

Trabalho 1:

Análise de Sistemas no Espaço de Estados

Professor responsável: Paulo Gil

João Carvalho nº 49341

Nikita Dyskin nº 49541

**Índice:**

Introdução 3

Modelação analítica 4

Obtenção do modelo em espaço de estados 4

Análise da estabilidade do sistema 6

Forma canónica diagonal 8

Análise através da resposta temporal do sistema 10

Matriz de transição de estados 10

Solução da equação de estado em regime não forçado 12

Simulação da resposta temporal do sistema em anel aberto 13

Projecto de controlador por retroacção de estado 16

Especificações do controlador 16

Controlabilidade do sistema 17

Projecto do controlador com efeito integral 19

Simulação do sistema 21

Conclusão 26

Anexo 27

# **Introdução**

Este trabalho tem como objectivos modelar o sistema da figura 1 em espaço de estados, analisar o seu comportamento dinâmico e fazer controlo por retroacção das variáveis de estado.

O sistema em estudo é constituído por três reservatórios, tendo os tanques 1 e 2 um diâmetro interno de 40 cm e o tanque 3 de 50 cm, duas eletrobombas no tanque 1 e no tanque 2, válvulas para regular o caudal e sensores de nível em todos os tanques.

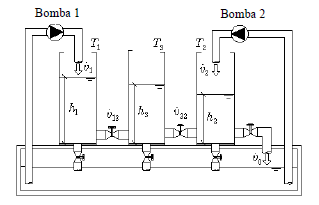


Figura 1-Sistema de vasos comunicantes

As electrobombas em funcionamento nominal, desprezando o efeito transitório, apresentam uma relação entre os caudais volúmicos e e as tensões aplicadas aos terminais das respectivas electrobombas, e , descrita por:

Os caudais volúmicos entre tanques adjacentes e os respectivos caudais volúmicos de descarga são genericamente descritos por

(1)

onde,

# **Modelação analítica**

## 2.1. Obtenção do modelo em espaço de estados

O modelo em espaços de estados foi obtido através da aplicação da equação do princípio da conservação da massa, dado por:

(2)

Sendo:

(3)

Considerando como variáveis de estado os níveis de líquido nos 3 tanques *(h1,h2,h3)***,** como entradas *(u1 e u2)* a tensão eléctrica aplicada, em Volt, aplicada a cada electrobomba e como saídas os níveis nos tanques *1* e *2* e o caudal volúmico de saída , determinou-se então a dinâmica dos tanques na forma de equações diferenciais e posteriormente a obtenção dos respectivos modelos descritos em espaço de estados.

Recorrendo às equações 1,2 e 3, temos:

Dinâmica do tanque 1:

Substituindo:

Vem que:

⇔ -

Dinâmica do tanque 2:

Seguindo o mesmo método de substituições para o tanque 2, obtém-se:

Dinâmica do tanque 3:

Repetindo o método, vem por fim:

Sendo as saídas do sistema, os níveis de líquido do tanque 1, do tanque 2 e o caudal volúmico de saída, temos:

𝑦1(𝑡) = ℎ1(𝑡)

𝑦2(𝑡) = ℎ2(𝑡)

𝑦3(𝑡) = (𝑡) = ℎ3(𝑡)

A dinâmica do sistema dada pelo conjunto de equações diferenciais pode ser escrita sob a forma de vectores e matrizes, resultando no modelo de estado que se segue. Que pode ser descrito de forma abreviada:

(4)

As equações anteriores, determinam o modelo de estado do sistema. Onde o vector x e as matrizes A,B e C são dadas por:

Sendo correspondentemente:

2.2. Análise de estabilidade do sistema

Os valores próprios do sistema correspondem aos valores próprios da matriz A, obtidos através de:

|  |  |
| --- | --- |
|  |  |

Desenvolvendo a expressão obtém-se a equação característica do modelo de estado.

0

Cuja solução é:

Como os valores próprios apresentam parte real <0, conclui-se que o sistema é estável. Utilizando a função *eig()* do *Matlab* confirma-se a correcta obtenção dos valores próprios por métodos analíticos.

Podemos também confirmar no Matlab que o sistema é estável através da resposta impulsional, como podemos observar na figura 2.

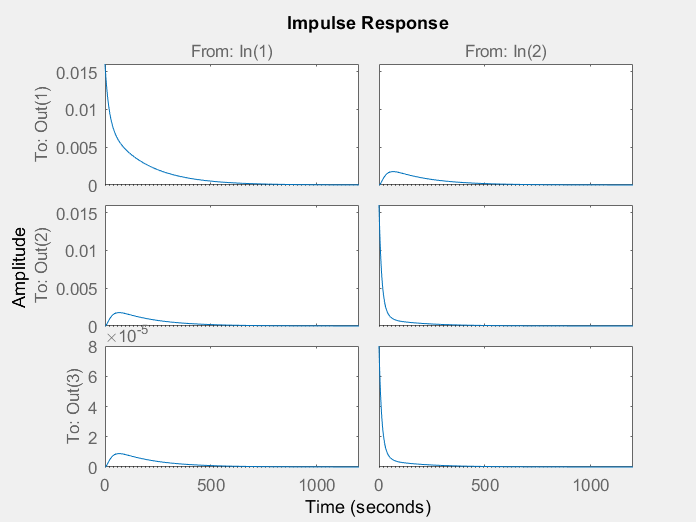


Figura 2-resposta impulsional do sistema

Os vectores próprios da matriz A são os vetores não nulos que são a solução da equação:

(5)

Ao resolvermo-la obtemos:

Recorrendo novamente à função *eig()* do *Matlab*, confirma-se que :

2.3. Forma canónica diagonal

Procuramos agora encontrar uma matriz de transformação de semelhança que converta o sistema obtido anteriormente na forma canónica diagonal. Como os valores próprios da matriz A são valores não inteiros e distintos, consideramos que a matriz de transformação de semelhança é dada pela matriz constituída pelos vectores próprios.

(6)

donde, det(T) = 0.97.

Como o determinante da matriz transformação é diferente de zero, conclui-se que é invertível, e, por isso podemos obter a forma canónica do sistema da seguinte forma:

(7)

Invertendo a matriz de transformação, obtém-se:

Assumindo a transformação de base,

(8)

E tendo em conta a descrição do sistema em espaços de estados, verifica-se

Fazendo os cálculos de transformação de base descritos em (6) para o sistema A, obtém-se:

1. **Análise através da resposta temporal do sistema**

3.1. Matriz de transição de estados

A equação de estado homogénea corresponde à parte do sistema cuja evolução não depende duma excitação externa, dependendo assim, único e exclusivamente das condições iniciais . Assim sendo, a solução do sistema homogéneo pode ser escrita da seguinte forma:

(9)

Sendo:

(10)

A matriz pode ser calculada a partir de duas formas:

Método 1: Forma canónica diagonal

A matriz de transição de estados associada à forma canónica diagonal é obtida da seguinte forma:

Agora, é possível obter a matriz de transição de estados através duma transformação de base, utilizando como matriz de transformação a matriz calculada em (6):

(11)

Chegamos então ao resultado:

Com os respectivos membros:

Método 2 (confirmação): Transformada de Laplace

Considerando condições iniciais não nulas, pode-se escrever:

(11)

Fazendo os cálculos, obtém-se:

=

Aplicando o operador Transformada de Laplace inverso, vem que:

Com respectivos membros:

Obtendo-se então a validade do primeiro método.

3.2. Solução da equação de estado em regime não forçado

A equação de estado em regime não forçado partindo da representação do sistema na forma canónica diagonal tem a seguinte forma:

(12)

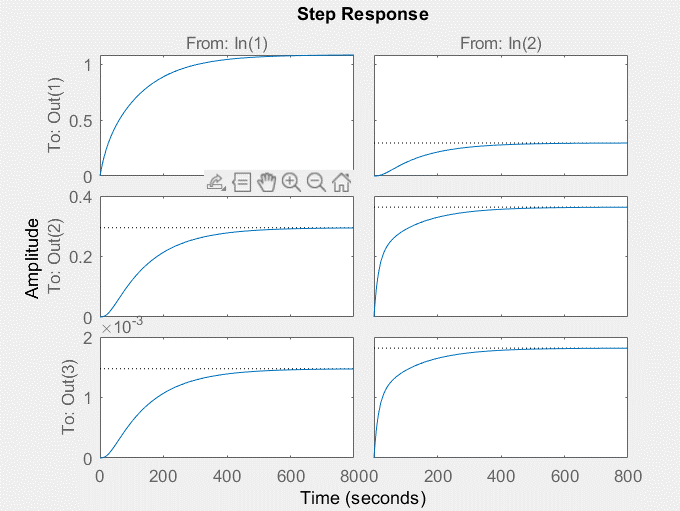
Considerando:

3.3. Simulação da resposta temporal do sistema em anel aberto

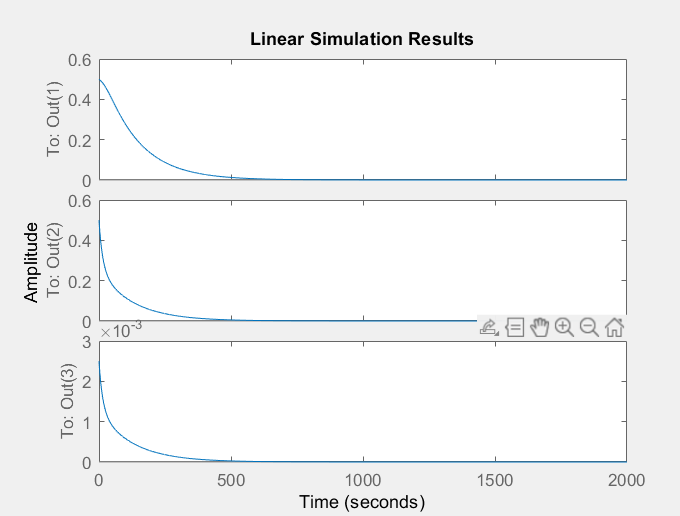
Considerando condições iniciais nulas e recorrendo às funções do *Matlab step()*, calculou-se a resposta ao degrau unitário do sistema em anel aberto.

Resposta ao degrau unitário:

*Figura 3 – Resposta ao degrau unitário do sistema*

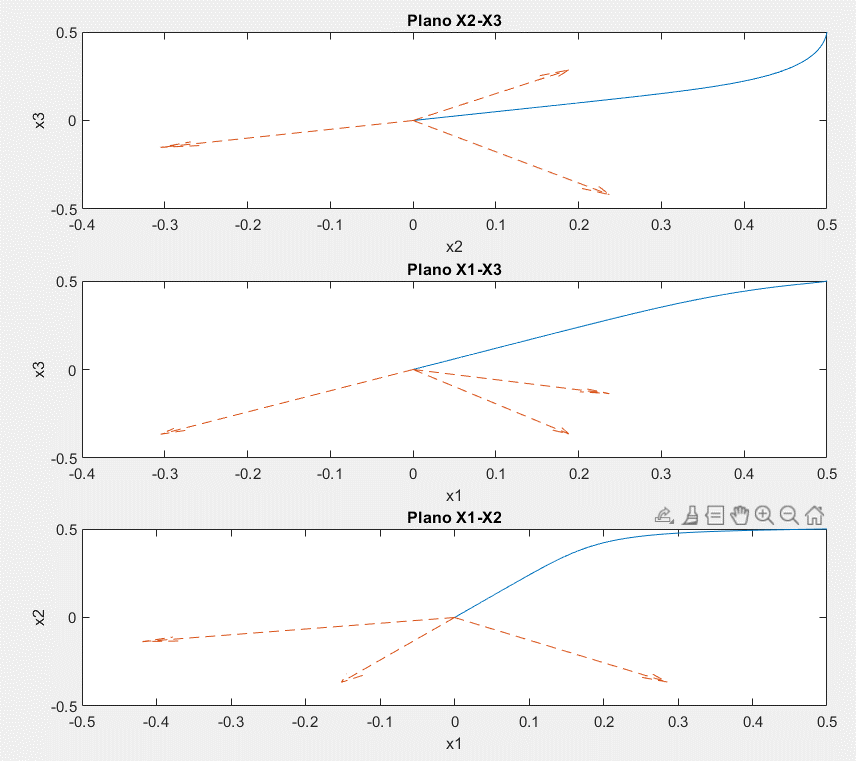


Agora, utilizando a função do *Matlab lsim(),* assumindo condições iniciais m, com as electrobombas em repouso (regime não forçado), observou-se o sistema a convergir para a origem. Para a simulação, utilizou-se um horizonte temporal de 2000 segundos, obteve-se a seguinte resposta:



*Figura 3 – Simulação do sistema em anel aberto, em regime não forçado com condições iniciais não nulas*

De seguida calculou-se a trajectória de estado nos 3 planos, descrita na base dos vectores de estado. O resultado foi:



*Figura 3 – Trajectória de estado no 3 planos*

1. **Projecto de controlador por retroacção de estado**

4.1 Especificações do controlador

Finalmente, procede-se ao projecto de um controlador por retroacção de estado com e sem efeito integral.

Para ambos os controladores, pretende-se que forcem o sistema a que tenha erro estático nulo na resposta ao degrau com 0.8 de amplitude, tempo de estabelecimento a 5% inferior a 60% (escolheu-se 50%) do tempo dominante de estabelecimento do sistema em anel aberto e que tenha sobreelevação nula.

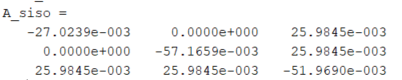
Para que o tempo de estabelecimento seja cumprido, começa-se por calcular o tempo de estabelecimento dominante do sistema em anel aberto, com o auxilio da função *Matlab stepinfo()*, cujo valor é , que multiplicado por 50%, é 199.5. O polo dominante, para que o sistema tenha tal dinâmica, é calculado através da expressão:

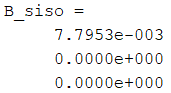
(13)

Como se pretende que a resposta do sistema tenha sobreelevação nula, a parte imaginária dos polos é nula.

4.2. Controlabilidade do sistema

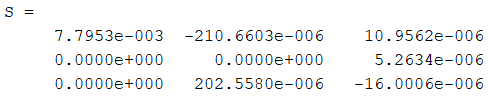
Para os projectos dos controladores, considerou-se o sistema SISO, apenas com a electrobomba 1 a funcionar e com a observação do nível de :



Cujo polinómio característico é dado por

Construindo a matriz de controlabilidade :



Verifica-se que a sua característica é igual a 3, pelo que o par (A,B) é completamente controlável.

4.3. Projecto do controlador por retroacção de estado com ganho adicional

Pela expressão (13), consideram-se os valores próprios desejados do sistema que respeitam as condições pretendidas para a dinâmica, como sendo:

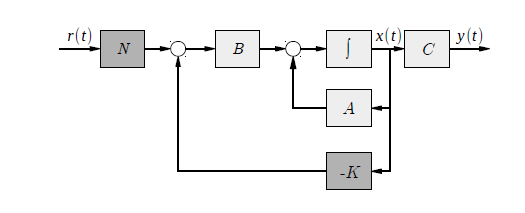
Cujo polinómio característico é dado por:

Pretende-se agora dimensionar o vector de ganhos K, calculado a partir de , com e toma a seguinte forma:

Pela dimensão do sistema, torna-se um esforço hercúleo calcular analiticamente a solução, pelo que se utiliza a função de *Matlab acker()* que calcula o vector de ganhos K de retroação que colocam o sistema com os valores próprios pretendidos, sendo ele:

O cálculo do ganho adicional tem como objectivo compensar o ganho estático do sistema em anel fechado. O gajo adicional é calculado a partir do seguinte sistema:

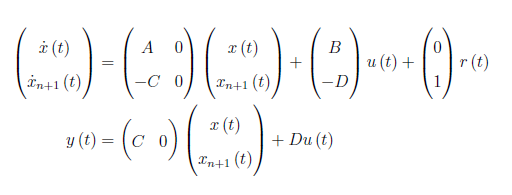
Pelo que,



*Figura 4 – Arquitectura do controlo de seguimento com ganho adicional*

4.4 Projecto do controlador com efeito integral

A inclusão do efeito integral no controlador por retroacção de variáveis de estado, torna-se desta forma bastante simples, se pensarmos que basta adicionar uma variável extra ao sistema cuja derivada é dada pelo erro de controlo, que depois é integrada e multiplicada por um ganho e de seguida somada com o estado de retroacção multiplicado por um ganho. Ficando então o sistema estendido, escrito da seguinte forma:



Como a matriz D=0 obtemos o sistema estendido:

Sendo o sistema original controlável, o sistema estendido também o é. Deste modo, podemos calcular a matriz de ganhos K.

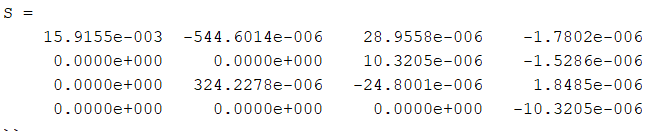
Escolhe-se como valores próprios com base na expressão (13), mas desta vez, como o sistema estendido é de quarta ordem, adicionamos um polo extra 1 década depois do polo dominante, para que este não condicione a dinâmica pretendida do sistema. Ficamos então com o conjunto de valores próprios:

Calculando o polinómio característico do sistema estendido, obtemos:

E como polinómio característico do sistema com a dinâmica pretendida, temos que:

Podemos finalmente calcular o vector , como sendo:

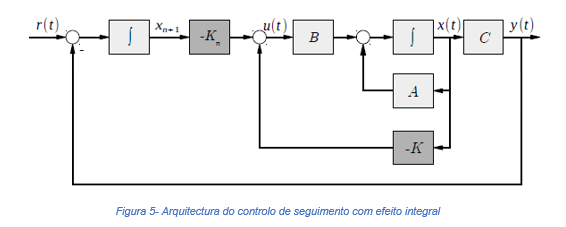
Com recurso à função *Matlab ctrb(),* calcula-se a matriz de controlabilidade S.



A partir do sistema estendido em anel aberto, obtemos a matriz modal.

Vem finalmente que:

Vem então que:



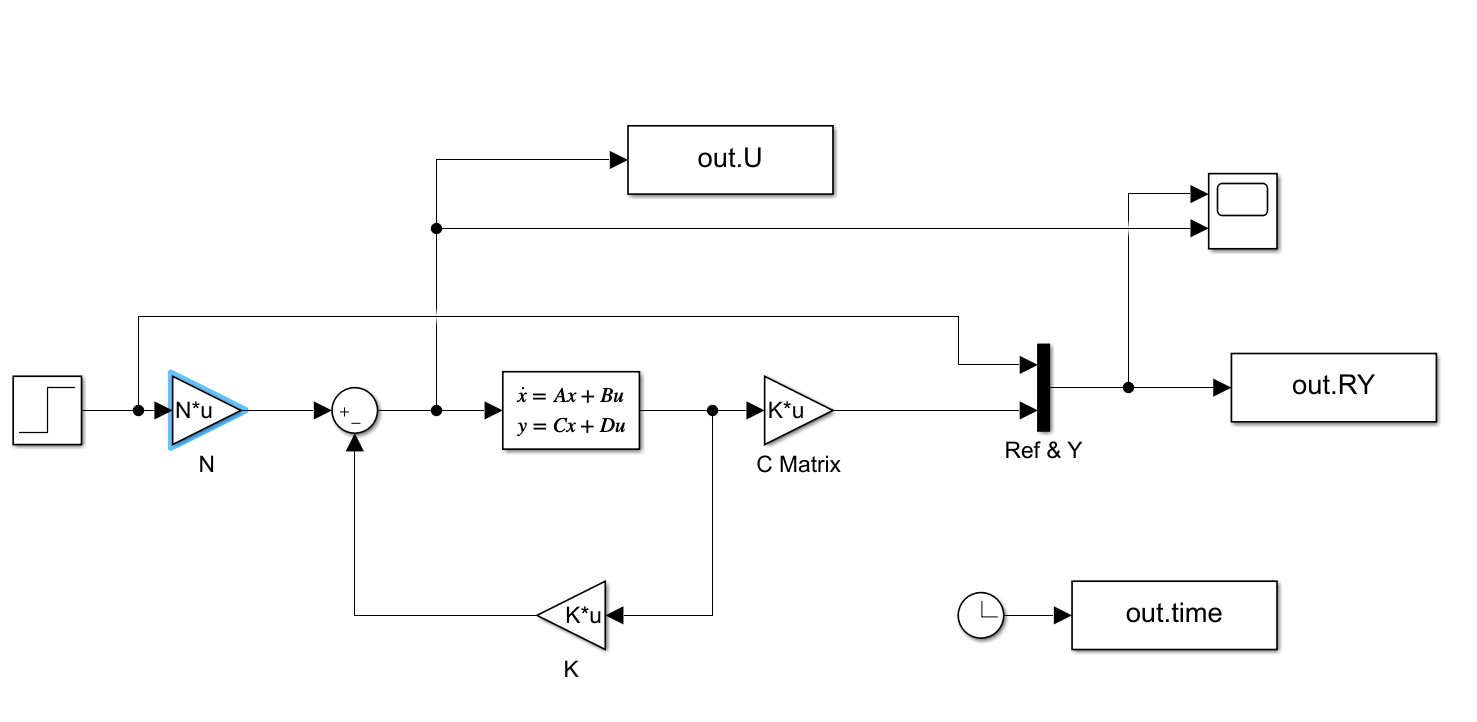
Em que a matriz de ganhos e a matriz de ganhos

4.5. Simulação do sistema

Para a simulação do sistema utilizou-se a *Toolbox Simulink* do *Matlab*. Simulou-se o sistema com o controlador de retroacção de estado com ganho de compensação e o controlador com retroacção de estado e efeito integral. Para ambas as arquitecturas, simulou-se primeiro sem perturbações e de seguida com uma perturbação de uma constante multiplicativa 0.8 na matriz de entrada.

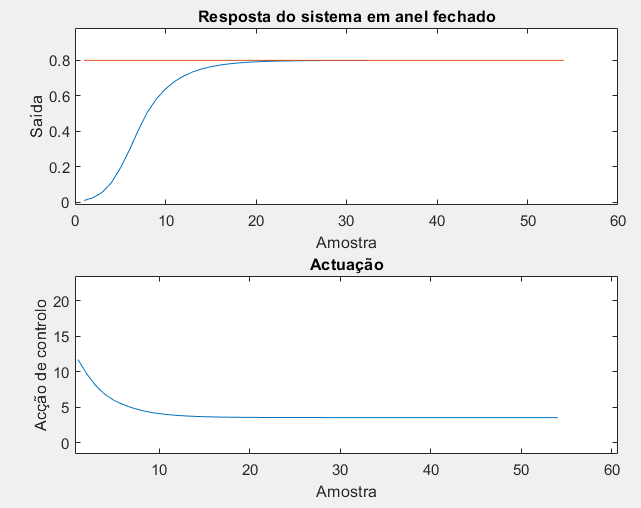
Controlador com retroacção de variáveis de estado com ganho adicional:

No *Simulink* contruiu-se o seguinte diagrama de blocos:

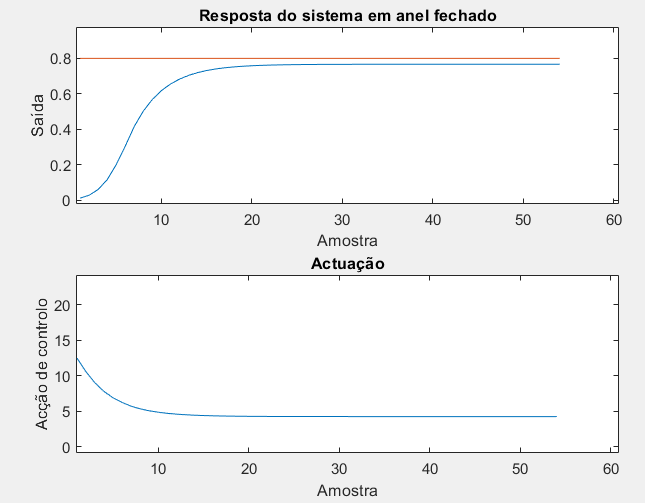


*Figura 6- Diagrama de blocos do sistema de controlo com retroacção de variáveis de estado e ganho adicional*

Cujo resultado da simulação é:



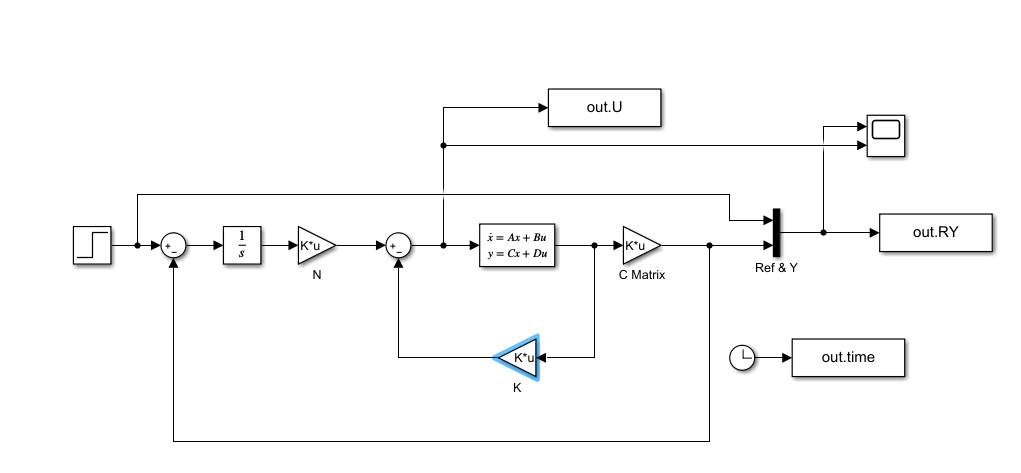
*Figura 7-* *Resultado do sistema* *de controlo com retroacção de variáveis de estado e ganho adicional sem perturbações*



*Figura 8-* *Resultado do sistema* *de controlo com retroacção de variáveis de estado e ganho adicional com perturbações*

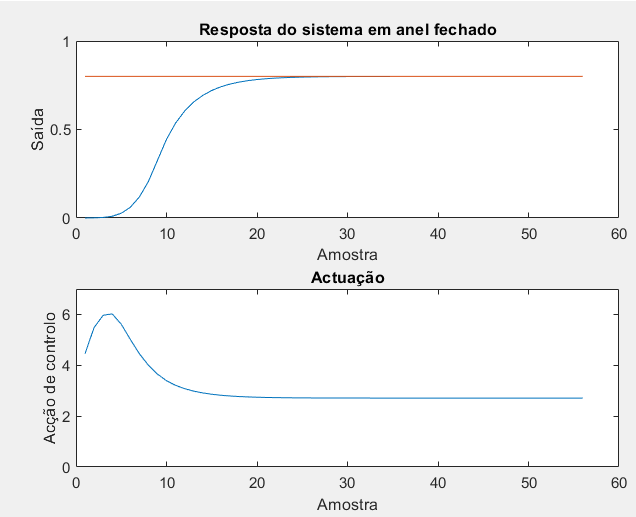
Controlador com retroacção de variáveis de estado com efeito integral:

No *Simulink* contruiu-se o seguinte diagrama de blocos:

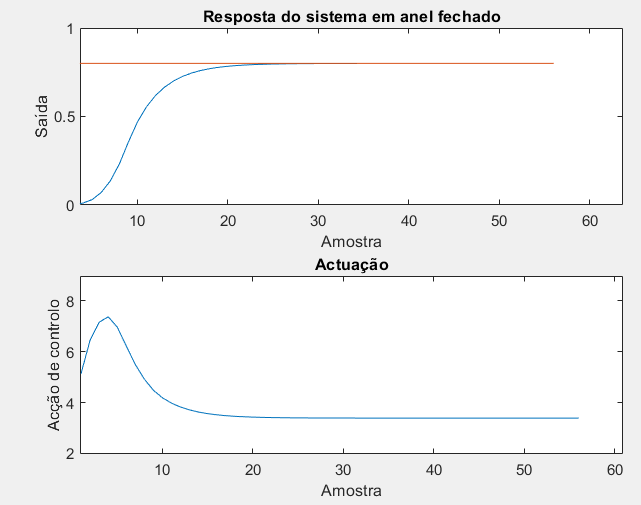


*Figura 9- Diagrama de blocos do sistema de controlo com retroacção de variáveis de estado com efeito integral*

Cujo resultado da simulação é:

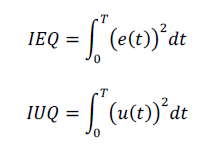


*Figura 10-* *Resultado do sistema* *de controlo com retroacção de variáveis de estado e efeito integral sem perturbações*

**

*Figura 11-* *Resultado do sistema* *de controlo com retroacção de variáveis de estado e efeito integral com perturbações*

Usando como métricas de comparação:



Obtemos:

Controlador com ganho adicional:

Sem perturbação: IEQ = 8.9916

IUQ = 44395

Error estático= 3.37e-7

Com perturbação: IEQ = 9.026

IUQ = 4831.1

Erro estático = 0.033

Controlador com efeito integral:

Sem perturbação: IEQ = 5.0005

IUQ = 8967.8

Erro estático = 3.47e-7

1. **Conclusão**

Com este trabalho, o grupo conseguiu com sucesso modelar um sistema físico em espaço de estados e controlá-lo.

Concluímos que na presença de uma perturbação que cause erro estático, o controlador de retroacção de estado com ganho adicional não é muito robusto, o que leva à necessidade de inclusão de efeito integral, visto que este integra o erro acumulado e isso faz com que desapareça o erro estático, já que este é constante ao longo do tempo e o efeito integral “tem em conta” todo o erro passado.

1. **Anexo**

Modelação analítica:

close all, clear all, clc;

format shortEng

format compact

%%

%Variables

syms s t;

A1 = 0.125; A2 = A1;

A3 = 0.196;

n13 = 4e-3;

n32 = 4e-3;

n0 = 5e-3;

n30 = 3e-4;

n10 = 3e-4;

v1 = 20e-4;

v2 = 20e-4;

%%

%State space representation System

A = [-(1/A1)\*(n13+n10) 0 n13/A1; 0 (-1/A2)\*(n32+n0) (1/A2)\*n32; (1/A3)\*n13 (1/A3)\*n32 -(1/A3)\*(n13+n30+n32)];

B = [(1/A1)\*v1 0; 0 (1/A2)\*v2; 0 0];

C = [1 0 0; 0 1 0; 0 n0 0];

D = zeros(3,2);

%%

%forma canonica, matrizes de transformacao e de transicao

[V,D2] = eig(A, 'nobalance', 'vector');

val\_prop = diag(D2);

T = V;

Ad = (T^-1) \* A \* T;

Bd = T^-1 \* B;

Cd = C \* T;

phid = diag(exp(D2\*t));

phiT = T \* phid \* T^-1; %transicao de estados atraves da matriz de transicao

phiT = vpa(phiT);

%transicao de estados com Transformada de laplace

sia = s \* eye(size(A,1),size(A,2)) - A;

phiS = adjoint(sia) / det(sia);

phi = vpa(ilaplace(phiS));

phi = vpa(phi);

%%

%Time response

sys = ss(A,B,C,D);

G = tf(sys);

figure();

step(G);

figure();

impulse(sys);

%%

%Simulação

x0 = [.5 .5 .5];

t = 0:0.05:2000;

u = zeros(length(t),2);

figure();

lsim(sys,u,t,x0)

[y,t,x] = lsim(sys,u,t,x0);

%%

%evolução do vector de estado

V32 = V(3:-1:2,1:3);

V31 = V(3:-1:1, 1:3), [V31(2,:)] = [];

V21 = V(2:-1:1, 1:3);

figure();

subplot(3,1,1), plot(x(:,3), x(:,2)), hold on, qv1 = quiver([0 0 0], [0 0 0], V32(1,:), V32(2,:), 0.5); hold off;

qv1.MaxHeadSize = .2; qv1.LineStyle='--';

title('Plano X2-X3'), ylabel('x3'), xlabel('x2')

subplot(3,1,2), plot(x(:,3), x(:,1)), hold on, qv1 = quiver([0 0 0], [0 0 0], V31(1,:), V31(2,:), 0.5); hold off;

qv1.MaxHeadSize = .2; qv1.LineStyle='--';

title('Plano X1-X3'), ylabel('x3'), xlabel('x1')

subplot(3,1,3), plot(x(:,2), x(:,1)), hold on, qv1 = quiver([0 0 0], [0 0 0], V21(1,:), V21(2,:), 0.5); hold off;

qv1.MaxHeadSize = .2; qv1.LineStyle='--';

title('Plano X1-X2'), ylabel('x2'), xlabel('x1')

Projecto controlador:

clear all

close all

clc

n13 = 2.5e-3;

n10 = 1.0e-4;

n32 = 2.5e-3;

n0 = 3e-3;

di = 0.35;

Atk = pi/4\*di^2;

% I. Model :: SISO

% -------------------------------

A\_siso = [-(n13+n10) 0 n13;0 -(n32+n0) n32;n13 n32 -(n13+n32)]/Atk;

B\_siso = [7.5e-4 0 0]'/Atk;

C\_siso = [0 1 0];

D\_siso = [0];

% II. State Feedback Control :: N

% -------------------------------

Ts\_design\_set = 0.5\*399; % << Ts\_design\_max

Dwn = 3/Ts\_design\_set;

lambda\_1 = -Dwn;

lambda\_2 = 5\*lambda\_1;

lambda\_3 = 8\*lambda\_1;

lambda\_spec = [lambda\_1 lambda\_2 lambda\_3]

K = acker(A\_siso,B\_siso,lambda\_spec);

Nxu = [A\_siso B\_siso;C\_siso D\_siso]^-1\*[zeros(3,1); 1];

N = Nxu(end)+ K\*Nxu(1:end-1);

% EFEITO INTEGRAL

clear all

close all

clc

n13 = 4.0e-3;

n10 = 3.0e-4;

n32 = 4.0e-3;

n30 = 3.0e-4;

n0 = 5.0e-3;

di12 = 0.4;

di3 = 0.5;

Atk12 = pi/4\*di12^2;

Atk3 = pi/4\*di3^2;

% I. Model :: SISO

% -------------------------------

A\_siso = [-((n13+n10)/Atk12) 0 (n13/Atk12);0 -((n32+n0)/Atk12) (n32/Atk12);(n13/Atk3) (n32/Atk3) -((n13+n32+n30)/Atk3)];

B\_siso = ([(20e-4) 0 0]')/Atk12;

C\_siso = [0 1 0];

D\_siso = [0];

sys = ss(A\_siso,B\_siso,C\_siso,0);

% II. State Feedback Control :: N

% -------------------------------

A\_=[A\_siso zeros(3,1); -C\_siso 0];

B\_ = [B\_siso; -D\_siso];

C\_ = [0 1 0];

Ts\_design\_set = 399\*0.5; % << Ts\_design\_max

Dwn = 3/Ts\_design\_set;

lambda\_1 = -Dwn;

lambda\_2 = 5\*lambda\_1;

lambda\_3 = 8\*lambda\_1;

lambda\_4 = 10\*lambda\_1;

lambda\_spec = [lambda\_1 lambda\_2 lambda\_3 lambda\_4]

K\_ = acker(A\_,B\_,lambda\_spec);

K=K\_(1:3);

N=K\_(4);