	ζ = 0.04

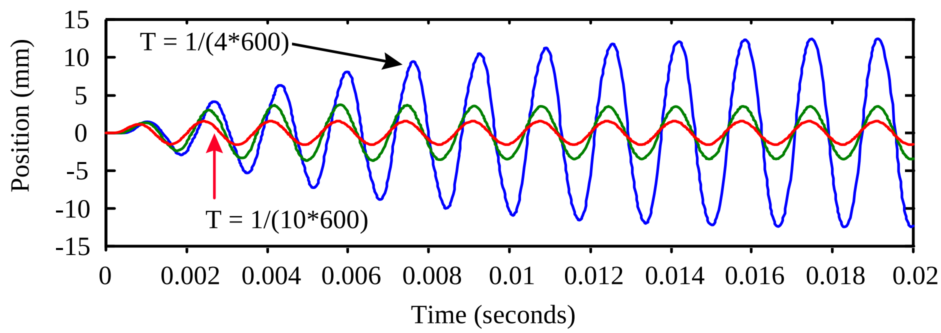
[E6 high light row 1]
A closed-loop simulation with the discrete controller and continuous plant was performed to evaluate the differences between the discrete controller as a function of the sampling period. The control system was designed for sinusoidal inputs, not step inputs, but we will use the step response of these controllers to view how the damping varies with changes in the sampling rate. The damping is indicative of the system stability. Note that as the sampling rate decreases and the sampling period increases the closed loop system becomes less damped and less stable. The percent overshoot due to the simulated step response increases with increasing sampling period. Note that even at sampling frequency equal to thirty times the design bandwidth of 600 Hertz, the response still deviates from the original continuous controller.

Content Slide 7: The controller sinusoidal response

Closed loop controller response to 600 Hz sinusoidal command using various sampling periods



Closed loop controller response to 600 Hz sinusoidal command using various sampling periods



[E6 highlight row 1]
Here we evaluate the control response to a 600 Hertz sinusoidal input. You can see that for sampling frequencies equal to 15, 20, and 30 times 600 Hz, the control response is similar. The response is what we would expect with a phase of about -45 degrees. The amplitude ratio of the continuous control would be about -3 dB for a peak of about 0.71. The response for the discrete controllers have a higher gain with a magnitude near 1.0.

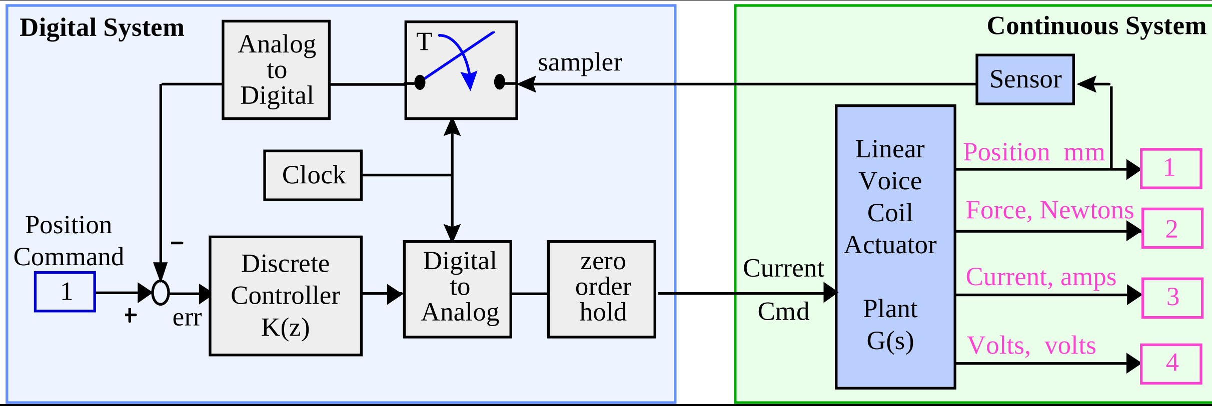
[E6 highlight row 2]
Note that control response for sampling frequencies of 4, 5, and 10 times 600 Hz have a gain greater than one. The process of going from a continuous design to a discrete implementation is now automated in most computer-aided control system design packages. But the results are far from guaranteed to function as desired because the discrete implementation is only an approximation of the continuous design. Simulating the mixed continuous and discrete system can help in finding any errors caused by this approximation process prior to going to the real hardware.

Content Slide 8: Coding Issues
Discrete transfer function, K(z) $$K(z)=\frac{u(z)}{e(z)}=\frac{\beta_2 z^2+\beta_1 z+\beta_0}{z^2+\alpha_1 z+\alpha_0}$$
Discrete control law as a difference equation. $u(kT) = \beta_2 e(kT)+\beta_1 e((k-1)T)+\beta_0 e((k-2)T)-\alpha_1 u((k-1)T)-\alpha_0 u((k-2)T)$

[E9 highlight row 2]
Before completing this discussion about control system implementation, we will briefly mention the issue of coding of this controls system in software. We're not going to address interrupt service routines or timing. We are interested in the computation of the control law.

If the processor being used to implement this controller has floating point capabilities, the task for implementing this discrete control law in software will be straightforward. If the selected processor does not support floating point and the control law has to be implement using fixed point integer math, then scaling is required to avoid the loss of resolution caused by overflow or underflow. Since the coefficients of our control law are not lightly to be integers, we need a way to deal with the fractional portion of the coefficient. One method is to break the coefficient up into the ratio of integer numbers. The process of multiplying a variable by a coefficient becomes a two step process comprised of a multiplication by the numerator and a division of the denominator. These numbers are selected so that the result of each step falls within the acceptable limits of the available word length in the selected processor. With the appropriate software tools, this fixed point implementation step can be evaluated in software prior to going to the actual hardware.

Content Slide 9: Summary



[E10]
In this module we provided a mathematical background to the z-transform to show its relationship to the Laplace transform. We used this background to show that the z-transform provides a mapping from the complex s-domain to the complex z-domain. We showed the relationship between stable pole of a continuous transfer function on the left half s-plane to stable poles of a discrete transfer function within the unit circle in the z-plane. We then discussed the implementation of the case study control law by showing simulated responses for the closed-loop system with a discrete controller for various sampling rates. The coefficients of the discrete controller were recalculated for each sampling rate. We took the anti-aliasing filter out of the feedback loop so that it would not affect the simulated results. Finally we concluded this discuss of controller implementation with a brief mention of the scaling that is required when calculating the control law using fixed point, integer math.

Development Notes:

Question 1
1) What is the z-transform? a) $z = e^{Ts}$ b) The Laplace Transform of a sampled data system, with a change of variables to avoid having to work with exponentials of the Laplace variable “s”. c) A discrete transfer function. d) A method for analyzing the stability of difference equations.
b) is correct. The z-transform is: The Laplace Transform of a sampled data system, with a change of variables to avoid having to work with exponentials of the Laplace variable “s”. $z = e^{Ts}$ is only a piece of the z-transform used to avoid having to work with exponentials. The summation due to the pulse-train is an important part of the z-transform.