

Projeto CDIG 2020: O vírus CODIG – Grupo 23

Tomás Martins Araújo
CDIG, 3MIEEC_A2
FEUP – Faculdade de Engenharia da
Universidade do Porto
Porto, Portugal
up201704738@fe.up.pt

Bernardo Campos Leichsenring Franco
CDIG, 3MIEEC_A2
FEUP – Faculdade de Engenharia da
Universidade do Porto
Porto, Portugal
up201707073@fe.up.pt

Guilherme Pereira Pinheiro
CDIG, 3MIEEC_A2
FEUP – Faculdade de Engenharia da
Universidade do Porto
Porto, Portugal
up201703867@fe.up.pt

Francisco Tiago de Espírito Santo e Caetano
CDIG, 3MIEEC_A2
FEUP – Faculdade de Engenharia da
Universidade do Porto
Porto, Portugal
up201705031@fe.up.pt

Resumo—Este relatório incide sobre a segunda parte do projeto da unidade curricular CDIG relativa a Espaço de Estados (*State Space Representation, Analysis and Synthesis*). É analisado um modelo compartimental, neste caso uma variação do modelo SIR, agora efetuando uma distinção entre os infetados mais graves e os menos graves.

Palavras-chave — População, Infetados, Suscetíveis, Recuperados, Controlador, Estimador, Espaço de Estados.

I. INTRODUÇÃO

Tendo sido estudadas previamente o método mais banal para o estudo da evolução de uma pandemia, mas também para a análise dos efeitos que as medidas de prevenção provocam, torna-se importante estudar modelos mais complexos e realistas. Partindo, mais uma vez, do modelo SIR, proposto por Kermack e McKendrick em 1927, propõe-se o modelo SI_dI_hR .

Este modelo difere do estudado anteriormente por realizar uma subdivisão do comportamento de infetados em dois compartimentos distintos. Para além dos compartimentos **S** (Suscetíveis) e **R** (Recuperados), surgem os compartimentos **I_d** (Infetados menos graves, que se encontram a recuperar no domicílio) e **I_h** (Infetados mais graves, que se encontram a recuperar no hospital). A doença infeta os suscetíveis. Os infetados eventualmente irão recuperar, sendo considerados temporariamente imunes, podendo regressar ao estado de suscetíveis. Esta é uma diferença considerável relativamente ao cenário da Parte I, em que se assumia que, uma vez recuperado, a imunidade estava assegurada. No entanto, se considerarmos $I = I_h + I_d$, mantém-se a relação $N = S + I + R$, com N igual à dimensão da população.

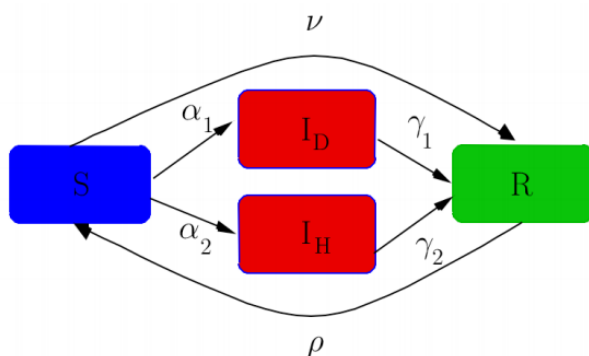


Figura 1 - Modelo SI_dI_hR

A dinâmica de propagação é expressa por um sistema de 4 equações diferenciais ordinárias:

$$\dot{S}(t) = -(\alpha_1 + \alpha_2)S(t) - \nu S(t) + \rho R(t)$$

$$\dot{I}_d(t) = \alpha_1 S(t) - \gamma_1 I_d(t)$$

$$\dot{I}_h(t) = \alpha_2 S(t) - \gamma_2 I_h(t)$$

$$\dot{R}(t) = \nu S(t) + \gamma_1 I_d(t) + \gamma_2 I_h(t) - \rho R(t)$$

Podemos considerar α_1 e α_2 as taxas de transição dos Suscetíveis para os compartimentos I_d e I_h , respetivamente. Cada um destes comportamentos apresenta as respetivas taxas de transição para o compartimento dos Recuperados, representadas por γ_1 e γ_2 . Considera-se ainda que cada indivíduo, depois de recuperado, tem imunidade temporária, passando depois de algum tempo à condição de Suscetível novamente, com taxa de transição ρ . Podemos representar estas variáveis por:

$$\alpha = \alpha_1 + \alpha_2 = R_t \cdot \frac{I_d(t)}{N}; R_t = R_0 - u$$

$$\alpha_1 = 0,85\alpha$$

$$\alpha_2 = 0,15\alpha$$

$$\gamma = \gamma_1 + \gamma_2 = \frac{1}{T_{inf}}$$

O parâmetro R_t corresponde ao número de reprodução, ou seja, o número de pessoas que uma pessoa infetada irá contagiar, assumindo que as outras pessoas não estão infetadas e não foram vacinadas. Pode ser controlado por u , que corresponde às medidas de contenção. R_0 toma o valor de 2,4 nesta simulação. T_{inf} , por seu turno, é a duração média da infeção em dias, assumindo inicialmente o valor 6.

Tendo estes fatores em conta, uma melhor representação do sistema de equações diferenciais seria:

$$\dot{S}(t) = -\alpha S(t) - \nu S(t) + \rho R(t)$$

$$\dot{I}_d(t) = 0,85\alpha S(t) - \gamma I_d(t)$$

$$\dot{I}_h(t) = 0,15\alpha S(t) - \gamma I_h(t)$$

$$\dot{R}(t) = \nu S(t) + \gamma(I_d(t) + I_h(t)) - \rho R(t)$$

II. ANÁLISE DO SISTEMA POR ESPAÇO DE ESTADOS EM TEMPO CONTÍNUO

Este capítulo refere-se ao primeiro ponto do enunciado fornecido. Serão analisadas várias questões, nomeadamente aquelas relacionadas com a linearização do sistema em torno

de um ponto de equilíbrio, bem como da obtenção do espaço de estados que o representa.

Serão avaliadas várias propriedades do espaço de estados obtidos, nomeadamente a controlabilidade, a observabilidade e a sua estabilidade. Para além disso, serão apresentadas várias representações possíveis para o espaço de estados.

A. Ponto de Funcionamento

Assuma-se um ponto de funcionamento para o sistema caracterizado por $(S, I_d, I_h, R) = (S^*, I_d^*, I_h^*, R^*)$. Vamos assumir que o ponto de funcionamento é um ponto de equilíbrio. Os pontos de equilíbrio são caracterizados por corresponderem a pontos em que se verifica que a variação das variáveis se anula, ou seja:

$$\dot{S}(t) = 0$$

$$\dot{I}_d(t) = 0$$

$$\dot{I}_h(t) = 0$$

$$\dot{R}(t) = 0$$

Tendo em conta estas condições podemos trabalhar o sistema de equações:

$$\begin{aligned} & \begin{cases} \alpha^* S^* + v^* S^* = \rho R^* \\ \gamma I_d^* = 0,85 \alpha^* S^* \\ \gamma I_h^* = 0,15 \alpha^* S^* \\ \gamma(I_h^* + I_d^*) + v^* S^* = \rho R^* \end{cases} \Leftrightarrow \\ & \Leftrightarrow \begin{cases} R^* = \frac{\alpha^* + v^*}{\rho} S^* \\ \gamma I_d^* = 0,85 \frac{(R_0 - U^*)}{N} I_d^* \gamma S^* \\ \frac{\gamma I_h^*}{\gamma I_d^*} = \frac{0,15 \alpha^* S^*}{0,85 \alpha^* S^*} \\ \gamma(I_h^* + I_d^*) + v^* S^* = \rho S^* \frac{\alpha^* + v^*}{\rho} \end{cases} \Leftrightarrow \\ & \Leftrightarrow \begin{cases} R^* = \frac{\alpha^* + v^*}{\rho} S^* \\ 0,85 U^* S^* = 0,85 R_0 S^* - N \\ \frac{I_h^*}{I_d^*} = \frac{0,15}{0,85} \\ \gamma(I_h^* + I_d^*) = \alpha^* S^* \end{cases} \Leftrightarrow \\ & \Leftrightarrow \begin{cases} R^* = \frac{\alpha^* + v^*}{\rho} S^* \\ U^* = R_0 - \frac{N}{0,85} \\ \frac{I_h^*}{I_d^*} = \frac{0,15}{0,85} \\ S^* = \frac{\gamma}{\alpha^*} (I_h^* + I_d^*) \end{cases} \end{aligned}$$

B. Ponto de Equilíbrio

Considere-se o possível ponto de equilíbrio definido por $N = 10^7$, $I^* = 2 \times 10^5$, $R^* = 2 \times 10^6$ e $S^* = 7,8 \times 10^6$. Considere-se também que $\rho = \frac{1}{60}$, $\gamma = \frac{1}{6}$ e $v^* = 0$. Para este ponto deve cumprir um requisito: anular a variação dos parâmetros. Isto pode ser verificado através de dois métodos, utilizando as equações diferenciais do sistema ou recorrendo às expressões obtidas no ponto anterior.

Em primeiro lugar, é necessário determinar os valores de I_d^* e I_h^* :

$$I^* = I_h^* + I_d^*; \frac{I_h^*}{I_d^*} = \frac{0,15}{0,85}$$

$$I^* = \frac{15}{85} I_d^* + I_d^* \Leftrightarrow I^* = \frac{100}{85} I_d^* \Leftrightarrow I_d^* = 1,7 \times 10^5$$

$$\therefore I_h^* = 3 \times 10^4$$

Recorrendo ao MATLAB, optou-se por se utilizar as expressões calculadas no ponto anterior para verificar que este se tratava efetivamente de um ponto de equilíbrio. Foram declaradas as variáveis com os valores assumidos, realizando-se as seguintes operações:

```
U=2.4-(N/(0.85*S));
disp(U);

Id=(85/100)*I;
disp(Id);
alfa=(Ro-U)*gama*Id/N;
disp(alfa);

%%Comprovar resultados
R_teste=((alfa+v)/p)*S;
disp(R_teste);

S_teste=(gama/alfa)*(I);
disp(S_teste);

Id_teste=(0.85*alfa*S)/gama;
disp(Id_teste);

Ih_teste=(0.15*alfa*S)/gama;
disp(Ih_teste);
```

Figura 2 - Código utilizado para comprovar o ponto de equilíbrio

Se as igualdades são válidas para um ponto de funcionamento definido como um ponto de equilíbrio, então se obtivermos igualdades verdadeiras, teremos uma prova concreta da sua autenticidade.

Os resultados obtidos foram os seguintes:

$$\Leftrightarrow \begin{cases} R_{\text{teste}} = R^* = 2000000 \\ I_{d_{\text{teste}}} = I_d^* = 170000 \\ I_{h_{\text{teste}}} = I_h^* = 30000 \\ S_{\text{teste}} = S^* = 7800000 \\ U = U^* = \frac{387}{434} \end{cases}$$

Fica então comprovado que este é um ponto de equilíbrio válido, bem como se encontra o valor de U para cumprir esta situação.

C. Linearização do Sistema

Podemos agora linearizar o sistema em torno do ponto de funcionamento (S^*, I_d^*, I_h^*, R^*) . Realizaremos uma aproximação por uma série de Taylor de primeira ordem, para S, I_d , I_h , R, U e V, com $S = S - S^*$, $I_d = I_d - I_d^*$, $I_h = I_h - I_h^*$, $R = R - R^*$, $U = U - U^*$ e $V = v - v^*$:

Seja $\dot{S} = f(t)$, $\dot{I}_d = g(t)$, $\dot{I}_h = h(t)$ e $\dot{R} = i(t)$. Por uma razão de simplificação, consideremos o ponto de equilíbrio como x. As derivadas parciais correspondem às derivadas parciais no ponto de equilíbrio assumido:

$$\dot{S} = f(x) + (S - S^*) \cdot \frac{\partial f}{\partial S} + (I_d - I_d^*) \cdot \frac{\partial f}{\partial I_d} + (I_h - I_h^*) \cdot \frac{\partial f}{\partial I_h} + (R - R^*) \cdot \frac{\partial f}{\partial R} + (U - U^*) \cdot \frac{\partial f}{\partial U} + (v - v^*) \cdot \frac{\partial f}{\partial v}$$

$$\dot{I}_d = g(x) + (S - S^*) \cdot \frac{\partial g}{\partial S} + (I_d - I_d^*) \cdot \frac{\partial g}{\partial I_d} + (I_h - I_h^*) \cdot \frac{\partial g}{\partial I_h} + (R - R^*) \cdot \frac{\partial g}{\partial R} + (U - U^*) \cdot \frac{\partial g}{\partial U} + (v - v^*) \cdot \frac{\partial g}{\partial v}$$

$$\dot{I}_h = h(x) + (S - S^*) \cdot \frac{\partial h}{\partial S} + (I_d - I_d^*) \cdot \frac{\partial h}{\partial I_d} + (I_h - I_h^*) \cdot \frac{\partial h}{\partial I_h} + (R - R^*) \cdot \frac{\partial h}{\partial R} + (U - U^*) \cdot \frac{\partial h}{\partial U} + (v - v^*) \cdot \frac{\partial h}{\partial v}$$

$$\dot{R} = i(x) + (S - S^*) \cdot \frac{\partial i}{\partial S} + (I_d - I_d^*) \cdot \frac{\partial i}{\partial I_d} + (I_h - I_h^*) \cdot \frac{\partial i}{\partial I_h} + (R - R^*) \cdot \frac{\partial i}{\partial R} + (U - U^*) \cdot \frac{\partial i}{\partial U} + (v - v^*) \cdot \frac{\partial i}{\partial v}$$

Sabemos que o valor destas funções para o ponto de funcionamento é nulo. Assim:

$$\dot{S} = S \left(-v^* - \frac{(R_0 - U^*) \gamma I_d^*}{N} \right) + I_d \left(-\frac{(R_0 - U^*) \gamma S^*}{N} \right) + \rho R + U \left(\frac{\gamma S^* I_d^*}{N} \right) - V S^*$$

$$\dot{I}_d = S \left(0,85 \frac{(R_0 - U^*) \gamma I_d^*}{N} \right) + I_d \left(0,85 \frac{(R_0 - U^*) \gamma S^*}{N} - \gamma \right) + U \left(-0,85 \frac{\gamma S^* I_d^*}{N} \right)$$

$$\dot{I}_h = S \left(0,15 \frac{(R_0 - U^*) \gamma I_d^*}{N} \right) + I_d \left(0,15 \frac{(R_0 - U^*) \gamma S^*}{N} \right) - I_h \gamma + U \left(-0,15 \frac{\gamma S^* I_d^*}{N} \right)$$

$$\dot{R} = S v^* + I_d \gamma + I_h \gamma - \rho R + V S^*$$

Se considerarmos:

$$\begin{cases} R^* = \frac{\alpha^* + v^*}{\rho} S^* \\ U^* = R_0 - \frac{N}{0,85} \\ \frac{I_h^*}{I_d^*} = \frac{0,15}{0,85} \\ S^* = \frac{\gamma}{\alpha^*} (I_h^* + I_d^*) \end{cases}$$

Podemos escrever:

$$\begin{cases} \dot{S} = S \left(-v^* - \frac{\gamma I_d^*}{0,85 S^*} \right) + I_d \left(-\frac{\gamma}{0,85} \right) + \rho R + U \left(\frac{\gamma S^* I_d^*}{N} \right) - V S^* \\ \dot{I}_d = S \left(\frac{\gamma I_d^*}{S^*} \right) - U \left(0,85 \frac{\gamma S^* I_d^*}{N} \right) \\ \dot{I}_h = S \left(0,15 \frac{\gamma I_h^*}{S^*} \right) + I_d \left(\frac{\gamma I_h^*}{I_d^*} \right) - I_h \gamma - U \left(0,15 \frac{\gamma S^* I_d^*}{N} \right) \\ \dot{R} = S v^* + I_d \gamma + I_h \gamma - \rho R + V S^* \end{cases}$$

D. Representação do Espaço de Estados

Após a linearização, obtemos um sistema linear invariante no tempo, cujo espaço de estados pode ser descrito por estas duas equações:

$$\dot{x}(t) = A x(t) + B u(t)$$

$$y(t) = C x(t) + D u(t)$$

Nas quais:

$$x(t) = \begin{bmatrix} S \\ I_d \\ I_h \\ R \end{bmatrix}, y(t) = \begin{bmatrix} I_h \\ I_d + I_h \end{bmatrix} \text{ e } u(t) = \begin{bmatrix} U \\ V \end{bmatrix}$$

Relativamente à primeira equação, pode ser obtida na forma pretendida, reescrevendo as equações resultantes da linearização na forma matricial:

$$\dot{x}(t) = \begin{bmatrix} -v^* - \frac{\gamma I_d^*}{0,85 S^*} & -\frac{\gamma}{0,85} & 0 & \rho \\ \frac{\gamma I_d^*}{S^*} & 0 & 0 & 0 \\ \frac{\gamma I_h^*}{S^*} & \frac{\gamma I_h^*}{I_d^*} & -\gamma & 0 \\ v^* & \gamma & \gamma & -\rho \end{bmatrix} \times \begin{bmatrix} S \\ I_d \\ I_h \\ R \end{bmatrix} + \begin{bmatrix} \frac{\gamma I_d^* S^*}{N} & -S^* \\ -0,85 \frac{\gamma I_d^* S^*}{N} & 0 \\ -0,15 \frac{\gamma I_d^* S^*}{N} & 0 \\ 0 & S^* \end{bmatrix} \times \begin{bmatrix} U \\ V \end{bmatrix}$$

E que, portanto:

$$A = \begin{bmatrix} -v^* - \frac{\gamma * I_d^*}{0,85 * S^*} & -\frac{\gamma}{0,85} & 0 & \rho \\ \frac{\gamma * I_d^*}{S^*} & 0 & 0 & 0 \\ \frac{\gamma * I_h^*}{S^*} & \frac{\gamma * I_h^*}{I_d^*} & -\gamma & 0 \\ v^* & \gamma & \gamma & -\rho \end{bmatrix}$$

$$B = \begin{bmatrix} \frac{\gamma * Id^* * S^*}{N} & -S^* \\ -0.85 \frac{\gamma * Id^* * S^*}{N} & 0 \\ -0.15 \frac{\gamma * Id^* * S^*}{N} & 0 \\ 0 & S^* \end{bmatrix}$$

Reescrever a segunda equação é um processo mais trivial:

$$y(t) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} S \\ I_d \\ I_h \\ R \end{bmatrix}$$

Como podemos observar, as entradas $u(t)$ não são relevantes na representação das saídas y . Temos:

$$C = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix} \text{ e } D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

E. Controlabilidade, Observabilidade e Estabilidade

A controlabilidade e a observabilidade serão testadas para vários conjuntos de combinações de entradas e saídas. A constante, neste caso, será a matriz A , que permanecerá imutável. Após a substituição das variáveis que a constituem pelos seus valores reais, será:

$$A = \begin{bmatrix} -\frac{1}{234} & -\frac{10}{51} & 0 & \frac{1}{60} \\ \frac{17}{4680} & 0 & 0 & 0 \\ \frac{1}{1560} & \frac{1}{34} & -\frac{1}{6} & 0 \\ 0 & \frac{1}{6} & \frac{1}{6} & -\frac{1}{60} \end{bmatrix}$$

A primeira combinação proposta é o teste da controlabilidade quando só temos uma entrada para atuar, ou seja, $u=U$. Neste caso, a matriz B será:

$$B1 = \begin{bmatrix} \frac{\gamma * Id^* * S^*}{N} \\ -0.85 \frac{\gamma * Id^* * S^*}{N} \\ -0.15 \frac{\gamma * Id^* * S^*}{N} \\ 0 \end{bmatrix} = \begin{bmatrix} 22100 \\ -18785 \\ -3315 \\ 0 \end{bmatrix}$$

A segunda combinação proposta é o teste da controlabilidade quando temos duas entradas para atuar, ou seja, $u=(U,V)$. Neste caso, a matriz B será:

$$B2 = \begin{bmatrix} \frac{\gamma * Id^* * S^*}{N} & -S^* \\ -0.85 \frac{\gamma * Id^* * S^*}{N} & 0 \\ -0.15 \frac{\gamma * Id^* * S^*}{N} & 0 \\ 0 & S^* \end{bmatrix} = \begin{bmatrix} 22100 & -7800000 \\ -18785 & 0 \\ -3315 & 0 \\ 0 & 7800000 \end{bmatrix}$$

Um sistema LIT é controlável se existir um vetor de entrada $u(t)$ para $0 \leq t \leq T$, com $T > 0$ e finito, tal que o sistema vai da condição inicial $x(0) = 0$ para qualquer estado x no intervalo de tempo T .

De certa forma, um sistema é dito controlável no tempo se for possível transferir o sistema de qualquer estado inicial $x(t_0)$ para qualquer estado $x(t_1)$ por meio de um vetor de controlo, num intervalo de tempo finito.

Seja um sistema de ordem n , dado por $\dot{x}(t) = Ax(t) + Bu(t)$. Para um sistema controlável basta analisar a equação dos estados, ou seja, o par de matrizes A e B .

Definindo a matriz de controlabilidade M :

$$M = [B \quad AB \quad A^2B \quad A^3B]$$

O sistema definido pelas matrizes (A,B) é controlável se a característica for igual ao mínimo entre o número de linhas e colunas, a característica máxima, que pode ser obtida pelo valor de n (no nosso caso, $n=4$). Uma alternativa, se a matriz em questão for quadrada, é utilizar o determinante. Se o seu valor for diferente de 0, então o sistema é controlável.

Utilizou-se o seguinte programa de MATLAB para verificar a controlabilidade dos casos apresentados:

```
%%Uma Entrada - Controlabilidade
M_one=[B_onein A*B_onein A*A*B_onein A*A*A*B_onein];
disp(rank(M_one));
if(rank(M_one)==4)
    disp('Controlável com uma entrada');
else
    disp('Não Controlável com uma entrada');
end
%%Duas Entradas - Controlabilidade
M_two=[B_twoin A*B_twoin A*A*B_twoin A*A*A*B_twoin];
disp(rank(M_two));
if(rank(M_two)==4)
    disp('Controlável com duas entradas');
else
    disp('Não Controlável com duas entradas');
end
```

Figura 3 - Código para verificar a controlabilidade

A matriz M_{one} calculada com B_{onein} , a matriz equivalente à matriz $B1$, corresponde ao primeiro caso apresentado. Por seu lado, M_{two} corresponde ao segundo caso, com B_{twoin} como equivalente a $B2$.

Os resultados obtidos permitem-nos concluir que, em ambos os casos, o sistema é não controlável, uma vez que a característica obtida para ambas as matrizes M é 2.

Para a verificação da observabilidade, a primeira combinação consiste em ter apenas uma saída no sistema, ou seja, $y=I_h$. Neste caso, a matriz C será:

$$C1 = [0 \quad 0 \quad 1 \quad 0]$$

A segunda combinação proposta é o teste da observabilidade quando temos duas saídas, ou seja, $u=(I_h, I_h + I_d)$. Neste caso, a matriz C será:

$$C2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

Um sistema é completamente observável se toda transição de estado afeta cada um dos elementos do vetor de saída.

Este conceito é útil na reconstrução (estimação) de variáveis de estado não mensuráveis, a partir de variáveis mensuráveis. É possível obter informação sobre (observar) o estado inicial a partir da entrada e da saída.

Definindo a matriz de observabilidade Q :

$$Q = \begin{bmatrix} C \\ CA \\ CA^2 \\ CA^3 \end{bmatrix}$$

À semelhança da controlabilidade, podemos concluir se um sistema é observável através da característica da sua matriz. Se esta for igual à característica máxima, então o sistema é observável. Uma alternativa, se a matriz em questão for quadrada, é utilizar o determinante. Se o seu valor for diferente de 0, então o sistema é observável.

Utilizou-se o seguinte programa de MATLAB para verificar a controlabilidade dos casos apresentados:

```
%%Uma Saida - Observabilidade
Q_one=[C_oneout; C_oneout*A; C_oneout*A*A; C_oneout*A*A*A];
disp(rank(Q_one));
if(rank(Q_one)==4)
    disp('Observável com uma saída');
else
    disp('Não Observável com uma saída');
end

%%Duas saídas - Observabilidade
Q_two=[C_twoout; C_twoout*A; C_twoout*A*A; C_twoout*A*A*A];
disp(rank(Q_two));
if(rank(Q_two)==4)
    disp('Observável com duas saídas');
else
    disp('Não Observável com duas saídas');
end
```

Figura 4 - Código utilizado para verificar a observabilidade

A matriz Q_{one} calculada com C_{oneout} , a matriz equivalente à matriz $C1$, corresponde ao primeiro caso apresentado. Por seu lado, Q_{two} corresponde ao segundo caso, com C_{twoout} como equivalente a $C2$.

Os resultados obtidos permitem-nos concluir que, em ambos os casos, o sistema é observável, uma vez que a característica obtida para ambas as matrizes Q é 4.

A estabilidade de um sistema depende da localização dos seus pólos em malha aberta. Para os determinar, igualamos o determinante característico da matriz A a zero:

$$\det([sI - A]) = 0$$

Recorreu-se ao MATLAB de modo a resolver a equação apresentada:

```
disp(det(s*eye(4)-A));
f=det(s*eye(4)-A) == 0;
j=solve(f,s);
disp(j);
```

Figura 5 - Código utilizado para solucionar a equação

Foram obtidos, tal como esperado, 4 polos:

$$\begin{cases} \text{polo}_1 = -\frac{1}{6} \\ \text{polo}_2 = 0 \\ \text{polo}_3 = -\frac{49}{4680} - j\frac{44}{1695} \\ \text{polo}_4 = -\frac{49}{4680} + j\frac{44}{1695} \end{cases}$$

O sistema é estável se toda a entrada limitada tem uma resposta limitada. Se todos os polos pertencerem ao semiplano esquerdo (SPE), podemos assegurar que é estável. No nosso

caso em específico, temos 3 polos no SPE e 1 na origem. O polo na origem faz com que esteja na fronteira da região de estabilidade. Se um dos zeros do sistema anular este polo na origem, podemos afirmar que é estável.

Existem duas formas de obter os zeros do sistema: uma delas é o cálculo da função de transferência, outra é a obtenção da solução da seguinte equação:

$$\det\left(\begin{bmatrix} sI - A & -B \\ C & D \end{bmatrix}\right) = 0$$

Serão calculados os zeros para duas situações: o caso original, com duas entradas e duas saídas; o caso que será utilizado em pontos futuros, com uma entrada e uma saída. Recorreu-se ao MATLAB para solucionar o problema:

```
%%Cálculo dos zeros: 2 entradas, 2 saídas
f=det([s*eye(4)-A -B_twoin; C_twoout [0 0;0 0]])==0;
j=solve(f,s);
disp(j);

%%Cálculo dos zeros: 1 entrada, 1 saída
f=det([s*eye(4)-A -B_onein; C_oneout 0])==0;
j=solve(f,s);
disp(j);
```

Figura 6 - Código utilizado para determinar os zeros

No caso original foi obtido apenas um zero, localizado na origem. Para o segundo caso, foram encontrados três zeros, um deles na origem, outro em $-\frac{1}{6}$ e, por fim, um em $-\frac{1}{60}$.

Estes resultados permitem-nos concluir que o polo causador de instabilidade no sistema é, em qualquer um dos casos, anulados por um dos zeros, tornando o sistema estável.

Um sistema é estabilizável se conseguirmos projetar um controlador que o torne estável, ou seja, se conseguirmos que em malha fechada o sistema fique estável. Isto é, apenas é estabilizável se todos os seus modos dinâmicos instáveis forem controláveis.

Para calcular os polos em malha fechada, devemos partir da seguinte expressão:

$$\det([sI - (A - BK)]) = 0$$

Em que K é a matriz $K = [k_4 \ k_3 \ k_2 \ k_1]$.

Mais uma vez, a solução para esta equação foi determinada com recurso ao MATLAB:

```
k=[k4 k3 k2 k1];
disp(s*eye(4)-(A-B_onein*k));
disp(det(x-(A-B_onein*k)));
f=det(x-(A-B_onein*k)) == 0;
j=solve(f,s);
disp(j);
```

Figura 7 - Código utilizado para determinar os polos

Os resultados obtidos foram os seguintes:

$$\begin{cases} -\frac{1}{6} \\ 0 \\ \text{expressão}_1 \text{ de } k \\ \text{expressão}_2 \text{ de } k \end{cases}$$

$$\text{expressão}_1 = \frac{3315k^2}{2} + \frac{18785k^3}{2} - 11050k^4 - 3 \left(\frac{1221025k^2}{4} + \frac{20757425k^2k^3}{6} - \frac{12210250k^2k^4}{3} + \frac{493k^2}{216} + \frac{352876225k^3}{36} - \frac{207574250k^3k^4}{9} + \frac{8381k^3}{648} + \frac{122102500k^4}{9} - \frac{68765k^4}{162} + \frac{11050k^4}{27} - \frac{14759}{197121600} \right)^{\frac{1}{2}} - \frac{49}{4680}$$

$$\text{expressão}_2 = \frac{3315k^2}{2} + \frac{18785k^3}{2} - 11050k^4 + 3 \left(\frac{1221025k^2}{4} + \frac{20757425k^2k^3}{6} - \frac{12210250k^2k^4}{3} + \frac{493k^2}{216} + \frac{352876225k^3}{36} - \frac{207574250k^3k^4}{9} + \frac{8381k^3}{648} + \frac{122102500k^4}{9} - \frac{68765k^4}{162} + \frac{11050k^4}{27} - \frac{14759}{197121600} \right)^{\frac{1}{2}} - \frac{49}{4680}$$

Tal como esperávamos, dois dos polos obtidos em malha aberta não são alteráveis. Isto deve-se ao facto de a característica obtida ser 2, representando este valor o máximo de polos controláveis, isto é, que podem ser colocados em qualquer posição pretendida. Não podemos, portanto, assegurar que o sistema é sempre estabilizável, uma vez que um dos polos fixos se localiza na origem.

F. Representações Canónicas Controlável, Observável e Diagonal

Para o caso de uma entrada e uma saída ($u=U$, $y=I_h$), tentaram-se escrever as representações canónicas controlável, observável e diagonal.

Em primeiro lugar, iremos analisar a forma canónica controlável. Sabemos, pelo que foi calculado anteriormente, que o sistema não é controlável, logo não será possível representá-lo na forma canónica controlável. No entanto, optamos por demonstrá-lo.

Para este cálculo, devemos obter a matriz M caracterizada por $M = [B \quad AB \quad A^2B \quad A^3B]$:

$$M = \begin{bmatrix} 22100 & \frac{32300}{9} & -\frac{65374}{707} & -\frac{3053}{3487} \\ -18785 & \frac{1445}{18} & \frac{7131}{547} & -\frac{439}{1307} \\ -3315 & \frac{85}{6} & \frac{1615}{702} & -\frac{798}{13463} \\ 0 & -\frac{11050}{3} & \frac{4165}{54} & \frac{2103}{1655} \end{bmatrix}$$

De seguida, devemos calcular a matriz W, definida por:

$$W = \begin{bmatrix} a_{n-1} & a_{n-2} & \dots & a_1 & 1 \\ a_{n-2} & a_{n-3} & \dots & 1 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_1 & 1 & \dots & 0 & 0 \\ 1 & 0 & \dots & 0 & 0 \end{bmatrix}$$

No nosso caso:

$$W = \begin{bmatrix} a_3 & a_2 & a_1 & 1 \\ a_2 & a_1 & 1 & 0 \\ a_1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Em que a_1 , a_2 e a_3 se obtêm através do polinómio característico de A:

$$\det([sI - A]) = s^4 + \frac{439s^3}{2340} + \frac{s^2}{234} + \frac{11s}{84240}$$

$$\text{Logo, } a_1 = \frac{439}{2340}, a_2 = \frac{1}{234} \text{ e } a_3 = \frac{11}{84240}.$$

$$W = \begin{bmatrix} \frac{11}{84240} & \frac{1}{234} & \frac{439}{2340} & 1 \\ \frac{1}{234} & \frac{439}{2340} & 1 & 0 \\ \frac{439}{2340} & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Poderíamos agora obter a matriz de mudança de base T, através de $T = MW$:

$$T = \begin{bmatrix} 0 & \frac{12155}{18} & 7735 & 22100 \\ 0 & -\frac{3757}{72} & -\frac{41327}{12} & -18785 \\ 0 & -\frac{221}{24} & -\frac{2431}{4} & -3315 \\ 0 & -\frac{5525}{9} & -\frac{11050}{3} & 0 \end{bmatrix}$$

Como podemos observar, a primeira coluna da matriz T é nula, o que significa que não pode ser invertida. Como a forma canónica seria obtida por $T^{-1}AT$, podemos comprovar que tal não é possível.

Para a obtenção destas matrizes, recorreu-se ao seguinte código em MATLAB:

```
%Forma Canónica Controlável
disp('Forma Canónica Controlável'); disp(' ');
syms s;
pol=[s 0 0 0; 0 s 0 0; 0 0 s 0; 0 0 0 s]-A;
disp('Determinante do polinómio característico de A para saber a3 a2 a1'); disp(det(pol));
a1=439/2340;
a2=1/234;
a3=11/84240;
W=[a3 a2 a1 1; a2 a1 1 0; a1 1 0 0; 1 0 0 0];
M=[B A*B A*A*B A*A*A*B];
T=M*W;
disp('Matriz de controlabilidade, M:'); disp(M);
disp('Matriz de Mudança de Base, T:'); disp(T);
disp('Nova A:'); disp(inv(T)*A*T);
disp('Nova B:'); disp(inv(T)*B);
disp('-----');
```

Figura 8 - Código utilizado para o cálculo das matrizes

O passo seguinte será colocar o sistema na forma canónica observável, para uma entrada e uma saída ($u=U$, $y=I_h$):

Em primeiro lugar, devemos obter a matriz Q para este caso:

$$Q = \begin{bmatrix} C \\ CA \\ CA^2 \\ CA^3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ \frac{1}{1560} & \frac{1}{34} & -\frac{1}{6} & 0 \\ -\frac{1}{365040} & -\frac{1989}{13} & \frac{1}{36} & \frac{1}{93600} \\ 0 & \frac{15867}{80599} & -\frac{373}{80599} & 0 \end{bmatrix}$$

A matriz W a utilizar é igual à calculada anteriormente. Podemos avançar para o cálculo da matriz $U = (WQ)^{-1}$:

$$U = \begin{bmatrix} -\frac{248997}{4} & \frac{82999}{8} & -\frac{8119}{48} & -\frac{828}{185} \\ \frac{9497}{7} & -\frac{9497}{42} & \frac{9497}{252} & -\frac{1909}{3107} \\ 0 & 0 & 0 & 1 \\ 622493 & -\frac{40595}{4} & \frac{4864}{37} & \frac{7309}{1787} \end{bmatrix}$$

Podemos obter a forma canónica observável, dada por:

$$A_o = U^{-1}AU, B = U^{-1}B \text{ e } C_o = CU$$

$$A_o = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -\frac{11}{84240} \\ 0 & 1 & 0 & -\frac{1}{234} \\ 0 & 0 & 1 & -\frac{439}{2340} \end{bmatrix}$$

$$B_o = \begin{bmatrix} 0 \\ \frac{221}{24} \\ 2431 \\ -3315 \end{bmatrix}$$

$$C_o = [0 \quad 0 \quad 0 \quad 1]$$

Para obter as matrizes apresentadas, recorreu-se ao MATLAB, do seguinte modo:

```
%Forma Canonica Observavel
disp('Forma Canonica Observavel'); disp(' ');
Q=[C; C*A; C*A*A; C*A*A*A];
U=inv(W*Q);
disp('Matriz de observabilidade, Q:'); disp(Q);
disp('Matriz de Mudança de Base, U:'); disp(U);
disp('Nova A:'); disp(inv(U)*A*U);
disp('Nova C:'); disp(C*U);
disp('-----');
```

Figura 9 - Código utilizado para o cálculo das matrizes

Há diversas formas de obter a forma canónica diagonal. Decidimos utilizar a fórmula dos vetores próprios, uma vez que o MATLAB nos fornece uma função (eig()) que nos permite calculá-los de uma forma muito mais rápida.

Designamos a matriz P, como a matriz de mudança de base, ou seja, a matriz com os vetores próprios.

$$P = \begin{bmatrix} -\frac{665}{923} & -\frac{665}{923} & 0 & \frac{283}{3822} \\ \frac{346}{9893} + \frac{355}{4094}i & \frac{346}{9893} - \frac{355}{4094}i & \frac{355}{4192} & -\frac{29}{17970} \\ \frac{83}{13448} + \frac{12874}{12874}i & \frac{83}{13448} - \frac{12874}{12874}i & \frac{117}{7829} & \frac{1153}{1726} \\ \frac{447}{658} - \frac{552}{5411}i & \frac{447}{658} + \frac{552}{5411}i & \frac{269}{270} & -\frac{659}{890} \end{bmatrix}$$

Para obter a forma pretendida:

$$A_d = P^{-1}AP, B_d = e \ C_d$$

$$A_d = \begin{bmatrix} -\frac{49}{4680} + \frac{44}{1695}i & 0 & 0 & 0 \\ 0 & -\frac{49}{4680} - \frac{44}{1695}i & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{6} \end{bmatrix}$$

$$B_d = \begin{bmatrix} -\frac{291404}{19} + 102132i \\ -\frac{291404}{19} - 102132i \\ \frac{1}{16169288644} + \frac{1}{134902129590}i \\ \frac{1}{15131305040007}i \end{bmatrix}$$

$$C_d = \left[\frac{83}{13448} + \frac{197}{12874}i \quad \frac{83}{13448} - \frac{197}{12874}i \quad \frac{117}{7829} \quad \frac{1153}{1726} \right]$$

O programa a utilizar no MATLAB é bastante simples:

```
%Forma Canonica Diagonal
disp('Forma Canonica Diagonal'); disp(' ');
[V,D]=eig(A);
P=V;
disp('Matriz de Mudança de Base, P:'); disp(P);
disp('Nova A:'); disp(inv(P)*A*P);
```

Figura 10 - Código utilizado para o cálculo das matrizes

Para todos os casos anteriores, D=0.

G. Função de Transferência

Neste ponto, iremos calcular a função de transferência a partir da representação no espaço dos estados para o caso de uma entrada e uma saída ($u=U$, $y=I_h$). A função de transferência pode ser obtida através de:

$$G(s) = C[sI - A]^{-1}B + D$$

Recorremos ao MATLAB, de modo a calcular a função:

```
disp('Função de transferência, G:');
syms s
G=C*inv([s 0 0 0; 0 s 0 0; 0 0 s 0; 0 0 0 s]- A)*B+D;
G=simplify(G);
disp(G);
```

Figura 11 - Código utilizado para calcular a função de transferência

A função de transferência obtida apresentava um zero e dois polos, sendo que os seus valores coincidem com aqueles previstos anteriormente (dos três zeros e dos quatro polos previstos, dois de cada anulam-se mutuamente, resultando uma função com um zero e dois polos):

$$G(s) = -775710 \frac{60s + 1}{14040s^2 + 294s + 11} \Leftrightarrow$$

$$\Leftrightarrow G(s) = -3315 \frac{s + \frac{1}{60}}{s^2 + \frac{49}{2340}s + \frac{11}{14040}}$$

H. Resposta Temporal Transitória

Serão considerados dois casos distintos para a análise temporal transitória:

- Condições iniciais nulas e controlo em degrau, em que U passa de U^* para $0.9U^*$;
- Controlo nulo e condições iniciais não nulas, em que I_d é igual a $1.2I_d^*$.

Para ambos os casos, recorremos a uma função do MATLAB, a função *lsim()* à qual são fornecidos vários parâmetros, nomeadamente o espaço de estados, o intervalo de tempo a ser visualizado, o sinal de entrada e, se necessárias, as condições iniciais.

Abordaremos, então, o caso i). Uma vez que não é possível materializar no MATLAB esta função (devido à forma como o programa interpreta o *step*), optamos por simular esta variação com uma equivalente em termo da diferença verificada. A entrada U passará de 0 a $-0.1U^*$. O *script* utilizado toma a seguinte forma:

```
%%Condições iniciais nulas
T = 0:0.01:600; % simulation time = 600 seconds
U = -0.1*U_eq+0*T; % input as a function of time
sys = ss(A,B,C,D); % construct a system model
figure(2);clf;
[Y, Tsim, X] = lsim(sys,U,T); % simulate
plot(Tsim,Y) % plot the output vs. time
title('Response to an Input with Zero Initial Conditions')
```

Figura 12 - Código utilizado para a resposta com condições iniciais nulas

A simulação foi efetuada para o equivalente a 600 unidades de tempo, tendo produzido os seguintes resultados:

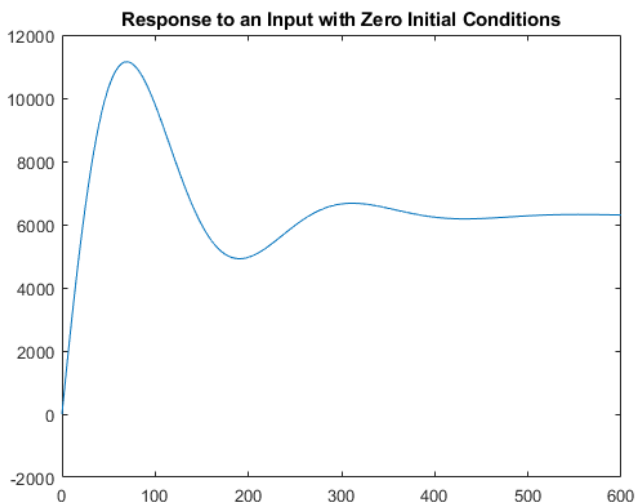


Figura 13 - Resposta Transitória com condições iniciais nulas

A diminuição nas medidas de contenção caracterizadas por U provocam um efeito na saída, que representa I_h , perto dos 12000 infetados. O resposta é sub-amortecida, tal como era esperado, devido à localização dos polos existentes (o polo em $-1/6$ e o polo na origem são ambos anulados, restando apenas os dois polos imaginários no SPE). A saída eventualmente estabiliza em torno dos 6000 infetados graves.

Resta-nos simular o caso ii). O enunciado é bastante incompleto acerca das condições iniciais pretendidas. O único requisito apresentado era que I_d fosse igual a $1.2I_d^*$. Uma vez que não faz sentido simular uma situação com condições iniciais em que não há suscetíveis, considerou-se

que, não existindo Infetados graves e recuperados, $S=N-I_d$, com $N=10^7$. O *script* utilizado toma a seguinte forma:

```
%%Condições iniciais não nulas
T = 0:0.01:600; % simulation time = 600 seconds
U = zeros(size(T)); % no input
X0 = [1e7-1.2*I_d_eq 1.2*I_d_eq 0 0]; % initial conditions of the three states
sys = ss(A,B,C,D); % construct a system model
figure(1);clf;
lsim(sys, U, T, X0) % simulate and plot the response (the output)
title('Response to Non-Zero Initial Conditions')
```

Figura 14 - Código utilizado para a resposta com condições iniciais não nulas

A simulação foi efetuada para o equivalente a 600 unidades de tempo, tal como no caso anterior, tendo produzido os seguintes resultados:

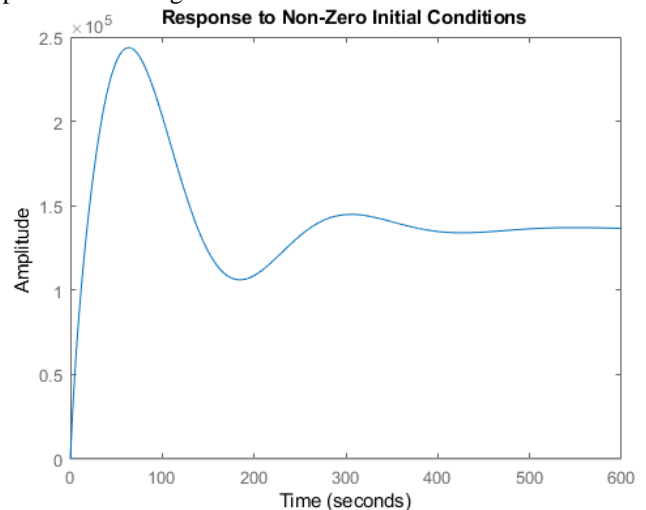


Figura 15 - Resposta Transitória com condições iniciais não nulas

Essencialmente, o comportamento é do mesmo tipo que o anterior. No entanto, a saída atinge valores muito mais altos. Seria semelhante a um dos picos, como os visualizados na parte I deste trabalho, em que não existia controlo aplicado. A saída acaba por estabilizar num valor próximo dos 150000, ainda bastante elevado.

III. PROJECTO DE CONTROLADOR EM TEMPO CONTÍNUO POR ESPAÇO DE ESTADOS

Será desenvolvido um controlador que nos permita regular o funcionamento do sistema, à semelhança do que foi realizado no trabalho anterior. Neste caso, no entanto, o dimensionamento será feito com recurso à análise do espaço de estados.

A. Controlador Linearmente Realimentado

Antes de avançarmos para o dimensionamento do controlador, devemos relembrar um fator importante que já foi previamente abordado. A matriz de controlabilidade para o caso proposto, um modelo de espaço de estados em que todo o estado é mensurável e $u=U$, apresenta uma característica igual a 2. Sabemos que a característica máxima para este caso é 4. Com base nisto, sabemos que apenas podemos mover 2 dos 4 polos em malha aberta, algo que devemos ter em conta para calcular a matriz K do controlador.

Primeiramente, começamos por observar os polos em malha aberta e os polos em malha fechada, anteriormente calculados:

$$\left\{ \begin{array}{l} \text{polo}_1 = -\frac{1}{6} \\ \text{polo}_2 = 0 \\ \text{polo}_3 = -\frac{49}{4680} - j\frac{44}{1695} \\ \text{polo}_4 = -\frac{49}{4680} + j\frac{44}{1695} \end{array} \right., \text{ em malha aberta.}$$

$$\left\{ \begin{array}{l} -\frac{1}{6} \\ 0 \end{array} \right., \text{ em malha fechada, com:}$$

expressão₁ de k
expressão₂ de k

$$\text{expressão}_1 = \frac{3315k_2}{2} + \frac{18785k_3}{2} - 11050k_4 - 3 \left(\frac{1221025k_2^2}{4} + \frac{20757425k_2k_3}{6} - \frac{12210250k_2k_4}{3} + \frac{493k_2}{216} + \frac{352876225k_3^2}{36} - \frac{207574250k_3k_4}{9} + \frac{8381k_3}{648} + \frac{122102500k_4^2}{9} - \frac{68765k_4}{162} + \frac{11050k_1}{27} - \frac{14759}{197121600} \right)^{\frac{1}{2}} - \frac{49}{4680}$$

$$\text{expressão}_2 = \frac{3315k_2}{2} + \frac{18785k_3}{2} - 11050k_4 + 3 \left(\frac{1221025k_2^2}{4} + \frac{20757425k_2k_3}{6} - \frac{12210250k_2k_4}{3} + \frac{493k_2}{216} + \frac{352876225k_3^2}{36} - \frac{207574250k_3k_4}{9} + \frac{8381k_3}{648} + \frac{122102500k_4^2}{9} - \frac{68765k_4}{162} + \frac{11050k_1}{27} - \frac{14759}{197121600} \right)^{\frac{1}{2}} - \frac{49}{4680}$$

Conforme podemos observar, dois dos polos obtidos em malha aberta não podem ser movidos em malha fechada (0 e -1/6). Um destes polos pode ser causador de instabilidade, se não for anulado por um zero do sistema, por exemplo.

A questão essencial que surge neste ponto é como dimensionar os outros dois polos. Optamos por tentar reduzir a sua influência no sistema, tornando-os puramente reais, mantendo a sua localização estável (SPE) e afastando-os do eixo imaginário. Ao torná-los puramente reais estaremos a eliminar as oscilações que se verificam na resposta original do sistema que pode ser obtida através do seguinte código:

```
figure(1);clf;
step(A,B,C,D);
```

Figura 16 -Código utilizado para a resposta de referência

Podemos observar a referida oscilação na sua resposta na imagem seguinte. Para além disso, o pico tem um valor bastante elevado em módulo:

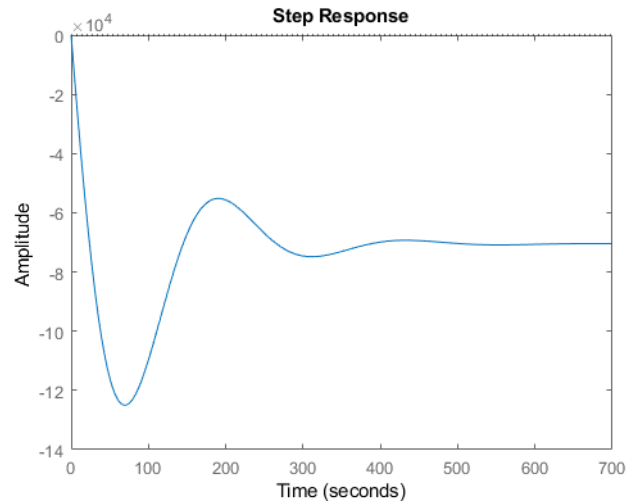


Figura 17 - Resposta de referência

Decidimos testar dois valores distintos para os pólos que pretendemos obter: -1 e -2. Tendo 4 graus de liberdade para obter duas soluções, decidimos fixar dois valores de K. Por isso, seleccionamos k₂=k₁=1. Recorremos ao seguinte programa em MATLAB para resolver o sistema de duas equações:

```
%Polos
syms s;
pol_ma=(s^4+0*s^3+0*s^2+0*s+0)-A;
eq_ma=det(pol_ma)==0;
s_ma=solve(eq_ma,s);
syms k4 k3 k2 k1;
pol_mf=(s^4+0*s^3+0*s^2+0*s+0)-A+B_onein*[k4 k3 k2 k1];
eq_mf=det(pol_mf)==0;
s_mf=solve(eq_mf,s);
disp('Polos malha aberta:');disp(s_ma);
disp('Polos malha fechada:');disp(s_mf);
sol1=subs(s_mf(3,1),{k2,k1},{1,1});
sol2=subs(s_mf(4,1),{k2,k1},{1,1});
disp(sol1);
disp(sol2);
eq1=sol1==1;
eq2=sol2==1;
final=solve([eq1 eq2],[k3 k4]);
disp('k3:');disp(double(final.k3));
disp('k4:');disp(double(final.k4));
```

Figura 18 - Código utilizado para o cálculo de k3 e k4

Obtivemos ao colocar os polos em -1:

$$K = \begin{bmatrix} 3813 & 4921 & 1 & 1 \\ 3812 & 4920 & 1 & 1 \end{bmatrix}$$

Ao colocar os polos em -2 o resultado foi o seguinte:

$$K = \begin{bmatrix} 959 & 938 & 1 & 1 \\ 958 & 937 & 1 & 1 \end{bmatrix}$$

Apesar de a diferença parecer bastante ligeira, veremos que produz efeitos notórios no sistema realimentado. Para visualizarmos a resposta ao degrau, podemos alterar ligeiramente o código da Figura 16:

```
figure(2);clf
step(double(A-B*k),B,C,D);
```

Figura 19 - Alterações efetuadas ao código da Figura 16

Para o caso da colocação dos dois polos móveis em -1, obtém-se a seguinte resposta:

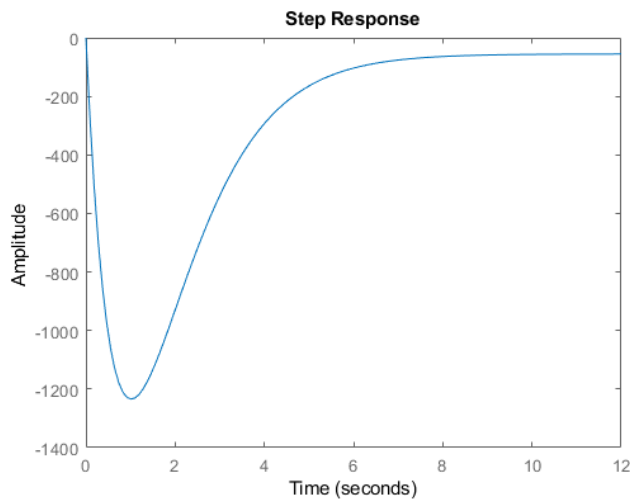


Figura 20 - Resposta do sistema com pólos em -1

É evidente a redução significativa do tempo de assentamento, que passou das centenas de segundos para aproximadamente 7 segundos. Para além disso, o pico está bastante mais pequeno. O sistema mantém-se estável, não apresentando agora oscilações. Apresentam-se, de seguida, os efeitos da escolha dos polos alternativos:

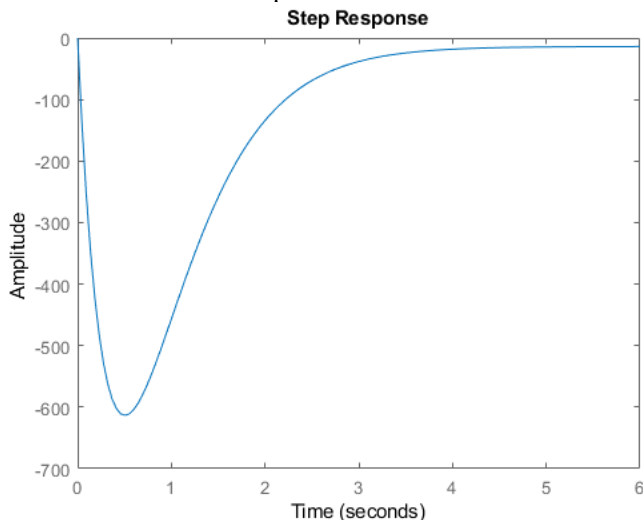


Figura 21 - Resposta do Sistema com pólos em -2

O tempo de assentamento tornou-se ainda menor com uma ligeira variação nos valores da matriz K. Para além disso, o valor em regime final é agora muito menor, bem como o módulo máximo do pico verificado.

Se continuássemos a diminuir o valor dos pólos esta tendência iria agravar-se, tornando-se o sistema mais rápido e mais suave. Um efeito potenciado pelo efeito do posicionamento dos polos e que já tinha sido previamente referido.

B. Projeto do Estimador de Estado

Considerando agora que apenas a saída do sistema é única, $y = I_h$, e diretamente acessível, todos os outros estados terão que ser estimados. Será projetado um estimador de estado tal que a norma do erro do estimador ao fim de dois dias seja um décimo da norma do erro inicial.

Ao fim de 2 dias, temos que ter um décimo da norma do erro inicial.

$$e^{2\beta} < 10^{-1} \Leftrightarrow \beta = -1.15129$$

Com base neste resultado, seleccionámos um $\beta = -2$, por ser o valor inteiro mais próximo que é menor do que o resultado.

O determinante característico é obtido através de:

$$pe(\lambda) = \det([\lambda I - (A - KeC)])$$

Podemos obter esta expressão simplificada através do MATLAB:

```
syms ke4 ke3 ke2 ke1 omega
ke = [ke4;ke3;ke2;ke1];
pe = det(omega*eye(4)-(A-ke*C_oneout));
disp(pe);
```

Figura 22 - Código utilizado para obter o polinomio

Tendo-se obtido:

$$pe(\lambda) = \lambda^4 + \lambda^3 \left(k_{e2} + \frac{439}{2340} \right) + \lambda^2 \left(k_{e2} \frac{49}{2340} + \frac{k_{e3}}{34} + \frac{k_{e4}}{1560} + \frac{1}{234} \right) + \lambda \left(\frac{k_{e1}}{93600} + k_{e2} \frac{11}{14040} + \frac{k_{e3}}{2040} + k_{e4} \frac{11}{93600} + \frac{11}{84240} \right) + \left(\frac{k_{e1} + k_{e2} + k_{e3} + k_{e4}}{561600} \right)$$

Para o valor de β definido, temos que:

$$pe(\lambda) = \lambda^4 + 8\lambda^3 + 24\lambda^2 + 32\lambda + 16$$

Assim, podemos resolver o sistema de quatro equações no MATLAB:

```
eq0=ke1/561600 + ke2/561600 + ke3/561600 + ke4/561600==16;
eq1=11/84240 + ke1/93600 + 11*ke2/14040 + ke3/2040 + 11*ke4/93600==32;
eq2=49*ke2/2340 + ke3/34 + ke4/1560 + 1/234==24;
eq3=ke2+439/2340==8;
eq_ke=solve([eq0 eq1 eq2 eq3],[ke1 ke2 ke3 ke4]);
disp('ke4:');disp(double(eq_ke.ke4));
disp('ke3:');disp(double(eq_ke.ke3));
disp('ke2:');disp(double(eq_ke.ke2));
disp('ke1:');disp(double(eq_ke.ke1));
```

Figura 23 - Código utilizado para determinar a matriz Ke

Tendo-se obtido:

$$Ke = \begin{bmatrix} -668087 \\ 92227 \\ 6 \\ 4539 \\ 581 \\ 9638308 \end{bmatrix}$$

Para a simulação, definiu-se um erro inicial máximo de norma 1, para o qual o estimador foi inicialmente dimensionado. Sabemos que o estado I_h é diretamente acessível através da saída do sistema, pelo que não faz sentido que o seu erro seja diferente de zero. Podemos utilizar como erro inicial a seguinte matriz:

$$e_0 = \begin{bmatrix} 0,7 \\ 0,7 \\ 0 \\ 0,7 \end{bmatrix}$$

Calculando a norma com recurso ao MATLAB, podemos concluir que a norma é menor que 1:

$$\|e_0\| = 0,9899$$

Foi necessário, para além do erro inicial, definir também condições iniciais para os estados que constituem x :

$$x(0) = \begin{bmatrix} S(0) \\ I_d(0) \\ I_h(0) \\ R(0) \end{bmatrix} = \begin{bmatrix} 9898000 \\ 10000 \\ 1000 \\ 1000 \end{bmatrix}$$

Sabendo que:

$$\dot{x} = Ax + BR - BK\tilde{x}; y = Cx$$

$$\dot{\tilde{x}} = (A - Bk)\tilde{x} + BR + K_e C(x - \tilde{x})$$

Logo:

$$\begin{bmatrix} \dot{\tilde{x}} \\ \dot{\tilde{x}} \end{bmatrix} = \begin{bmatrix} A & -BK \\ K_e C & A - BK - K_e C \end{bmatrix} \begin{bmatrix} x \\ \tilde{x} \end{bmatrix} + \begin{bmatrix} B \\ B \end{bmatrix} R$$

Para este caso, K é uma matriz nula, uma vez que estamos a considerar apenas os efeitos do estimador. O observador construído é um observador de *full-order*. Construímos, portanto, o seguinte programa em MATLAB de modo a estudar a evolução do estimador em relação ao sinal estimado:

```
e0 = 0.5;
vecnorm([e0;e0;e0;e0])
t = 0:0.01:5;
u = zeros(size(t));
xo = [10000000-1010000-1000; 100000; 1000; 1000];
xe = xo-[e0;e0;e0;e0]; % x~ = x - e

Ak = double([A, zeros(size(A)); ke*C, A-ke*C]); % Sem Controlo
Bk = [B; B];
Ck = [C, zeros(size(C))];
Dk = D;

sistema_2b = ss(Ak,Bk,Ck,Dk);

[y,t,x] = lsim(sistema_2b,u,t,[xo; xe]);

% Distribuir variáveis
n=4;
x_aux = x(:,1:n);
s = x_aux(:,1); id = x_aux(:,2); ih = x_aux(:,3); r = x_aux(:,4);

x_est = x(:,n+1:end);
s_est = x_est(:,1); id_est = x_est(:,2); ih_est = x_est(:,3); r_est = x_est(:,4);
```

Figura 24 - Código para a determinação da evolução das variáveis

Para a análise dos estados, optou-se por representar a sua evolução temporal em separado, de modo a visualizar cada um com maior detalhe:

```
% Simulação
figure(1);
clf;
plot(t,s,'-k', t,s_est,':k');
legend('S','S_{est}');
figure(2);
clf;
plot(t,id,'-r', t,id_est,':r');
legend('Id','Id_{est}');
figure(3);
clf;
plot(t,ih,'-b', t,ih_est,':b');
legend('Ih','Ih_{est}');
figure(4);
clf;
plot(t,r,'-g', t,r_est,':g');
legend('R','R_{est}');
```

Figura 25 - Código utilizado para representar a evolução das variáveis

Os resultados parecem condizer com o comportamento de um estimador. À medida que o tempo progride, o comportamento das variáveis estimadas aproxima-se bastante do comportamento real das variáveis reais, sendo que a disparidade entre os valores se torna bastante baixa.

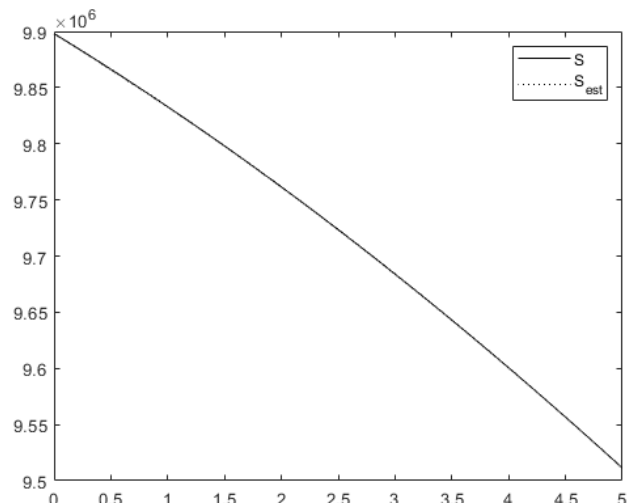


Figura 26 - Evolução de S e do seu estimador

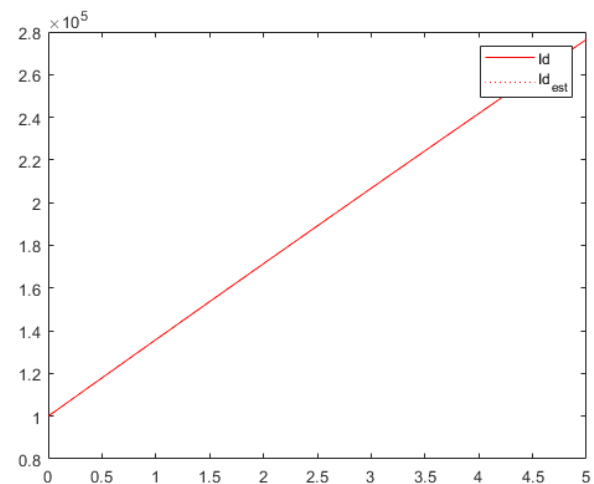


Figura 27 - Evolução de Id e do seu estimador

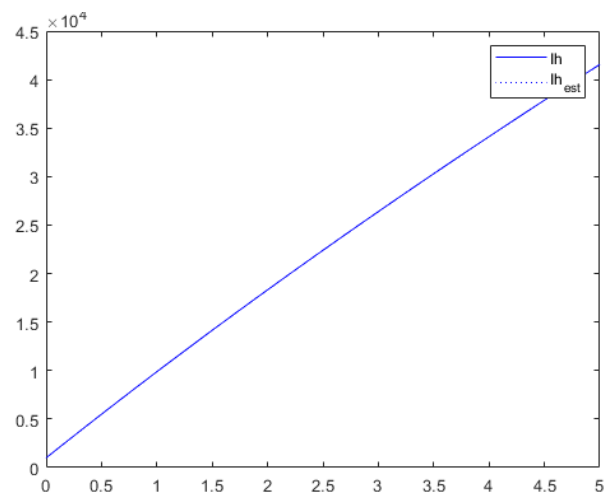


Figura 28 - Evolução de Ih e do seu estimador

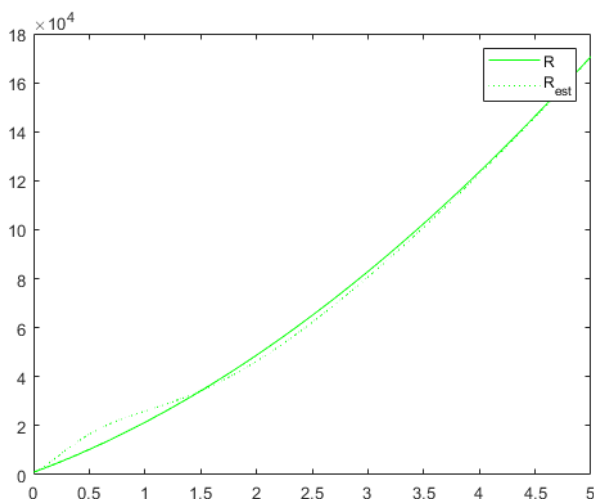


Figura 29 - Evolução de R e do seu estimador

Verifica-se que os estimadores que apresentam uma maior disparidade com os resultados que pretendíamos obter são relativos aos Suscetíveis e aos Recuperados. Apesar de a evolução ser promissora, a sua estabilização parece estar bastante longe de 1/10 do erro original propostos e dimensionados e que se deveriam verificar volvidos dois dias de simulação. Um facto evidente na escala de grandeza das variáveis em questão.

Os únicos estados cujos estimadores parecem gozar de algum sucesso são, sem qualquer dúvida, os estados relativos aos números de infetados, em que é praticamente indetetável as diferenças entre o estimador e o valor real obtido. Para demonstrar o efeito provocado na norma do erro, utilizou-se o MATLAB:

```
figure(5);
clf;
plot(t, vecnorm(transpose(x_aux-x_est)));
figure(6); clf;
plot(t, abs(x_aux(:,1)-x_est(:,1)));
figure(7); clf;
plot(t, abs(x_aux(:,2)-x_est(:,2)));
figure(8); clf;
plot(t, abs(x_aux(:,3)-x_est(:,3)));
figure(9); clf;
plot(t, abs(x_aux(:,4)-x_est(:,4)));
```

Figura 30 - Código utilizado para visualizar a evolução do erro

Este código permite-nos, também, visualizar individualmente o módulo do erro de cada estado, algo essencial para facilitar o processo de interpretação da norma do vetor de erro.

Como podemos observar, o padrão seguido pela norma do vetor do erro aproxima-se bastante do erro verificado em R que, por ser muito maior do que os outros todos, acaba por tornar impossível cumprir os objetivos inicialmente propostos para o observador de estado:

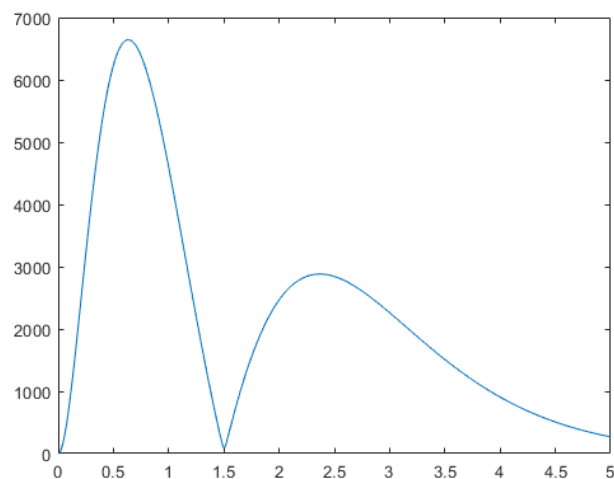


Figura 31 - Evolução da norma do erro

O erro de S é substancialmente superior ao dimensionado, apresentando duas oscilações antes de estabilizar. Em termos relativos, o erro em relação à variável que estamos a estimar é extremamente pequeno:

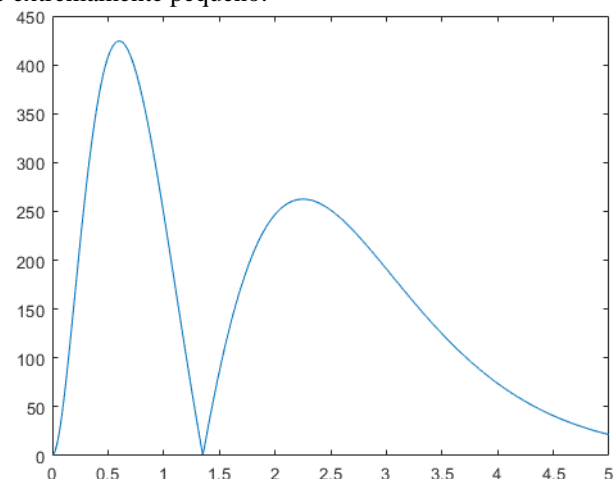


Figura 32 - Evolução do erro de S

Mais uma vez, não são cumpridos os objetivos iniciais, apesar de, à semelhança do erro em S, o erro em Id relativamente aos valores da variável são muito pequenos:

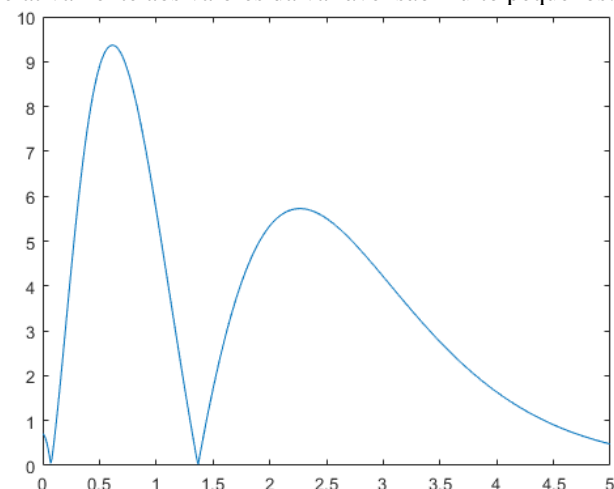


Figura 33 - Evolução do erro de Id

Sendo esta a variável diretamente acessível, o erro é praticamente inexistente:

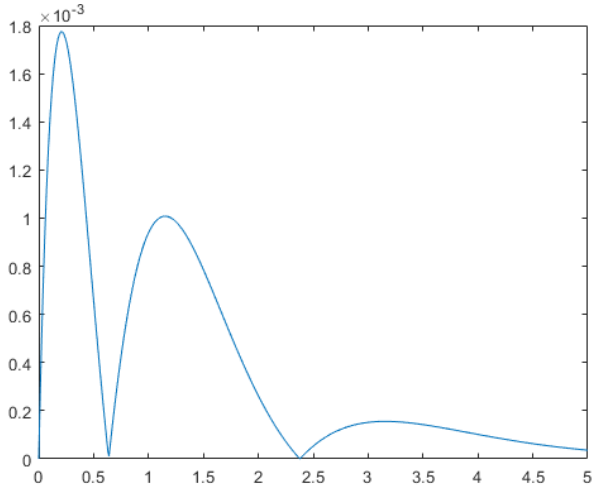


Figura 34 - Evolução do erro de I_h

O estimador de R é a causa maior da disparidade entre os erros obtidos e os pretendidos. Como podemos observar, esta curva é quase igual à curva da norma do vetor do erro, devido à magnitude dos valores do erro:

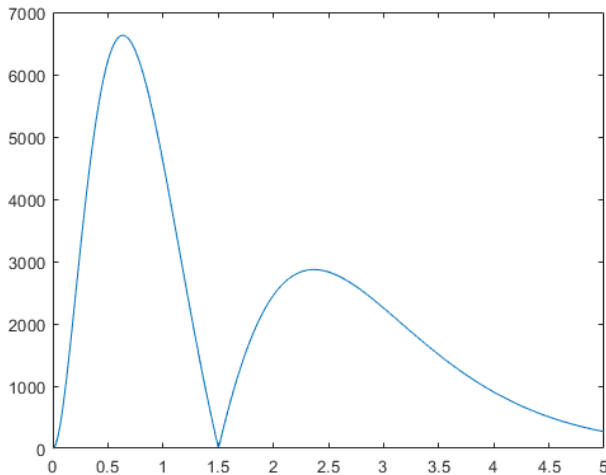


Figura 35 - Evolução do erro de R

C. Sistema com Controlador e Estimador

A figura seguinte mostra a interligação entre o sistema original, o estimador de estado e a realimentação de estado feita a partir do estimador:

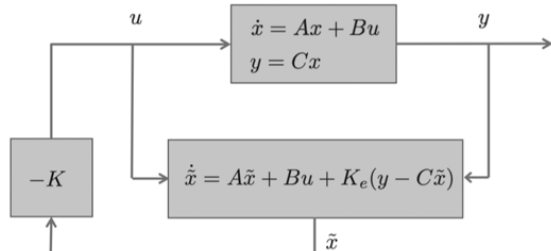


Figura 36 - Sistema com Estimador e Controlador

Para determinar a função de transferência de malha fechada vamos considerar que u passará a ser $u = -K\tilde{x} + r$, onde r será a entrada externa (notemos que se $K = 0$ - sistema

em malha aberta - temos $u = r$). Isto corresponde ao seguinte diagrama de blocos:

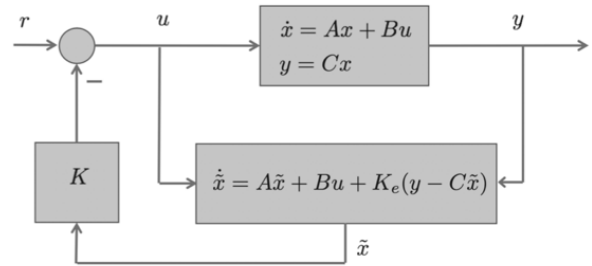


Figura 37 - Sistema apresenta uma entrada externa r

Podemos, com isto, estabelecer as seguintes relações:

$$\dot{x} = Ax + Bu; y = Cx$$

$$\dot{\tilde{x}} = A\tilde{x} + Bu + K_e(y - C\tilde{x})$$

$$u = -K\tilde{x} + r$$

Aplicando a transformada de Laplace nas relações acima temos:

$$sX(s) = AX(s) + BU(s)$$

$$Y(s) = CX(s)$$

$$sX(s) = A\tilde{X}(s) + BU(s) + K_e(Y(s) - C\tilde{X}(s))$$

$$U(s) = -K\tilde{X}(s) + R(s)$$

Das duas primeiras, vem que:

$$Y(s) = C(sI - A)^{-1}BU(s)$$

Substituindo a última equação na penúltima temos ainda

$$\tilde{X}(s) = (sI - A + BK + K_eC)^{-1}[BR(s) + K_eY(s)]$$

Com isto, obtemos

$$U(s) = -K(sI - A + BK + K_eC)^{-1}[BR(s) + K_eY(s)] + R(s) \Leftrightarrow$$

$$\Leftrightarrow U(s) = (1 - K(sI - A + BK + K_eC)^{-1}B)R(s) - K(sI - A + BK + K_eC)^{-1}K_eY(s)$$

Substituindo agora na relação entre $U(s)$ e $Y(s)$ acima, iremos obter:

$$\frac{Y(s)}{R(s)} = \frac{C(sI - A)^{-1}B(1 - K(sI - A + BK + K_eC)^{-1}B)}{1 + C(sI - A)^{-1}BK(sI - A + BK + K_eC)^{-1}K_e}$$

Com o seguinte código no MATLAB, podemos obter a expressão final:

```
syms s
e=s*eye(4);
yr = C*inv(e-A)*B*(1-K*inv(e-A+B*K+K_e*C)*B)/(1+C*inv(e-A)*B*K*inv(e-A+B*K+K_e*C)*K_e);
exp=simplify(yr);
disp(exp);
```

Figura 38 - Código utilizado para a Função de Transferência

$$\frac{Y(s)}{R(s)} = -\frac{3315s^5 + \frac{106301}{4}s^4 + \frac{240007}{3}s^3 + 107406s^2 + 54808s + 884}{s^6 + \frac{30151}{3015}s^5 + \frac{14720}{359}s^4 + \frac{9775}{111}s^3 + \frac{9258}{89}s^2 + \frac{9537}{149}s + \frac{47313}{2957}}$$

D. Resposta e Desempenho do Sistema Controlado

Era-nos pedido que analisássemos a resposta do sistema controlado (via estimador), verificando o desempenho obtido. Tendo obtido a função de transferência, poderíamos recorrer a um sinal de referência, neste caso o degrau unitário da função $step()$, para visualizar a resposta do sistema. Para além disso, com recurso à função $stepinfo()$, é possível obter os indicadores de desempenho.

```

syms s
e=s*eye(4);
yr = C*inv(e-A)*B*(1-K*inv(e-A+B*K+Ke*C)*B)/(1+C*inv(e-A)*B*K*inv(e-A+B*K+Ke*C)*Ke);
exp=simplify(yr);
disp(exp);
s = tf('s');
exp_zeros=3315*s^5+106301/4*s^4+240007/3*s^3+107406*s^2+54808*s+884;
exp_poles=s^6+30151/3015*s^5+14720/359*s^4+9775/111*s^3+9258/89*s^2+9537/149*s+47313/2957;
g=exp_zeros/exp_poles;
step(q);
ST=stepinfo(q)

```

Figura 39 - Código utilizado para obter a resposta do sistema

A resposta do sistema, devido ao sinal negativo da função de transferência, teria originalmente esta forma:

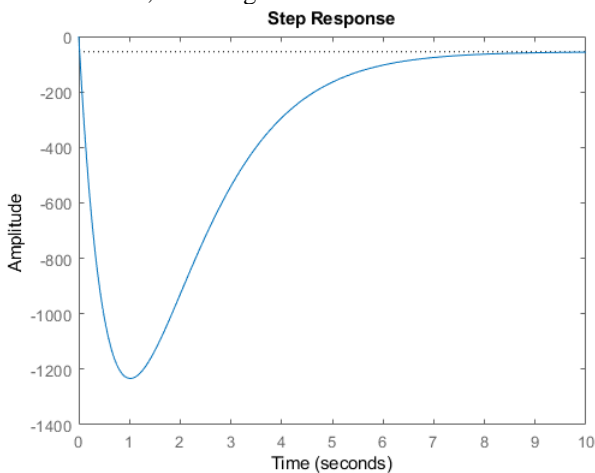


Figura 40 - Resposta do sistema ao degrau

Para facilitar a interpretação da sua análise visual, o sinal foi removido, tal como se comprova no código. Esta particularidade não afetará os indicadores de desempenho:

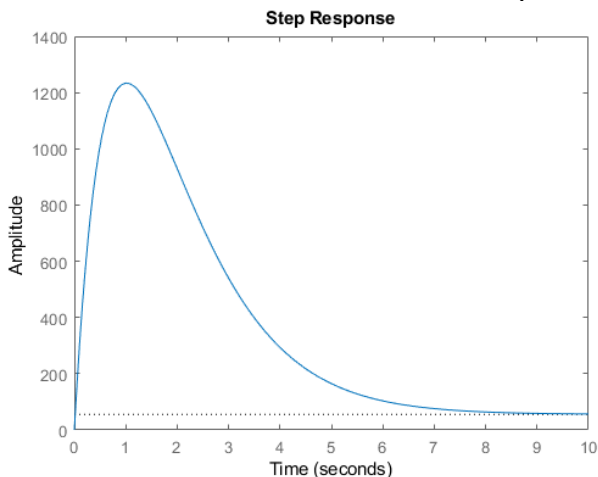


Figura 41 - Resposta do sistema simétrico ao sinal

Relativamente aos indicadores de desempenho apresentados, denota-se um tempo de assentamento relativamente pequeno, mas com um overshoot bastante elevado, tal como se pode verificar pela dimensão do pico relativamente ao valor final que adquire. O crescimento

inicial é, também ele, bastante acelerado, tal como o *rise time* comprova:

```

RiseTime: 0.0141
SettlingTime: 6.8436
SettlingMin: 56.1655
SettlingMax: 1.2335e+03
Overshoot: 2.1326e+03
Undershoot: 0
Peak: 1.2335e+03
PeakTime: 1.0076

```

Figura 42 - Indicadores de desempenho do sistema

IV. ESPAÇO DE ESTADOS EM TEMPO DISCRETO

A. Modelo de espaços de estado em tempo discreto

No que toca ao modelo equivalente em tempo discreto com $\Delta=1$, temos de considerar o seguinte modelo:

$$x_{k+1} = A_t x_k + B_t u_k, \quad y_k = C_t x_k + D_t u_k$$

Com:

$$A_t = e^{A\Delta} \quad B_t = A^{-1}(e^{A\Delta} - I)B$$

Estes passos foram feitos em Matlab. Devido a alguma dificuldade de interpretação sobre o que era pedido no guião, não ficou claro se seria pretendido fazer esta análise para um sistema com uma entrada e uma saída, ou para um sistema com duas entradas e duas saídas. Desta forma decidimos fazer para os dois casos. Iremos primeiramente expor os resultados para o primeiro caso, ou seja, uma entrada e uma saída:

$$A_t = \begin{bmatrix} 0.9954 & -0.1942 & 0.0013 & 0.0165 \\ 0.0036 & 0.996 & 0 & 0 \\ 0.0006 & 0.0270 & 0.8465 & 0 \\ 0.0004 & 0.1676 & 0.1522 & 0.9835 \end{bmatrix}$$

$$B_t = \begin{bmatrix} 23879 \\ -18743 \\ -3308 \\ -1829 \end{bmatrix}$$

$$C_1 = [0 \quad 0 \quad 1 \quad 0]$$

```

inversa=ilaplace(inv(s*eye(4)-A));
expA=subs(inversa,t,1);
A_til=expA;
integral=subs(inversa,t,1-x);
B_til=int(integral*B,x,0,1);

```

Figura 43 - Código utilizado para o cálculo do sistema discreto para uma entrada e uma saída

Agora para o segundo caso, duas entradas e duas saídas, onde temos um $B_t 2$ e um C_2 :

$$B_{t2} = \begin{bmatrix} 23900 & -7717900 \\ -18700 & -141000 \\ -3300 & -2500 \\ -1800 & 7734400 \end{bmatrix}$$

$$C_2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

```
inversa=ilaplace(inv(s*eye(4)-A));
expA=subs(inversa,t,1);
A_til=expA;
integral=subs(inversa,t,1-x);
B2_til=int(integral*B2,x,0,1);
```

Figura 44 - Código utilizado para o cálculo do sistema discreto para duas entradas e duas saídas

Como temos uma saída estática, as matrizes C_t e D_t serão, respetivamente, as matrizes C e D calculadas anteriormente.

B. Função de transferência Z

Para a função de transferência temos de efetuar o seguinte cálculo:

$$FT = C_t(zI - A_t)^{-1}B_t$$

Utilizamos o Matlab para efetuar este cálculo, tendo obtido o resultado:

$$G(z) = -\frac{(3307.5z - 3252.9)}{z^2 - 1.9785z + 0.9793}$$

Para a função obtida em 1g) utilizamos o matlab para a discretização da função. O resultado foi o seguinte:

$$G(z) = -\frac{(3307.5z - 3252.9)}{z^2 - 1.9785z + 0.9793}$$

Podemos observar que as funções de transferências são exatamente iguais. Apesar de o processo ser diferente, as expressões são as esperadas. No exercício 1, alínea g, obtivemos a função de transferência em s e passamos para a expressão no domínio discreto, em z . No exercício 3, alínea b, calculamos as matrizes de transição de estados e obtivemos a expressão em z através da fórmula mostrada em cima. Com base neste acontecimento, concluímos que o cálculo do modelo de estados em tempo discreto está 100% correto.

C. Projetar um controlador que leva o seu sistema ao seu ponto de equilíbrio

Para este último desafio, foi-nos sugerido investigar o dead-beat control.

Usando colocação de polos, *pole placement*, para um sistema controlável, podemos colocar os polos em qualquer local (no caso do dead-beat control, é desejado colocar todos os polos na origem), a partir de uma matriz de ganho (*state feedback gain matrix*), normalmente representada por K . Isto é a definição de dead-beat control.

Com polos na origem, o sistema tende a convergir para zero em n amostras, sendo n a ordem do sistema. Esta técnica de colocação de polos constitui também a mais rápida resposta na convergência, sendo esta n amostras. Mais rápido não é possível devido ao número de termos do numerador. No nosso caso, como temos um sistema de ordem 4, queremos que ele convirja em 4 amostras de tempo.

Em primeiro lugar, temos de calcular a controlabilidade. A matriz controlabilidade é a seguinte:

$$M = [B_t \quad A_t B_t \quad A_t^2 B_t \quad A_t^3 B_t]$$

Calculando o rank desta matriz, vemos que é igual a 3, não constituindo portanto um sistema controlável. No entanto, ao calcular os polos em malha fechada, observamos que temos dois polos fixos e dois polos dependentes dos valores de K .

$$\begin{cases} \exp\left(-\frac{1}{6}\right) \\ 1 \\ \text{expressão}_1 \text{ de } k \\ \text{expressão}_2 \text{ de } k \end{cases}$$

Devido à enorme expressão dada pela ferramenta MatLab, iremos apenas mostrar o código a usar.

```
syms kt1 kt2 kt3 kt4 s
dett=det(s*eye(4)-A_til+B_til*[kt1 kt2 kt3 kt4])==0;
sol_dett=solve(dett,s);
simplify(sol_dett(3,1))
simplify(sol_dett(4,1))
```

Figura 45 - Código utilizado para o cálculo dos polos do sistema

Consequentemente utilizamos apenas as duas primeiras colunas da matriz de controlabilidade, M , na nova matriz P . Depois, adicionamos 2 vetores linearmente independentes. Este método veio do livro cujo nome está na bibliografia no elemento [6].

Suppose that the system of Equation (10-1) is not completely state controllable. Then the rank of the controllability matrix is less than n , or

$$\text{rank}[B \mid AB \mid \cdots \mid A^{n-1}B] = q < n$$

This means that there are q linearly independent column vectors in the controllability matrix. Let us define such q linearly independent column vectors as f_1, f_2, \dots, f_q . Also, let us choose $n - q$ additional n -vectors $v_{q+1}, v_{q+2}, \dots, v_n$ such that

$$P = [f_1 \mid f_2 \mid \cdots \mid f_q \mid v_{q+1} \mid v_{q+2} \mid \cdots \mid v_n]$$

is of rank n . Then it can be shown that

$$\hat{A} = P^{-1}AP = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}, \quad \hat{B} = P^{-1}B = \begin{bmatrix} B_{11} \\ 0 \end{bmatrix}$$

Figura 46 - Método de colocação de polos para sistemas não controláveis

Após isto, calculamos \hat{A} e \hat{B} e selecionamos os coeficientes $A_{11}, A_{12}, A_{21}, A_{22}, B_{11}$ e B_{12} necessários para o cálculo do determinante que iremos calcular de seguida. O código é o seguinte:

```
disp('Alínea C');
M_til=[B_til A_til*B_til A_til^2*B_til A_til^3*B_til];
rank(M_til)
[V,D]=eig(A);
P=V;
new_P=[B_til A_til*B_til [0;0;1;0] [0;1;1;0]];
Achapeu=inv(new_P)*A_til*new_P;
Bchapeu=inv(new_P)*B_til;
A11=[Achapeu(1,1) Achapeu(1,2) Achapeu(2,1) Achapeu(2,2)];
A12=[Achapeu(1,3) Achapeu(1,4) Achapeu(2,3) Achapeu(2,4)];
A21=[0 0 0 0];
A22=[Achapeu(3,3) Achapeu(3,4) Achapeu(4,3) Achapeu(4,4)];
B11=[Bchapeu(1,1) Bchapeu(2,1)];
B12=[0;0];
```

Figura 47 - Código usado para os cálculos iniciais e seleção dos coeficientes da nova matriz P

Seguidamente, temos de calcular o determinante e igualar a 0 para obtermos os valores de k_1 :

Then we have

$$\begin{aligned}
 |s\mathbf{I} - \mathbf{A} + \mathbf{BK}| &= |\mathbf{P}^{-1}(s\mathbf{I} - \mathbf{A} + \mathbf{BK})\mathbf{P}| \\
 &= |s\mathbf{I} - \mathbf{P}^{-1}\mathbf{A}\mathbf{P} + \mathbf{P}^{-1}\mathbf{BK}\mathbf{P}| \\
 &= |s\mathbf{I} - \hat{\mathbf{A}} + \hat{\mathbf{B}}\hat{\mathbf{K}}| \\
 &= \left| s\mathbf{I} - \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{0} & \mathbf{A}_{22} \end{bmatrix} + \begin{bmatrix} \mathbf{B}_{11} \\ \mathbf{0} \end{bmatrix} [\mathbf{k}_1 \quad \mathbf{k}_2] \right| \\
 &= \begin{vmatrix} s\mathbf{I}_q - \mathbf{A}_{11} + \mathbf{B}_{11}\mathbf{k}_1 & -\mathbf{A}_{12} + \mathbf{B}_{11}\mathbf{k}_2 \\ \mathbf{0} & s\mathbf{I}_{n-q} - \mathbf{A}_{22} \end{vmatrix} \\
 &= |s\mathbf{I}_q - \mathbf{A}_{11} + \mathbf{B}_{11}\mathbf{k}_1| \cdot |s\mathbf{I}_{n-q} - \mathbf{A}_{22}| = 0
 \end{aligned}$$

Figura 48 - Cálculo do determinante

Estes valores de \mathbf{k}_1 serão necessários para vermos a resposta dos sistema em malha aberta (sendo a entrada $U = -Kx$).

Para os restantes valores de \mathbf{K} definimos 0 para cada um, uma vez que os polos obtidos do segundo determinante do código mostrado a seguir são exatamente os polos obtidos do sistema em malha fechada que não podemos mexer, isto é, $\exp(-1/6)$ e 1.

```

%det(sIq - A11 + B11*k1)
syms k11 k12
k1=[k11 k12];
det1=det(s*eye(2)-A11+B11*k1);
sol_det1=solve(det1==0,s);
eq1=sol_det1(1)==0;
eq2=sol_det1(2)==0;
sol_det1=solve([eq1 eq2],[k11 k12]);
k11 = sol_det1.k11;
k12 = sol_det1.k12;

Kchapeu=[k11 k12 0 0];
K=Kchapeu*inv(new_P);

%det(sIn-q - A22) - Confirmar os valores dos polos que nao podemos mexer
syms k21 k22
det2=det(s*eye(2)-A22);
sol_det2=solve(det2==0,s); %Confirmado

```

Figura 49 – Código para o cálculo do determinante

Por último, vimos a resposta do sistema em malha fechada a um degrau pela seguinte equação, representativa de uma realimentação:

$$\mathbf{A}_t - \mathbf{B}_t\mathbf{K}$$

```

sys = ss(double(A_til-B_til*K),double(B_til),C,0,1);
step(sys);

```

Figura 50 - Código para o Teste

E obtemos o seguinte resultado:

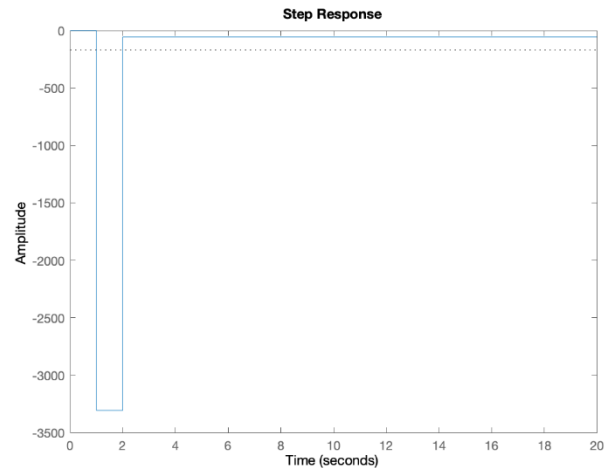


Figura 51 - Resultado final na resposta ao degrau

Como podemos ver, o sistema estabiliza após 2 amostras, sendo 2 o número de polos que colocamos, comprovando o que estava previsto. Apesar de este método constituir o método de convergência mais rápida, consideramos que provavelmente será melhor utilizar outro método que não revele um pico de amplitude tão elevado.

É importante referir que não é possível obter as 4 amostras pedidas pelo professor no guião, pelas diversas razões que mostramos ao longo desta alínea.

Terminamos assim a nossa análise.

BIBLIOGRAFIA

- [1] Ogata, Katsuhiko, *Discrete Time Control Systems*, Second Edition, University of Minnesota, 1995.
- [2] I. Kar, *Module 4: Time Response of discrete time systems – Lecture Note 2*.
- [3] No Author declared, *Linearization of Differential Equation Models*.
- [4] *SEIR and SEIRS model*, Institute for Disease Modeling (IDM), 2019.
- [5] ACM, FAF, FLP, *Projeto CDIG 2020: O vírus CODIG*, FEUP-DEEC, 2020.
- [6] Ogata, Katsuhiko, *Modern Control Engineering*, Fifth Edition, University of Minnesota, 1995.

Anexos: Código MATLAB utilizado

Exercício 1b)

```
format rational

N=10^7;
I=2e5;
R=2e6;
S=7.8e6;
p=1/60;
gama=1/6;
v=0;
Ro=2.4;

U=2.4-(N/(0.85*S));
disp(U);

Id=(85/100)*I;
disp(Id);
alfa=(Ro-U)*gama*Id/N;
disp(alfa);

%% Comprovar resultados
R_teste=((alfa+v)/p)*S;
disp(R_teste);

S_teste=(gama/alfa)*(I);
disp(S_teste);

Id_teste=(0.85*alfa*S)/gama;
disp(Id_teste);

Ih_teste=(0.15*alfa*S)/gama;
disp(Ih_teste);
```

Exercício 1e)

```
N=1e7;
I_eq=2e5;
Id_eq=0.85*I_eq;
Ih_eq=0.15*I_eq;
R_eq=2e6;
S_eq=7.8e6;
gama=1/6;
p=1/60;
v_eq=0;
Ro=2.4;

A=[-v_eq-gama*Id_eq/(0.85*S_eq) -gama/0.85 0 p; Id_eq*gama/S_eq 0 0 0; gama*Ih_eq/S_eq
gama*Ih_eq/Id_eq -gama 0; v_eq gama gama -p];

B_twain=[Id_eq*S_eq*gama/N -S_eq; -0.85*Id_eq*S_eq*gama/N 0; -0.15*Id_eq*S_eq*gama/N 0; 0
S_eq];

B_onein=[Id_eq*S_eq*gama/N; -0.85*Id_eq*S_eq*gama/N; -0.15*Id_eq*S_eq*gama/N; 0];

C_twoout=[0 0 1 0; 0 1 1 0];
```

```

C_oneout=[0 0 1 0];

%%Uma Entrada - Controlabilidade
M_one=[B_onein A*B_onein A*A*B_onein A*A*A*B_onein];
disp(rank(M_one));
if(rank(M_one)==4)
    disp('Controlável com uma entrada');
else
    disp('Não Controlável com uma entrada');
end

%%Duas Entradas - Controlabilidade
M_two=[B_twoin A*B_twoin A*A*B_twoin A*A*A*B_twoin];
disp(rank(M_two));
if(rank(M_two)==4)
    disp('Controlável com duas entradas');
else
    disp('Não Controlável com duas entradas');
end

%%Uma Saida - Observabilidade
Q_one=[C_oneout; C_oneout*A; C_oneout*A*A; C_oneout*A*A*A];
disp(rank(Q_one));
if(rank(Q_one)==4)
    disp('Observável com uma saída');
else
    disp('Não Observável com uma saída');
end

%%Duas saídas - Observabilidade
Q_two=[C_twoout; C_twoout*A; C_twoout*A*A; C_twoout*A*A*A];
disp(rank(Q_two));
if(rank(Q_two)==4)
    disp('Observável com duas saídas');
else
    disp('Não Observável com duas saídas');
end

disp('---');
%Polos
syms s;
pol_ma=[s 0 0 0; 0 s 0 0; 0 0 s 0; 0 0 0 s]- A);
eq_ma=det(pol_ma)==0;
s_ma=solve(eq_ma,s);
syms k4 k3 k2 k1
pol_mf=[s 0 0 0; 0 s 0 0; 0 0 s 0; 0 0 0 s]- A +B_onein*[k4 k3 k2 k1]);
eq_mf=det(pol_mf)==0;
s_mf=solve(eq_mf,s);
disp('Polos malha aberta:');disp(s_ma);
disp('Polos malha fechada:');disp(s_mf);
sol1=subs(s_mf(3,1),{k2,k1},{1,1});
sol2=subs(s_mf(4,1),{k2,k1},{1,1});
disp(sol1);
disp(sol2);
eq1=sol1== -2;
eq2=sol2== -2;
final=solve([eq1 eq2],[k3 k4]);
disp('k3:');disp(double(final.k3));
disp('k4:');disp(double(final.k4));

```

Exercício 1f)

```
%Constantes definidas, umas dadas e outras calculadas
format rational
N=1e7;
I_eq=2e5;
Id_eq=0.85*I_eq;
Ih_eq=0.15*I_eq;
R_eq=2e6;
S_eq=7.8e6;
gama=1/6;
p=1/60;
v_eq=0;
Ro=2.4;

%Matriz A
A=[-v_eq-gama*Id_eq/(0.85*S_eq) -gama/0.85 0 p; Id_eq*gama/S_eq 0 0 0; gama*Ih_eq/S_eq
gama*Ih_eq/Id_eq -gama 0; v_eq gama gama -p];

%Uma entrada, u=U
B=[Id_eq*S_eq*gama/N; -0.85*Id_eq*S_eq*gama/N; -0.15*Id_eq*S_eq*gama/N; 0];

%Uma saída, y=Ih
C=[0 0 1 0];

%Forma Canonica Controlavel
disp('Forma Canonica Controlavel'); disp(' ');
syms s;
pol=([s 0 0 0; 0 s 0 0; 0 0 s 0; 0 0 0 s]- A);
disp('Determinante do polinomio caracteristico de A para saber a3 a2 a1'); disp(det(pol));
a1=439/2340;
a2=1/234;
a3=11/84240;
W=[a3 a2 a1 1 ; a2 a1 1 0 ; a1 1 0 0; 1 0 0 0];
M=[B A*B A*A*B A*A*A*B];
T=M*W;
disp('Matriz de controlabilidade, M:'); disp(M);
disp('Matriz de Mudança de Base, T:'); disp(T);
%disp('Nova A:'); disp(inv(T)*A*T);
%disp('Nova B:'); disp(inv(T)*B);
disp('-----');

%Forma Canonica Observavel
disp('Forma Canonica Observavel'); disp(' ');
Q=[C; C*A; C*A*A; C*A*A*A];
U=inv(W*Q);
disp('Matriz de observabilidade, Q:'); disp(Q);
disp('Matriz de Mudança de Base, U:'); disp(U);
disp('Nova A:'); disp(inv(U)*A*U);
disp('Nova B:'); disp(inv(U)*B);
disp('Nova C:'); disp(C*U);
disp('-----');

%Forma Canonica Diagonal
disp('Forma Canonica Diagonal'); disp(' ');
[V,D]=eig(A);
P=V;
disp('Matriz de Mudança de Base, P:'); disp(P);
disp('Nova A:'); disp(inv(P)*A*P);
disp('Nova B:'); disp(inv(P)*B);
disp('Nova C:'); disp(C*P);
disp('-----');
```

Exercício 1g)

%Constantes definidas, umas dadas e outras calculadas

```
N=1e7;  
I_eq=2e5;  
Id_eq=0.85*I_eq;  
Ih_eq=0.15*I_eq;  
R_eq=2e6;  
S_eq=7.8e6;  
gama=1/6;  
p=1/60;  
v_eq=0;  
Ro=2.4;
```

%Matriz A

```
A=[-v_eq-gama*Id_eq/(0.85*S_eq) -gama/0.85 0 p; Id_eq*gama/S_eq 0 0 0; gama*Ih_eq/S_eq  
gama*Ih_eq/Id_eq -gama 0; v_eq gama gama -p];
```

%Uma entrada, u=U

```
B=[Id_eq*S_eq*gama/N; -0.85*Id_eq*S_eq*gama/N; -0.15*Id_eq*S_eq*gama/N; 0];
```

%Uma saída, y=Ih

```
C=[0 0 1 0];  
D=[0];
```

disp('Função de transferência, G:');
syms s

```
G=C*inv([s 0 0 0; 0 s 0 0; 0 0 s 0; 0 0 0 s]- A)*B+D;  
G=simplify(G);  
disp(G);
```

Exercício 1h)

format rational

```
N=1e7;  
I_eq=2e5;  
Id_eq=0.85*I_eq;  
Ih_eq=0.15*I_eq;  
R_eq=2e6;  
S_eq=7.8e6;  
gama=1/6;  
p=1/60;  
v_eq=0;  
Ro=2.4;  
U_eq=Ro-N/(0.85*S_eq);
```

%Matriz A

```
A=[-v_eq-gama*Id_eq/(0.85*S_eq) -gama/0.85 0 p; Id_eq*gama/S_eq 0 0 0; gama*Ih_eq/S_eq  
gama*Ih_eq/Id_eq -gama 0; v_eq gama gama -p];
```

%Uma entrada, u=U

```
B=[Id_eq*S_eq*gama/N; -0.85*Id_eq*S_eq*gama/N; -0.15*Id_eq*S_eq*gama/N; 0];
```

%Uma saída, y=Ih

```
C=[0 0 1 0];
```

```
D=0;
```

%%Condições iniciais não nulas


```

T = 0:0.01:600; % simulation time = 600 seconds
U = zeros(size(T)); % no input
X0 = [1e7-1.2*Id_eq 1.2*Id_eq 0 0]; % initial conditions of the three states
sys = ss(A,B,C,D); % construct a system model
figure(3);clf;
lsim(sys, U, T, X0) % simulate and plot the response (the output)
title('Response to Non-Zero Initial Conditions')

```

```

%%Condições iniciais nulas
T = 0:0.01:600; % simulation time = 600 seconds
U = -0.1*U_eq+0*T; % input as a function of time
sys = ss(A,B,C,D); % construct a system model
figure(2);clf;
[Y, Tsim, X] = lsim(sys,U,T); % simulate
plot(Tsim,Y) % plot the output vs. time
title('Response to an Input with Zero Initial Conditions')

```

Exercício 2a)

```

N=1e7;
I_eq=2e5;
Id_eq=0.85*I_eq;
Ih_eq=0.15*I_eq;
R_eq=2e6;
S_eq=7.8e6;
gama=1/6;
p=1/60;
v_eq=0;
Ro=2.4;

```

```

A=[-v_eq-gama*Id_eq/(0.85*S_eq) -gama/0.85 0 p; Id_eq*gama/S_eq 0 0 0; gama*Ih_eq/S_eq
gama*Ih_eq/Id_eq -gama 0; v_eq gama gama -p];

```

```

B_twin=[Id_eq*S_eq*gama/N -S_eq; -0.85*Id_eq*S_eq*gama/N 0; -0.15*Id_eq*S_eq*gama/N 0; 0
S_eq];

```

```

B_onein=[Id_eq*S_eq*gama/N; -0.85*Id_eq*S_eq*gama/N; -0.15*Id_eq*S_eq*gama/N; 0];

```

```

C_twoout=[0 0 1 0; 0 1 1 0];

```

```

C_oneout=[0 0 1 0];

```

%%Uma Entrada - Controlabilidade

```

M_one=[B_onein A*B_onein A*A*B_onein A*A*A*B_onein];
disp(rank(M_one));
if(rank(M_one)==4)
    disp('Controlável com uma entrada');
else
    disp('Não Controlável com uma entrada');
end

```

%%Duas Entradas - Controlabilidade

```

M_two=[B_twin A*B_twin A*A*B_twin A*A*A*B_twin];
disp(rank(M_two));
if(rank(M_two)==4)
    disp('Controlável com duas entradas');
else
    disp('Não Controlável com duas entradas');
end

```

```

%% Uma Saída - Observabilidade
Q_one=[C_oneout; C_oneout*A; C_oneout*A*A; C_oneout*A*A*A];
disp(rank(Q_one));
if(rank(Q_one)==4)
    disp('Observável com uma saída');
else
    disp('Não Observável com uma saída');
end

```

```

%% Duas saídas - Observabilidade
Q_two=[C_twoout; C_twoout*A; C_twoout*A*A; C_twoout*A*A*A];
disp(rank(Q_two));
if(rank(Q_two)==4)
    disp('Observável com duas saídas');
else
    disp('Não Observável com duas saídas');
end

```

```

disp('---');
% Polos
syms s;
pol_ma=(s 0 0 0; 0 s 0 0; 0 0 s 0; 0 0 0 s)- A);
eq_ma=det(pol_ma)==0;
s_ma=solve(eq_ma,s);
syms k4 k3 k2 k1
pol_mf=(s 0 0 0; 0 s 0 0; 0 0 s 0; 0 0 0 s)- A +B_onein*[k4 k3 k2 k1]);
eq_mf=det(pol_mf)==0;
s_mf=solve(eq_mf,s);
disp('Polos malha aberta:');disp(s_ma);
disp('Polos malha fechada:');disp(s_mf);
sol1=subs(s_mf(3,1),{k2,k1},{1,1});
sol2=subs(s_mf(4,1),{k2,k1},{1,1});
disp(sol1);
disp(sol2);
eq1=sol1==-1;
eq2=sol2==-1;
final=solve([eq1 eq2],[k3 k4]);
disp('k3:');disp(double(final.k3));
disp('k4:');disp(double(final.k4));

```

Exercício 2b)

```

N=1e7;
I_eq=2e5;
Id_eq=0.85*I_eq;
Ih_eq=0.15*I_eq;
R_eq=2e6;
S_eq=7.8e6;
gama=1/6;
p=1/60;
v_eq=0;
Ro=2.4;
U_eq=Ro-N/(0.85*S_eq);

A=[-v_eq-gama*Id_eq/(0.85*S_eq) -gama/0.85 0 p; Id_eq*gama/S_eq 0 0 0; gama*Ih_eq/S_eq
gama*Ih_eq/Id_eq -gama 0; v_eq gama gama -p];

B_twoin=[Id_eq*S_eq*gama/N -S_eq; -0.85*Id_eq*S_eq*gama/N 0; -0.15*Id_eq*S_eq*gama/N 0; 0
S_eq];

```

```
B_onein=[Id_eq*S_eq*gama/N; -0.85*Id_eq*S_eq*gama/N; -0.15*Id_eq*S_eq*gama/N; 0];
```

```
C_twoout=[0 0 1 0; 0 1 1 0];
```

```
C_oneout=[0 0 1 0];
```

```
L=acker(A',C_oneout',[-2 -2 -2 -2]);
```

```
syms ke4 ke3 ke2 ke1 omega
```

```
ke = [ke4;ke3;ke2;ke1];
```

```
pe = det(omega*eye(4)-(A-ke*C_oneout));
```

```
disp(pe);
```

```
eq0=ke1/561600 + ke2/561600 + ke3/561600 + ke4/561600==16;
```

```
eq1=11/84240 + ke1/93600 + 11*ke2/14040 + ke3/2040 + 11*ke4/93600==32;
```

```
eq2=49*ke2/2340 + ke3/34 + ke4/1560 + 1/234==24;
```

```
eq3=ke2+439/2340==8;
```

```
eq_ke=solve([eq0 eq1 eq2 eq3],[ke1 ke2 ke3 ke4]);
```

```
disp('ke4:');disp(double(eq_ke.ke4));
```

```
disp('ke3:');disp(double(eq_ke.ke3));
```

```
disp('ke2:');disp(double(eq_ke.ke2));
```

```
disp('ke1:');disp(double(eq_ke.ke1));
```

```
disp(L);
```

```
ke=[double(eq_ke.ke4); double(eq_ke.ke3); double(eq_ke.ke2); double(eq_ke.ke1)];
```

```
B=B_onein;
```

```
C=C_oneout;
```

```
D=0;
```

```
e0 = 0.7;
```

```
vecnorm([e0;e0;0;e0])
```

```
t = 0:0.01:5;
```

```
u = zeros(size(t));
```

```
xo = [10000000-101000-1000 ; 100000 ; 1000 ; 1000];
```

```
xe = xo-[e0;e0;0;e0]; %  $\tilde{x} = x - e$ 
```

```
Ak = double([A , zeros(size(A)); ke*C , A-ke*C]); % Sem Controle
```

```
Bk = [B ; B];
```

```
Ck = [C , zeros(size(C))];
```

```
Dk = D;
```

```
sistema_2b = ss(Ak,Bk,Ck,Dk);
```

```
[y,t,x] = lsim(sistema_2b,u,t,[xo ; xe]);
```

```
% Distribuir variáveis
```

```
n=4;
```

```
x_aux = x(:,1:n);
```

```
s = x_aux(:,1); id = x_aux(:,2); ih = x_aux(:,3); r = x_aux(:,4);
```

```
x_est = x(:,n+1:end);
```

```
s_est = x_est(:,1); id_est = x_est(:,2); ih_est = x_est(:,3); r_est = x_est(:,4);
```

```
% Simulação
```

```
figure(1);
```

```

clf;
plot(t,s,'-k', t,s_est,'k');
legend('S','S_{est}');
figure(2);
clf;
plot(t,id,'-r', t,id_est,'r');
legend('Id','Id_{est}');
figure(3);
clf;
plot(t,ih,'-b', t,ih_est,'b');
legend('Ih','Ih_{est}');
figure(4);
clf;
plot(t,r,'-g', t,r_est,'g');
legend('R','R_{est}');
figure(5);
clf;
plot(t,vecnorm(transpose(x_aux-x_est)));
figure(6); clf;
plot(t,abs(x_aux(:,1)-x_est(:,1)));
figure(7); clf;
plot(t,abs(x_aux(:,2)-x_est(:,2)));
figure(8); clf;
plot(t,abs(x_aux(:,3)-x_est(:,3)));
figure(9); clf;
plot(t,abs(x_aux(:,4)-x_est(:,4)));

```

Exercício 2c)

```

N=1e7;
I_eq=2e5;
Id_eq=0.85*I_eq;
Ih_eq=0.15*I_eq;
R_eq=2e6;
S_eq=7.8e6;
gama=1/6;
p=1/60;
v_eq=0;
Ro=2.4;

A=[-v_eq-gama*Id_eq/(0.85*S_eq) -gama/0.85 0 0; Id_eq*gama/S_eq 0 0 0; gama*Ih_eq/S_eq
gama*Ih_eq/Id_eq -gama 0; v_eq gama gama -p];

B=[Id_eq*S_eq*gama/N; -0.85*Id_eq*S_eq*gama/N; -0.15*Id_eq*S_eq*gama/N; 0];

C=[0 0 1 0];

K=[3813/3812 4921/4920 1 1];

Ke=[-668087;92227/6;4539/581;9638308];

syms s
e=s*eye(4);
yr = C*inv(e-A)*B*(1-K*inv(e-A+B*K+Ke*C)*B)/(1+C*inv(e-A)*B*K*inv(e-A+B*K+Ke*C)*Ke);
exp=simplify(yr);
disp(exp);
s = tf('s');
exp_zeros=3315*s^5+106301/4*s^4+240007/3*s^3+107406*s^2+54808*s+884
exp_poles=s^6+30151/3015*s^5+14720/359*s^4+9775/111*s^3+9258/89*s^2+9537/149*s+47313/2957
;
g=exp_zeros/exp_poles;

```

```
step(g);
ST=stepinfo(g)
```

Exercício 3

%Constantes definidas, umas dadas e outras calculadas

```
format
N=1e7;
I_eq=2e5;
Id_eq=0.85*I_eq;
Ih_eq=0.15*I_eq;
R_eq=2e6;
S_eq=7.8e6;
gama=1/6;
p=1/60;
v_eq=0;
Ro=2.4;
```

%Matriz A

```
A=[-v_eq-gama*Id_eq/(0.85*S_eq) -gama/0.85 0 p; Id_eq*gama/S_eq 0 0 0; gama*Ih_eq/S_eq
gama*Ih_eq/Id_eq -gama 0; v_eq gama gama -p];
```

%Uma entrada, u=U

```
B=[Id_eq*S_eq*gama/N; -0.85*Id_eq*S_eq*gama/N; -0.15*Id_eq*S_eq*gama/N; 0];
```

%Duas saidas

```
B2=[Id_eq*S_eq*gama/N -S_eq; -0.85*Id_eq*S_eq*gama/N 0; -0.15*Id_eq*S_eq*gama/N 0; 0 S_eq];
```

%Uma saida, y=Ih

```
C=[0 0 1 0];
```

%Duas saidas

```
C2=[0 0 1 0; 0 1 1 0];
```

```
D=[0;0;0;0];
```

```
syms s t x z;
disp('Alinea A');
```

```
disp('Uma entrada e uma saida');
inversa=ilaplace(inv(s*eye(4)-A));
expA=subs(inversa,t,1);
A_til=expA;
integral=subs(inversa,t,1-x);
B_til=int(integral*B,x,0,1);
disp('A_til'); disp(double(A_til));
disp('B_til'); disp(double(B_til));
disp('C_til'); disp(double(C));
```

```
disp('Duas entradas e duas saidas');
inversa=ilaplace(inv(s*eye(4)-A));
expA=subs(inversa,t,1);
A_til=expA;
integral=subs(inversa,t,1-x);
B2_til=int(integral*B2,x,0,1);
disp('A_til'); disp(double(A_til));
disp('B2_til'); disp(double(B2_til));
disp('C2_til'); disp(double(C2));
```

```
disp('-----');
```

```
disp('Alinea B');
expZ=C*inv(z*eye(4)-A_til)*B_til;
```

```

expZ=simplify(expZ);
expressao=-((3307.5*z-3252.9)/(z^2-1.9785*z+0.9793));
disp(expressao);

disp('-----');

disp('Alinea C');
M_til=[B_til A_til*B_til A_til*A_til*B_til A_til*A_til*A_til*B_til];
[V,D]=eig(A);
P=V;
new_P=[B_til A_til*B_til [0;0;1;0] [0;1;1;0]];
Achapeu=inv(new_P)*A_til*new_P;
Bchapeu=inv(new_P)*B_til;
A11=[Achapeu(1,1) Achapeu(1,2);Achapeu(2,1) Achapeu(2,2)];
A12=[Achapeu(1,3) Achapeu(1,4);Achapeu(2,3) Achapeu(2,4)];
A21=[0 0;0 0];
A22=[Achapeu(3,3) Achapeu(3,4);Achapeu(4,3) Achapeu(4,4)];
B11=[Bchapeu(1,1);Bchapeu(2,1)];
B12=[0;0];

%det(sIq - A11 + B11k1)
syms k11 k12
k1=[k11 k12];
det1=det(s*eye(2)-A11+B11*k1);
sol_det1=solve(det1==0,s);
eq1=sol_det1(1)==0;
eq2=sol_det1(2)==0;
sol_det1=solve([eq1 eq2],[k11 k12]);
k11 = sol_det1.k11;
k12 = sol_det1.k12;

Kchapeu=[k11 k12 0 0];
K=Kchapeu*inv(new_P);

%det(sIn-q - A22) - Confirmar os valores dos polos que nao podemos mexer
syms k21 k22
det2=det(s*eye(2)-A22);
sol_det2=solve(det2==0,s); %Confirmado

sys = ss(double(A_til-B_til*K),double(B_til),C,0,1);
step(sys);

```