

IDENTIFICAÇÃO E MODELAGEM DA PLATAFORMA ROBÓTICA DIDÁTICA

GEORGE ACIOLI JÚNIOR, ALEQUINE B. DE LIMA, PÉRICLES R. BARROS

*Laboratório de Instrumentação Eletrônica e Controle,
Departamento de Engenharia Elétrica, Universidade Federal de Campina Grande
Rua Aprígio Veloso, 822 – Bodocongó, Campina Grande-PB, 58429-900*

E-mails: alequine.lima@ee.ufcg.edu.br , georgeacioli@dee.ufcg.edu.br ,
prbarros@dee.ufcg.edu.br

Abstract— This article has as main objective to show the modelling and implementation of a didactic robotic platform, aiming the application of this one on experiments related to system identification. Therefore, there has been studied two modelling techniques of robots, geometrically and algebraically, allowing the implementation of a program to collect data and to excite the system. The utilization of motor-servos with appropriated sensors and the communication system embedded was also a differential, allowing that major focus was given to the identification process. To the data collect, it was used the OPC (Open Platform Communications), in addition to MATLAB, where were also fulfilled the identification process, generating a quantity of acceptable functions composed of up three poles, resulting in the selection of the best functions at the end of ten essays.

Keywords— identification, robotic, modelling

Resumo— Este artigo tem como objetivo apresentar a modelagem e implementação de uma plataforma robótica didática, visando a aplicação desta em experimentos relacionados à identificação de sistemas. Para tanto, foram estudadas duas técnicas de modelagem de robôs, geométrica e algébrica, permitindo a implementação de um programa para coleta de dados e excitação do sistema. A utilização de servos motores com sensores apropriados e sistema de comunicação próprios fora também um diferencial, permitindo que maior enfoque fosse dado ao processo de identificação. Para a coleta de dados, utilizou-se OPC (Open Platform Communications), além do MATLAB, onde fora também realizado o processo de identificação, gerando uma quantidade de funções aceitáveis compostas de até três polos, resultando na escolha das melhores funções ao fim dos ensaios.

Palavras-chave— identificação, robótica, modelagem

1 Introdução

Na última década, observou-se grande crescimento no desenvolvimento de novas tecnologias e no aperfeiçoamento de outras menos recentes. A robótica, por sua vez figura nesse âmbito, como exemplo de tecnologia futurista e que mesmo com tanto desenvolvimento não parece ter atingindo sua maturidade. Embora existam robôs atualmente capazes de realizarem salvamentos, ou que completem de forma relativamente autônoma, ainda não temos sistemas robóticos inteiramente independentes e suficientemente precisos para realizar certas atividades.

De modo a estimular o desenvolvimento de atividades com robôs e procedimentos de manufatura, o adentrando com mais dedicação no campo da robótica o Laboratório de Instrumentação Eletrônica e Controle (LIEC), da UFCG adquiriu um manipulador robótico da série “AX-12A Smart Robotic Arm”, montado e distribuído pela CrustCrawler (www.crustcrawler.com/products/AX-18F Smart Robotic Arm/docs/AX12A18A Smart Robotic Arm Specifications.pdf)

Neste trabalho tem-se como intuito apresentar uma plataforma didática voltada para o aprendizado da robótica, utilizando-se de abordagens de modelagem para o manipulador, aproveitando-se dos conceitos de robótica apresentados no último século e, consequentemente, aprimorar o comportamento deste robô, para que a movimentação satisfaça a realização de tarefas um pouco mais elaboradas do que as observadas até então.

Na seção 2, são apresentadas as condições utilizadas para o prosseguimento com a modelagem. Em seguida são apresentados os métodos para a modelagem matemática do braço robótico, que foram a convenção de Denavit-Hartenberg (1955, Denavit & Hartenberg) e a abordagem geométrica. Estes conceitos foram utilizados segundo a abordagem de Vidyasagar (2005, Vidyasagar), que construiu didaticamente o que fora exposto por Denavit-Hartenberg. Além disso, um trabalho semelhante realizado por Laura Griggs (2014, Griggs. L & Fahimi. F) fora utilizado como referência na implementação dos parâmetros de Denavit-Hartenberg. Na seção 3, são apresentados os equipamentos utilizados e em seguida, na seção 4 é apresentado o programa elaborado para auxílio na aquisição de dados dos servos-motores e criação e arquivos compatíveis com o MATLAB®, utilizando de uma abordagem semelhante àquela utilizada por Mensink (2008, Mensink). Posteriormente, na seção 5 apresentam-se os resultados e uma análise, que precede as sugestões para trabalhos futuros.

2 Modelagem da plataforma robótica

Nesta seção, apresenta-se o desenvolvimento do modelo cinemático do robô, por duas vias distintas: A modelagem geométrica e a modelagem matricial.

O manipulador “Smart Robotic Arm” possui as dimensões como na tabela 1, associadas de cima para baixo, com os eixos da figura 1:

Tabela 1 Dimensões do manipulador:

L_{12}	L_{23}	L_{34}	L_{45}
8.57 cm	17.2 cm	6.43 cm	4.2 cm

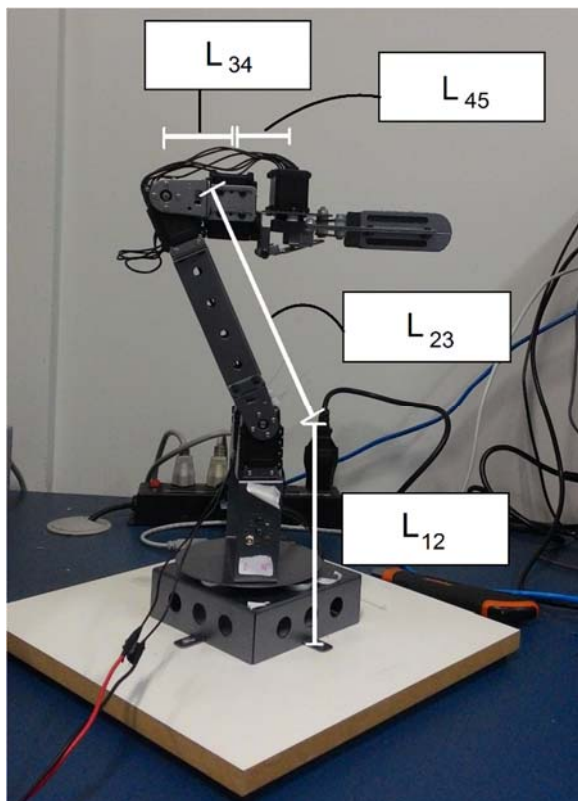


Figura 1: Manipulador utilizado no desenvolvimento da plataforma didática.

Para a modelagem de um manipulador robótico, convém analisar o que seria a cinemática direta e a inversa. A cinemática direta é um problema que envolve as relações entre as juntas individuais e o posicionamento orientação do “ponto de interesse”. Os ângulos entre as juntas e o comprimento das juntas são as variáveis em questão.

A cinemática inversa é o problema oposto, onde não temos os dados dos ângulos e desejamos adquiri-los a partir dos demais dados.

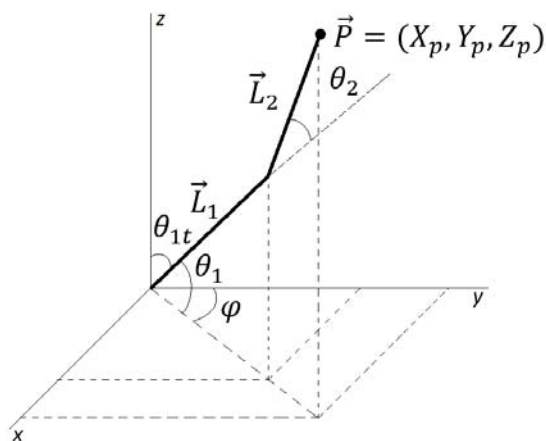


Figura 2: Representação do mecanismo robótico no espaço tridimensional e seus ângulos

Para que fosse possível o desenvolvimento do estudo adequado para o mecanismo, resolveu-se estabelecer alguns critérios e considerações:

- As juntas do robô são unicamente de revolução, isto é não possuem deslocamento de deslizamento.
- Adota-se o seguinte modelo via coordenadas esféricas, considerando o eixo ombro e o eixo base como sendo únicos de modo a facilitar o entendimento e o desenvolvimento da matemática envolvida. Não há alteração no modelo, já que o giro do eixo base produz uma rotação idêntica no eixo ombro.
- O eixo ombro também é a referência (0,0,0).

2.1 Cinemática direta (abordagem geométrica)

Através da figura 2, é possível observar que decomposição de $\vec{L}_1$ no plano xy e no eixo z, nos levará as seguintes equações:

$$\vec{L}_1 = \sin \varphi \cdot \cos \theta_1 L_1 \hat{a}_x + \cos \varphi \cdot \cos \theta_1 L_1 \hat{a}_y + \sin \theta_1 L_1 \hat{a}_z \quad (1)$$

E também, de forma análoga a L1, temos para L2:

$$\vec{L}_2 = \sin \varphi \cdot \cos(\theta_1 + \theta_2) L_2 \hat{a}_x + \cos \varphi \cdot \cos(\theta_1 + \theta_2) L_2 \hat{a}_y + \sin(\theta_1 + \theta_2) L_2 \hat{a}_z \quad (2)$$

Seja então $\vec{P} = \vec{L}_1 + \vec{L}_2$, onde

$$\vec{P} = X_p \hat{a}_x + Y_p \hat{a}_y + Z_p \hat{a}_z \quad (3)$$

Das soma entre as equações (1) e (2), obtemos P, e temos, igualando termo a termo com a equação (3):

$$X_p = \sin \varphi \cdot (\cos \theta_1 L_1 + \cos(\theta_1 + \theta_2) L_2) \quad (4)$$

$$Y_p = \cos \varphi \cdot (\cos \theta_1 L_1 + \cos(\theta_1 + \theta_2) L_2) \quad (5)$$

$$Z_p = L_1 \sin(\theta_1) + \sin(\theta_1 + \theta_2) L_2 \quad (6)$$

Dessa forma é possível obter o posicionamento cartesiano a partir das leituras dos ângulos θ_1, θ_2 e φ cedidas pelos servos-motores AX-12A e conhecimento das dimensões $\vec{L}_1$ e $\vec{L}_2$.

2.2 Cinemática inversa (abordagem geométrica)

Por sua vez, é interessante que, dadas as coordenadas do ponto de interesse, o robô consiga calcular os ângulos associados para cada eixo.

Através da lei dos cossenos, é possível determinar o ângulo θ_{1t} , como visto na figura 2:

$$\theta_{1t} = \arccos \left(\frac{Y^2 + Z^2 + X^2 + L_1^2 - L_2^2}{2L_1 \sqrt{Y^2 + Z^2 + X^2}} \right)$$

Uma vez determinado θ_{1t} , e sabendo que:

$$\theta_1 = \arctan \frac{\sqrt{Y^2 + X^2}}{Z} - \arccos \left(\frac{Y^2 + Z^2 + X^2 + L_1^2 - L_2^2}{2L_1 \sqrt{Y^2 + Z^2 + X^2}} \right) \quad (8)$$

Onde θ_1 é o ângulo que deve ser adicionado à posição atual do eixo ombro.

Analogamente para o ângulo θ_2 , e fazendo-se uso do ângulo auxiliar θ_{2t} da figura 3, temos:

$$Y^2 + Z^2 + X^2 = L_1^2 + L_2^2 + 2L_1L_2 \cos \theta_2$$

$$\theta_2 = \arccos \left(\frac{Y^2 + Z^2 + X^2 - L_1^2 - L_2^2}{2L_1L_2} \right)$$

Onde θ_2 é o ângulo que deve ser adicionado à posição atual do eixo cotovelo.

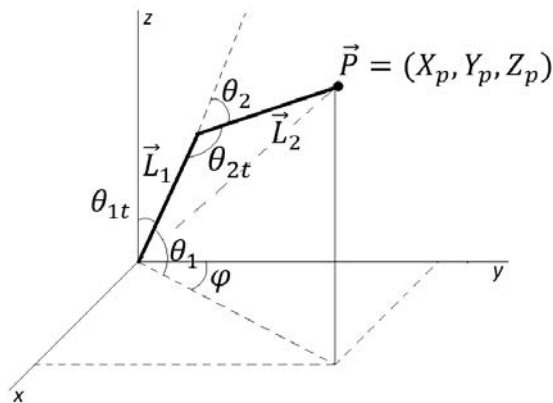


Figura 3: Representação gráfica para a obtenção da cinemática inversa do manipulador

Para a determinação do ângulo a ser estabelecido no eixo base, obtemos:

$$\varphi = \arctan \frac{X}{Y} \quad (11)$$

Entretanto, ainda há a existência de outra possibilidade, tal que θ_1 assume maior valor, mas θ_2 assume valor negativo, levando à mesma posição de interesse. Obtemos então:

$$\theta_1 = \arctan \frac{\sqrt{Y^2 + X^2}}{Z}$$

$$\mp \arccos \left(\frac{Y^2 + Z^2 + X^2 + L_1^2 - L_2^2}{2L_1\sqrt{Y^2 + X^2}} \right) \quad (12)$$

$$\theta_2 = \pm \arccos \left(\frac{Y^2 + Z^2 + X^2 - L_1^2 - L_2^2}{2L_1L_2} \right) \quad (13)$$

$$\varphi = \arctan \frac{X}{Y} \quad (14)$$

Outra opção é a utilização da convenção de Denavit-Hartenberg para a construção de um modelo, mais focado no algebrismo que na geometria do sistema.

2.3 Cinemática direta (Denavit-Hartenberg)

A convenção mais utilizada para a modelagem adequada de manipuladores é a convenção proposta por Denavit-Hartenberg (1955, Denavit & Hartenberg), que define a matriz A_i que associa os sistemas coordenados, de modo que:

$$A_i = \begin{bmatrix} c_{\theta_i} & -s_{\theta_i}c_{\alpha_i} & s_{\theta_i}s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i}c_{\alpha_i} & -c_{\theta_i}s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (15)$$

Os parâmetros α (torção da junta em relação à anterior), θ (giro de uma junta em relação a outra), d (deslocamento de uma junta em relação à anterior) e a (comprimento da haste em relação à anterior) são obtidos a partir da observação do equipamento.

Como mostrado pelo livro de Vidyasagar (2005, Vidyasagar), e de acordo com a matemática produzida por J. Denavit e R.S. Hartenberg (1955, Denavit & Hartenberg), para cada caso temos uma abordagem diferente.

Ao final das sucessivas operações matriciais é possível obter as coordenadas do ponto de interesse, construindo o que chamamos de “cinemática direta” do manipulador.

Até o momento, optou-se por não explorar a cinemática inversa utilizando as matrizes DH devido a problemas que podem acontecer na inversão de matrizes, e sim o método geométrico.

Em Vidyasagar (2005, Vidyasagar), é possível encontrar uma lista como passos para a modelagem correta e didática.

Desse modo, a convenção de Denavit-Hartenberg define as equações de cinemática direta para o manipulador, mapeando as variáveis de cada junção e para o posicionamento e orientação ponto de interesse. Segundo o próprio Vidyasagar, a decomposição das matrizes DH para a solução do problema de cinemática inversa em manipuladores com juntas de revolução é dispendiosa e exagerada para um manipulador de pequena ordem, sendo então deixada de lado para manipuladores mais complexos e com juntas esféricas.

Assim como sugerido por Laura Griggs (2014, Griggs. L & Fahimi. F) que utilizou o mesmo dispositivo para aplicações na medicina, construiu-se uma tabela com os dados do manipulador em questão, obtemos:

Tabela 2 Parâmetros de D-H do mecanismo

Junta/Par.	θ_i	α_i	a_i	d_i
Junta 1	θ_1	$\pi/2$	2.22 cm	0
Junta 2	θ_2	0	6.855 cm	0
Junta 3	θ_3	$-\pi/2$	0	8.25 cm
Junta 4	θ_4	0	0	0

E o modelo através de sistemas coordenados, como na figura 4.

3 Plataforma robótica didática

Para cada junta, o braço mecânico possui um servo motor produzido pela empresa Robotis (www.robotis.com). O servo-motor AX-12a é apresentado como sendo um motor DC inteligente, que nos permite realizar o controle até certo ponto sem a necessidade de um controle externo. Além de leves, são possíveis de utilizar em rede, isto é conectados mas mesmo assim operantes de forma independente, devido a utilização de ID's – números individuais de identificação de cada servo – que podem ser de 0 até 253. A precisão das informações fornecidas e recebidas pelo servo são através de 10 bits, permitindo uma boa margem para o tratamento de erro.

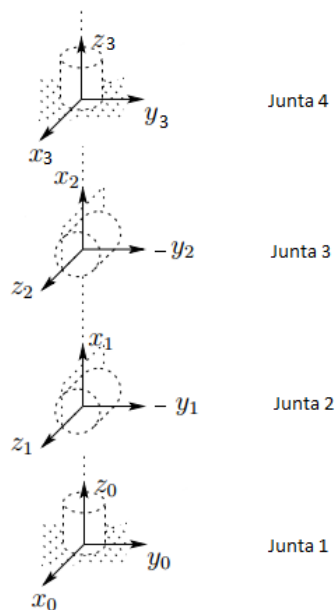


Figura 4: Representação das juntas do mecanismo

3.1 Informações Técnicas

- Resolução - 0.35°
- Ângulo de Operação - 300°
- Tensão de Trabalho - 7V~10V (Recomenda-se 9.6V)
- Máxima Corrente Suportada - 900mA
- Temperatura de Operação - 5 °C negativos até +85 °C
- Sinal de Comando – Pacotes de Dados
- Protocolo - Half duplex Asynchronous Serial Communication (8bit, 1stop, No Parity)
- Conexão - TTL (Em cadeia)
- ID - (0 à 253), e BROADCAST ID 254
- Velocidade de Comunicação - 7343bps à 1 Mbps
- Dados Disponíveis - Posição, Temperatura, Carga, Tensão de Entrada, etc.

3.2 Arquitetura dos servos

Para a comunicação dos servos, é observado a utilização de apenas um fio para os pacotes enviados e recebidos para os servos-motores, fato este que pode nos trazer problemas devido as “colisões” entre pacotes. Para a utilização adequada do servo, é interessante conhecer as principais funções por ele fornecidas, como ID, BaudRate, GoalPosition, Present Position, que fornecem respectivamente a identidade do servo, a taxa de transmissão de bits, a posição destino e a posição atual. Estas vão desde funcionalidades triviais, até detalhes de configuração.

É possível utilizar as instruções abaixo como forma de comandar os motores, junto com os pacotes de instrução. A fabricante, ROBOTIS, disponibiliza toda uma API – conjunto de funções que permite realisar a interface computador/dispositivo – de modo a

simplificar o processo de controle dos motores, individualmente ou em conjunto.

Tabela 1 Especificações das funções na utilização da memória

Instrução	Função	Valor	Param.
PING	Não possui ação. Utilizado apenas para verificar existência de ID	0x01	0
READ	Leitura de valores em diferentes posições de memória	0x02	2
WRITE	Escrita de valores	0x03	2~
REG WRITE	Escreve valores mas espera ACTION	0x04	2~
ACTION	Autoriza execução	0x05	0
RESET	Atribui os valores de fábrica	0x06	0
SYNC WRITE	Utilizado para controle de múltiplos servos ao mesmo tempo	0x83	4~

3.3 Comunicação dos servos

Na tentativa de facilitar a comunicação diretamente com o computador, adotamos o uso de um dispositivo, dito USB2DYNAMIXEL. Através dele, é possível controlar os motores AX-12a utilizando programação em C, C++ ou C# e sem a necessidade de um microcontrolador.

Dada entrada particular dos motores AX-12a, no formato TTL, este é visto como uma alternativa prática de possibilitar o controle através do computador. Além da possibilidade de conversão entre USB e TTL, temos também a opção de utilizar como conversor entre RS485, RS232 e USB.

Dessa forma é possível associar os motores e os utilizar apenas com o Visual Studio 2012 instalado e o driver apropriado.

4 Programa

As funções disponibilizadas pela API foram utilizadas na construção de uma base sólida que permitisse realizar ensaios periódicos afim de realizar a identificação e a coleta de dados. Como forma de utilizar os motores AX-12a no braço robótico, foram criadas classes em C#, que preservassem as funcionalidades dos motores e permitissem a operação em conjunto. Funcionalidades associadas a duas posições de memória foram tratadas através de funções com terminações “word”.

A função “dxl_set_txpacket_instruction()” também foi muito utilizada, em conjunto com as especificações “INST_REG_WRITE” e “INST_ACTION” que possibilitam a operação simultânea dos motores.

Após a criação da classe eixo, foi de interesse isolar cada eixo dentro de suas condições físicas, tais como “agarrar”, “girar verticalmente”. Dessa forma, foram criadas as classes “Base, Ombro, Cotovelo, Punho e Mão”, deixando a forma de programar mais intuitiva.

Para a Base, foram atribuídas funcionalidades tais como ângulo, velocidade e limites de posicionamento, visando unidades mais práticas, como “Graus”. Praticamente todas as funcionalidades da classe eixo foram reaproveitadas. Para as demais Ombro, Cotovelo, Punho, pouco se modifica, exceto pelo nome das propriedades, onde temos a presença de “ângulo_antebraço” ou ainda “ângulo_braço”. Para a Mão, especificou-se a propriedade “fechada”, resultando no fechamento ou não da garra, diferentemente das outras classes, que dependem do valor em graus para operar.

Para a execução dos ensaios, foi elaborado um programa visando aprimorar e tornar mais eficaz a utilização dos servos, que aqui foram considerados como constituintes de cada eixo. De modo geral, o programa construído é separado em 3 vertentes de trabalho.

Figura 5 Aba "ensaio", responsável pela aquisição de dados

Na primeira, intitulada “Ensaio”, como visto na figura 5, é possível selecionar qual eixo excitar utilizando uma onda quadrada. Além disso, é gerado um arquivo de saída, que pode ser exportado como “.m”, para o MATLAB, ou como “.txt”.

Na segunda vertente, Posicionamento Cartesiano, como visto na figura 6, temos a implementação das equações de modelagem do robô, permitindo posicionar o braço robótico nas posições desejadas de acordo com coordenadas cartesianas no que diz respeito as juntas e ao ponto de interesse. É notável também a presença das opções por trajeto, uma vez que é possível obter o mesmo posicionamento para o ponto de interesse utilizando trajetos diferentes.

Figura 6 Aba responsável pelo posicionamento simplificado do robô no espaço

Temos ainda os botões Liberar e Capturar que permitem, respectivamente: Liberar os motores do torque a ser imprimido, e capturar a posição atual do ponto de interesse, em coordenadas cartesianas, com os respectivos ângulos do motor de punho e do motor da mão robótica.

Figura 7 Aba responsável pela funcionalidade OPC do programa

Por sua vez, a opção OPC, na terceira vertente de mesmo nome, na figura 7, permite utilizar o servidor OPC instalado no computador para gerar diferentes sinais, como senóide, onda quadrada, degrau, RPBS, etc. Por fim, temos a opção de gerar sequências desejadas. Com o auxílio do botão liberar, posicionamos o robô na posição de interesse e salvamos em um arquivo “.pos”. E continuamos a adicionar diversas posições, até que o usuário ache suficiente. Desse modo, ao pressionar Executar, o robô reexecutará todos os passos salvos no arquivo.

5 Identificação

Por sua vez, nos ensaios realizados, utilizou-se um sinal de onda quadrada com 3 ciclos, para cada motor. Esse sinal oscilava entre $\pm 10\%$ em torno de uma posição central. Foram ciclos que mostraram uma resposta periódica e portanto não necessitou-se de outros ciclos. A amplitude escolhida era de modo a não atravessar zonas onde o servo-mecanismo aquece-se

além do desejado, uma vez que para maiores amplitudes de oscilação, o servo operaria de forma mais agressiva. O período, por sua vez, fora escolhido de modo a permitir a observação do fenômeno oscilatório completo. Estes ensaios, foram a princípio baseados nos ensaios promovidos por Arno Mensink (2008, Mensink) que utilizou-se de um sinal degrau.

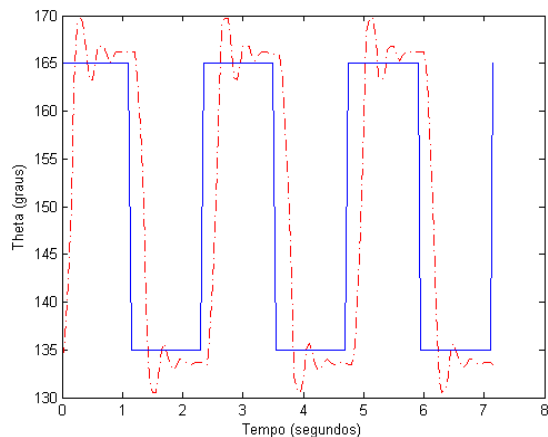


Figura 8 Gráfico correspondente à aquisição de dados do eixo ombro

Para o caso mais preocupante, o ombro, graficamente representado na figura 8, onde o sinal em azul representa uma referência imposta na forma de onda quadrada e o sinal em vermelho tracejado representa a devida resposta do atuador, devido as fortes oscilações, decidiu-se realizar uma sequência de experimentos, para a determinação de uma função de transferência que represente o comportamento.

$$H_o(s) = \frac{31.95}{s^2 + 7.072s + 31.93} \quad (16)$$

Obtiveram-se diversas funções, sendo a mais conveniente, para o eixo ombro, a função de transferência da equação 16, obtida a partir do uso da ferramenta "System Identification" do MATLAB, que fora configurado para inicializar utilizando N4SID seguido do método dos mínimos quadrados não linear.

Os ajustes obtidos foram respectivamente de 92.27%, 88.77%, 95.93%, 93.77%, 80.7%.

Para um segundo conjunto de valores inseridos e obtidos, experimentou-se utilizar a mesma função de transferência obtida e verificar se o ajuste era válido para o sinal. A validade fora atestada segundo o erro percentual entre o sinal de saída original e o sinal de saída após a passagem pela função de transferência obtida. Desta vez, obteve-se um ajuste de 85.1% para um sinal PRBS – sequência pseudoaleatória binária – e 84.99% para um sinal senoidal, mostrando a validade da função de transferência.

6 Conclusões e Trabalhos Futuros

Observa-se que com a construção do programa e com a devida modelagem do sistema robótico fora possível efetuar o processo de identificação e assim obter modelos que representaram o sistema de forma satisfatória. A extensão do sistema de identificação

para um sistema de identificação e controle poderá ser efetuada no futuro, inclusive adaptando à utilização de micro controladores. Além disso, existe toda uma base matemática preparada para que seja possível construir modelos virtuais e integra-los ao dispositivo mecânico, de modo a obter simulações e controle remoto.

Fora perceptível em todos os ensaios realizados a existência de um erro de regime permanente o que é um forte indicador da necessidade de um controlador que possua ação integral, intencionando remover este erro, o que será foco de um trabalho mais adiante. Este erro de regime é indesejado uma vez que, tratando-se de um manipulador, não será possível atingir as posições desejadas com precisão, ainda que a modelagem esteja correta.

Agradecimentos

Agradecimentos ao LIEC – UFCG pela estrutura concedida.

Agradecimentos à ATECEL por ter fomentando parte da pesquisa.

Referências Bibliográficas

- Denavit, Jacques e Hartenberg, Richard Scheunemann (1955). "A kinematic notation for lower-pair mechanisms based on matrices". *Trans ASME J. Appl. Mech* **23**: 215–221.
- Griggs, Lauren, e Farbod, Fahimi. "Introduction and Testing of an Alternative Control Approach for a Robotic Prosthetic Arm." *The Open Biomedical Engineering Journal* 8 (2014): 93–105.
- Mensink, Arno, Characterization and modeling of a dynamixel servo. Technical report, University of Twente, 2008. Individual research assignment.
- Spong, M. W. Hutshinson S. e Vidyasagar, M. Robot Modeling and Control. John Wiley & Sons, Ins., First Edition, 2005.
- [www.crustcrawler.com/products/AX-18F Smart Robotic Arm/docs/AX12A18A Smart Robotic Arm Specifications.pdf](http://www.crustcrawler.com/products/AX-18F_Smart_Robotic_Arm/docs/AX12A18A_Smart_Robotic_Arm_Specifications.pdf) – Acesso em 25 de agosto de 2016
- support.robotis.com/en/product/dynamixel/ax_series/dxl_ax_actuator.htm – Acesso em 25 de agosto de 2016
- www.robotis.com – Acesso em 25 de agosto de 2016