

CENTRO UNIVERSITÁRIO FEI
LEONARDO DA SILVA COSTA

**AQUISIÇÃO DOS MODELOS DINÂMICOS DE ROBÔS OMNIDIRECIONAIS
UTILIZANDO IDENTIFICAÇÃO DE SISTEMAS**

São Bernardo do Campo

2021

LEONARDO DA SILVA COSTA

**AQUISIÇÃO DOS MODELOS DINÂMICOS DE ROBÔS OMNIDIRECIONAIS
UTILIZANDO IDENTIFICAÇÃO DE SISTEMAS**

Relatório Final de Iniciação Científica apresentado
ao Centro Universitário FEI, como parte dos requi-
sitos do Programa PIBIC-FEI. Orientado pelo Prof.
Dr. Flavio Tonidandel

São Bernardo do Campo

2021

RESUMO

Este projeto propõe o desenvolvimento de um software capaz de adquirir os dados necessários para a geração dos modelos dinâmicos dos robôs da categoria *RoboCup-SSL*. Também são mostrados cinco tipos de representação utilizadas para a identificação de sistemas, sendo elas: *Autoregression with Extra Inputs* (ARX), *Autoregressive Moving Average with Exogenous Inputs* (ARMAX), *Output Error* (OE) e Box-Jenkins (BJ) e o modelo de Função de Transferência (FT). Após a conclusão dos testes foram escolhidos três modelos: ARX, ARMAX e FT. Utilizando os modelos obtidos é possível desenvolver sistemas de controle de forma mais rápida, pois, o processo de modelagem da planta leva um tempo menor do que utilizando a modelagem a partir das leis da física.

Palavras-chave: *Autoregression with Extra Inputs*. Futebol de robôs. *Autoregressive Moving Average with Exogenous Inputs*. *Output Error*. Box-Jenkins. *transfer function*. *Small-Size League*. *RoboCup*.

LISTA DE ILUSTRAÇÕES

Figura 1 – Robô utilizado pela equipe RoboFEI.	10
Figura 2 – Diagrama de blocos do sistema de controle de posição.	10
Figura 3 – Diagrama de blocos do software de aquisição de dados.	11
Figura 4 – Resultado do teste de linearidade para o eixo X.	13
Figura 5 – Resultado do teste de linearidade para o eixo Y.	13
Figura 6 – Resultado do teste de linearidade para o ângulo.	14
Figura 7 – TDF do sinal de entrada e saída em X.	14
Figura 8 – TDF do sinal de entrada e saída em Y.	14
Figura 9 – TDF do sinal de entrada e saída em θ	14
Figura 10 – Gerador de sinais PRBS	15
Figura 11 – PRBS gerado através do MATLAB	15
Figura 12 – Exemplo de sinal GBN.	16
Figura 13 – Descontinuidade de θ	25
Figura 14 – Variação contínua de θ	25
Figura 15 – nMRSE para modelos OE de ordem variada.	27
Figura 16 – Modelos obtidos antes da parametrização do sinal GBN.	28
Figura 17 – Modelos obtidos após parametrização do sinal GBN.	29
Figura 18 – Modelos obtidos antes da parametrização do sinal PRBS.	29
Figura 19 – Modelos obtidos após parametrização do sinal PRBS.	30
Figura 20 – Software de aquisição de dados.	31
Figura 21 – Interface de carregamento dos dados.	31
Figura 22 – Interface de identificação dos modelos.	32
Figura 23 – Início do Experimento #3.	33
Figura 24 – Comparação entre os modelos gerados para o eixo X utilizando o nMRSE.	34
Figura 25 – Comparação entre os modelos gerados para o eixo Y utilizando o nMRSE.	34
Figura 26 – Comparação entre os modelos gerados para o eixo θ utilizando o nMRSE.	35
Figura 27 – Comparação entre os modelos gerados utilizando o MRSE.	35
Figura 28 – Comparação entre os modelos gerados para o eixo X utilizando o nMRSE.	36
Figura 29 – Comparação entre os modelos gerados para o eixo Y utilizando o nMRSE.	36
Figura 30 – Comparação entre os modelos gerados para o eixo θ utilizando o nMRSE.	36
Figura 31 – Comparação entre os modelos gerados utilizando o MRSE.	37
Figura 32 – Predição de 20 passos à frente utilizando os dados do experimento um como entrada.	38
Figura 33 – Predição de 20 passos à frente utilizando os dados do experimento dois como entrada.	39
Figura 34 – Resposta ao degrau para cada eixo do robô.	40
Figura 35 – Simulação dos modelos de FT obtidos.	41

Figura 36 – Diagrama de blocos utilizado no projeto do controlador para o eixo X	42
Figura 37 – <i>PIDTuner</i>	42
Figura 38 – Diagrama de blocos finalizado para o eixo X	43
Figura 39 – Resposta ao degrau para os eixos do robô.	43

LISTA DE TABELAS

Tabela 1 – Linhas gerais de parametrização do sinal GBN.	17
--	----

SUMÁRIO

1	Introdução	8
1.1	Objetivo	8
1.2	Justificativa	9
1.3	Sistema Utilizado	9
2	Revisão Bibliográfica	12
2.1	Identificação de Sistemas	12
2.1.1	Introdução	12
2.1.2	Características do sistema	12
2.2	Formas de Perturbação de um Sistema	15
2.2.1	Pseudo Random Binary Signal (PRBS)	15
2.2.2	Generalized Binary Noise (GBN)	16
2.3	Representações Utilizadas para Identificação de Sistemas	17
2.3.1	Função de Transferência	17
2.3.2	Representações em Tempo Discreto	18
2.3.2.1	<i>AutoRegressive with eXogenous Inputs (ARX)</i>	19
2.3.2.2	<i>AutoRegressive Moving Average with eXogenous Inputs (ARMAX)</i>	20
2.3.2.3	<i>Output Error (OE)</i>	21
2.3.2.4	<i>Box-Jenkins (BJ)</i>	22
2.4	Trabalhos Correlatos	22
3	Metodologia	24
3.1	Normalização do Ângulo do Robô	25
4	Resultados Finais	27
4.1	Escolha do Formato dos Modelos	27
4.2	Parametrização dos Sinais	28
4.2.1	GBN	28
4.2.2	PRBS	29
4.3	Software de Aquisição de Dados	30
4.3.1	Aplicativo em MATLAB para Identificação dos Modelos	30
4.4	Modelos ARX e ARMAX	32
4.4.1	Experimentos Realizados	32
4.4.2	ARX	33
4.4.3	ARMAX	36
4.4.4	Comparação Entre os Modelos ARX e ARMAX Finais	37
4.5	Controlador de Posição Utilizando os Modelos de FT Estimados	39
4.5.1	Sinal de Entrada	40
4.5.2	Identificação das Funções de Transferência	40
4.5.3	Projeto do Controlador de Posição	41

4.5.4	Implementação do Controlador	42
4.5.5	Resultados	43
5	Conclusão	45
	APÊNDICE A – Modelo ARX	46
	APÊNDICE B – Modelo ARMAX	48
	REFERÊNCIAS	50

1 INTRODUÇÃO

A categoria *Small Size League* (SSL) do futebol de robôs da *RoboCup*¹ é uma das mais dinâmicas de toda a competição devido aos passes e chutes rápidos. A velocidade da bola pode chegar a até $6,5m/s$. Portanto, para ser capaz de executar e interceptar passes nesta velocidade, assim como realizar outras jogadas ágeis, é necessário dominar o controle da movimentação dos robôs.

Para implementar um bom controlador é necessário ter em mãos o modelo dinâmico do robô. A fim de criar este modelo é preciso conhecer as características físicas dos robôs e também os parâmetros do campo, a partir das quais a modelagem é feita utilizando as leis da física clássica. Devido à dificuldade de modelar o sistema com todas as variáveis que influenciam na movimentação do robô e também a dificuldade na obtenção de algumas constantes, como o atrito com o carpete, existem formas de determinar modelos de uma planta a partir da medição da entrada e saída do sistema. A esse tipo de modelagem dá-se o nome de identificação de sistemas (LJUNG, 1998).

De acordo com Beer e Johnston (1991) um modelo cinemático é aquele que trata da geometria do sistema e leva em conta a posição, velocidade, aceleração e tempo sem relação às suas causas. O modelo dinâmico é aquele que levará em conta as forças e torques envolvidos. Embora o modelo identificado neste trabalho não terá como variáveis de entrada as forças aplicadas, ele será obtido com base nos dados reais, portanto, será capaz de incorporar grande parte dos efeitos que afetam o sistema, desde que eles estejam presentes nos dados utilizados para identificar o modelo.

1.1 OBJETIVO

Este estudo tem como objetivo desenvolver um software para aquisição dos dados necessários para criar os modelos dinâmicos dos robôs de forma automática e apresentar cinco tipos de representações utilizadas na identificação de sistemas, sendo elas, o *Autoregression with Extra Inputs* (ARX) (AGUIRRE, 2004; LJUNG, 1998), *Autoregressive Moving Average with Exogenous Inputs* (ARMAX) (AGUIRRE, 2004; LJUNG, 1998), *Output Error* (OE) (AGUIRRE, 2004; LJUNG, 1998), *Box-Jenkins* (BJ) (AGUIRRE, 2004; LJUNG, 1998) e o modelo de Função de Transferência (FT) (OGATA; YANG, 2002). Dentre os quais, três representações serão escolhidas para serem testadas no robô da categoria SSL da equipe RoboFEI. Por fim, utilizando o MATLAB serão obtidos os modelos dinâmicos dos robôs, que serão comparados para descobrir qual fornece maior precisão a respeito do comportamento do robô.

¹ Competição em nível mundial que se desenrola todos os anos. Visa o estudo e desenvolvimento da Inteligência Artificial (IA) e da Robótica

1.2 JUSTIFICATIVA

Devido à necessidade de possuir um bom controlador de posição dos robôs é preciso que se tenha em mãos os modelos dinâmicos dos robôs. Porém, alguns motivos dificultam a obtenção destes modelos a partir das leis da física. Alguns destes motivos são: dificuldade de obtenção de alguns parâmetros, como o atrito com o carpete; ou variações na estrutura mecânica de cada robô como, por exemplo, utilizar robôs com motores mais desgastados junto com robôs com motores novos. Dados estes empecilhos ao obter o modelo dinâmico do robô, a técnica de identificação de sistemas mostra-se de grande utilidade, já que através dela obtém-se o modelo de qualquer planta a partir de dados reais a respeito da entrada e da saída do sistema, assim, quaisquer fenômenos físicos presentes no processo serão levados em conta no modelo final.

Além de empregar a técnica de identificação de sistemas é necessário que a obtenção dos dados e criação dos modelos sejam feitas de forma rápida e sem demandar conhecimento do operador no assunto, assim, qualquer membro da equipe será capaz de obter os modelos de cada robô. Isto é de grande importância, pois, a aquisição dos modelos deverá ser feita para seis até onze robôs antes de cada competição, já que alguns parâmetros poderão mudar em relação ao ambiente de laboratório. Pensando nisso, será desenvolvido um software que irá realizar os testes necessários em cada robô de forma automática, obtendo assim os dados requeridos para criar os modelos dinâmicos, estes dados serão devolvidos ao usuário que deverá fornecê-los ao MATLAB que será o responsável por gerar os modelos.

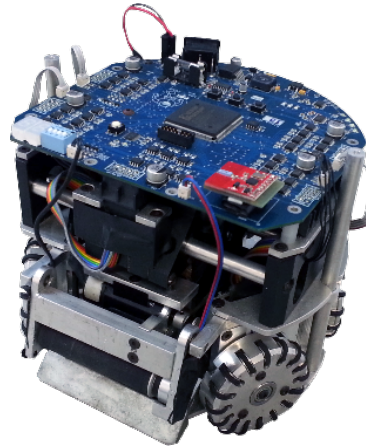
Com a obtenção de tais modelos torna-se possível a implementação de técnicas de controle sofisticadas de forma mais rápida. De acordo com Raimundo (2013) um controlador muito utilizado que faz o uso de modelos da planta é o *Model Predictive Control* (MPC). Este é um controlador que pode ser de grande uso para a equipe pois ele é capaz de trabalhar com sistemas multivariáveis, que é o caso do SSL, onde, têm-se três entradas, velocidades em X , Y e rotação, e três saídas, posição em X , Y e ângulo da frente do robô.

Outra possibilidade de uso do modelo é para realizar a previsão da posição futura do robô. O modelo obtido será uma função que leva em conta as posições e velocidades anteriores do robô e gera a posição atual do mesmo, portanto, para prever uma posição futura basta realimentar o resultado do modelo e gerar as velocidades correspondentes para aquela posição. Essas velocidades são geradas a partir de um controlador Proporcional Integral Derivativo (PID) (OGATA; YANG, 2002), ou seja, são fáceis de obter dada a posição atual. Dessa forma é possível realizar melhoras no controlador sem usar técnicas complexas e melhorar o desempenho das jogadas.

1.3 SISTEMA UTILIZADO

O sistema estudado neste trabalho trata-se do robô F180 da categoria SSL da RoboCup. Uma imagem do robô utilizado pela equipe RoboFEI pode ser vista na Figura 1

Figura 1 – Robô utilizado pela equipe RoboFEI.

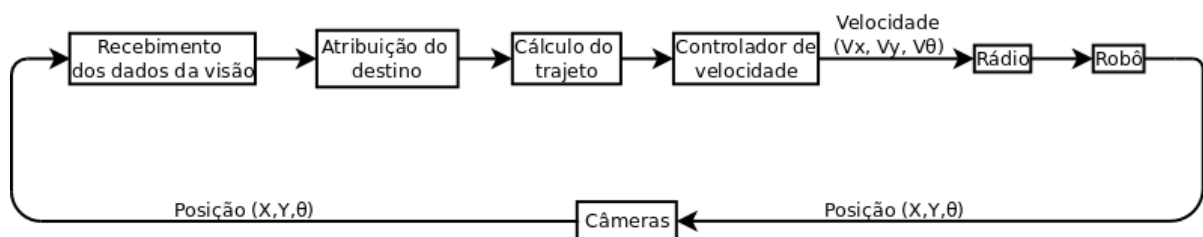


Fonte: Autor.

O robô utiliza um sistema de movimentação omnidirecional composto de quatro rodas com uma angulação de 33° , desta forma é possível que ele se movimente em qualquer direção de forma independente.

O sistema é controlado através do *software* de estratégia da equipe, que capta as informações dos robôs em campo (Posição em X, Y e ângulo) através de câmeras fixadas acima do campo, processa essas informações e envia, através do sistema de rádio, os dados necessários para que os robôs chutem, driblem e se movimentem. Na Figura 2 é possível ver um diagrama de blocos que representa o sistema de controle de posição dos robôs. É importante ter conhecimento sobre como o sistema opera, pois, existem algumas diferenças ao realizar o processo de identificação em malha aberta e malha fechada. Em Yuwana e Seborg (1982) têm-se um exemplo do uso de identificação em malha fechada, neste trabalho utilizou-se do método de identificação de sistemas para facilitar o processo de calibração de um controlador a partir da resposta ao degrau do sistema real em malha fechada.

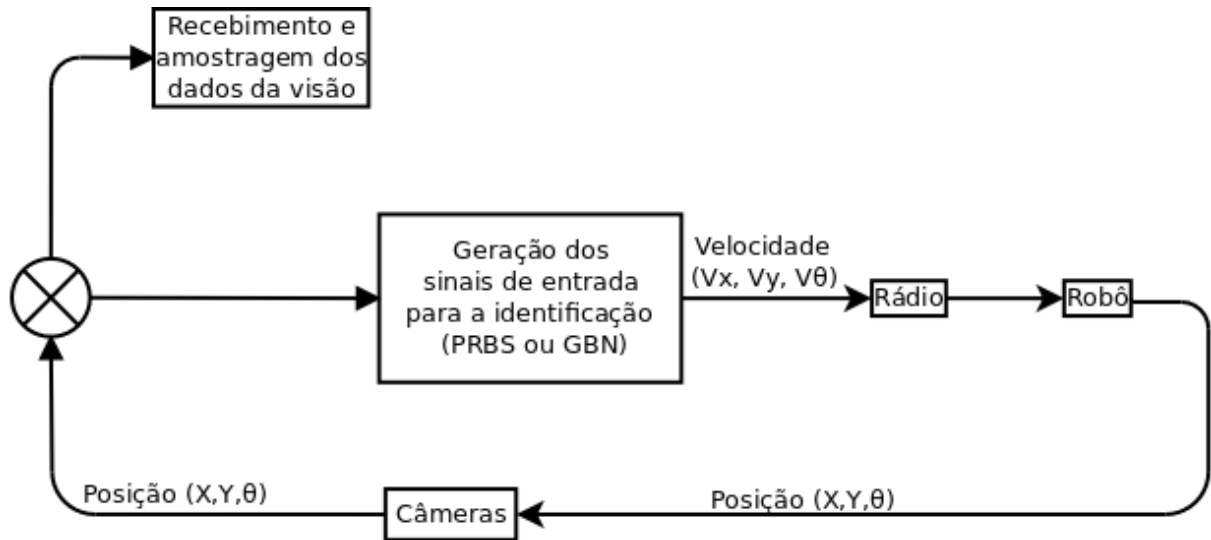
Figura 2 – Diagrama de blocos do sistema de controle de posição.



Fonte: Autor.

Como o objetivo deste projeto é obter somente o modelo do robô, o processo de identificação será feito em malha aberta, i.e. não haverá um controlador fazendo o controle das velocidades do robô. Este sistema pode ser visto no diagrama da Figura 3.

Figura 3 – Diagrama de blocos do software de aquisição de dados.



Fonte: Autor.

Embora trate-se um sistema de múltiplas entradas e múltiplas saídas (*Multiple Input Multiple Output* - MIMO), devido à capacidade de o robô conseguir se movimentar de forma independente em cada eixo espera-se que seja possível tratá-lo como três sistemas separados, um para o eixo X , Y e rotação. Porém, devido aos efeitos de escorregamento das rodas, principalmente no instante de partida, e algumas imperfeições mecânicas, notou-se que o robô pode deslocar-se em um segundo eixo ao receber um sinal de entrada. Como os modelos ARX e ARMAX são capazes de lidar com sistemas MIMO, isto não será um problema, basta realizar a excitação simultânea das entradas do robô.

De acordo com Aguirre (2004), ao excitar sistemas com múltiplas entradas, é importante que não haja correlação entre os sinais de entrada utilizados na identificação. Para garantir isso, basta que a *seed* do *Pseudo Random Binary Signal* (PRBS) e do gerador aleatório do *Generalized Binary Noise* (GBN) seja diferente para cada sinal.

Como o modelo de FT é empregado para sistemas *Single Input Single Output* (SISO), ele será estimado a partir da excitação independente de cada um dos eixos.

2 REVISÃO BIBLIOGRÁFICA

2.1 IDENTIFICAÇÃO DE SISTEMAS

2.1.1 Introdução

Para diversas áreas do conhecimento o ser humano procura modelos para representar fenômenos físicos para que assim possa-se obter certo domínio do que pode acontecer no presente e até no futuro. Um modelo nada mais é do que uma representação matemática a respeito do comportamento de um determinado sistema. Todo modelo possui uma característica em comum que é tentar unir diversas observações em algum tipo de padrão (LJUNG, 1987).

De acordo com Aguirre (2004), o ser humano mesmo sem perceber está sempre gerando e adaptando modelos a fim de realizar tarefas do dia a dia, por exemplo, dirigir um carro. Embora a tarefa de gerar modelos mentais para as tarefas corriqueiras do dia pareça ser simples, como que isto pode ser feito quando deseja-se obter o modelo de algum sistema dinâmico mais complexo, como uma torre de destilação (SRINIWAS et al., 1995). Assim como os seres humanos obtêm seus modelos mentais a partir de sua experiência e dados obtidos ao longo de sua vida, a técnica de identificação de sistemas tem o mesmo princípio, que é o de gerar modelos a respeito de um sistema a partir dos dados de entrada e saída do mesmo.

De acordo com Aguirre (2004), existem diversas formas de classificar as técnicas de modelagem. Uma delas classifica os métodos em modelagem caixa branca, modelagem caixa preta e modelagem caixa cinza. A modelagem caixa branca ou então modelagem pela física é a mais conhecida, nela é preciso um grande conhecimento a respeito das leis físicas que regem o sistema a ser estudado e também de diversas constantes, que são obtidas através de testes. Essa forma de modelagem possui um grande nível de dificuldade de ser realizada devido à necessidade de possuir grande conhecimento do sistema. A modelagem caixa preta é a que se enquadra na área de identificação de sistemas, pois, nesse tipo de modelagem o conhecimento necessário a respeito do sistema é pouco ou nenhum, basta ter acesso a medições realizadas na entrada e saída do sistema. A modelagem caixa cinza é uma mistura dos outros dois tipos de modelagem, ela pode ser mais "clara" ou "escura" dependendo do nível de informações que se possui do sistema, um exemplo desse tipo de modelagem pode ser visto em Aguirre (2004).

2.1.2 Características do sistema

Para que seja possível escolher um modelo que seja capaz de representar o sistema estudado é preciso ter conhecimento de algumas características que o sistema pode ter que influenciam em seu comportamento. Algumas dessas características são:

- a) **Invariância no tempo:** Significa que a resposta do sistema não depende do tempo absoluto (LJUNG, 1987). Não se deve confundir esta propriedade com a de o

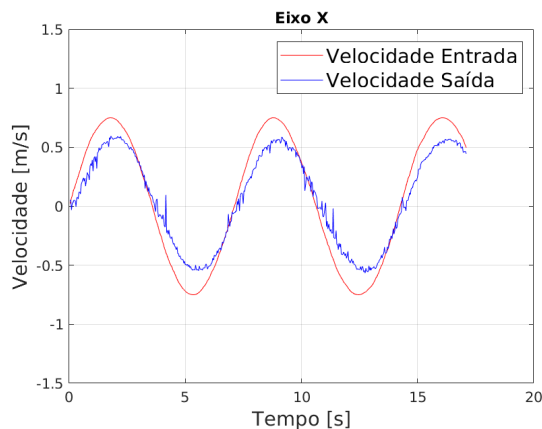
sistema estar estático, um sistema invariante no tempo implica que a dinâmica que rege o sistema não muda com o tempo (AGUIRRE, 2004).

- b) Linearidade: De acordo com Aguirre (2004), o princípio da linearidade implica que o sistema obedece o princípio da superposição, por exemplo, um sistema com as entradas a e b produz as saídas y_a e y_b e ao ser excitado pela entrada $a + b$ sua saída deverá ser $y_a + y_b$.

Sabendo que o sistema obedece à estas propriedades, podem ser utilizadas algumas simplificações que facilitam a decisão sobre qual tipo de modelo utilizar.

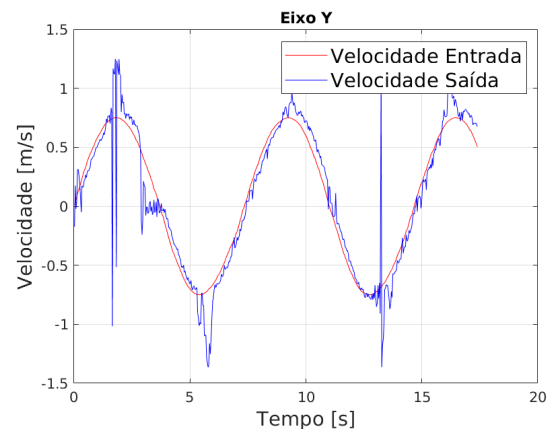
Em Hosseini, Johansen e Fatehi (2011), têm-se uma coletânea de métodos utilizados na detecção do grau de não linearidade de um sistema. A fim de identificar se o robô a ser utilizado pode ser considerado como um Sistema Linear Invariante no Tempo (SLIT). Neste trabalho foi utilizada a técnica de Análise Harmônica (HOSSEINI; JOHANSEN; FATEHI, 2011). Esta é uma técnica que aparenta ser bem simples de ser aplicada e pode ser feita no domínio do tempo. Para utilizá-la basta excitar o sistema utilizando um sinal senoidal da forma $y(t) = A * \sin(\omega * t)$, e então, de acordo com Haber (1985), se o sistema é SLIT então não deve aparecer nenhuma sub harmônica no sinal de saída diferente das quais estão no sinal de entrada. Nas Figuras 4, 5 e 6 têm-se o resultado obtido com os testes realizados, para a entrada o sinal utilizado foi de $V_i(t) = 0.75 * \sin(t)$ para $i = x, y$ [SI] e $V_\theta(t) = 5.4 * \sin(t)$ [SI].

Figura 4 – Resultado do teste de linearidade para o eixo X.



Fonte: Autor.

Figura 5 – Resultado do teste de linearidade para o eixo Y.

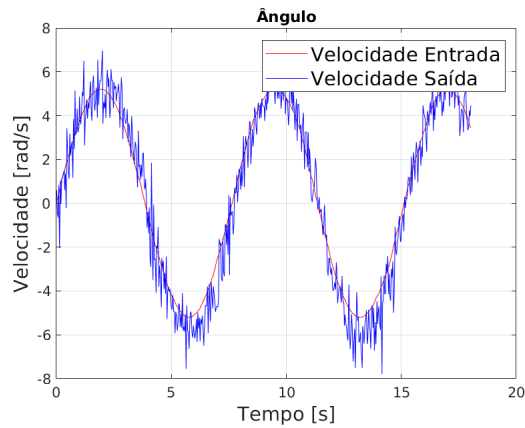


Fonte: Autor.

Nos testes realizados, ao desconsiderar os ruídos presentes na leitura, nota-se que não há grandes distorções no sinal de saída.

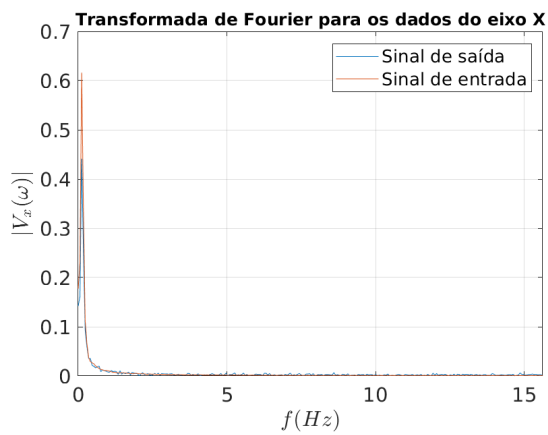
Também é possível confirmar que não há presença de sub harmônicas no sinal de saída ao calcular a Transformada Discreta de Fourier (TDF). Nas Figuras 7, 8 e 9 é possível ver o resultado deste cálculo. Isso mostra que o sistema pode ser considerado linear sem grandes perdas na precisão do modelo.

Figura 6 – Resultado do teste de linearidade para o ângulo.

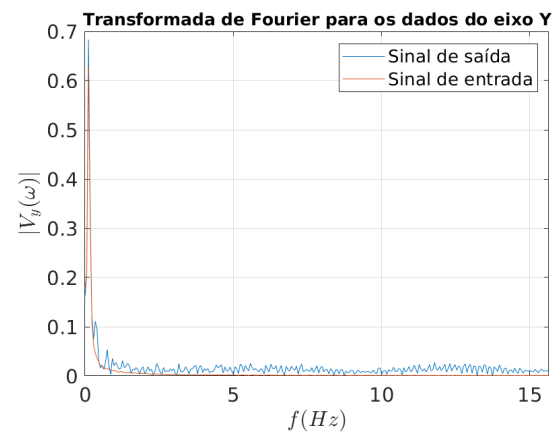


Fonte: Autor.

Figura 7 – TDF do sinal de entrada e saída em X. Figura 8 – TDF do sinal de entrada e saída em Y.

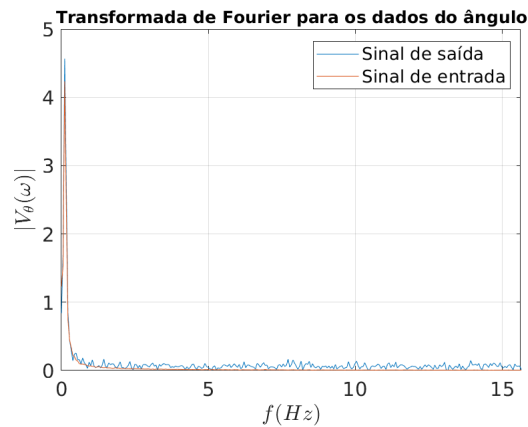


Fonte: Autor.



Fonte: Autor.

Figura 9 – TDF do sinal de entrada e saída em θ .



Fonte: Autor.

2.2 FORMAS DE PERTURBAÇÃO DE UM SISTEMA

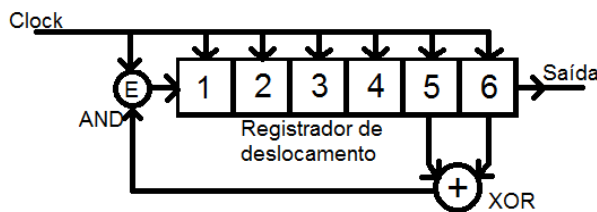
A fim de estimar o modelo do sistema é preciso que sejam coletados os dados de entrada e saída, porém, o sistema não pode ser excitado de qualquer maneira a desejo do operador. Existem formas específicas de perturbar o sistema para que os dados coletados tenham a maior quantidade possível de informações a respeito do sistema, assim, a capacidade do modelo de prever o comportamento do sistema será satisfatória.

Nesta seção serão apresentadas duas formas de perturbação utilizando sinais binários pseudo aleatórios. Estes sinais possuem apenas dois níveis $+A$ e $-A$, sendo A a amplitude máxima do sinal. A troca de um nível para outro ocorre sempre a cada instante $t = T_b$, de forma constante, e o período do sinal $T = NT_b$, onde N é um número ímpar (AGUIRRE, 2004).

2.2.1 Pseudo Random Binary Signal (PRBS)

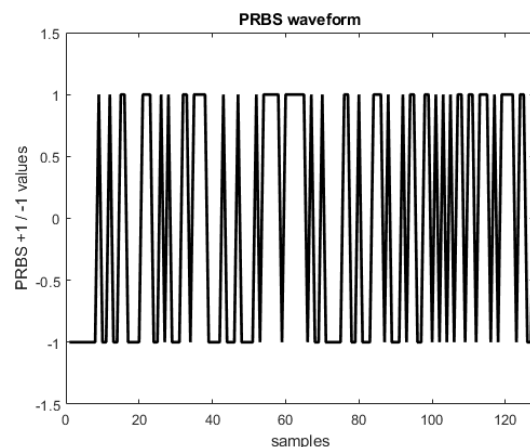
É o tipo mais comum de sinal binário pseudo aleatório devido ao seu simples funcionamento. Embora seu comportamento seja pseudo aleatório, a forma como a sequência de valores é gerada é determinística, isso significa que o mesmo sinal pode ser gerado diversas vezes. Como mostrado na Figura 10, um simples circuito digital é capaz de gerar um PRBS e na Figura 11 têm-se um exemplo de um PRBS com duração de 128 intervalos, a quantidade de intervalos gerada depende do número de bits do registrador de deslocamento, podendo ser calculado através da seguinte expressão: $N = 2^n - 1$, onde n é o número de bits (AGUIRRE, 2004).

Figura 10 – Gerador de sinais PRBS



Fonte: Adaptado de Aguirre (2004).

Figura 11 – PRBS gerado através do MATLAB



Fonte: Autor.

Como dito anteriormente, nem todo sinal pode ser utilizado para se obter um bom modelo, em Ljung (1987) são mostradas diversas propriedades estatísticas de um sinal que o tornam ótimo para a identificação de sistemas. No caso do PRBS, duas propriedades interessantes

são que sua média tende assintoticamente à zero e sua covariância se aproxima à do ruído branco à medida que sua duração tende ao infinito (ORENSTEIN, 2013).

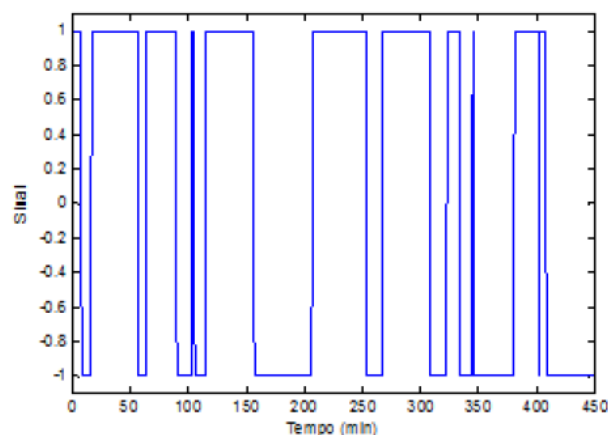
A parametrização de qualquer sinal para a excitação de um sistema não é uma tarefa simples e pode variar consideravelmente de um sistema para o outro, porém, existem algumas dicas que podem ser aplicadas de forma geral. De acordo com Aguirre (2004), é importante que o período do sinal PRBS não seja menor que a menor constante de tempo do sistema (τ_{min}). Além disso, Aguirre (2004) propõe uma forma de estimar o tempo de troca de nível (T_b). Esse é um parâmetro importante, pois, caso seja muito pequeno, o sistema não terá tempo suficiente para responder às mudanças de nível e se for muito longo o sinal irá ficar semelhante à um degrau, o que não é interessante para o processo de identificação do modelo. Portanto, para Aguirre (2004), o valor de T_b pode ser determinado a partir da Equação 1. Onde para sistemas lineares é comum escolher um valor próximo do limite inferior.

$$\frac{\tau_{min}}{10} \leq T_b \leq \frac{\tau_{min}}{3} \quad (1)$$

2.2.2 Generalized Binary Noise (GBN)

De acordo com Orenstein (2013), o ruído binário é um tipo de sinal que pode assumir somente dois níveis. Diferente do PRBS o GBN é totalmente aleatório, a probabilidade de troca de nível p é o que define se no próximo instante o nível atual será mantido, em Tulleken (1990) mostra-se que o valor de p deve ser definido de acordo com o tipo de sistema a ser identificado, variando de acordo com a ordem do sistema. Na Figura 12 tem-se um exemplo de um GBN com $p = 0.95$.

Figura 12 – Exemplo de sinal GBN.



Fonte: Orenstein (2013).

Para a parametrização deste tipo de sinal é possível modificar duas variáveis, a probabilidade de troca p e o intervalo de troca mínimo T . Em Orenstein (2013) é apresentada a Tabela 1 para ajudar na parametrização deste tipo de sinal a partir do tipo do processo, por exemplo,

se ele é de primeira ou segunda ordem. Onde, T^* é o tempo de troca mínimo ótimo, τ_s é a constante de tempo do sistema, p^* é a probabilidade de troca ótima e $E(T^*)$ é a média de trocas ótima. Assim, conhecendo o tipo do processo e sua constante de tempo, um bom valor de p e T pode ser determinado facilmente utilizando a Tabela 1.

Tabela 1 – Linhas gerais de parametrização do sinal GBN.

Tipo de processo	T^*/τ_s	p^*	$E(T^*)/\tau_s$
Primeira ordem	1/32	0.94	0.5
Segunda ordem sub-amortecido fase mínima	0.04	0.8	0.2
Segunda ordem sub-amortecido fase não mínima	1/15	0.8	1/3
Segunda ordem amortecido	0.05 a 0.10	0.9 a 0.95	1

Fonte: Orenstein (2013).

2.3 REPRESENTAÇÕES UTILIZADAS PARA IDENTIFICAÇÃO DE SISTEMAS

No processo de identificação de um sistema deve-se escolher qual a forma a ser utilizada da equação matemática que descreve o sistema, ou seja, o tipo de representação do modelo do sistema, para realizar esta decisão é necessário um certo conhecimento a respeito das características do sistema (Invariância no tempo e Linearidade). Após a conclusão dos testes mostrados na seção 2.1.2 será posição saber com certeza se o sistema é linear ou não. Por questões didáticas, nesta proposta serão apresentadas técnicas para sistemas lineares. Cinco das técnicas mais utilizadas serão discutidas a seguir com a finalidade de escolher três para aplicação neste trabalho.

2.3.1 Função de Transferência

De acordo com Aguirre (2004), dentre as representações de sistemas lineares a mais utilizada é a função de transferência (FT), que é definida como a transformada de Laplace da resposta ao impulso $h(t)$ do sistema, para condições iniciais nulas. A FT também pode ser obtida como sendo o quociente das transformadas de Laplace da saída do sistema e sua entrada (Equação 2) (OGATA; YANG, 2002).

$$H(s) = \frac{Y(s)}{X(s)} \quad (2)$$

Onde $Y(s)$ é a transformada de Laplace da saída do sistema e $X(s)$ é a transformada de Laplace da entrada do sistema.

Utiliza-se a notação $H(s)$ para uma FT contínua no tempo ou $H(k)$ para uma FT discreta no tempo. Dependendo do tipo de variável utilizada na resposta ao impulso do sistema, contínua ou discreta no tempo, deve-se utilizar diferentes tipos de transformada, sendo a transformada de Laplace utilizada para variáveis contínuas e a transformada Z utilizada para variáveis discretas.

A FT é um método interessante de representação de sistemas, pois, a partir dela é possível realizar diversas análises a fim de compreender o comportamento do sistema. Por exemplo, a partir da resposta em frequência do sistema, que é obtida substituindo-se s por $j\omega$ na FT (AGUIRRE, 2004), é possível aplicar o Critério de Estabilidade de Nyquist (OGATA; YANG, 2002), em que é possível determinar a estabilidade do sistema em malha fechada sem precisar determinar os polos da equação característica do sistema.

Outra vantagem é que com a FT em mãos, pode-se determinar o lugar das raízes, e assim, um controlador para o sistema pode ser facilmente projetado utilizando técnicas de controle clássico, por exemplo, controladores *lag* e *lead* (OGATA; YANG, 2002).

2.3.2 Representações em Tempo Discreto

Já que a coleta de dados é feita por amostragem, faz sentido que sejam utilizadas representações em tempo discreto, até por motivos de simplificação. De acordo com Ljung (1987), a forma mais simples de representar a relação entre entrada e saída é através de uma equação a diferenças como mostra a Equação 3.

$$y(k) + a_1y(k-1) + \dots + n_a y(k-n_a) = b_1u(k-1) + \dots + n_b u(k-n_b) + v(k) \quad (3)$$

Onde n_a, n_b determinam a ordem de cada polinômio, $u(k)$ é o sinal de entrada, $y(k)$ é a saída do sistema e $v(k)$ é o ruído branco.

Neste caso, os parâmetros de ajuste do modelo são: $\theta = [a_1 \dots a_n \ b_1 \dots b_n]^T$

Sendo q^{-a} o operador de atraso ($y(k)q^{-a} = y(k-a)$) é possível utilizar outra forma de representar a Equação 3. Definindo os polinômios $A(q)$ e $B(q)$ (Equação 4).

$$A(q) = 1 + a_1q^{-1} + \dots + a_{n_a}q^{-n_a}; B(q) = b_1q^{-1} + \dots + b_{n_b}q^{-n_b} \quad (4)$$

Têm-se que a Equação 3 pode ser escrita conforme a Equação 5.

$$y(k)A(q) = u(k)B(q) + v(k) \quad (5)$$

A partir disto é possível definir o modelo geral, mostrado na Equação 6 (AGUIRRE, 2004).

$$y(k) = \frac{B(q)}{F(q)A(q)}u(k) + \frac{C(q)}{D(q)A(q)}v(k) \quad (6)$$

Onde $C(q)$, $D(q)$ e $F(q)$ são os polinômios a seguir: $C(q) = 1 + c_1q^{-1} + \dots + c_{n_c}q^{-n_c}$; $D(q) = 1 + d_1q^{-1} + \dots + d_{n_d}q^{-n_d}$; $F(q) = 1 + f_1q^{-1} + \dots + f_{n_f}q^{-n_f}$.

Sendo n_c, n_d e n_f a ordem dos polinômios $C(q)$, $D(q)$ e $F(q)$, respectivamente.

Desta forma fica claro que os polinômios $A(q)$, $B(q)$, $C(q)$, $D(q)$ e $F(q)$ são os coeficientes das entradas, saídas e ruídos anteriores, a partir dos quais determina-se o estado atual da saída.

A partir do modelo geral descrito na Equação 6 surgem diversos casos especiais, que serão mostrados nas seções seguintes.

Vale a pena notar que a Equação 6 foi derivada considerando-se apenas uma entrada ($u(k)$) e uma saída ($y(k)$). Por estar lidando com um sistema MIMO de três entradas e três saídas, ao obter os modelos, no lugar de polinômios serão obtidas matrizes de polinômios. Dessa forma, os polinômios $A(q)$, $B(q)$, $C(q)$, $D(q)$ e $F(q)$ serão matrizes de tamanho $m \times n$, onde m é o número de linhas, n é o número de colunas. Para o sistema estudado $m = n$ e ambos são iguais ao número de entradas e saídas do sistema (três).

2.3.2.1 *AutoRegressive with eXogenous Inputs (ARX)*

Utilizando $C(q) = D(q) = F(q) = 1$ obtém-se o modelo auto regressivo com entradas externas ou *AutoRegressive with eXogenous inputs* (ARX). Dessa forma obtém-se a seguinte representação (Equação 7) (AGUIRRE, 2004).

$$y(k) = \frac{B(q)}{A(q)}u(k) + \frac{1}{A(q)}v(k) \text{ ou } A(q)y(k) = B(q)u(k) + v(k) \quad (7)$$

Nesta representação assume-se que o sistema e o ruído possuem polos comuns e nenhum tipo de dinâmica no numerador do ruído. Este tipo de representação também é considerado um modelo de erro na equação, devido aos polos comuns entre o sistema e o ruído, é dito que o ruído é incluído junto com o sinal de entrada no sistema (AGUIRRE, 2004; LJUNG, 1998).

Para este tipo de modelo é necessário fazer a estimação de $n_a + n_b$ parâmetros, que indicam a ordem dos polinômios $A(q)$ e $B(q)$, esses coeficientes podem ser obtidos utilizando o método dos mínimos quadrados. Ao longo deste trabalho o modelo ARX será referido como $ARX(n_a, n_b)$ com atraso n_k , onde n_k é o atraso entre a entrada e a saída (LJUNG, 1998; TANGIRALA, 2014).

Levando em conta que o modelo obtido terá um formato matricial, devido ao número de entradas e saídas. Para o sistema abordado, utilizando um modelo $ARX(2,2)$ com atraso 0, será obtido algo com o formato da Equação 8.

$$\begin{bmatrix} 1 + a_{1_x}q^{-1} + a_{2_x}q^{-2} & 0 + a_{3_x}q^{-1} + a_{4_x}q^{-2} & 0 + a_{5_x}q^{-1} + a_{6_x}q^{-2} \\ 0 + a_{1_y}q^{-1} + a_{2_y}q^{-2} & 1 + a_{3_y}q^{-1} + a_{4_y}q^{-2} & 0 + a_{5_y}q^{-1} + a_{6_y}q^{-2} \\ 0 + a_{1_\theta}q^{-1} + a_{2_\theta}q^{-2} & 0 + a_{3_\theta}q^{-1} + a_{4_\theta}q^{-2} & 1 + a_{5_\theta}q^{-1} + a_{6_\theta}q^{-2} \end{bmatrix} \times \begin{bmatrix} x(k) \\ y(k) \\ \theta(k) \end{bmatrix} = \begin{bmatrix} 1 + b_{1_x}q^{-1} + b_{2_x}q^{-2} & 0 + b_{3_x}q^{-1} + b_{4_x}q^{-2} & 0 + b_{5_x}q^{-1} + b_{6_x}q^{-2} \\ 0 + b_{1_y}q^{-1} + b_{2_y}q^{-2} & 1 + b_{3_y}q^{-1} + b_{4_y}q^{-2} & 0 + b_{5_y}q^{-1} + b_{6_y}q^{-2} \\ 0 + b_{1_\theta}q^{-1} + b_{2_\theta}q^{-2} & 0 + b_{3_\theta}q^{-1} + b_{4_\theta}q^{-2} & 1 + b_{5_\theta}q^{-1} + b_{6_\theta}q^{-2} \end{bmatrix} \times \begin{bmatrix} u_x(k) \\ u_y(k) \\ u_\theta(k) \end{bmatrix} + \begin{bmatrix} v_x(k) \\ v_y(k) \\ v_\theta(k) \end{bmatrix} \quad (8)$$

Embora no caso matricial seja possível especificar a ordem de cada um dos polinômios das matrizes, a não ser que seja dito o contrário, os modelos sempre serão estimados com a mesma ordem para uma mesma matriz, dessa forma, a representação $ARX(n_a, n_b)$ com atraso n_k continua sendo válida.

A predição de um passo à frente para este tipo de modelo pode ser calculada a partir da Equação 9. A saída predita é identificada pelo chapéu (\hat{y}) na variável de saída (TANGIRALA, 2014).

$$\hat{y}(k) = B(q^{-1})u(k) + (1 - A(q^{-1}))y(k). \quad (9)$$

Para adaptar a equação de predição ao formato da Equação 8 basta usar a matriz de identidade no lugar da constante unitária que subtrai o polinômio $A(q^{-1})$ na Equação 9.

Devido à simplicidade de obtenção e uso deste tipo de modelo, ele é considerado um bom chute inicial à respeito de uma boa estrutura de modelo para se utilizar (TANGIRALA, 2014).

2.3.2.2 *AutoRegressive Moving Average with eXogenous Inputs (ARMAX)*

A partir do modelo geral (Equação 6), utilizando $D(q) = F(q) = 1$. Assim, obtém-se o modelo auto regressivo com média móvel com entradas externas ou *AutoRegressive Moving Average with eXogenous inputs* (ARMAX), mostrado a seguir (Equação 10).

$$y(k) = \frac{B(q)}{A(q)}u(k) + \frac{C(q)}{A(q)}v(k) \quad (10)$$

Neste modelo, assim como o ARX, assume-se polos comuns entre o sistema e o ruído, porém é considerada a dinâmica do polinômio $C(q)$ no numerador. De acordo com Ljung (1998), a inclusão desse polinômio faz com que o valor atual de $y(k)$ dependa dos valores previstos anteriormente. Este polinômio também torna o ruído $v(k)$ uma média móvel do ruído branco.

De acordo com Ljung (1998), a técnica de máxima verossimilhança é utilizada para estimar os coeficientes deste tipo de modelo.

O modelo ARMAX pode ser visto como uma versão melhorada do tipo ARX. Devido à introdução de um termo de média móvel ($C(q)$) ele é capaz de modelar melhor os ruídos presentes na planta, dessa forma, fornecendo um modelo mais flexível para o ruído do que o modelo ARX. Dada esta característica ele tem grandes aplicações na área industrial (TANGIRALA, 2014).

Para realizar a predição de um passo à frente com este tipo de modelo utiliza-se a Equação 11 (TANGIRALA, 2014).

$$\hat{y}(k) = \frac{B(q^{-1})}{A(q^{-1})}u(k) + \left(1 - \frac{C(q^{-1})}{A(q^{-1})}\right)y(k) \quad (11)$$

De forma análoga ao modelo ARX, para o ARMAX também serão obtidas matrizes de polinômios, a utilização será parecida, porém, claramente existirá a adição de uma matriz com os polinômios $C(q)$ na equação.

O modelo ARMAX será referido como: $\text{ARMAX}(n_a, n_b, n_c)$ com atraso n_k , onde n_k é o atraso entre a entrada e a saída.

2.3.2.3 Output Error (OE)

Esta representação pode ser obtida a partir do modelo geral (Equação 6), fazendo com que $A(q) = C(q) = D(q) = 1$. Ao desconsiderar polos comuns entre o sinal de entrada e o ruído, é dito que este modelo é do tipo erro na saída, isto é o que dá nome à esta representação (*Output Error* - OE) (AGUIRRE, 2004).

Dessa forma é desconsiderada qualquer dinâmica presente no ruído, assim, o ruído presente no sistema é somente um ruído branco puro. De acordo com Ljung (1998), uma característica desta representação é que a saída do sistema, $y(k)$, irá depender apenas das entradas anteriores. Esta representação pode ser vista na Equação 12.

$$y(k) = \frac{B(q)}{F(q)}u(k) + v(k) \quad (12)$$

Devido à não linearidade nos parâmetros a serem encontrados neste tipo de modelo, utiliza-se geralmente o método de minimização de Gauss-Newton (LJUNG, 1998).

De acordo com Tangirala (2014), para prever um passo à frente com este modelo deve-se utilizar a Equação 13.

$$\hat{y}(k) = \frac{B(q^{-1})}{F(q^{-1})}u(k) \quad (13)$$

A partir da Equação 13 nota-se facilmente que para realizar a predição da saída futura, este tipo de modelo utiliza somente as entradas anteriores, em outras palavras, ele trabalha em malha aberta. Tangirala (2014) afirma que esta é uma característica importante deste tipo de modelo para a modelagem de processos em malha aberta, pois, como ele não é condicionado para modelar os ruídos da planta ele é capaz de prover uma boa estimativa da função de transferência da mesma.

Analisando Tangirala (2014) nota-se uma característica importante de conhecer-se a respeito deste modelo, as predições de um passo à frente e de infinitos passos à frente irão coincidir para um modelo do tipo OE, portanto, utilizando este tipo de modelo não é possível prever o estado do sistema alguns passos no futuro. Como no cenário da SSL é de grande utilidade possuir um modelo capaz de indicar a posição futura dos robôs, por exemplo, para utilização com um filtro de Kalman, o modelo OE não é interessante de ser obtido para esta aplicação.

2.3.2.4 Box-Jenkins (BJ)

A partir do modelo geral (Equação 6), utilizando $A(q) = 1$, obtém-se o modelo Box-Jenkins (BJ), mostrado na Equação 14.

$$y(k) = \frac{B(q)}{F(q)}u(k) + \frac{C(q)}{D(q)}v(k) \quad (14)$$

Assim como o modelo OE, o BJ também é considerado um modelo do tipo erro na saída, porém, nele considera-se toda a dinâmica presente no ruído. Diferente do modelo ARMAX, a FT do ruído ($\frac{C(q)}{D(q)}$) não possui parâmetros em comum com a FT do sistema ($\frac{B(q)}{F(q)}$) (AGUIRRE, 2004). De acordo com Ljung (1998) esta representação se adapta melhor às situações em que o ruído é inserido no final do processo, por exemplo, ruído de medição, que é uma característica interessante para a aplicação atual devido à presença de ruído na medição da posição dos robôs, causado pelas regiões de *overlap*.

De acordo com Hipel, McLeod e Lennox (1977), o método proposto em McLeod (1977) é capaz de produzir resultados satisfatórios na estimação dos parâmetros para este modelo.

Embora o modelo BJ seja um dos mais flexíveis, devido à parametrização independente do sinal de entrada e do ruído, este tipo de modelo tem uma complexidade elevada na estimação de seus parâmetros e por isso não costuma ser escolhido a priori (TANGIRALA, 2014). Neste trabalho a estimação dos modelos é feita utilizando as ferramentas disponibilizadas pelo MATLAB, portanto, o trabalho empregado na estimação dos parâmetros não será levado em conta. Porém, é importante saber que não se deve utilizar um modelo que leve ao efeito conhecido como *overfitting*, que é quando um modelo é parametrizado de tal forma que se ajuste demais aos dados utilizados na identificação, fazendo com que sua resposta para outras situações não seja satisfatória. O mesmo vale para a decisão de qual o grau que deve-se utilizar no modelo estimado. Portanto, dada a grande quantidade de parâmetros no modelo BJ, é necessário tomar cuidado para que não ocorra *overfitting* (TANGIRALA, 2014).

2.4 TRABALHOS CORRELATOS

Em Pereira (2000) é possível ver uma aplicação muito similar ao que será feito neste projeto para a SSL. Neste trabalho é feita a identificação e controle de um micro robô móvel da categoria *Very Small Size League* (VSSL), que é uma categoria mais simples da SSL. Embora não seja o mesmo tipo de robô, na VSSL é utilizado um robô não holonômico, muito pode ser aproveitado. Na parte de identificação o autor utiliza três modelos ARX(2,2) com atraso 8. E para o sinal de entrada é feito o uso de um sinal do tipo PRBS.

Mendes (2012) também trata da identificação de sistemas aplicada para um robô da categoria VSSL, porém, propões um tratamento diferente para a definição da estrutura do modelo. Neste trabalho o autor da um tratamento especial para a zona morta presente no sistema. Para

a identificação do modelo foram utilizados dois modelos ARX(2,4) com atraso 0. No sinal de entrada também utilizou-se um sinal do tipo PRBS.

Nota-se também que em Mendes (2012) a forma de modelagem pode ser considerada como caixa cinza, pois, antes de realizar a identificação do sistema foi feita toda a modelagem cinemática e dinâmica do robô e a partir disso é que foi estimado um modelo com um formato parecido com o que foi deduzido teoricamente, ou seja, a técnica de identificação de sistemas foi utilizada somente para obter as constantes físicas do sistema, como, atrito com o chão. Neste trabalho a identificação do modelo será feita completamente caixa preta, utilizando o mínimo/nenhum conhecimento prévio sobre o sistema.

Ambos os trabalhos foram de grande valor para esta pesquisa, pois, fornecem uma noção da ordem de modelo que deve-se utilizar para obter uma resposta com precisão suficiente. Mesmo sendo outro tipo de robô e categoria de jogo, ambas as ligas compartilham características em comum, como, o sistema de visão global que envia a posição dos robôs no campo, tornando os conhecimentos de fácil adaptação e utilização de uma liga para outra.

3 METODOLOGIA

Foi feita uma pesquisa a respeito de cada um dos três modelos de identificação de sistemas escolhidos e também sobre as formas de perturbação do sistema a fim de obter os dados necessários para utilizar tais modelos, como o PRBS ou o GBN.

O software responsável por realizar os testes necessários em cada robô foi desenvolvido e devidamente integrado com o MATLAB. Com os dados em mãos foi utilizada a ferramenta *System Identification Toolbox* do MATLAB, para gerar os modelos dinâmicos do robô. Esses modelos foram gerados a partir das três formas de modelagem estudadas anteriormente, que serão testadas a fim de determinar qual possui maior precisão em representar o comportamento do robô.

Foram utilizadas três métricas para determinar a adequação dos modelos gerados, sendo elas: a forma gráfica, onde é verificado se o modelo é capaz de reproduzir valores próximos dos reais para uma amostra diferente da utilizada para gerá-lo; o cálculo do erro médio quadrático relativo (*Mean Relative Squared Error* - MRSE) e da raiz do erro médio quadrático normalizado (*normalized Root Mean Squared Error* - nRMSE) conforme as Equações 15 e 16. Ambas são métricas utilizadas na literatura e a última é calculada automaticamente pelo MATLAB (ORENSTEIN, 2013).

$$MRSE = \frac{1}{k} \sum_{i=1}^k \sqrt{\frac{\sum_{j=1}^N \hat{e}_{ij}^2}{\sum_{j=1}^N y_{ij}^2}} 100\% \quad (15)$$

$$nRMSE = \left(1 - \frac{\sqrt{\sum_{j=1}^N \hat{e}_{ij}^2}}{\sqrt{\sum_{j=1}^N (y_j - \bar{y})^2}} \right) 100\% \quad (16)$$

Onde k é o número de saídas do sistema, N é o número de pontos medidos, \hat{e} é o erro da medida menos a previsão no instante j , y é a medida no instante j , \bar{y} é a média do sinal de saída. Para um modelo bem ajustado aos dados, o valor do MRSE tende à 0% e o nRMSE tende à 100%, se o nRMSE for igual à zero, isso significa que uma linha reta com o valor da média é melhor que o modelo gerado.

Os modelos foram testados através da avaliação gráfica e das métricas MRSE e nRMSE, o procedimento foi feito utilizando o método de validação cruzada. De acordo com Ljung (1987) este método consiste em gerar o modelo a partir de um conjunto de dados, e simular a resposta do modelo utilizando um segundo conjunto de dados obtido através de um outro experimento. Para isso, serão coletadas cerca de três amostras de dados para diferentes perturbações na entrada, com uma dessas amostras serão gerados os modelos do robô, com as outras duas amostras o modelo será testado. Todos os testes citados serão feitos utilizando o MATLAB.

Em Ljung (1987) também é definido o método de simulação para a validação do modelo. Nesse método é feito o cálculo da resposta do modelo sem utilizar as saídas anteriores (como num modelo do tipo OE). Embora este método de validação não seja interessante no caso estudado, é importante saber que o MATLAB, ao simular um modelo, utiliza esta técnica

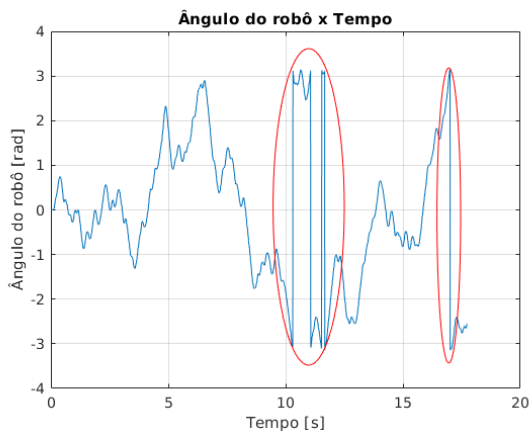
independentemente da estrutura utilizada, ou seja, ao simular um modelo ARX, o mesmo irá apresentar o mesmo comportamento de um modelo OE de mesma ordem. Portanto, para observar a resposta real do modelo é preciso fazer com que o MATLAB faça a predição um passo à frente, dessa forma, a resposta do modelo será calculada usando os valores de entrada e saída anteriores.

A fim de comprovar a efetividade dos modelos de FT foi feita a modelagem de um controlador Proporcional Integral Derivativo (PID), utilizando a ferramenta *PIDTuner* do MATLAB, para substituir o atual controlador de posição do robô que a equipe utiliza.

3.1 NORMALIZAÇÃO DO ÂNGULO DO ROBÔ

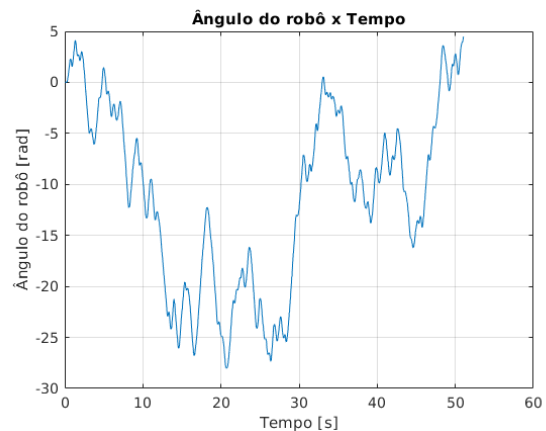
Uma característica encontrada na amostragem dos dados é que o ângulo do robô (θ) varia de $-\pi < \theta < +\pi$, o que acaba causando uma descontinuidade no valor desta variável, e assim o modelo gerado não é capaz de se ajustar satisfatoriamente nos dados. Na Figura 13 é possível ver que, ao passar do valor de π o ângulo muda imediatamente para $-\pi$ e vice-versa, os lugares onde isso acontece estão circulado em vermelho.

Figura 13 – Descontinuidade de θ .



Fonte: Autor.

Figura 14 – Variação contínua de θ .



Fonte: Autor.

O problema com isso é que este não é um comportamento facilmente interpretável do ponto de vista matemático, isso faz com o que o modelo não possua um comportamento adequado para previsões maiores que um passo.

É importante notar que o mesmo acontece utilizando outro intervalo de ângulos, por exemplo $0 < \theta < 2\pi$.

Para resolver este problema decidiu-se utilizar o ângulo sem fazer restrições quanto ao valor que o mesmo pode assumir, ou seja, o ângulo θ irá variar de $-\infty$ a $+\infty$ [rad] considerando como positivo o sentido anti-horário, dessa forma, ao realizar duas voltas completas no sentido anti-horário o ângulo θ será de 4π rad, por exemplo. Dessa forma o ângulo pode variar sem nenhuma descontinuidade, independente de como o robô se movimenta, isso pode ser visto na

Figura 14, nesse exemplo nota-se que o ângulo varia de forma contínua em todo o intervalo de 5 rad a -30 rad .

4 RESULTADOS FINAIS

4.1 ESCOLHA DO FORMATO DOS MODELOS

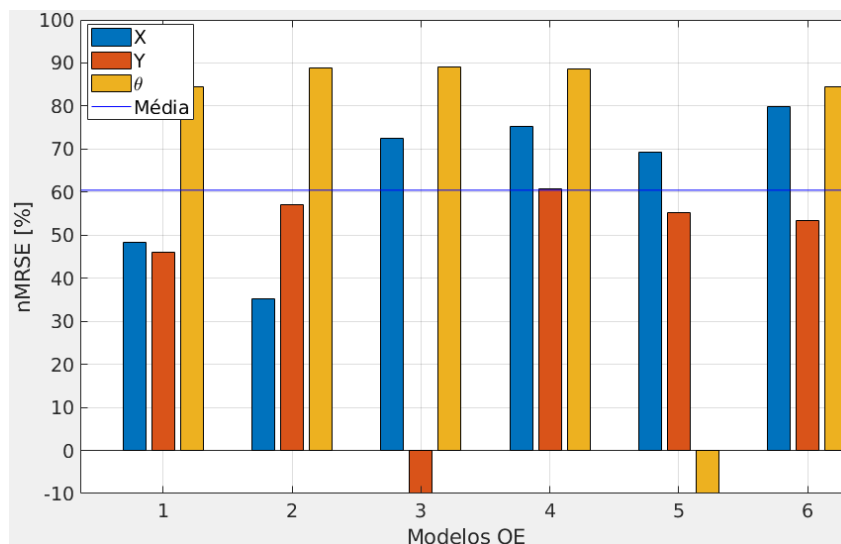
Neste trabalho foram apresentados os modelos do tipo FT, ARX, ARMAX, OE e BJ. Entretanto, destes cinco, três serão escolhidos para comparação e determinação de qual melhor representa o comportamento do robô.

A partir do estudo bibliográfico realizado e algumas análises empíricas foi possível perceber que os modelos do formato OE e BJ não eram apropriados para esta aplicação.

Embora o modelo BJ seja o mais flexível de todos os formatos apresentados, de acordo com Tangirala (2014) ele é o mais complexo de ser estimado, pois, deve ser feita a escolha da ordem de quatro polinômios. Além disso, Tangirala (2014) sugere que deve-se priorizar modelos simples, de forma a diminuir o risco de *overfitting*. Por esses motivos, o modelo BJ não foi escolhido.

Ao realizar alguns testes com o modelo do tipo OE nota-se que a performance deste formato não é muito boa para o sistema estudado neste trabalho. Isso pode ocorrer devido à característica do modelo OE de não tentar descrever as características do ruído presente nos dados (TANGIRALA, 2014). Na Figura 15 é possível ver uma rápida comparação entre alguns modelos OE de ordem variada.

Figura 15 – nMRSE para modelos OE de ordem variada.



Fonte: Autor.

Através da Figura 15 é possível notar que na média, o modelo OE alcança a marca dos 60% de nMRSE.

Portanto, sobram os modelos ARX, ARMAX e FT. Como apontado na revisão bibliográfica, o modelo ARX é um dos formatos mais simples de modelo polinomial e, dependendo da aplicação, também é capaz de fornecer bons modelos. O modelo ARMAX, devido à sua

melhor capacidade de modelar o ruído também é um bom candidato. O modelo de FT, embora possua a limitação de modelar somente sistemas SISO, possui uma praticidade considerável no projeto de controladores, e por isso também será estimado.

4.2 PARAMETRIZAÇÃO DOS SINAIS

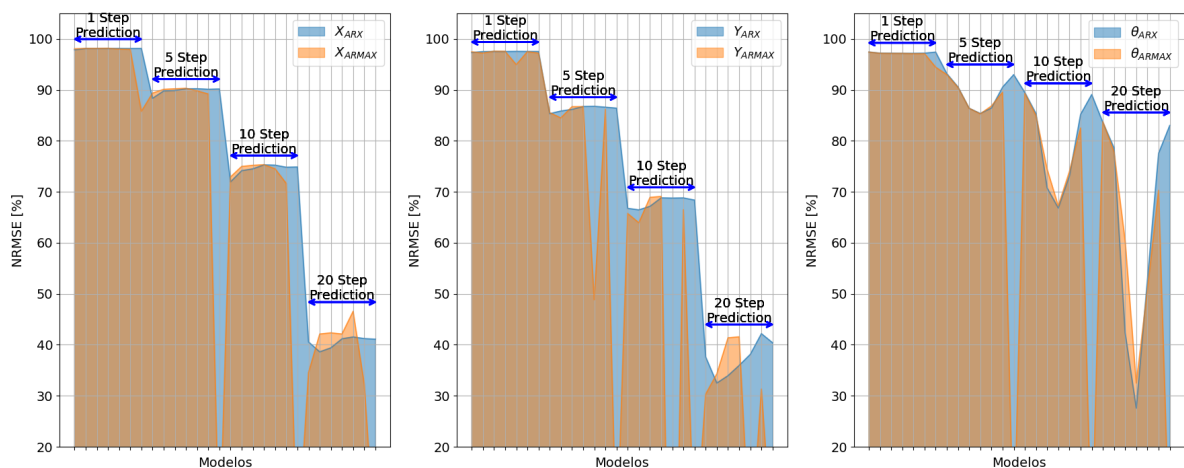
Para os testes finais, foram feitos diversos experimentos a fim de determinar uma parametrização dos sinais PRBS e GBN que gerasse bons resultados. Com os modelos ARX e ARMAX gerados antes de uma boa parametrização dos sinais de excitação obteve-se previsões de até seis passos a frente com erro aceitável. Portanto, com o sinal parametrizado da forma correta este limite de previsão deve aumentar.

4.2.1 GBN

A melhor parametrização para o sinal GBN foi obtida ao utilizar uma probabilidade de troca diferente para cada eixo do robô. Os valores de p determinados para os eixos X , Y e θ são: 0.35, 0.35 e 0.8 e as amplitudes são: 0.5 m/s, 0.5 m/s e 2.5 rad/s, respectivamente.

Na Figura 16 é possível ver o resultado obtido com o sinal GBN que gerou os melhores modelos antes da parametrização melhorada. Nota-se que na Figura 16, para previsões de 20 passos à frente, os modelos mal conseguiram atingir a marca de 40% de nRMSE.

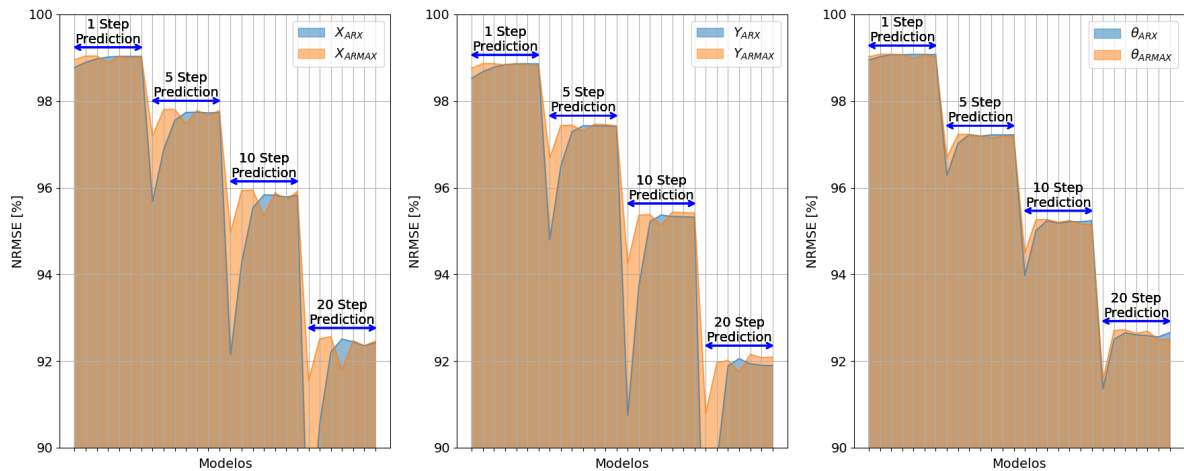
Figura 16 – Modelos obtidos antes da parametrização do sinal GBN.



Fonte: Autor.

Com a nova parametrização do sinal, e um melhor ajuste das ordens dos modelos, os resultados foram superiores à 92% de $nRMSE$ para previsões de 20 passos à frente, que resulta em aproximadamente 300 ms. A comparação entre os modelos obtidos pode ser vista na Figura 17.

Figura 17 – Modelos obtidos após parametrização do sinal GBN.



Fonte: Autor.

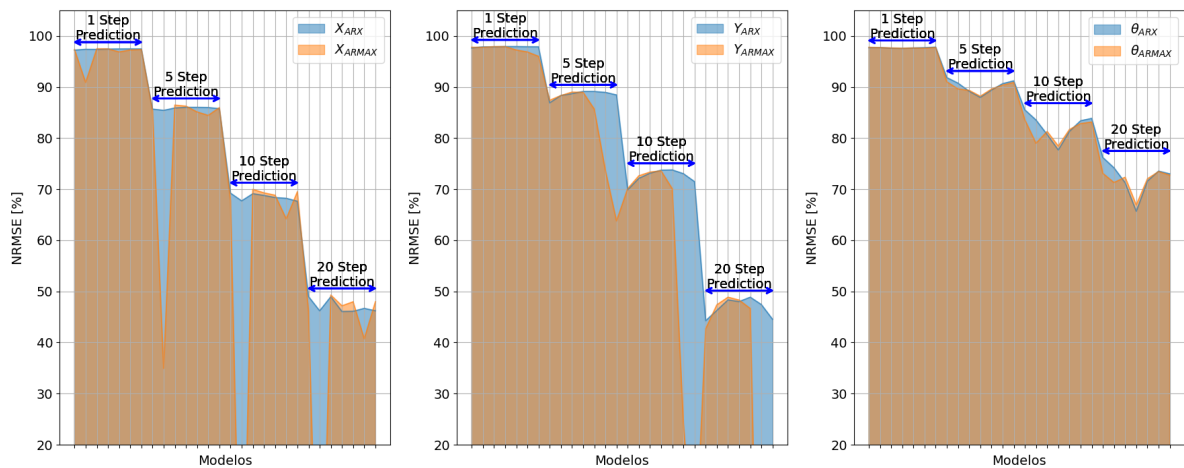
4.2.2 PRBS

Para o PRBS foi possível obter bons resultados sem necessitar de uma parametrização específica para cada eixo, os valores obtidos são: $T_b = 250 \text{ ms}$ e as amplitudes para os eixos X , Y e θ são: 0.2 m/s , 0.2 m/s e 2 rad/s , respectivamente.

De forma similar ao que foi feito para o sinal GBN, também foi feita a simulação com o sinal PRBS que gerou os melhores modelos antes da parametrização melhorada, o resultado disso pode ser visto na Figura 18.

Analisando a Figura 18 fica evidente que, a performance antes da parametrização era muito ruim para predições de 20 passos à frente, os valores alcançados pelos modelos não chegavam à 50% de $nRMSE$.

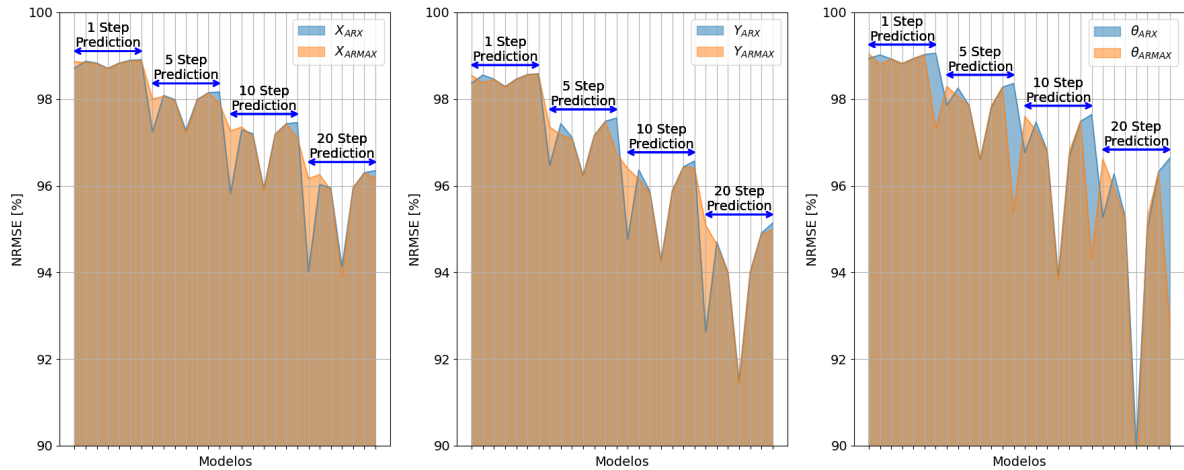
Figura 18 – Modelos obtidos antes da parametrização do sinal PRBS.



Fonte: Autor.

Como pode ser visto na Figura 19, com a nova parametrização foi possível obter modelos que superam a marca dos 94% de nRMSE.

Figura 19 – Modelos obtidos após parametrização do sinal PRBS.



Fonte: Autor.

4.3 SOFTWARE DE AQUISIÇÃO DE DADOS

O software de aquisição de dados foi completamente implementado e funcionou corretamente. Nele é possível realizar os testes utilizando os sinais PRBS, GBN, degrau e senoidal. O sinal do tipo degrau foi adicionado para facilitar a obtenção da constante de tempo do sistema, utilizada para a parametrização dos sinais de entrada para os dados de identificação do modelo, assim como para estimar os modelos de FT. O sinal do tipo senoidal foi adicionado para facilitar o teste de linearidade do sistema.

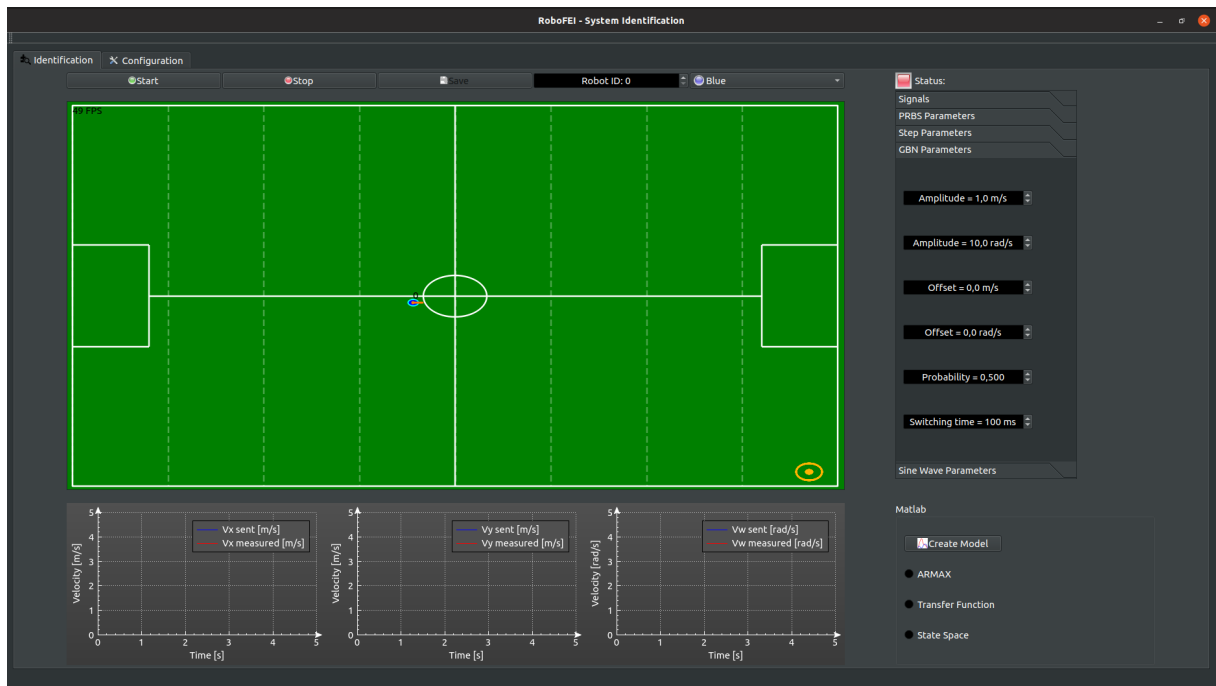
Na Figura 20 é possível ver uma imagem da interface do software. Para a configuração dos sinais, é possível determinar a amplitude do sinal aplicado em m/s para os eixos lineares e rad/s para o eixo de rotação, o *offset* do sinal, ou seja, o quanto o usuário quer que o sinal seja deslocado (para cima ou para baixo) e os parâmetros específicos de cada tipo de sinal.

4.3.1 Aplicativo em MATLAB para Identificação dos Modelos

Para facilitar o processo de identificação dos modelos, foi criado um aplicativo utilizando o MATLAB. Ele é responsável por ler os dados coletados pelo software de aquisição e gerar os modelos conforme o usuário desejar. O aplicativo pode ser visto nas Figuras 21 e 22.

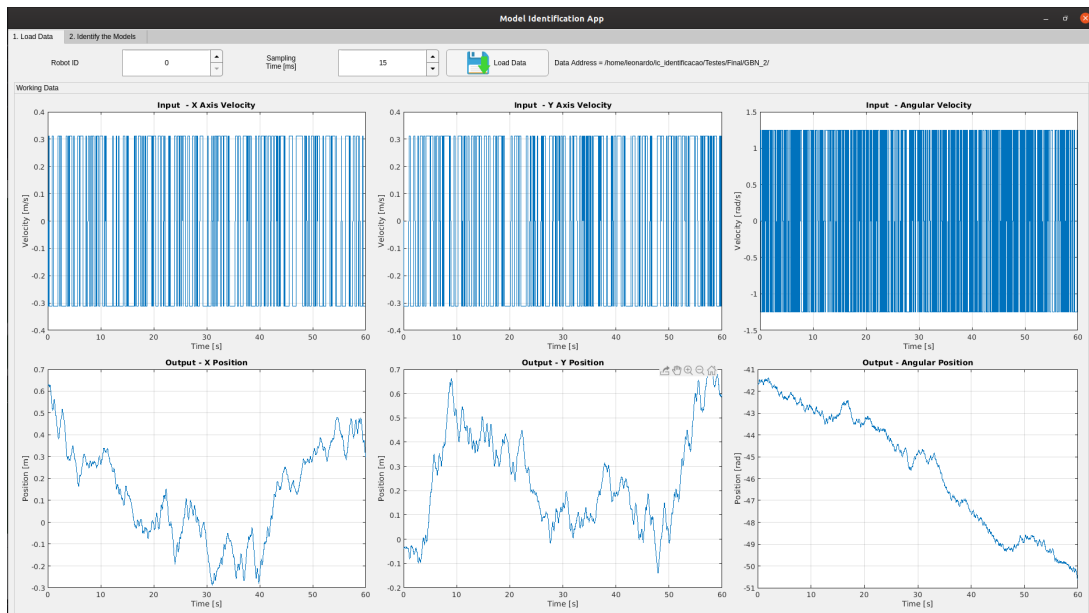
O aplicativo é distribuído em duas abas, a primeira (Figura 21) é onde é feito o carregamento dos dados do experimento realizado. Nesta aba é realizada a configuração do tempo de amostragem utilizado no experimento e também é possível visualizar os gráficos dos dados de entrada e saída do robô.

Figura 20 – Software de aquisição de dados.



Fonte: Autor.

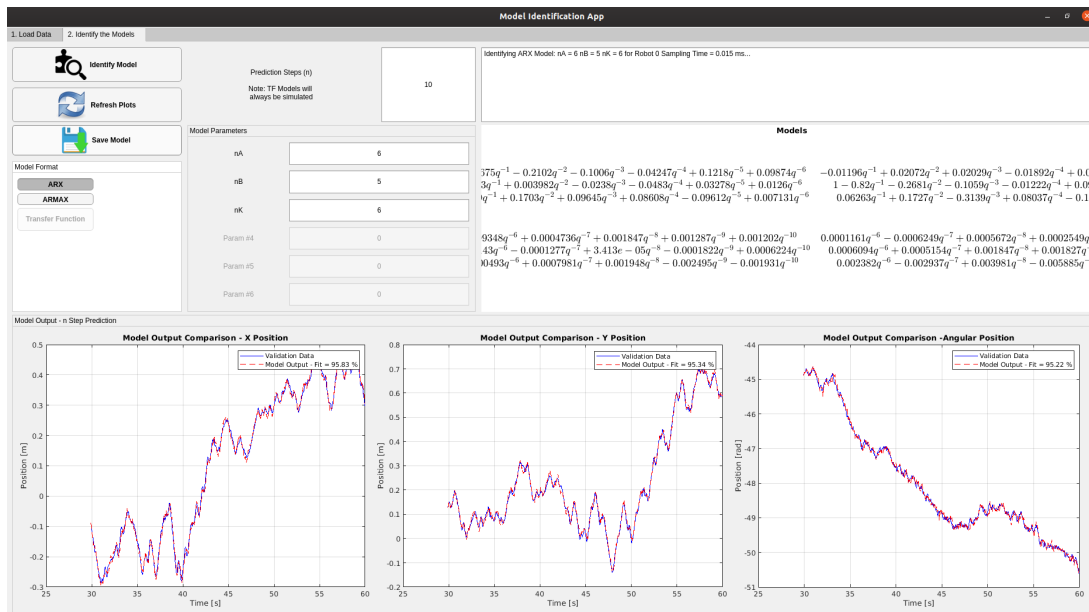
Figura 21 – Interface de carregamento dos dados.



Fonte: Autor.

A segunda aba é onde é realizada a configuração do modelo a ser estimado. Nela é possível escolher o tipo de modelo, i.e. ARX, ARMAX ou FT, a ordem de cada polinômio ou número de polos e zeros e quantos passos à frente será feita a simulação. Também é possível visualizar o modelo identificado e os gráficos de simulação.

Figura 22 – Interface de identificação dos modelos.



Fonte: Autor.

4.4 MODELOS ARX E ARMAX

Nesta seção serão apresentados os modelos ARX e ARMAX estimados, assim como a análise dos mesmos.

Os modelos foram obtidos de forma o mais caixa preta (ver Seção 2.1.1) possível, assim, demandando o mínimo de conhecimento sobre o sistema por parte do operador. A ideia é que os modelos possam ser facilmente estimados ao trocar o ambiente em que o robô será utilizado, por exemplo, ao chegar numa competição, trocar o carpete do laboratório ou utilizar rodas com um atrito diferente. Dessa forma, caso algum controlador utilize o modelo, ele estará sempre com uma representação fiel do comportamento do robô.

Nos modelos identificados, foi considerado como sendo a saída a posição do robô (X, Y, θ) e como entrada a velocidade enviada. A velocidade medida não foi considerada devido ao ruído presente nas leituras.

Os modelos foram comparados em relação á capacidade de previsão de 1, 5, 10 e 20 passos à frente, lembrando que cada passo equivale à um período de amostragem.

4.4.1 Experimentos Realizados

Como dito no Capítulo 3, os modelos serão comparados utilizados dados de três experimentos distintos. Com um conjunto de dados foi feita a identificação dos modelos e com os outros dois foi feita a comparação. É importante ressaltar que o coeficiente de correlação entre os sinais de entrada de todos os experimentos é menor do que 0.05 em valor absoluto.

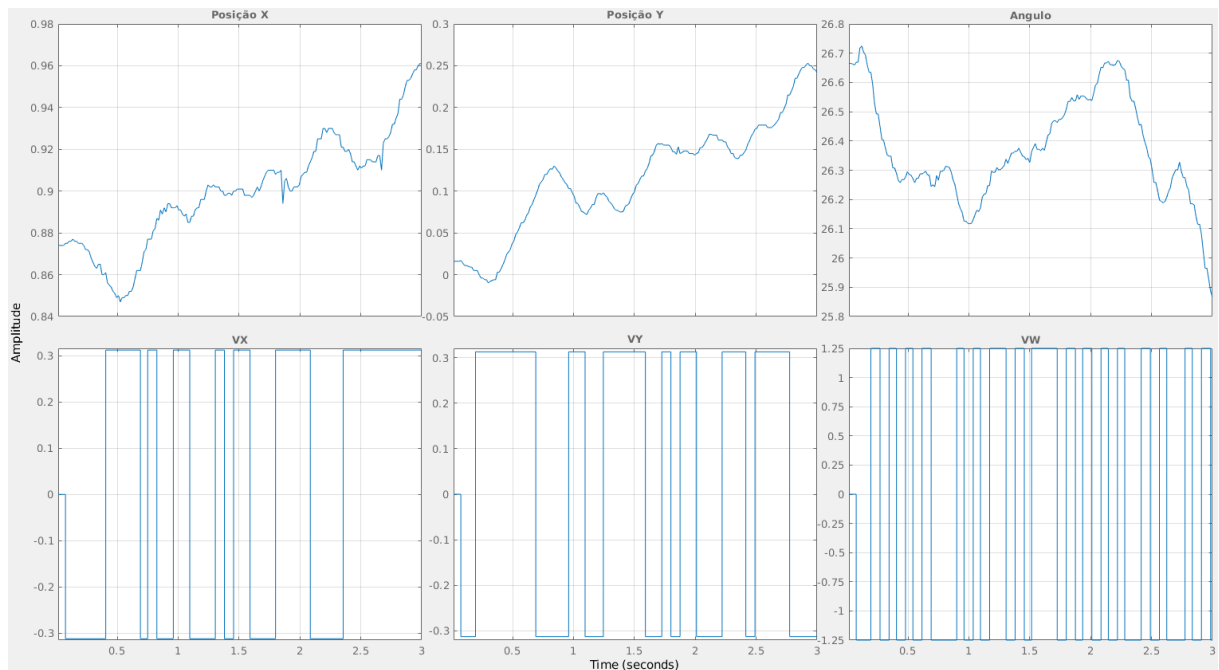
Os parâmetros dos sinais de entrada para cada experimento são:

- Experimento #1: Sinal PRBS, $T_b = 250 \text{ ms}$ e amplitudes para os eixos X , Y , θ de 0.25 m/s , 0.25 m/s , 1.85 rad/s , respectivamente.
- Experimento #2: Sinal PRBS, $T_b = 250 \text{ ms}$ e amplitudes para os eixos X , Y , θ de 0.5 m/s , 0.5 m/s , 1.85 rad/s , respectivamente.
- Experimento #3: Sinal GBN, probabilidades para os eixos X , Y , θ de 0.35 , 0.35 , 0.8 e amplitudes de 0.3 m/s , 0.3 m/s , 1.25 rad/s , respectivamente.

Todos os dados foram coletados utilizando um período de amostragem de 15 ms . Nos experimentos foi utilizado o robô real no laboratório da equipe RoboFEI, os testes foram feitos em sequência utilizando o mesmo robô.

Para geração e envio dos sinais de excitação foi utilizado o *software* de aquisição de dados desenvolvido (Seção 4.3). O robô foi posicionado no centro do campo, de forma a evitar colisões com as paredes, e então os sinais devidamente parametrizados foram enviados para o robô. A Figura 23 ilustra os três primeiros segundos do Experimento #3, os sinais gerados são enviados simultaneamente para ambos os eixos do robô.

Figura 23 – Início do Experimento #3.



Fonte: Autor.

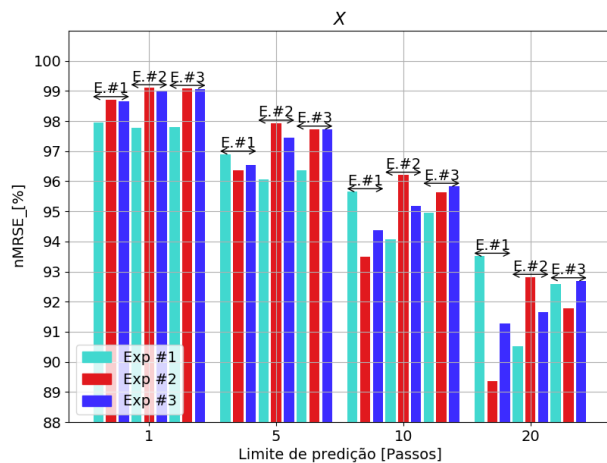
4.4.2 ARX

Para que as Figuras 17, 16, 19 e 18 não ficassem congestionadas, foram omitidas as ordens dos modelos utilizados, porém, o modelo ARX que obteve os melhores resultados foi o

ARX(6,5) com atraso 6. Portanto, os parâmetros para o modelo ARX que serão utilizados até o fim do trabalho são $n_a = 6$, $n_b = 5$ e $n_k = 6$.

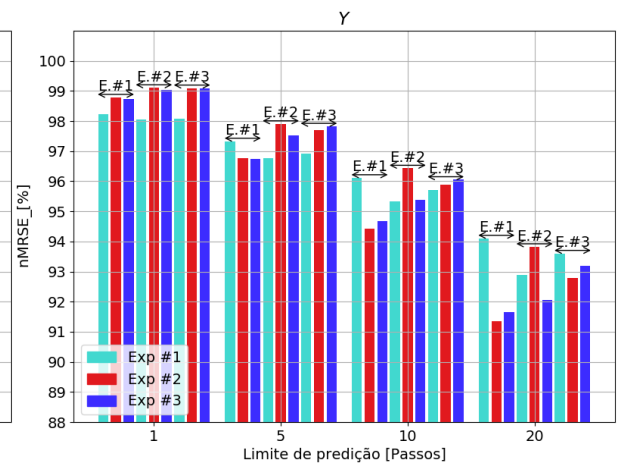
Para avaliar a performance do modelo ARX, foi feita a validação cruzada utilizando os três experimentos realizados, ou seja, o modelo foi gerado a partir de um conjunto de dados e testado nos outros dois. Esse processo foi repetido três vezes, variando o experimento utilizado na identificação. As Figuras 24, 25 e 26 mostram o desempenho do modelo ARX utilizando o nMRSE como métrica. A Figura 27 mostra a mesma comparação, porém, utilizando o *MRSE* como métrica. As setas indicam qual experimento foi utilizado para identificar o modelo que gerou tal resposta, por exemplo, E.#1 significa que os dados do experimento um foram utilizados para identificar o modelo.

Figura 24 – Comparação entre os modelos gerados para o eixo *X* utilizando o nMRSE.



Fonte: Autor.

Figura 25 – Comparação entre os modelos gerados para o eixo *Y* utilizando o nMRSE.



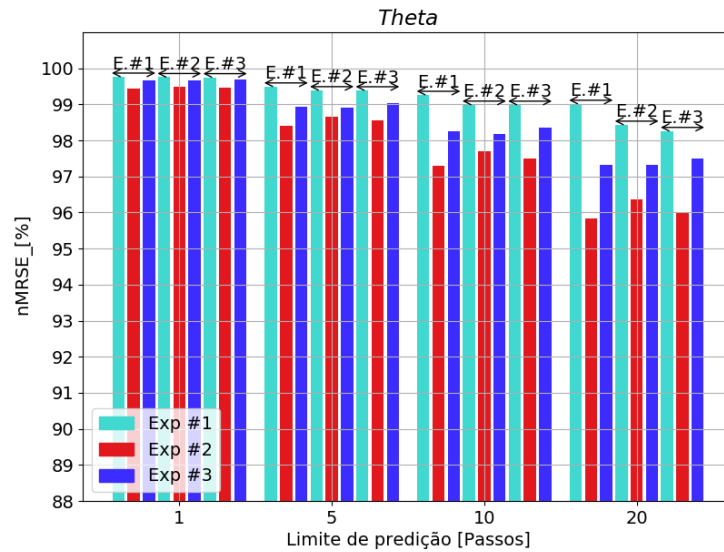
Fonte: Autor.

Analisando as Figuras 24, 25 e 26 fica evidente que a precisão do modelo decai conforme o limite de predição aumenta, este é um resultado óbvio. Entretanto, nota-se que no geral, esta perda de precisão não chega a superar cerca de 7% para os eixos *X*, *Y* e cerca de 4% para o eixo θ , no nMRSE.

A Figura 27 é um pouco mais simples de analisar, pois, o MRSE é calculado para o modelo como um todo. Nota-se também que o MRSE aumenta de acordo com a quantidade de passos de predição, porém, esse aumento fica em torno de 2.5% para todos os modelos.

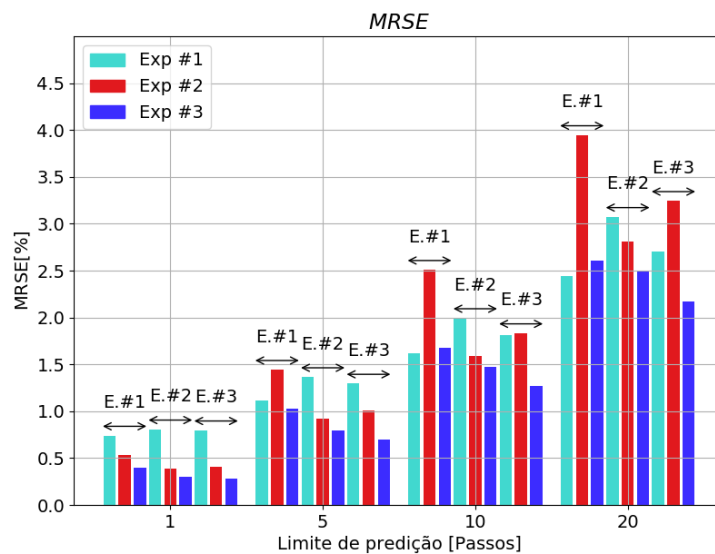
Outro ponto interessante que é possível observar é que o modelo gerado a partir do E.#1 não obteve uma boa performance ao ser utilizado nos E.#2 e E.#3. Isso ocorre muito provavelmente devido à incapacidade do modelo de generalizar a resposta em amplitudes mais altas do que a utilizada na identificação do mesmo. Esse efeito foi observado num experimento em que o modelo foi estimado utilizando dados com aproximadamente 0.1 de amplitude para todos os eixos, e então testado em campo com o robô se movimentando à aproximadamente 1.5 a 2 *m/s*, nesse caso foi possível notar um erro elevado na resposta do modelo. Essa falta de precisão pode

Figura 26 – Comparação entre os modelos gerados para o eixo θ utilizando o nMRSE.



Fonte: Autor.

Figura 27 – Comparação entre os modelos gerados utilizando o MRSE.



Fonte: Autor.

ocorrer devido à excitação de características não lineares do sistema que aparecem em baixas velocidade, como é o caso das zonas mortas. De acordo com Aguirre (2004), é importante que características não lineares não sejam excitadas quando deseja-se um modelo linear.

Das Figuras 24, 25 e 26, o modelo estimado com os dados do E.#3 é o que possui melhor nMRSE para os dados dos outros dois experimentos. Na Figura 27 os modelos estimados a partir dos experimentos #2 e #3 apresentam um MRSE próximo para dados diferentes do utilizado na identificação.

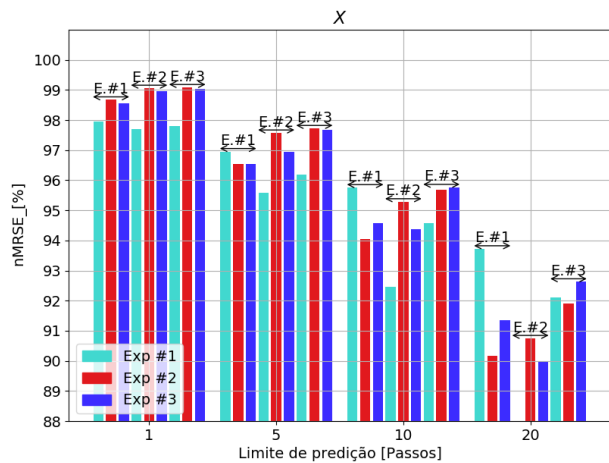
Dessa forma, o melhor modelo ARX obtido foi o estimado a partir dos dados do experimento número três. Este modelo pode ser visto no Apêndice A.

4.4.3 ARMAX

De forma similar ao modelo ARX, a partir dos testes realizados nas Figuras 17, 16, 19 e 18 o melhor modelo ARMAX foi o ARMAX(3,2,2) com atraso 5. Portanto, os parâmetros para o modelo ARMAX que serão utilizados até o fim do trabalho são $n_a = 3$, $n_b = 2$, $n_c = 2$ e $n_k = 5$.

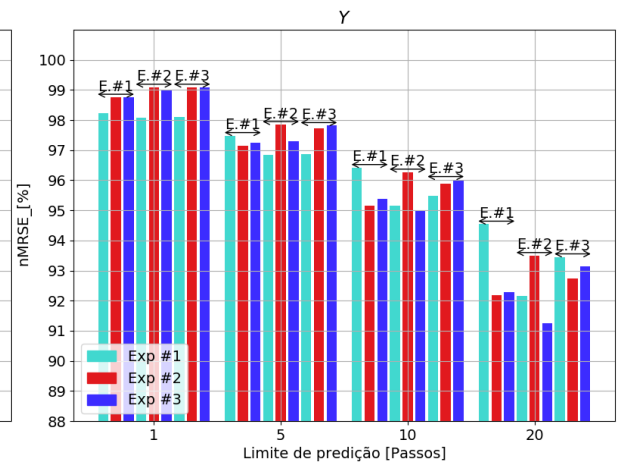
Os modelos ARMAX também foram analisados utilizando a validação cruzada. O resultado disso pode ser visto nas Figuras 28, 29 e 30 utilizando o nMRSE como métrica, e na Figura 31 utilizando o MRSE como métrica.

Figura 28 – Comparação entre os modelos gerados para o eixo X utilizando o nMRSE.



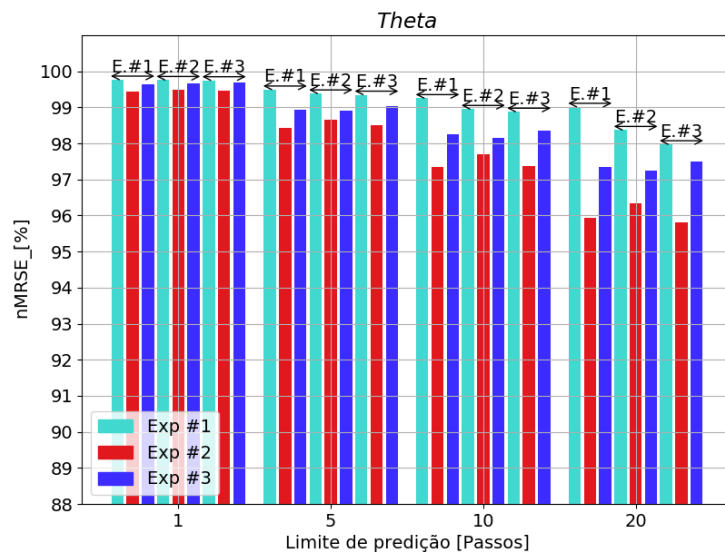
Fonte: Autor.

Figura 29 – Comparação entre os modelos gerados para o eixo Y utilizando o nMRSE.



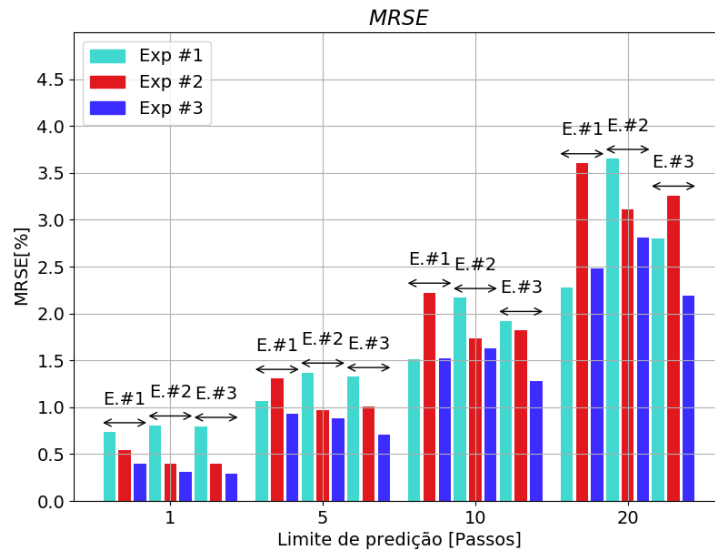
Fonte: Autor.

Figura 30 – Comparação entre os modelos gerados para o eixo θ utilizando o nMRSE.



Fonte: Autor.

Figura 31 – Comparação entre os modelos gerados utilizando o MRSE.



Fonte: Autor.

Analisando as Figuras 28, 29 e 30, nota-se que a precisão dos modelos decai ao aumentar o limite de predição, para os eixos X , Y e θ houve um decréscimo de cerca de 7%, 6% e 3%, respectivamente, no nMRSE.

A Figura 31 também demonstra este aumento do erro de acordo com o limite de predição. Para o MRSE, o aumento do erro é de cerca de 3%.

De acordo com as Figuras 28, 29 e 30, o experimento três foi o que gerou o modelo com melhor performance no dados dos outros experimentos no geral. No MRSE (Figura 31), os resultados são bem próximos, porém, a performance dos modelos identificados com o experimento três foi um pouco superior à dos outros experimentos.

Dessa forma, o melhor modelo ARMAX obtido foi o estimado a partir dos dados do experimento número três. Este modelo pode ser visto no Apêndice A.

4.4.4 Comparação Entre os Modelos ARX e ARMAX Finais

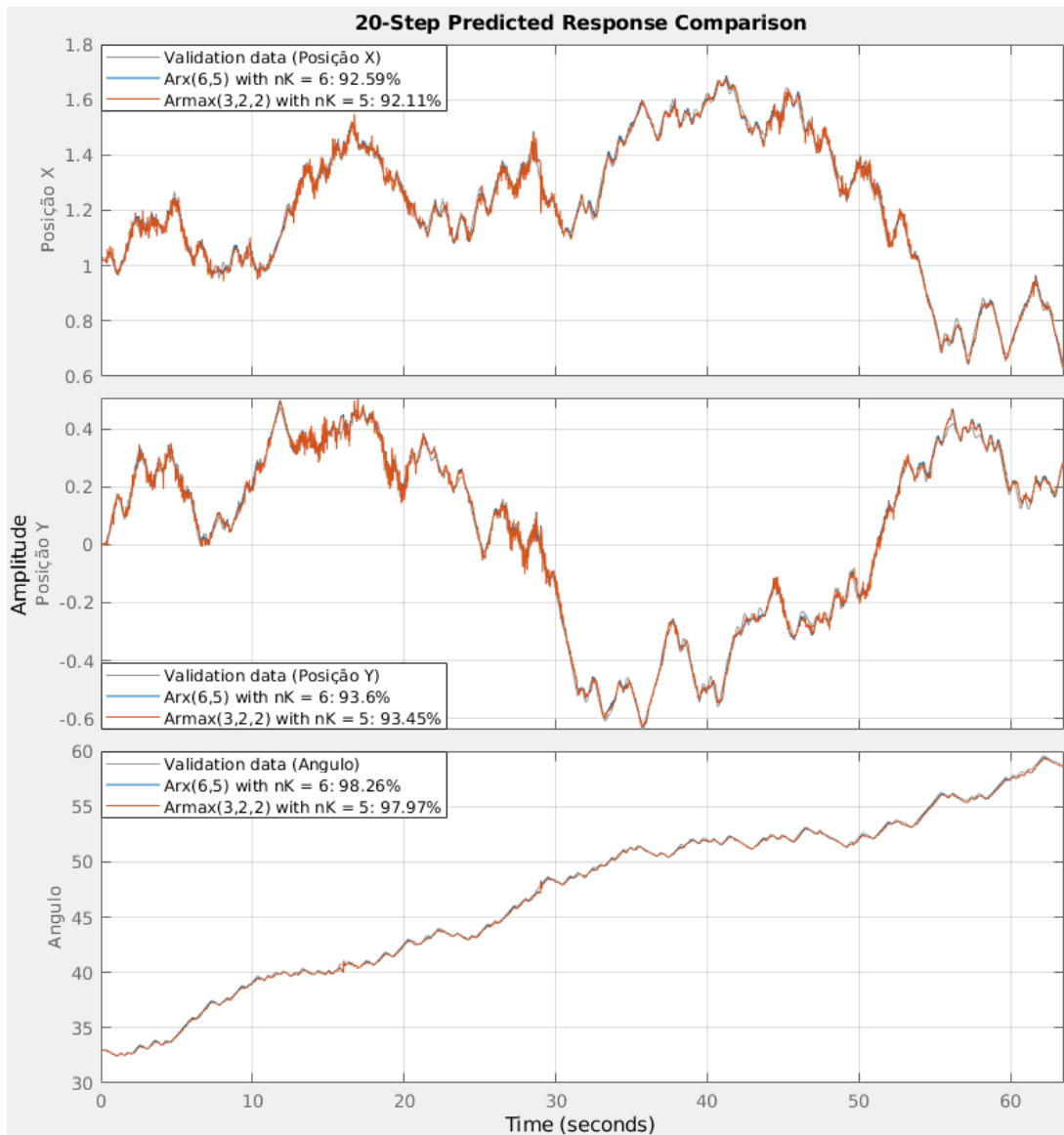
Como visto nas Seções 4.4.2 e 4.4.3, a partir dos gráficos gerados foi possível determinar os modelos que melhor representam o comportamento do robô. Esses modelos são aqueles que apresentaram um pequeno erro para dados desconhecidos, i.e., que não fizeram parte do processo de identificação.

Enfim, como análise final basta decidir qual o formato é mais adequado para ser utilizado.

Nas Figuras 32 e 33 é possível observar a resposta dos modelos identificados a partir do experimento três nos outros dois experimentos.

As simulações foram feitas utilizando a predição de 20 passos, pois, este é o ponto em que a precisão dos modelos começa a decair significativamente.

Figura 32 – Predição de 20 passos à frente utilizando os dados do experimento um como entrada.

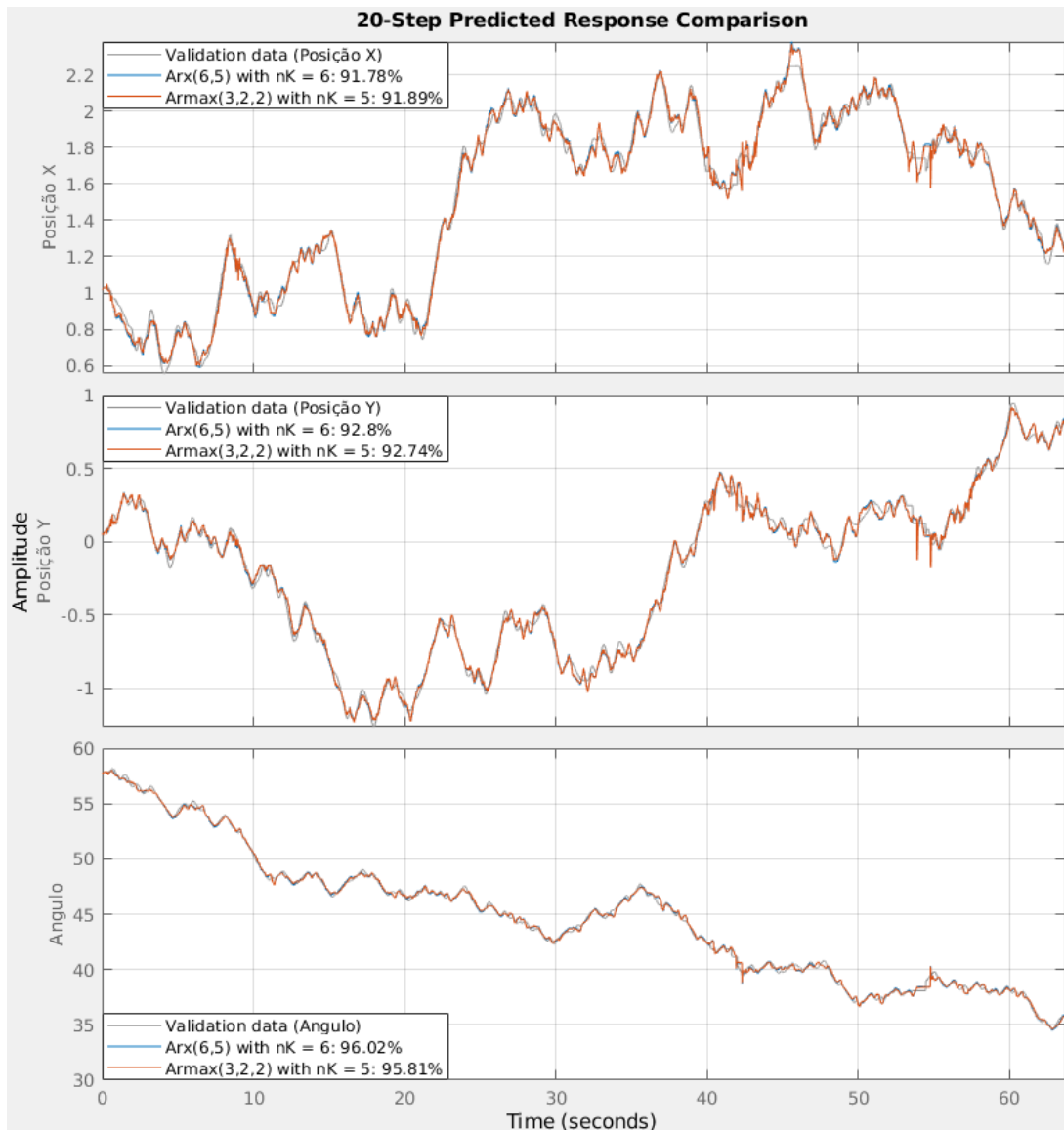


Fonte: Autor.

Na Figura 32, nota-se um ruído considerável que não aparece na Figura 33, isto ocorre devido à possível presença de não linearidades que o robô apresenta em baixas velocidades. Como na maioria das vezes os robôs estão se movimentando pelo campo em velocidades superiores à 1 m/s , isto não é um problema.

Analisando as Figuras 24 a 33 é possível afirmar que a precisão dos modelos ARX e ARMAX são muito próximas no geral, são poucos casos em que um modelo tem mais de 1% de diferença para o outro, tanto no nMRSE quanto no MRSE. Portanto, de acordo com as métricas estabelecidas, ambos os modelos são capazes de representar fielmente o comportamento do robô. Uma forma de desempatar esta escolha é levar em conta a complexidade do modelo. Embora o modelo ARX apresente praticamente a mesma performance do modelo ARMAX, a

Figura 33 – Predição de 20 passos à frente utilizando os dados do experimento dois como entrada.



Fonte: Autor.

sua complexidade é muito maior, seus polinômios tem ordens de 6 a 5, enquanto que o modelo ARMAX possui polinômios de ordem 3 a 2. Dessa forma, o modelo mais apropriado para uso é o ARMAX, devido ao seu alto nMRSE, baixo MRSE e baixa ordem dos polinômios A,B e C.

4.5 CONTROLADOR DE POSIÇÃO UTILIZANDO OS MODELOS DE FT ESTIMADOS

Ao utilizar os modelos ARX e ARMAX é possível modelar de forma satisfatória diversos tipos de processos, porém, para projetar um controlador utilizando este tipo de modelo não é uma tarefa tão simples quanto a identificação dos mesmos. Por esse motivo, o terceiro formato de modelo escolhido foi a FT. Embora não seja possível modelar o robô como sendo um

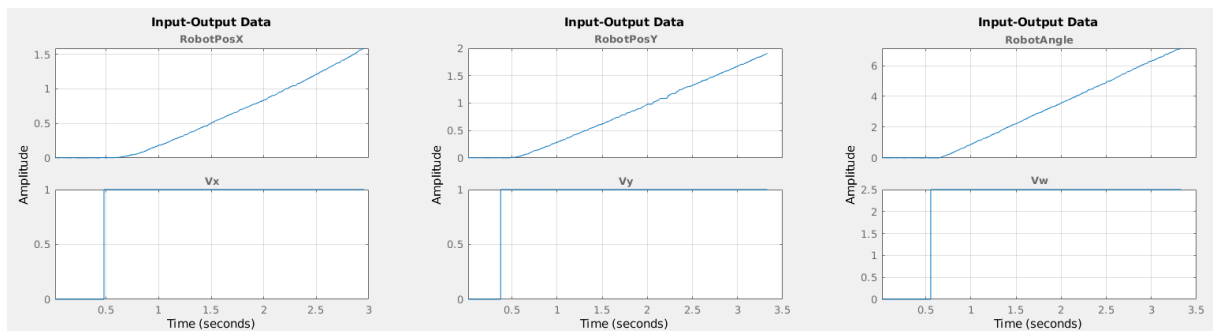
sistema MIMO, o uso da FT traz grande facilidade na hora de projetar controladores simples, mas eficazes.

4.5.1 Sinal de Entrada

O sinal de entrada escolhido para a identificação das funções de transferência é o degrau. Embora ele seja considerado um sinal pobre de informações para a identificação de um modelo, no caso da FT é possível estimar seus parâmetros com boa precisão utilizando este tipo de sinal (XU; CHEN; XIONG, 2015; AGUIRRE, 2004).

Na Figura 34, é possível ver os dados obtidos após realizar a resposta ao degrau para cada um dos eixos do robô. De forma similar aos modelos ARX e ARMAX, para as funções de transferência, a entrada é a velocidade que o robô deve se movimentar e a saída é a sua posição no campo.

Figura 34 – Resposta ao degrau para cada eixo do robô.



Fonte: Autor.

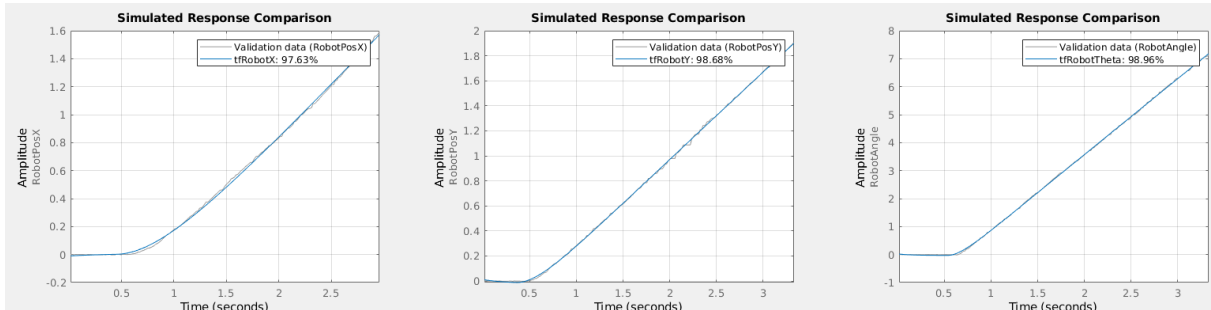
4.5.2 Identificação das Funções de Transferência

Um cuidado especial que deve ser tomado ao estimar modelos de FT é que é importante que os dados iniciem sempre do zero, ou seja, as condições iniciais devem ser nulas. Embora o MATLAB tenha a opção de utilizar condições iniciais não nulas, será mais simples de trabalhar com o modelo na hora de projetar o controlador caso as mesmas sejam nulas.

Após realizar a resposta ao degrau e ajustar os dados obtidos, para identificar os modelos basta definir quantidade de polos e zeros que a FT irá possuir. De forma similar à escolha da ordem dos polinômios nos modelos ARX e ARMAX, esta tarefa consiste de tentativa e erro até encontrar uma quantidade adequada de polos e zeros que represente de forma satisfatória o sistema. Porém, de forma geral, os sistemas costumam possuir entre um e três polos no máximo. Com isso em mente, realizando poucos testes foi possível determinar que ambos os eixos do robô podem ser modelados como sendo funções de transferência de dois polos e nenhum zero.

Na Figura 35, é possível observar a simulação do modelo obtido para cada eixo. Nota-se que para ambos os eixos os modelos obtiveram um $nRMSE$ superior à 97%.

Figura 35 – Simulação dos modelos de FT obtidos.



Fonte: Autor.

Nas Equações (17), (18) e (19) é possível ver as funções de transferência estimadas para cada eixo do robô.

$$\frac{P_x(z)}{V_x(z)} = \frac{0.0003494}{z^2 - 1.97z + 0.97} \quad (17)$$

$$\frac{P_y(z)}{V_y(z)} = \frac{0.0006378}{z^2 - 1.94z + 0.9399} \quad (18)$$

$$\frac{P_\theta(z)}{V_\theta(z)} = \frac{0.001623}{z^2 - 1.901z + 0.9009} \quad (19)$$

4.5.3 Projeto do Controlador de Posição

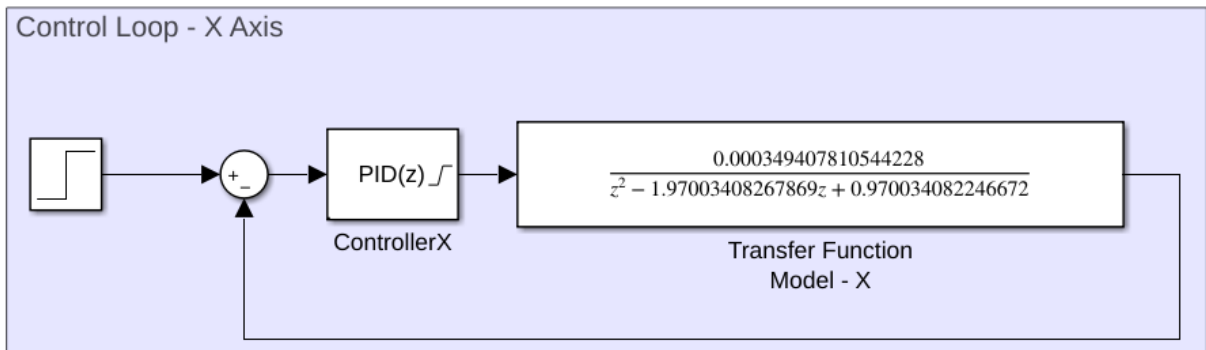
Com as funções de transferência estimadas é possível realizar o projeto dos controladores de posição do robô.

O controlador a ser projetado é o PID. Ele foi escolhido por ser um controlador robusto e por ser facilmente calibrado utilizando o MATLAB (OGATA; YANG, 2002).

Para projetar o controlador, a maneira mais simples é montar o diagrama de blocos do sistema no *Simulink*. O diagrama feito para o eixo X pode ser visto na Figura 36, para os outros eixos o diagrama é o mesmo, com exceção dos parâmetros da FT.

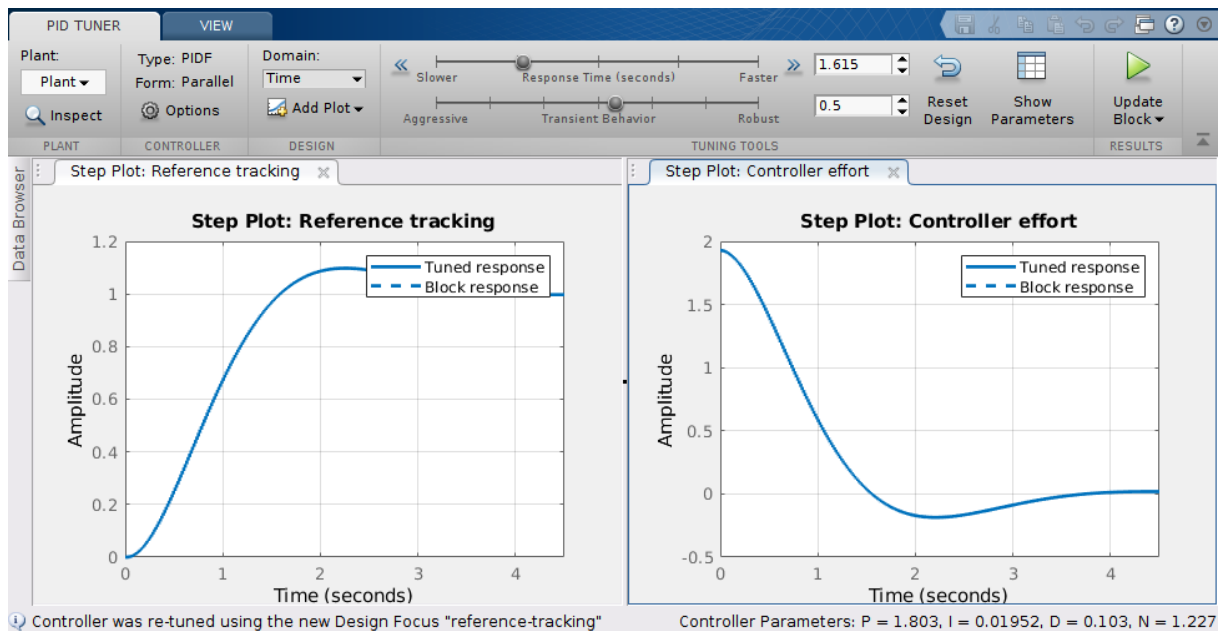
A partir do diagrama de blocos, basta utilizar a opção *Tune* do bloco do controlador PID para abrir a ferramenta *PIDTuner*. Com essa ferramenta é possível calibrar o controlador apenas ajustando o tempo de resposta desejado do sistema e o comportamento no transiente, isso pode ser visto na Figura 37.

Figura 36 – Diagrama de blocos utilizado no projeto do controlador para o eixo X .



Fonte: Autor.

Figura 37 – *PIDTuner*.



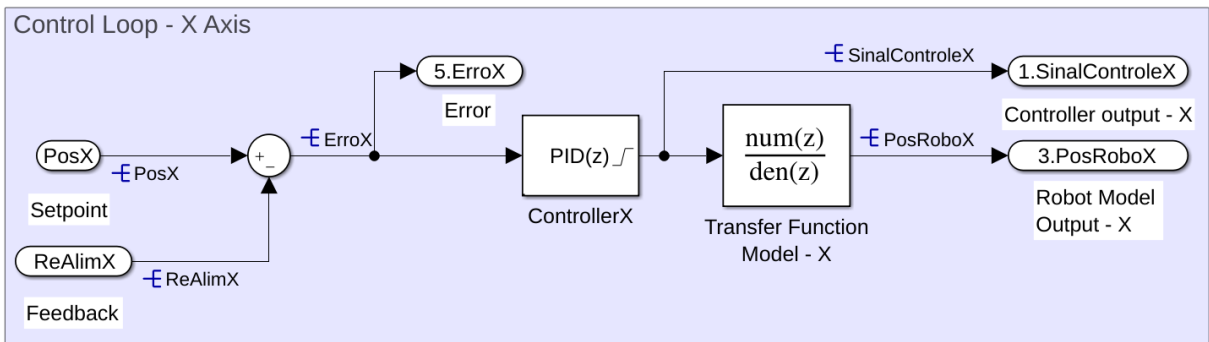
Fonte: Autor.

4.5.4 Implementação do Controlador

Após finalizado o projeto dos três controladores, com alguns ajustes no diagrama de blocos é possível exportá-lo para a linguagem $C++$ utilizando o próprio *Simulink*. Dessa forma o controlador poderá ser utilizado no software de estratégia da equipe.

O diagrama de blocos adaptado para o eixo X pode ser visto na Figura 38.

Figura 38 – Diagrama de blocos finalizado para o eixo X.

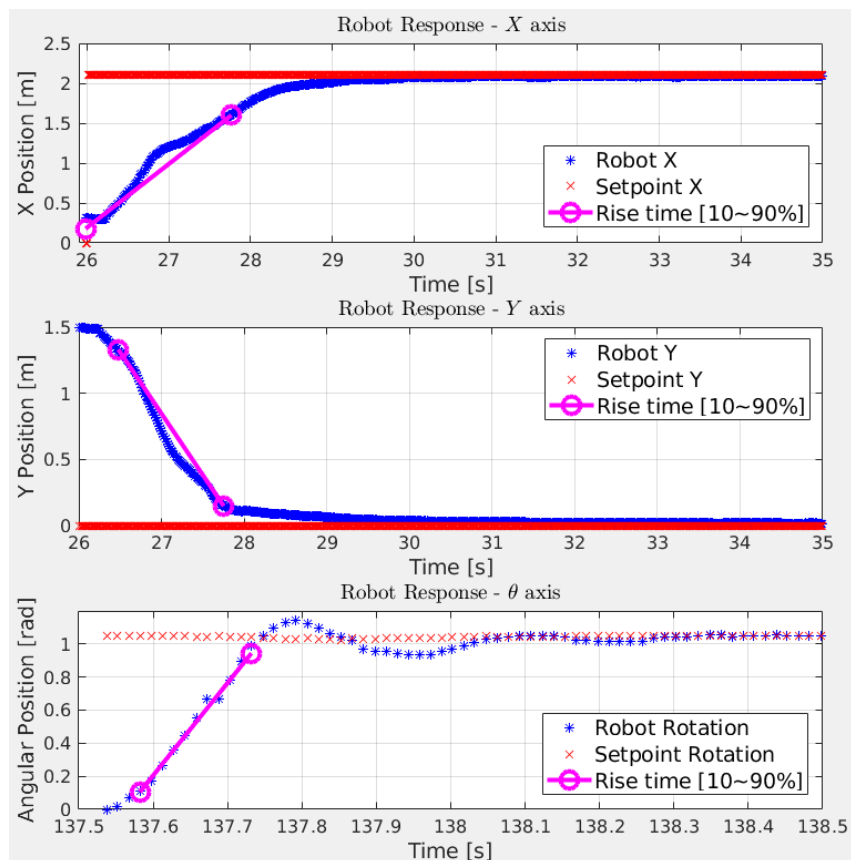


Fonte: Autor.

4.5.5 Resultados

Com o novo controlador de posição a melhora na precisão dos robôs é visivelmente notável. Devido à ação integral do PID é possível zerar o erro em estado estacionário. A resposta ao degrau para ambos os eixos do robô pode ser vista na Figura 39.

Figura 39 – Resposta ao degrau para os eixos do robô.



Fonte: Autor.

Também é importante ressaltar que devido à capacidade do controlador lidar com pequenas variações nos parâmetros do sistema, o mesmo controlador funcionou satisfatoriamente para todos os robôs da equipe. Embora com o processo de identificação de sistemas seja possível possuir um controlador personalizado para cada robô, isso não parece ser necessário.

Embora o modelo de FT possa não ser mais preciso que os modelos ARX e ARMAX, ele ainda assim é capaz de fornecer bons resultados no projeto de controladores clássicos, que neste caso, foi uma melhora significativa.

Vale a pena lembrar que, no caso do projeto de um *Model Predictive Control* (MPC), a capacidade de previsão do estado futuro do sistema é crucial, portanto, os modelos ARX e ARMAX são mais adequados para esta aplicação.

5 CONCLUSÃO

O objetivo deste trabalho foi desenvolver um software de aquisição de dados para a aquisição dos modelos dinâmicos de um robô móvel omnidirecional da categoria SSL e realizar a identificação deste modelo utilizando a técnica de identificação de sistemas. Para isso, foi feita a apresentação dos formatos de modelos clássicos da técnica de identificação de sistemas, que são: FT, ARX, ARMAX, OE e BJ, e dos tipos de sinais comumente utilizados, PRBS e GBN.

Utilizando o nMRSE e o MRSE foi possível analisar a performance de diversas parametrizações de modelos e de sinais de entrada, a fim de determinar qual a ordem de modelo e parametrização do sinal de entrada, de forma a obter um modelo com a melhor precisão possível.

A partir do estudo realizado foi definido que os modelos testados seriam: ARX, ARMAX e FT. Com isso, o modelo ARMAX mostrou-se como sendo o melhor para modelar o comportamento do robô.

O modelo de FT teve sua eficiência comprovada ao utilizá-lo no projeto de um controlador PID para a posição do robô, que proporcionou uma grande melhora no desempenho dos robôs da equipe RoboFEI.

APÊNDICE A – MODELO ARX

Este apêndice apresenta o modelo ARX obtido neste trabalho.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \times \begin{bmatrix} x(k) \\ y(k) \\ \theta(k) \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \times \begin{bmatrix} u_x(k) \\ u_y(k) \\ u_\theta(k) \end{bmatrix} + \begin{bmatrix} v_x(k) \\ v_y(k) \\ v_\theta(k) \end{bmatrix} \quad (20)$$

Onde,

$$a_{11} = 1 + (-885q^{-1} - 211.2q^{-2} - 129.9q^{-3} + 35.48q^{-4} + 94.06q^{-5} + 96.39q^{-6}) \times 1e-3$$

$$a_{12} = (-3.627q^{-1} - 6.891q^{-2} + 32.06q^{-3} - 21.35q^{-4} + 13.24q^{-5} - 13.32q^{-6}) \times 1e-3$$

$$a_{13} = (-0.8738q^{-1} + 0.4977q^{-2} - 1.596q^{-3} + 0.4087q^{-4} + 2.459q^{-5} - 0.89q^{-6}) \times 1e-3$$

$$a_{21} = (3.293q^{-1} - 11.11q^{-2} + 3.368q^{-3} - 26.81q^{-4} + 44.46q^{-5} - 12.9q^{-6}) \times 1e-3$$

$$a_{22} = 1 + (-880.1q^{-1} - 255.1q^{-2} - 64.01q^{-3} + 5.209q^{-4} + 114.7q^{-5} + 79.14q^{-6}) \times 1e-3$$

$$a_{23} = (-1.853q^{-1} + 2.451q^{-2} - 0.6699q^{-3} + 1.535q^{-4} + 0.5743q^{-5} - 2.047q^{-6}) \times 1e-3$$

$$a_{31} = (180.2q^{-1} - 14.53q^{-2} - 217.3q^{-3} + 174.2q^{-4} - 216.7q^{-5} + 93.35q^{-6}) \times 1e-3$$

$$a_{32} = (-95.2q^{-1} - 50.93q^{-2} + 195.8q^{-3} - 98.66q^{-4} + 185.9q^{-5} - 135.6q^{-6}) \times 1e-3$$

$$a_{33} = 1 + (-1123q^{-1} - 99.82q^{-2} + 93.6q^{-3} + 91.72q^{-4} + 40.44q^{-5} - 2.777q^{-6}) \times 1e-3$$

$$b_{11} = (0.6201q^{-6} + 0.9395q^{-7} + 0.8685q^{-8} + 1.479q^{-9} + 1.719q^{-10}) \times 1e-3$$

$$b_{12} = (-0.06138q^{-6} + 0.4715q^{-7} + 0.1788q^{-8} - 0.01466e - 02q^{-9} + 0.2167q^{-10}) \times 1e-3$$

$$b_{13} = (-0.0185q^{-6} - 0.01672q^{-7} + 0.06551q^{-8} + 0.008555q^{-9} - 0.05865q^{-10}) \times 1e-3$$

$$b_{21} = (0.3192q^{-6} - 0.03809q^{-7} - 0.05706q^{-8} + 0.2062q^{-9} - 0.09036q^{-10}) \times 1e-3$$

$$b_{22} = (0.7866q^{-6} + 0.446q^{-7} + 1.708q^{-8} + 1.527q^{-9} + 1.262q^{-10}) \times 1e-3$$

$$b_{23} = (-0.02034q^{-6} + 0.02992q^{-7} - 0.01636q^{-8} + 0.04124q^{-9} - 0.05365q^{-10}) \times 1e-3$$

$$b_{31} = (0.8151q^{-6} + 1.802q^{-7} - 1.527q^{-8} + 2.741q^{-9} - 1.456q^{-10}) \times 1e-3$$

$$b_{32} = (0.2854q^{-6} + 1.733q^{-7} - 3.848q^{-8} + 1.635q^{-9} - 0.1987q^{-10}) \times 1e-3$$

$$b_{33} = (2.228q^{-6} + 1.693q^{-7} + 1.636q^{-8} + 0.8824q^{-9} + 0.1856q^{-10}) \times 1e-3$$

APÊNDICE B – MODELO ARMAX

Este apêndice apresenta o modelo ARMAX obtido neste trabalho.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \times \begin{bmatrix} x(k) \\ y(k) \\ \theta(k) \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \times \begin{bmatrix} u_x(k) \\ u_y(k) \\ u_\theta(k) \end{bmatrix} + \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} \times \begin{bmatrix} v_x(k) \\ v_y(k) \\ v_\theta(k) \end{bmatrix} \quad (21)$$

Onde,

$$\begin{aligned} a_{11} &= 1 - 2.238q^{-1} + 1.577q^{-2} - 0.339q^{-3} \\ a_{12} &= -0.009126q^{-1} + 0.02219q^{-2} - 0.01303q^{-3} \\ a_{13} &= 0.01187q^{-1} - 0.01945q^{-2} + 0.007578q^{-3} \\ a_{21} &= -0.01716q^{-1} + 0.02858q^{-2} - 0.01137q^{-3} \\ a_{22} &= 1 - 2.307q^{-1} + 1.709q^{-2} - 0.4021q^{-3} \\ a_{23} &= -0.009473q^{-1} + 0.02006q^{-2} - 0.01059q^{-3} \\ a_{31} &= 0.3336q^{-1} - 0.6256q^{-2} + 0.2918q^{-3} \\ a_{32} &= -0.12q^{-1} + 0.2246q^{-2} - 0.1041q^{-3} \\ a_{33} &= 1 - 2.372q^{-1} + 1.888q^{-2} - 0.5161q^{-3} \\ \\ b_{11} &= -0.0007504q^{-5} + 0.001807q^{-6} \\ b_{12} &= -0.0001105q^{-5} + 0.0002622q^{-6} \\ b_{13} &= -5.317e - 05q^{-5} + 8.376e - 05q^{-6} \\ b_{21} &= 6.64e - 05q^{-5} - 1.362e - 05q^{-6} \\ b_{22} &= -0.0005729q^{-5} + 0.001574q^{-6} \\ b_{23} &= 1.179e - 05q^{-5} - 2.555e - 05q^{-6} \\ b_{31} &= 0.0005984q^{-5} + 0.0001707q^{-6} \\ b_{32} &= 0.0003465q^{-5} - 0.0003667q^{-6} \\ b_{33} &= 0.000616q^{-5} + 0.001062q^{-6} \\ \\ c_1 &= 1 - 1.325q^{-1} + 0.5609q^{-2} \\ c_2 &= 1 - 1.397q^{-1} + 0.6124q^{-2} \\ c_3 &= 1 - 1.221q^{-1} + 0.5131q^{-2} \end{aligned}$$

REFERÊNCIAS

- AGUIRRE, Luis Antonio. **Introdução à identificação de sistemas–Técnicas lineares e não-lineares aplicadas a sistemas reais**. [S.l.]: Editora UFMG, 2004.
- BEER, Ferdinand Pierre; JOHNSTON, Jr. **ER Mecânica Vetorial para Engenheiros, v. 1 e 2, São Paulo: Makron**. [S.l.]: McGraw-Hill, Brasil, 1991.
- HABER, R. Nonlinearity tests for dynamic processes. **IFAC Proceedings Volumes**, Elsevier, v. 18, n. 5, p. 409–414, 1985.
- HIPEL, Keith William; MCLEOD, Angus Ian; LENNOX, William C. Advances in Box-Jenkins modeling: 1. Model construction. **Water Resources Research**, Wiley Online Library, v. 13, n. 3, p. 567–575, 1977.
- HOSSEINI, Seyed M; JOHANSEN, Tor A; FATEHI, Alireza. Comparison of nonlinearity measures based on time series analysis for nonlinearity detection, 2011.
- LJUNG, Lennart. System identification. In: SIGNAL analysis and prediction. [S.l.]: Springer, 1998. P. 163–173.
- _____. **System identification: theory for the user**. [S.l.]: Prentice-hall, 1987.
- MCLEOD, AI. Improved Box–Jenkins estimators. **Biometrika**, Oxford University Press, v. 64, n. 3, p. 531–534, 1977.
- MENDES, Ellon Paiva. **Identificação em tempo real de modelo dinâmico de robô móvel com acionamento diferencial e zona morta**. 2012. Diss. (Mestrado) – Universidade Federal do Rio Grande do Norte.
- OGATA, Katsuhiko; YANG, Yanjuan. **Modern control engineering**. [S.l.]: Prentice hall India, 2002. v. 4.
- ORENSTEIN, Leonardo Poubel. **Procedimento para identificação de sistemas dinâmicos em ambiente industrial**. 2013. Tese (Doutorado) – Dissertação (Mestrado) Universidade Federal do Rio de Janeiro-UFRJ.
- PEREIRA, GAS. Identificação e controle de micro-robôs móveis. **Belo Horizonte, UFMG**, 2000.
- RAIMUNDO, Augusto dos Santos. Simulação de um controlador preditivo baseado em modelo com termo integral, 2013.

SRINIWAS, G Ravi et al. Nonlinear identification and control of a high-purity distillation column: a case study. **Journal of Process Control**, Elsevier, v. 5, n. 3, p. 149–162, 1995.

TANGIRALA, Arun K. **Principles of system identification: theory and practice**. [S.l.]: Crc Press, 2014.

TULLEKEN, Herbert JAF. Generalized binary noise test-signal concept for improved identification-experiment design. **Automatica**, Elsevier, v. 26, n. 1, p. 37–49, 1990.

XU, Ling; CHEN, Lei; XIONG, Weili. Parameter estimation and controller design for dynamic systems from the step responses based on the Newton iteration. **Nonlinear Dynamics**, Springer, v. 79, n. 3, p. 2155–2163, 2015.

YUWANA, Minta; SEBORG, Dale E. A new method for on-line controller tuning. **AIChE Journal**, Wiley Online Library, v. 28, n. 3, p. 434–440, 1982.

ZICKLER, Stefan et al. CMDragons 2009 extended team description. In: PROC. 14th International RoboCup Symposium, Singapore. [S.l.: s.n.], 2010.