AER4410
Digital Control Applications

LEVEL CONTROL PROJECT REPORT

Full Project

Student Name: Roaa Tareq Mohammed
Sec.1 BN.24

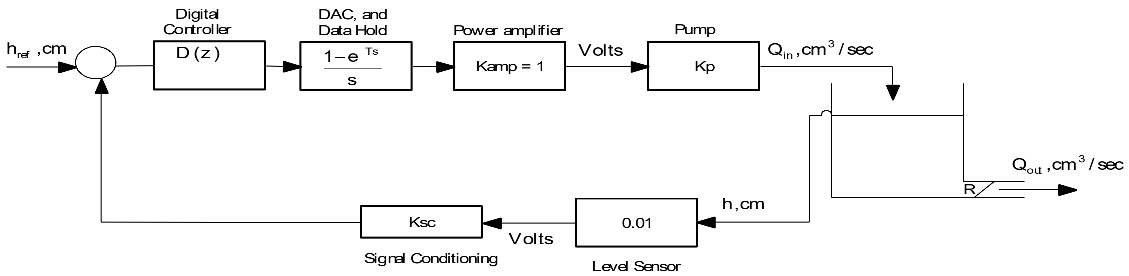
Submitted to: Prof. Osama Mohamady
Due Date: Dec.13^{th}, 2022

Table of Contents

Problem Statement

Considering the water level control system shown, we applied 10 volts step to the pump. Some readings were taken from the level sensor (10 m volts/cm sensitivity), the sensor reading settles after T_s sec on a value V volts.

$B.no.$	T_s	V
24	$10 + (B.no,/10) = 12.4\ sec$	$0.3\ volts$



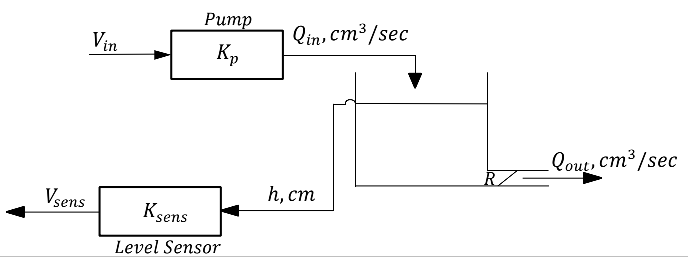
Question 1

Treating the pump as a gain $\left(K_p\ \frac{cm^3/sec}{volt}\right)$, determine this gain and the pressure resistance in the outlet pipe.

ρ_{water}	g	Tank Cross-section	K_{sens}
$1\ gm/cm^3$	$981\ cm/sec^2$	15x15 cm^2	0.01

Solution

Mathematical Modelling

$A\dot{h} = q_{in} - q_{out} \xrightarrow{\mathcal{L}} Ahs = Q_{in} - Q_{out}$	
$Q_{in} = K_p V_{in}$	
$Q_{out} = \frac{\rho g h}{R}$	
$V_{sens} = K_{sens} h$	

$$Ahs = K_p V_{in} - \frac{\rho g h}{R} \rightarrow \boxed{G(s) = \frac{h}{V_{in}} = \frac{K_p/A}{s + \frac{\rho g}{AR}}} \ \& \ \boxed{G_s(s) = \frac{V_{sens}}{V_{in}} = \frac{K_{sens} K_p/A}{s + \frac{\rho g}{AR}}}$$

Parameters Identification

The time constant

$$\tau = \frac{AR}{\rho g}$$

The settling time ($Given: T_s = 12.4\ sec$)

$$T_s = 4\tau = 4\frac{AR}{\rho g} = 12.4$$

Hence,

$$\begin{aligned} R &= \frac{1}{4}\frac{\rho g T_s}{A} \\ &= \boxed{13.516} \end{aligned}$$

Using the final value theorem

$$\begin{aligned} V_{sens}|_{final} = V &= \lim_{s\to 0} s \cdot V_{in} G_s(s) \\ &= \lim_{s\to 0} \cancel{s} \cdot \frac{10}{\cancel{s}} G_s(s) \\ &= \lim_{s\to 0} 10 \left(\frac{K_{sens}K_p/A}{s + \frac{\rho g}{AR}} \right) \end{aligned}$$

Therefore,

$$V = 10\left(\frac{K_{sens}K_p}{\frac{\rho g}{R}}\right)$$

$$\begin{aligned} K_p &= \frac{V \cdot \frac{\rho g}{R}}{10\ K_{sens}} \\ &= \boxed{217.74194} \end{aligned}$$

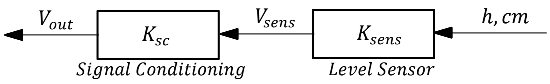
Hence, the transfer function of the pump and tank plant

$$\boxed{G(s) = \frac{h}{V_{in}} = \frac{\frac{30}{31}}{s + \frac{10}{31}}}$$

where $\tau = \frac{31}{10} = 3.1\ sec$

Question 2

It's required to design a signal conditioning unit for the level sensor in a level range $0 \rightarrow 50$ cm.



Solution

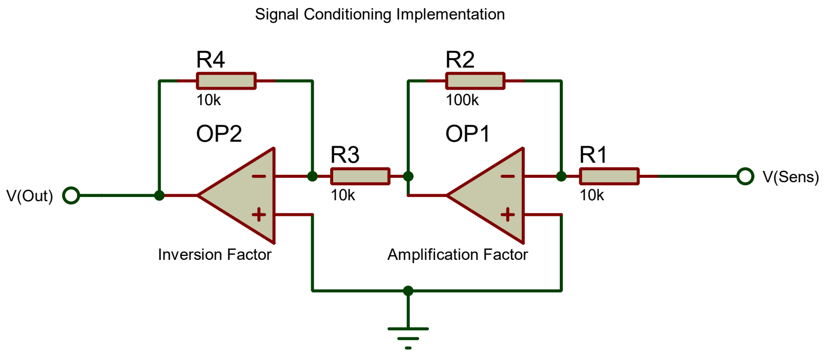
The sensor measures the water level with 10 m volts/cm sensitivity. Therefore, for the level range $0 \rightarrow 50$cm, the sensor will give voltage range

$$50cm \cdot 10m\frac{volts}{cm} = 0.5volts \Longrightarrow V_{sens}: 0 \rightarrow 0.5V$$

An amplification factor is required to utilize the resolution of Arduino input voltage range $(0 \rightarrow 5V)$. The signal conditioning factor is calculated as follows

$$\begin{aligned} K_{sc} &= \frac{5V}{0.5V} \\ &= \boxed{10} \end{aligned}$$

The signal conditioning implementation using two OpAmps: one for amplification and the other for inversion.



Question 3

To discretize this system with 0.1 sec sampling period, is this sampling period suitable? if not, select a suitable sampling period.

Solution

The criteria set on sampling period of simple lag systems is

$$T_s \leq \frac{\tau}{10}$$

For $\tau = 3.1$, sampling period is limited by

$$T_s \leq 0.31$$

Therefore, 0.1 sec sampling period is suitable for this system.

Question 4

Design a digital controller to achieve 5 sec settling time, 5 % overshoot, and 5 % steady state error.

Solution

The digital controller is required to achieve the following characteristics:

- $T_s = 5\ sec$

$$T_s = \frac{4}{\zeta\omega_n} = 5 \rightarrow \zeta\omega_n = 0.8$$

- $M_p = 5\%$

$$M_p = e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}} = 0.05 \rightarrow \zeta = 0.69$$

$$\omega_n = \frac{\zeta\omega_n}{\zeta} = 1.5942\ s^{-1}\ and\ \omega_d = \omega_n\sqrt{1-\zeta^2} = 0.8392\ s^{-1}$$

The desired closed-loop poles are

$$z_{1,2} = e^{s_{1,2}T} = e^{-\zeta\omega_n T}\angle \pm \omega_d T \quad \begin{aligned} &= 0.92312\angle \pm 0.08392 \\ &= 0.91987 + 0.077377\ i \end{aligned}$$

Therefore, the desired closed-loop characteristic equation is

$$z^2 - 1.83974\ z + 0.85215 = 0$$

And the closed-loop transfer function is

$$T(z) = \frac{a}{z^2 - 1.83974\ z + 0.85215}$$

where a can be chosen to satisfy the steady state error.

- $E_{ss} = 5\%$

$$E_{ss} = \lim_{z\to 1}(1 - z^{-1})R(z)\big(1 - T(z)\big) = \lim_{z\to 1}\cancel{(1 - z^{-1})} \cdot \frac{1}{\cancel{(1 - z^{-1})}} \cdot \big(1 - T(z)\big) = 1 - T(1)$$

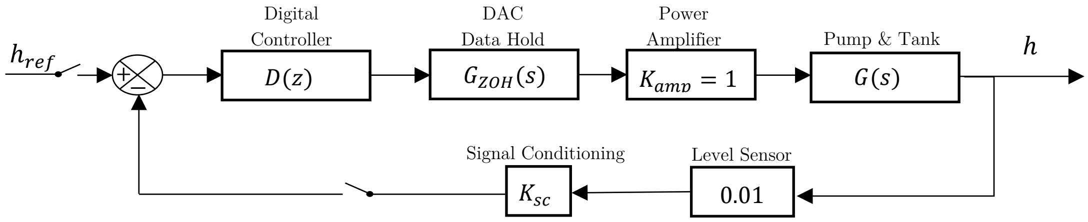
$$\therefore 1 - T(1) = 0.05 \rightarrow T(1) = 0.95$$

$$T(1) = \frac{a}{0.01241} = 0.95 \rightarrow a = 0.0117895$$

Hence, the closed-loop transfer function is

$$\boxed{T(z) = \frac{0.0117895}{z^2 - 1.83974\ z + 0.85215}}$$

The system can be represented by the following block diagram



The closed-loop transfer function

$$\frac{h}{h_{ref}} = \frac{D(z)G_p(z)}{1 + D(z)G_pH(z)}$$

Function	S-domain	Z-domain
$G_p = G_{ZOH}K_{amp}G$	$(1-e^{-sT})\cdot\frac{30/31}{s(s+10/31)} \rightarrow \begin{cases} a = \frac{10}{31} \\ T = 0.1 \end{cases}$	$\cancel{(1-z^{-1})}\cdot\frac{0.09523}{\cancel{(1-z^{-1})}(z-0.96826)}$ [1]
$H = K_{sc}\cdot K_{sens}$	$(10)(0.01) = 0.1$	0.1
$G_pH = G_p\cdot H$	$(1-e^{-sT})\cdot\frac{30/31}{s(s+10/31)}\cdot 0.1$	$\frac{0.009523}{z-0.96826}$
Therefore,		
$\boxed{G_p(z) = \frac{0.09523}{z-0.96826}}$		$\boxed{G_pH(z) = \frac{0.009523}{z-0.96826}}$

Hence, the transfer function of the digital controller can be written as follows:

$$\begin{aligned} D(z) &= \frac{T(z)}{G_p(z) - G_pH(z)T(z)} \\ &= \frac{\left(\frac{0.0117895}{z^2 - 1.83974\,z + 0.85215}\right)}{\left(\frac{0.09523}{z-0.96826}\right) - \left(\frac{0.009523}{z-0.96826}\right)\left(\frac{0.0117895}{z^2 - 1.83974\,z + 0.85215}\right)} \\ &= \frac{(0.0117895)(z-0.96826)}{(0.09523\,)(z^2 - 1.83974\,z + 0.85215) - (0.009523)(0.0117895)} \\ &= \boxed{\frac{0.0117895\,z - 0.011415}{0.09523\,z^2 - 0.175198\,z + 0.081038}}\ or\ \boxed{\frac{0.0117895\,z^{-1} - 0.011415\,z^{-2}}{0.09523 - 0.175198\,z^{-1} + 0.081038\,z^{-2}}} \end{aligned}$$

The difference equation of the control action

$$D(z) = \frac{0.0117895\,z^{-1} - 0.011415\,z^{-2}}{0.09523 - 0.175198\,z^{-1} + 0.081038\,z^{-2}} = \frac{U(z)}{E(z)}$$

$$\boxed{0.09523\,u(k) - 0.175198\,u(k-1) + 0.081038\,u(k-2) = 0.0117895\,e(k-1) - 0.011415\,e(k-2)}$$

[1] $Z\left[\frac{a}{s(s+a)}\right] = \frac{(1-e^{-aT})z^{-1}}{(1-z^{-1})(1-e^{-aT}z^{-1})}$

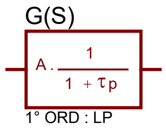
Question 5

Write a program on Arduino to perform the digital controller function.

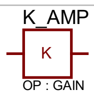
Solution

Building the circuit on Proteus

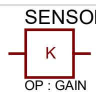
- Write the $G(s)$ in the following form

	$G(s) = \frac{\frac{30}{31}}{s + \frac{10}{31}} \cdot \frac{\frac{31}{10}}{\frac{31}{10}} = \frac{3}{3.1\, s + 1} \rightarrow \begin{cases} A = 3 \\ \tau_p = 3.1 \end{cases}$

- The amplification gain K_{amp}

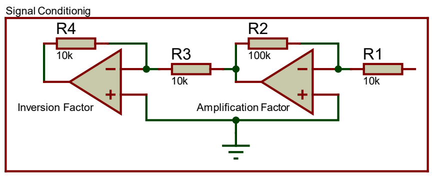
	$K = 1$

- The sensor gain K_{sens}

	$K = 0.01$

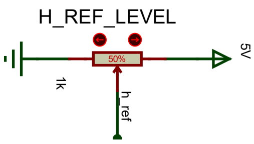
NB: The input and output are assigned to specific terminals on the gain block

- The signal conditioning block

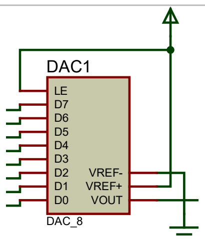


Amplification OPAMP	Inversion OPAMP
$\frac{R2}{R1} = 10 \rightarrow K_1 = -10$	$\frac{R4}{R3} = 1 \rightarrow K_2 = -1$
$K_{sc} = K_1 \cdot K_2 = 10$	

- Liquid level reference point is set using a potentiometer

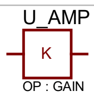
	The input form the potentiometer ranges form 0 to $5V$. This signal is then amplified in the code to reach the liquid level range of $0 \rightarrow 50\ cm$.

- An external DAC unit is used since Arduino UNO do not support DAC

	The control action is calculated using the digital controller difference equation and then provided through PORTD 8 pins to the 8 pins of the DAC unit.

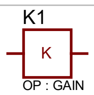
- The control action compensation gain U_{amp}

Based on the results obtained in Simulink model section, the range of the control action is measured by applying the maximum input (maximum tank level). The maximum value of the controller action is found to be 33.2. The microcontroller output ranges from 0 to 5, therefore, an amplification gain K is applied to DAC output for signal conditioning.

	$K = \dfrac{33.2}{5} = 6.64$

- The liquid level gain K_1

The liquid level passed to the microcontroller ranges from 0 to 5, while the real liquid level range is $0 \rightarrow 50cm$. An amplification gain K is multiplied by this signal to compare it with the time response.

	$K = 10$

Proteus Circuit

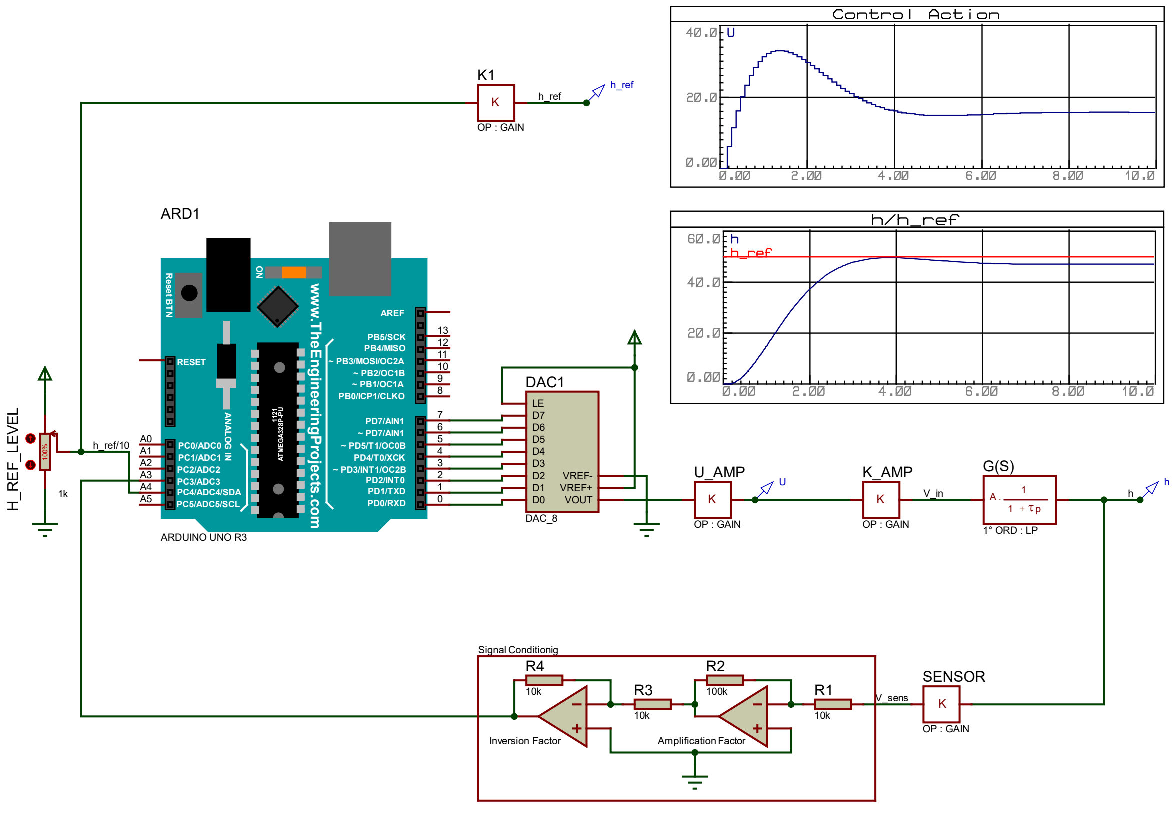


Figure 1 System simulating circuit in Proteus

Time response in Proteus

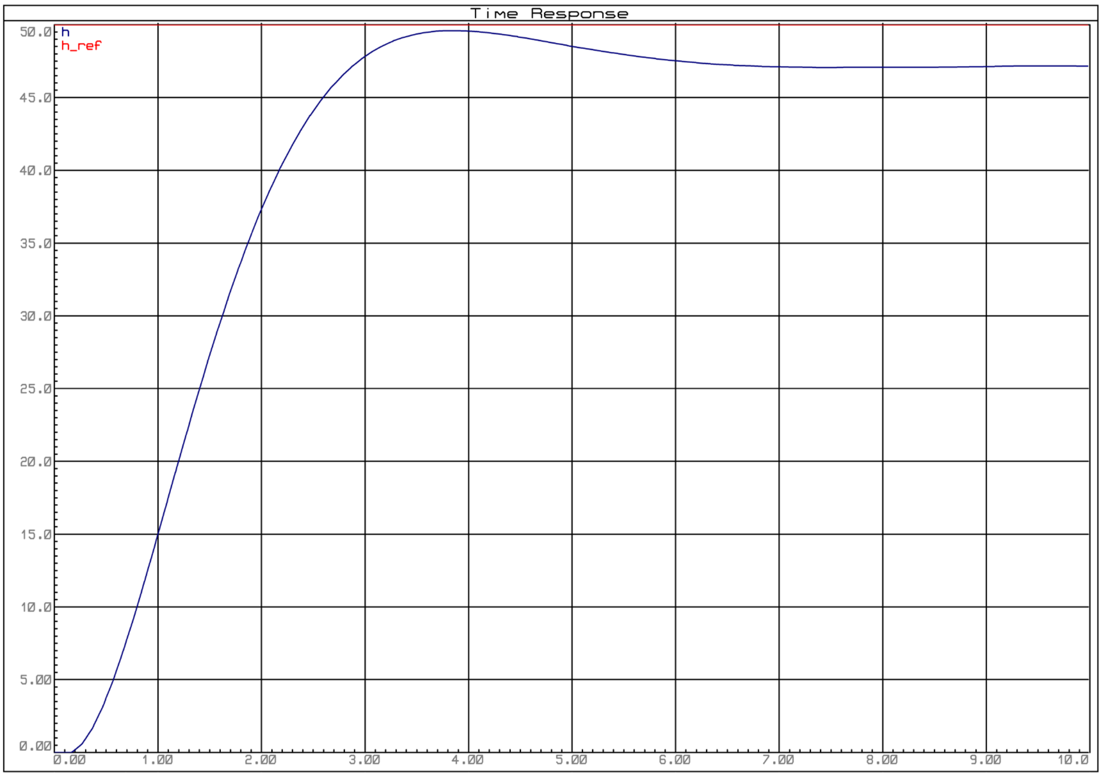


Figure 2 Time response of the system compared to the input in Proteus

Control Action in Proteus

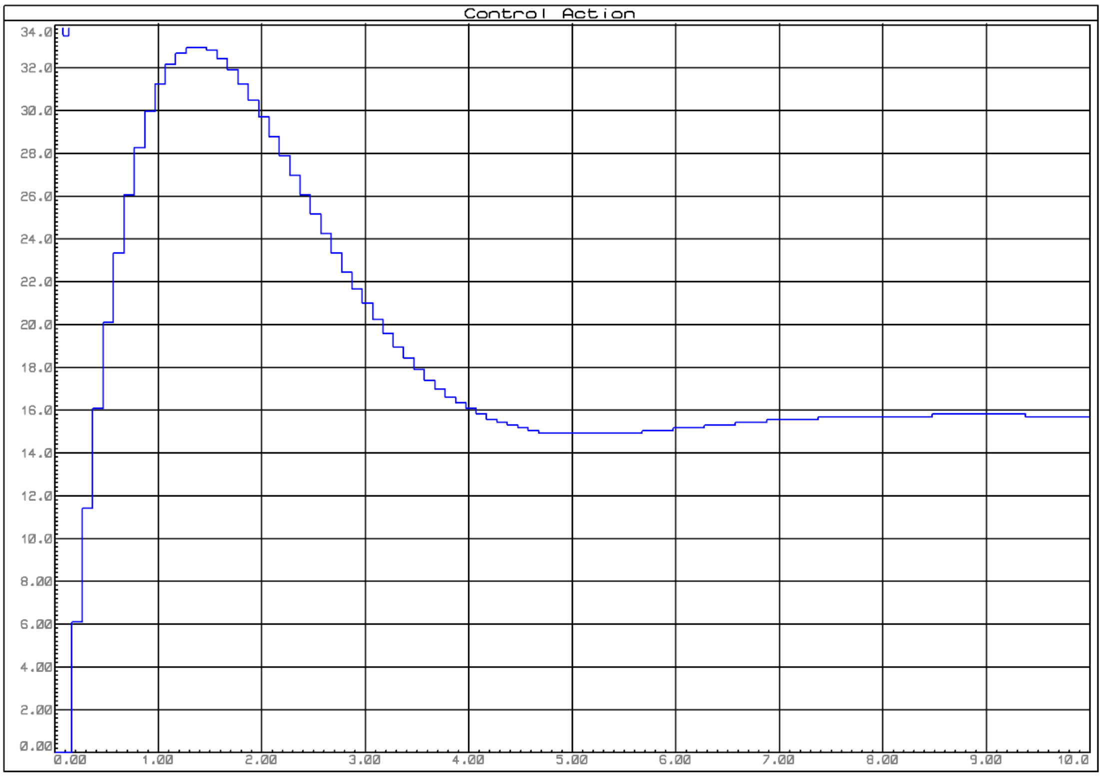


Figure 3 Digital controller action in Proteus

Writing the program on Arduino UNO R3

Pin connections

Input	A3	V_{out}: Signal condition output
	A4	h_{ref}: Liquid level reference point
Output	PORTD (PD0-PD7)	8-bit DAC

Digital controller difference equation

$$u(k) = 1.83974\, u(k-1) - 0.85097\, u(k-2) + 0.1238\, e(k-1) - 0.11987\, e(k-2)$$

Arduino Program

```
float href, Vout, BN, t;
float u[3] = {0,0,0}; // u = {u(k), u(k-1), u(k-2)}
float e[3] = {0,0,0}; // e = {e(k), e(k-1), e(k-2)}

void setup() {
  // Define PortD pins as Output
  DDRD = B11111111;
}

void loop() {
  // Read current time
  t = millis();

  // Read the output from the conditioned signal measured by the sensor
  Vout = analogRead(A3)/1023.0*5.0;       // [/1023.0*5.0] => 0->5
  // Read the reference point
  href = analogRead(A4)/1023.0*5.0*10.0; // [/1023.0*5.0*10] => 0->50 cm

  // Summation point operation
  e[0] = href - Vout;
  // Control action using the digital controller difference equation
  u[0] = 1.83974*u[1]-0.85097*u[2]+0.1238*e[1]-0.11987*e[2];

  // Output to the DAC point
  BN = u[0]/33.2*255.0; // [/33.2*255.0] => 0->255 (33.2:highest u)
  PORTD = BN;

  // Update values each sampling period
  u[2] = u[1];
  u[1] = u[0];

  e[2] = e[1];
  e[1] = e[0];

  delay(100-(millis()-t));
}
```

Simulink model

Pump & Tank Transfer Function	$G(s) = \dfrac{30/_{31}}{s + 10/_{31}}$
Digital Controller	$D(z) = \dfrac{0.0117895\, z - 0.011415}{0.09523\, z^2 - 0.175198\, z + 0.081038} = \dfrac{0.1238\, z - 0.11987}{z^2 - 1.83974\, z + 0.85097}$
Feedback Gain	$K_{feedback} = K_{sc} \cdot K_{sens} = 0.1$

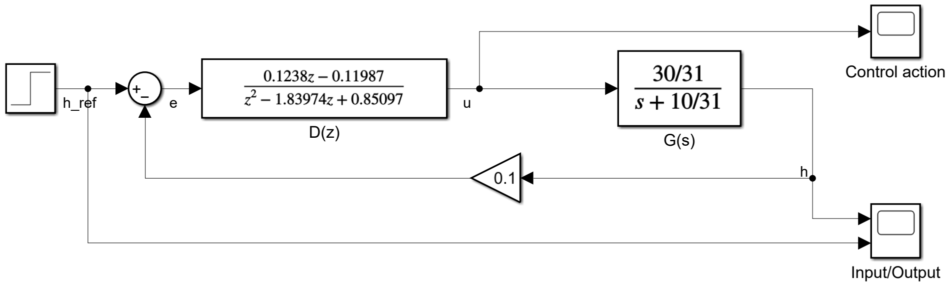


Figure 4 System simulating circuit in Simulink

Time Response in Simulink

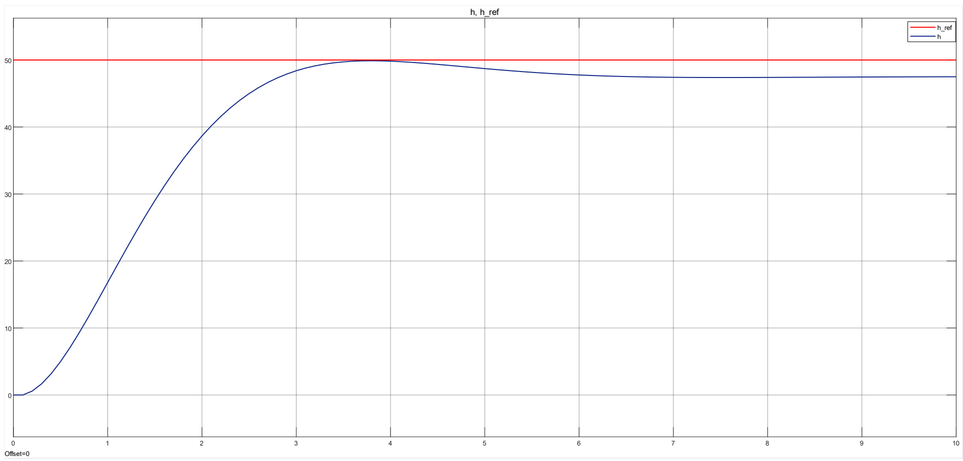


Figure 5 Time response of the system compared to the input in Proteus

Control Action in Simulink

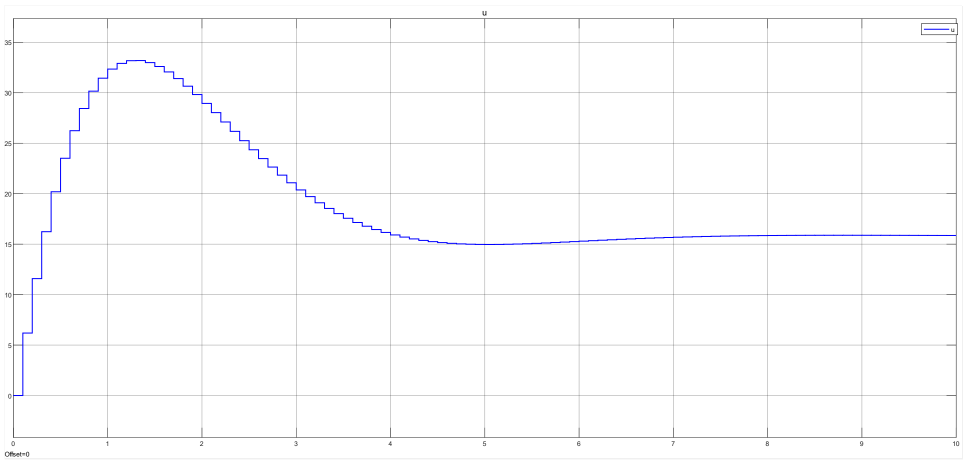


Figure 6 Digital controller action in Simulink

Question 6

By using MATLAB, get the sampling frequency that could be used to analyze the system and design the controller continuous.

Solution

MATLAB Script Calculating the Sampling Frequency

```
clc
clear all
```

Pump and Tank Transfer Function

```
sys = tf(30/31,[1 10/31]);
```

Bandwidth Frequency of the System

```
wb = bandwidth(sys) % The gain of sys drops to 3 dB below its DC value at wb rad/s
```

```
wb = 0.3218
```

Sampling frequency

```
ws = 10*wb          % The minimum sampling frequency that is suitable to analyze the system
```

```
ws = 3.2182
```

Published with MATLAB® R2022a

The continuous controller design MATLAB skript is attached in the following pages.

Table of Contents


```
clc;
clear all
```

Controller Design Criteria

```
M = 0.05;
Ts = 5;
Ess = 0.05;
```

System Desired Characteristics

```
zeta = fzero(@(x) M-exp(-pi*x/sqrt(1-x^2)),0.6);
Wn = 4/Ts/zeta;
```

Systen Definition

Pump and Tank System Transfer Function

$$G\left(s\right)=\frac{A}{\tau s+1}$$

```
tau = 3.1;
A = 3;

s = tf('s');
G = A/(tau*s+1)
```

```
G =

      3
  ---------
  3.1 s + 1

Continuous-time transfer function.
```

Feedback Gain (Sensor and Signal Conditioning)

```
H = 0.01*10;
```

Controller and Closed Loop TF

Assume the controller of the following form: $C(s) = \frac{k(\tau s+1)}{s^2+a_1 s+a_0}$

Forward Path: $G_F(s) = C(s)\,G(s) = \frac{A}{\tau s+1} \cdot \frac{k(\tau s+1)}{s^2+a_1 s+a_0} = \frac{\mathrm{kA}}{s^2+a_1 s+a_0}$

Closed Loop TF: $G_c(s) = \frac{G_F(s)}{1+G_F(s)H(s)} = \frac{\mathrm{kA}}{s^2+a_1 s+(H(s)\mathrm{kA}+a_0)}$

Apply the Design Criteria to Calculate k, $a_1$and a_0

$$E_{\mathrm{ss}} = \lim_{s\to 0} s.\,\frac{1}{s}\,(1 - G_c(s))$$

$$G_c(s) = \frac{\mathrm{kA}}{s^2 + 2\zeta\omega_n\, s + \omega_n^2}$$

```
k = (1-Ess)*Wn^2/A;
a1 = 2*zeta*Wn;
a0 = Wn^2-A*H*k;
```

Controller

```
C = k*(tau*s+1)/(s^2+a1*s+a0)
```

```
C =

    1.233 s + 0.3977
  ---------------------
  s^2 + 1.547 s + 1.137

Continuous-time transfer function.
```

Forward Path

```
GF = C*G
```

```
GF =


          3.699 s + 1.193

  --------------------------------

  3.1 s^3 + 5.795 s^2 + 5.071 s

                         + 1.137


Continuous-time transfer function.
```

Continuous

```
Gc = feedback(GF,H)
```

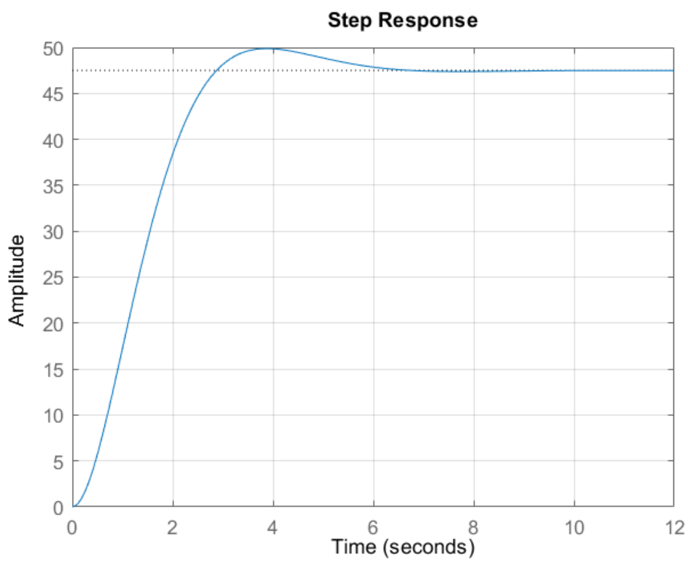
```
Gc =

         3.699 s + 1.193
  --------------------------------
  3.1 s^3 + 5.795 s^2 + 5.441 s
                         + 1.256

Continuous-time transfer function.
```

System Characteristics

```
infoC=stepinfo(Gc);
step(Gc*50);
xlim([0,12])
hold on; grid on; box on;
```



Optimum Sampling Time Computation

```
T=linspace(tau/10,0.01,10);
Gz = cell(length(T),1);  % Forward path in z-domain
Gc = cell(length(T),1);   % Closed loop in z-domain
info = cell(length(T),1); % System characteristics
Error = 1e-2;
Tsampling = 0;

for i=1:length(T)
    Gz{i} = c2d(GF,T(i));
```

```
    Gc{i} = feedback(Gz{i},H);
    info{i} = stepinfo(Gc{i});
    if abs(infoC.SettlingTime-info{i}.SettlingTime)/infoC.SettlingTime<Error && ...
     abs(infoC.Overshoot-info{i}.Overshoot)/infoC.Overshoot<Error
        Tsampling=T(i);
        M = info{i}.Overshoot;
        Ts = info{i}.SettlingTime;
        break
    end
end
```

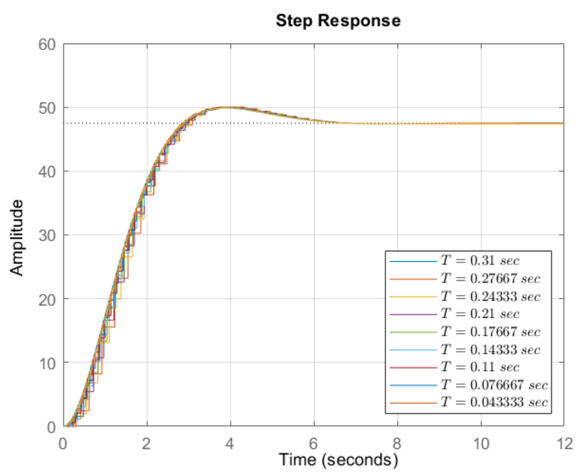
Conclusion

```
if Tsampling==0
    disp('The range set for settling time does not meet the error requirement,
    try smaller settling time');
else
    index = find(T==Tsampling);
    legendList = cell(1,index);
    for i=1:index
    legendList{i}=['$T= ', num2str(T(i)), '\;sec$'];
    step(Gc{i}*50);
    end
    grid on; box on;
    legend(legendList,'interpreter','latex','Location','SouthEast')

    figure
    step(Gc{index}*50);
    legend(legendList{index},'interpreter','latex','Location','SouthEast')
    xlim([0,12])

    fprintf('Sampling time =')
    display(Tsampling);

    fprintf('The continuous controller in s-domain')
    display(C);
    fprintf('The continuous controller in z-domain')
    Cz=zpk(c2d(C,Tsampling,'impulse'));
    display(Cz);
end
```



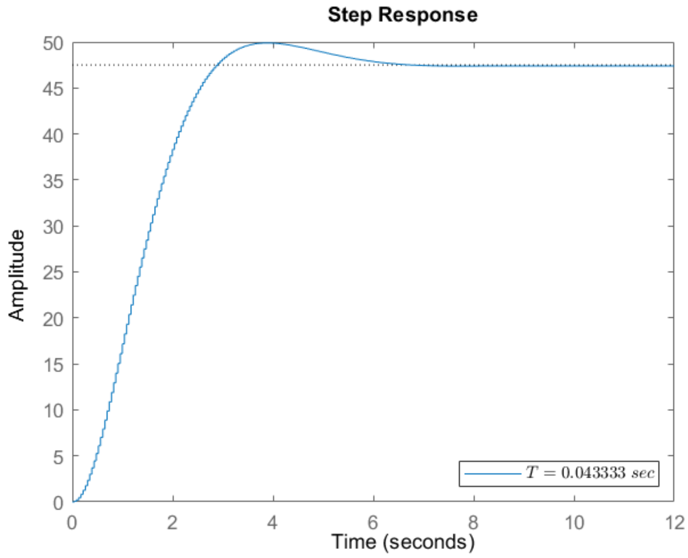


Figure 1: Designed Controller with sampling time 0.0433 sec

```
Sampling time =
Tsampling = 0.0433
The continuous controller in s-domain
C =

    1.233 s + 0.3977
  ---------------------
  s^2 + 1.547 s + 1.137

Continuous-time transfer function.
The continuous controller in z-domain
Cz =

   0.05343 z (z-0.9854)
  -----------------------
  (z^2 - 1.933z + 0.9352)

Sample time: 0.043333 seconds
Discrete-time zero/pole/gain model.
```