

# ESTA008-17 – Sistemas de Controle II

2º Quadrimestre de 2019

Aula 20 – Aula Prática 5

Implementação do compensadores por avanço e por atraso de fase com auxílio do MATLAB via resposta em frequência pelo método analítico – Projeto de compensador por atraso-avanço de fase com auxílio do MATLAB via resposta em frequência pelo método de tentativa-e-erro, método do Ogata e método analítico

Profs. André Luís da Silva e Magno Enrique Mendoza Meza

**Aviso:** O relatório deverá ser entregue na data indicada, pelo professor, após realizada a implementação com todas as contas determinadas, os itens documentados, e as observações, discussões e comentários conceituais realizados. **NÃO SERÃO ACEITOS RELATÓRIOS FEITOS A MÃO, EXCETO QUANDO INDICADO NO ROTEIRO.**

## 1 Objetivo

O objetivo desta aula é realizar a implementação dos compensadores por atraso-avanço de fase para o servomecanismo Quanser, utilizando técnicas no domínio da frequência. Será utilizado o servomecanismo Quanser e todo o procedimento será assistido por um técnico de laboratório.

## 2 Introdução

É muito importante ter realizado as simulações com cuidado na última aula, para certificar-se que o procedimento experimental será factível, principalmente, no que diz respeito à amplitude de controle.

O trabalho consiste simplesmente em fornecer ao técnico ou ao professor o bloco Simulink projetado para cada compensador, junto dos respectivos parâmetros. Ele irá inserir os mesmos no diagrama Simulink conectado à placa Q4 via QuaRC, o qual realizará a compilação e gravação para controle digital.

Devem ser aplicadas as entradas onda quadrada e onda triangular simuladas na aula prática 2. Prestar atenção para usar os mesmos tempos, frequências e amplitudes.

Seus dados de entrada de referência, saída e amplitude de controle devem ser salvos em arquivos de dados “.mat” e posteriormente usados para traçar os respectivos gráficos.

## 3 Procedimentos

Os procedimentos podem ser resumidos conforme seguem.

### 3.1 Compensador de Atraso-Avanço

1. Forneça ao técnico seu bloco Simulink contendo o compensador de atraso-avanço;
2. Rode o script `parametros_atraso_avanco.m` no qual você salvou os parâmetros deste compensador;
3. Configure o diagrama Simulink de integração HIL (*hardware in the loop*) para aplicar uma entrada quadrada. A frequência é 0,2 Hz e a amplitude  $\pi/4$  [rad], tempo 10 segundos. Por características de implementação do QuaRC, o procedimento deve ser parado manualmente (botão “stop” do Simulink) próximo do término dos 10 segundos. Salve seus resultados de entrada de referência, saída lida e amplitude de controle num arquivo “.mat” para posteriormente gerar os gráficos com o MATLAB;
4. Configure o diagrama Simulink de integração HIL (*hardware in the loop*) para aplicar uma entrada triangular. A frequência é 0,2 Hz e a amplitude  $\pi/4$  [rad], tempo 10 segundos. Por características de implementação do QuaRC, o procedimento deve ser parado manualmente próximo do término dos 10 segundos. Salve seus resultados de entrada de referência, saída lida e amplitude de controle num arquivo “.mat” para posteriormente gerar os gráficos com o MATLAB.

## 4 Relatório

Insira no relatório uma descrição da montagem de hardware, conforme sintetizado no documento `Guia_Experimentos` fornecido com o guia desta aula. Consulte o manual do fabricante para substanciar sua explanação. Seja breve e sintético, relute em somente traduzir todo o volume de informação lançado pelo fabricante.

Apresente os gráficos de referência, saída e controle coletados experimentalmente para cada controlador e cada tipo de referência. Sempre indique os parâmetros do compensador e os parâmetros da referência.

Compare seus resultados reais obtidos nesta aula com os dados correlatos de simulação determinados na aula prática 2. Indique as discrepâncias e semelhanças. Substancie sua avaliação com análises técnicas e conceituais sobre os fatos apurados.

## 5 Observações para a implementação dos compensadores

### 5.1 Plotagem dos dados

As variáveis que armazenam os dados de simulação e os dados reais da planta são: (i) `data_pos`, a qual armazena os dados do sinal de referência e da posição do eixo do motor; (ii) `data_vm`, a qual armazena o esforço de controle aplicado no motor.

O conteúdo de `data_pos` é como segue

`data_pos(:,1)` contém os dados dos instantes de tempo;  
`data_pos(:,2)` contém os dados do sinal de referência;  
`data_pos(:,3)` contém os dados da posição angular do eixo do motor;

e o conteúdo de `data_vm` é

`data_vm(:,1)` contém os dados dos instantes de tempo;  
`data_vm(:,2)` contém os dados do esforço de controle aplicado ao motor;

Para plotar os dados armazenados nas variáveis `data_pos` e `data_vm` recomenda-se fortemente utilizar os scripts 2 e 3 do Anexo A.

## 5.2 Execução para implementação dos compensadores no Simulink

Recomenda-se utilizar o MATLAB 2009 para realizar todas as aulas práticas, pois outras versões podem gerar problemas de incompatibilidade nos arquivos do Simulink. A ordem da execução dos scripts do MATLAB para a implementação é a seguinte:

1° Executar script `setup_srv02_exp02_pos.m`;

2° Executar script

- `parametros_atraso_avanco.m` para um compensador por atraso-avanço de fase, script 1 no Anexo A.

3° Abrir arquivo

- `q_srv02_pos_lag_lead.mdl` para um compensador por atraso-avanço de fase.

4° Configuração DAQ: Duplo click no bloco de inicialização HIL no subsistema SRV02-ET, arquivo

- `q_srv02_pos_lag_lead.mdl` para um compensador por atraso-avanço de fase.

nos quais está dentro do subsistema SRV02-ET **Position**, e certifique-se de que está configurado para o dispositivo DAQ que está instalado no seu sistema;

5° Clicar no **QuaRC | Build** para compilar o diagrama do Simulink. Espere alguns instante ate a compilação finalizar;

6° Se o passo anterior executou sem problema, então selecione **QuaRC | Start** para começar a executar o controlador. Os escopos devem exibir as respectivas respostas;

7° Quando uma resposta adequada é obtida, clique no botão **Stop** na barra de ferramentas do diagrama de Simulink, ou selecione do menu **QuaRC | Stop**, para parar a execução do código;

8° Se não aconteceu erro nenhum, execute o arquivo `plot_datas.m`. Este arquivo exporta as figuras para o formato PNG com os nomes desejados para os arquivos, por exemplo, `figura_step.npg` e `figura_step_control.npg`, respectivamente;

9° Na linha de comandos do MATLAB digitar `save identificacao_grupo_onda_yyyy_xxxx`, no qual se deve substituir `yyyy` por quadrada ou triangular, e `xxxx` por avanço, atraso, atraso-avanço. O comando `save` armazena todas as variáveis do Workspace do MATLAB no respectivo arquivo `identificacao_grupo_onda_yyyy_xxxx.mat`, o qual pode ser carregado no Workspace do MATLAB com o comando `load identificacao_grupo_onda_yyyy_xxxx`.

## A Scripts úteis para aula prática

SCRIPT MATLAB 1: Programa para definir os parâmetros do compensador por avanço de fase, arquivo `parametros_atraso_avanco.m`.

```
% Modelo nominal (engrenagem alta, carga disco)
K=1.53; % [rad/V.s]
tau=0.0254; % [s]
```

```
% Parametros do compensador
Kv=;
Kc=;
alpha=;
T1=;
beta=;
T2=;
```

SCRIPT MATLAB 2: Programa para plotar os dados de referência, posição angular e esforço de controle, arquivo `plot_datas.m`

```
% (c) - Magno Enrique Mendoza Meza
figure(1)
plot(data_pos(:,1),180/pi*data_pos(:,2),'r',data_pos(:,1),180/pi*data_pos(:,3),...
    'b');
grid;
title('Sinal de referência e Posição do Eixo');
xlabel('Tempo [s]');
ylabel('\theta [graus]')
legend('Ref.','\theta','Location','NorthWest');
size_font;
print('-f1','figura_step','-dpng'); % Exporta a figure(1) para formato PNG com ...
    o nome 'figura_step'

figure(2)
plot(data_vm(:,1),data_vm(:,2),'b');
grid;
title('Esforço de controle');
xlabel('Tempo [s]');
ylabel('Tensão [V]')
size_font;
print('-f2','figura_step_control','-dpng'); % Exporta a figure(2) para formato ...
    PNG com o nome 'figura_step_control'
```

SCRIPT MATLAB 3: Programa para definir tamanho da figura, tamanho da fonte e espessura das curvas, arquivo `size_font.m`.

```
% (c) - Magno Enrique Mendoza Meza
% Estabelece a posição da figura, a largura e comprimento
% set(gcf, 'units','normalized','outerposition',[0 0 1 1]);
set(gcf, 'Units','centimeters', 'Position',[2 3 56 42]/2.5)
% Estabelece tamanho da fonte do título e etiqueta dos eixos
set(findall(gcf,'Type','text'),'FontSize',15)
% Estabelece espessura linha e cor dos dados
set(findall(gcf,'Type','line'),'LineWidth',3)
% Estabelece tamanho fonte etiqueta eixos, cor e espessura linhas
set(findall(gcf,'Type','axes'),'FontSize',14,'LineWidth',3,'XColor',...
    'black','YColor','black')
```