

Comparação de métodos lineares e não-lineares para identificação de sistemas com materiais piezoelétricos e transmissão acústica

Daniel Pereira da Costa

Comparação de métodos lineares e não-lineares para identificação de sistemas com materiais piezoelétricos e transmissão acústica

Aluno: Daniel Pereira da Costa

Orientador: Helon Vicente Hultmann Ayala

Trabalho apresentado como requisito parcial à conclusão do curso de Engenharia de Controle e Automação na Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, Brasil.

Agradecimentos

Gostaria de agradecer a minha mãe, pai e irmão por sempre me apoiar em relação os estudos, me motivando cada vez mais a continuar sempre aprendendo.

Agradeço ao meu orientador, Helon Vicente Hultmann Ayala, por ter me orientado ao longo do projeto e por se mostrar sempre disposto a ajudar.

Gostaria de agradecer também ao meu amigo Matheus Patrick Soares Barbosa, por me dar apoio durante todo o meu projeto, me ajudando na parte técnica, teórica e com a escrita dessa dissertação.

Resumo

Materiais piezoelétricos são materiais capazes de produzir uma corrente elétrica quando submetidos a um estresse mecânico. Por outro lado, esses materiais são fisicamente deformados quando um campo elétrico é aplicado sobre eles. Nesse trabalho, os piezoelétricos são usados para a aquisição de dados através de um túnel acústico e assim aplicar técnicas de identificação de sistemas para o processo de modelagem do sistema. Este trabalho tem como objetivo a identificação de um sistema de transmissão acústica através de métodos de identificação de sistemas caixa-preta. Em específico, são usados os modelos lineares *AutoRegressive with eXogenous inputs* (ARX) e *AutoRegressive Moving Average with eXogenous inputs* (ARMAX) e modelos baseados em redes neurais do tipo *Nonlinear AutoRegressive with eXogenous inputs* (NARX). Esse projeto engloba todas as etapas de um processo de identificação de sistemas, desde a aquisição dos dados até a obtenção dos resultados. Os resultados obtidos são comparados com o modelo ARX e com o modelo ARMAX. A conclusão da predição mostra que o melhor resultado se obteve com o modelo ARMAX.

Palavras-chave: Transmissão piezoacústica, estruturas inteligentes, identificação não-linear, redes neurais artificiais

Comparison of linear and nonlinear methods for systems identification with piezoelectric materials and acoustic transmission

Abstract

Piezoelectric materials are materials capable of producing an electric current when subjected to mechanical stress. On the other hand, these materials are physically deformed when an electric field is applied to them. In this work, piezoelectrics are used for data acquisition through an acoustic tunnel and thus apply systems identification techniques to the system modeling process. This work aims to identify an acoustic transmission system through black box system identification methods. Specifically, the AutoRegressive with eXogenous Inputs (ARX) and AutoRegressive Moving Average with eXogenous Inputs (ARMAX) linear models and Nonlinear AutoRegressive with eXogenous Inputs (NARX) model with artificial neural network structure are used. This project encompasses all stages of a systems identification process, from data acquisition to results. The results obtained are compared with the ARX model and the ARMAX model. The prediction conclusion shows that the best result was obtained with the ARMAX model.

Keywords: Piezoacoustic transmission, intelligent structures, nonlinear identification, artificial neural networks

Lista de Figuras

1	Montagem do experimento	1
2	Esquema representando a configuração final do experimento.	3
3	Multisenso	4
4	Chirp	5
5	Square Sweep	5
6	Sinal de entrada e de saída	7
7	Sinal Chirp reamostrado (vermelho) vs original (azul)	8
8	Sinal Chirp reamostrado (vermelho) vs original (azul)	8
9	Plot da densidade spectral da entrada e saída dos sinais originais	9
10	Plot da densidade spectral da entrada e saída dos sinais reamostrados	9
11	Plot da densidade spectral da entrada e saída dos sinais reamostrados e filtrados	10
12	Sinal de entrada utilizado para a modelagem	10
13	Sinal de saída utilizado para a modelagem	11
14	Etapas do processo de Identificação de Sistemas	12
15	Estrutura interna de uma rede neural artificial	15
16	Camadas de uma rede neural artificial.	16
17	Estrutura de uma rede neural Cascade FeedForward.	17
18	Estrutura de uma rede neural wavelet.	18
19	Erro do melhor modelo ARX com OSA (azul) e FRS (vermelho).	20
20	Erro do melhor modelo ARMAX com OSA (azul) e FRS (vermelho).	21
21	Erro do modelo melhor NARX com OSA (azul) e FRS (vermelho).	21
22	Autorrelação dos Resíduos a esquerda e a correlação cruzada entre os resíduos e as entradas a direita.	22
23	Saída do modelo ARMAX OSA e a saída real	23
24	Saída do modelo ARMAX FR e a saída real	23

Lista de Tabelas

- 1 Todos os modelos ordenados por ordem crescente de R^2 em FRS. 22

Sumário

Lista de Figuras	iv
Lista de Tabelas	v
1 Introdução	1
2 Aquisição de dados	3
a Montagem do experimento	3
b Sinais de excitação	3
1 Multiseno	4
2 Chirp	4
3 Square Sweep	5
c Sinal de Estudo	6
3 Processamento dos Dados	7
a Método de Reamostragem	7
b Sinal Reamostrado	8
c Sinal para Modelagem	10
4 Identificação de Sistemas	12
a Identificação de Sistema Lineares	12
1 ARX	13
2 ARMAX	14
b Identificação de Sistemas Não lineares	14
1 NARX	15
c Validação do Modelo	18
1 OSA e FR	18
2 Métrica Quantitativa de Erro	19
5 Resultados	20
a ARX	20
b ARMAX	20
c NARX	21
d Resumo dos resultados	22
6 Conclusões & Trabalho Futuro	24
7 Referências	25
A Método de Reamostragem	26
B Geração do Sinal Reamostrado e Filtrado	28
C Código para iteração das ordens dos modelos ARMAX e ARX	29
D Código para a variação dos parâmetros do modelo NARX	30

1 Introdução

A piezoeletricidade é definida como a capacidade de alguns materiais se tornarem eletricamente polarizados quando sujeitos a algum tipo de stress. Mais precisamente, materiais piezoelétricos conseguem produzir corrente elétrica quando submetidos a esforços mecânicos. Esse processo também é reversível, e assim eles são fisicamente deformados quando submetidos a estresse mecânico. Graças a esse efeito, esse material possui diversas aplicações em diferentes áreas da engenharia, como a produção e detecção de som, geração de tensões elevadas, de frequências eletrônicas e diversas outras utilidades. Um dos usos desses atuadores corresponde à comunicação através de duas placas metálicas, representando as paredes de um contêiner metálico. Esse cenário se torna interessante em aplicações em que o contêiner não pode sofrer penetrações físicas e também quando outras técnicas através da parede, como acoplamento indutivo, capacitivo e de ressonância magnética, não são viáveis devido ao forte efeito Faraday [1] [2]. Nessa aplicação o material piezoelétrico se encontra posicionado na parede externa do contêiner, onde se gera uma onda acústica que se propaga através do material, também chamado de túnel acústico, que então é recuperada por um segundo piezo elétrico posicionado na parede interna. Uma vez que o sinal é captado na parede interna, ele pode ser usado para diverso fins, como por exemplo, para a transmissão de informações sobre o próprio fluido no interior do contêiner.

A composição física do túnel acústico influencia consideravelmente a eficiência da transmissão, e o uso de túneis com multicamadas, como metal e água o que acaba aumentando a complexidade matemática da modelagem. Em Identificação de Sistemas, o modelo caixa preta depende apenas dos sinais de entrada e saída, e não requer um conhecimento do funcionamento físico do sistema, o que o torna uma excelente técnica para essas aplicações.

Este projeto se foca na modelagem do tipo caixa preta de uma transmissão acústica multicamadas, em que os dois piezoelétricos se encontram imersos em água. Para isso, um experimento físico foi montado, conforme o *test bench* da Figura 1.



Figure 1: Montagem do experimento

Os dados foram obtidos sobre condições específicas pré-definidas pelo projeto. Com base nas aquisições realizadas, as próximas etapas consistiam em realizar um tratamento dos dados, fazer a construção do modelo e estudar a sua validação. Um pré-processamento dos dados foi necessário devido a existência de ruídos e também pelo fato do estudo do sistema ser realizado em apenas uma determinada banda de frequência, 900kHz e 1.35MHz, banda em que se observava a resposta com a melhor intensidade.

A abordagem proposta é implementada com três modelos de caixa preta. O objetivo principal era de criar um modelo capaz de identificar divergências no sistema quando ocorressem mudanças no meio e calcular a resposta de frequência offline ideal. O sistema foi excitado com três diferentes sinais, a fim de analisar o seu comportamento em diferentes condições. Dado a melhor resposta obtida, o desempenho de três diferentes modelos foi estudado, a fim de se analisar o melhor entre eles, com base na suas

complexidades. Apenas as entrada e saída do sistema foram utilizadas, sendo assim, o modelo pode ser usado para uma avaliação online da resposta de frequência ou para rastrear alterações no sistema quando aplicado ao monitoramento.

De posse dos dados medidos, a próxima etapa se deu na construção de um modelo não-linear capaz de reproduzir, de forma satisfatória, o sistema. Os modelos estudados possuíam tanto características lineares e também não-lineares: modelo *AutoRegressive with eXogenous inputs* (ARX), *AutoRegressive Moving Average with eXogenous inputs* (ARMAX), e o modelo baseado em redes neurais artificiais do tipo *Nonlinear AutoRegressive with eXogenous inputs* (NARX). Os resultados obtidos com cada um dos modelos foram listados com base na métrica escolhida e a ordem do modelo em questão.

Dentre as diversas estruturas matemáticas capazes de modelar este sistema, o projeto se focou no uso de redes neurais para identificação de sistemas não lineares NARX. Essa é uma classe poderosa de modelos que se mostrou muito eficiente para a modelagem de sistemas não lineares e especialmente, séries temporais. Os demais modelos foram utilizados como forma de comparação.

Este projeto se encontra dividido em quatro seções. Na seção 2, se encontra a explicação para cada um dos sinais utilizados para a excitação do sistema. Em seguida, em 3 o pré-processamento dos sinais de entrada e saída é explicado e detalhado. Na seção 4 os modelos estudados são exemplificados e descritos e na seção seguinte os seus resultados são analisados. Por último uma conclusão dos resultados com detalhes do trabalho futuro é apresentada.

2 Aquisição de dados

a Montagem do experimento

Os dados obtidos para a modelização do projeto foram obtidos através de um experimento criado para o estudo da transmissão acústica na água. Este último contava com a existência de dois piezoelétricos, um para a transmissão do sinal de entrada e o segundo para a captação do sinal de resposta. Esses atuadores apresentam algumas vantagens em relação a outros tipos de atuadores. Com materiais piezoelétricos, é possível gerenciar pequenos deslocamentos na faixa de sub nanômetros a várias centenas de micrômetros [3]. Os atuadores piezoelétricos podem ser menores e mais leves, em várias formas, em comparação com os atuadores tradicionais. Um aspecto importante é que o mesmo permite um controle de alta precisão e também de alta velocidade. Por outro lado, esses materiais apresentam uma desvantagem no processo de modelagem, devido a existência de histerese estática e dinâmica e as próprias vibrações.

Para gerar os sinais de excitação desejados necessitou-se do uso de um gerador de funções arbitrárias. Mais especificamente, se usou um *Tektronix AFG3022C*. O uso de um amplificador de sinal também se mostrou necessária, devido a baixa amplitude do sinal na saída do gerador. Por fim, para a visualização dos sinais de saída e de entrada, um osciloscópio foi utilizado. A montagem final dos equipamentos se encontra na Figura 2.

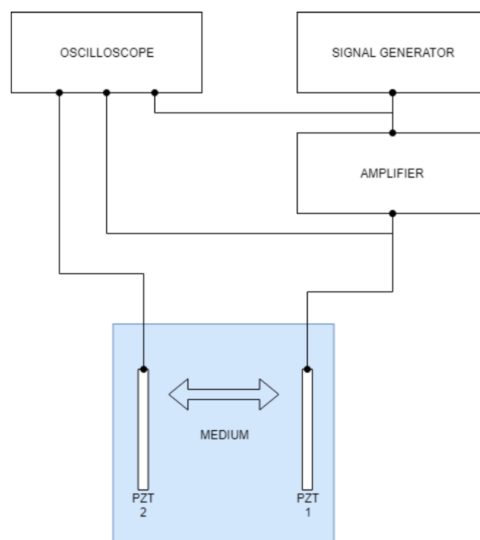


Figure 2: Esquema representando a configuração final do experimento.

O túnel acústico em questão é composto por dois atuadores piezoelétricos acoplados, cada um, a um lado das placas de aço de 5mm de espessura com uma camada de epóxi. As duas placas são espaçadas em 100mm e imersas em água destilada a temperatura ambiente. Dessa maneira, o sistema final é um composto de cinco camadas e três materiais (epóxi, aço e água) onde energia podem ser transmitida.

O sinal de excitação do modelo era gerado pelo equipamento *Tektronix AFG3022C* e seu espectro era analisado no osciloscópio a fim de verificar a sua consistência. Esse mesmo sinal, era amplificado e analisado no osciloscópio. Em seguida, se excitava o piezoelétrico que convertia o sinal elétrico em um sinal físico que era transmitido através da água destilada. O segundo atuador recebia este sinal, o convertia em sinal elétrico e então o seu espectro era observado no osciloscópio.

b Sinais de excitação

Os sinais de entrada do sistema são escolhidos com base no propósito do experimento de estudo. Os mesmos são gerados com a intenção de excitar o modelo em diversas banda de frequências, com diferentes amplitudes e analisar a sua resposta. O objetivo era de se estudar como o sistema respondia em diferentes cenários, a fim de se encontrar aquele que o sistema apresenta uma melhor resposta.

Como mencionado na primeira seção a banda de frequência de estudo já era conhecida: 900kHz-1.35Mhz. Três tipos sinais de excitação foram gerados nessa faixa, e, dentre eles, um foi o escolhido para se realizar o processo de modelagem.

1 Multiseno

O multiseno é uma soma de senoides harmonicamente relacionados. Pelo fato deste tipo de sinal variar a sua frequência ao longo de um período, ele permite realizar a medição da resposta em todo um conjunto de frequências.

Em formulação matemática o mesmo é representado pela equação 1, onde A_r representa a amplitude e ϕ_0 a fase.

$$u(t) = \sum_{k=1}^F A_r \cos(2\pi f_0 k t T_s + \phi_r) \quad (1)$$

Na Figura 3 se observa o multiseno de 400MVpp de amplitude, utilizado para um dos testes. O intervalo no tempo nessa figura foi escolhido com objetivo de facilitar a visualização dos dados.

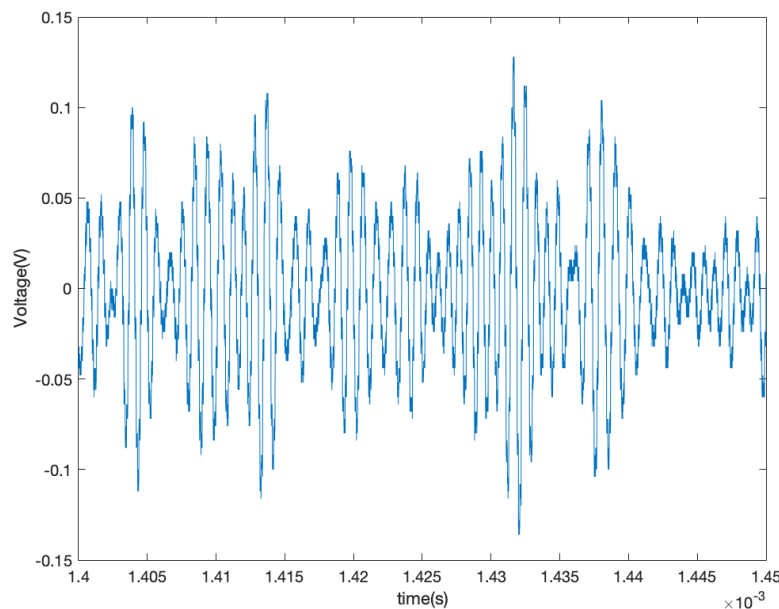


Figure 3: Multiseno

2 Chirp

O chirp é caracterizado por variar linearmente todas as frequências de uma determinada banda em um período. Devido a esta grande varredura, o mesmo apresenta uma grande energia. A função correspondente no domínio do tempo para um chirp linear senoidal é representado pela seguinte formulação matemática:

$$x(t) = \sin[\phi_0 + 2\pi(\frac{c}{2}t^2 + f_0 t)] \quad (2)$$

onde $c = \frac{f_1 - f_0}{T}$, f_1 e f_2 correspondem as frequência mínima e máxima do chirp, ϕ_0 a fase inicial do seno em $t = 0$.

Na Figura 4, o chirp utilizado foi criado variando o mesmo na banda de frequência em estudo, com uma amplitude de 80mVpp e com período de 100Ms. O intervalo no tempo da Figura 4 foi escolhido com objetivo de facilitar a visualização dos dados.

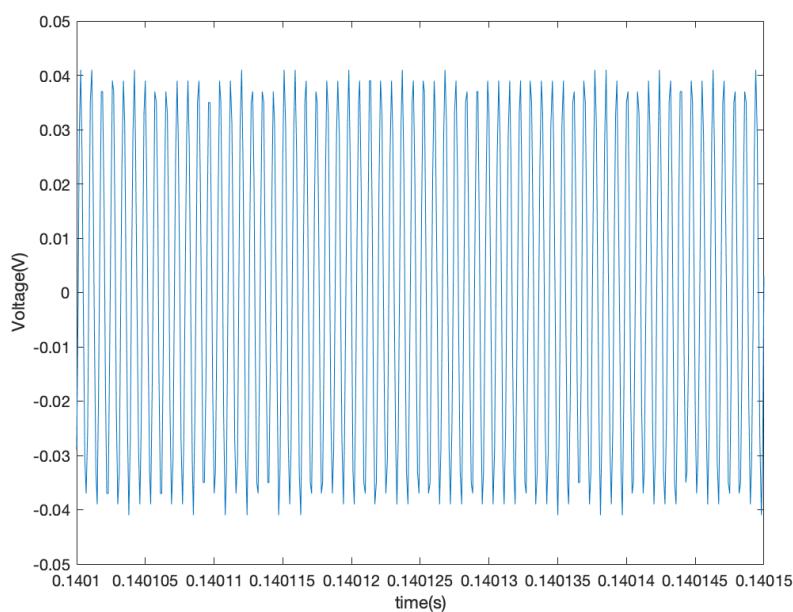


Figure 4: Chirp

3 Square Sweep

Seguindo a mesma lógica do sinal chirp, o square sweep é uma onda quadrada variando a frequência em uma determinada banda.

No teste realizado, uma das amplitudes utilizada é de 50mVpp, com um sweep de 100ms. O intervalo no tempo da figura 5 foi escolhido com objetivo de facilitar a visualização dos dados.

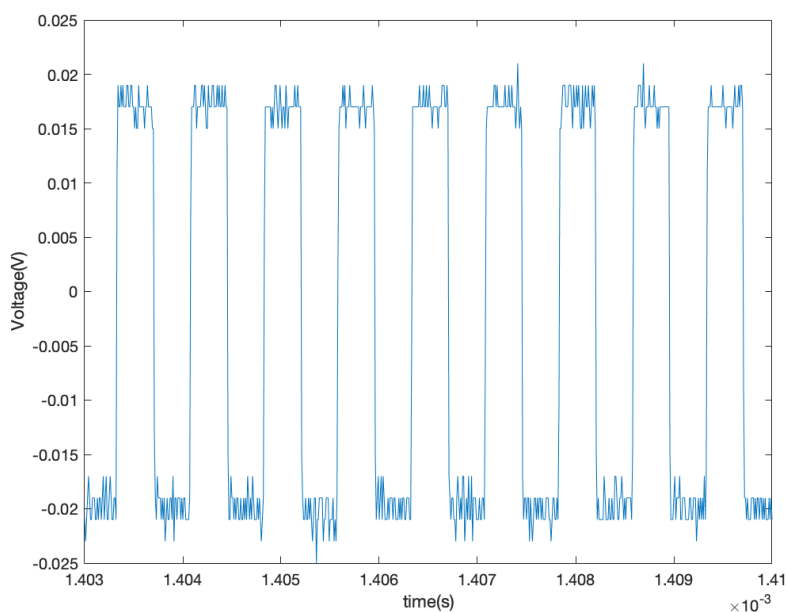


Figure 5: Square Sweep

c Sinal de Estudo

Diversos testes com diversas variações dos sinais de excitação listados acima, variando amplitude e frequência, foram realizados. O sinal escolhido para se construir os modelos foi o chirp representado na Figura 4. É válido destacar, que o chirp foi escolhido pois um outro trabalho já havia sido realizado com os mesmo dado, permitindo assim, um estudo comparativo.

Com a aquisição de dados, um total de 10 milhões de pontos foram gerados para 100ms do sinal, a frequência de amostragem foi de 100Mhz. Essa frequência elevada foi escolhida a fim de uma grande quantidade de dados pudesse ser gerada, e para que nenhum dado relevante fosse perdido. Na etapa de processamento dos dados, esses dados serão reamostrados, e apenas uma intervalo de tempo do sinal será usada para o processo de modelagem.

3 Processamento dos Dados

De posse dos dados medidos, a próxima etapa consiste em realizar um pré-processamento dos dados a fim de se eliminar possíveis ruídos, valores fora do intervalo desejados e verificar a consistência do mesmo. Na Figura 6 é possível observar os sinais de entrada e de saída. É importante ressaltar que o sinal de entrada está representado apenas em um curto intervalo de tempo, para facilitar a sua visualização, e que o sinal completo possui o mesmo intervalo de tempo que o sinal de saída.

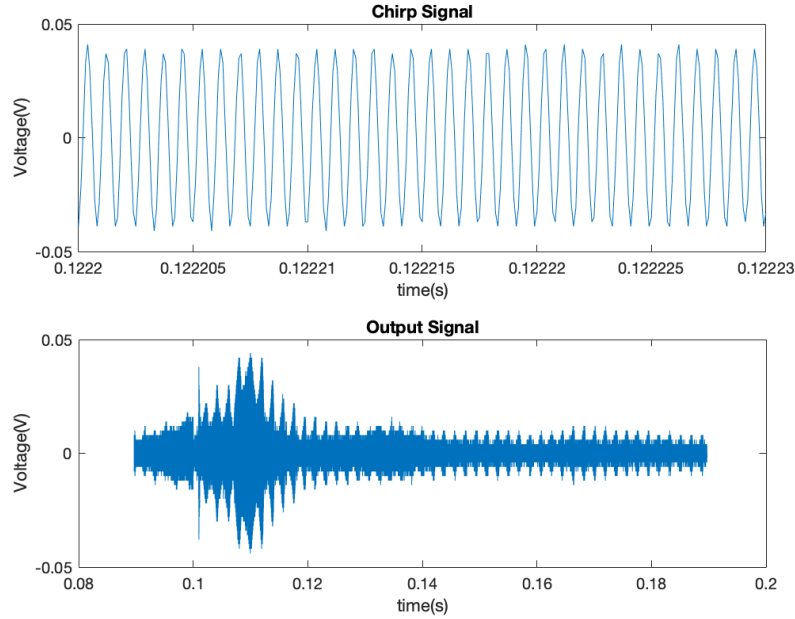


Figure 6: Sinal de entrada e de saída

Como dados obtidos foram *superamostrados*, 10 milhões de pontos foram gerados, é possível aplicar uma técnica de reamostragem para se obter a frequência ideal de amostragem.

a Método de Reamostragem

O método de reamostragem consiste em retirar diversas amostras do dado de origem, de forma a não alterar o formato do sinal. O período de reamostragem não deve ser grande demais a fim de que informações relevantes do dado original não sejam perdidas, mas também não deve ser muito pequeno de modo a inviabilizar a tarefa de modelagem por impor custo computacional excessivo. O teorema de Nyquist-Shannon versa sobre a frequência mínima para reconstrução do sinal. Este teorema especifica a taxa mínima de amostragem para que o sinal possa ser completamente reconstruído ou recuperado apenas por essas amostras [4]. Considerando a f_{max} como a frequência máxima do sinal de original, a frequência mínima de amostragem, f_s segue a seguinte fórmula:

$$f_s \geq 2f_{max} \quad (3)$$

A máxima frequência de interesse para o sinal de estudo é 1.35MHz, dessa maneira, a frequência mínima de amostragem é 2.7MHz, ou seja, um período de $3.7 \cdot 10^{-7}s$.

Apesar de se saber a frequência mínima para a amostragem, não se sabe qual é o período de amostragem ideal. Para isso, o método desenvolvido em [5] permite obter esse T_s . O objetivo desse método consiste em calcular um fator de dizimação $\Delta \in \mathbb{N}$, que representa a nova escala do sinal reamostrado. As funções de covariância são calculadas como:

$$\begin{aligned} r_{y^*} &= E[(y^*(k) - \overline{y^*(k)})(y^*(k - \tau) - \overline{y^*(k)})] \\ r_{y^{*2}} &= E[(y^{*2}(k) - \overline{y^{*2}(k)})(y^{*2}(k - \tau) - \overline{y^{*2}(k)})] \end{aligned} \quad (4)$$

onde os atrasos nos primeiros mínimos de cada função, representados por τ_{y^*} e $\tau_{y^{*2'}}$ são usadas para calcular τ_m^* com a equação $\tau_m^* = \min[\tau_{y^*}, \tau_{y^{*2'}}]$. Dessa maneira, Δ é calculado a fim de que τ_m fique dentro do intervalo 10 e 20, onde $\tau_m = \tau_m^* / \Delta$.

Com esse método, a nova frequência de reamostragem foi de 10^7 , reduzindo os pontos para um total de 1 milhão. Um script Matlab foi gerado para este método e disponibilizado no apêndice A.

b Sinal Reamostrado

Uma vez a frequência de reamostragem calculada, é possível comparar os gráficos dos dois sinais, tanto da entrada (chirp) quanto da saída, a fim de verificar a eficiência da reamostragem. As figuras 7 e 8 comparam a entrada e saída do modelo, antes e depois da reamostragem. Essas figuras representam apenas um intervalo do sinal, a fim de facilitar a sua visualização. É possível notar que não houve alteração relevante no sinal original, o que mostra a consistência do método utilizado.

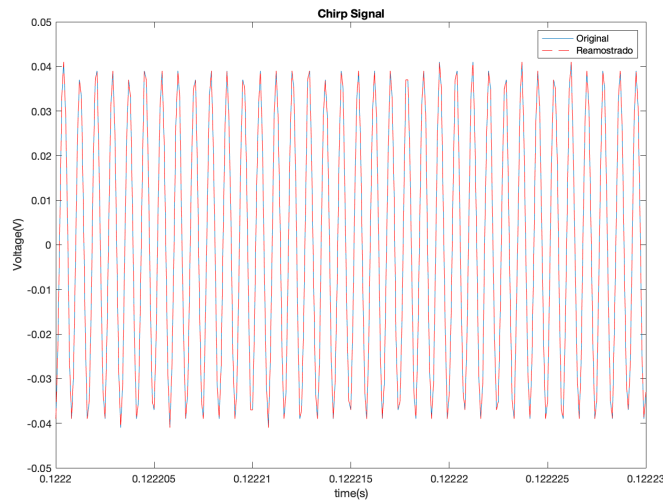


Figure 7: Sinal Chirp reamostrado (vermelho) vs original (azul)

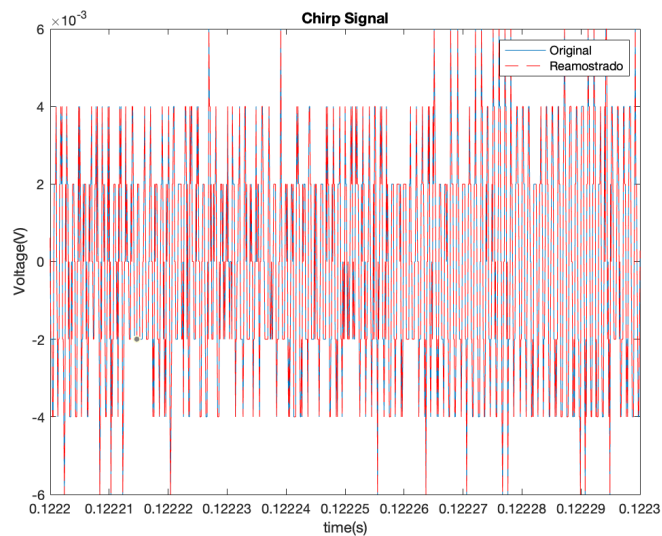


Figure 8: Sinal Chirp reamostrado (vermelho) vs original (azul)

Uma análise no domínio da frequência também se mostrou necessária para verificar se não houve alteração

na banda de frequência dos sinais.

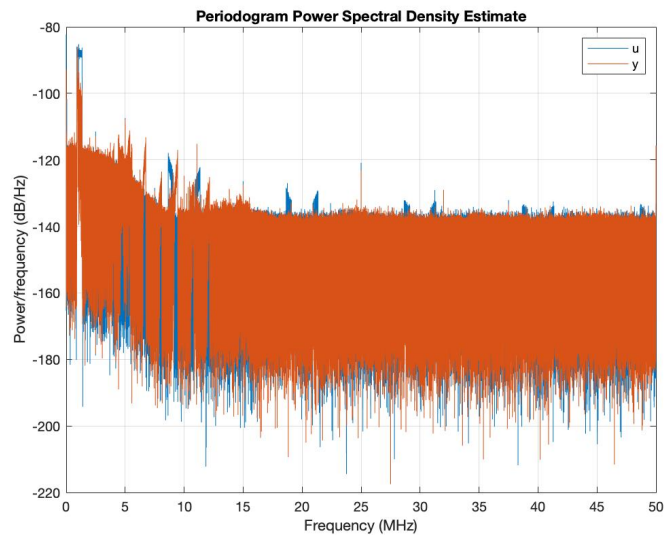


Figure 9: Plot da densidade spectral da entrada e saída dos sinais originais

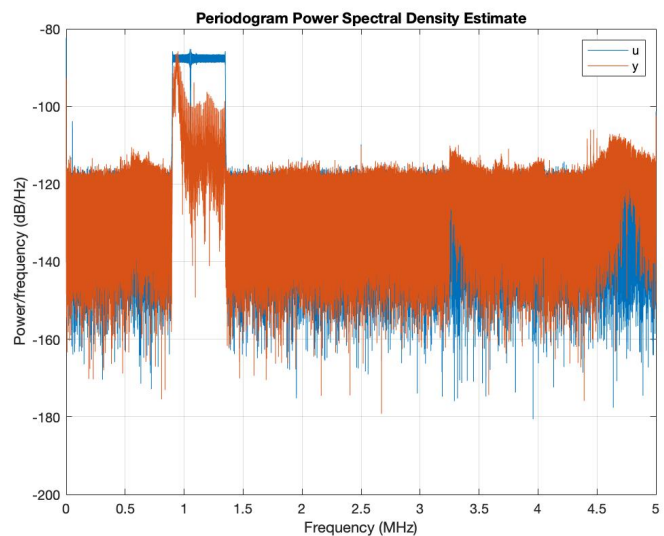


Figure 10: Plot da densidade spectral da entrada e saída dos sinais reamostrados

Como é possível observar nas figuras 10 e 9, existe um pico de resposta entre as frequências 900kHz e 1350kHz, que são as frequências de interesse. Algumas frequências aparecem na saída devido a existência de não-linearidade e também aos ruídos. Para retirar-las um filtro passa-banda de ordem 5 foi aplicado, obtendo o resultado representado na Figura 11.

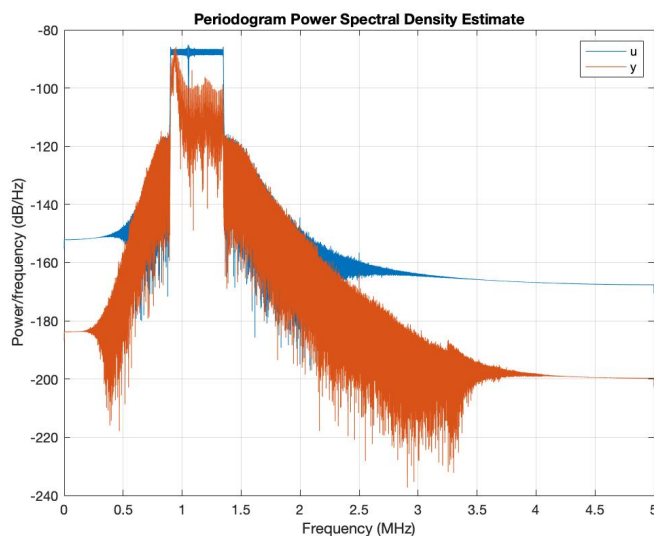


Figure 11: Plot da densidade spectral da entrada e saída dos sinais reamostrados e filtrados

Um script Matlab foi gerado para o processo de reamostragem e disponibilizado no apêndice B.

c Sinal para Modelagem

Uma vez o processo de reamostragem finalizado, um intervalo arbitrário do sinal de trabalho foi escolhido para realizar o processo de modelagem. Um total de 2000 amostras foram escolhidas. Esta escolha foi feita para permitir a criação e comparação de diferentes modelos em tempo hábil, uma vez que o tempo computacional para criação do modelo é proporcional à quantidade de dados utilizados. Dessa maneira, é possível observar na Figura 12 os sinais utilizados para a modelagem.

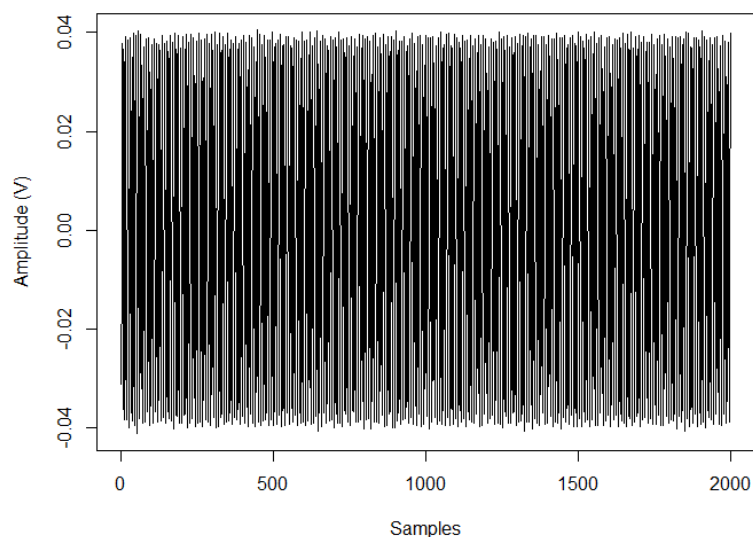


Figure 12: Sinal de entrada utilizado para a modelagem

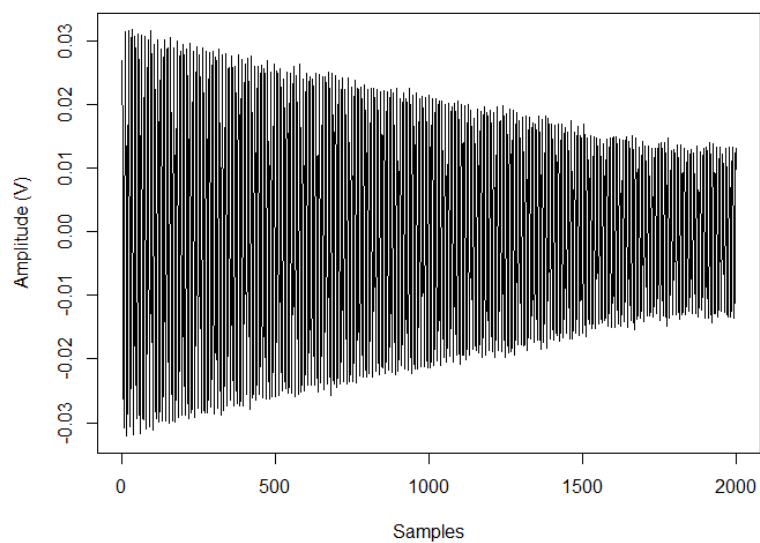


Figure 13: Sinal de saída utilizado para a modelagem

4 Identificação de Sistemas

Virtualmente todos os sistemas que existem podem ser representados através de um modelo matemático. Esses exemplos podem variar desde a vibração de um carro até um sistema biomédico. Mesmo os modelos sendo bastante diferentes, eles podem ser considerados como um sistema com entradas e saídas que possuem alguma relação que pode ser descrita por um modelo matemático. Dessa maneira, se estes últimos podem ser modelados, é possível criar um programa capaz de simular o comportamento real. Consequentemente, isso permite que diversas simulações sejam realizadas com muito mais facilidade e velocidade, permitindo estudar a resposta do sistema para diferentes entradas [6].

Identificação de sistemas consiste em calcular esse modelo matemático com base na relação entrada/saída do sistema. Mesmo em condições ideais, criar um modelo para um sistema dinâmico baseado em dados de entrada/saída não é trivial. Muitas vezes, não se sabe se o sistema em questão é linear ou não, ou ainda quantos parâmetros são necessários, seus tipos, qual tipo de não-linearidade está presente e ainda outras variáveis. Apesar da sua complexidade, esse método se mostrou prático e eficaz em modelagem [6].

Para o processo de identificação, diversas etapas devem ser seguidas para que o modelo seja construído com eficácia. O primeiro passo consiste na aquisição de dados. Os dados devem ser obtidos com muito cuidado, pois dados errôneos ou com muitos ruídos, irão definitivamente gerar uma simulação insatisfatória. Em seguida, é necessário definir qual modelo será utilizado. Posteriormente, os parâmetros, deste modelo são calculados e por fim o modelo é validado de acordo com a métrica escolhida. Se os resultados não estiverem dentro do desejado, é necessário voltar cada uma das etapas até que o problema seja encontrado. A Figura 14 representa o processo de identificação de sistemas:

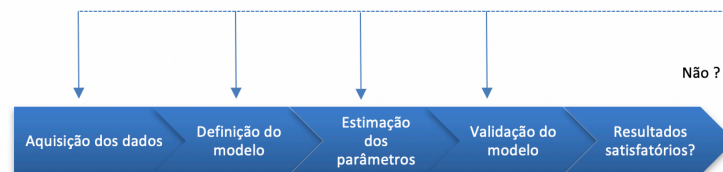


Figure 14: Etapas do processo de Identificação de Sistemas

Existem três formas de realizar esse processo de modelagem. A primeira é chamada de *Grey Box*. Essa abordagem leva em consideração um conhecimento físico do sistema, ou seja, é possível construir a estrutura do modelo com base nos dados experimentais e na física do próprio sistema. A segunda abordagem, é conhecida de *Black Box*. Nesse método, a modelagem é realizada com base apenas na relação entrada/saída, não existe um conhecimento prévio do seu funcionamento interno, reduzindo significativamente o trabalho de modelagem [7]. Por último existe a abordagem *White Box*, na qual o modelo é derivado diretamente de alguns princípios levando em consideração a conexão entre os componentes do sistema, porém diferentemente da primeira abordagem, não existe a dependência de dados experimentais [8].

Escolher entre uma dessas abordagens não é um trabalho simples. O uso do primeiro método possui a vantagem de que os relacionamentos físicos conhecidos são incorporados, e assim nenhum parâmetro deve ser desperdiçado estimando o que já é conhecido. Por um outro lado, esse processo de modelagem demanda tempo e uma alta demanda computacional na parte de minimização dos erros preditos.

Devido a complexidade física do experimento, uma abordagem caixa preta ou *Black Box* foi preferida. Este processo geralmente é um processo de tentativa e erro, em que diversos parâmetros de várias estruturas são estimados e comparados. Os primeiros modelos a serem construídos foram os modelos lineares mais simples e depois os mais complexos.

Para esse tipo de abordagem é possível então usar modelos padrão, que por experiência são conhecidos por serem capazes de lidar com uma ampla gama de dinâmicas de sistema diferentes.

a Identificação de Sistema Lineares

Sistemas lineares são definidos como sistemas que satisfazem o princípio da superposição. Este princípio afirma que a resposta causada por dois ou mais estímulos juntos, pode também ser causada pela junção

das respostas de cada um dos estímulos separados. Para um sistema ser considerado linear, ele deve respeitar as equações abaixo.

$$\begin{aligned} F(x + y) &= F(x) + F(y) \\ F(\alpha x) &= \alpha F(x) \end{aligned} \quad (5)$$

Para esse tipo de abordagem, existem dois modelos lineares que foram estudados: AutoRegressive with eXogenous inputs (ARX) e AutoRegressive Moving Average with eXogenous inputs (ARMAX). As estruturas de cada um desses modelos de caixa preta seguem o formato da equação dos sistemas lineares invariantes no tempo [9]:

$$A(z^{-1})y(k) = \frac{B(z^{-1})}{F(z^{-1})}u(k) + \frac{C(z^{-1})}{D(z^{-1})}e(k) \quad (6)$$

$u(k)$ representa a entrada do sistema, $y(k)$ a saída e $e(k)$ o erro de modelagem. A , B , C , D e F são polinômios. A entrada $u(k)$ é para o controle do sistema, e tanto $y(k)$ quanto $u(k)$ são rastreadas e gravadas. As saídas e entradas são completamente ou parcialmente mensuráveis, enquanto a perturbação do sistema não é classificada diretamente. O erro é medido através da diferença do valor predito com o valor real da saída [9].

1 ARX

O modelo ARX é o mais simples que incorpora o sinal de estímulo. A sua solução sempre satisfaz o mínimo global da função de perda, e a sua solução é sempre única. Porém a sua estrutura não controla os ruídos, e assim os distúrbios fazem parte da dinâmica do sistema. No entanto essa influência pode ser reduzida com uma boa relação sinal/ruído.

Definindo $C(z^{-1}) = D(z^{-1}) = F(z^{-1}) = 1$ na equação 6, se chega à:

$$y(k) = \frac{B(z^{-1})}{A(z^{-1})}u(k) + \frac{e(k)}{A(z^{-1})} \quad (7)$$

O modelo pode ser escrito pela equação diferencial linear:

$$y(k) + a_1y(k-1) + \dots + a_{n_a}y(k-n_a) = b_0u(k-d) + \dots + b_{n_b}u(k-n_b) + e(t) \quad (8)$$

Na equação 8 é possível observar os termos n_a e n_b que apresentam as ordens do modelo. Os termos $\theta = [a_1a_2 \dots a_{n_a}b_0 \dots b_{n_b}]$ são os parâmetros ajustáveis do modelo [10].

O modelo ARX utilizado para fazer a predição, utilizou o método dos mínimos quadrados para fazer a estimação de seu parâmetros.

Método dos mínimos quadrados

Este método é usado para calcular a melhor linha de ajuste aos dados. Dado os dados de entrada e saída $(u_1, y_1), \dots, (u_N, y_N)$ e o modelo representado pela Eq.(8), o vetor de regressão é definido como $\phi(k) = [-y(k-1), -y(k-2), \dots, -y(k-n_a), u(k-1), u(k-2), \dots, u(k-n_b)]^T$. O vetor de parâmetros com $n_a + n_b$ dimensões é definido como $\theta = [a_1a_2 \dots a_{n_a}b_0 \dots b_{n_b}]$. É possível reescrever a Eq.(8) utilizando esses novos parâmetros:

$$y(k) = \phi^T(k)\theta + \xi(k) \quad (9)$$

Considerando $p = 1 + \max(n_a, n_b)$ e N como sendo a quantidade de medições realizadas, é possível chegar a equação:

$$\begin{bmatrix} y(p) \\ y(p+1) \\ \vdots \\ y(N) \end{bmatrix} = \begin{bmatrix} \phi^T(p) \\ \phi^T(p+1) \\ \vdots \\ \phi^T(N) \end{bmatrix} \theta + \begin{bmatrix} \xi(p) \\ \xi(p+1) \\ \vdots \\ \xi(N) \end{bmatrix} \quad (10)$$

Renomeando as variáveis: $\mathbf{y} = \Phi\theta + \xi$. O vetor de estimação dos parâmetros $\hat{\theta}$ é calculado com objetivo de se minimizar o erro gerado pela diferença entre o valor predito

$$\hat{y}(k) = \phi^T(k)\hat{\theta} \quad (11)$$

e o valor real de $y(k)$. Assim, o erro pode ser definido como: $e(k) = y(k) - \hat{y}(k)$ ou na forma matricial $\mathbf{e} = \mathbf{y} - \hat{\mathbf{y}}$. Definindo o vetor $\hat{\theta}$ como o vetor que minimiza o resíduo:

$$\hat{\theta} = \arg_{\theta} \min J(\theta) \quad (12)$$

onde $J = \mathbf{e}^T \mathbf{e}$. J representa a soma dos quadrados dos resíduos, e assim minimizando este parâmetro, melhor será o resultado das predições do modelo. O erro se minimiza através da equação:

$$\nabla J = 0 \quad (13)$$

Obtendo assim:

$$\hat{\theta} = [\Phi^T \Phi]^{-1} \Phi^T \mathbf{y} \quad (14)$$

2 ARMAX

Diferentemente do modelo acima, o modelo ARMAX leva em consideração o parâmetro do erro. Esse modelo se torna mais útil ao se dominar distúrbios que entraram no início do processo. A sua estrutura é a mesma do modelo ARX porém com um termo de erro da média móvel.

Fazendo $F(z^{-1}) = D(z^{-1}) = 1$ na equação 6, e escrevendo na sua forma diferencial, se obtém a seguinte equação.

$$y(k) + a_1 y(k-1) + \dots + a_{n_a} y(k-n_a) = b_0 u(k) + \dots + b_{n_b} u(k-n_b) + e(k) + c_1 e(k-1) + \dots + c_{n_c} e(k-n_c) \quad (15)$$

n_a , n_b e n_c representam as ordens do modelo referente a saída, entrada e o erro, respectivamente. Os parâmetros ajustáveis do modelo $\theta = [a_1 a_2 \dots a_{n_a} b_0 \dots b_{n_b} c_0 \dots c_{n_c}]$

Para estimação de parâmetros o modelo ARMAX do projeto, utilizou o *prediction error method* (PEM).

PEM

Esse método usa uma otimização numérica para minimizar a função de custo, uma norma ponderada do erro de previsão.

Esse erro é dado pela equação:

$$\epsilon(k, \theta) = y(k) - \hat{y}(k|\theta) \quad (16)$$

Onde θ é o vetor dos parâmetros desconhecidos. A função de custo a ser minimizada pelo método é:

$$V_N(\theta, Z^N) = \frac{1}{N} \sum_{k=1}^N (y(k) - \hat{y}(k|\theta))^2 \quad (17)$$

onde Z^n é um conjunto de dados de entrada e saída [10].

b Identificação de Sistemas Não lineares

Sistemas não lineares são definidos como os sistemas que não satisfazem ao princípio da superposição descrito na Eq.(5). Apesar dessa definição ser abrangente, ele acaba se tornando necessária devido a enorme quantidade de sistemas não lineares, o que tornaria impossível escrever uma descrição que cobrisse todas as classes [6].

Dentre os diversos modelos, nesse projeto o modelo não-linear utilizado para a realização dos testes e obtenção do resultado foi o modelo NARX com base em rede neurais artificiais.

1 NARX

O modelo NARX é uma generalização não linear do modelo ARX. Esse modelo é aplicado a uma variedade de sistemas dinâmicos não lineares, incluindo também séries temporais. Ao invés de ser descrito por uma equação que representa uma soma ponderada dos regressores, como apresentado na Eq.(8), esse modelo apresenta uma função de mapeamento não linear F , que o torna mais flexível. Essa equação está definida abaixo.

$$y(t) = F[y(t-1), y(t-2), \dots, y(t-n_a), u(t-1), u(t-2), \dots, u(t-n_b)] + \xi(t); \quad (18)$$

onde n_a e n_b apresentam as ordens da saída e da entrada, respectivamente. Existem diversas funções não lineares para o NARX, as entradas para F são regressores do modelo.

O uso de redes neurais artificiais para a resolução de problemas de sistemas de identificação equivale à: escolher a ordem das entradas $u(t)$ e saídas $y(t)$ da Eq. (18) defasadas do modelo para que o sistema físico seja representado de forma satisfatória; e também equivale à estimar os parâmetros de $F[\cdot]$ para essas entradas escolhidas, de forma que o modelo seja válido de acordo com alguma medida predefinida.

Rede Neurais Artificiais

Redes neurais artificiais é um maquina desenhada para imitar a forma que o cérebro humano realiza uma determinada tarefa. O processo de aprendizagem se dá através dos neurônios da rede. Segundo [11], esse mecanismo é caracterizado por ser altamente paralelizado e capaz de armazenar conhecimento para um futuro uso. Em comparação com o cérebro humano, uma rede neural artificial também aprende a partir de seu ambiente e o seu conhecimento vêm dos pesos sinápticos entre os seus neurônios.

A estrutura interna de uma rede neural MLP (Multi-Layer Perceptron) está representa na Figura 15. Uma rede MLP, é uma das mais conhecidas estruturas de redes, ela é um aproximador universal capaz de aproximar uma função contínua arbitrária [11].

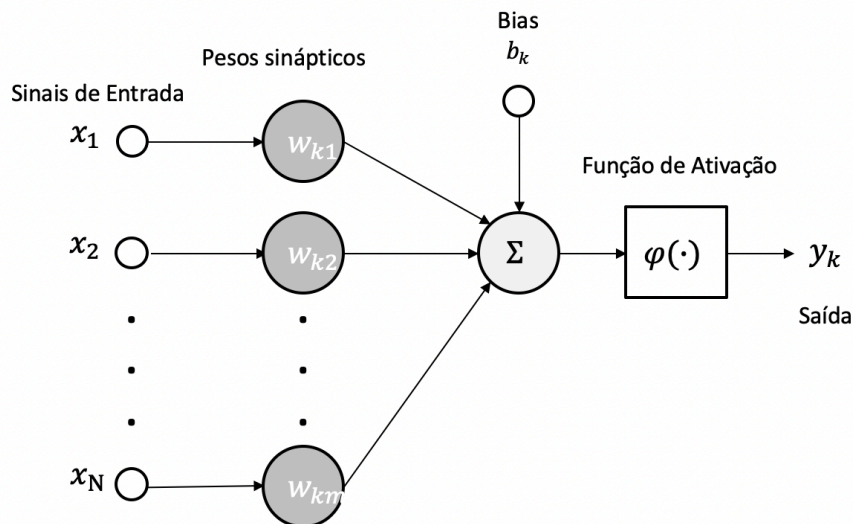


Figure 15: Estrutura interna de uma rede neural artificial

A estrutura de uma rede basicamente é dividida em neurônios, camadas e função de ativação. Em um cérebro humano, os neurônios se comunicam uns com os outros através de complexas conexões. Em um rede artificial essas conexões são feitas através de algoritmos. Cada conexão possui um peso, que pode ser tanto negativo ou positivo. Como observado na Fig.15, o neurônio é responsável por fazer a soma de cada uma das respectivas entradas (x_1, x_2, \dots, x_m), multiplicadas pelos seus respectivos pesos ($w_{k1}, w_{k2}, \dots, w_{km}$). A saída do neurônio passa então por uma função de ativação ($\phi(\cdot)$). O objetivo dessa função é de introduzir uma não linearidade na saída do neurônio, o que permite que a rede realize tarefas mais complexas, caso contrário a mesma corresponderia a um modelo de regressão linear. Existem três tipos de funções que são as mais conhecidas: arco-tangente,

$$f(x) = \tan^{-1}(x) \quad (19)$$

sigmoide

$$f(x) = \frac{1}{1 + e^{-x}} \quad (20)$$

e ReLU

$$f(x) = \begin{cases} 0, & x \leq 0 \\ x, & x > 0 \end{cases} \quad (21)$$

As diferenças entre elas correspondem ao domínio da imagem de cada uma e também ao grau de diferenciabilidade de cada uma.

Os neurônios são agrupados em camadas. Basicamente, esses modelos possuem uma camada de entrada, uma de saída e uma ou mais camadas escondidas. A camada de entrada corresponde aos neurônios que recebem as entradas e a camada de saída aos últimos neurônios responsáveis por gerar o resultado final. As camadas escondidas se encontram entre essas duas últimas camadas, elas são responsáveis por aumentar o grau da capacidade de resolução da rede, porém a demanda computacional também é maior [12]. Na Fig.16 se observa cada uma das camadas de uma rede neural artificial.

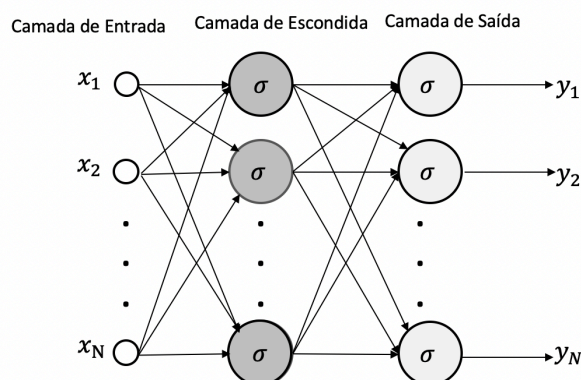


Figure 16: Camadas de uma rede neural artificial.

Diferentes tipos de redes neurais usam princípios diferentes para determinar suas próprias regras. Cada tipo de rede possui uma especialidade e uma força única. Nesse projeto quatro arquiteturas diferentes de redes neurais foram analisadas com intuito de se obter o melhor resultado: FeedForward, Cascade FeedForward, Sigmoid e Wavenet.

FeedForward

Conhecida por ser uma das mais simples redes, essa rede é caracterizada pelo fato dos dados seguirem uma única direção e sentido. Os dados seguem o fluxo da camada de entrada até a camada de saída, como algumas outras redes. Como representado na Figura 15 a soma dos produtos da entradas é calculada e fornecida para a próxima camada.

Cascade FeedForward

Esse tipo de arquitetura apresenta similaridades com a arquitetura FeedForward, porém possui uma diferença. Em uma rede FeedForward a camada de entrada e de saída estão interligadas de uma forma indireta, pelo fato das entradas passarem pelos neurônios da camada escondida. Já no caso da configuração Cascade FeedForward a entrada e a saída estão conectadas de uma forma direta além da forma indireta presente no outro tipo de rede. A vantagem desse método é que ele acrescenta um relacionamento não linear entre a entrada e a saída, além de não eliminar o relacionamento linear entre os dois [13].

Na Figura 17 se observa que a camada de entrada se encontra interliga com a camada de saída, além da configuração normal de uma rede neural FeedForward.

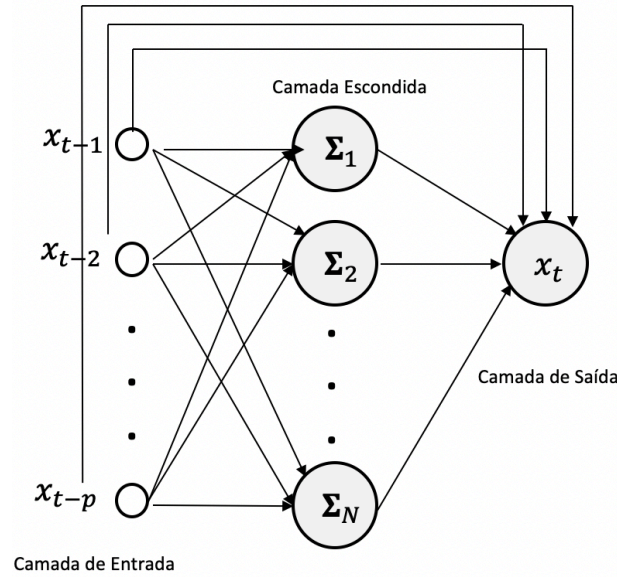


Figure 17: Estrutura de uma rede neural Cascade FeedForward.

Redes Sigmoid e Wavenet

Uma das aplicações dessas estruturas de rede é para a estimativa de funções. Dada uma série de amostras passadas dos valores da função, é possível treinar a rede para que ela aprenda a composição da função e, portanto, calcular um valor esperado para uma determinada entrada. Tanto redes wavelet quanto sigmoid seguem o mesmo princípio de arquitetura porém a classe de função de cada uma são diferentes. Uma função sigmoide foi discutida anteriormente, e sua equação se encontra 20. Já wavelets são uma classe de função que podem ser definidas como funções de onda com de tamanho reduzido, denominadas de $\psi(\cdot)$. Essas funções, diferentemente das senoides por exemplo, crescem e depois decaem em um período finito de tempo. Essa função é definida nos eixos dos reais, possui sua integral igual zero em seu domínio, a integral do seu quadrado nesse mesmo domínio equivale à 1 e ainda satisfaz a condição de admissibilidade.

Essas redes combinam a teoria das funções de wavelet e sigmoid com a de redes neurais. Essas redes consistem em uma rede neural feedforward com uma camada escondida, cujas funções de ativações correspondem a cada uma dessas funções.

$$\psi_{\lambda,t}(u) = \psi\left(\frac{u-t}{\lambda}\right) \quad (22)$$

onde λ e t são parâmetros de dilatação e contração da onda, respectivamente.

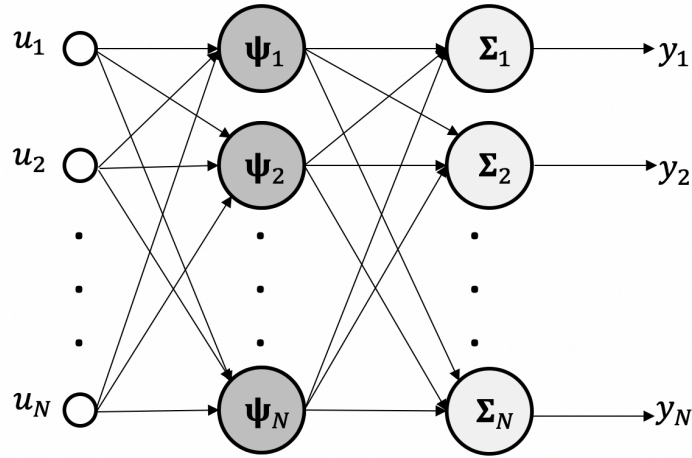


Figure 18: Estrutura de uma rede neural wavelet.

Como representado na Figura 18, se observa a camada de entrada recebendo uma ou mais entradas, uma camada de saída em um ou mais combinadores e uma camada oculta. Os neurônios da camada oculta possuem suas funções de ativação extraídas de uma base wavelet [14]. A mesma estrutura é observada para uma rede sigmoid.

c Validação do Modelo

A validação do modelo escolhido é uma parte fundamental da identificação de sistemas. O objetivo consiste em testar o modelo escolhido, a fim de verificar que o mesmo seja capaz de representar o conjunto de dados de forma adequada. Esse processo representa a última etapa do processo de identificação, e assim, será nessa parte que o resultado irá mostrar se o estudo foi corretamente realizado ou se ainda precisa ser melhorado, como representado na Figura 14.

Uma técnica usada para analisar a performance do modelo consiste em realizar previsões com base em valores passados, ou ainda com base em previsões passadas. Essa duas abordagens são definidas respectivamente como o teste de previsão OSA (*One Step Ahead*) e o FR (*Free Run Simulation*).

1 OSA e FR

O resíduo da *One Step Ahead Residual* é definido pela equação abaixo, onde \hat{y} é o valor predito.

$$\xi(t) = y(t) - \hat{y}(t) \quad (23)$$

Esse teste quantitativo é capaz de mostrar a aderência do modelo, pois os dados mais recentes são utilizados para o cálculo de \hat{y} . Para um modelo ARX de segunda ordem, apresentado em 24

$$y(k) = ay(k-1) + by(k-2) + cu(k-1) + e(k) \quad (24)$$

As previsão da predição OSA, começando de 3, seriam:

$$\begin{cases} \hat{y}(3) = ay(2) + by(1) + cu(2) \\ \hat{y}(4) = ay(3) + by(2) + cu(3) \\ \vdots \\ \hat{y}(k) = ay(k-1) + by(k-2) + cu(k-1) \\ \vdots \end{cases} \quad (25)$$

Como representado no sistema acima, a predição OSA é calculada com valores precedentes do modelo, e o seus erros são redefinidos a cada iteração [6]. Por essa razão os seus erros normalmente são menores

em comparação com os erros de previsões de janelas maiores. Devido a esse tipo de abordagem, é mais difícil de validar o modelo utilizando apenas o *one step ahead*. Essa janela de horizonte de previsão é comumente usada como uma primeira abordagem para se analisar o resultado, pois se o seu resultado não seja satisfatório, o resultado utilizando uma janela maior ou com FR será, em grande parte do tempo, pior. Para assegurar o resultados dos modelos, o FR, também foi estudada para o projeto.

A previsão da métrica residual FR é obtida através de previsões passadas. Os erros dessa métrica são acumulados ao longo de cada iteração e assim, permite analisar a validade do modelo com uma maior facilidade. Considerando o modelo apresentado na equação 18, as primeiras previsões são inicializadas com valores reais dos dados:

$$\begin{cases} \hat{y}(1) = y(1) \\ \hat{y}(2) = y(2) \\ \vdots \\ \hat{y}(k) = y(k) \\ \vdots \end{cases} \quad (26)$$

As previsões da previsão são realizadas sobre previsões passadas de acordo com o sistema:

$$\begin{cases} \hat{y}_s(n_y + 1) = F(\hat{y}_s(n_y), \hat{y}_s(n_y - 1), \dots, \hat{y}_s(1), u(n_u), u(n_u - 1), \dots, u(1)) \\ \vdots \\ \hat{y}_s(t) = F(\hat{y}_s(t - 1), \hat{y}_s(t - 2), \dots, \hat{y}_s(t - n_y), u(t - 1), u(t - 2), \dots, u(t - n_u)) \\ \vdots \end{cases} \quad (27)$$

Por fim os resíduos são calculados através da equação: $\xi(t) = y_s(t) - \hat{y}(t)$.

Junto com essas duas métricas de resíduos apresentadas, é necessário o uso de métricas capazes de quantificar a eficiência do modelo. As métricas escolhidas para desempenhar essa função foram calculadas com base no resíduos.

2 Métrica Quantitativa de Erro

A ideia básica por trás das métricas de erro é quantificar a capacidade do modelo identificado em reproduzir a dinâmica presente nos dados adquiridos. As métricas quantitativas se tornam importantes pois elas permitem investigar possíveis problemas de complexidade na simulação e também na generalização do modelo. Elas facilitam a análise dos resultados, permitindo um estudo do valor numérico obtido para o erro residual.

Uma das métricas que se permite quantificar o erro é o coeficiente de correlação múltipla (R^2).

O erro métrico quadrático é definido através da seguinte equação abaixo, onde $\xi(t)$ representa o erro:

Nesse projeto a métrica de estudo foi o R^2 , pelo fato do seu valor não depender da amplitude do modelo, tendo 1 como seu valor máximo. O R^2 é definido de acordo com a equação 28, onde $\bar{y}(t)$ representa o valor médio da sequência:

$$R^2 = 1 - \frac{\sum_{t=1}^N [\xi(t)]^2}{\sum_{t=1}^N [y(t) - \bar{y}(t)]^2} \quad (28)$$

É importante ressaltar que, para o R^2 , quanto mais perto de zero pior é a representação do sistema pelo modelo, e quanto mais perto de 1 melhor essa representação.

5 Resultados

Uma vez que os dados são pré-processados conforme detalhado na seção 3, os modelos escolhidos e as métricas definidas, a parte final consiste em testar cada um dos modelos e verificar o melhor resultado obtido. Como discutido anteriormente a métrica quantitativa utilizada foi o R^2 . Tanto as predições OSA quanto FR foram escolhidas. Cada um dos três modelos ARX, ARMAX e NARX tiveram suas ordens e arquiteturas modificadas a fim de se maximar a métrica residual para cada um. É importante ressaltar que os melhores modelos foram escolhidos com base no R^2 da FR.

Todo o processo de pré-processamento dos dados, criação dos modelos e geração dos resultados foram realizados utilizando o software MATLAB [15].

a ARX

As ordens do modelo ARX foram variadas em um intervalo específico até que o melhor resultado fosse encontrado. Os valores das ordens do modelo foram definidos como todas as combinações para $n_a, n_b \in [1, 10]$. Para a estimação de seus parâmetros o método dos mínimos quadrados foi utilizado. A ordem do melhor modelo obtido é: $n_a = 1$ e $n_b = 5$, com um R^2 FR de 89.90%, como apresentado na tabela 1.

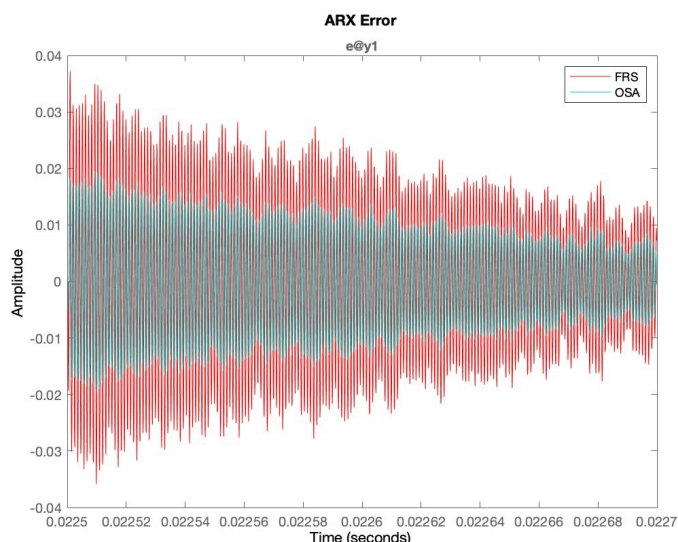


Figure 19: Erro do melhor modelo ARX com OSA (azul) e FRS (vermelho).

b ARMAX

Assim como para o modelo ARX, o modelo ARMAX teve seus parâmetros variados. n_a , n_b e n_c variam no intervalo de 1 a 10. Um script MATLAB foi gerado para realização das iterações e disponibilizado no apêndice C. As iterações dos demais modelos foram realizadas utilizando este mesmo código com apenas algumas alterações.

Para a estimação de seus parâmetros, o método utilizado foi o PEM. O melhor modelo obtido foi com as seguintes ordens: $n_a = 10$, $n_b = 2$ e $n_c = 6$; com um R^2 de FR de 90.60%.

Na Figura 20 é possível observar os erros residuais tanto de OSA e FRS para o melhor modelo ARX. O erro da simulação livre é maior que a do um passo à frente. É possível notar que os erros acompanham o formato do sinal de saída representado na Figura 13.

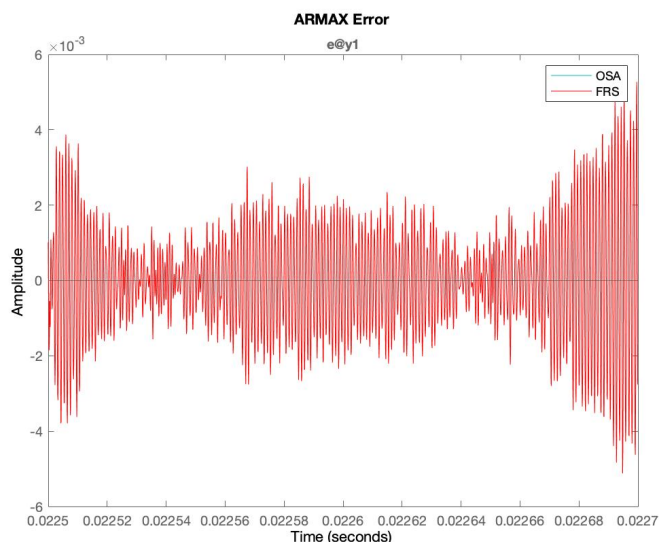


Figure 20: Erro do melhor modelo ARMAX com OSA (azul) e FRS (vermelho).

c NARX

Já para o modelo NARX além de se fazer mudanças nas alterações da ordens do modelo, se mostrou necessário mudar alguns parâmetros da rede neural. É importante lembrar que os resultados desse modelo foram gerados utilizando uma rede neural feedforward, cascade feedforward, sigmoid e wavenet. As ordens do modelo foram variadas em um intervalo maior que os dos modelos anteriores. n_a e n_b foram variados no intervalo $[1 : 20]$. As camadas escondidas da rede foram alteradas entre 1 e 2, e a quantidade de neurônios em cada uma delas variou entre 4, 5 e 10. É válido notar que o uso da rede feedforward e da cascade feedforward exigia uma alta demanda computacional e assim um elevado tempo para convergir. Dentre todas as variações o modelo que apresentou um desempenho melhor foi a rede sigmoid. O melhor R^2 FRS obtido foi de 89.20% com as ordens $n_a = 16$ e $n_b = 15$. A fim de se variar os parâmetros do modelo, um script MATLAB foi gerado, e disponibilizado no apêndice D para um estudo mais detalhado.

Na Figura 21 se observa o erro dos dois horizontes de predição escolhido.

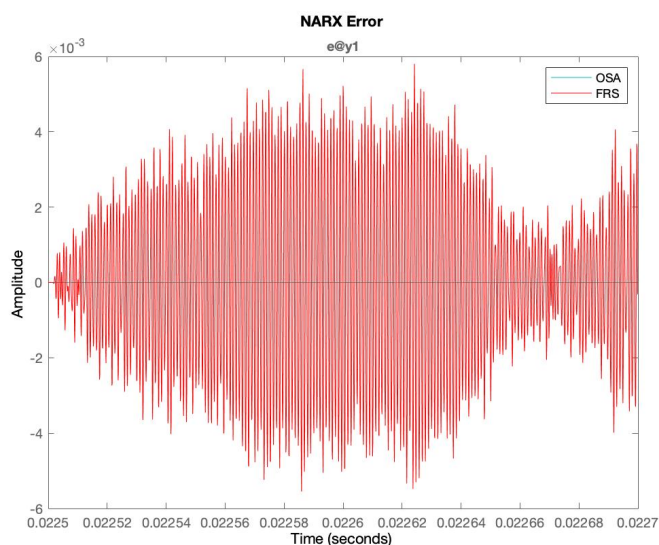


Figure 21: Erro do modelo melhor NARX com OSA (azul) e FRS (vermelho).

Para este modelo o algoritmo de aprendizado utilizado foi aquele dentre os seguintes que apresentasse a

primeira descida que levasse a uma redução no custo de estimativa: *Subspace Gauss-Newton least squares search*, *Levenberg-Marquardt least squares search*, *Steepest descent least squares search* e *Adaptive subspace Gauss-Newton search*.

d Resumo dos resultados

Os melhores resultados de cada um dos modelos estão listados na tabela abaixo, em ordem crescente do R^2 em FRS. Na tabela se observa um resultado melhor para a métrica em um passo a frente do que para a simulação livre, como já era esperado. Apesar do modelo ARX ter obtido um R^2 melhor em FRS, apresentou melhor resultado para OSA. Nota-se também que as predições em OSA são, para todos os modelos, muito próximas da unidade. Isto demonstra a importância da utilização da métrica baseada em FR, de modo a esclarecer a real qualidade das previsões.

Table 1: Todos os modelos ordenados por ordem crescente de R^2 em FRS.

Modelo	n_b	n_a	n_c	n_k	R2 OSA	R2 FRS
NARX	19	17	NA	NA	1	0.8830
NARX	17	16	NA	NA	1	0.8830
NARX	18	18	NA	NA	1	0.8880
NARX	15	16	NA	NA	1	0.8920
ARX	3	4	NA	1	0.992	0.8950
ARX	1	2	NA	1	0.992	0.8960
ARX	4	2	NA	1	0.992	0.8970
ARX	5	1	NA	1	0.999	0.8990
ARMAX	1	5	2	1	1	0.9000
ARMAX	1	5	5	1	1	0.9010
ARMAX	2	10	7	1	1	0.9040
ARMAX	1	8	3	1	1	0.9040
ARMAX	2	10	6	1	1	0.9060

O modelo que se obteve o melhor resultado foi o modelo linear ARMAX em termos de erro em FR. É possível observar na Fig.22 que a autocorrelação dos resíduos se encontra fora da faixa do intervalo de confiança, indicando que há dinâmica presente nos resíduos e que o modelo ainda pode ser melhorado. Por outro lado, analisando o gráfico das correlações cruzadas, é possível afirmar com intervalo de confiança de 99%, que os resíduos não são correlacionados com entradas passadas.

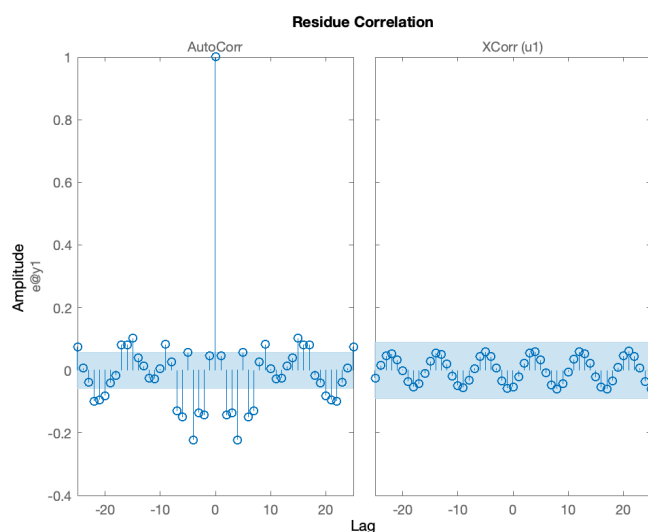


Figure 22: Autorrelação dos Resíduos a esquerda e a correlação cruzada entre os resíduos e as entradas a direita.

Apesar desse melhor modelo não levar em consideração o aspecto não linear do sistema ele se mostrou

eficaz na modelagem, como é possível observar nas figuras 23 e 24. Isto se deve por conta do modelo ARMAX melhorar a estimação dos parâmetros através dos termos de média móvel. O modelo NARX baseado em redes neurais não apresenta este mecanismo. Possivelmente, o resultado da rede neural poderia ter sido melhorado se o modelo fosse separado em dois componentes, um linear e outro não-linear. Nesta situação, o modelo não linear seria utilizado para modelar os resíduos do modelo linear, de modo que a saída final dos dois modelos seria composta pela sua soma.

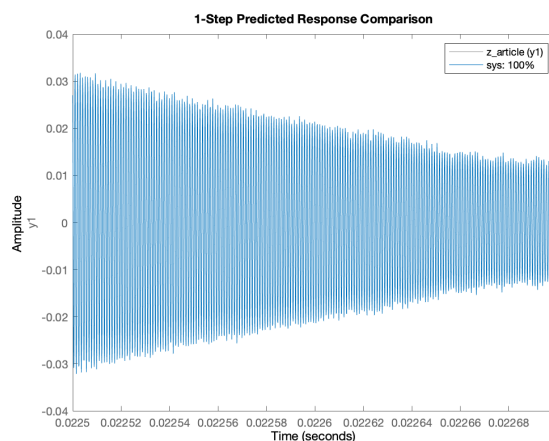


Figure 23: Saída do modelo ARMAX OSA e a saída real

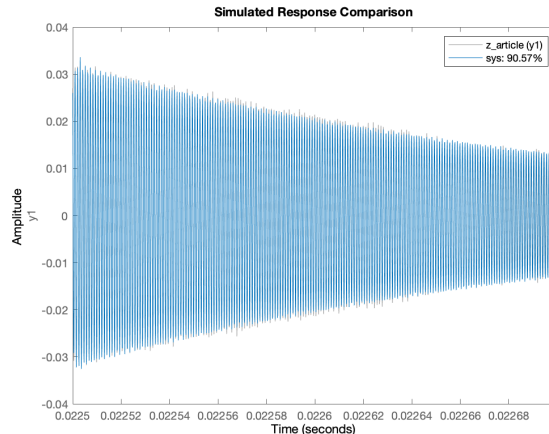


Figure 24: Saída do modelo ARMAX FR e a saída real

6 Conclusões & Trabalho Futuro

Esse trabalho mostrou o uso de técnicas de identificação de sistemas para a modelagem de um sistema de transmissão acústica através de múltiplas camadas com atuadores piezoelétricos com modelos de caixa preta. Dentre os 3 modelos apresentados o que se obteve melhor resultado foi o modelo ARMAX. Os resultados do modelo ARX e NARX não foram muito diferentes. O objetivo do projeto de comparar métodos de identificação, lineares e não lineares, para a modelagem do sistema de transmissão acústica com material piezoelétrico utilizando técnicas de identificação de sistemas foi concluído com êxito. Contudo, os modelos testados ainda podem ser melhorados com o uso de métodos avançados para identificação, tais como ensembles [16]. Desta forma, o modelo com redes neurais artificiais pode ser construído a fim de modelar toda a dinâmica do sistema que o modelo linear não foi capaz de fazê-lo. A capacidade do uso de redes neurais para séries temporais é um ferramente bastante eficaz e se acredita que melhores resultados ainda podem ser obtidos. Os parâmetros da rede podem ainda ser variados com uma maior quantidade de combinações entre cada um dos intervalos de seus parâmetros. Uma outra ideia, consiste em testar o modelo com outras formas de arquitetura de redes além das três que foram testadas.

Modelos futuros podem ser construídos utilizando outros tipos de sinal como por exemplo o sinal multisenoidal de fase aleatória. Ainda é necessário estudar a influência das condições operacionais sobre o desempenho do modelo, como a distância da placas piezoelétricas, temperatura e tipo de fluido.

7 Referências

- [1] D.-X. Yang, Z. Hu, H. Zhao, H.-F. Hu, Y.-Z. Sun, and B.-J. Hou, "Through-metal-wall power delivery and data transmission for enclosed sensors: A review," *Sensors*, vol. 15, no. 12, pp. 31 581–31 605, 2015.
- [2] D. Shoudy, G. Saulnier, H. Scarton, P. Das, S. Roa-Prada, J. Ashdown, and A. Gavens, "P3f-5 an ultrasonic through-wall communication system with power harvesting," in *2007 IEEE Ultrasonics Symposium Proceedings*. IEEE, 2007, pp. 1848–1853.
- [3] G.-Y. Gu, L.-M. Zhu, C.-Y. Su, H. Ding, and S. Fatikow, "Modeling and control of piezo-actuated nanopositioning stages: A survey," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 1, pp. 313–332, 2014.
- [4] E. Lai, "2 - converting analog to digital signals and vice versa," in *Practical Digital Signal Processing*, E. Lai, Ed. Oxford: Newnes, 2003, pp. 14 – 49. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/B9780750657983500023>
- [5] L. A. Aguirre, "A Bird's Eye View of Nonlinear System Identification," *arXiv e-prints*, p. arXiv:1907.06803, Jul 2019.
- [6] S. A. Billings, *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains*. John Wiley & Sons, 2013.
- [7] P. Theodor, "Modeling of dynamic systems: by lennart ljung and torkel glad; prentice hall information and system sciences series; prentice hall; englewood cliffs, nj, usa; 1994 isbn: 0-13-597097-0 - book review," *Control Engineering Practice*, vol. 3, p. 897–898, 06 1995.
- [8] G. De Nicolao, "System identification: Problems and perspectives," in *12th Workshop on Qualitative Reasoning*, vol. 3, 1997, pp. 1–2.
- [9] M. Saifizi, M. Ab Muin, S. Yaacob, and M. Mohamad, "Comparison of arx and armax models for thermoelectric refrigerator," in *IOP Conference Series: Materials Science and Engineering*, vol. 50, no. 1. IOP Publishing, 2013, p. 012032.
- [10] S. Rachad, B. Nsiri, and B. Bensassi, "System identification of inventory system using arx and armax models," *International Journal of Control and Automation*, vol. 8, pp. 2 83 – 2 94, 12 2015.
- [11] S. Haykin, *Neural Networks and learning machines*. Upper Saddle River: Prentice Hall, 2009.
- [12] G. Panchal, A. Ganatra, Y. Kosta, and D. Panchal, "Behaviour analysis of multilayer perceptrons with multiple hidden neurons and hidden layers," *International Journal of Computer Theory and Engineering*, vol. 3, no. 2, pp. 332–337, 2011.
- [13] B. Warsito, R. Santoso, H. Yasin *et al.*, "Cascade forward neural network for time series prediction," in *Journal of Physics: Conference Series*, vol. 1025, no. 1. IOP Publishing, 2018, p. 012097.
- [14] D. Veitch, "Wavelet neural networks," *Produced by University of UK*, 2005.
- [15] "Matlab 9.6.0.1150989 (r2019a) update 4," Natick, Massachusetts, 2019.
- [16] D. Aleksovski, J. Kocijan, and S. Džeroski, "Model-tree ensembles for noise-tolerant system identification," *Advanced Engineering Informatics*, vol. 29, no. 1, pp. 1–15, 2015.

A Método de Reamostragem

```

1 %% Import data from text file
2 % Script for importing data from the following text file:
3 %
4 % filename: /Users/danieldacosta/Documents/TCC_Matlab/Model/270519/tek0002.csv
5 %
6 % Auto-generated by MATLAB on 08-Sep-2019 18:24:56
7
8 clc;
9 clear;
10 close all;
11 %% Setup the Import Options
12 opts = delimitedTextImportOptions("NumVariables", 4);
13
14 % Specify range and delimiter
15 opts.DataLines = [22, 10000022];
16 opts.Delimiter = ",";
17
18 % Specify column names and types
19 opts.VariableNames = ["TIME", "GER", "AMP", "PIEZO"];
20 opts.VariableTypes = ["double", "double", "double", "double"];
21 opts.ExtraColumnsRule = "ignore";
22 opts.EmptyLineRule = "read";
23
24 % Import the data
25 tbl = readtable("/Users/danieldacosta/Documents/TCC_Matlab/Model/270519/tek0002.csv", opts);
26
27 %% Convert to output type
28 Ts_orig = 1e-8;
29 time = tbl.TIME;
30 DATA = [tbl.GER tbl.AMP tbl.PIEZO];
31 % Removendo tempos repetidos, devido a elevada frequência de amostragem
32 [time1, indtime] = unique(time);
33 DATA1 = DATA(indtime,:);
34 nans = any(isnan(DATA1));
35 DATA2 = DATA1(~nans,:); % Eliminado os valores NANS
36 time2 = time1(~nans);
37
38 figure
39 plot(time2, DATA2(:,1), 'k-')
40 figure
41 plot(time2, DATA2(:,3), 'k-')
42
43 figure
44 periodogram(DATA2(:, [1 3]), [], [], 1/Ts_orig)
45 legend('u', 'y')
46
47 % filtrar o sinal para a banda de interesse
48 [B,A] = butter(5, [0.16 0.3]); % passband in normalized frequency
49 figure
50 freqz(B,A)
51 dataOut = filter(B,A,DATA2);
52 figure
53 periodogram(dataOut(:, [1 3]), [], [], 1/Ts_orig)
54 legend('u', 'y')
55
56 timeVector = time2;
57 PIEZO_trunc = dataOut(:,3);
58 GER_trunc = dataOut(:,1);
59 AMP_trunc = dataOut(:,2);
60
61 %% Linear Autocorrelation
62 [r, lags] = autocorr(PIEZO_trunc, 'NumLags', length(timeVector)-2);
63 TF = islocalmin(r);
64 index_min_linear = find(TF, 1, 'first');
65 r_min_linear = r(index_min_linear);
66 tau_min = lags(index_min_linear);
67
68 figure
69 plot(lags, r)
70 hold on
71 plot(tau_min, r_min_linear)
72 legend('r', 'r_{min}')
73
74 %% Non Linear Autocorrelation
75 PIEZO2_trunc = PIEZO_trunc.^2;
76 [r2, lags2] = autocorr(PIEZO2_trunc, 'NumLags', length(timeVector)-2);
77 TF2 = islocalmin(r2);
78 index_min_nonlinear = find(TF2, 1, 'first');
79 r_min_linear2 = r(index_min_nonlinear);

```

```
80 tau_min2 = lags2(index_min_nonlinear);
81
82 figure
83 plot(lags2,r2)
84 hold on
85 plot(tau_min2, r_min_linear2)
86 legend('r2','r2_{min}')
87
88 %The smallest lag;
89 tau_m = min(abs(tau_min), abs(tau_min2));
90 if tau_m > 20
91     Δ = tau_m/20; %decimation factor. tau*_m should be between [10,20] or [5,25]
92 elseif tau_m < 5
93     Δ = 10/tau_m;
94 end
95
96 Ts_reduced = Ts_orig*Δ;
97 fSampling = 1/Ts_reduced;
```

B Geração do Sinal Reamostrado e Filtrado

```

1 %Daniel Pereira da Costa
2
3 %Building new models using Chirp signal from June;
4 % tek002 - Chirp 900 a 1350 80MVpp. Sweep time: 100Ms
5
6 clc;
7 clear;
8 close all;
9
10 %% Setup the Import Options
11 opts = delimitedTextImportOptions("NumVariables", 4);
12
13 % Specify range and delimiter
14 opts.DataLines = [22, 10000022];
15 opts.Delimiter = ",";
16
17 % Specify column names and types
18 opts.VariableNames = ["TIME", "GER", "AMP", "PIEZO"];
19 opts.VariableTypes = ["double", "double", "double", "double"];
20 opts.ExtraColumnsRule = "ignore";
21 opts.EmptyLineRule = "read";
22
23 % Import the data
24 tbl = readtable("/Users/danieldacosta/Documents/TCC_Matlab/Model/270519/tek0002.csv", opts);
25
26 %% Resampling Signal to new Sampling Rate
27 Ts_orig = 1e-8;
28 Ts_reduced = 1e-7;
29 time = tbl.TIME;
30 DATA = [tbl.GER tbl.AMP tbl.PIEZO];
31
32 % pro algum motivo tem tempos repetidos
33 [time1, indtime] = unique(time);
34 DATA1 = DATA(indtime,:);
35 nans = any(isnan(DATA1));
36 DATA2 = DATA1(~nans,:); % por algum motivo tinhamos NaNs
37 time2 = time1(~nans);
38
39 [DATAr,timer] = resample(DATA2,time2,1/Ts_reduced,1,1);
40 [DATAr2,timer2] = resample(DATA1,time1,1/Ts_orig,1,1); % apenas corrige as amostras nao espacadas igualmente
41
42 % Original Data vs. Resample:
43 % 1- vejam que o sinal nao possui uma curva bem definida (dar zoom na
44 % horizontal - x), acredito que isto se deva a resolucao da medicao. seria
45 % interessante comparar com os dados com amplitude maior da nova aquisicao
46 figure
47 plot(time2,DATA2(:,1), 'k-',timer,DATAr(:,1), 'b--')
48 figure
49 plot(time2,DATA2(:,3), 'k-',timer,DATAr(:,3), 'b--')
50
51 figure
52 periodogram(DATAr2(:,[1 3]),[],[],1/Ts_orig)
53 legend('u','y')
54 figure
55 periodogram(DATAr(:,[1 3]),[],[],1/Ts_reduced)
56 legend('u','y')
57 % algumas frequencias aparecem na saida devido (acredito) as
58 % nao-linearidade
59
60 %% Filtrar o sinal para a banda de interesse
61 [B,A] = butter(5,[0.16 0.3]); % passband in normalized frequency
62 figure
63 freqz(B,A)
64 dataOut = filter(B,A,DATAr);
65 figure
66 periodogram(dataOut(:,[1 3]),[],[],1/Ts_reduced)
67 legend('u','y')
68
69 figure
70 plot(timer,dataOut(:,1), 'k-',timer,DATAr(:,1), 'b--')
71 figure
72 plot(timer,dataOut(:,3), 'k-',timer,DATAr(:,3), 'b--')
73
74 z = iddata(dataOut(:,3),dataOut(:,1),Ts_reduced);

```

C Código para iteração dos ordens dos modelos ARMAX e ARX

```
1 %% Estimate ARMAX Models Iteratively
2 fid = fopen('ARMAX_Results_article.csv','a+');
3 fprintf(fid, '\n');
4
5 na = 1:10;
6 nb = 1:10;
7 nc = 1:10;
8 nk = 1;
9 ct = 1;
10 for i = 1:length(na)
11     na_ = na(i);
12     for m = 1:length(nb)
13         nb_ = nb(m);
14         for j = 1:length(nc)
15             nc_ = nc(j);
16             for k = 1:1
17                 nk_ = nk(k);
18                 models = armax(z_article,[na_ nb_ nc_ nk_]);
19                 kstep = [1 2 3 10 20 50 100];
20                 fprintf(fid, '%d;%d;%d;', na_,nb_,nk_);
21                 nk_step = length(kstep);
22                 for l=1:nk_step
23                     [y,fit,x0] = compare(z_article,models,kstep(l));
24                     fprintf(fid, '%d;', fit);
25                 end
26                 fprintf(fid, '\n');
27             end
28         end
29     end
30 end
31 end
32 fclose(fid)
```

D Código para a variação dos parâmetros do modelo NARX

```

1 % Daniel Pereira da Costa
2 % Load input and output data from file: ResampledData.sid
3
4 clc;
5 clear;
6 close all;
7
8 %% PreprocessType = 2; % 1 or 2 – Subtract offset or trend from time-domain signals contained in iddata ...
   objects
9
10 na = 10;
11 nb = 10;
12
13 opt = nlarxOptions;
14 opt.Display = 'full';
15 opt.SearchOptions.MaxIterations = 10;
16 % Neural Network Configuration
17
18 %% FeedForward
19 net = feedforwardnet(5);
20
21 net_estimator = neuralnet(net);
22 mdl2 = nlarx(z_article,[na nb 1], net_estimator,opt);
23
24 kstep = [1 2 3 10 20 50 100];
25 %% CascadeForwardnet
26 net = cascadeforwardnet(5);
27
28 net_estimator = neuralnet(net);
29 mdl2 = nlarx(z_range,[na nb 1], net_estimator,opt);
30
31 kstep = [1 2 3 10];
32
33 %%
34 N_layer1 = 5;
35 N_layer2 = 4;
36 net = cascadeforwardnet(N_layer1);
37
38 net_estimator = neuralnet(net);
39
40 fid=fopen('NARX_CF_Results_article.csv','a+');
41
42 fprintf(fid, '\nna,nb,Neuron1,R2_OSA1,R2_OSA2,R2_OSA3,R2_OSA10,R2_OSA20,R2_OSA50,R2_OSA100,Free_Run\n');
43 fprintf(fid, '\n');
44 nk = length(kstep);
45 for na_ = 10:10
46     for nb_ = 10:10
47         mdl2 = nlarx(z_article,[na_ nb_ 1], net_estimator,opt);
48         kstep = [1 2 3 10 20 50 100];
49         nk = length(kstep);
50         fprintf(fid, '%d; %d; %d;', na_, nb_, N_layer1);
51         for i=1:nk
52             [y, fit, x0] = compare(z_article, mdl2, kstep(nk));
53             fprintf(fid, '%d;', fit);
54         end
55         fprintf(fid, '\n');
56     end
57 end
58 fclose(fid);

```