

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DEPARTAMENTO ACADÊMICO DE ENGENHARIA ELETRÔNICA

ELEC20 – S71 - (1/2024)

FÁBIO ZHAO YUAN WUANG - 2358310

VICTOR AUGUSTO DEL MONEGO – 2378345

Este relatório tem como propósito responder as questões feitas no trabalho final da disciplina Controle 1. A dupla Fábio/Victor possui número **p** igual a 5, e os enunciados já foram ajustados para tal:

A figura 01 abaixo representa o sistema a ser analisado para todas as questões:

$$G(s) = \frac{10p}{s(s+p)(0.1s+p)}$$

*Figura 1: Sistema a ser tratado*

- 1) **Faça uma análise em malha aberta para este sistema, indicando:**
  - a) **Polos e zeros. A partir do diagrama de lugar das raízes, comente sobre a resposta temporal esperada em malha fechada dados os polos dominantes.**
  - b) **Utilizando um controlador proporcional determine utilizando o critério de Routh-Hurwitz, o ganho K do controlador, a partir do qual o sistema se torna instável. Discuta a estabilidade do sistema.**
  - c) **Qual o tipo de sistema e sua constante de erro associada. É possível reduzir o erro associado á metade, utilizando o controle proporcional sem que o sistema perca a estabilidade?**

O código abaixo foi utilizado para responder à questão 1:

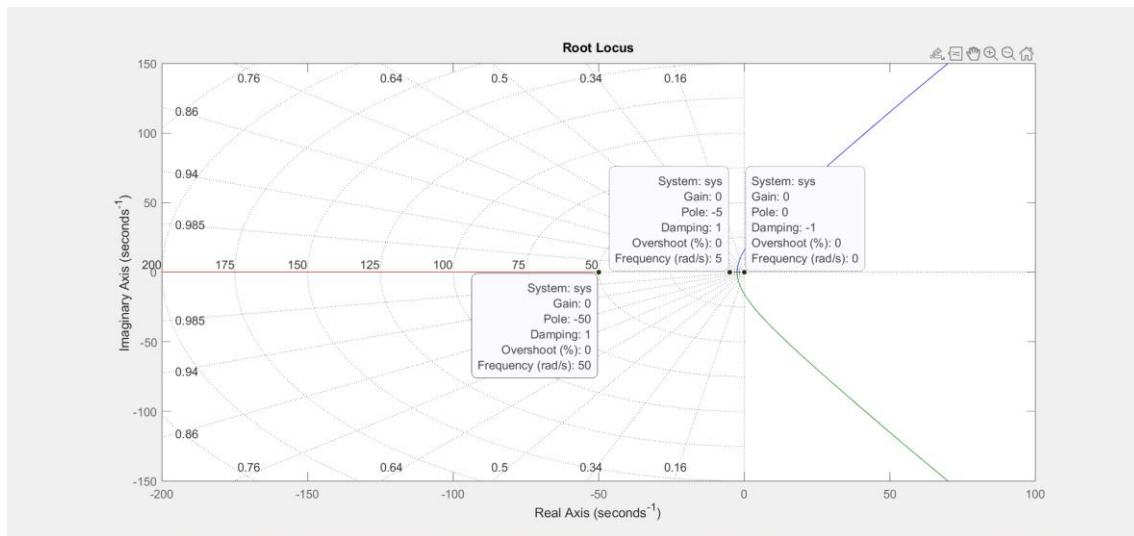
```
% comandos iniciais
clear all
close all
clc
% declarando o sistema
s = tf('s');
sys = (10*5)/(s*(s+5)*(0.1*s+5))
zpk(sys)
% polos e zeros
pole(sys)
%lugar das raízes
figure(1);
rlocus(sys)
%Critério de estabilidade de routh Hurwitz
K = 1;
```

```

K_crit = 27.49;
sys_fb = feedback((K_crit)*sys,1)
figure(2)
step(sys_fb)
%Constante de erro
%sistema é do tipo 1 (possui polo na origem)
Ktst = K_crit/2
Kv = 10*Ktst*5/(5*5)
errv = 1/Kv

```

Abaixo estão anexos os polos, e o diagrama de lugar das raízes.



Analisando o diagrama, e considerando o polo em zero e em -50, é de se esperar que o polo em -50 defina o ganho crítico do sistema, bem como as suas demais propriedades e, portanto, sua estabilidade, por estar mais próximo do eixo imaginário.

A obtenção do K crítico (K de Routh), foi feita pela tabela de Routh, ilustrada no PDF de apoio “Trabalho Final – Obtenção do K de Routh”. Realizando os cálculos, o sistema se torna instável a partir de  $K = 27,5$ . Partindo deste ponto, a conclusão que podemos tirar é que o sistema possui uma boa margem de escalabilidade quando se trata de estabilidade, visto que existe uma boa janela de ganho que o sistema pode receber antes de se desestabilizar.

Os cálculos para este sistema deram um erro aproximado de 0,037, e considerando a redução deste erro pela metade utilizando o cálculo proporcional, o trecho de código abaixo ilustra esta possibilidade:

```

%Constante de erro
%sistema é do tipo 1 (possui polo na origem)
Ktst = K_crit/2
Kv = 10*Ktst*5/(5*5)
errv = 1/Kv

% Visto que Kv é proporcional ao Ktst, e
% Kv é inversamente proporcional ao errv
% então, o único jeito de reduzir o erro associado à

```

```

% metade é dobrando o Kv, isto é, temos de
% dobrar o Ktst. Portanto, se reduzimos o erro à metade
% temos de dobrar o Ktst (I). Tendo (I) em mente:

% Para controladores com  $K_{tst} < K_{routh}/2$ ,
% ao dobrar o valor de Ktst, garantimos a estabilidade
% já que  $2*K_{tst} < K_{routh}$ , e, de (I)
% conseguimos reduzir o erro à metade.

% Para controladores com  $K_{routh}/2 < K_{tst} < K_{routh}$ ,
% ao dobrarmos Ktst, afim de reduzir a metade
% o erro associado, não garantimos a
% estabilidade, já que  $K_{routh} < 2*K_{tst}$ .

```

**2) Projetar um controlador utilizando resposta em frequência tal que a margem de fase seja de  $(50 \pm 1)^\circ$ . Este controlador é de avanço ou atraso de fase. Justifique**

O código abaixo foi utilizado para responder à questão 02:

```

%declarando o sistema
s = tf('s');
sys = (10*5)/(s*(s+5)*(0.1*s+5));

%ganho arbitrário
K = 4.53;
%margem máxima usando a margem desejada
phi = 50 + 5;
%diagrama de Bode do sistema descompensado
figure(1)
margin(sys)

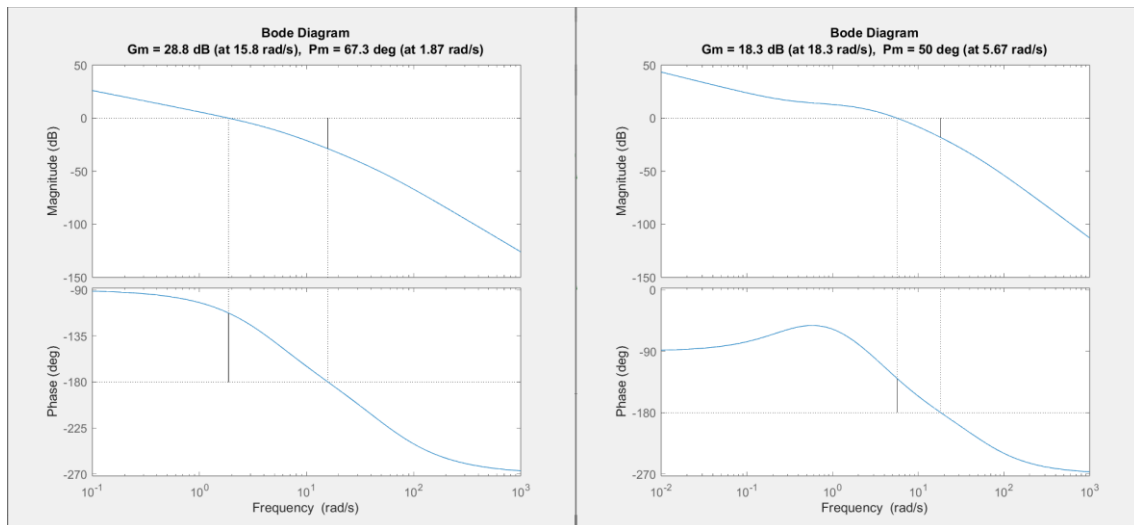
%determinação dos parâmetros de fase
alph = 10^(5/20);
tau = 1/(0.1*3.03);
z = 1/tau;
p = 1/alph*tau;

%Sistema compensado com controlador de atraso de fase
figure(2)
D = K*(s+z)/(s+p);
margin(D*sys)

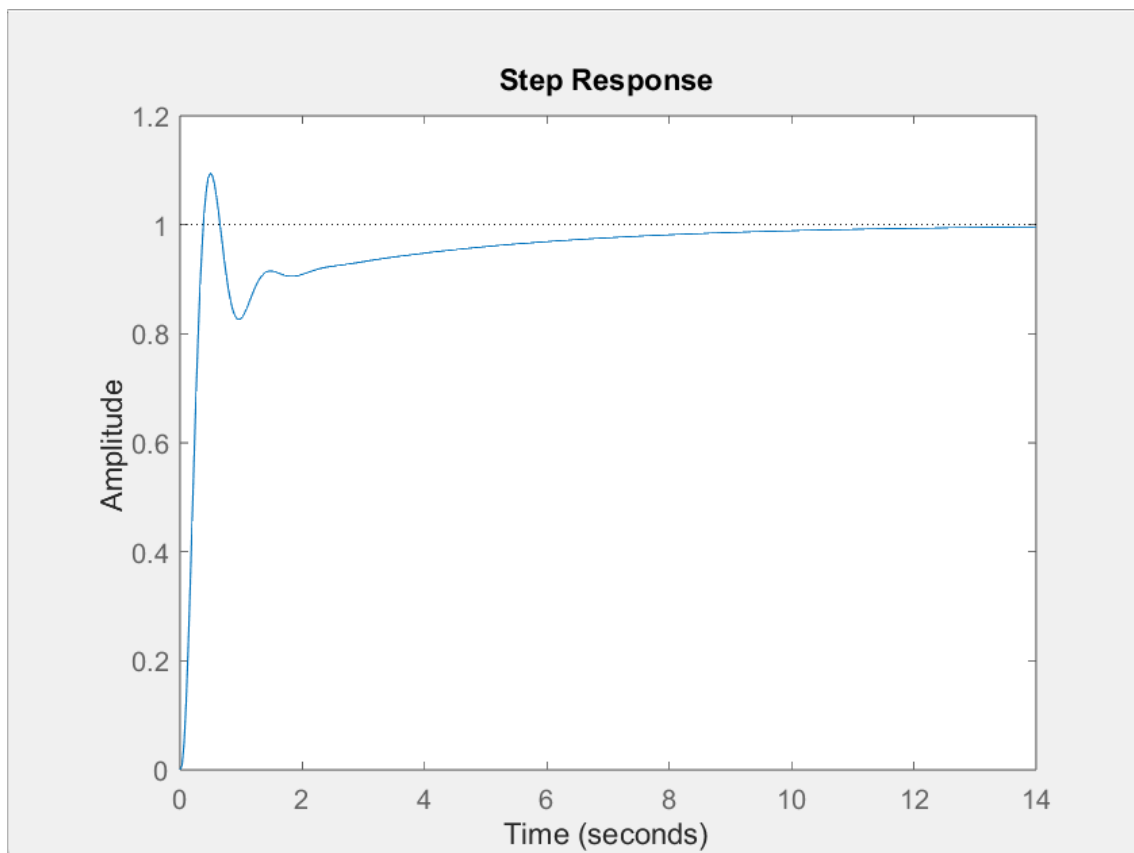
%teste de sobressalto
figure(3)
step(feedback(D*sys,1))
stepinfo(feedback(D*sys,1))

```

Abaixo, os gráficos de bode do sistema descompensado, bem como o sistema após a compensação de fase:



O approach utilizado foi um controlador por atraso de fase, pois era cômodo que a porcentagem de overshoot fosse baixa. Tendo isso em mente, definindo um overhead de 7 graus para a margem desejada, e ajustando o ganho a fim de tornar a margem precisa chegamos em um valor de  $K = 4,53$ . A justificativa para utilizar um controlador de atraso de fase é ilustrada abaixo:



A resposta ao degrau do sistema manteve um overshoot relativamente baixo de 9,5%, o que é um efeito característico de um controlador de atraso de fase

**3) Projete um controlador por alocação de polos utilizando o lugar das raízes tal que o tempo de assentamento seja menor que 0,2 segundos e a taxa de amortecimento seja de 0,5. Comente o projeto e justifique os resultados.**

O código abaixo foi utilizado para responder à questão 3:

```
clear all
close all
clc

%declaração do sistema
s = tf('s');
sys = (10*5)/(s*(s+5)*(0.1*s+5));

%lugar das raízes
rlocus(sys)
grid on;
hold on;

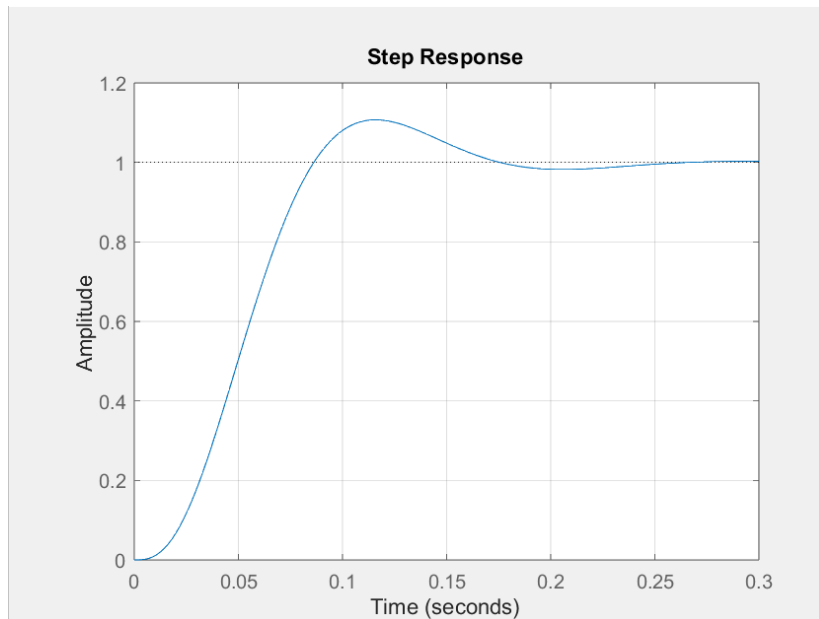
% Polos desejados
polos_desejados = [-20 + 34.64i, -20 - 34.64i];
plot(real(polos_desejados), imag(polos_desejados), 'rx',
'MarkerSize', 10, 'LineWidth', 2)

%matrizes na forma canônica de controlabilidade
A = [0 1 0 ; 0 0 1 ; 0 -25 -5.5];
B = [0 ; 0 ; 1];
C = [50 0 0];

%Operações matriciais para construção do sistema (slides aula 16)
P = [-20+1i*34.64 -20-1i*34.64 -50.5];
K= place(A,B,P);
I=eye(size(A));
MJ=C*inv(-A+B*K)*B;
J=1/MJ;

%controlador feito
controller=ss(A-B*K, B*J,C, []);
figure(2)
step(controller)
hold on;
grid on;
stepinfo(controller)
damp(controller)
```

O PDF de apoio “Trabalho Final – Primeira Tentativa de Projeto de Alocação de Polos” ilustra o raciocínio (apresentado na aula 16) de projeto de controlador utilizando realimentação de estado. O código do Matlab apresenta um polo real mais próximo do polo mais distante do sistema base, para um melhor overshoot. Tendo estas coisas em mente, as imagens abaixo ilustram a resposta ao degrau, os dados de respostas, e o fator de amortecimento dos polos deste controlador.



```

RiseTime: 0.0527
TransientTime: 0.1632
SettlingTime: 0.1632
SettlingMin: 0.9063
SettlingMax: 1.1070
Overshoot: 10.6998
Undershoot: 0
Peak: 1.1070
PeakTime: 0.1167

```

Pole	Damping	Frequency (rad/seconds)	Time Constant (seconds)
$-2.00e+01 + 3.46e+01i$	$5.00e-01$	$4.00e+01$	$5.00e-02$
$-2.00e+01 - 3.46e+01i$	$5.00e-01$	$4.00e+01$	$5.00e-02$
$-5.05e+01$	$1.00e+00$	$5.05e+01$	$1.98e-02$

Analisando as informações apresentadas, vemos que o tempo de assentamento permanece menor que 0,2 segundos, e a taxa de amortecimento dos polos complexos conjugados se situa em 0,5, de acordo com o requisitado.

O projeto por Alocação de Polos requer um trabalho braçal maior do que os demais, porém calcular os polos necessários para se manter a característica necessitada possui uma chance de acerto maior quando se trata de tempo de assentamento e amortecimento, o que pode ser uma vantagem para o projetista.

- 4) Projete um controlador tipo PID que tenha resposta temporal em malha fechada semelhante ao controlador da questão 3. Comente o projeto e justifique os resultados.

O código a seguir foi utilizado para responder à questão 4:

```
clear all
close all
clc

%objetivo:
%atingir as especificações do exercicio anterior
%overshooting = 10.6%
%settling time = 0.161s
%Rise time = 0.052

%declarando o sistema
s=tf('s');
sys = (10*5)/(s*(s+5)*(0.1*s+5));

%Tentativa 1:
%ver a freq de cruzamento
margin(sys)
figure(1)

%frequência de cruzamento de fase obtida observando o gráfico
wcf = 15.8;

%achar o K utilizando G
G = -28.8;
K = 10^(-G/20);
%comparando com o K_routh, foi decidido utilizar
%o próprio K_routh
Kr = 27.49;
Tu = 2*pi/wcf;

%Parâmetros tabelados da tabela de Ziegler-Nichols
Kp = 0.6*Kr;
Ki = 2*Kp/Tu;
Kd = Kp*Tu/8;
C = Kp + Ki/s + Kd*s;

%fechamento da malha
sys_fb_c = feedback(C*sys, 1);
sys_fb = feedback(sys, 1);
sys_crit_fb = feedback(sys*(Kr), 1);

%comparação entre a malha fechada do
%sistema sem o controlador (vermelho)
%e com o controlador

step(sys_fb, "red")
hold on
step(sys_fb_c)

%por este método, podemos ver que o
%overshooting e o tempo de assentamento
%não foram cumpridas.
```

```

%Por isso, decidimos criar um pseudo-PID
%e, das características do pseudo-PID
%achar uma expressão para um PID que
%tenha um comportamento prox do pseudo

%tentativa 2:
%utilizando controlSystemDesigner() e na
%tentativa e erro
%foi possível chegar na seguinte expressão
%para o pseudo-PID com as especificações desejadas:
%Cm = 0.0075205*(0.2*s+1)*(1+2*10^3*s)/(s*(1+0.001*s))
%utilizando Cm como base, foi possível projetar
% o seguinte PID:
a1 = 3.008*1000;
a2 = 15.04*1000;
a3 = 0.007521*1000;
b1 = 1000;

%a1, a2, a3, b1 são coeficientes dos polinomios
%do numerador e do denominador de Cm
%que quando escritos em termos de Kp, Ki, Kd:

Kpm = (-a3+a2*b1)/(b1^2);
Kim = a3/b1;
Kdm = (a3-a2*b1+a1*(b1^2))/(b1^3);

%daqui, equacionamos o PID:
Cm1 = Kpm + Kim/s + Kdm*s;
sys_fb_cm1 = feedback(Cm1*sys, 1);

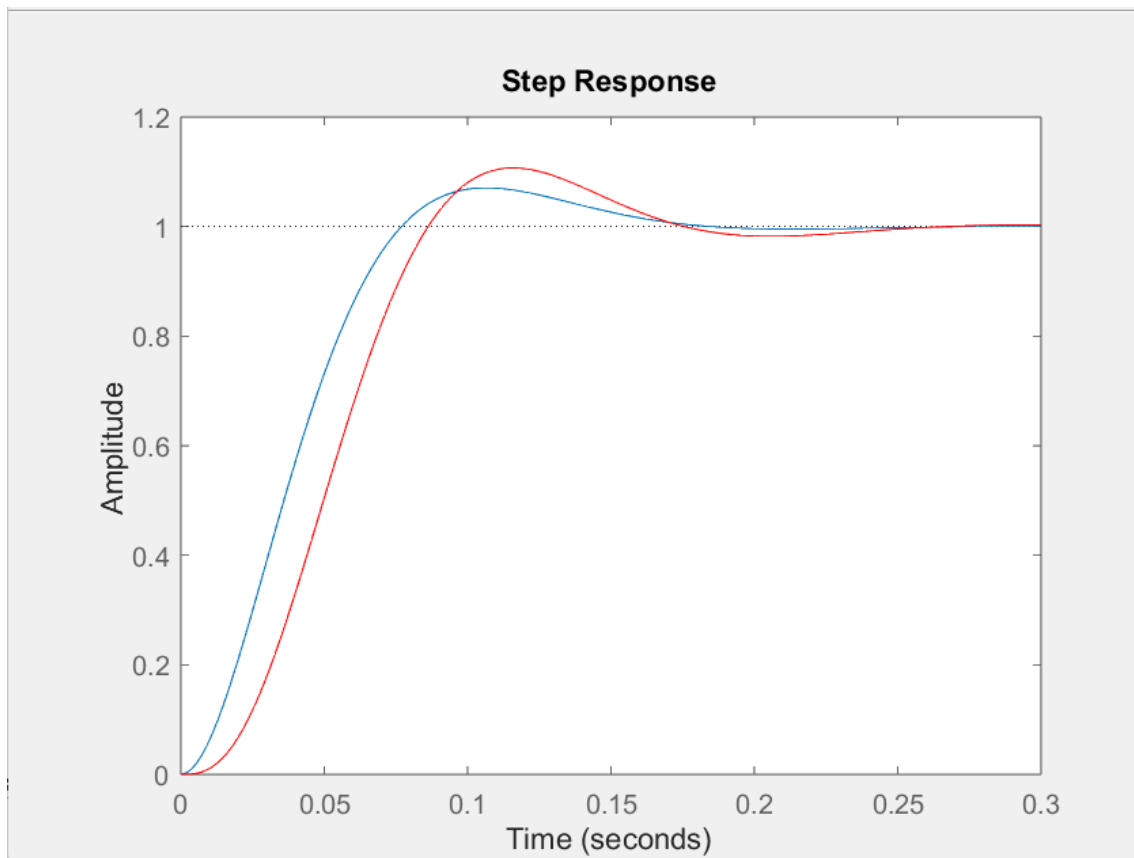
%comparação gráfica das duas malhas fechadas
%PID <- azul
%controlador ex3 <- vermelho
figure()
step(sys_fb_cm1)
hold on

A = [0 1 0 ; 0 0 1 ; 0 -25 -5.5];
B = [0 ; 0 ; 1];
C = [50 0 0];
P = [-20+1i*34.64 -20-1i*34.64 -50.5 ];
K= place(A,B,P);
MJ=C*inv(-A+B*K)*B;
J=1/MJ;
controller=ss(A-B*K, B*J,C,[]);
step(controller, "red")

```



Abaixo, temos a resposta ao degrau do sistema PID comparada com o sistema de alocação de polos, que está em vermelho, e suas informações de degrau:



struct with fields:

```
RiseTime: 0.0509
TransientTime: 0.1558
SettlingTime: 0.1558
SettlingMin: 0.9063
SettlingMax: 1.0706
Overshoot: 7.0640
Undershoot: 0
Peak: 1.0706
PeakTime: 0.1069
```

struct with fields:

```
RiseTime: 0.0527
TransientTime: 0.1632
SettlingTime: 0.1632
SettlingMin: 0.9063
SettlingMax: 1.1070
Overshoot: 10.6998
Undershoot: 0
Peak: 1.1070
PeakTime: 0.1167
```

Baseando-se no gráfico e dados como fontes de análise, percebemos que o controlador PID consegue estabelecer uma performance ainda melhor que o controlador por alocação de polos, visto que ele apresenta tanto overshoot quanto tempo de assentamento menor. O ponto negativo é que como o próprio código mostra, o controlador PID é extremamente sensível a alterações, e a manipulação de cada parte do controlador acaba sendo um desafio de projeto, e sendo um custo alto pela performance apresentada.

**5) Compare o desempenho dos 3 controladores acima e indique as dificuldades/facilidades dos projetos.**

Analisando os 3 controladores, podemos tomar algumas observações:

- O controlador por compensação de fase, possui um overshoot relativamente baixo, mesmo que não o mais baixo dos 3, e um longo tempo de assentamento. Quando o objetivo é manter a margem de fase, é uma boa alternativa pois o controlador consegue compensar a fase do sistema sem causar alterações drásticas no sobrepasso da resposta do sistema, sendo as únicas desvantagens o tempo de assentamento e a sensibilidade que o sistema tem ao ganho, que pode desregular a margem de fase desejada mesmo com alterações pequenas.
- O controlador por PID é o controlador mais flexível de todos os controladores, visto que cada “parte” do controlador pode ser ajustada para as devidas características de projeto necessárias. A desvantagem deste controlador se apresenta na sua estabilidade, ou ao menos a sensibilidade de sua estabilidade. Modificar as partes do controlador para atingir certos patamares torna-se um desafio para o projetista. Mesmo atingindo uma precisão maior que o controlador por alocação de polos, o PID possui um grau de liberdade tão alto, com tantas variáveis, que sua projeção se torna em certo ponto instável, viabilizando o uso de algo como um controlador de alocação de polos.
- O controlador por alocação de polos possui a vantagem de ter um overshoot e tempo de assentamento baixos em uma boa estabilidade, mesmo que não tão baixos quanto os parâmetros obtidos pelo PID. De todos, é o controlador mais preciso nestes dois quesitos, o que o torna eficiente para aplicações precisas por alcançar patamares de precisão altos, e ser estável. O custo desta eficiência é que o projeto do controlador exige uma quantidade considerável de trabalho técnico, em comparação com os outros.

Em conclusão, a aplicação do controlador evidentemente impacta no nível de precisão e construção que deve ser investido.

**OBS:** Os PDFs de apoio mencionados no relatório acima serão enviados em anexo com os códigos do Matlab, e o Relatório em si.