

DESENVOLVIMENTO DE OBJETOS EDUCACIONAIS INTERATIVOS EM SCILAB PARA O ENSINO DE CONTROLE E SERVOMECANISMOS

Cássio Rodrigues Mello

**DESENVOLVIMENTO DE OBJETOS
EDUCACIONAIS INTERATIVOS EM SCILAB
PARA O ENSINO DE CONTROLE E
SERVOMECANISMOS**

Aluno: Cássio Rodrigues Mello

Orientadores: Ana Maria Beltran Pavani

William de Souza Barbosa



Agradecimentos

Agradeço aos meus pais por todo o apoio e confiança ao longo de toda a minha formação.

Aos meus orientadores Ana Pavani e William Barbosa e a toda equipe do LAMBDA envolvida no projeto de objetos educacionais utilizando o Scilab.

E finalmente, aos amigos de graduação com os quais compartilhei vários momentos na PUC.

Resumo

Objetos educacionais são elementos fundamentais no desenvolvimento do ensino a distância. Esses objetos podem acessar os recursos tecnológicos de maneira mais interativa para estimular os alunos e melhor desenvolver as competências curriculares.

A criação e a distribuição desses objetos educacionais são facilitadas pelo uso de software gratuito e de código aberto, como o Scilab, que possui boa parte dos recursos do Matlab, como por exemplo, um equivalente ao Simulink chamado Xcos.

Este trabalho consiste no desenvolvimento de dois objetos educacionais sobre modelagem, simulação e controle de um pêndulo invertido e de um motor DC utilizando realimentação de estados, controle PID e recursos gráficos disponíveis no Scilab.

Palavras-chave: Objetos educacionais; Scilab; Pêndulo invertido; Motor DC



Development of Interactive Learning Objects using Scilab for Teaching Servomechanisms

Abstract

Learning objects are fundamental elements for the development of the distance learning. These objects can access technological resources in a more interactive way to better engage students and improve the achievement of curricular skills.

The creation and distribution of these learning objects are eased by using a free and open source software, such as Scilab, which has most of Matlab's features, for example, an equivalent to Simulink called Xcos.

This work consists on the development of two learning objects about modeling, simulating and controlling an inverted pendulum and a DC motor using full-state feedback, a PID controller and some graphical resources available on Scilab.

Keywords: Learning Objects; Scilab; Inverted pendulum; DC motor

Agradecimentos	i
Resumo	ii
Abstract.....	iii
Sumário	iv
1. Introdução.....	1
2. O Projeto.....	2
a. Objetos educacionais.....	2
b. Scilab.....	2
3. Conteúdo dos Objetos Educacionais.....	4
a. Motivação.....	4
b. Pêndulo invertido	4
c. Motor DC.....	5
d. Formato de apresentação.....	5
4. Referências Bibliográficas.....	6
Anexo 1 – Roteiro do pêndulo invertido – Parte 1.....	7
Anexo 2 – Conteúdo da Lupinha 1	9
Anexo 3 – Código Scilab 1	10
Anexo 4 – Código Scilab 2	11
Anexo 5 – Código Scilab 3	12
Anexo 6 - Roteiro do pêndulo invertido – Parte 2	13
Anexo 7 – Conteúdo da lupinha 2.....	15
Anexo 8 – Código Scilab 4.....	16
Anexo 9 – Conteúdo da Lupinha 3	17
Anexo 10 – Código Scilab 5	18
Anexo 11 - Roteiro do pêndulo invertido – Parte 3.....	20
Anexo 12 – Código Scilab 6	22
Anexo 13 – Conteúdo da lupinha 4	23
Anexo 14 – Código Scilab 7	24
Anexo 15 – Código Scilab 8	27
Anexo 16 – Roteiro do pêndulo invertido – Parte 4.....	28
Anexo 17 – Código Scilab 9	30
Anexo 18 – Código Scilab 10.....	31
Anexo 19 – Código Scilab 11	34
Anexo 20 – Código Scilab 12.....	36
Anexo 21 – Código Scilab 13.....	39
Anexo 22 – Roteiro do Motor DC – Parte 1	42



Anexo 23 – Conteúdo da lupinha 5	44
Anexo 24 – Código Scilab 14	45
Anexo 25 – Código Scilab 15	47
Anexo 26 – Roteiro do Motor DC – Parte 2	48
Anexo 27 – Código Scilab 16	51
Anexo 28 – Conteúdo da lupinha 6	52
Anexo 29 – Código Scilab 17	53
Anexo 30 – Código Scilab 18	54
Anexo 31 – Roteiro do Motor DC – Parte 3	55
Anexo 32 – Conteúdo da lupinha 7	56
Anexo 33 – Código Scilab 19	57
Anexo 34 – Roteiro do Motor DC – Parte 4	58
Anexo 35 – Código Scilab 20	60
Anexo 36 – Código Scilab 21	61
Anexo 37 – Código Scilab 22	63
Anexo 38 – Código Scilab 23	64
Anexo 39 – Código Scilab 24	66



1. Introdução

A ideia deste trabalho de fim de curso surgiu a partir de um projeto que já estava sendo desenvolvido no LAMBDA (Laboratório de Automação de Museus, Bibliotecas Digitais e Arquivos – Vice Reitoria Acadêmica - PUC-Rio) sob a coordenação da professora Ana Pavani, que consistia na utilização do Scilab como ferramenta de cálculo e simulação de forma remota para possibilitar a criação de objetos educacionais interativos.

A equipe do LAMBDA trabalhou na instalação e configuração dos Scilab em servidores de teste, no desenvolvimento de conteúdo e dos códigos de programação. Posteriormente, o Scilab foi instalado nos servidores no RDC (Rio Datacentro – PUC-Rio) e os objetos publicados através do Sistema Maxwell, um repositório da produção científica da PUC-Rio.

2. O Projeto

a. Objetos educacionais

O desenvolvimento de recursos computacionais e da internet trouxeram a possibilidade de se desenvolver novas ferramentas educacionais que pudessem auxiliar o professor e o aluno no processo de ensino e aprendizagem.

Neste contexto, surgiram os objetos educacionais, que foram alvo de diferentes estudos abordando suas características, e podem ser definidos como um granular e reutilizável fragmento de informação independente de mídia. [1]

Os objetos educacionais são definidos também como a aplicação da orientação ao objeto, desenvolvida na década de 70 para melhorar a organização de softwares ao mundo da aprendizagem. [2]

Para garantir as características de reuso, foi necessária a criação de repositórios, como o sistema Maxwell da PUC-Rio, auxiliando na organização e catalogação que possibilitam a busca e a recuperação de objetos de forma mais eficiente.

Buscando a independência de plataformas e a interoperabilidade entre objetos, foram desenvolvidos padrões a serem seguidos na produção de objetos educacionais, como o padrão LOM (Learning Object Metadata) do IEEE (Institute of Electrical and Electronics Engineers) [3].

b. Scilab

O Scilab é um software gratuito e de código aberto para computação numérica, semelhante ao Matlab que fornece um poderoso ambiente computacional para aplicações científicas e de engenharia[4]. Foi desenvolvido na França, em 1990, por pesquisadores do INRIA (Institut National de Recherche en Informatique et en Automatic) e do ENPC (Ecole des Ponts ParisTech).[5] Atualmente, é mantido pela Scilab Enterprises [4].

O Scilab possui uma grande coleção de bibliotecas para áreas como álgebra linear, polinômios e funções racionais, integração numérica, processamento de sinais, sistemas de controle, otimização, estatística e também recursos gráficos.

Por ser um software gratuito, alcançou certa popularidade, [6] pois a utilização de softwares comerciais como o Matlab tem se mostrado muito custosa[7], especialmente no meio acadêmico e científico onde a licença geralmente é vendida por máquina.

c. Estrutura

Os objetos educacionais interativos combinam teoria e simulação. Eles podem ser acessados, através do Sistema Maxwell por qualquer dispositivo que possua um navegador com acesso à internet. Geralmente, possuem campos para entrada dos parâmetros por parte dos usuários. Ao entrar com os dados, o site envia o código Scilab para o servidor, que processa e devolve o resultado, que pode ser na forma de texto ou imagem para ser interpretado pelo usuário. Este processo está representado no diagrama da figura 1.

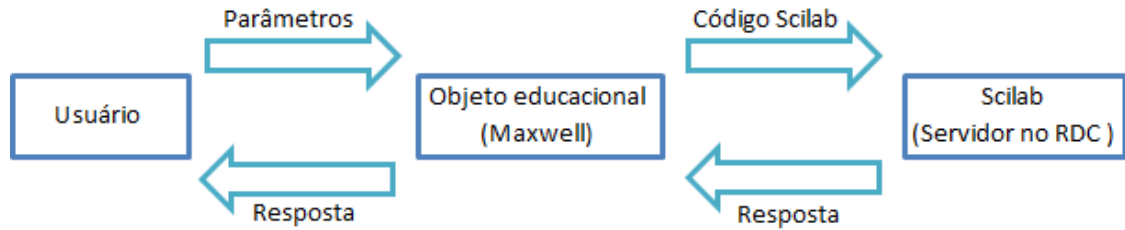


Figura 1 – Diagrama de acesso ao Scilab

3. Conteúdo dos Objetos Educacionais

a. Motivação

A motivação para a escolha da disciplina de Controle e Servomecanismos e dos assuntos do pêndulo invertido e motor DC para este projeto é devido a grande quantidade de recursos, principalmente recursos gráficos, como o Diagrama de Nyquist, Lugar das Raízes e resposta impulsional, já disponíveis na biblioteca de sistemas de controle do Scilab. A utilização desses recursos na disciplina proporciona um grande ganho de produtividade, pois possibilita a visualização de forma rápida e interativa dos efeitos da variação dos parâmetros de entrada do sistema.

Deste modo, é possível, por exemplo, ajustar os termos proporcional, integral e derivativo de um controlador PID através do método de tentativa e erro, observando o efeito das alterações na resposta impulsional do sistema. Essa observação proporciona que o aluno perceba, por conta própria, se está se aproximando ou se afastando da resposta desejada, e aprenda a tomar decisões sobre o quanto incrementar ou decrementar cada termo, passando a não mais ajustar os termos de forma aleatória e sim, de forma consciente.

Estudos como o de Wenjiang, Dong e Fan [8] mostram melhorias no ensino de computação numérica para engenharia ao utilizar o Scilab. Magyar e Žáková [9] apresentam as vantagens e possibilidades criadas ao se desenvolver um laboratório virtual para acesso via navegador Web, que não exige a instalação de nenhum software adicional à máquina do usuário.

Finalmente, Tona [10] apresenta um estudo que conclui que estudantes apresentam melhores resultados de aprendizado ao utilizar ferramentas de simulação como ferramentas de apoio no curso de controle de processos.

b. Pêndulo invertido

O pêndulo invertido é um exemplo clássico da engenharia no ensino da teoria de controle e integra conhecimentos de elétrica, eletrônica e mecânica.

O sistema consiste em um carrinho com uma haste que pode se deslocar angularmente. A haste tende a cair sob efeito da gravidade, portanto é um sistema naturalmente instável. O sistema pode ser estabilizado aplicando uma força horizontal ao carrinho, fazendo a haste ficar na vertical.

O roteiro do objeto educacional está dividido em quatro partes: na primeira é apresentado o sistema, seus elementos, as variáveis envolvidas, as entradas e saídas, as aproximações necessárias durante a modelagem, as funções de transferência, o espaço de estados e a resposta impulsional inicial, do sistema ainda instável; na segunda parte, o sistema é estabilizado através de controlador PID e o resultado pode ser verificado através da resposta impulsional e do Diagrama de Nyquist; na terceira o controlador PID é ajustado com o auxílio do Lugar das Raízes para que o sistema atinja um determinado comportamento em relação ao tempo de acomodação; e, finalmente, na quarta parte, é apresentado o método de controle por realimentação de estados, que permite que os polos do sistema sejam reposicionados de forma arbitrária pelo usuário.

A abordagem de técnicas de controle no acionamento de motores a corrente contínua é um campo bastante explorado na engenharia devido a sua versatilidade nas aplicações. Os de pequeno porte são muito utilizados em brinquedos e equipamentos portáteis pelo fato de poderem ser acionados por meio de pilhas e baterias. Já os de grande porte são utilizados para tração elétrica em trens e metrô pelo fato de permitirem fácil e precisa variação de velocidade.

Os motores CC de excitação independente podem ter sua velocidade controlada através da variação da corrente de campo ou da tensão de armadura. Neste objeto educacional trataremos deste segundo caso, então consideraremos a corrente de campo sempre constante, desta forma o fluxo magnético produzido no campo também será constante. A tensão de armadura e a velocidade de rotação da máquina passam a ter então uma relação direta.

O roteiro do objeto educacional está dividido em quatro partes: na primeira é apresentado o sistema, seus elementos, as variáveis envolvidas, as entradas e saídas, o diagrama de blocos, as funções de transferência e o espaço de estados; na segunda parte, a estabilidade do sistema é analisada através da resposta ao degrau, do diagrama de Nyquist e do diagrama de Bode; na terceira, o controle PID é usado para ajustar características da resposta do sistema, e o resultado pode ser observado pela resposta ao degrau; e, finalmente, na quarta parte, é apresentado o método de controle por realimentação de estados, que permite que os polos do sistema sejam reposicionados de forma arbitrária pelo usuário.

d. Formato de apresentação

Os roteiros dos objetos educacionais apresentados em anexo a este trabalho seguem o padrão descrito a seguir.

Os objetos começam com uma breve descrição do sistema, a modelagem e algumas aproximações necessárias para obtenção das funções de transferência e do espaço de estados.

A teoria é apresentada ao longo do texto, conforme a necessidade, e de forma resumida, voltada para usuários já familiarizados com a disciplina. Para os não tão familiarizados com os assuntos, trechos adicionais da teoria podem ser acessados em determinados pontos ao longo do texto ao clicar no ícone de uma lupa. Ao lado de cada lupa existe um número para identificá-la, o conteúdo a ser acessado pela lupa 1, por exemplo, está no Anexo 2, nomeado, Conteúdo da Lupinha 1.

Foram inseridos ao longo do texto alguns campos de formulário para entrada dos parâmetros necessários à simulação. Junto aos campos estão indicadas as variáveis a quais se referem e a unidade de medida a ser utilizada. Os campos já vêm preenchidos com valores iniciais de exemplo para dar ao usuário uma ideia da ordem de grandeza dos valores normalmente associados àquela variável.

Junto aos campos de formulário foi inserida uma imagem de um botão de enviar com o número dos códigos Scilab que devem ser executados ao enviar o formulário. Na parte de anexos podem ser encontrados todos os códigos utilizados, identificados com o respectivo número.

O retângulo azul tracejado que aparece ao redor de algumas imagens indica que esta é uma imagem gerada pelo Scilab a partir dos dados enviados pelo usuário. Próximo de cada retângulo existe uma indicação sobre qual código gera a imagem.

Nos códigos em Scilab em anexo, as variáveis pintadas de cor verde são as que devem ser substituídas pelos valores obtidos a partir do formulário.

4. Referências Bibliográficas

- [1] MUZIO, J.; HEINS, T.; MUNDELL, R. Experiences with Reusable eLearning Objects: From Theory to Practice. Victoria, Canadá. 2001.
- [2] SHEPHERD, C. Objects of interest. 2000.
- [3] IEEE Learning Technology Standards Committee (LTSC). Draft Standard for Learning Object Metadata (IEEE 1484.12.1-2002). Julho de 2002. Disponível em: http://129.115.100.158/txlor/docs/IEEE_LOM_1484_12_1_v1_Final_Draft.pdf Acesso em: 17 jul. 15
- [4] Scilab Enterprises, About Scilab. Disponível em: <https://www.scilab.org/scilab/about> Acesso em: 17 jul. 15
- [5] C. Gomez. Engineering and scientific computing with Scilab. Birkhauser, 1999.
- [6] Feng Xia, Longhua Ma e Zhe Peng. "Programming scilab in ARM linux". Em: SIGSOFT Softw. Eng. Notes 33 (5 2008), 10:1-10:5. issn: 0163-5948.
- [7] J.W.R. Cano e M.F.R. Ospina. "Scilab as a Tool to Increase Learning in Courses of Communication Systems". Em: Electronics, Robotics and Automotive Mechanics Conference, 2008. CERMA'08. IEEE. 2008, pp. 201-205.
- [8] L.Wenjiang, N. Dong e T. Fan. "Application of Scilab in teaching of engineering numerical computations". Em: Open-source Software for Scientific Computation (OSSC), 2009 IEEE International Workshop on. IEEE. 2009, pp. 8890.
- [9] Z. Magyar e K. Žáková. "Using SciLab for building of virtual lab". Em: Information Technology Based Higher Education and Training (ITHET), 2010 9th International Conference on. IEEE. 2010, pp. 280283.
- [10] P. Tona. "Teaching process control with scilab and scicos". Em: American Control Conference, 2006. IEEE. 2006, 6pp.

Anexo 1 – Roteiro do pêndulo invertido – Parte 1

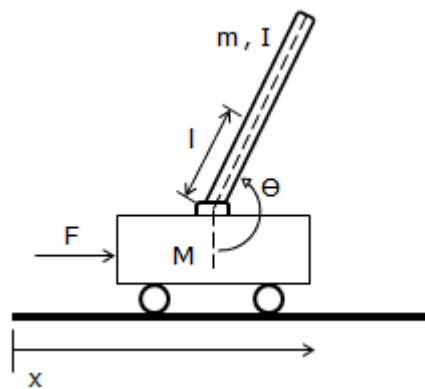
Modelo por variáveis de estados e função de transferência

O pêndulo invertido é um exemplo clássico da engenharia no ensino da teoria de controle e integra conhecimentos de elétrica, eletrônica e mecânica.

O sistema consiste em um carrinho com uma haste que pode se deslocar angularmente. A haste tende a cair sob efeito da gravidade, portanto é um sistema naturalmente instável. O sistema pode ser estabilizado aplicando uma força horizontal ao carrinho, fazendo a haste ficar na vertical.

O sistema tem como entrada a força F , que move o carro horizontalmente e, como saídas, a posição angular do pêndulo θ e a posição horizontal do carro x .

A figura a seguir apresenta o carro com o pêndulo e os seus componentes.



Onde:

$M \rightarrow$ Massa do carro

$m \rightarrow$ Massa do pêndulo

$b \rightarrow$ Coeficiente de fricção do carro

$l \rightarrow$ Distância ao centro de massa do pêndulo

$I \rightarrow$ Momento de inércia do pêndulo

$F \rightarrow$ Força aplicada ao carro

$\theta \rightarrow$ Posição angular do pêndulo

$x \rightarrow$ Posição do carro

Para detalhes sobre a modelagem do sistema, clique na lupinha.

Função de transferência $G(s)$ de Posição do pêndulo-Força F :

$$G(s) = \frac{\Phi(s)}{F(s)} = \frac{\frac{ml}{q}s}{s^3 + \frac{b(I+ml^2)}{q}s^2 - \frac{(M+m)mgl}{q}s - \frac{bmgl}{q}} \left[\frac{\text{rad}}{\text{N}} \right]$$

Onde: $q = (M+m)(I+ml^2) - (ml)^2$

Função de transferência $H(s)$ de posição do carro-Força F :

$$H(s) = \frac{X(s)}{F(s)} = \frac{\frac{(I+ml^2)s^2 - gml}{q}}{s^4 + \frac{b(I+ml^2)}{q}s^3 - \frac{(M+m)mgl}{q}s^2 - \frac{bmgl}{q}s} \left[\frac{\text{m}}{\text{N}} \right]$$

O Sistema pode ser representado também através do espaço de estados:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-(I+ml^2)b}{I(M+m)+Mml^2} & \frac{m^2gl^2}{I(M+m)+Mml^2} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-mlb}{I(M+m)+Mml^2} & \frac{mgl(M+m)}{I(M+m)+Mml^2} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{I+ml^2}{I(M+m)+Mml^2} \\ 0 \\ \frac{ml}{I(M+m)+Mml^2} \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u$$

Então, dados os parâmetros do sistema:

$$\begin{aligned} M &= 0.5 \text{ Kg} & I &= 0.006 \text{ Kg.m}^2 \\ m &= 0.2 \text{ Kg} & g &= 9.8 \text{ m/s}^2 \\ b &= 0.1 \text{ N/m/s} & l &= 0.3 \text{ m} \end{aligned}$$

Enviar

123

As funções de transferência tornam-se:

$$\begin{aligned} G(s) &= \frac{4.54545s}{-4.45455 - 31.1818s + 0.18182s^2 + s^3} \\ H(s) &= \frac{-44.5455 + 1.81818s^2}{-4.45455s - 31.1818s^2 + 0.18182s^3 + s^4} \end{aligned}$$

Código Scilab 1

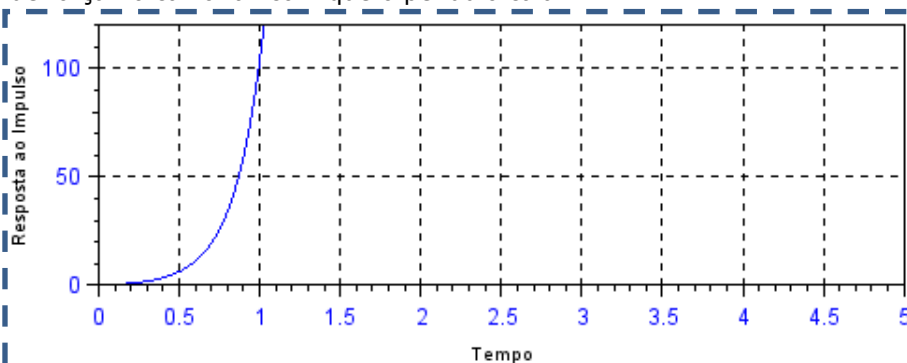
O espaço de estado torna-se:

$$\begin{cases} \dot{X}(t) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & -0.182 & 2.673 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -0.455 & 31.18 & 0 \end{pmatrix} X(t) + \begin{pmatrix} 0 \\ 1.818 \\ 0 \\ 4.545 \end{pmatrix} U(t) \\ Y(t) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} X(t) \end{cases}$$

Código Scilab 2

Onde: $\dot{X}(t) = \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix}$ e $X(t) = \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix}$

Como dito anteriormente, este sistema sem qualquer mecanismo de controle é instável. Isto pode ser verificado através da resposta impulsional, ou seja, a partir da posição inicial, uma pequena aplicação de força no carro faz com que o pêndulo caia.



Código Scilab 3

Anexo 2 – Conteúdo da Lupinha 1

Após a modelagem, através do diagrama de corpo livre e do cálculo do somatório de forças, chega-se a duas equações que descrevem o comportamento do sistema. No entanto, para aplicar as técnicas de controle, essas equações precisam ainda ser linearizadas. O sistema será linearizado no ponto onde se deseja que ele fique em equilíbrio, isto é, com $\theta = \pi$, e, considerando ϕ como sendo uma pequena variação em relação à posição de equilíbrio, então podemos definir a posição do pêndulo como $\theta = \pi + \phi$ e realizar as seguintes aproximações:

$$\cos\theta \cong -1$$

$$\sin\theta \cong -\phi$$

$$\dot{\theta}^2 = \dot{\phi}^2 = 0$$

Aplicando as aproximações, chegamos, finalmente, às equações:

$$(I + ml^2)\ddot{\phi} - mgl\phi = ml\ddot{x}$$

$$(M + m)\ddot{x} + b\dot{x} - ml\ddot{\phi} = F$$

Para chegarmos às funções de transferência, é preciso aplicar a transformada de Laplace nas equações linearizadas, que se tornam:

$$(I + ml^2)\Phi(s)s^2 - mgl\Phi(s) = mlX(s)s^2$$

$$(M + m)X(s)s^2 + bX(s)s - ml\Phi(s)s^2 = F(s)$$

Combinando as equações, obtemos finalmente as funções de transferência para posição do pêndulo e do carro em relação à força F aplicada no carro.

Anexo 3 – Código Scilab 1

```

M = 0.5;
m = 0.2;
b = 0.1;
I = 0.006;
g = 9.8;
l = 0.3;

q = (M+m)*(I+m*l^2)-(m*l)^2;
s=poly(0,'s');
format(8);
G=(m*I*s/q)/(s^3 + (b*(I + m*l^2))*s^2/q - ((M + m)*m*g*l)*s/q - b*m*g*l/q);
H=((I+m*l^2)/q)*s^2 - (m*g*l/q))/(s^4 + (b*(I + m*l^2))*s^3/q - ((M + m)*m*g*l)*s^2/q - b*m*g*l*s/q);
subplot(2,1,1);
str = prettyprint(G);
str2 = 'G(s)=';
str3=[str2,str];
xstring(0, 0, str3);
t=get('hdl');
t.font_foreground=1;
t.font_size=5;
t.alignment = 'left';
t.font_angle=0;
t.text_box = [1,1];
t.text_box_mode = 'filled';
t.box='off';
t.foreground=5;
t.fill_mode='on';
t.background=7;
a=get('current_axes');
a.margins = [0.075,0.01,0.05,0.1];
subplot(2,1,2);
str = prettyprint(H);
str2 = 'H(s)=';
str3=[str2,str];
xstring(0, 0, str3);
t=get('hdl');
t.font_foreground=1;
t.font_size=5;
t.alignment = 'left';
t.font_angle=0;
t.text_box = [1,1];
t.text_box_mode = 'filled';
t.box='off';
t.foreground=5;
t.fill_mode='on';
t.background=7;
a=get('current_axes');
a.margins = [0.075,0.01,0.05,0.1];
xset('window',0); xset('wdim', 450, 200);

```

Anexo 4 – Código Scilab 2

```

M = 0.5;
m = 0.2;
b = 0.1;
I = 0.006;
g = 9.8;
l = 0.3;

p = I*(M+m)+M*m*l^2;
format(6);
A = [0      1      0      0;
      0 -(I+m*l^2)*b/p (m^2*g*l^2)/p  0;
      0      0      0      1;
      0 -(m*l*b)/p      m*g*l*(M+m)/p  0];
B = [ 0;
      (I+m*l^2)/p;
      0;
      m*l/p];
C = [1 0 0 0;
      0 0 1 0];
D = [0;
      0];
ss=syslin('c',A,B,C);
str = prettyprint(ss);
xstring(0, 0, str);
t=get('hdl');
t.font_foreground=1;
t.font_size=5;
t.alignment = 'left';
t.font_angle=0;
t.text_box=[1,1];
t.text_box_mode='filled';
t.box='off';
t.foreground=5;
t.fill_mode='on';
t.background=7;
a=get('current_axes');
a.margins=[0,0,0,0];
xset('window',0); xset('wdim', 500, 200);

```

Anexo 5 – Código Scilab 3

```

M = 0.5;
m = 0.2;
b = 0.1;
I = 0.006;
g = 9.8;
l = 0.3;

q = (M+m)*(I+m*l^2)-(m*l)^2;
tf=5;
s=poly(0,'s');
G=(m*I*s/q)/(s^3 + (b*(I + m*l^2))*s^2/q - ((M + m)*m*g*l)*s/q - b*m*g*l/q);
t=linspace(0,tf,500);
imp_res=csim('imp',t,G);
plot(t,imp_res);xgrid();xtitle(' ','Tempo','Resposta ao Impulso');
xset('window',0); xset('wdim', 500,250);
a=get('current_axes');
a.font_size=2;
a.labels_font_color=2;
a.data_bounds=[0,0,-1;5,120,1];
a.box='on';
a.margins = [0.1,0.01,0.05,0.22];

```

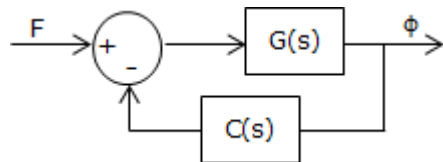
Anexo 6 - Roteiro do pêndulo invertido – Parte 2

Estabilização por controle PID

Como é possível então estabilizar o sistema?

Uma forma de fazer isto é através de um controlador PID.

O que se deseja fazer é adicionar um controlador $C(s)$ na malha de realimentação, então o novo sistema seria:



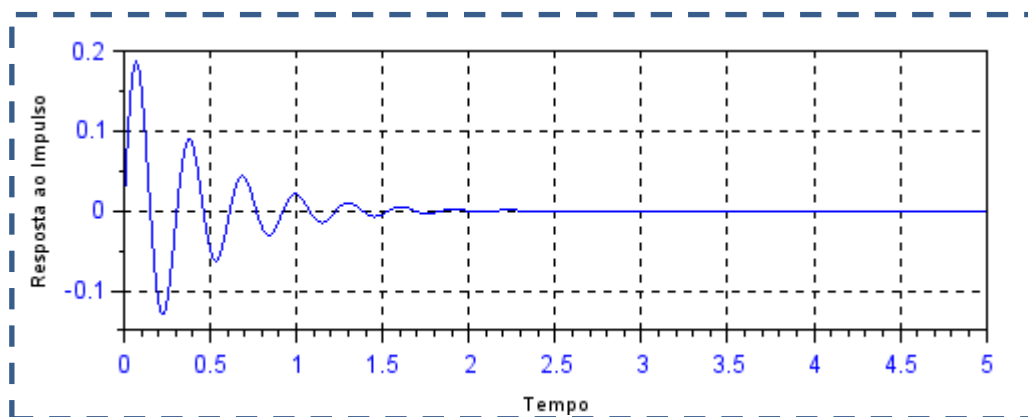
Para mais detalhes sobre o controlador PID, clique na lupinha. ₂

Podemos então tentar encontrar valores de K_p , K_I e K_D que deixam o sistema com as características desejadas. O efeito do controlador no sistema pode ser avaliado através da resposta ao impulso.

$M =$ Kg $I =$ Kg.m²
 $m =$ Kg $g =$ m/s²
 $b =$ N/m/s $l =$ m

$K_p =$ $K_I =$ $K_D =$

Enviar ₄



Código Scilab 4

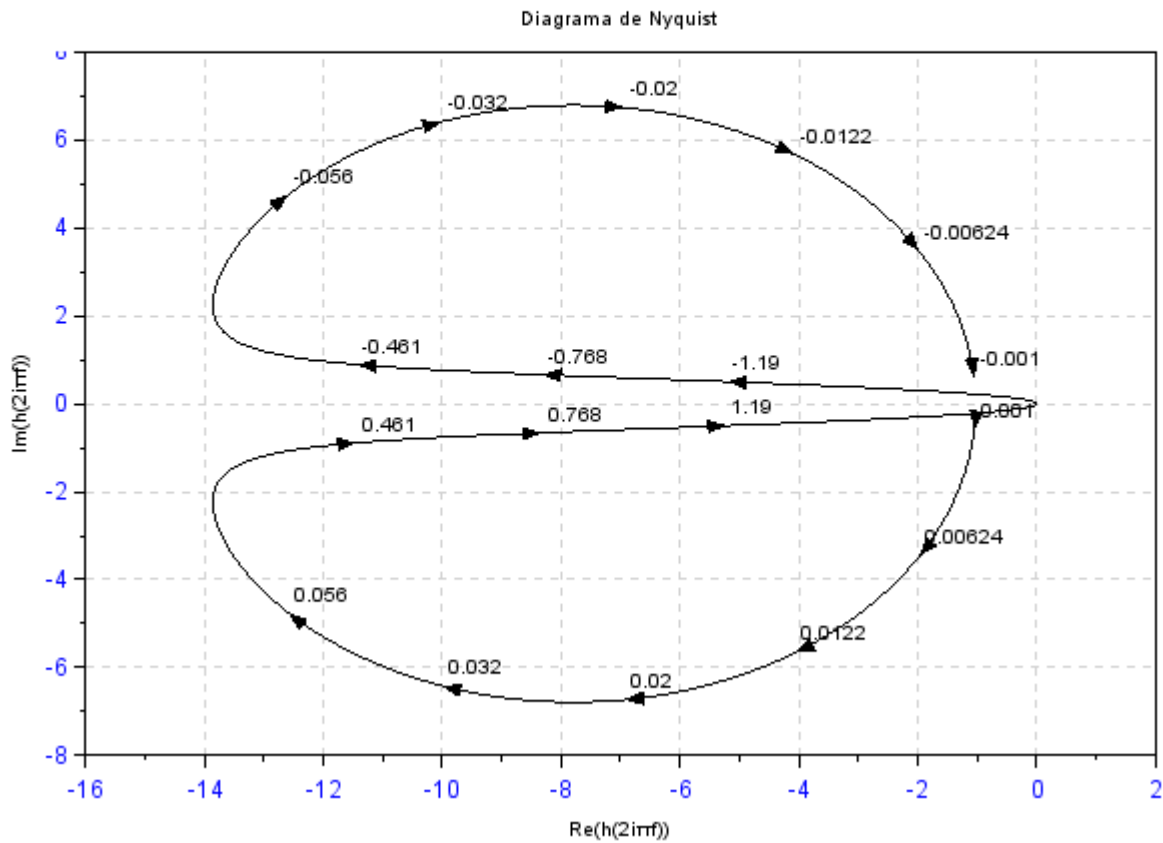
A condição de estabilidade do sistema também pode ser analisada através do diagrama de Nyquist e da FTMA.

Para mais detalhes sobre o critério de estabilidade de Nyquist, clique na lupinha. ₃

Eixo real de a Eixo Imaginário de a

Enviar ₅

$$FTMA = \frac{4.545 + 454.5s + 4.545s^2}{-4.455 - 31.18s + 0.182s^2 + s^3} \quad \text{Polos: } \begin{pmatrix} 5.565 \\ -5.604 \\ -0.143 \end{pmatrix}$$



Código Scilab 5

Anexo 7 – Conteúdo da lupinha 2

O controlador PID é composto por três componentes: Um termo proporcional K_P , um integral K_I e um derivativo K_D .

Sua função de transferência tem então a seguinte forma:

$$K_P + \frac{K_I}{s} + K_D s$$

Cada um dos termos causa diferentes efeitos no sistema:

K_P → Reduz o tempo de subida, também reduz o erro com a referência (porém sem nunca eliminá-lo) e aumenta o Overshoot.

K_I → Tem a capacidade de eliminar o erro com a referência, porém, a resposta transiente será afetada. O tempo de assentamento e o overshoot aumentam.

K_D → Aumenta a estabilidade do sistema, reduzindo o overshoot e o tempo de assentamento.

Anexo 8 – Código Scilab 4

```

M = 0.5;
m = 0.2;
b = 0.1;
I = 0.006;
g = 9.8;
l = 0.3;
kp=100;
ki=1;
kd=1;

q = (M+m)*(I+m*l^2)-(m*l)^2;
s=poly(0,'s');
G=(m*I*s/q)/(s^3 + (b*(I + m*l^2))*s^2/q - ((M + m)*m*g*l)*s/q - b*m*g*l/q);
C=(kd*s^2+kp*s+ki)/s;
G2=G/(1+G*C);
t=linspace(0,5,500);
imp_res=csim('imp',t,G2);
plot(t,imp_res);xgrid();xtitle(' ','Tempo','Resposta ao Impulso');
xset('window',0); xset('wdim', 500, 250);
a=get('current_axes');
a.font_size=2;
a.labels_font_color=2;
a.box='on';
a.margins = [0.1,0.01,0.05,0.22];

```

Anexo 9 – Conteúdo da Lupinha 3

Para que o sistema seja estável, ele não deve ter zeros da FTMF no SPAD. O critério de estabilidade de Nyquist diz que o número Z de zeros no SPAD pode ser calculado através de

$$Z = P - N$$

Onde:

$P \rightarrow$ Número de polos da FTMF no SPAD (que é igual ao número de polos da FTMA no SPAD).

$N \rightarrow$ Número de voltas no sentido horário que o diagrama de Nyquist dá em torno do ponto -1 .

Anexo 10 – Código Scilab 5

```

M = 0.5;
m = 0.2;
b = 0.1;
I = 0.006;
g = 9.8;
l = 0.3;

kp=100;
ki=1;
kd=1;

ri=-16;
rf=2;
ia=-8;
ib=8;

format(8);
s=poly(0,'s');
q = (M+m)*(I+m*l^2)-(m*l)^2;
G=(m*s/q)/(s^3 + (b*(I + m*l^2))*s^2/q - ((M + m)*m*g*l)*s/q - b*m*g*l/q);
C=(kd*s^2+kp*s+ki)/s;
G2=G*C;
den=(s^3 + (b*(I + m*l^2))*s^2/q - ((M + m)*m*g*l)*s/q - b*m*g*l/q);
p=roots(den);
subplot(1,1,1);
H=syslin('c',(G2));
nyquist(H);
a=get('current_axes');
a.font_size=2;
a.labels_font_color=2;
a.box='on';
a.data_bounds=[ri,ia,-1;rf,ib,1];
a.margins = [0.085,0.01,0.25,0.1];
subplot(4,1,1);
str = prettyprint(G2);
str2 = 'FTMA = ';
str4 = prettyprint(p);
str5 = 'Polos: ';
str3=[str2,str,str5,str4];
xstring(0, 0, str3);
t=get('hdl');
t.font_foreground=1;
t.font_size=5;
t.alignment = 'left' ;
t.font_angle=0 ;
t.text_box = [1,1] ;
t.text_box_mode = 'filled' ;
t.box='off';
t.foreground=5;

```



```
t.fill_mode='on';  
t.background=7;  
a=get('current_axes');  
a.margins = [0.05,0.0,0.0,0.0];  
xset('window',0);  
xset('wdim', 600, 600);
```

Anexo 11 - Roteiro do pêndulo invertido – Parte 3

Ajuste do controlador PID através do Lugar das Raízes

Outro método para analisar a estabilidade é através do lugar das raízes. Este método permite desenhar no plano complexo as trajetórias de localização das raízes de um polinômio a partir da variação de um parâmetro, no caso, um ganho K .

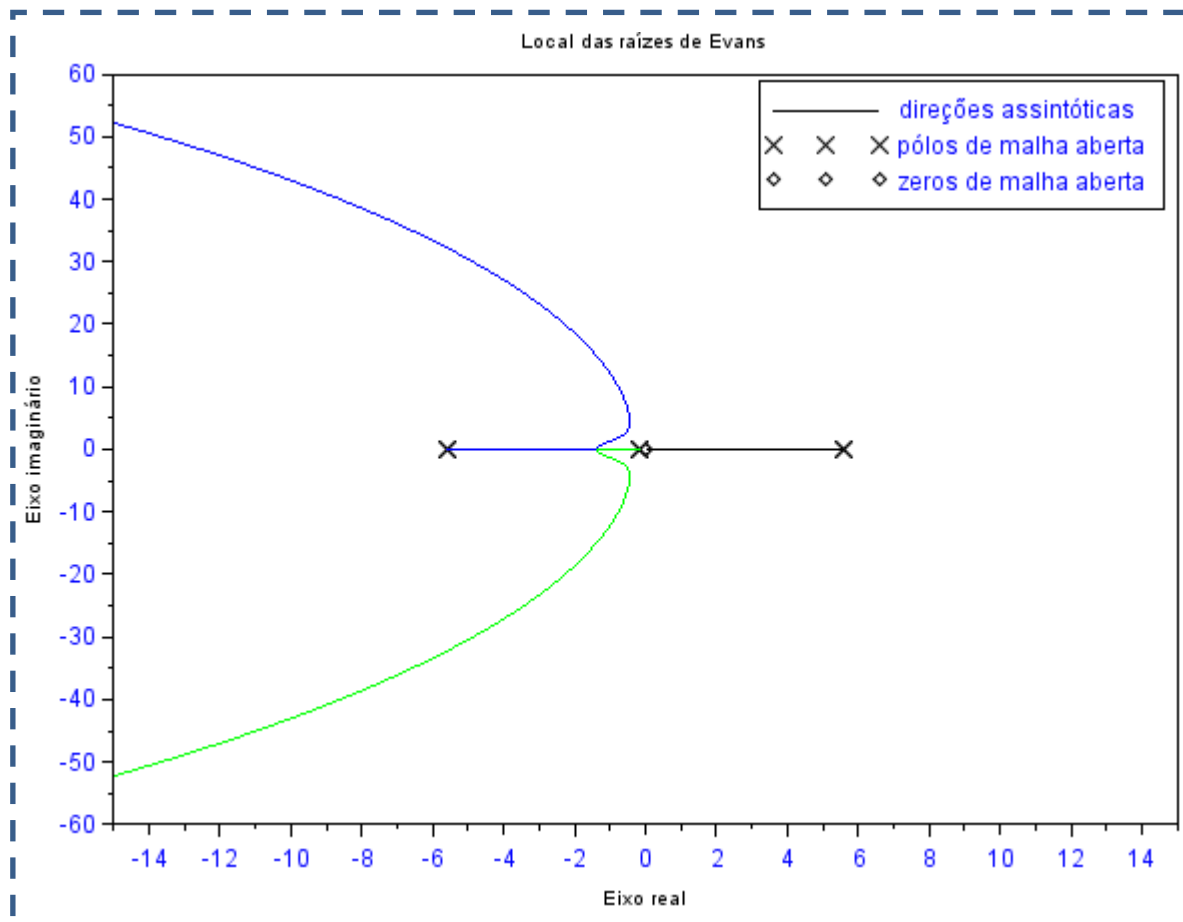
Dados os parâmetros do sistema e do controlador PID, podemos gerar o lugar das raízes:

$M =$ Kg $I =$ Kg.m²
 $m =$ Kg $g =$ m/s²
 $b =$ N/m/s $l =$ m

$K_p =$ $K_I =$ $K_D =$

Eixo real de a Eixo Imaginário de a

Enviar



Código Scilab 6

A partir do Lugar das raízes podemos ajustar o controlador para obedecer alguns requisitos como, por exemplo, que o tempo de acomodação seja de até 0.5s (critério de 2%).

Para mais informações sobre como ajustar o sistema a partir do lugar das raízes, clique na lupinha.



Podemos calcular o ganho K associado a qualquer ponto do lugar das raízes:

Ponto = + j

Enviar

K = 3.025

Tempo de subida = 0.046 s

Tempo de Pico = 0.082 s

Tempo de acomodação (critério de 2%) = 0.5 s

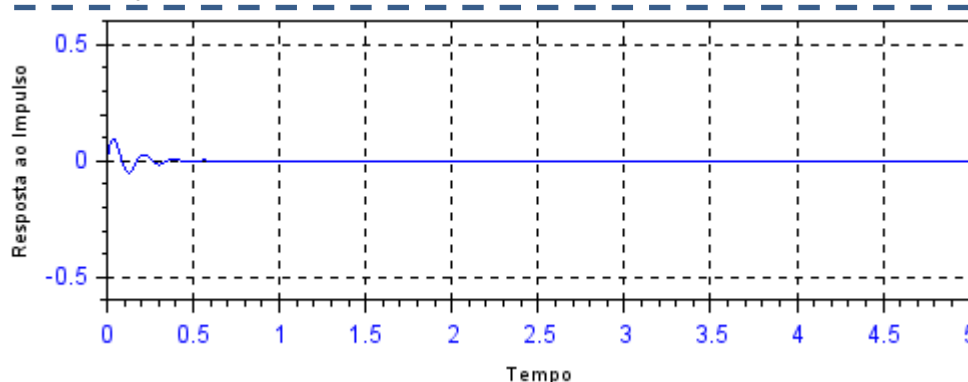
Máximo sobressinal = 0.521

Código Scilab 7

O comportamento final do sistema, reagindo a um impulso para um ganho K aplicado ao controlador, pode ser então verificado:

K =

Enviar



Código Scilab 8

Anexo 12 – Código Scilab 6

```

ri=-15;
rf=15;
ia=-50;
ib=50;

M = 0.5;
m = 0.2;
b = 0.1;
I = 0.006;
g = 9.8;
l = 0.3;
q = (M+m)*(I+m*l^2)-(m*l)^2;

kp=100;
ki=1;
kd=1;

s=poly(0,'s');
G=(m*l*s/q)/(s^3 + (b*(I + m*l^2))*s^2/q - ((M + m)*m*g*l)*s/q - b*m*g*l/q);
C=(kd*s^2+kp*s+ki)/s;
G2=G*C;
xset('window',0); xset('wdim', 600, 500);
evans(G2);
a=get('current_axes');
a.font_size=2;
a.data_bounds=[ri,ia,-1;rf,ib,1];
a.labels_font_color=2;
a.box='on';
a.margins = [0.08,0.01,0.05,0.1];

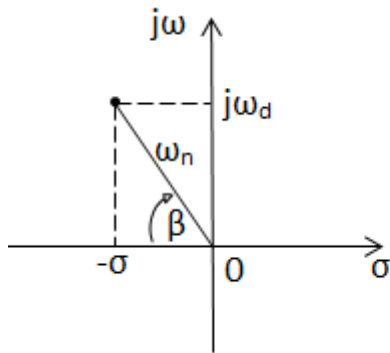
```

Anexo 13 – Conteúdo da lupinha 4

Para o tempo de acomodação t_s (critério de 2%):

$$t_s = 0.5s \rightarrow t_s = \frac{4}{\sigma} \rightarrow \sigma = \frac{4}{0.5} \rightarrow \sigma = 8$$

Essas informações estão contidas no lugar das raízes da seguinte forma:



$$s = -\sigma \pm j\omega_d$$

$$\omega_d = \omega_n \sqrt{1 - \zeta^2}$$

$$\sigma = \zeta \omega_n$$

$$\cos \beta = \zeta$$

Então, através do lugar das raízes, buscamos um ponto em que $-\sigma = -8$

Logo, se existir um ponto que a parte real é igual a -0,8 (aproxime a visualização do lugar das raízes em torno da região de interesse para encontrar um valor mais preciso para o ponto), e sabendo que todos os pontos do lugar das raízes obedecem à equação:

$$1 + K \cdot G(s)C(s) = 0$$

Podemos calcular o ganho K associado a este ponto e ainda outras características do sistema.

Anexo 14 – Código Scilab 7

```

M = 0.5;
m = 0.2;
b = 0.1;
I = 0.006;
g = 9.8;
l = 0.3;

kp=100;
ki=1;
kd=1;

rea=-8;
im=38.5;

re=-rea;
q = (M+m)*(I+m*l^2)-(m*l)^2;
format(6);
s=poly(0,'s');
G=(m*I*s/q)/(s^3 + (b*(I + m*l^2))*s^2/q - ((M + m)*m*g*l)*s/q - b*m*g*l/q);
C=(kd*s^2+kp*s+ki)/s;
G2=G*C;
h=re+im*i;
G4=(m*I*h/q)/(h^3 + (b*(I + m*l^2))*h^2/q - ((M + m)*m*g*l)*h/q - b*m*g*l/q);
C4=(kd*h^2+kp*h+ki)/h;
k1=-1/(G4*C4);
k=abs(k1);
wn=sqrt((im^2)+(re^2));
am=re/wn;
pi=3.14159265359;
tp=pi/im;
ts=4/re;
be=acos(am);
tr=(pi-be)/im;
mp=exp(-pi*(am/sqrt(1-am^2)));
xset('window',0); xset('wdim', 450, 120);
subplot(5,1,1);
str = prettyprint(k);
str2 = ' K =';
str3=[str2,str];
xstring(0, 0, str3);
t=get('hdl');
t.font_foreground=1;
t.font_size=3.8;
t.alignment = 'left' ;
t.font_angle=0 ;
t.text_box = [1,1] ;
t.text_box_mode = 'off' ;
t.box='off';
t.foreground=5;
t.fill_mode='on';

```

```

t.background=7;
a=get('current_axes');
a.margins = [0,0,0,0];
subplot(5,1,2);
str = prettyprint(tr);
str2 = 'Tempo de subida =';
str1 ='s';
str3=[str2,str,str1];
xstring(0, 0, str3);
t=get('hdl');
t.font_foreground=1;
t.font_size=3.8;
t.alignment = 'left' ;
t.font_angle=0 ;
t.text_box = [1,1] ;
t.text_box_mode = 'off' ;
t.box='off';
t.foreground=5;
t.fill_mode='on';
t.background=7;
a=get('current_axes');
a.margins = [0,0,0,0];
subplot(5,1,3);
str = prettyprint(tp);
str2 = 'Tempo de Pico =';
str1 ='s';
str3=[str2,str,str1];
xstring(0, 0, str3);
t=get('hdl');
t.font_foreground=1;
t.font_size=3.8;
t.alignment = 'left' ;
t.font_angle=0 ;
t.text_box = [1,1] ;
t.text_box_mode = 'off' ;
t.box='off';
t.foreground=5;
t.fill_mode='on';
t.background=7;
a=get('current_axes');
a.margins = [0,0,0,0];
subplot(5,1,4);
str = prettyprint(ts);
str2 = 'Tempo de acomodação (critério de 2%) =';
str1 ='s';
str3=[str2,str,str1];
xstring(0, 0, str3);
t=get('hdl');
t.font_foreground=1;
t.font_size=3.8;
t.alignment = 'left' ;
t.font_angle=0 ;
t.text_box = [1,1] ;
t.text_box_mode = 'off' ;

```



```
t.box='off';
t.foreground=5;
t.fill_mode='on';
t.background=7;
a=get('current_axes');
a.margins = [0,0,0,0];
subplot(5,1,5);
str = prettyprint(mp);
str2 = ' Máximo sobressinal =';
str3=[str2,str];
xstring(0, 0, str3);
t=get('hdl');
t.font_foreground=1;
t.font_size=3.8;
t.alignment = 'left' ;
t.font_angle=0 ;
t.text_box = [1,1] ;
t.text_box_mode = 'off' ;
t.box='off';
t.foreground=5;
t.fill_mode='on';
t.background=7;
a=get('current_axes');
a.margins = [0,0,0,0];
```

Anexo 15 – Código Scilab 8

```

M = 0.5;
m = 0.2;
b = 0.1;
I = 0.006;
g = 9.8;
l = 0.3;

kp=100;
ki=1;
kd=1;

k=3.025;

s=poly(0,'s');
q = (M+m)*(I+m*l^2)-(m*l)^2;
G=(m*l*s/q)/(s^3 + (b*(I + m*l^2))*s^2/q - ((M + m)*m*g*l)*s/q - b*m*g*l/q);
C=(kd*s^2+kp*s+ki)/s;
G2=G/(1+k*C);
t=linspace(0,5,500);
imp_res=csim('imp',t,G2);
plot(t,imp_res);xgrid();xtitle(' ','Tempo','Resposta ao Impulso');
xset('window',0); xset('wdim', 500, 250);
a=get('current_axes');
a.font_size=2;
a.labels_font_color=2;
a.data_bounds=[0,-0.5,-1;5,0.5,1];
a.box='on';
a.margins = [0.1,0.01,0.05,0.22];

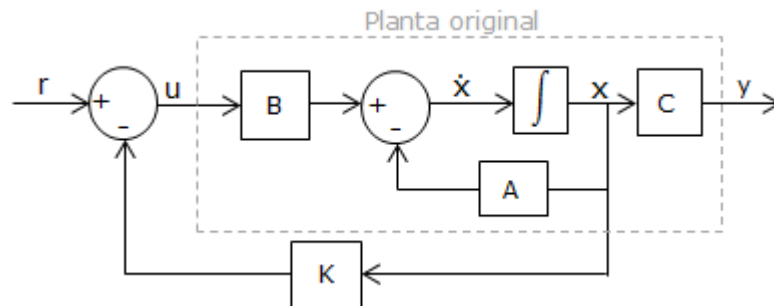
```

Anexo 16 – Roteiro do pêndulo invertido – Parte 4

Controle por realimentação de estados

Outra forma de controlar o sistema é através da realimentação de estados, que consiste em realimentar todos os estados do sistema, multiplicados por uma matriz de ganhos K em vez de usar a saída do sistema como realimentação. Isto permite reposicionar os polos do sistema para torná-lo estável.

A imagem a seguir representa o diagrama de blocos da planta original com o controle por realimentação de estados.



Como visto anteriormente, dados os parâmetros do sistema do pêndulo invertido, o espaço de estados do sistema sem controle torna-se:

$$\begin{aligned} M &= 0.5 \text{ Kg} & I &= 0.006 \text{ Kg.m}^2 \\ m &= 0.2 \text{ Kg} & g &= 9.8 \text{ m/s}^2 \\ b &= 0.1 \text{ N/m/s} & l &= 0.3 \text{ m} \end{aligned}$$

Enviar

9 10 11

$$\left\{ \begin{aligned} \dot{X}(t) &= \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & -0.182 & 2.673 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -0.455 & 31.18 & 0 \end{pmatrix} X(t) + \begin{pmatrix} 0 \\ 1.818 \\ 0 \\ 4.545 \end{pmatrix} U(t) \\ Y(t) &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} X(t) \end{aligned} \right.$$

Código Scilab 9

A instabilidade do sistema pode ser verificada através dos polos de malha aberta, que são os autovalores da matriz A. Se o sistema possuir polos no SPAD, ele é instável.

Polos:
P1 = 0
P2 = -5.6040941
P3 = -0.1428316
P4 = 5.5651076

Código Scilab 10

Através da realimentação, é possível obter um novo sistema em que os polos de malha fechada podem ser reposicionados arbitrariamente se o par (A, B) for controlável.

O sistema é controlável se o posto da matriz $A_{n \times n}$ for igual a n, neste caso, n precisa ser igual 4.

Matriz de Controlabilidade:

$$\begin{pmatrix} 0 & 1.818 & -0.331 & 12.21 \\ 1.818 & -0.331 & 12.21 & -4.429 \\ 0 & 4.545 & -0.826 & 141.9 \\ 4.545 & -0.826 & 141.9 & -31.32 \end{pmatrix}$$

Posto = 4

Código Scilab 11

Após verificar a controlabilidade, podemos então determinar a nova posição desejada para os polos e, através da fórmula de Ackerman, calcular o valor de K que realiza o ajuste necessário.

O objetivo principal para atingir a estabilidade é reposicionar o polo que está no SPAD, mas os outros polos também podem ser ajustados.

P1 = P2 = P3 = P4 =

Enviar

12 13

K1 = 0
K2 = 0.013
K3 = 13.87
K4 = 2.474

Código Scilab 12

Finalmente, podemos aplicar o ganho K na malha de realimentação e chegar ao sistema controlado. O sistema original era da forma:

$$\dot{x} = A \cdot x + B \cdot u$$

$$y = C \cdot x$$

O sistema controlado será da forma:

$$\dot{x} = (A - B \cdot k) \cdot x + B \cdot u$$

$$y = C \cdot x$$

Então, podemos verificar a nova posição dos polos do Sistema realimentado através dos autovalores de (A-B.k):

Polos:
P1 = 0
P2 = -0.15
P3 = -6
P4 = -5.6

Código Scilab 13

Se o sistema não tiver mais polos no SPAD após a aplicação dos ganhos, então ele terá se tornado estável.

Anexo 17 – Código Scilab 9

```

M = 0.5;
m = 0.2;
b = 0.1;
I = 0.006;
g = 9.8;
l = 0.3;

p = I*(M+m)+M*m*l^2;
format(6);
A = [0      1      0      0;
      0 -(I+m*l^2)*b/p (m^2*g*l^2)/p  0;
      0      0      0      1;
      0 -(m*l*b)/p      m*g*l*(M+m)/p  0];
B = [ 0;
      (I+m*l^2)/p;
      0;
      m*l/p];
C = [1 0 0 0;
      0 0 1 0];
D = [0;
      0];
ss=syslin('c',A,B,C);
str = prettyprint(ss);
xstring(0, 0, str);
t=get('hdl');
t.font_foreground=1;
t.font_size=5;
t.alignment = 'left';
t.font_angle=0;
t.text_box=[1,1];
t.text_box_mode='filled';
t.box='off';
t.foreground=5;
t.fill_mode='on';
t.background=7;
a=get('current_axes');
a.margins=[0,0,0,0];
xset('window',0); xset('wdim', 500, 200);

```

Anexo 18 – Código Scilab 10

```

M = 0.5;
m = 0.2;
b = 0.1;
I = 0.006;
g = 9.8;
l = 0.3;

p = I*(M+m)+M*m*l^2;

A = [0      1      0      0;
      0 -(I+m*l^2)*b/p (m^2*g*l^2)/p 0;
      0      0      0      1;
      0 -(m*l*b)/p      m*g*l*(M+m)/p 0];
B = [ 0;
      (I+m*l^2)/p;
      0;
      m*l/p];
C = [1 0 0 0;
      0 0 1 0];
D = [0;
      0];

ss=syslin('c',A,B,C);
po=spec(A);
format(6);
subplot(5,1,1);
str2 = 'Polos: ';
xstring(0, 0, str2);
t=get('hdl');
t.font_foreground=1;
t.font_size=3.8;
t.alignment = 'left' ;
t.font_angle=0 ;
t.text_box = [1,1] ;
t.text_box_mode = 'off' ;
t.box='off';
t.foreground=5;
t.fill_mode='on';
t.background=7;
a=get('current_axes');
a.margins = [0,0,0,0];
subplot(5,1,2);
str = prettyprint(po(1));
str2 = 'P1 =';
str3=[str2,str];
xstring(0, 0, str3);
t=get('hdl');
t.font_foreground=1;
t.font_size=3.8;
t.alignment = 'left' ;
t.font_angle=0 ;

```

```

t.text_box = [1,1] ;
t.text_box_mode = 'off' ;
t.box='off';
t.foreground=5;
t.fill_mode='on';
t.background=7;
a=get('current_axes');
a.margins = [0,0,0,0];
subplot(5,1,3);
str = prettyprint(po(2));
str2 = 'P2 =';
str3=[str2,str];
xstring(0, 0, str3);
t=get('hdl');
t.font_foreground=1;
t.font_size=3.8;
t.alignment = 'left' ;
t.font_angle=0 ;
t.text_box = [1,1] ;
t.text_box_mode = 'off' ;
t.box='off';
t.foreground=5;
t.fill_mode='on';
t.background=7;
a=get('current_axes');
a.margins = [0,0,0,0];
subplot(5,1,4);
str = prettyprint(po(3));
str2 = 'P3 =';
str3=[str2,str];
xstring(0, 0, str3);
t=get('hdl');
t.font_foreground=1;
t.font_size=3.8;
t.alignment = 'left' ;
t.font_angle=0 ;
t.text_box = [1,1] ;
t.text_box_mode = 'off' ;
t.box='off';
t.foreground=5;
t.fill_mode='on';
t.background=7;
a=get('current_axes');
a.margins = [0,0,0,0];
subplot(5,1,5);
str = prettyprint(po(4));
str2 = 'P4 =';
str3=[str2,str];
xstring(0, 0, str3);
t=get('hdl');
t.font_foreground=1;
t.font_size=3.8;
t.alignment = 'left' ;
t.font_angle=0 ;

```



```
t.text_box = [1,1] ;  
t.text_box_mode = 'off' ;  
t.box='off';  
t.foreground=5;  
t.fill_mode='on';  
t.background=7;  
a=get('current_axes');  
a.margins = [0,0,0,0];  
xset('wdim', 250, 120);
```


Anexo 19 – Código Scilab 11

```

M = 0.5;
m = 0.2;
b = 0.1;
I = 0.006;
g = 9.8;
l = 0.3;

p = I*(M+m)+M*m*I^2;
A = [0      1      0      0;
      0 -(I+m*I^2)*b/p (m^2*g*I^2)/p 0;
      0      0      0      1;
      0 -(m*I*b)/p      m*g*I*(M+m)/p 0];
B = [ 0;
      (I+m*I^2)/p;
      0;
      m*I/p];
C = [1 0 0 0;
      0 0 1 0];
D = [0;
      0];
format(6);
ss=syslin('c',A,B,C);
cont=cont_mat(ss);
posto=rank(cont);
xset('wdim', 500, 150);
str = prettyprint(cont);
str2 = 'Matriz de Controlabilidade: ';
str3=[str2,str];
xstring(0, 0, str3);
t=get('hdl');
t.font_foreground=1;
t.font_size=3.5;
t.alignment = 'left' ;
t.font_angle=0 ;
t.text_box = [1,1] ;
t.text_box_mode = 'off' ;
t.box='off';
t.foreground=5;
t.fill_mode='on';
t.background=7;
a=get('current_axes');
a.margins = [0,0,0,0];
str = prettyprint(posto);
str2 = 'Posto =';
str3=[str2,str];
xstring(0, 0, str3);
t=get('hdl');
t.font_foreground=1;
t.font_size=3.8;
t.alignment = 'left' ;
t.font_angle=0 ;
t.text_box = [1,1] ;

```



```
t.text_box_mode = 'off' ;  
t.box='off';  
t.foreground=5;  
t.fill_mode='on';  
t.background=7;  
a=get('current_axes');  
a.margins = [0,0,0,0];
```

Anexo 20 – Código Scilab 12

```

M = 0.5;
m = 0.2;
b = 0.1;
I = 0.006;
g = 9.8;
l = 0.3;

p1=0;
p2=-5.6;
p3=0.15;
p4=-6;

format(6);
p = I*(M+m)+M*m*l^2;
A = [0      1      0      0;
      0 -(I+m*l^2)*b/p (m^2*g*l^2)/p  0;
      0      0      0      1;
      0 -(m*l*b)/p      m*g*l*(M+m)/p  0];
B = [ 0;
      (I+m*l^2)/p;
      0;
      m*l/p];
C = [1 0 0 0;
      0 0 1 0];
D = [0;
      0];

ss=syslin('c',A,B,C);
p=[p1 p2 p3 p4];
function [K] = acker_coef(A,B,alpha);
sa = size(A);
n = sa(1);
Com = cont_mat(A,B);
Comi = inv(Com);
alpha = [alpha,1];
alphaC = alpha(1) * eye(n,n);
for i = 2:n+1;
alphaC = alphaC+alpha(i)*A^(i-1);
end;
K = [zeros(1,n-1),1]*Comi*alphaC;
endfunction;
function [K] = acker(A,B,p);
n = length(p);
s = poly(0,'s');
equation = 1;
for i = 1:n;
equation = equation*(s-p(i));
end;
alpha = coeff(equation);
K = acker_coef(A,B,alpha);
endfunction;
ac=acker(A,B,p);

```

```

subplot(4,1,1);
str = prettyprint(ac(1));
str2 = 'K1 =';
str3=[str2,str];
xstring(0, 0, str3);
t=get('hdl');
t.font_foreground=1;

t.font_size=3.8;
t.alignment = 'left' ;
t.font_angle=0 ;
t.text_box = [1,1] ;
t.text_box_mode = 'off' ;
t.box='off';
t.foreground=5;
t.fill_mode='on';
t.background=7;
a=get('current_axes');
a.margins = [0,0,0,0];
subplot(4,1,2);
str = prettyprint(ac(2));
str2 = 'K2 =';
str3=[str2,str];
xstring(0, 0, str3);
t=get('hdl');
t.font_foreground=1;
t.font_size=3.8;
t.alignment = 'left' ;
t.font_angle=0 ;
t.text_box = [1,1] ;
t.text_box_mode = 'off' ;
t.box='off';
t.foreground=5;
t.fill_mode='on';
t.background=7;
a=get('current_axes');
a.margins = [0,0,0,0];
subplot(4,1,3);
str = prettyprint(ac(3));
str2 = 'K3 =';
str3=[str2,str];
xstring(0, 0, str3);
t=get('hdl');
t.font_foreground=1;
t.font_size=3.8;
t.alignment = 'left' ;
t.font_angle=0 ;
t.text_box = [1,1] ;
t.text_box_mode = 'off' ;
t.box='off';
t.foreground=5;
t.fill_mode='on';
t.background=7;
a=get('current_axes');

```

```
a.margins = [0,0,0,0];  
subplot(4,1,4);  
str = prettyprint(ac(4));  
str2 = 'K4 =';  
str3=[str2,str];  
xstring(0, 0, str3);  
t=get('hdl');  
t.font_foreground=1;  
t.font_size=3.8;  
t.alignment = 'left' ;  
t.font_angle=0 ;  
t.text_box = [1,1] ;  
t.text_box_mode = 'off' ;  
t.box='off';  
t.foreground=5;  
t.fill_mode='on';  
t.background=7;  
a=get('current_axes');  
a.margins = [0,0,0,0];  
xset('wdim', 250, 100);
```

Anexo 21 – Código Scilab 13

```

M = 0.5;
m = 0.2;
b = 0.1;
I = 0.006;
g = 9.8;
l = 0.3;

p1=0;
p2=-5.6;
p3=-0.15;
p4=-6;

p = I*(M+m)+M*m*I^2;
A = [0      1      0      0;
      0 -(I+m*I^2)*b/p (m^2*g*I^2)/p 0;
      0      0      0      1;
      0 -(m*I*b)/p      m*g*I*(M+m)/p 0];
B = [ 0;
      (I+m*I^2)/p;
      0;
      m*I/p];
C = [1 0 0 0;
      0 0 1 0];
D = [0;
      0];
ss=syslin('c',A,B,C);
p=[p1 p2 p3 p4];
function [K] = acker_coef(A,B,alpha);
sa = size(A);
n = sa(1);
Com = cont_mat(A,B);
Comi = inv(Com);
alpha = [alpha,1];
alphaC = alpha(1) * eye(n,n);
for i = 2:n+1;
alphaC = alphaC + alpha(i) * A^(i-1);
end;
K = [zeros(1,n-1),1]*Comi*alphaC;
endfunction;
function [K] = acker(A,B,p);
n = length(p);
s = poly(0,'s');
equation = 1;
for i = 1:n;
equation = equation*(s-p(i));
end;
alpha = coeff(equation);
K = acker_coef(A,B,alpha);
endfunction;
ac=acker(A,B,p);
k=[ac(1) ac(2) ac(3) ac(4)];
abk=A-B*k;

```

```

po=spec(abk);
subplot(5,1,1);
str2 = 'Polos: ';
xstring(0, 0, str2);
t=get('hdl');
t.font_foreground=1;
t.font_size=3.8;
t.alignment = 'left' ;
t.font_angle=0 ;
t.text_box = [1,1] ;
t.text_box_mode = 'off' ;
t.box='off';
t.foreground=5;
t.fill_mode='on';
t.background=7;
a=get('current_axes');
a.margins = [0,0,0,0];
subplot(5,1,2);
str = prettyprint(po(1));
str2 = 'P1 =';
str3=[str2,str];
xstring(0, 0, str3);
t=get('hdl');
t.font_foreground=1;
t.font_size=3.8;
t.alignment = 'left' ;
t.font_angle=0 ;
t.text_box = [1,1] ;
t.text_box_mode = 'off' ;
t.box='off';
t.foreground=5;
t.fill_mode='on';
t.background=7;
a=get('current_axes');
a.margins = [0,0,0,0];
subplot(5,1,3);
str = prettyprint(po(2));
str2 = 'P2 =';
str3=[str2,str];
xstring(0, 0, str3);
t=get('hdl');
t.font_foreground=1;
t.font_size=3.8;
t.alignment = 'left' ;
t.font_angle=0 ;
t.text_box = [1,1] ;
t.text_box_mode = 'off' ;
t.box='off';
t.foreground=5;
t.fill_mode='on';
t.background=7;
a=get('current_axes');
a.margins = [0,0,0,0];
subplot(5,1,4);

```

```
str = prettyprint(po(3));
str2 = 'P3 =';
str3=[str2,str];
xstring(0, 0, str3);
t=get('hdl');
t.font_foreground=1;
t.font_size=3.8;
t.alignment = 'left' ;
t.font_angle=0 ;
t.text_box = [1,1] ;
t.text_box_mode = 'off' ;
t.box='off';
t.foreground=5;
t.fill_mode='on';
t.background=7;
a=get('current_axes');
a.margins = [0,0,0,0];
subplot(5,1,5);
str = prettyprint(po(4));
str2 = 'P4 =';
str3=[str2,str];
xstring(0, 0, str3);
t=get('hdl');
t.font_foreground=1;
t.font_size=3.8;
t.alignment = 'left' ;
t.font_angle=0 ;
t.text_box = [1,1] ;
t.text_box_mode = 'off' ;
t.box='off';
t.foreground=5;
t.fill_mode='on';
t.background=7;
a=get('current_axes');
a.margins = [0,0,0,0];
xset('wdim', 250, 120);
```


Anexo 22 – Roteiro do Motor DC – Parte 1

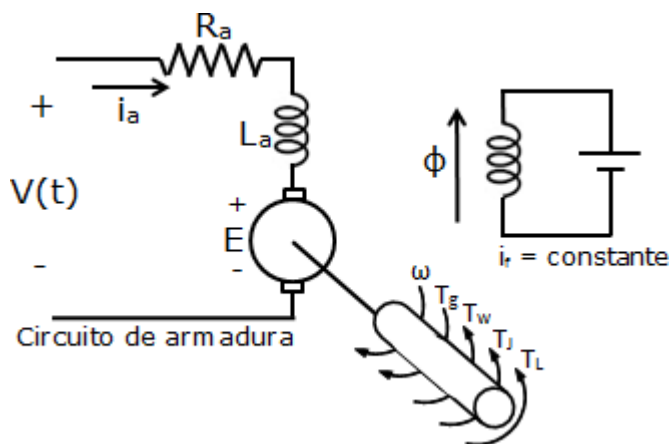
Modelo por variáveis de estados e função de transferência

A abordagem de técnicas de controle no acionamento de motores a corrente contínua é um campo bastante explorado na engenharia devido a sua versatilidade nas aplicações.

Os de pequeno porte são muito utilizados em brinquedos e equipamentos portáteis pelo fato de poderem ser acionados por meio de pilhas e baterias. Já os de grande porte são utilizados para tração elétrica em trens e metrô pelo fato de permitirem fácil e precisa variação de velocidade.

Os motores CC de excitação independente podem ter sua velocidade controlada através da variação da corrente de campo I_f ou da tensão de armadura V_a . Neste objeto educacional trataremos deste segundo caso, então consideraremos a corrente de campo sempre constante.

A figura a seguir apresenta a modelagem do motor de corrente contínua com excitação independente e os seus componentes.



Onde:

$R_a \rightarrow$ Resistência de armadura

$i_a \rightarrow$ Corrente de armadura

$L_a \rightarrow$ Indutância do enrolamento de armadura

$E \rightarrow$ Força contra eletromotriz induzida

$V \rightarrow$ Tensão de Armadura

$\Phi \rightarrow$ Fluxo magnético


$\omega \rightarrow$ velocidade angular de rotação do eixo

$T_g \rightarrow$ Conjugado eletromagnético desenvolvido pelo motor

$T_w \rightarrow$ Conjugado de perdas

$T_j \rightarrow$ Conjugado devido a inércia do eixo

$T_L \rightarrow$ Conjugado da carga

Para informações sobre a modelagem do sistema, clique na lupinha. 

O Sistema trata a tensão de armadura $V(s)$ e o torque da carga $T_L(s)$ como variáveis de entrada e a velocidade $\Omega(s)$ ou a posição do eixo $\Theta(s)$ como variáveis de saída.

A função de transferência tensão-velocidade $G_1(s)$ é:

$$G_1(s) = \frac{\Omega(s)}{V(s)} = \frac{K_t}{(sL_a + R_a)(sJ + B) + K_e K_t}$$

Multiplicando-a por $1/s$ obtém-se a função de transferência tensão-posição $H_1(s)$:

$$H_1(s) = \frac{\Omega(s)}{V(s)} \frac{1}{s} = \frac{\Theta(s)}{V(s)} = \frac{K_t}{s(sL_a + R_a)(sJ + B) + sK_e K_t}$$

A função de transferência torque da carga-velocidade $G_2(s)$ é:

$$G_2(s) = \frac{\Omega(s)}{T_f(s) + T_L(s)} = \frac{-\frac{1}{sJ + B}}{1 + \frac{K_e K_t}{(sL_a + R_a)(sJ + B)}}$$

Multiplicando-a por $1/s$ obtém-se a função de transferência torque da carga-posição $H_2(s)$:

$$H_2(s) = \frac{\Omega(s)}{T_r(s) + T_L(s)} \frac{1}{s} = \frac{\Theta(s)}{T_r(s) + T_L(s)} = \frac{-\frac{1}{sJ+B}}{s + \frac{sK_e K_t}{(sL_a + R_a)(sJ+B)}}$$

O sistema pode ser representado também na forma de espaço de estados, usando como variáveis de estado a velocidade angular e a corrente de armadura. A tensão de armadura é tratada como entrada e a velocidade angular como saída.

$$\frac{d}{dt} \begin{bmatrix} \dot{\theta} \\ i \end{bmatrix} = \begin{bmatrix} \frac{-b}{J} & \frac{k}{J} \\ \frac{-K}{L} & \frac{-R}{L} \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ i \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} v$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ i \end{bmatrix}$$

Então, dados os parâmetros de um motor de corrente contínua:

$$\begin{aligned} R_a &= \boxed{1} \, \Omega & J &= \boxed{0.01} \, \text{Kg.m}^2 \\ L_a &= \boxed{0.5} \, \text{H} & B &= \boxed{0.1} \, \text{Kg.m}^2/\text{s} \\ K_t = K_e &= \boxed{0.01} \, \text{Nm/A} \end{aligned}$$

Enviar

14 15

As funções de transferência tornam-se:

$$\begin{aligned} G1(s) &= \frac{2}{20.02 + 12s + s^2} \\ H1(s) &= \frac{2}{20.02s + 12s^2 + s^3} \\ G2(s) &= \frac{-200 - 99.9998s}{20.0002 + 12s + s^2} \\ H2(s) &= \frac{-200 - 99.9998s}{20.0002s + 12s^2 + s^3} \end{aligned}$$

Código Scilab 14

E o espaço de estados torna-se:

$$\begin{cases} \dot{X}(t) = \begin{pmatrix} -10 & 1 \\ -0.02 & -2 \end{pmatrix} X(t) + \begin{pmatrix} 0 \\ 2 \end{pmatrix} U(t) \\ Y(t) = \begin{pmatrix} 1 & 0 \end{pmatrix} X(t) \end{cases}$$

Código Scilab 15

Onde: $\dot{X}(t) = \begin{bmatrix} \dot{\theta} \\ i \end{bmatrix}$ e $X(t) = \begin{bmatrix} \theta \\ i \end{bmatrix}$

Anexo 23 – Conteúdo da lupinha 5

A equação elétrica pode ser obtida a partir da Lei de Kirchoff no circuito de armadura. A partir das leis de Newton para o movimento rotacional pode-se determinar as equações mecânicas. E, finalmente, a relação entre a parte elétrica e a mecânica é obtida pela Lei de Faraday-Neumann-Lenz.

Aplicando a Transformada de Laplace, obtêm-se as equações no domínio da frequência:

$$V(s) = R_a i_a(s) + sL_a I_a(s) + E(s)$$

$$T_g(s) = J s \omega(s) + B \omega(s) + T_f(s) + T_L(s)$$

$$T_g(s) = K_t I_a(s)$$

$$E(s) = K_e \omega(s)$$

Onde:

$J \rightarrow$ Momento de inércia total do rotor e da carga em relação ao eixo.

$B \rightarrow$ Constante de atrito viscoso.

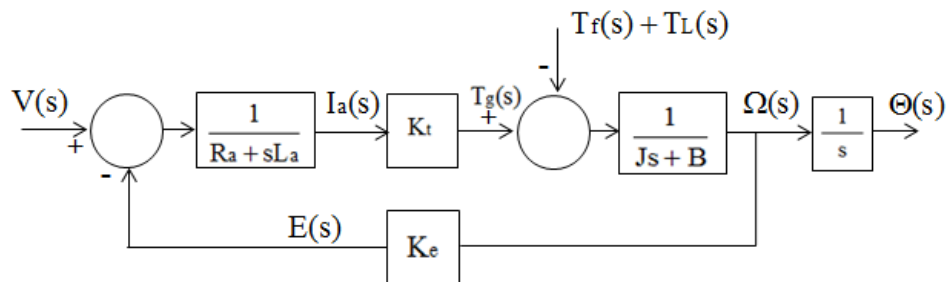
$T_f \rightarrow$ Conjugado referente ao atrito na carga e no motor (é parte do conjugado de perdas T_w).

$K_t \rightarrow$ Constante de torque do motor.

$K_e \rightarrow$ Constante da força eletromotriz.

Note que $K_t = K_e$, pois a potência mecânica desenvolvida deve ser igual a potência elétrica absorvida no roto.

Estas equações podem ser representadas pelo seguinte diagrama de blocos:



Anexo 24 – Código Scilab 14

```

ra=1;
la=0.5;
kt=0.01;
j=0.01;
b=0.1;

format(6);
s=poly(0,'s');
Ga=(kt)/((ra*j*s+b*ra+la*s*s*j+la*s*b+kt*kt));
c=coeff(Ga.den);
G1=(kt/c(3))/((ra*j*s+b*ra+la*s*s*j+la*s*b+kt*kt)/c(3));
H1=G1/s;
num=-1/(j*s+b);
den=1+((kt*kt)/(la*s+ra)*(j*s+b));
Gb=(num)/(den);
d=coeff(Gb.den);
G2=(num/d(3))/(den/d(3));
H2=G2/s;
format(8);
subplot(4,1,1);
str = prettyprint(G1);
str2 = 'G1(s)=';
str3=[str2,str];
xstring(0, 0, str3);
t=get('hdl');
t.font_foreground=1;
t.font_size=5;
t.alignment = 'left' ;
t.font_angle=0 ;
t.text_box = [1,1] ;
t.text_box_mode = 'filled' ;
t.box='off';
t.foreground=5;
t.fill_mode='on';
t.background=7;
a=get('current_axes');
a.margins = [0.075,0.01,0.05,0.1];
subplot(4,1,2);
str = prettyprint(H1);
str2 = 'H1(s)=';
str3=[str2,str];
xstring(0, 0, str3);
t=get('hdl');
t.font_foreground=1;
t.font_size=5;
t.alignment = 'left' ;
t.font_angle=0 ;

t.text_box = [1,1] ;
t.text_box_mode = 'filled' ;
t.box='off';
t.foreground=5;

```

```

t.fill_mode='on';
t.background=7;
a=get('current_axes');
a.margins = [0.075,0.01,0.05,0.1];
subplot(4,1,3);
str = prettyprint(G2);
str2 = 'G2(s)=';
str3=[str2,str];
xstring(0, 0, str3);
t=get('hdl');
t.font_foreground=1;
t.font_size=5;
t.alignment = 'left' ;
t.font_angle=0 ;
t.text_box = [1,1] ;
t.text_box_mode = 'filled' ;
t.box='off';
t.foreground=5;
t.fill_mode='on';
t.background=7;
a=get('current_axes');
a.margins = [0.075,0.01,0.05,0.1];
subplot(4,1,4);
str = prettyprint(H2);
str2 = 'H2(s)=';
str3=[str2,str];
xstring(0, 0, str3);
t=get('hdl');
t.font_foreground=1;
t.font_size=5;
t.alignment = 'left' ;
t.font_angle=0 ;
t.text_box = [1,1] ;
t.text_box_mode = 'filled' ;
t.box='off';
t.foreground=5;
t.fill_mode='on';
t.background=7;
a=get('current_axes');
a.margins = [0.075,0.01,0.05,0.1];
xset('window',0); xset('wdim', 400, 350);

```

Anexo 25 – Código Scilab 15

```
ra=1;
la=0.5;
kt=0.01;
j=0.01;
b=0.1;

A = [-b/j  kt/j;
      -kt/la -ra/la];
B = [0;
      1/la];
C = [1  0];
D = 0;
ss=syslin('c',A,B,C);
str = prettyprint(ss);
xstring(0, 0, str);
t=get('hdl');
t.font_foreground=1;
t.font_size=5;
t.alignment = 'left' ;
t.font_angle=0 ;
t.text_box = [1,1] ;
t.text_box_mode = 'filled' ;
t.box='off';
t.foreground=5;
t.fill_mode='on';
t.background=7;
a=get('current_axes');
a.margins = [0,0,0,0];
xset('window',0); xset('wdim', 500, 150);
```

Anexo 26 – Roteiro do Motor DC – Parte 2

Análise de estabilidade por Nyquist e resposta em frequência

Qual será o comportamento deste sistema? Em que condições ele é estável?

Um dos possíveis métodos para realizar esta análise é através do gráfico de resposta ao degrau da função de transferência tensão-velocidade G_1 . Nele é possível observar como varia a velocidade angular após a aplicação de um degrau de tensão e se a velocidade se estabiliza após determinado tempo.

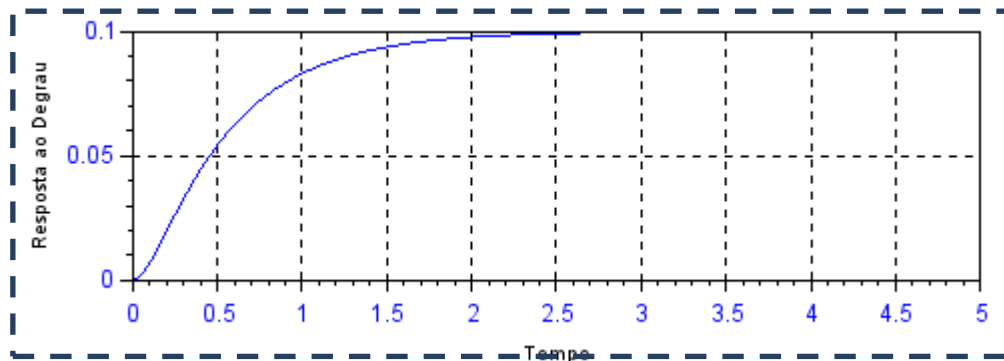
Então, dados os parâmetros de um motor de corrente contínua:

$$\begin{aligned} R_a &= \boxed{1} \, \Omega & J &= \boxed{0.01} \, \text{Kg.m}^2 \\ L_a &= \boxed{0.5} \, \text{H} & B &= \boxed{0.1} \, \text{Kg.m}^2/\text{s} \\ K_t = K_e &= \boxed{0.01} \, \text{Nm/A} \end{aligned}$$

Enviar

16 18

A resposta ao degrau será:



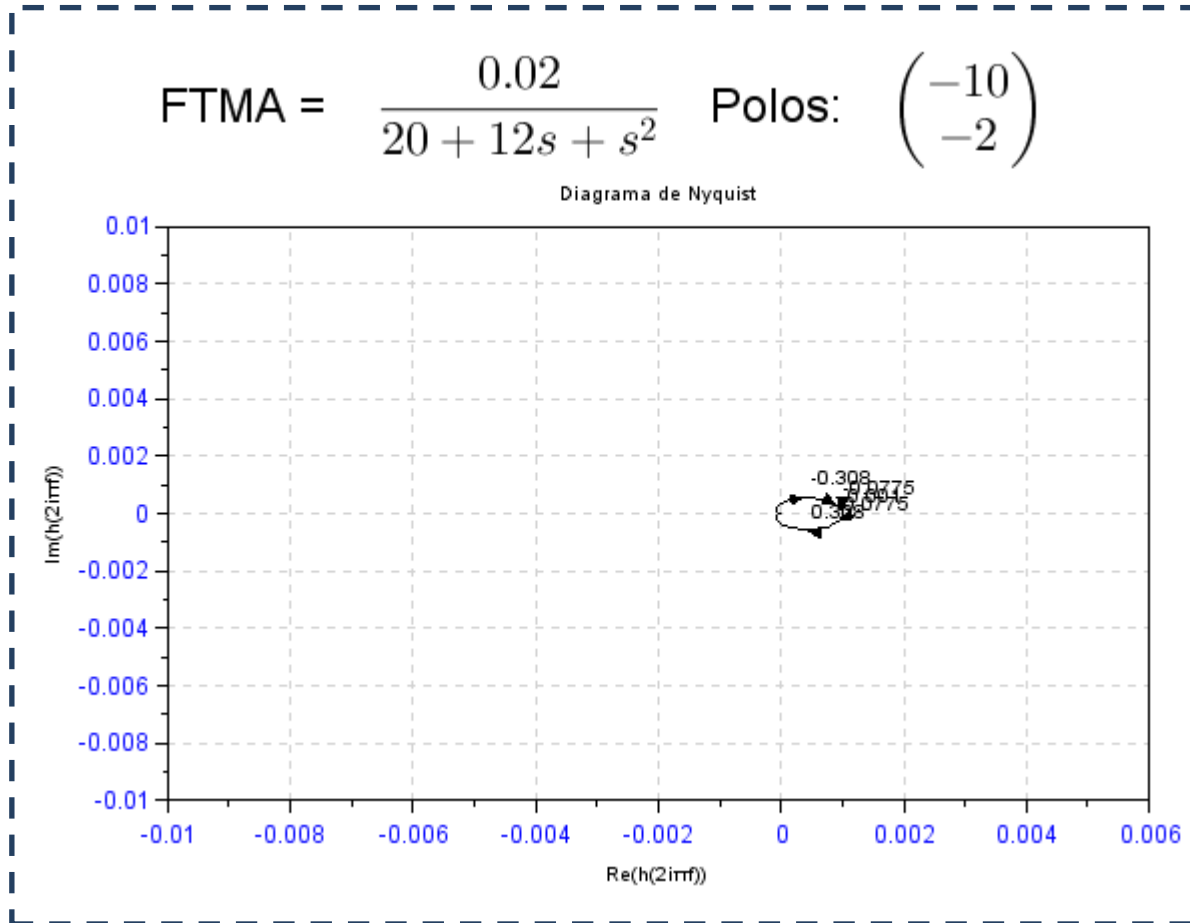
Código Scilab 16

A condição de estabilidade do sistema também pode ser analisada através do diagrama de Nyquist e da FTMA.

Para mais detalhes sobre o critério de estabilidade de Nyquist, clique na lupinha. 

Eixo real de a Eixo Imaginário de a

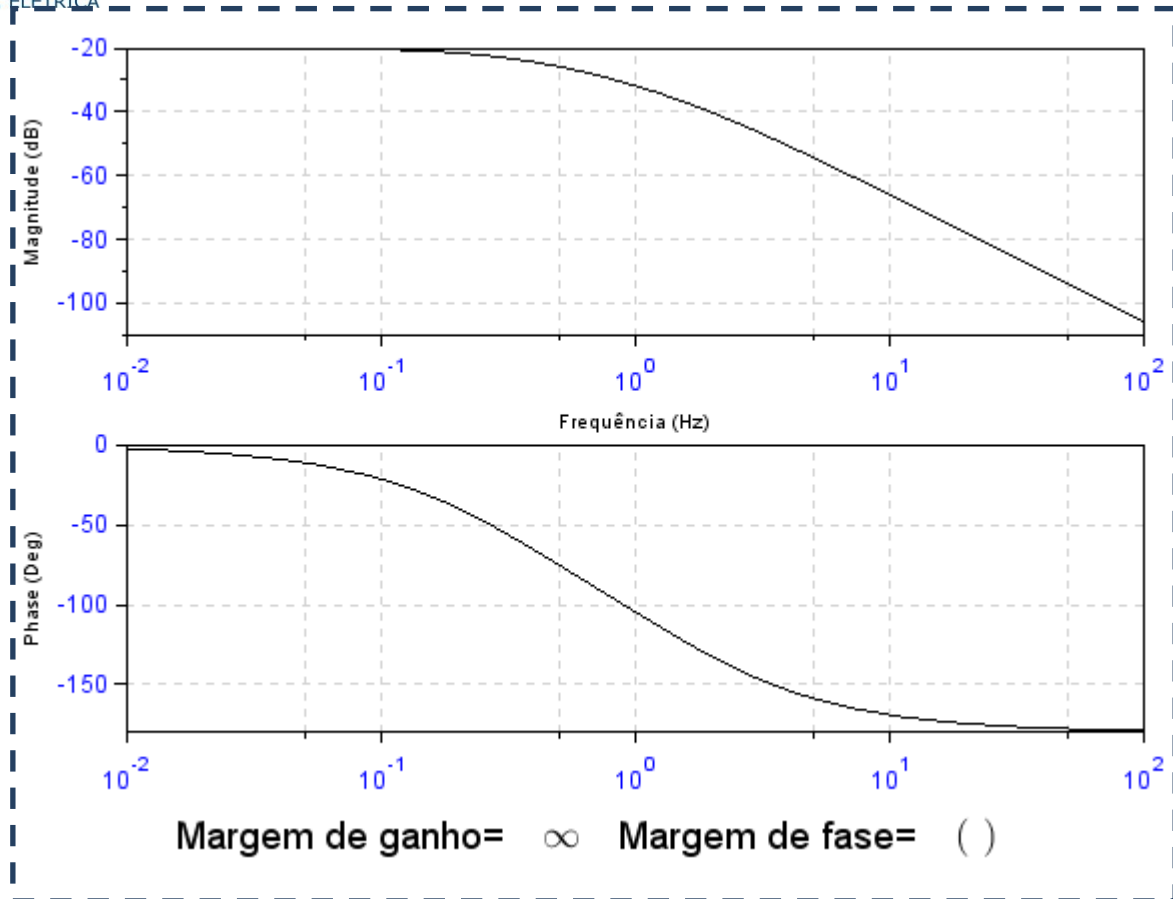
17



Código Scilab 17

Após verificar a estabilidade, pode-se ainda tirar conclusões sobre o sistema a partir da sua resposta em frequência, o diagrama de Bode.

A partir dele pode-se determinar, por exemplo, a Margem de Ganho que é uma medida de o quanto se pode incrementar ou decrementar o ganho do sistema para que ele alcance o ponto de estabilidade crítica.



Código Scilab 18

Note que a margem de ganho infinita significa que o sistema não pode ter a sua condição de estabilidade alterada pelo ganho.

Anexo 27 – Código Scilab 16

```
ra=1;
la=0.5;
kt=0.01;
j=0.01;
b=0.1;

tf=5;
s=poly(0,'s');
G1=(kt/(la*j))/((ra*j*s+b*ra+la*s*s*j+la*s*b+kt*kt)/(la*j));
t=linspace(0,tf,500);
imp_res=csim('step',t,G1);
plot(t,imp_res);xgrid();xtitle('','Tempo','Resposta ao Degrau');
xset('window',0); xset('wdim', 500, 250);
a=get('current_axes');
a.font_size=2;
a.labels_font_color=2;
a.box='on';
a.margins = [0.11,0.015,0.1,0.25];
```

Anexo 28 – Conteúdo da lupinha 6

Para que o sistema seja estável, ele não deve ter zeros da FTMF no SPAD. O critério de estabilidade de Nyquist diz que o número Z de zeros no SPAD pode ser calculado através de

$$Z = P - N$$

Onde:

$P \rightarrow$ Número de polos da FTMF no SPAD (que é igual ao número de polos da FTMA no SPAD)

$N \rightarrow$ Número de voltas no sentido horário que o diagrama de Nyquist dá em torno do ponto -1 .

Anexo 29 – Código Scilab 17

```

ra=1;
la=0.5;
kt=0.01;
j=0.01;
b=0.1;

re=-0.01;
rb=0.005;
ia=-0.01;
ib=0.01;

s=poly(0,'s');
Ga=(kt*kt)/((ra*j*s+b*ra+la*s*s*j+la*s*b));
c=coeff(Ga.den);
num=(kt*kt)/c(3);
den=(ra*j*s+b*ra+la*s*s*j+la*s*b)/c(3);
G1=(num)/(den);
p=roots(den);
subplot(1,1,1);
H=syslin('c',(G1));
nyquist(H);
a=get('current_axes');
a.font_size=2;
a.labels_font_color=2;
a.box='on';
a.data_bounds=[re,ia,-1;rb,ib,1];
a.margins = [0.12,0.04,0.25,0.1];
subplot(4,1,1);
str = prettyprint(G1);
str2 = 'FTMA = ';
str4 = prettyprint(p);
str5 = 'Polos: ';
str3=[str2,str,str5,str4];
xstring(0, 0, str3);
t=get('hdl');
t.font_foreground=1;
t.font_size=5;
t.alignment = 'left' ;
t.font_angle=0 ;
t.text_box = [1,1] ;
t.text_box_mode = 'filled' ;
t.box='off';
t.foreground=5;
t.fill_mode='on';
t.background=7;
a=get('current_axes');
a.margins = [0.1,0.1,0.05,0.1];
xset('window',0);
xset('wdim', 600, 500);

```

Anexo 30 – Código Scilab 18

```

ra=1;
la=0.5;
kt=0.01;
j=0.01;
b=0.1;

s=poly(0,'s');
G1=(kt)/((ra*j*s+b*ra+la*s*s*j+la*s*b+kt*kt));
H=syslin('c',(G1));
xset('window',0); xset('wdim', 600, 500);
subplot(2,1,1);
gainplot(H,0.01,100);
a=get('current_axes');
a.font_size=2;
a.labels_font_color=[2,2];
a.box='on';
a.margins = [0.1,0.03,0.1,0.25];
subplot(2,1,2);
phaseplot(H,0.01,100);
a=get('current_axes');
a.font_size=2;
a.labels_font_color=[2,2];
a.box='on';
a.margins = [0.1,0.03,0,0.35];
subplot(9,1,9);
str = prettyprint(g_margin(H));
str2 = 'Margem de ganho=';
str4=prettyprint(p_margin(H));
str5= '    Margem de fase=';
str3=[str2,str,str5,str4];
xstring(0, 0, str3);
t=get('hdl');
t.font_foreground=1;
t.font_size=5;
t.alignment = 'left' ;
t.font_angle=0 ;
t.text_box = [1,1] ;
t.text_box_mode = 'filled' ;
t.box='off';
t.foreground=5;
t.fill_mode='on';
t.background=7;
a=get('current_axes');

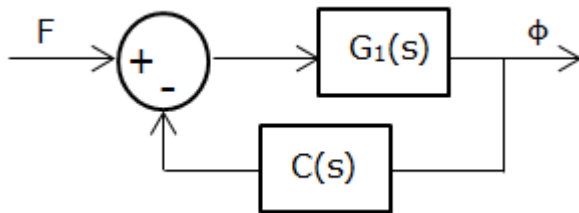
```

Anexo 31 - Roteiro do Motor DC – Parte 3

Ajuste por controle PID

O sistema já é estável, mas ainda assim, pode-se usar um controlador PID para ajustar as características como o tempo que o sistema demora a entrar em regime permanente e o Overshoot.

O que se deseja fazer é adicionar um controlador $C(s)$ na malha de realimentação, então o novo sistema seria:



Para mais detalhes sobre o controlador PID, clique na lupinha.

Podemos então tentar encontrar valores de K_p , K_I e K_D que deixam o sistema com as características desejadas, podemos analisar o efeito do controlador no sistema através da resposta ao degrau.

Então, dados os parâmetros de um motor de corrente contínua:

$$\begin{aligned} R_a &= \boxed{1} \, \Omega & J &= \boxed{0.01} \, \text{Kg.m}^2 \\ L_a &= \boxed{0.5} \, \text{H} & B &= \boxed{0.1} \, \text{Kg.m}^2/\text{s} \\ K_t = K_e &= \boxed{0.01} \, \text{Nm/A} \end{aligned}$$

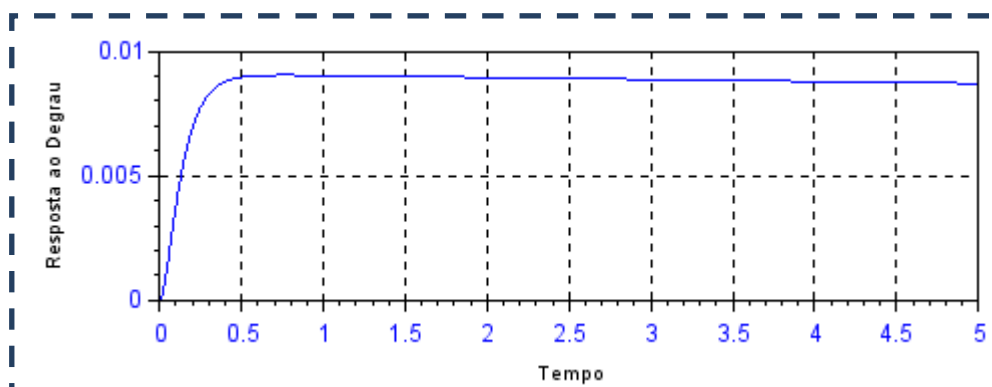
E do controlador PID:

$$K_p = \boxed{100} \quad K_I = \boxed{1} \quad K_D = \boxed{10}$$

Enviar

19

A resposta ao degrau será:



Código Scilab 19

Anexo 32 – Conteúdo da lupinha 7

O controlador PID é composto por três componentes: Um termo proporcional K_P , um integral K_I e um derivativo K_D .

Sua função de transferência tem então a seguinte forma:

$$K_P + \frac{K_I}{s} + K_D s$$

Cada um dos termos causa diferentes efeitos no sistema:

$K_P \rightarrow$ Reduz o tempo de subida, também reduz o erro com a referência (porém sem nunca eliminá-lo) e aumenta o Overshoot.

$K_I \rightarrow$ Tem a capacidade de eliminar o erro com a referência, porém, a resposta transiente será afetada. O tempo de assentamento e o overshoot aumentam.

$K_D \rightarrow$ Aumenta a estabilidade do sistema, reduzindo o overshoot e o tempo de assentamento.

Anexo 33 – Código Scilab 19

```
ra=1;
la=0.5;
kt=0.01;
j=0.01;
b=0.1;
kp=100;
ki=1;
kd=10;

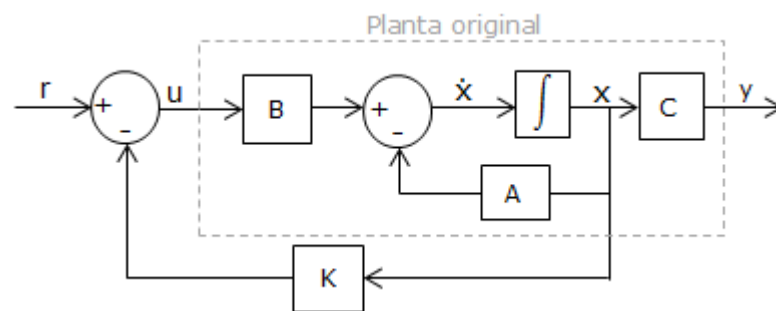
tf=5;
s=poly(0,'s');
G=(kt/(la*j))/((ra*j*s+b*ra+la*s*s*j+la*s*b+kt*kt)/(la*j));
C=(kd*s^2+kp*s+ki)/s;
G2=G/(1+G*C);
t=linspace(0,5,500);
imp_res=csim('step',t,G2);
plot(t,imp_res);xgrid();xtitle('','Tempo','Resposta ao Degrau');
xset('window',0); xset('wdim', 500, 250);
a=get('current_axes');
a.font_size=2;
a.labels_font_color=2;
a.box='on';
a.margins = [0.14,0.015,0.1,0.25];
```


Anexo 34 – Roteiro do Motor DC – Parte 4

Controle por realimentação de estados

Outra forma de controlar o sistema é através da realimentação de estados, que consiste em realimentar todos os estados do sistema, multiplicados por uma matriz de ganhos K em vez de usar a saída do sistema como realimentação. Isto permite reposicionar os polos do sistema para alterar suas características.

A imagem a seguir representa o diagrama de blocos da planta original com o controle por realimentação de estados.



Como visto anteriormente, dados os parâmetros do sistema do motor DC:

$$\begin{aligned} R_a &= \frac{1}{0.5} \Omega & J &= \frac{0.01}{0.1} \text{ Kg.m}^2 \\ L_a &= \frac{0.5}{0.1} \text{ H} & B &= \frac{0.1}{0.1} \text{ Kg.m}^2/\text{s} \\ K_t = K_e &= \frac{0.01}{0.1} \text{ Nm/A} \end{aligned}$$

Enviar

20 21 22

O espaço de estados do sistema sem controle torna-se:

$$\begin{cases} \dot{X}(t) = \begin{pmatrix} -10 & 1 \\ -0.02 & -2 \end{pmatrix} X(t) + \begin{pmatrix} 0 \\ 2 \end{pmatrix} U(t) \\ Y(t) = (1 \ 0) X(t) \end{cases}$$

Código Scilab 20

A posição inicial dos polos são os autovalores da matriz A . Eles podem ser verificados:

Polos:

$$P1 = -9.997$$

$$P2 = -2.003$$

Código Scilab 21

Através da realimentação, é possível obter um novo sistema em que os polos de malha fechada podem ser reposicionados arbitrariamente se o par (A, B) for controlável.

O sistema é controlável se o posto da matriz $A_{n \times n}$ for igual a n , neste caso, n precisa ser igual 2.

Matriz de Controlabilidade:
Posto = 2

$$\begin{pmatrix} 0 & 2 \\ 2 & -4 \end{pmatrix}$$

Código Scilab 22

Após verificar a controlabilidade, podemos então determinar a nova posição desejada para os polos:

P1 = P2 =

Enviar

23 24

E então, através da fórmula de Ackerman, calcular o valor de K que realiza o ajuste necessário:

K1 = -7.01

K2 = 1.5

Código Scilab 23

Finalmente, podemos aplicar o ganho K na malha de realimentação e chegar ao sistema controlado. O sistema original era da forma:

$$\dot{x} = A \cdot x + B \cdot u$$

$$y = C \cdot x$$

O sistema controlado será da forma:

$$\dot{x} = (A - B \cdot k) \cdot x + B \cdot u$$

$$y = C \cdot x$$

Então, podemos verificar a nova posição dos polos do Sistema realimentado através dos autovalores de $(A-B.k)$:

Polos:

P1 = -12

P2 = -3

Código Scilab 24

Anexo 35 – Código Scilab 20

```
ra=1;
la=0.5;
kt=0.01;
j=0.01;
b=0.1;

A = [-b/j  kt/j;
      -kt/la -ra/la];
B = [0;
      1/la];
C = [1  0];
D = 0;
ss=syslin('c',A,B,C);
str = prettyprint(ss);
xstring(0, 0, str);
t=get('hdl');
t.font_foreground=1;
t.font_size=5;
t.alignment = 'left' ;
t.font_angle=0 ;
t.text_box = [1,1] ;
t.text_box_mode = 'filled' ;
t.box='off';
t.foreground=5;
t.fill_mode='on';
t.background=7;
a=get('current_axes');
a.margins = [0,0,0,0];
xset('window',0); xset('wdim', 500, 150);
```

Anexo 36 – Código Scilab 21

```

ra=1;
la=0.5;
kt=0.01;
j=0.01;
b=0.1;

A = [-b/j  kt/j;
      -kt/la -ra/la];
B = [0;
      1/la];
C = [1  0];
D = 0;

ss=syslin('c',A,B,C);
po=spec(A);
format(6);
subplot(3,1,1);
str2 = 'Polos: ';
xstring(0, 0, str2);
t=get('hdl');
t.font_foreground=1;
t.font_size=3.8;
t.alignment = 'left' ;
t.font_angle=0 ;
t.text_box = [1,1] ;
t.text_box_mode = 'off' ;
t.box='off';
t.foreground=5;
t.fill_mode='on';
t.background=7;
a=get('current_axes');
a.margins = [0,0,0,0];
subplot(3,1,2);
str = prettyprint(po(1));
str2 = 'P1 =';
str3=[str2,str];
xstring(0, 0, str3);
t=get('hdl');
t.font_foreground=1;
t.font_size=3.8;
t.alignment = 'left' ;
t.font_angle=0 ;
t.text_box = [1,1] ;
t.text_box_mode = 'off' ;
t.box='off';
t.foreground=5;
t.fill_mode='on';
t.background=7;
a=get('current_axes');
a.margins = [0,0,0,0];
subplot(3,1,3);

```



```
str = prettyprint(po(2));  
str2 = 'P2 =';  
str3=[str2,str];  
xstring(0, 0, str3);  
t=get('hdl');  
t.font_foreground=1;  
t.font_size=3.8;  
t.alignment = 'left' ;  
t.font_angle=0 ;  
t.text_box = [1,1] ;  
t.text_box_mode = 'off' ;  
t.box='off';  
t.foreground=5;  
t.fill_mode='on';  
t.background=7;  
a=get('current_axes');  
a.margins = [0,0,0,0];  
xset('wdim', 250, 130);
```

Anexo 37 – Código Scilab 22

```

ra=1;
la=0.5;
kt=0.01;
j=0.01;
b=0.1;

A = [-b/j  kt/j;
      -kt/la -ra/la];
B = [0;
      1/la];
C = [1  0];
D = 0;
format(6);
ss=syslin('c',A,B,C);
cont=cont_mat(ss);
posto=rank(cont);
xset('wdim', 350, 120);
str = prettyprint(cont);
str2 = 'Matriz de Controlabilidade: ';
str3=[str2,str];
xstring(0, 0, str3);
t=get('hdl');
t.font_foreground=1;
t.font_size=3.8;
t.alignment = 'left' ;
t.font_angle=0 ;
t.text_box = [1,1] ;
t.text_box_mode = 'off' ;
t.box='off';
t.foreground=5;
t.fill_mode='on';
t.background=7;
a=get('current_axes');
a.margins = [0,0,0,0];
str = prettyprint(posto);
str2 = 'Posto =';
str3=[str2,str];
xstring(0, 0, str3);
t=get('hdl');
t.font_foreground=1;
t.font_size=3.8;
t.alignment = 'left' ;
t.font_angle=0 ;
t.text_box = [1,1] ;
t.text_box_mode = 'off' ;
t.box='off';
t.foreground=5;
t.fill_mode='on';
t.background=7;

```

Anexo 38 – Código Scilab 23

```

p1=-12;
p2=-3;

ra=1;
la=0.5;
kt=0.01;
j=0.01;
b=0.1;

A = [-b/j  kt/j;
      -kt/la -ra/la];
B = [0;
      1/la];
C = [1  0];
D = 0;
ss=syslin('c',A,B,C);
p=[p1 p2];
function [K] = acker_coef(A,B,alpha);
sa = size(A);
n = sa(1);
Com = cont_mat(A,B);
Comi = inv(Com);
alpha = [alpha,1];
alphaC = alpha(1) * eye(n,n);
for i = 2:n+1;
alphaC = alphaC+alpha(i)*A^(i-1);
end;
K = [zeros(1,n-1),1]*Comi*alphaC;
endfunction;
function [K] = acker(A,B,p);
n = length(p);
s = poly(0,'s');
equation = 1;
for i = 1:n;
equation = equation*(s-p(i));
end;
alpha = coeff(equation);
K = acker_coef(A,B,alpha);
endfunction;
ac=acker(A,B,p);
subplot(2,1,1);
str = prettyprint(ac(1));
str2 = 'K1 =';
str3=[str2,str];
xstring(0, 0, str3);
t=get('hdl');
t.font_foreground=1;
t.font_size=3.8;
t.alignment = 'left' ;
t.font_angle=0 ;
t.text_box = [1,1] ;

```

```
t.text_box_mode = 'off' ;  
t.box='off';  
t.foreground=5;  
t.fill_mode='on';  
t.background=7;  
a=get('current_axes');  
a.margins = [0,0,0,0];  
subplot(2,1,2);  
str = prettyprint(ac(2));  
str2 = 'K2 =';  
str3=[str2,str];  
xstring(0, 0, str3);  
t=get('hdl');  
t.font_foreground=1;  
t.font_size=3.8;  
t.alignment = 'left' ;  
t.font_angle=0 ;  
t.text_box = [1,1] ;  
t.text_box_mode = 'off' ;  
t.box='off';  
t.foreground=5;  
t.fill_mode='on';  
t.background=7;  
a=get('current_axes');  
a.margins = [0,0,0,0];  
xset('wdim', 250, 120);
```


Anexo 39 – Código Scilab 24

```

p1=-12;
p2=-3;

ra=1;
la=0.5;
kt=0.01;
j=0.01;
b=0.1;

A = [-b/j  kt/j;
      -kt/la -ra/la];
B = [0;
      1/la];
C = [1  0];
D = 0;

ss=syslin('c',A,B,C);
p=[p1 p2];
function [K] = acker_coef(A,B,alpha);
sa = size(A);
n = sa(1);
Com = cont_mat(A,B);
Comi = inv(Com);
alpha = [alpha,1];
alphaC = alpha(1) * eye(n,n);
for i = 2:n+1;
alphaC = alphaC + alpha(i) * A^(i-1);
end;
K = [zeros(1,n-1),1]*Comi*alphaC;
endfunction;
function [K] = acker(A,B,p);
n = length(p);
s = poly(0,'s');
equation = 1;
for i = 1:n;
equation = equation*(s-p(i));
end;
alpha = coeff(equation);
K = acker_coef(A,B,alpha);
endfunction;
ac=acker(A,B,p);
k=[ac(1) ac(2)];
abk=A-B*k;
po=spec(abk);
subplot(3,1,1);
str2 = 'Polos: ';
xstring(0, 0, str2);
t=get('hdl');
t.font_foreground=1;
t.font_size=3.8;
t.alignment = 'left' ;

```

```
t.font_angle=0 ;
t.text_box = [1,1] ;
t.text_box_mode = 'off' ;
t.box='off';
t.foreground=5;
t.fill_mode='on';
t.background=7;
a=get('current_axes');
a.margins = [0,0,0,0];
subplot(3,1,2);
str = prettyprint(po(1));
str2 = 'P1 =';
str3=[str2,str];
xstring(0, 0, str3);
t=get('hdl');
t.font_foreground=1;
t.font_size=3.8;
t.alignment = 'left' ;
t.font_angle=0 ;
t.text_box = [1,1] ;
t.text_box_mode = 'off' ;
t.box='off';
t.foreground=5;
t.fill_mode='on';
t.background=7;
a=get('current_axes');
a.margins = [0,0,0,0];
subplot(3,1,3);
str = prettyprint(po(2));
str2 = 'P2 =';
str3=[str2,str];
xstring(0, 0, str3);
t=get('hdl');
t.font_foreground=1;
t.font_size=3.8;
t.alignment = 'left' ;
t.font_angle=0 ;
t.text_box = [1,1] ;
t.text_box_mode = 'off' ;
t.box='off';
t.foreground=5;
t.fill_mode='on';
t.background=7;
a=get('current_axes');
a.margins = [0,0,0,0];
xset('wdim', 250, 120);
```