

ECA414 – Sistemas de Controle I

Experiência 10 – Identificação de Sistemas

Nome: _____ R.A.: _____

Nome: _____ R.A.: _____

Nome: _____ R.A.: _____

1 Introdução

Uma forma bastante útil de extrair a função de transferência de um sistema é utilizando algoritmos de identificação de sistemas. Ambientes de software como Matlab ou LabVIEW™ disponibilizam algoritmos que identificam as funções de transferência a partir do estímulo e da resposta do sistema.

No LabVIEW™ 2017 é possível percorrer o manual sobre função de identificação existentes em **C:\Program Files (x86)\National Instruments\LabVIEW 2017\manuals** e encontrar o arquivo **Slreference.pdf** que contém os cálculos realizados nos diversos algoritmos de identificação. Na Figura 1 há uma amostra do manual.

Figura 1 – Imagens do manual de identificação de Sistemas do LabVIEW™

Algorithm References

The coefficients $a(k)$ of an AR model can be computed by following the Levinson recursive algorithm, where

$$\hat{a}_m(k) = a_{m-1}(k) + \hat{k}_m a_{m-1}(m-k), k=1, \dots, m-1 \quad (2)$$
$$\hat{a}_m(m) = \hat{k}_m \quad (3)$$
$$\hat{\rho}_m = (1 - \hat{k}_m^2) \rho_{m-1} \quad (4)$$
$$\hat{k}_m = \frac{-2 \sum_{i=1}^{m-1} \hat{e}_{m-1}^*(i) \hat{e}_{m-1}^*(i-1)}{\sum_{i=1}^{m-1} \hat{e}_{m-1}^*(i)^2 + \sum_{i=1}^{m-1} \hat{e}_{m-1}^*(i-1)^2}, m=1, 2, \dots, n_a \quad (5)$$
$$\begin{aligned} \hat{e}_m^f(n) &= \hat{e}_{m-1}^f(n) - \hat{k}_m \hat{e}_{m-1}^f(n-1) \\ \hat{e}_m^b(n) &= \hat{e}_{m-1}^b(n) - \hat{k}_m \hat{e}_{m-1}^b(n) \\ \hat{e}_m^*(n) &= \hat{e}_m^f(n) = x(n), m=1, \dots, n_a \end{aligned} \quad (6)$$

5. References

[1] Guanghu Hu, *Digital Signal Processing*, Tsing Hua University Press.

si_Est ARX Model (SISO).vi

1. Introduction

This section includes the algorithms and references used by the si_Est ARX Model (SISO).vi, which is the core VI to estimate the coefficients of an ARX (SISO) model using the input-output data of a system. The si_Est ARX Model (SISO).vi is located in the labview\vi.lib\addons\System Identification\Parametric Estimation\Subs\lib directory.

2. Function Description

The ARX (SISO) model is described as

$$A(q)y(n) = B(q)u(n-k) + e(n) \quad (1)$$

where $A(q) = 1 + \sum_{i=1}^N a_i q^{-i}$, $B(q) = \sum_{i=0}^{N-1} b_i q^{-i}$, k is the delay of the system, and $u(n)$, $y(n)$, and $e(n)$ are the input, output, and disturbance of a system, respectively. q is the backward shift operator, which means

$$q^{-1}y(n) = y(n-1)$$

The purpose of this VI is to estimate the coefficients $[a_1, a_2, \dots, a_N]$ and $[b_0, b_1, \dots, b_{N-1}]$ using the input-output data of a system.

Algorithm References

3. APIs

Para Name	In Out	Type	Description
stimulus signal	In	1D DBL Array	The input signal of a system
response signal	In	1D DBL Array	The output signal of a system
model parameters	In	Cluster containing A order, B order, and delay; A order, B order, and delay are n_a , n_b , and k , respectively	
ARX model	Out	Cluster	-
std deviations	Out	Cluster	The standard deviations of A coefficients and B coefficients
noise variance	Out	DBL	Variance of $e(n)$
covariance matrix	Out	2D DBL Array	-
noises	Out	1D DBL Array	$e(n)$

4. Algorithms

The least square solution for the ARX model estimation is (Equation 7.34, Raf 1):

$$\hat{\theta}_{LS}^T = \left[\frac{1}{N} \sum_{t=1}^N \phi(t) \phi^T(t) \right]^{-1} \frac{1}{N} \sum_{t=1}^N \phi(t) y(t) \quad (2)$$

where

$$\phi(t) = [-y(t-1) \dots -y(t-N_a) \quad u(t) \quad u(t-1) \dots u(t-N_b+1)]^T$$

You can rewrite Equation 2 as the solution of the linear equations:

$$AX = Y \quad (3)$$

where

$$A = \begin{bmatrix} \phi^T(p) \\ \phi^T(p+1) \\ \vdots \\ \phi^T(N) \end{bmatrix}, X = \begin{bmatrix} a_1 \\ a_2 \\ b_0 \\ \vdots \\ b_{N_b-1} \end{bmatrix}, \text{ and } Y = \begin{bmatrix} y(p) \\ y(p+1) \\ \vdots \\ y(N) \end{bmatrix}$$

5. References

2 Identificando a Função de Transferência

Utilizaremos uma das funções de identificação para extrair a função de transferência do motor CC, utilizando o kit Quanser Cube. Para isso ligue o kit (instruções em atividade anterior) e abra o código **Identificação_Motor.vi**.

Figura 2 – Painel Frontal do código Identificação_Motor.vi

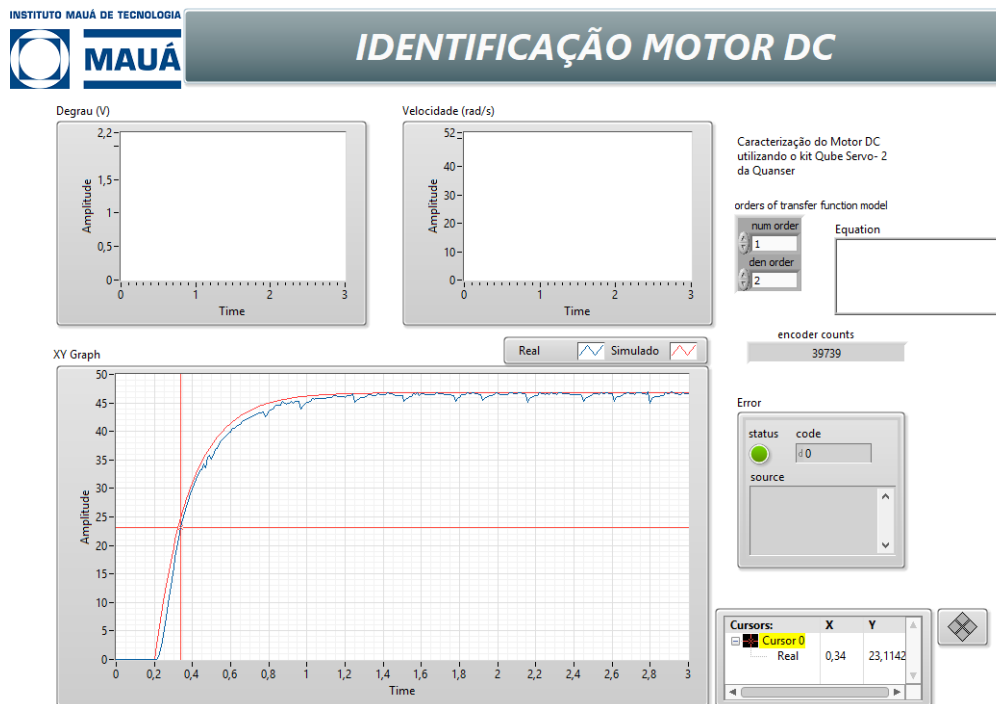
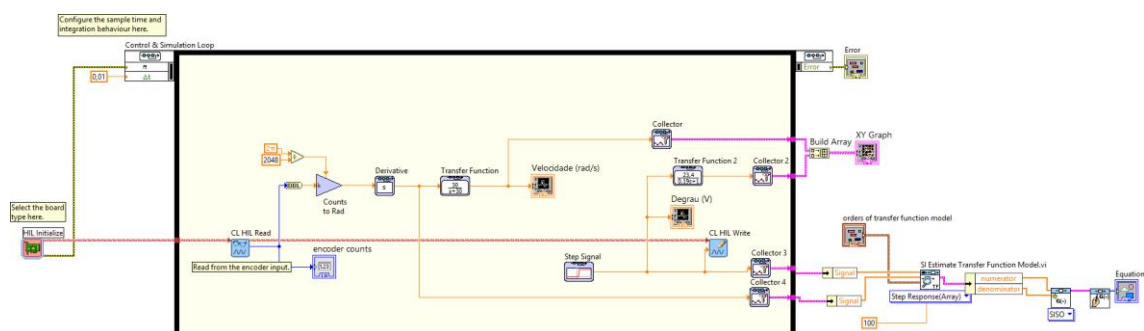


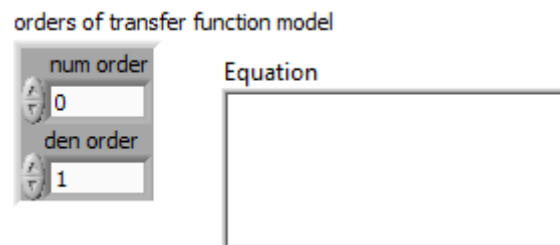
Figura 3 – Diagrama de Blocos do código Identificação_Motor.vi



Este código estimula o sistema com uma entrada do tipo degrau e após o término do *loop* insere os vetores do estímulo e da resposta no algoritmo de identificação. Será possível observar a função de transferência no indicador *Equation*.

Como o sistema que analisa a Velocidade em função da tensão aplicada é considerado de primeira ordem (conforme atividades anteriores), altere os parâmetros de numerador e denominador para 0 e 1, respectivamente, conforme a Figura 4.

Figura 4 – Parâmetro e Resposta da Identificação

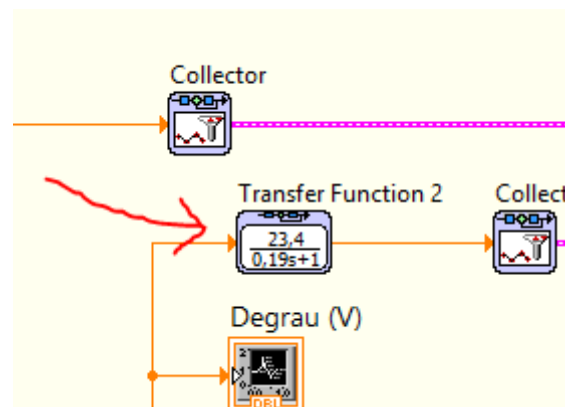


Execute o código e anote a função de transferência indicada:

$F(S) =$

Em seguida substitua a **função de simulação** pela função de transferência encontrada e execute novamente a simulação. Basta dar um duplo clique na função e alterar os valores.

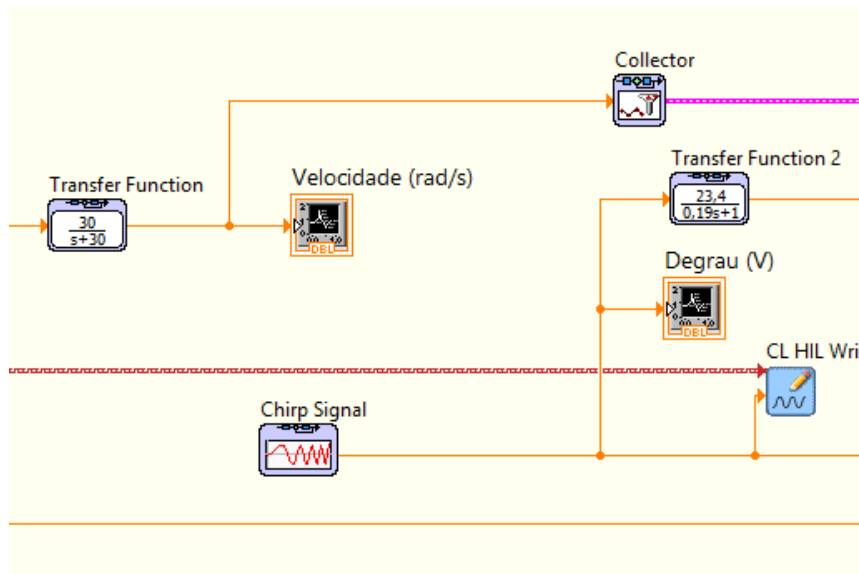
Figura 5 – Substituição da função de transferência simulada



Execute o código novamente e registre o gráfico obtido.

Troque sinal de entrada do sistema (degrau) por uma função do tipo *Chirp Signal*.

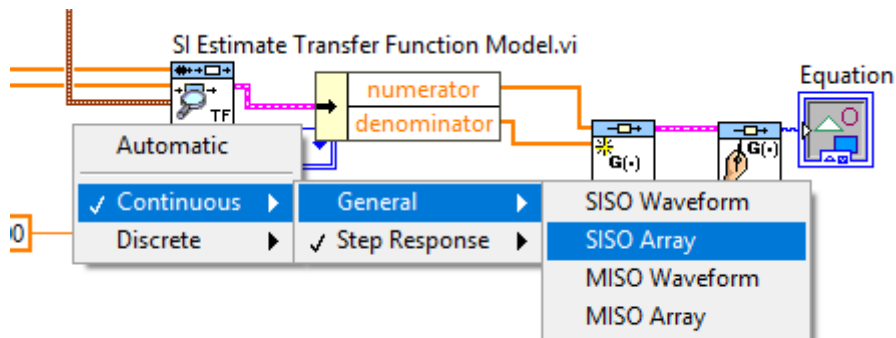
Figura 6 – Adicionando a função *Chirp Signal*



Execute novamente o código.

Provavelmente não será possível identificar a função de transferência, visto que esta está configurada apenas para entrada do tipo degrau. Altere as opções da função para Continuous/General/SISO Array, conforme a Figura 7.

Figura 7 – Alterando as propriedades da função de identificação



Execute o código novamente e analise a resposta.

Anote a conclusões sobre os ensaios realizados.

3 Identificando a Função de Transferência em Sistema de Posição

Abra o código **Controle_PD_exercício.vi** estudado em atividade anterior. Este código implementa um controlador proporcional derivativo em relação à posição do eixo/disco acoplado ao motor CC.

Figura 8 – Painel Frontal do código **Controle_PD_exercício.vi**

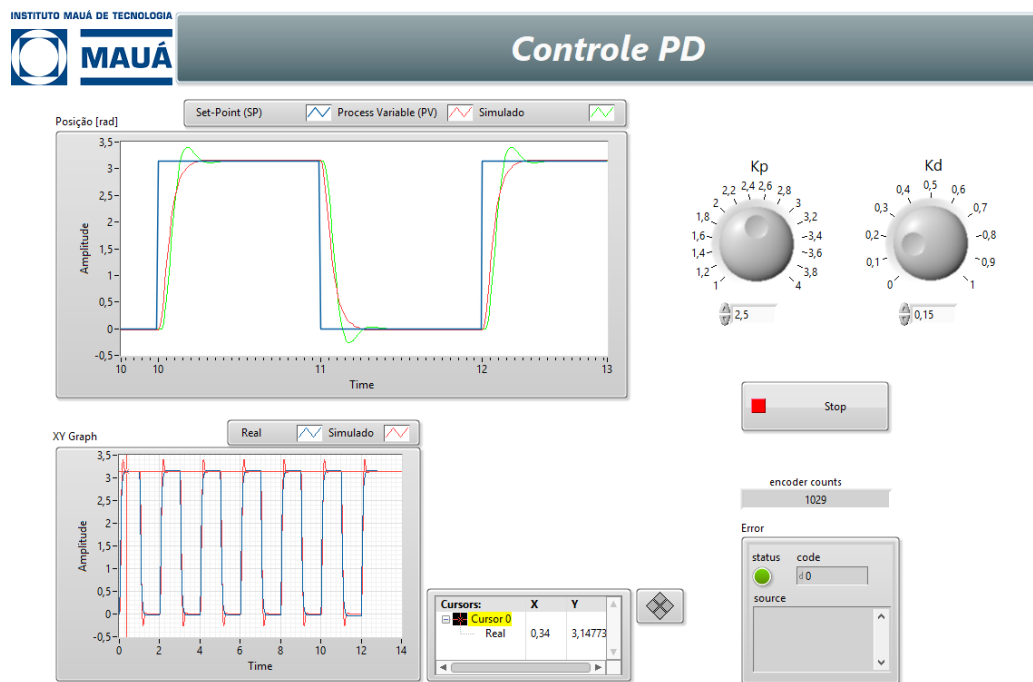
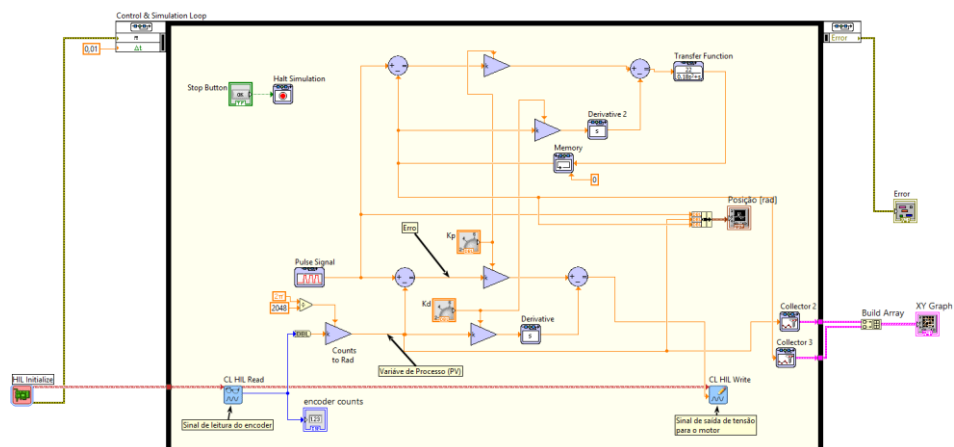


Figura 9 – Diagrama de blocos do código **Controle_PD_exercício.vi**



Copie a parte do código de identificação utilizado no item 2 para este código com o intuito de identificar a função de transferência para este sistema de segunda ordem.

Para não retirar as funções do controlador coloque valores nos ganhos K_p e K_d de modo que o controlador não interfira no sistema. Anote os valores.

$K_p =$

$K_d =$

Em seguida ajuste a ordem do sistema que se deseja obter e execute o código.

Analise os resultados e anote a função obtida.

$G(S) =$

Deduza a função de transferência a partir da função obtida no exercício 2. Compare com a função obtida.

Anote as análises e conclusões sobre o experimento.