Project Definition

The project is to model a spacecraft as a double integrator plant G(s)=(1/s)*(1/s). We will build an analog and digital control system with G(s) as above and the controller is C(s)=K (s+1)/(s+10). This is a "lead" compensator (similar to a PD). The plant is unstable and this C(s) stabilizes it under unity feedback. With varying the gain K, we will look at various underdamped responses.

Discrete-Time Simulation of First-Order Response

Source:

1. http://www.mathpages.com/home/kmath198/2-1/2-1.htm
2. http://www.mathpages.com/home/kmath249/kmath249.htm

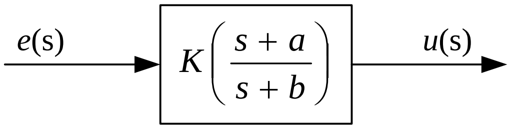
Since the dawn of time, engineers have been modeling real-world systems by **nth order constant coefficient linear differential equation**.

The first order lead/lag transfer function is a commonly used mathematical model for representing simple, continuous, dynamic response. A first order lead/lag coupling of two variables implies the relationship

$$b \cdot u + \frac{du}{dt} = K\left(a \cdot e + \frac{de}{dt}\right) \qquad \text{1-1}$$

where e is the independent variable, u is the dependent variable, t is time, K is the gain, and the coefficients a and b are coupling parameters. In the continuous domain this relationship uniquely determines the output response $u(t)$ for any given initial conditions and input function $e(t)$.

To allow us to use algebra *versus* calculus to simplify our nth order differential equation we map it into the **Laplace** domain function with $e(s)$ as an independent "input" signal, as shown below. Now we have our real-world system modeled by a **nth order polynomial**.



There exists a variety of algorithms for simulating (1-1) in a digital computer, and these algorithms differ both in their computational complexity and in their computed response characteristics.

There are two reasons for the diversity of digital simulations. First, according to equation (1-1) the change in the output during any interval of time depends on the value of the input continuously throughout the interval, but in digital systems the input is specified only by a sequence of discrete values. Hence the relationship is underspecified in the discrete-time domain, and can only be solved by making an assumption as to the behavior of the input variable between the discrete values. Second, practical constraints on execution time, execution frequency, and memory storage often lead to the use of modified solutions based on various simplifying assumptions.

Lead Compensator (general form)

$$C(s)=\frac{u(s)}{e(s)}=K\frac{(s+a)}{(s+b)}$$

Integration Methods

Source:

1. http://www.facstaff.bucknell.edu/mastascu/econtrolhtml/Sampled/Sampled5.html
2. http://www.facstaff.bucknell.edu/mastascu/econtrolhtml/Sampled/Sampled3.html

Now it is time to convert our continuous time state-space solution into its discrete time approximation (**Z-Transform**) and from there our **difference equations**. This involves finding a discrete approximation to a continuous integral (i.e., find the area under a curve). Because this is of course impossible to do with perfect accuracy (discrete does not equal continuous) we are reduced to approximations of which there seem to be three good ones, with two the de facto standard. Corollary: A single state-space solution can generate multiple Z-Transforms (all of which are correct, yet yielding different answers). As usual engineers like to give these two de facto standards multiple names.

First-order approximation: **Euler** Integration or Rectangular or Impulse Invariant Transformation: we approximate the area under a curve of width *dt* by a rectangle. To move from the Laplace (s) domain to the Z transform we replace every s with its Z-equivalent.

$$s=\frac{(z-1)}{zT}$$

Difference Equations:

Euler/Rectangular $$u_n=\frac{1}{(1+bT)}u_{n-1}+K\frac{(1+aT)}{(1+bT)}e_n-K\frac{1}{(1+bT)}e_{n-1}$$

Second-order approximation: Improved Euler or **Bilinear** or Trapezoidal or Predictor-Corrector or Heun: we approximate the area under a curve of width *dt* by a trapezoid. To move from the Laplace (s) domain to the Z transform we replace every s with its Z-equivalent.

$$s=\frac{2}{T}\cdot\frac{(1-z^{-1})}{(1+z^{-1})}=\frac{2}{T}\cdot\frac{(z-1)}{(z+1)}$$

Difference Equations:

Bilinear/Trapezoidal $$u_n=\frac{(2-bT)}{(2+bT)}u_{n-1}+K\frac{(2+aT)}{(2+bT)}e_n+K\frac{(aT-2)}{(2+bT)}e_{n-1}$$

Fourth-order approximation: **Runge Kutta** or Runge Kutta 4 algorithm: this is not one of the de facto standards.

System Design

The following is my baseline design to meet our project goals. You are encouraged to implement your own solution using any processor and hardware complement you wish.

Processor

- Arduino Duemilanove w/ ATmega328P

ADC

- The Atmel 328P microcontroller includes a 10-bit unipolar ADC subsystem,
- clk_{adc} = clk_{sys} / 128 = 125 KHz @ clk_{sys} = 16 MHz
- ADC in auto-trigger mode, 13.5 clk_{adc} / conversion = 9.259 ksps (10^3 samples / second)

DAC

- Microchip MCP4921 12-bit, digital-to-to analog converter (DAC)
- Unipolar or bipolar output
- 3-wire SPI serial interface
- Single-supply operation +4.5v to 5.25v

Plant

The plant will be modeled in hardware as a double integrator G(s)=(1/s)*(1/s).

Single Supply Op-Amps

source
Single Supply Op Amps
Maxim Application Note 656 - Design Trade-Offs for Single-Supply Op Amps

When working with a unipolar (0-Vcc, as opposed to bipolar +/-Vcc) supply, certain design considerations must be made. For instance, the output of a 411 Op-Amp can only get within about 1.5 volts of the power supply rails. If we were to power this chip from a single 5 volt supply this would only allow us to have outputs from about 1.5 to 3.5 volts.

Rail-To-Rail Op Amps

When doing single supply design it is common to use what are called rail to rail op amps. These amplifiers can output voltage very near the power supply voltages (or rails). You have already seen an example of one such amplifier, the LM324, whose output can go almost to the negative rail (ground), but still can't get close to the positive rail (Vcc). If you do you design carefully, you could use a 324 in this lab. It has the advantage of being quite cheap.

Another option is to use an op amp that goes near both supply rails. These tend to be much more expensive, but easier to work with. In the lab you will find two rail to rail op amps from Texas Instruments. the TLCV2772, and the TLC2774. The op amps on the chips are identical - the 2772 is an 8 pin DIP with 2 amplifiers, the 2774 is a 14 pin DIP with 4 amplifiers.

Input Voltage Range

Input common-mode voltage range is one of the first issues a designer should consider in specifying a single-supply op amp. The first impulse is to eliminate this concern by specifying a Rail-to-Rail input capability. Certain penalties must be paid, however, for true rail-to-rail operation. Most low-voltage op amps like the LM324 have input common-mode voltage ranges that include the negative supply rail, but only some allow inputs that extend to the positive rail, or even beyond the supply rails (Beyond-The-Rails™). Others allow input voltages only within one or two volts of the positive rail. Op amps that allow signals only to the negative rail will be referred to as ground-sensing amplifiers. Those that allow signals to either rail will be referred to as rail-to-rail input amplifiers.

Common-mode voltage range (CMVR) or Input Voltage Range (IVR) is the range of common mode signal for which the amplifier's operation remains linear. If we let the voltage present on the "-" input equal V1, and the voltage on the "+" input equal V2, then the common mode voltage is VCM = (V1+V2)/2. Some op amps, for instance, will only allow the common mode voltage of a signal to come within a diode drop or so of the power supply rails.

Impedance Matching

Figure 3 shows how to match impedances in the classic inverting and noninverting op-amp configurations. The inverting configuration (Figure 4) offers one way to eliminate changes in the input bias current by keeping the amplifier's common-mode input voltage constant at a reference voltage (V_{REF}). The output is given by $V_{OUT} = (-V_{IN} \times R2/R1) + V_{REF}(1 + R2/R1)$. If R2 = R1, this becomes $V_{OUT} = -V_{IN} + 2V_{REF}$. For $V_{REF} = 2V$ and V_{IN} between 0V and 3V, V_{OUT} ranges from 4V to 1V. The common-mode range is fixed, so CMR errors are eliminated as well.

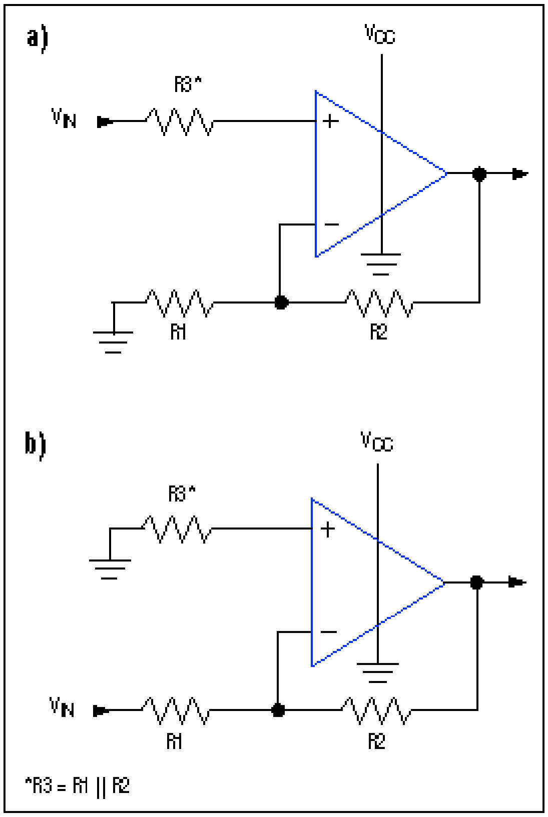


Figure 3. Matching the resistance at the inverting and noninverting nodes minimizes offset errors caused by input bias currents for both the noninverting (a) and inverting (b) configurations.

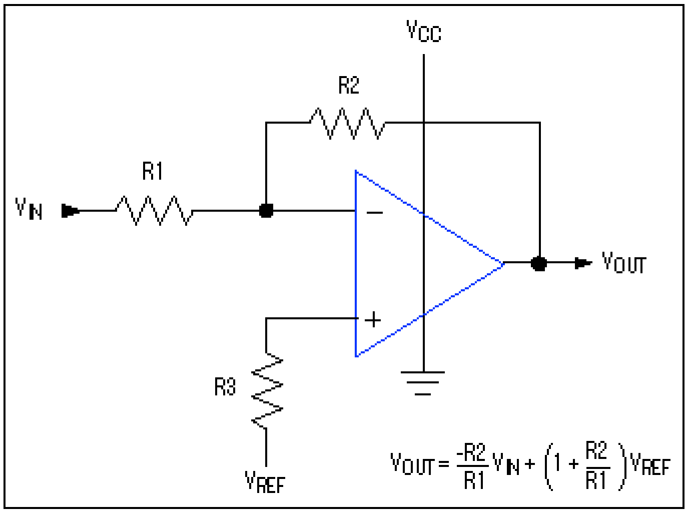

VCC
R2
R1
VIN
−
+
VOUT
R3
VREF
$V_{OUT} = \frac{-R2}{R1} V_{IN} + \left(1 + \frac{R2}{R1}\right) V_{REF}$

Figure 4. By holding the common-mode input voltage constant, the inverting-amplifier configuration eliminates common-mode rejection errors.

LM324 – Low Power Quad Operational Amplifier (LM124 Dual Operational Amplifier)

- Supports both dual +/-1.5v to +/- 16v and single 3v to 32v supplies.
- In the linear mode the input common-mode voltage range includes ground and the output voltage can also swing to ground, even though operated from only a single power supply voltage

- Input Voltage Range from 0 to V^{+}– 2
- Output Voltage Range from 0 to V^{+}– 1.5
- The unity gain cross frequency and input bias current are temperature compensated

TL082 dual supply

- CMVR = -12 V
- Note: This JFET device is designed for filter applications

Software

Step function input, error generation, and lead compensator will all be generated in software.

Program Language(s)

Three (3) Software Solutions to be Implemented (or at least attempted)

Arduino

Variable type is `long` 32-bit signed arithmetic operations (-2,147,483,648 to 2,147,483,647)

AVR Studio GCC Compiler ANSI X3.159-1989 and ISO/IEC 9899:1990 ("ANSI-C") standard

Variable type is `signed long long int, int64_t` 64-bit signed arithmetic operations

AVR Atmel Assembler

32-bit signed arithmetic operations (-2,147,483,648 to 2,147,483,647)

Project Reports

Step1 - Analog Control

1 Title Page, with your photo, and Table of Contents

2 Write-up, Introduction with Problem Statement

3 Simulation using Matlab, Simulink, or PSpice. Simulation input parameters including gain. Simulation output should be annotated to show rise time, peak amplitude, and settling time.

4 Schematic using OrCad, Eagle, etc. Schematic should include call-outs and optionally a bill-of-materials

5 Real world picture (e.g. photo of oscilloscope screen or screen capture for PC based oscilloscopes). The picture should be annotated to show rise time, peak amplitude, and settling time.

6 Conclusion, including tabular comparison of performance parameters (simulation vs. experimental).

Step 2 - Digital Wire - Sinewave

Using an 10-bit ADC input a sine wave from a waveform generator and then using a 12-bit SPI DAC output the sine wave to an oscilloscope. To complete this lab, most of you worked with the following three modules:

- 10-bit ADC ATmega subsystem
- ATmega AVR micoprocessor
- 12-bit Serial Peripheral Interface (SPI) DAC

1 Title Page, Photo of Yourself, and Table of Contents. Please include your photo so I can start placing names with faces.

2 Introduction with Problem Statement and Proposed Solution. Include a short discussion of each module comprising the solution (ADC, AVR, DAC). Reference the block diagram on page 3. Include a more detailed discussion of the software used to implement the solution. Because this software was provided to you, please demonstrate that you understand it. Reference the program listing also included on page 3.

3 Block Diagram showing Waveform Generator, ADC, AVR, DAC, and Oscilloscope. Software Program list file.

4 Photo of oscilloscope screen or screen capture for PC based oscilloscopes. The picture should be annotated to show waveform frequency and sample time.

Step 3 - Digital Control

Title Page, with photo and Table of Contents

Write-up containing paragraphs entitled Introduction, Problem Statement, and Proposed Solution. Your proposed solution should provide a fairly detailed discussion of your hardware/software implementation referencing your schematic and software code as needed. For example, Plant design and analog circuit implementation, Op-amp selection, compensator design and (software or hardware) implementation , summing junction design and (software or hardware) implementation, plus hardware Integration gain stage, level shifters and clamping circuit.

Simulation using Matlab, Simulink , or PSpice. If you are using the Simulink Double Integrator diagram provided by Dr. Shahian please update to (saturation and gain) reflect your implementation. Material should, as a minimum, include a simulation diagram (or Matlab function) and four (4) waveforms: Output of the compensator, Output of integrator #1, Output of integrator #2, Plant Output / Input to the software or hardware summing junction. See bs_doubleint simulink scope views for specific examples. The waveform showing the output of the plant, should be annotated to show rise time, peak amplitude, and settling time.

Schematic using OrCad, Eagle, etc. Schematic should include call-outs and optionally a bill-of-materials

Software code *if different* than the program provided by the lab instructor.

Experimental Results, including real world pictures (e.g. photo of oscilloscope screen or screen capture for PC based oscilloscopes). The picture showing the output of the plant should be annotated to show rise time, peak amplitude, overshoot, and settling time.

Tabular Comparison showing simulation versus measured values, with percent errors.

Conclusion, with a discussion of simulation versus measured data. Plus, what factors need to be considered when porting an analog control system to the digital domain?

Step 4 - Design of Digital Control Circuit

Title Page, with photo and Table of Contents

Write-up containing paragraphs entitled Introduction, Problem Statement, and Proposed Solution. Your proposed solution should provide a fairly detailed discussion of how you went from the design specifications (ex. steady state response) to your hardware/software implementation referencing your schematic and software code as needed.

Simulation using Matlab, Simulink , or PSpice. The waveform showing the output of the plant, should be annotated to show how design meets specifications.

Schematic using OrCad, Eagle, etc. Schematic should include call-outs and optionally a bill-of-materials

Software difference equation code, plus any other code *if different* than the program provided by the lab instructor.

Experimental Results, including real world pictures (e.g. photo of oscilloscope screen or screen capture for PC based oscilloscopes). The picture showing the output of the plant should be annotated to reflect design specifications.

Tabular Comparison showing design specifications, simulation and experimentally measured values.

Conclusion, with a discussion of your success or failure in meeting the design specifications.

Schedule

March 5th Build of analog control system with $G(s)=(1/s)*(1/s)$ and "lead" compensator $K(s)=K\ (s+1)/(s+10)$ complete. Step 1 of Project Documentation complete as defined above.

March 19th Open loop response of digital control system hardware to sinusoidal input. Maximum sample frequency measured. Step 2 of Project Documentation complete as defined above.

May 7th Build of digital control system with $G(s)=(1/s)*(1/s)$ and "lead" compensator $K(s)=K\ (s+1)/(s+10)$ complete. Step 3 of Project Documentation complete as defined above (*to be defined*).

May 14th Build of analog and digital control system with *new* factor and/or a *new* "lead/lag" compensator.