ANALYSIS OF A CONTROL SYSTEM USING P,PI,PID CONTROLLERS USING CONTROL SYSTEM ANALYSIS METHODS IN S & Z DOMAIN

PROJECT REVIEW REPORT

(Vth SEMESTER)

PROJECT COMPILED BY:
ADITYA SHARMA-2K18/EE/012
ANANT GUPTA-2K18/EE/025

CERTIFICATE

This is to certify that the project title ***"Analysis of a control system using P,PI,PID controller using control system analysis methods in s & z domain"*** is a record of work done by Aditya Sharma and Anant Gupta students of B.tech (Electrical Engineering), Delhi Technological University as a part of the minor project.This project was carried out under my supervision.

This project is absolutely genuine and does not indulge in plagiarism of any kind. The references taken in making the project have been declared at the end of this report.

ACKNOWLEDGMENT

We would like to express our deep sense of respect and gratitude to our project mentor, Dr Bhavnesh Jaint, Professor, Department of Electrical Engineering, DTU for providing the opportunity of carrying out this project and being the guiding force behind this work.

We are deeply indebted to her for the support, advice and encouragement she provided without which the project could not be executed

ABSTRACT

Engineers design the control systems starting with the performance requirements that the plant or system needs. Then an appropriate controller is chosen and its model along with the plant is used to analyze the system. The control system designs are analyzed using such things as root locus, bode plots, step response plot etc. to evaluate controller configuration and gain values for desired outputs for the system.

Here , we try to develop a MATLAB code is programmed where a suitable gain value of control system is found out for a PID controller using root locus technique so as to get a stable response for the system and then analysing the system for different types of controllers in both S-domain and Z-domain. MATLAB now has developed a number of functions that make it easy to convert one model to another model or to plot different types of plots like bode plots, step response plots etc.

INDEX

INTRODUCTION

Different types of controllers:

P Controller

Proportional controller, in engineering and process control, is a type of linear feedback control system in which a correction is applied to the controlled variable which is proportional to the difference between the desired value (setpoint, SP) and the measured value (process variable, PV). Two classic mechanical examples are the toilet bowl float proportioning valve and the fly-ball governor. Output power is directly proportional to control error.

The higher the proportion coefficient, the less the output power at the same control error. Proportional control can be recommended for fast-response systems with a large transmission coefficient. To adjust the proportional controller you should first set the maximum proportion coefficient where in the output power decreases to zero. When the measured value is stabilized, set a specified value and gradually reduce the proportion coefficient and the control error will decrease. If there are periodic oscillations in the system, the proportion coefficient should be increased so that control error is minimal periodic oscillations decrease to the limit.

A drawback of proportional control is that it cannot eliminate the residual SP − PV error in processes with compensation e.g. temperature control, as it requires an error to generate a proportional output. To overcome this the PI controller was devised, which uses a proportional term (P) to remove the gross error, and an integral term (I) to eliminate the residual offset error by integrating the error over time to produce an "I" component for the controller output. The proportional controller helps in reducing the steady-state error, thus makes the system more stable.

PI Controller

The PI controller is the most popular variation, even more than full PID controllers. Output power equals to the sum of proportion and integration coefficients. A PI controller (proportional-integral controller) is a special case of the PID controller in which the derivative (D) of the error is not used.The lack of derivative action may make the system more steady in the steady state in the case of noisy data. This is because derivative action is more sensitive to higher-frequency terms in the inputs.Without derivative action, a PI-controlled system is less responsive to real (non-noise) and relatively fast alterations in state and so the system will be slower to reach setpoint and slower to respond to perturbations than a well-tuned PID system may be.

The higher the proportion coefficient, the less the output power at the same control error. The higher the integration coefficient, the slower the accumulated integration coefficient. PI control provides zero control error and is insensitive to interference of the measurement channel. The PI control disadvantage is slow reaction to disturbances.

To adjust the PI controller you should first set the integration time equal to zero, and the maximum proportion time. Then by decreasing the coefficient of proportionality, achieve periodic oscillations in the system. Close to the optimum value of the coefficient of proportionality is twice higher than that at which any hesitation, and close to the optimum value of the integration time constant - is 20% less than the oscillation period. It is needed for non-integrating processes, meaning any process that eventually returns to the same output given the same set of inputs and disturbances

PID Controller

A proportional–integral–derivative controller (PID controller or three-term controller) is a control loop mechanism employing feedback that is widely used in industrial control systems and a variety of other applications requiring continuously modulated control. A PID controller continuously calculates an *error value* {\displaystyle e(t)}e(t) as the difference between a desired setpoint (SP) and a measured process variable (PV) and applies a correction based on proportional, integral, and derivative terms (denoted *P*, *I*, and *D* respectively), hence the name.

Output power equals to the sum of three coefficients: proportional, integral and differential. The higher the proportion coefficient, the less the output power at the same control error. The higher the integration coefficient, the slower the accumulated integration coefficient. The higher the differentiation coefficient, the greater the response of the system to the disturbance.

The PID controller is used in inertial systems with relatively low noise level of the measuring channel. The advantage of PID is fast warm up time, accurate setpoint temperature control and fast reaction to disturbances. Manual tuning PID is extremely complex, so it is recommended to use the autotune function. PID controllers, when used alone, can give poor performance when the PID loop gains must be reduced so that the control system does not overshoot, oscillate or hunt about the control setpoint value. They also have difficulties in the presence of non-linearities, may trade-off regulation versus response time, do not react to changing process behavior (say, the process changes after it has warmed up), and have lag in responding to large disturbances.

PROBLEM STATEMENT

The calculations for control system parameters sometimes become very tedious, time taking and complex to be always done by hand and now in this fast moving world , where different control systems are developed by engineers everyday , so we tried to develop a MATLAB code so as to create an easiness in choosing controllers etc. for the system so as to create an easiness in developing control systems for engineers.

Here , a PID controller is being designed for a system whose plant function is given. The block diagram is designed showing the gain, controller function block and plant function block with unity feedback path. We need to select a gain so as to get the system to give a stable well damped response for the PID controller. Then for the same gain , the control system is analyzed for P & PI controller for bode plot , step response etc. in s-domain and z-domain.

BASIC VALUES USED IN THE EXAMPLE TAKEN FOR THE CONTROL SYSTEM FOR THIS PROJECT

PLANT FUNCTION:

Transfer function: 1/((s+2)(s+6))= 1/(s^2+8*s+12)

Zeros of the plant function: none

Poles of the plant function: a= -2;

b= -6;

CONTROLLER FUNCTION(PID):

Transfer function: (s^2+5*s+15.25)/s

Different gains as derived from basic formula of PID controller :

Kp=5;

Kd=1;

Ki=15.25; \

Basic equation of PID controller:

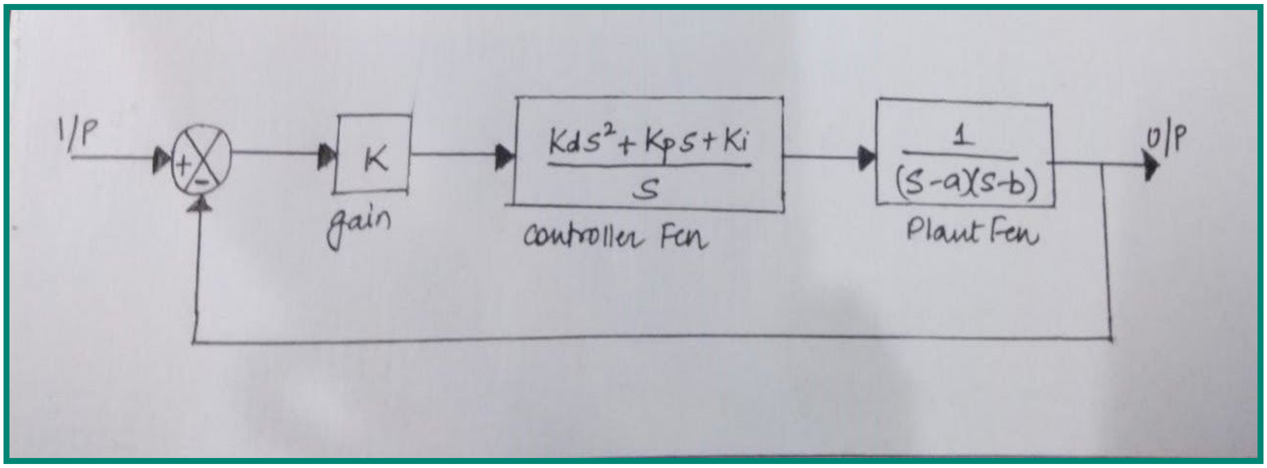
Kp+Ki/s+Kd*s = (Kd*s^2+Kp*s+Ki)/s

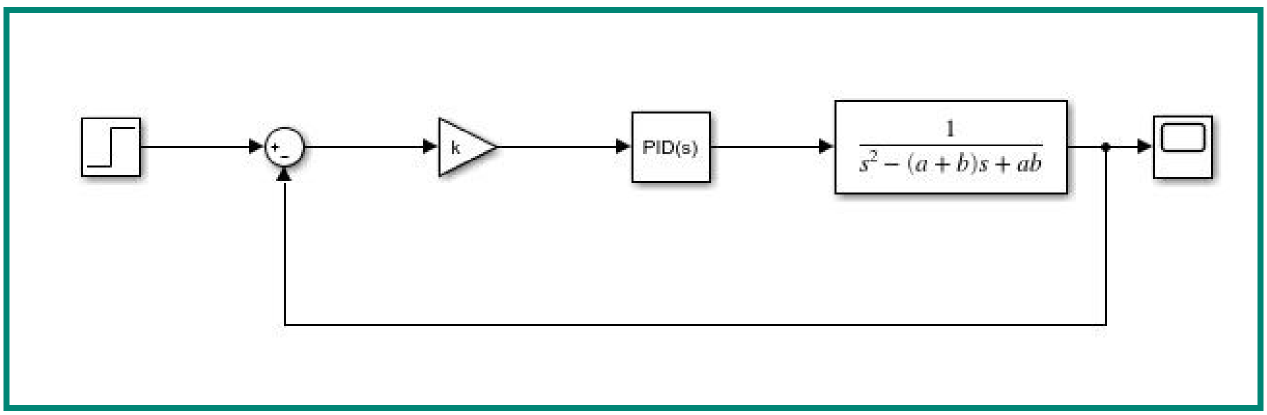
Zeros of PID controller: -2.5-3*j; -2.5+3*j;

Poles of PID controller: 0

Value of Gain block: K

BLOCK DIAGRAMS OF THE EXAMPLE FOR THE CONTROL SYSTEM USED IN THIS PROJECT:





COMPLETE MATLAB CODE USED:

```
%defining plant fcn

>> num=[1];

>> den=[1 8 12];

>> p=tf(num,den);

p

p =

      1

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

  s^2 + 8 s + 12

Continuous-time transfer function.

>> %defining values of controller fcn

>> kp=5;

>> kd=1;

>> ki=15.25;

>> %defining value of poles used

>> a=-2;

>> b=-6;
```

```
>> ztf=[-2.5-3*j;-2.5+3*j];

>> ptf=[0;a;b];

>> [numtf,dentf]=zp2tf(ztf,ptf,1);

>> rlocus(numtf,dentf);

>> [k,poles]=rlocfind(numtf,dentf);

Select a point in the graphics window

selected_point =-1.1600e+02 + 4.4409e-16i

>> k

k = 112.8389

>> %defining pid controller now

>> c1=pid(kp,ki,kd);

>> tf1=feedback(k*c1*p,1);

>> tf1

tf1 =

     112.8 s^2 + 564.2 s + 1721

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

  s^3 + 120.8 s^2 + 576.2 s + 1721

>> ts=0.1;
```

```
>> z1=c2d(tf1,ts);

>> z1

z1 =

     0.9855 z^2 - 1.468 z + 0.5987

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

  z^3 - 1.5 z^2 + 0.6164 z - 5.65e-06

Sample time: 0.1 seconds

>> step(tf1,z1);

>> step(z1);

>> step(tf1,z1);

>> bode(tf1,z1);

>> nyquist(tf1,z1);

>> i1=stepinfo(tf1);

>> i2=stepinfo(z1);

>> %defining pi controller

>> c2=pid(kp,ki);

>> tf2=feedback(c2*p,1);

>> tf2
```

```
tf2 =

        5 s + 15.25

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

  s^3 + 8 s^2 + 17 s + 15.25

>> z2=c2d(tf2,ts);

>> z2

z2 =

  0.0213 z^2 + 0.002369 z - 0.01332

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

  z^3 - 2.328 z^2 + 1.788 z - 0.4493

Sample time: 0.1 seconds

>> step(tf2,z2);

>> bode(tf2,z2);

>> nyquist(tf2,z2);

>> i3=stepinfo(tf2);

>> i4=stepinfo(z2);

>> %defining p controller

>> c3=pid(kp);
```

```
>> tf3=feedback(c3*p*k,1);

>> tf3

tf3 =

        564.2

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

  s^2 + 8 s + 576.2

>> z3=c2d(tf3,ts);

>> z3

z3 =

     1.371 z + 0.9866

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

  z^2 + 0.958 z + 0.4493

Sample time: 0.1 seconds

>> step(tf3,z3);

>> bode(tf3,z3);

>> nyquist(tf3,z3);

>> i5=stepinfo(tf3);

>> i6=stepinfo(z3);
```

FINDING VALUE OF K(GAIN) WITH USE OF PID CONTROLLER

For finding the value of gain to get a stable system , the root locus of the system was plotted in the MATLAB code, and the poles of the system were chosen close to the zeros so that at that gain poles and zeros cancel. On choosing we got the values K=112.8389 and poles=(-116,-2.42+3*j,-2.42-3*j). It can clearly be seen that on choosing gain as K=112.8389, the complex poles cancel each other and the poles =-116 is very far, so it will give a stable system with very fast response time.

MATLAB code used:

```
ztf=[-2.5-3*j;-2.5+3*j];

ptd=[0;a;b];

[numtf , dentf]=zp2tf(ztf , ptf , 1);

rlocus(numtf ,dentf);

[K , poles]=rlocfind(numtf , dentf);

K
```

Point chosen: -1.1600e+02 + 4.4409e-16i

Value of K found out : 112.8389

RESPONSE WITH PID CONTROLLER

The output of a PID controller, which is equal to the control input to the plant, is calculated in the time domain from the feedback error as follows:

$$u(t) = K_p e(t) + K_i \int e(t)dt + K_p \frac{de}{dt}$$

First, let's take a look at how the PID controller works in a closed-loop system using the schematic shown above. The variable (e) represents the tracking error, the difference between the desired output (r) and the actual output (y). This error signal (e) is fed to the PID controller, and the controller computes both the derivative and the integral of this error signal with respect to time. The control signal (u) to the plant is equal to the proportional gain (K_p) times the magnitude of the error plus the integral gain (K_i) times the integral of the error plus the derivative gain (K_d) times the derivative of the error.This control signal (u) is fed to the plant and the new output (y) is obtained. The new output (y) is then fed back and compared to the reference to find the new error signal (e). The controller takes this new error signal and computes an update of the control input. This process continues while the controller is in effect.

EQUATIONS OF PID CONTROLLER: Kp+Ki/s+Kd*s = (Kd*s^2+Kp*s+Ki)/s

EQUATIONS OF TRANSFER FUNCTION OF THE SYSTEM:

(K*Kd*s^2+K*Kp*s+K*Ki)/(s^3+(K*Kd-(a+b))*s^2+(K*Kp+a*b)*s+K*Ki)

TRANSFER FUNCTIONS IN BOTH S-DOMAIN AND Z-DOMAIN

Transfer function of controller:

c1= (s^2+5*s+15.25)/s

Overall transfer function of PID system:

tf1=(112.8*s^2+564.2*s+1721)/(s^3+120.8*s^2+576.2*s+1721)

z1=(0.9855*z^2-1.4682*z+0.59)/(z^3-1.5*z^2+0.6164*z-5.65e-06)

BLOCK DIAGRAMS OF THE PID SYSTEM:

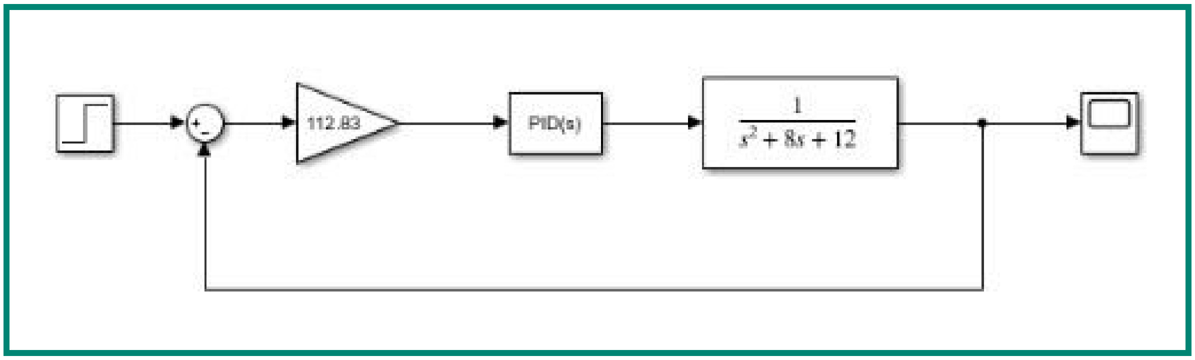


fig1. Equivalent control system circuit for pid control system

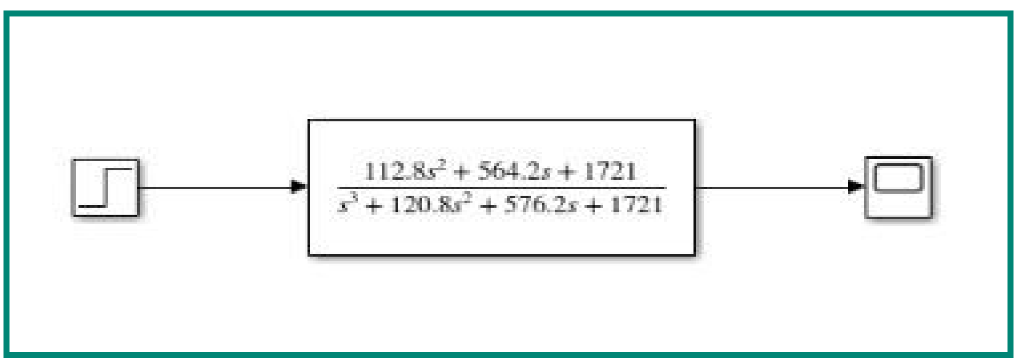
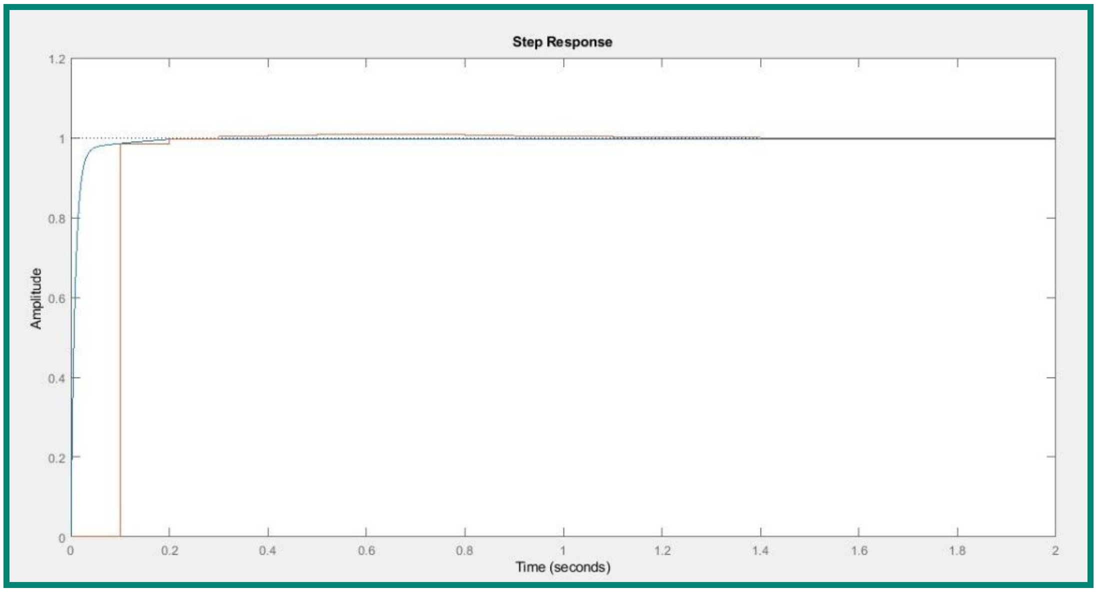
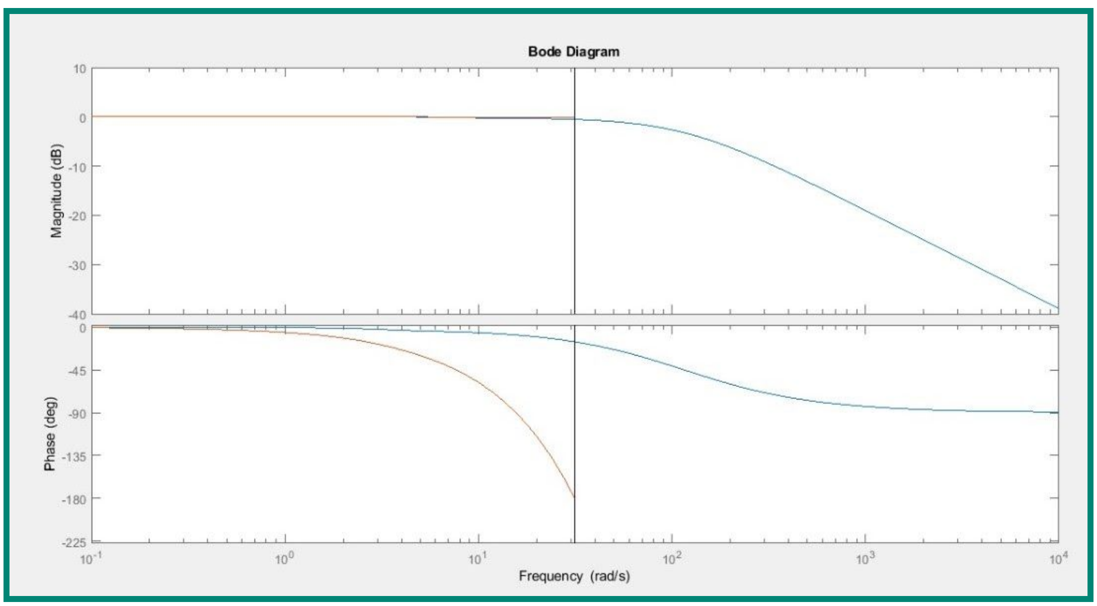
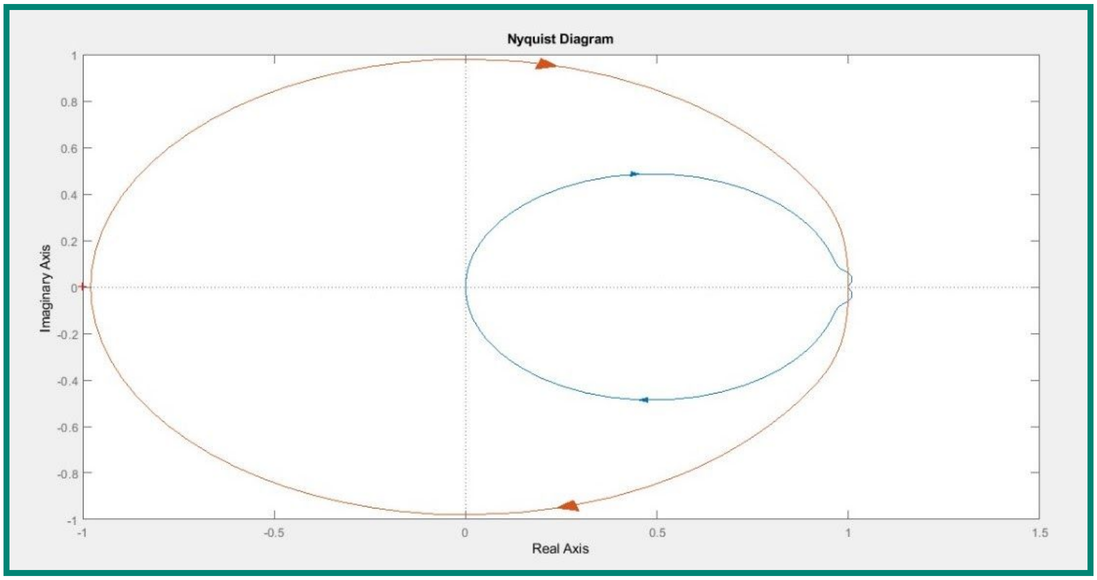


fig2. Equivalent open loop circuit for fig1 circuit

GRAPH RESPONSES OF PID CONTROLLER







INFORMATION ACHIEVED THROUGH PID SYSTEM RESPONSES

PROPERTY	S-DOMAIN	Z-DOMAIN
Rise time	0.0212	0
Settling time	0.0628	0.100
Settling min.	0.9011	0.9855
Settling max.	0.9964	1.0084
Overshoot	0	0.8363
Undershoot	0	0
Peak	0.9964	1.0084
Peak time	0.2060	0.600

RESPONSE WITH PI CONTROLLER

A P.I Controller is a feedback control loop that calculates an error signal by taking the difference between the output of a system, which in this case is the power being drawn from the battery, and the set point.Before proceeding to PID control, let's investigate PI control. From the table, we see that the addition of integral control (K_i) tends to decrease the rise time, increase both the overshoot and the settling time, and reduce the steady-state error.

ASSUMPTION:-

We have not taken into account the value of gain(K) for this controller for the simple reason being that proportional gain (Kp) also reduces the rise time and increases the overshoot as does the P-controller. Since we already do not have a very large Kp and Ki , we need to neglect the effect of K since it is very large and does not affect the flow of responses.

EQUATIONS OF PI CONTROLLER:- Kp+Ki/s = (Kp*s+Ki)/s

EQUATIONS OF TRANSFER FUNCTION OF THE SYSTEM:-

(Kp*s+Ki)/(s^3-(a+b)*s^2+(a*b+Kp)*s+Ki)

TRANSFER FUNCTIONS IN BOTH S-DOMAIN AND Z-DOMAIN

Transfer function of controller:

c2 = 5+(15.25/s) = (5*s+15.25)/s

Overall transfer function of PID system:

tf2 = (5*s+15.25)/(s^3+8*s^2+17*s+15.25)

z2 = (0.0213*z^2+0.002369*z-0.01332)/(z^3-2.328*z^2+1.788*z-0.4493)

BLOCK DIAGRAMS OF THE PI SYSTEM:

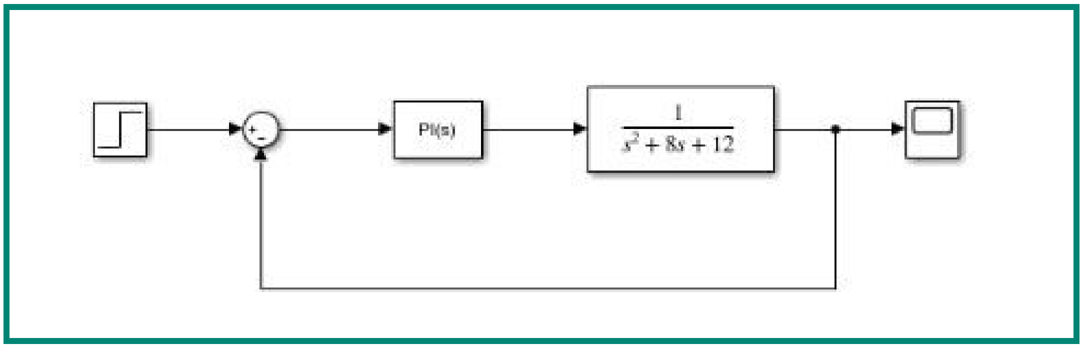


Fig3. Equivalent control system circuit for pi control system

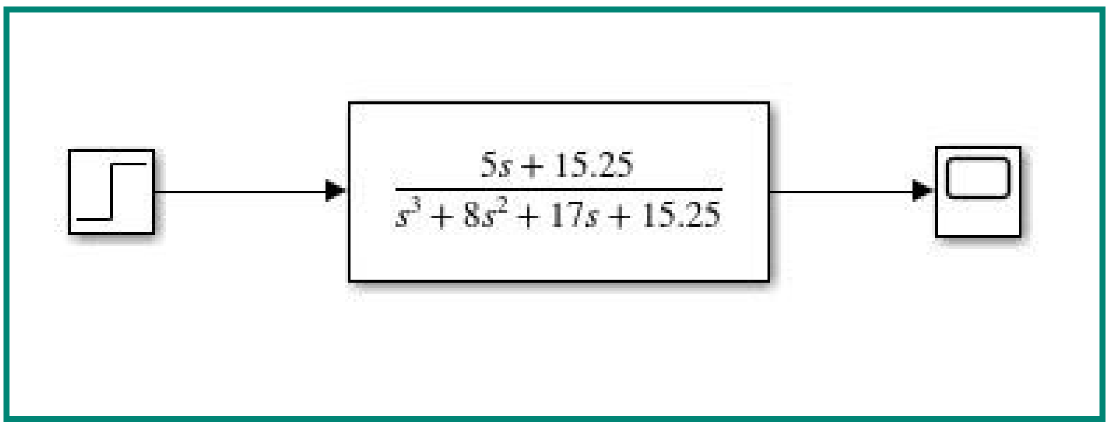
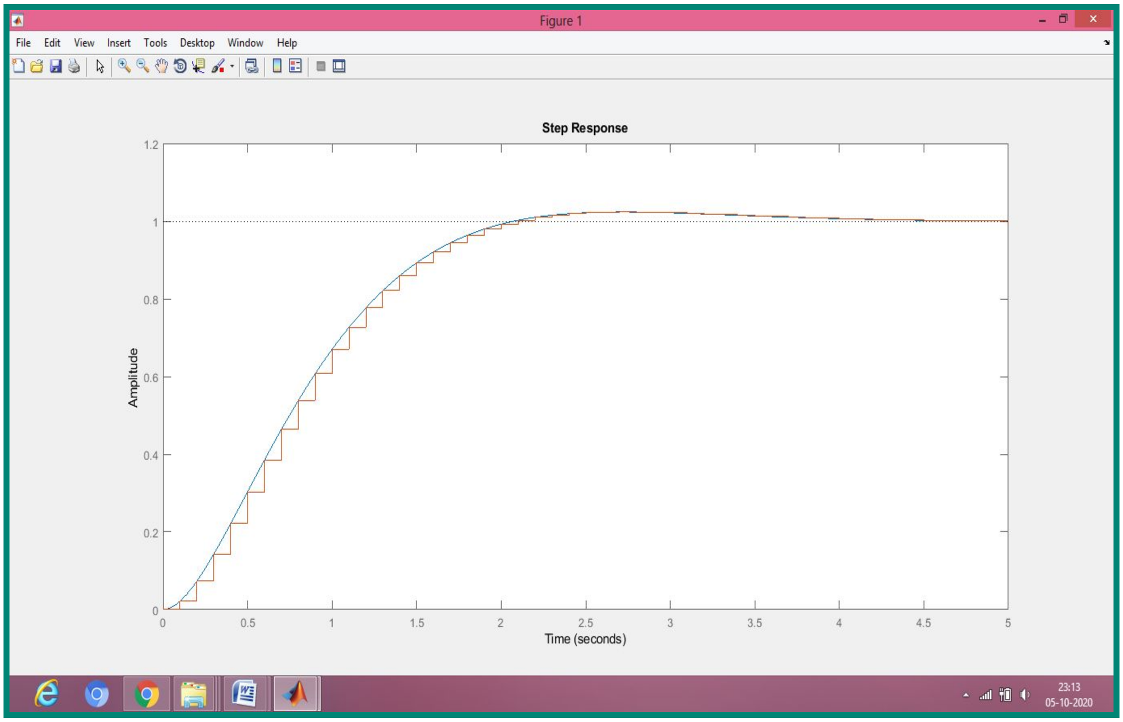
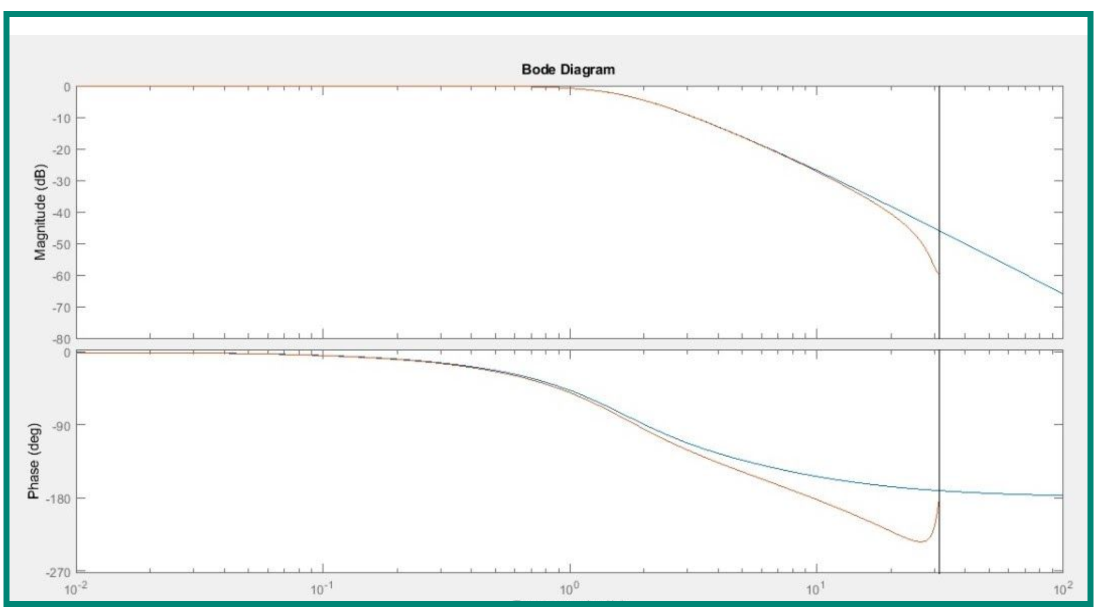
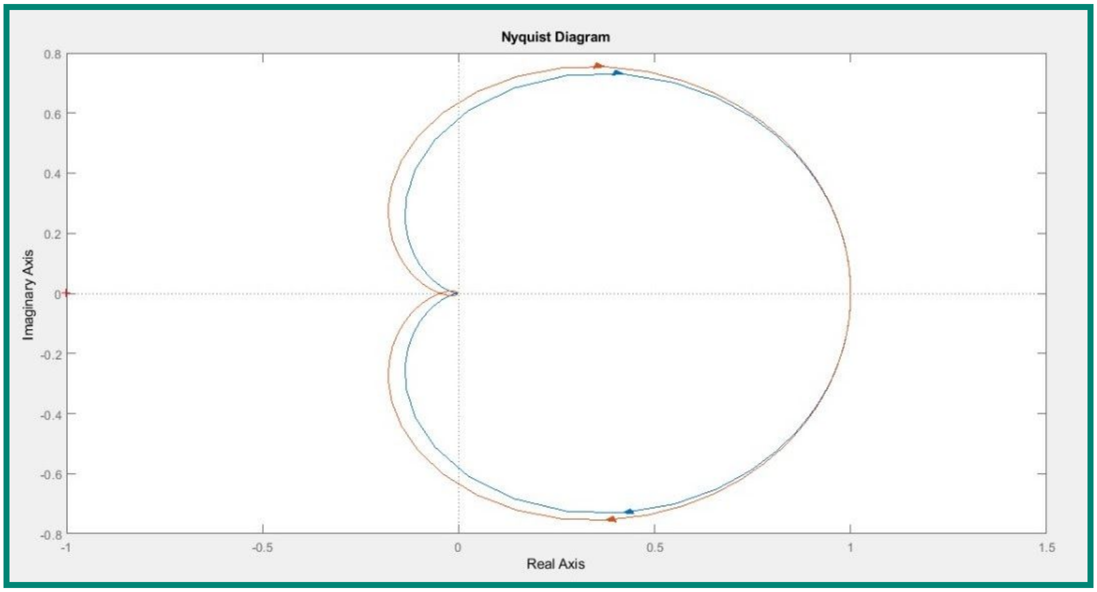


Fig4. equivalent open loop circuit for pi controller system

GRAPH RESPONSES OF PI CONTROLLER







INFORMATION GAINED THROUGH PI SYSTEM RESPONSES

PROPERTY	S-DOMAIN	Z-DOMAIN
Rise time	1.2840	1.300
Settling time	3.1049	3.200
Settling min.	0.9029	0.9208
Settling max.	1.0236	1.0236
Overshoot	2.3591	2.3583
Undershoot	0	0
Peak	1.0236	1.0236
Peak time	2.7198	2.7

RESPONSE WITH P CONTROLLER

Proportional control, in engineering and process control, is a type of linear feedback control system in which a correction is applied to the controlled variable which is proportional to the difference between the desired value and the measured value.P controller is mostly used in first order processes with single energy storage to stabilize the unstable process. The main usage of the P controller is to decrease the steady state error of the system. As the proportional gain factor K increases, the steady state error of the system decreases.It is the simplest form of continuous control that can be used in a closed-looped system. P-only control minimizes the fluctuation in the process variable, but it does not always bring the system to the desired set point. It provides a faster response than most other controllers, initially allowing the P-only controller to respond a few seconds faster. However, as the system becomes more complex.

EQUATIONS OF P CONTROLLER:- Kp

EQUATIONS OF TRANSFER FUNCTION OF THE SYSTEM:-

(Kp*K)/(s^2-(a+b)*s+(a*b+Kp*K))

TRANSFER FUNCTIONS IN BOTH S-DOMAIN AND Z-DOMAIN

Transfer function of controller:

c3 = 5;

Overall transfer function of P system:

tf3 = (564.2)/(s^2+8*s+576.2)

z3 = (1.371*z+0.9866)/(z^2+0.958*z+0.4493)

BLOCK DIAGRAMS OF THE P SYSTEM:

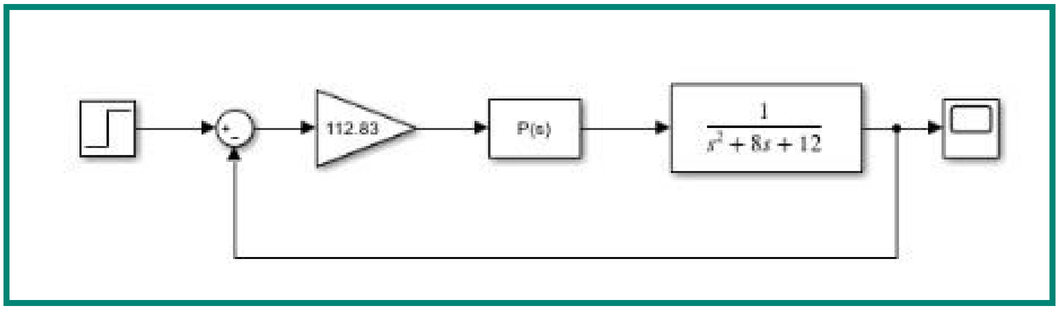


fig1. Equivalent control system circuit for p control system

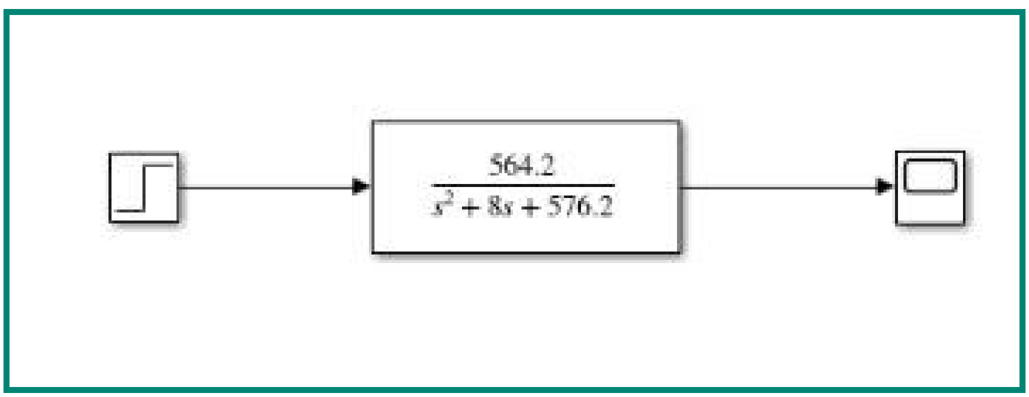
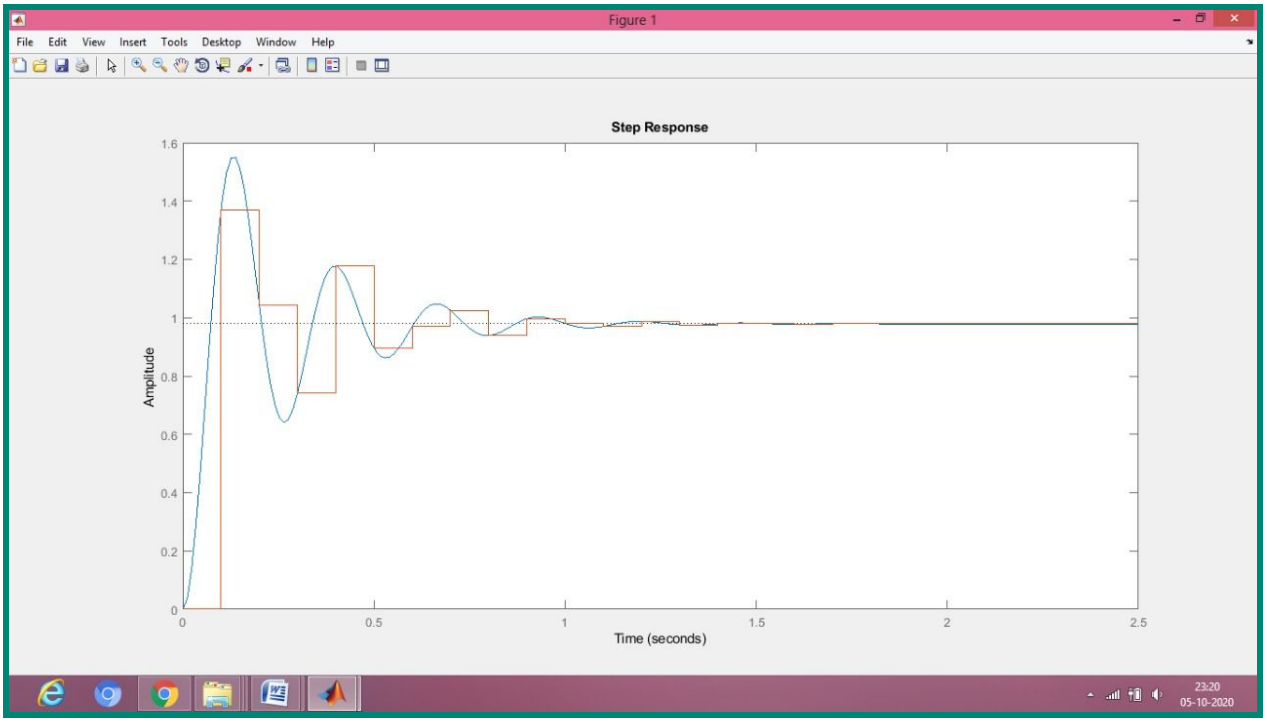
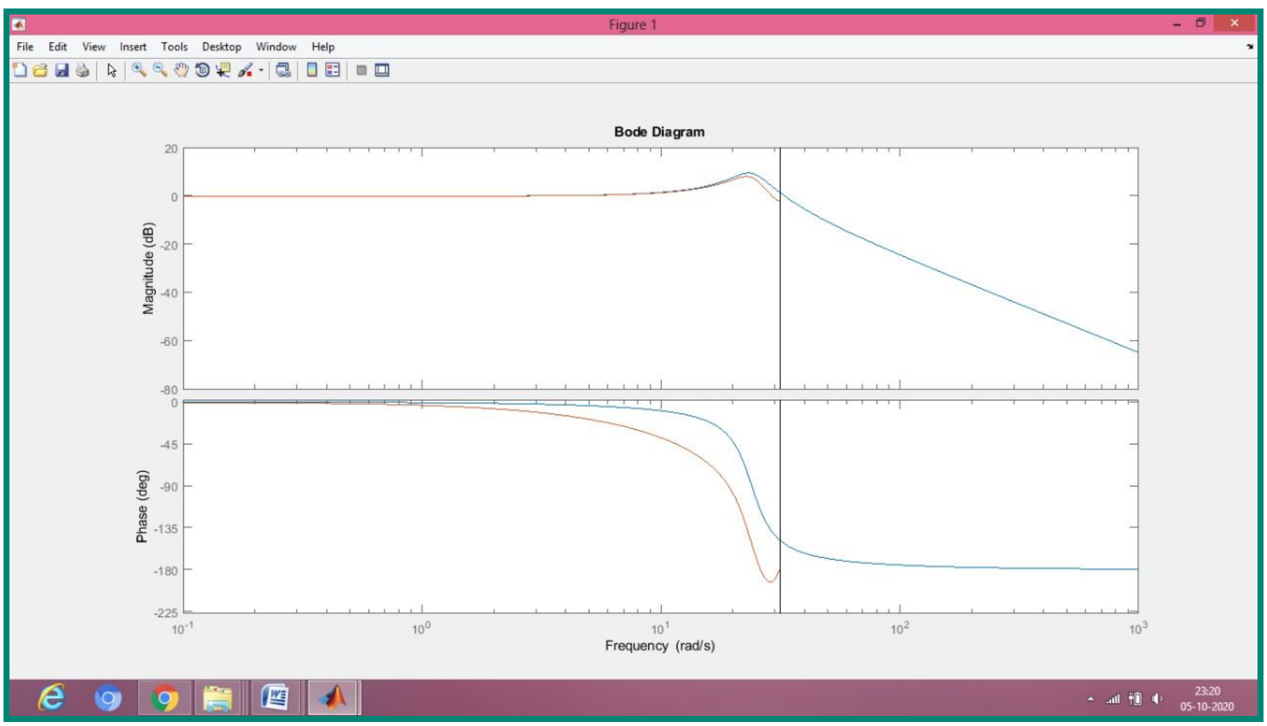
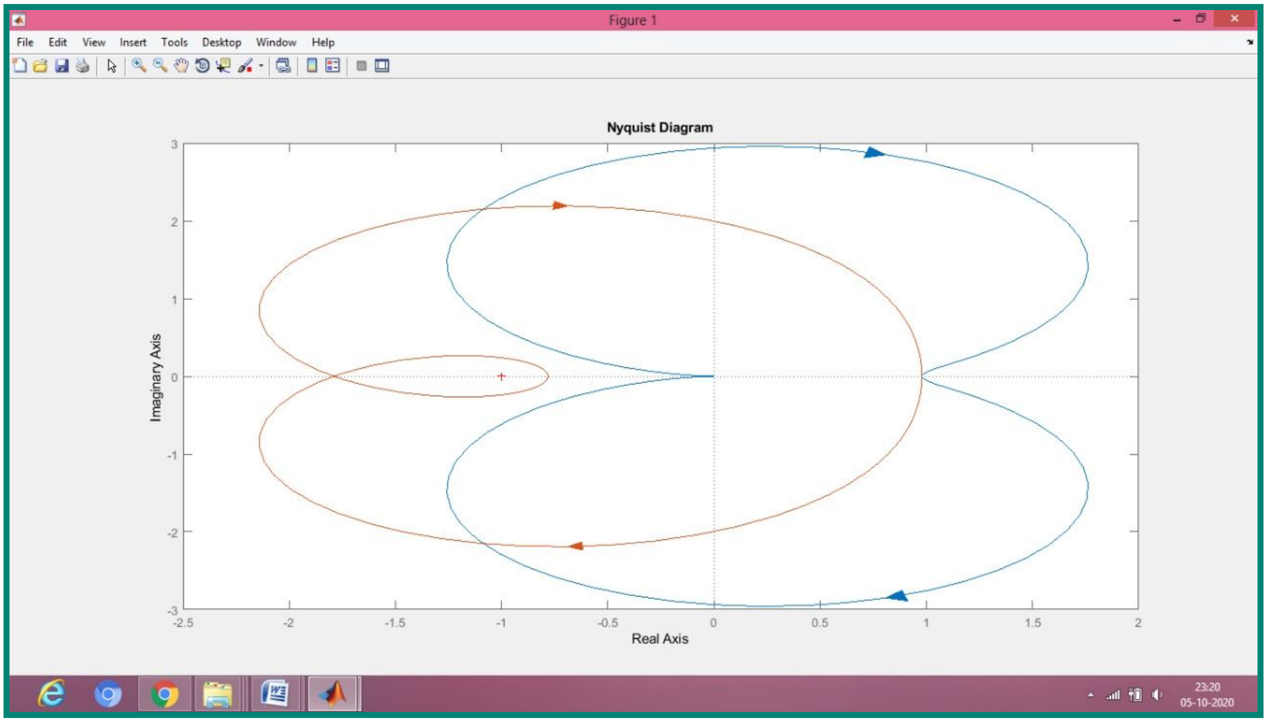


fig2. Equivalent open loop circuit for fig1 circuit

GRAPH RESPONSES OF P CONTROLLER







INFORMATION GAINED THROUGH P SYSTEM RESPONSES

PROPERTY	S-DOMAIN	Z-DOMAIN
Rise time	0.0494	0
Settling time	0.9552	0.9000
Settling min.	0.6402	0.7410
Settling max.	1.5502	1.3706
Overshoot	58.3154	39.9761
Undershoot	0	0
Peak	1.5502	1.3706
Peak time	0.1382	0.1000

CONCLUSION/RESULT(COMPARISONS)

❑ PID CONTROLLER:

- ❑ Proportional control improves rise time.
- ❑ Derivative control reduces overshoot.
- ❑ Integral control reduces steady state error.
- ❑ So, response plotted and the information gathered we have plotted a plot with no overshoot and very fast rise time and very little steady state error.

❑ PI CONTROLLER:

- ❑ With proportional control and addition of integral control(Ki) there is a trend to decrease rise time, increase both overshoot and settling time.
- ❑ Pi controllers generally have reduced steady state error.
- ❑ For the PI controller here we have not taken into account value of K(gain) as mentioned in the assumption in the PI controller because Kp and Ki both reduce rise time and increase the overshoot(double effect)
- ❑ The response plotted shows that Ki eliminated the steady state error in this case.

❑ P CONTROLLER:

- ❑ Kp reduces rise time.
- ❑ It also increases the overshoot.
- ❑ Kp reduces steady state error.
- ❑ The plot shows a reduced rise time and the steady state error and increased overshoot and decreased settling time by a small amount.

❑ DIFFERENCES IN S-DOMAIN AND Z-DOMAIN

- ❑ The z domain **is** the discrete S domain where by definition Z= exp **S** Ts with Ts **is** the sampling time. ... Also the discrete time functions and systems can be easily mathematically described and synthesized in the Z-domain exactly like the S-domain for continuous time systems and signals.

REFERENCES

- Control System Engineering- I.J Nagrath & M.Gopal
- https://ctms.engin.umich.edu/CTMS/index.php?example=Introduction&section=ControlPID
- https://in.mathworks.com/help/control/ug/creating-discrete-time-models.html