Embedded Scientific Computing Homework

September 23, 2020

Problem 01: Signed Binary Number Representations

Let $A = +107_{10}$, $B = -97_{10}$, and $C = -7_{10}$.

(A) Determine the 8-bit signed binary representations for A, B, and C.

(B) Determine the 8-bit signed binary representations for $-A$, $-B$, and $-C$.

(C) Perform the long-hand addition to show that A and $-A$ are addititve inverses.

(D) Determine the range of numbers representable using the 16-bit and 32-bit unsigned binary representations.

Problem 02: PIC18F1220 Microprocessor Arithmetic Support

Looking at the datasheets for a Microchip PIC18F1220 processor, answer the following questions:

(A) What kinds of addition are supported directly in hardware? (i.e., for what bit widths, using what number representations, supported directly with assembly language addition instructions)?

(B) How is carry and overflow information made available to the programmer?

(C) What hardware support does the processor provide for multiplication and division?

Problem 03: C Variable Bit Lengths

(A) On the personal computer of your choice, write C code that declare three variables as unsigned short integers; then, set two of the variables to the values 5 and 6, respectively. Do the subtraction 5-6, putting the result in the third variable. Display that variable using the printf command so that you can explicitly state the format in which you would like the results printed. "%u" is used when "printf-ing" unsigned variables in decimal format. Then answer the following questions:

  (i) What value does the result have?

  (ii) Given the result in (i), how many bits are used for an unsigned short integer? How do you know?

(B) Repeat (A) using unsigned integers instead of unsigned short integers

(C) Repeat (A) using unsigned long integers instead of unsigned short integers

Problem 04: Representing Numbers Using an $[n, -f]$ (Signed) Fixed-Point Representation Scheme

(A) For the number $M = -468.421875$ determine the round-off approximation $\overline{\beta_r}$ for M, the rounded value $\overline{M}$, and the round-off error ΔM using the $[16, -3]$ (signed) fixed-point representation scheme.

(B) For the number $M = -468.421875$ determine the round-off approximation $\overline{\beta_r}$ for M, the rounded value $\overline{M}$, and the round-off error ΔM using the $[16, -5]$ (signed) fixed-point representation scheme.

(C) For the number $M = -468.421875$ determine the round-off approximation $\overline{\beta_r}$ for M, the rounded value $\overline{M}$, and the round-off error ΔM using the $[16, -8]$ (signed) fixed-point representation scheme.

(D) Find the smallest n and f such that $M = -468.421875$ has an $[n, -f]$ (signed) fixed-point representation.

Problem 05: Representing Numbers Using an $[n, -f]$ Unsigned Fixed-Point Representation Scheme

(A) For the number $M = +98.6$ determine the round-off approximation $\overline{\beta_r}$ for M, the rounded value $\overline{M}$, and the round-off error ΔM using the $[16, -3]$ unsigned fixed-point representation scheme.

(B) For the number $M = +98.6$ determine the round-off approximation $\overline{\beta_r}$ for M, the rounded value $\overline{M}$, and the round-off error ΔM using the $[16, -5]$ unsigned fixed-point representation scheme.

(C) For the number $M = +98.6$ determine the round-off approximation $\overline{\beta_r}$ for M, the rounded value $\overline{M}$, and the round-off error ΔM using the $[16, -8]$ unsigned fixed-point representation scheme.

(D) For the number $M = +98.6$ determine the round-off approximation $\overline{\beta_r}$ for M, the rounded value $\overline{M}$, and the round-off error ΔM using the $[32, -24]$ unsigned fixed-point representation scheme.

(E) Find the smallest n and f such that $M = +98.6$ has an $[n, -f]$ unsigned fixed-point representation.

Problem 06: Interpreting Bit Patterns (Theoretical)

Let β be the bit pattern given by

$$\beta = 1011\ 0001\ 0111\ 1010\ 1111\ 0000\ 0000\ 0000.$$

(A) What number does β represent using the 32-bit unsigned binary representation scheme?

(B) What number does β represent using the 32-bit signed binary representation scheme?

(C) What number does β represent using the 32-bit IEEE 754 basic single precision floating-point representation scheme? (You may use an "IEEE-754 Floating Point Converter" such as can be found on the internet.)

Problem 07: Interpreting Bit Patterns (Applied)

In C on a workstation or laptop, check your answers in Question 07 by setting variables of various types to the bit pattern

$$\beta = 1011\ 0001\ 0111\ 1010\ 1111\ 0000\ 0000\ 0000;$$

you may use the hexadecimal pattern $B17AF000_{16}$. Then print the decimal value with a $printf$ statement. Note that you will have to be careful to use the proper $printf$ statement formatting for a given variable type. Do the printed values match your answers in question 06? If not, why not? Turn in your C code, the output, and your analysis.

Problem 08: Cubic Splines for Discrete Data Points

Construct the periodic cubic spline approximation for $g(t)$, where $g(t)$ is known for the following support points:

$$\begin{aligned} g(0) &= 1 \\ g(1) &= 0 \\ g(3) &= \frac{3}{4} \\ g(6) &= -1 \end{aligned} \tag{1}$$

Submit

(A) the cubic splines in a table with two columns, one column for the interval over which the cubic is accurate and one column for the cubic,

(B) a graph of both the function and the polynomials on the specified range.

Problem 09: Cubic Splines - Bessel Function

Find a natural cubic spline approximation for $J_0(x)$ the Bessel function of the first kind[1] on the interval $[0, 10]$. Use the interpolation points

$$\{0, 2.40482555769577, 3.83170597020751, 5.52007811028631,\\ 7.01558666981561, 8.65372791291101, 10\}.$$

Submit

[1]The MATLAB function for the Bessel function of the first kind $J_\nu(x)$ is *besselj(nu,x)*.

(A) the cubic splines in a table with two columns, one column for the interval over which the cubic is accurate and one column for the cubic,

(B) a graph of both the function and the polynomials on the specified range.

Problem 10: Least Squares Approximation

(A) The natural mollifier $\eta(x)$ is the function defined by

$$\eta(x) = \begin{cases} exp\left(\frac{1}{|x|^2-1}\right), & |x| < 1 \\ 0, & |x| \geq 1 \end{cases}.$$

This function is useful because it is non-zero only on the interval $[-1, 1]$, and has a continuous n^{th} derivative for all non-negative n. Using 101 evenly spaced fitting points on the interval $[-1, 1]$, determine the tenth-order least squares approximation to $\eta(x)$. Be sure to display the polynomial coefficients, and plot the approximation and $\eta(x)$ on the interval.

(B) Using 101 evenly spaced fitting points on the interval $[0, 10]$ determine the eighth-order least square approximation for $J_0(x)$ the Bessel function of the first kind. Obtain the eighth-order least square approximation for $J_0(x)$ using 1001 evenly spaced fitting points. Compare and contrast the two approximations. Plot the approximations and $J_0(x)$ on the interval.

Problem 11: (Lagrange) Barycentric Polynomial Approximation Problem

The gaussian function is defined by

$$f(x) = e^{-x^2}.$$

(A) Write the barycentric form of the interpolating polynomial $p_3(x)$ for the gaussian function on the interval $[-1, 1]$ using four evenly spaced fitting points.

(B) Use MATLAB to plot the gaussian function and the $p_3(x)$ you obtained in (A) on the same axes. Also plot the error on a separate graph.

(C) Use MATLAB to plot a tenth-order interpolating polynomial $p_{10}(x)$ using the barycentric form of the polynomial and eleven evenly spaced fitting points on the same axis as the gaussian. Also plot the error on a separate graph.

(D) Compare and contrast the two interpolating polynomials.

Problem 12: (Lagrange) Barycentric Polynomial Approximation Problem

Evenly spaced fitting points are often not well-suited for polynomial interpolation. A famous example is given by Runge to interpolate

$$f(x) = \frac{1}{1 + 25x^2}$$

on the interval $[-1, 1]$.

(A) Find a fifth-order approximation to $f(x)$ using six evenly spaced fitting points. Be sure to plot $f(x)$ and $p_5(x)$ on the same axes, and also plot the error $f(x) - p_5(x)$.

(B) Find a ninth-order approximation to $f(x)$ using ten evenly space fitting points. Be sure to plot $f(x)$ and $p_9(x)$ on the same axes, and also plot the error $f(x) - p_9(x)$.

(C) Find a 99^{th}-order approximation to $f(x)$ using one hundred evenly space fitting points. Be sure to plot $f(x)$ and $p_{99}(x)$ on the same axes, and also plot the error $f(x) - p_{99}(x)$.

(D) Compare the results for these three approximations.

Problem 13: Chebyshev Approximation

The gaussian function is defined by

$$f(x) = e^{-x^2}.$$

(A) Write the barycentric form of the interpolating polynomial $p_3(x)$ for the gaussian function on the interval $[-1, 1]$ using the extremal Chebyshev points.

(B) Use MATLAB to plot the gaussian function and the $p_3(x)$ you obtained in (A) on the same axes. Also plot the error on a separate graph.

(C) Use MATLAB to plot a tenth-order interpolating polynomial $p_{10}(x)$ using the barycentric form of the polynomial and the extremal Chebyshev points on the same axis as the gaussian. Also plot the error on a separate graph.

(D) Compare and contrast the two interpolating polynomials.

Problem 14: Barycentric Chebyshev Polynomial Approximation Problem

Evenly spaced fitting points are often not well-suited for polynomial interpolation. A famous example is given by Runge to interpolate

$$f(x) = \frac{1}{1 + 25x^2}$$

on the interval $[-1, 1]$.

(A) Find a fifth-order approximation to $f(x)$ using six extremal Cheyshev points. Be sure to plot $f(x)$ and $p_5(x)$ on the same axes, and also plot the error $f(x) - p_5(x)$.

(B) Find a ninth-order approximation to $f(x)$ using ten extremal Cheyshev fitting points. Be sure to plot $f(x)$ and $p_9(x)$ on the same axes, and also plot the error $f(x) - p_9(x)$.

(C) Find a 99^{th}-order approximation to $f(x)$ using one hundred extremal Cheyshev points. Be sure to plot $f(x)$ and $p_{99}(x)$ on the same axes, and also plot the error $f(x) - p_{99}(x)$.

(D) Compare the results for these three approximations with the three approximations from Problem 13.

——————The problems above this line will not change significantly[2] —————-

Problem 15: To Be Determined

Problem 16: Interpretation of a Processor

Look at the datasheets for a Microchip dsPIC30F1010 processor, and consider both the datapath (which Microchip calls the core and the DSP engine. Answer the following questions

[2] I may clarify or correct errors

1. What kinds of addition are supported directly in hardware? (i.e., for what bit widths, using what number representations, supported directly with assembly language addition instructions)?

2. How is carry and overflow information made available to the programmer?

3. What hardware support does the processor provide for multiplication, division, and multiply-and-accumulate operations? Is there hardware for these operations in the datapath (which Microchip calls the core, in the DSP engine, or both?

4. How does the programmer control whether the arithmetic being done in the DSP engine is "regular arithmetic" or saturation arithmetic?

Problem 17: Discretization of a Continuous-Time Filter

Consider the continuous-time system with transfer function

$$H_c(s) = \frac{1}{s^2 + \sqrt{2}s + 1}.$$

A discrete-time approximation to the system using the $[16, -8]$ two's complement representation scheme is to be designed. Problem 10 only deals with the infinite-precision discrete-time approximation of the filter. Problem 14 will revisit the problem by approximating the coefficients of a different discretization with fixed-point approximations.

(A) Using (forward) Euler's approximation, determine a discrete-time approximation with transfer function $H_{d,Euler}(z)$ using the sampling time $\Delta t = 1\ ms$.

(B) Determine the poles and zeroes of $H_{d,Euler}(z)$, noting that the poles are complex conjugates of each other.

(C) Plot the frequency responses of $H_{d,Euler}(z)$ and of $H_{d,exact}(z)$.

(D) Determine the L_p-norm error of the approximation $H_{d,Euler}(z)$ for $H_{d,exact}(z)$ for $p \in \{1, 2, \infty\}$. That is determine $\|H_{d,exact} - H_{d,Euler}\|_{L_p(\mathbb{T})}$ for the three values of p.

Problem 18: Approximation of the Coefficients of a Digital Filter

Suppose that you want to implement a digital filter with two zeros at $z = -1$ and a complex conjugate pair of poles at $z = 0.875 \pm i\ 0.211$.

(A) What is the transfer function $H(z)$ for this filter?

(B) What is the difference equation associated with this filter?

(C) In Matlab, plot a frequency response plot (magnitude and phase) with the unit-less magnitude (not in dB) and the x-axis on a linear scale. Assume that the sample time is $\Delta t = 100ms$, and label the x-axis in rad.

(D) Suppose that the difference equation associated with this filter is implemented with coefficients that are approximated using a $[16, -6]$ two's complement representation scheme. Where are the resulting zeros and poles of the approximate transfer function $\overline{H(z)}$?

(E) Use Matlab to plot the frequency responses (magnitude and phase) of the two versions of this filter on the same graph: $H(z)$ and $\overline{H(z)}$. Be sure you look at the result to see whether it is reasonable. (Based on the pole and zero positions, what do you expect?). Be sure that the axes are also labeled appropriately. Comment on any differences you see between the two filters. Do the differences make sense? (Note that the analysis so far assumes that once the coefficients are approximated, the rest of the math is "perfect", with no rounding or truncation needed.)

(F) Use Matlab to plot the frequency responses (magnitude and phase) or the error of the approximation $\overline{H(z)}$ for $H(z)$.

(G) Determine the L_p-norm error of the approximation $\overline{H(z)}$ for $H(z)$ for $p \in \{1, 2, \infty\}$. That is determine $\left\|H - \overline{H}\right\|_{L_p(\mathbb{T})}$ for the three values of p.

Problem 19: Approximation of the Real and Imaginary Parts of the Poles of a Digital Filter

Suppose that you want to implement a digital filter with two zeros at $z = -1$ and a complex conjugate pair of poles at $z = 0.875 \pm i\ 0.211$.

(A) Suppose the real parts and the imaginary parts of this filter are approximated using a $[16, -6]$ two's complement representation scheme. What are the resulting pole locations?

(B) What is the transfer function $\overline{H(z)}$ for this filter approximation?

(C) What is the difference equation associated with $\overline{H(z)}$?

(D) Use Matlab to plot the frequency responses (magnitude and phase) of the two versions of this filter on the same graph: $H(z)$ and $\overline{H(z)}$. Be sure you look at the result to see whether it is reasonable. (Based on the pole and zero positions, what do you expect?). Be sure that the axes are also labeled appropriately. Comment on any differences you see between the two filters. Do the differences make sense? (Note that the analysis so far assumes that once the coefficients are approximated, the rest of the math is "perfect", with no rounding or truncation needed.)

(E) Use Matlab to plot the frequency responses (magnitude and phase) or the error of the approximation $\overline{H(z)}$ for $H(z)$.

(F) Determine the L_p-norm error of the approximation $\overline{H(z)}$ for $H(z)$ for $p \in \{1, 2, \infty\}$. That is determine $\left\|H - \overline{H}\right\|_{L_p(\mathbb{T})}$ for the three values of p.

Problem 20: Approximation of a Digital Filter

Suppose that you want to implement a digital filter with two zeros at $z = -1$ and a complex conjugate pair of poles at $z = 0.875 \pm i\ 0.211$. (Note that this filter was used in Problem 11)

(A) Using Matlab, generate a step response for this system using the exact coefficients and double-precision floating point math.

(B) Direct Form I

- Draw the computation structure for this filter using the Direct Form I implementation
- Write C code to implement the difference equations using the Direct Form I implementation scheme and simulate the response of this filter to a unit step. Use the code skeleton given on the course Brightspace that reads inputs from a file and adds the outputs one at a time to an output file. You will only need to include the difference equation(s). Use double-precision floating point math so the effects of approximation of coefficients and variables are not introduced. Import (copy and paste from the file into MATLAB is sufficient) the output into MATLAB to plot the step response.

(C) Direct Form II

- Draw the computation structure for this filter using the Direct Form II implementation.
- Write the state-space represention that is the Direct Form II implementation.
- Write C code to implement the difference equations using the Direct Form II implementation scheme and simulate the response of this filter to a unit step. Use the code skeleton given on the course Brightspace that reads inputs from a file and adds the outputs one at a time to an output file. You will only need to include the difference equation(s). Use double-precision floating point math so the effects of approximation of coefficients and variables are not introduced. Import (copy and paste from the file into MATLAB is sufficient) the output into MATLAB to plot the step response.

(D) Coupled Form

- Determine the Coupled Form implementation by doing the following
  - Write the state space representation associated with the Direct Form implementation structure (this was done in an earlier part of the problem), paying close attention to the associated A_{DF} matrix.
  - Use MATLAB to find the eigenvalues and eigenvectors associated with the A_{DF} matrix. The MATLAB command *eig* may be helpful.
  - Use the real part u and the imaginary part v of one of the eigenvectors $u+iv$ to create the similarity transform $P = \begin{pmatrix} u & v \end{pmatrix}$.
  - Apply this similarity transform to the Direct Form II state-space representation.
- Draw the computation structure for this filter using the Coupled Form implementation
- Write C code to implement the difference equations using the Coupled Form implementation scheme and simulate the response of this filter to a unit step. Use the code skeleton given on the course Brightspace that reads inputs from a file and adds the outputs one at a time to an output file. You will only need to include the difference equation(s). Use double-precision floating point math so the effects of approximation of coefficients and variables are not introduced. Import (copy and paste from the file into MATLAB is sufficient) the output into MATLAB to plot the step response.

Problem 21A: Discretization of a Filter - Direct Form and Coupled Form

Consider the continuous-time system with transfer function

$$H_c(s) = \frac{1}{s^2 + \sqrt{2}s + 1}.$$

Using Tustin's method, a discrete-time approximation is given by

$$H_{d,Tustin}(z) = \frac{2^{20}\frac{(\Delta t)^2}{4+2\Delta t\sqrt{2}+(\Delta t)^2}(z+1)^2}{z^2 + z\frac{2(\Delta t)^2-8}{4+2\Delta t\sqrt{2}+(\Delta t)^2} + \frac{4-2\Delta t\sqrt{2}+(\Delta t)^2}{4+2\Delta t\sqrt{2}+(\Delta t)^2}}.$$

Note that a discrete-time approximation using Euler's method was designed for this system in Problem 11. The factor 2^{20} in the numerator is used to keep the numbers reasonable. After an output is calculated, the output's value can be divided easily by 2^{20} to obtain a final output.

(A) Using the sampling time $\Delta t = 2^{-10}$ s find the numbers for the coefficients. Determine the poles and zeroes of $H_d(z)$, noting that the poles are complex conjugates of each other (Note that you need more than four places to the right of the decimal point.). Plot the frequency response of $H_d(z)$.

(B) Determine the smallest number f_D such that approximating the denominator coefficients using the $[32, -f_D]$ two's complement approximation scheme does *not* result in any real poles. That is, the poles of $H_{d,Tustin}(z)$ must still be complex conjugates of each other.

(C) Determine the smallest number f_C such that approximating the real and imaginary parts of the poles using the $[32, -f_D]$ two's complement approximation scheme does *not* result in any real poles.

(D) Set $f = \max\{f_C, f_D\}$. Use the $[32, -f]$ two's complement representation scheme for the rest of this problem.

(E) Suppose the Direct Form I/II implementation scheme is used. Determine approximated system transfer function $\overline{H_{d,DF}(z)}$, where each multiplier is approximated with a multiplication by the best number with a $[32, -f]$ two's complement representation, the resulting poles and zeroes of $\overline{H_{d,DF}(z)}$. Plot the frequency response of $\overline{H_{d,DF}(z)}$.

(F) Suppose the Coupled Form implementation scheme is used. The resulting system may include an additional delay. Determine approximated system transfer function $\overline{H_{d,CF}(z)}$, where each multiplier is approximated with a multiplication by the best number with a $[32, -f]$ two's complement representation, the resulting poles and zeroes of $\overline{H_{d,CF}(z)}$. Plot the frequency response of $\overline{H_{d,CF}(z)}$.

(G) On the same set of axes, plot the frequency responses of $H_d(z)$, $\overline{H_{d,DF}(z)}$ and $\overline{H_{d,CF}(z)}$. Compare the frequency responses.

(H) Determine the $L_\infty(\mathbb{T})$-norm errors between the exact discretization $H_{d,exact}(z)$ and each of the approximations $\overline{H_{d,DF}(z)}$ and $\overline{H_{d,CF}(z)}$.

Problem 21B: Discretization of a Filter - δ Operator Form

Consider the continuous-time system with transfer function

$$H_c(s) = \frac{1}{s^2 + \sqrt{2}s + 1}.$$

Using Tustin's method, a discrete-time approximation is given by

$$H_{d,Tustin}(z) = \frac{2^{20}\frac{(\Delta t)^2}{4+2\Delta t\sqrt{2}+(\Delta t)^2}(z+1)^2}{z^2 + z\frac{2(\Delta t)^2-8}{4+2\Delta t\sqrt{2}+(\Delta t)^2} + \frac{4-2\Delta t\sqrt{2}+(\Delta t)^2}{4+2\Delta t\sqrt{2}+(\Delta t)^2}}.$$

Note that a discrete-time approximation using Euler's method was designed for this system in Problem 11. The factor 2^{20} in the numerator is used to keep the numbers reasonable. After an output is calculated, the output's value can be divided easily by 2^{20} to obtain a final output. In Problem 14A, you determined the number of fraction bits required for the Direct Form I/II implementation and the Coupled Form Implementation. Use the same number of fraction bits f in this problem.

1. Suppose the δ-operator implementation scheme is used. Determine approximated system transfer function $\overline{H_{d,\delta}(z)}$, where each multiplier is approximated with a multiplication by the best number with a $[32, -f]$ two's complement representation, the resulting poles and zeroes of $\overline{H_{d,\delta}(z)}$. Plot the frequency response of $\overline{H_{d,\delta}(z)}$.On the same set of axes, plot the frequency responses of $H_d(z)$, $\overline{H_{d,DF}(z)}$, $\overline{H_{d,CF}(z)}$, $\overline{H_{d,\delta}(z)}$. Compare the frequency responses.

2. Determine the $L_\infty(\mathbb{T})$-norm errors between the exact discretization $H_{d,exact}(z)$ and each of the approximations $\overline{H_{d,DF}(z)}$, $\overline{H_{d,CF}(z)}$, $\overline{H_{d,\delta}(z)}$.

3. Write C code to implement the difference equations for the δ-operator implementations of this system, and simulate the response of each of these filters to a unit step. Use double-precision floating point math so the effects of approximation of the variables are not introduced.

Problem 22: Number of Integer Bits for a Filter

Consider the system that has the state-space representation

$$\begin{aligned} w[k+1] &= \begin{pmatrix} -3.5855 & -3.0095 & 24.0525 \\ -4.1710 & -3.0190 & 32.1050 \\ -1.2928 & -1.0048 & 9.0263 \end{pmatrix} w[k] + \begin{pmatrix} 1 \\ 2 \\ 0.5 \end{pmatrix} x[k] \\ y[k] &= \begin{pmatrix} 0.7365 & 1.2990 & -4.6692 \end{pmatrix} w[k]. \end{aligned}$$

(A) Draw the implementation diagram for this respresentation.

(B) Determine the three transfer functions given below. I recommend using MATLAB (and the command $ss2tf$) to help you do so.

- $G_{x\mapsto w_1}(z) = \frac{W_1(z)}{X(z)}$
- $G_{x\mapsto w_2}(z) = \frac{W_2(z)}{X(z)}$
- $G_{x\mapsto w_3}(z) = \frac{W_3(z)}{X(z)}$

(C) Determine the impulse responses $g_{x\mapsto w_1}[k]$, $g_{x\mapsto w_2}[k]$, and $g_{x\mapsto w_3}[k]$.

(D) Determine the ℓ_1 norms for each of the impulse responses found in part (C).

(E) Use the ℓ_1 norms found in part (C) to determine the number of integer bits required for the signals $w_1[k]$, $w_2[k]$, and $w_3[k]$.

(F) Using the number of integer bits required for $w_1[k]$, $w_2[k]$, and $w_3[k]$, determine the number of integer bits required to avoid overflow for the outputs of each multiplier and each adder.

Problem 23: Number of Fraction Bits for a Filter

Consider the system that has the state-space representation

$$\begin{aligned} w[k+1] &= \begin{pmatrix} -3.5855 & -3.0095 & 24.0525 \\ -4.1710 & -3.0190 & 32.1050 \\ -1.2928 & -1.0048 & 9.0263 \end{pmatrix} w[k] + \begin{pmatrix} 1 \\ 2 \\ 0.5 \end{pmatrix} x[k] \\ y[k] &= \begin{pmatrix} 0.7365 & 1.2990 & -4.6692 \end{pmatrix} w[k]. \end{aligned}$$

Note that this problem is a continuation of Problem 15, where you determined the number of integer bits required for the signals in this filter.

(A) Determine the transfer functions noted below. I recommend using MATLAB (and the command $ss2tf$) to help you do so.

- $G_{\Delta x\to y}(z) = \frac{Y(z)}{\Delta X(z)}$
- $G_{\Delta w_1\to y}(z) = \frac{Y(z)}{\Delta W_1(z)}$
- $G_{\Delta w_2\to y}(z) = \frac{Y(z)}{\Delta W_2(z)}$
- $G_{\Delta w_3\to y}(z) = \frac{Y(z)}{\Delta S_1(z)}$

(B) Determine the impulse responses $g_{\Delta x\mapsto y}[k]$, $g_{\Delta w_1\mapsto y}[k]$, $g_{\Delta w_2\mapsto y}[k]$, $g_{\Delta w_3\mapsto y}[k]$

(C) Determine the ℓ_1 norms for every transfer function obtained in Part (B).

(D) Use the norms obtained in Part (C) to determine how many fraction bits are needed for the signals $x[k]$, $w_1[k]$, $w_2[k]$ and $w_3[k]$, if it is required that

$$\|\{\Delta y[k]\}\|_{\ell\infty} \leq 2^{-10}.$$

(E) Summarize Problem 15 and Problem 16 by writing the required fixed-point representation schemes for each signal $x[k]$, $w_1[k]$, $w_2[k]$ and $s_1[k]$.

(F) Write the equations for each of the signals $w_1[k+1]$, $w_2[k+1]$, $w_3[k+1]$, and $y[k]$ using additions, multiplications, and shift operators.