

Monografia de Graduação

UTILIZAÇÃO DE REDES NEURAIS ARTIFICIAIS NO CONTROLE DE SISTEMAS MECÂNICOS INCERTOS

JOÃO DEODATO BATISTA DOS SANTOS

Natal, fevereiro de 2012

JOÃO DEODATO BATISTA DOS SANTOS

**UTILIZAÇÃO DE REDES NEURAS ARTIFICIAIS NO CONTROLE DE
SISTEMAS MECÂNICOS INCERTOS**

Monografia produzida a título de Trabalho de Conclusão de Curso de Graduação em Engenharia Mecânica, apresentada à Coordenação do Curso de Engenharia Mecânica da Universidade Federal do Rio Grande do Norte, como requisito à obtenção da nota da atividade DEM0551 – Trabalho de Conclusão de Curso II.

Orientador: Prof. Dr. Wallace Moreira Bessa.

NATAL

2011

FOLHA DE APROVAÇÃO

JOÃO DEODATO BATISTA DOS SANTOS

UTILIZAÇÃO DE REDES NEURAIS ARTIFICIAIS NO CONTROLE DE SISTEMAS
MECÂNICOS INCERTOS

Monografia produzida a título de Trabalho de Conclusão de Curso de Graduação em Engenharia Mecânica, apresentada à Coordenação do Curso de Engenharia Mecânica da Universidade Federal do Rio Grande do Norte, como requisito à obtenção da nota da atividade DEM0551 – Trabalho de Conclusão de Curso II.

Natal, 22 de dezembro de 2011.

BANCA EXAMINADORA/COMISSÃO DE AVALIAÇÃO

Wallace Moreira Bessa, D.Sc.– Orientador

Raimundo Carlos Silvério Freire Júnior, D.Sc

Márcio Valerio de Araújo, M.Sc.

UTILIZAÇÃO DE REDES NEURAI ARTIFICIAIS NO CONTROLE DE SISTEMAS MECÂNICOS INCERTOS

João Deodato Batista dos Santos¹

RESUMO

Atualmente há uma constante busca por métodos que minimizem o consumo de energia, em virtude da preocupação em reduzir os níveis da poluição e da necessidade cada vez maior do desenvolvimento de equipamentos mais seguros. Neste contexto, o desenvolvimento de controladores mais eficientes permite que esses objetivos possam ser atingidos. No entanto, a maioria dos sistemas físicos existentes na natureza apresentam características não lineares, o que dificulta o seu controle através das abordagens lineares tradicionais. Além disso, muitas vezes estes sistemas apresentam incertezas ou características dinâmicas difíceis de serem determinadas com exatidão, o que por sua vez também dificulta a obtenção de sistemas de controle mais eficientes. Este trabalho tem caráter expositivo e busca apresentar uma nova estratégia de controle, baseada na associação de metodologias de controle não linear com redes neurais do tipo ADALINE comparando-a com a aproximação de funções por mínimos quadrados. Para ilustrar o desenvolvimento da lei de controle, a estratégia proposta é aplicada ao controle do movimento unidirecional um submarino robô e de um sistema eletro-hidráulico. Resultados numéricos obtidos computacionalmente confirmam a capacidade do controlador em lidar com as não linearidades e incertezas do sistema controlado.

PALAVRAS-CHAVE: ADALINE, Mínimos quadrados, Controle adaptativo, Sistemas não lineares

¹ João Deodato Batista dos Santos, Graduando em Engenharia Mecânica, Universidade Federal do Rio Grande do Norte

LISTA DE FIGURAS

Figura 1	Trajectoria senoidal para movimentação unidirecional em “z” de um submarino robô, com 30% de perturbação comparando o ajuste através de Adaline, de mínimos quadrados e sem ajuste dos pesos	19
Figura 2	Comparação das diferenças entre a movimentação desejada é a obtida para a trajetória senoidal para movimento unidirecional em z de um submarino robô, com 30% de perturbação .	19
Figura 3	Trajectoria triangular para movimentação unidirecional em “z” de um submarino robô, com 30% de perturbação comparando o ajuste através de Adaline, de mínimos quadrados e sem ajuste dos pesos.	20
Figura 4	Comparação das diferenças entre a movimentação desejada é a obtida para a trajetória triangular para movimento unidirecional em z de um submarino robô, com 30% de perturbação externa.	20
Figura 5	Trajectoria quadrada para movimentação unidirecional em “z” de um submarino robô, com 30% de perturbação comparando o ajuste através de Adaline, de mínimos quadrados e sem ajuste dos pesos.	21
Figura 6	Comparação das diferenças entre a movimentação desejada é a obtida para a trajetória em quadrada para movimento unidirecional em z de um submarino robô, com 30% de perturbação externa.	21
Figura 7	Sistema eletrohidráulico.	23
Figura 8	Não linearidade de zona morta.	25
Figura 9	Trajectoria senoidal para movimentação unidirecional em “z” de um atuador eletrohidráulico, com 30% de perturbação comparando o ajuste através de Adaline, de mínimos quadrados e sem ajuste dos pesos.	29

- Figura 10** Comparação das diferenças entre a movimentação desejada e a obtida para a trajetória senoidal para movimento de um sistema eletro-hidráulico, com 30% de perturbação externa **29**
- Figura 11** Trajetória triangular para movimentação unidirecional em “z” de um submarino robô, com 30% de perturbação externa, comparando ajustes através de Adaline e sem ajuste. **30**
- Figura 12** Comparação das diferenças entre a movimentação desejada e a obtida para a trajetória triangular para movimento de um sistema eletro-hidráulico, com 30% de perturbação externa. **30**
- Figura 13** Trajetória triangular para movimentação unidirecional em “z” de um submarino robô, com 30% de perturbação externa, comparando ajustes através de Adaline e sem ajuste. **31**
- Figura 14** Comparação das diferenças entre a movimentação desejada e a obtida para a trajetória em patamares para movimento unidirecional em z de um submarino robô, com 30% de perturbação externa. **31**

SUMÁRIO

1	INTRODUÇÃO	7
2	OBJETIVOS	8
2.1	OBJETIVO GERAL	8
2.2	OBJETIVOS ESPECÍFICOS	8
3	REVISÃO BIBLIOGRÁFICA	9
3.1	REDES NEURAIS ARTIFICIAIS	9
3.2	ADELINE	9
3.3	MÍNIMOS QUADRADOS	10
3.4	RUNGE-KUTTA	12
4	METODOLOGIA	14
5	APLICAÇÕES	15
5.1	PARA SISTEMA DE SEGUNDA ORDEM	15
5.2	PARA SISTEMA DE TERCEIRA ORDEM	23
6	CONCLUSÃO	33
	REFERÊNCIAS	
	APÊNDICE	

1 INTRODUÇÃO

As redes neurais artificiais são técnicas computacionais que apresentam seu funcionamento inspirado na estrutura neural de organismos inteligentes. Tem a capacidade de adquirir conhecimento a partir da interação com o ambiente e as experiências proporcionadas por essa interação. Por isso são amplamente utilizadas quando há necessidade de imitar o processo de inferência humana.

No entanto, sua aplicação ultrapassa seu uso tradicional, sendo utilizada na aproximação de funções e na modelagem de sistemas dinâmicos, através da identificação de parâmetros a partir de séries temporais. Desta forma, pode-se aplicar redes neurais artificiais, RNA's, no reconhecimento dos parâmetros que regem o comportamento de sistemas complexos e, eventualmente, reproduzi-los e controlá-los.

Atualmente realiza-se um esforço considerável com a intenção de combinar estratégias de controle não-linear com as metodologias de inteligência artificial, com objetivo de utilizar as redes neurais para melhorar o desempenho dos controladores normalmente adotados, visando principalmente o controle de sistemas mecânicos, em especial o tratamento de sistemas com elevado grau de incerteza.

Um sistema mecânico incerto consiste em um sistema onde se desconhece ou não se torna possível prever antecipadamente os seus parâmetros de funcionamento. Isso poderá acontecer devido a diferença entre o modelo matemático do sistema e a realidade, por terem comportamento não-linear ou simplesmente o sistema real apresentar perturbações e imprevisibilidades que podem ser causadas pelo ambiente onde ele se insere, dificultando o controle eficiente desses sistemas através de técnicas de controle convencionais.

No contexto atual, onde as questões ambientais e de sustentabilidade são preocupações constantes, o desenvolvimento de técnicas de controle inteligentes podem permitir um consumo mais eficiente de energia, o que tem-se mostrado extremamente importante. De forma paralela, o crescente número de trabalhos nos últimos anos dedicados ao problema de controle inteligente de sistemas mecânicos confirma o grande interesse pelo tema. Neste trabalho aplica-se as metodologias de redes neurais na modelagem de sistemas mecânicos incertos para viabilizar sua aplicação no desenvolvimento de sistemas de controle inteligente.

2 OBJETIVOS

2.1 OBJETIVO GERAL

Construir o algoritmo de controle para um sistema mecânico incerto de segunda e terceira ordem, utilizando Redes Neurais Artificiais como aproximador de funções afim de obter uma metodologia de controle que se adeque melhor as necessidades do problema, analisando suas peculiaridades à medida que o sistema funciona.

2.2 OBJETIVOS ESPECÍFICOS

Entender o funcionamento dos diferentes métodos de reconhecimento de parâmetros contidos na literatura;

Implementar os algoritmos de mapeamento dos parâmetros a serem usados para o controle;

Testar os algoritmos através de simuladores para comprovar e compreender se sua aplicação será eficiente quando utilizado para o controle do sistema mecânico;

Analisar sua viabilidade como controlador.

3 REVISÃO BIBLIOGRÁFICA

3.1 REDES NEURAIS ARTIFICIAIS

As redes neurais artificiais compreendem um conjunto de metodologias de cálculo inspirado nas estruturas neurais de organismos inteligentes. Como pode ser observado em JANG, SUN e MIZUTANI (1997) ou em CASTRO e CASTRO (2000), as redes neurais artificiais (RNAs) são metodologias de cálculo paralelo distribuído, compostas por unidades de cálculo simples, neurônios artificiais, que calculam determinadas funções matemáticas normalmente não lineares. Estas são dispostas em uma ou mais camadas interligadas por um grande número de conexões. Na forma mais comum trabalha em uma única direção.

Os modelos mais utilizados possuem pesos que ponderam cada conexão. Nesses pesos, armazena-se o conhecimento adquirido pelo modelo. Dessa forma, os pesos são ajustados através da interação com os valores de entrada e saída aplicado à rede. Esses métodos possibilitam a aprendizagem das características do sistema no qual a rede neural está inserida.

O aprendizado de uma RNA consiste em um processo iterativo de ajustes dos parâmetros da rede, que são os pesos das conexões. Após o processo de aprendizado, são nesses pesos que se guarda o conhecimento adquirido do ambiente externo à rede. Dessa forma, o aprendizado é definido como o processo pelo qual os parâmetros livres de uma rede neural são ajustados por meio de uma forma contínua de estímulo pelo ambiente externo, sendo cada tipo específico de aprendizado definido pela maneira particular como ocorrem os ajustes dos parâmetros livres.

3.2 ADALINE

Segundo BRAGA; CARVALHO; LUDERMIR (2007), o Adaline é um tipo de RNA que pode ser classificado como um aproximador de funções a partir de um conjunto de pontos. Seu sistema de aprendizado caracteriza-se pela minimização da função do somatório dos erros quadráticos, realizando ponderações a cada interação na direção do erro mínimo, para produzir uma função de forma a melhor representar esse conjunto de pontos.

Admitindo uma função de aproximação do tipo:

$$f(x) = c_0 \cdot \varphi_0(x) + c_1 \cdot \varphi_1(x) + \dots + c_m \cdot \varphi_m(x) \quad (1)$$

Onde os “c” são as constantes que se pretende encontrar e os φ são funções que dependem do tipo da natureza dos dados. A função somatório dos erros quadráticos se torna:

$$R = \sum_{i=0}^{i=n} r_i^2 = \sum_{i=0}^{i=n} (Y_i - f(X_i))^2 = \sum_{i=0}^{i=n} ((Y_i - c_0 \cdot \varphi_0(x) + c_1 \cdot \varphi_1(x) + \dots + c_m \cdot \varphi_m(x)))^2 \quad (2)$$

Calculando o gradiente da função somatório dos erros quadráticos (2), para encontrar o valor mínimo dessa função, e conseqüentemente encontrar uma função que melhor aproxime o conjunto de pontos e admitindo que em um número finito de interações a fórmula recursiva retornaria um valor com um erro admissível. Encontra-se a seguinte fórmula recursiva:

$$\Delta c_{ij} = \eta \cdot e_i \cdot \varphi_{ij} \quad (3)$$

$$e_i = Y_i - (c_0 \cdot \varphi_0(X_{i-1}) + c_1 \cdot \varphi_1(X_{i-1}) + \dots + c_m \cdot \varphi_m(X_{i-1})) \quad (4)$$

$$\text{ou } e_i = Y_i - f(X_{i-1})$$

Onde tem-se que $f(X_{i-1})$ na equação (4) é o valor estimado pela rede a partir do conhecimento adquirido nas interações anteriores. Y_i : Valor retornada pelo sistema. e_i : o erro a cada interação, que é a diferença entre o valor retornado e o valor estimado pela rede. E “ η ” é o parâmetro de convergência, ou também denominada de constante de aprendizagem da rede.

Este método pode ser utilizado para a aproximação de funções da mesma forma que o método de mínimos quadrados, com apenas uma diferença, o Adaline pode proceder a essa aproximação ao longo das medições de forma gradativa e continua.

3.3 MÍNIMOS QUADRADOS

O método de mínimos quadrados é uma técnica de aproximação de funções também conhecido como regressão linear. É o método de aproximação de funções mais utilizado na atualidade. Podendo ser encontrado na literatura como em STRANG(1988) e LAWSON (1997), este método procura uma função que melhor corresponda ao conjunto de dados ao qual se refere, tentando minimizar a soma dos quadrados das diferenças, equação (2), entre o valor estimado e os dados observados.

Como exemplo um conjunto de dados do tipo $\{(a_1, b_1); (a_2, b_2); (a_3, b_3) \dots (a_n, b_n)\}$ que se deseja aproximar por uma reta.

Resulta em um sistema de equações (5), que só apresentará uma solução única quando b for um múltiplo de a .

$$\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \hat{d} + \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} \hat{c} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \quad (5)$$

$$\text{organizando} \quad \begin{bmatrix} 1 & a_1 \\ 1 & a_2 \\ \vdots & \vdots \\ 1 & a_n \end{bmatrix} \begin{bmatrix} \hat{d} \\ \hat{c} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \quad \text{ou} \quad A \hat{x} = B$$

Se o vetor A não for múltiplo B , não existe apenas uma única solução para a equação (5). Então o método de mínimos quadrados procura uma combinação de (\hat{d}, \hat{c}) que minimize o somatório dos erros quadrados. Derivando a equação (2) e igualando a zero obtêm-se a seguinte expressão.

$$2 \cdot \left(\sum_{i=1}^{i=n} (a_i \hat{c} + \hat{d} - b_i) \cdot a_i + (a_i \hat{c} + \hat{d} - b_i) \cdot 1 \right) = 0 \quad (6)$$

Organizando a equação 6 na forma de um sistema matricial resulta em :

$$A^t A \hat{x} = A^t B \quad (7)$$

Para uma função de aproximação do tipo de (1) o método de mínimos quadrados pode ser escrito na forma mais geral da seguinte forma:

$$\begin{bmatrix} \sum_{i=0}^{i=n} \varphi_0(X_i) \cdot \varphi_0(X_i) & \cdots & \sum_{i=0}^{i=n} \varphi_0(X_i) \cdot \varphi_m(X_i) \\ \vdots & \ddots & \vdots \\ \sum_{i=0}^{i=n} \varphi_m(X_i) \cdot \varphi_0(X_i) & \cdots & \sum_{i=0}^{i=n} \varphi_m(X_i) \cdot \varphi_m(X_i) \end{bmatrix} \cdot \begin{bmatrix} c1 \\ c2 \\ \vdots \\ cm \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{i=n} Y_i \cdot \varphi_0(X_i) \\ \sum_{i=0}^{i=n} Y_i \cdot \varphi_1(X_i) \\ \vdots \\ \sum_{i=0}^{i=n} Y_i \cdot \varphi_m(X_i) \end{bmatrix} \quad (8)$$

Se todas as colunas de A são linearmente independentes então a matriz $A^t A$ é simétrica, quadrada e inversível e, portanto, será possível aproximar uma função através desse método. No entanto, se uma ou mais colunas de A forem linearmente dependentes, não será

possível obter solução através desse método.

3.4 RUNGE -KUTTA

Os métodos de Rung-Kutta formam uma família de métodos de resolução numérica de equações diferenciais ordinárias. Estes métodos foram desenvolvidos pelos matemáticos, Carl David Tolmé Runge e Martin Wilhelm Kutta por volta de 1900. Segundo SENNE (2000) esses métodos são bastante utilizados na prática, pois são autoinicializáveis, o que quer dizer que a partir da condição inicial é possível conhecer os demais pontos da solução. Não requer o cálculo de derivadas, permite com certa facilidade a alteração do tamanho do passo e utiliza n avaliações da função a cada passo onde n é o grau do método.

Da série de Taylor:

$$y_{i+1} = y(x_{i+1}) = y(x_i + h) = y_i + h \cdot \dot{y}_i + \frac{h^2}{2} \cdot \ddot{y}(x_i) + Et \quad (9)$$

fazendo : $\dot{y} = f(x, y)$

$$y_{i+1} = y_i + h \cdot f(x, y) + \left(\frac{h^2}{2}\right) \cdot \left(\frac{d(f(x, y))}{dx}\right) + Et$$

como $\frac{d(f(x, y))}{dx} \approx \frac{1}{h} [f(x_{i+1}, y_{i+1}) - f(x_i, y_i)]$

$$y_{i+1} \approx y_i + h \cdot f(x, y) + \left(\frac{h}{2}\right) \cdot ([f(x_{i+1}, y_{i+1}) - f(x_i, y_i)]) + Et$$

e $y_{i+1} = y_i + h \cdot f(x_i, y_i)$

$$y_{i+1} \approx y_i + \left(\frac{h}{2}\right) \cdot ([f(x_{i+1}, y_{i+1}) + f(x_i, y_i)]) + Et$$

Fazendo $k_1 = h \cdot f(x_i, y_i); k_2 = h \cdot f(x_{i+1}, y_i + k_1)$

Chegando assim na equação do Runge Kutta de 2ª ordem $y_{i+1} = y_i + \frac{1}{2}(k_1 + k_2)$

utilizando uma dedução semelhante para a expansão da série de Taylor de quarta ordem chega-se a seguinte equação.

$$y_{i+1} = y_i + \frac{1}{6}(k_1 + 2 \cdot k_2 + 2 \cdot k_3 + k_4) \quad (10)$$

$$k_1 = h \cdot f(x_i, y_i)$$

$$k_2 = h \cdot f\left(x_i + \frac{h}{2}, y_i + \frac{k_1}{2}\right)$$

$$k_3 = h \cdot f\left(x_i + \frac{h}{2}, y_i + \frac{k_2}{2}\right)$$

$$k_4 = h \cdot f(x_i + h, y_i + k_1)$$

Para este trabalho foi utilizado tal metodologia a fim de conceber um simulador para realizar os testes necessários para avaliar qual é melhor forma de abordar o problema de estudo.

4 METODOLOGIA

Para realizar esse estudo, primeiro foi feita uma pesquisa bibliográfica, buscando na literatura, embasamento teórico suficiente para formular os algoritmos contidos neste estudo.

A análise adotada parte da simulação do sistema mecânico, testando se é possível controlar esse sistema através das estratégias abordadas no estudo e, dessa forma, comparar a eficiência de cada método.

Com vistas a alcançar os objetivos propostos, buscou-se integrar a literatura pertinente ao assunto com o embasamento teórico necessário para se entender a utilização das ferramentas de redes neurais, cuja finalidade é escolher o melhor método a ser utilizado para modelar o sistema mecânico incerto em estudo.

Para se testar a rede, desenvolveu-se um simulador que consiste em um algoritmo de resolução de equações diferenciais ordinárias por Runge-Kutta, de quarta ordem, resolvendo a equação do modelo matemático que consta na literatura, representando o modelo ideal, com um acréscimo de um termo extra que representa as incertezas do sistema “real”. Sendo a melhor abordagem aquela que conseguir adaptar-se mais rapidamente ao sistema gerando o menor erro possível.

5 APLICAÇÕES

Com o intuito de desenvolver um controle para um sistema mecânico incerto, adotou-se uma abordagem de controle onde se considera desconhecer, a priori, parte das características do sistema de forma que a medida que o sistema funciona o controlador se adequa as condições e variações do sistema e atinja o resultado o mais próximo possível do desejado permanecendo dentro de um valor de erro tolerado.

Para isso a metodologia de cálculo adotada consiste em estimar através de expansão de série de Taylor, as variáveis que se deseja controlar. Essa expansão é feita até o termo da série onde o erro de truncamento seja considerado admissível. Permitindo estimar para o instante seguinte a partir das condições atuais como o sistema estará se comportando, e, assim, calcular a intensidade da interferência do controlador sobre o sistema para atingir as condições desejada. As variáveis desconhecidas do sistema são aproximadas ao fim de cada interação por RNA, Adaline ou ao fim de um número predeterminado de interações por mínimos quadrados, minimizando o erro de aproximação e permitindo que o controlador estime de forma mais precisa as condições futuras do sistema e conseqüentemente minimize o erro de rastreamento .

5.1 PARA SISTEMA DE SEGUNDA ORDEM.

Para exemplificar o processo de cálculo da lei de controle adaptativo, será utilizado nesse texto como exemplo de sistema mecânico incerto de segunda ordem um submarino robô autônomo (no inglês AUV), mais especificamente ao controle de profundidade desse submarino.

A equação que rege esse sistema, desconsiderando os efeitos das correntes marinhas que pode ser obtida através de um balanço de forças é:

$$M \cdot \dot{v} = M \cdot g - A + f(t) - b \cdot |v| \cdot v \quad (11)$$

$$\text{fazendo : } p = \sqrt{\frac{|M \cdot g - A + f(t)|}{b}}$$

$$M \cdot \dot{v} = b(p^2 - |v| \cdot v)$$

Onde M é a massa do submarino, g a constante de aceleração da gravidade, A a força natural de flutuabilidade, que faz o submarino tender a boiar em caso de pane dos propulsores, $f(t)$ é a força utilizada pelos motores sobre o submarino, v a velocidade do submarino, e b a constante de arrasto.

Expandindo essa função (11) em série de Taylor até o segundo termo e aceitando o erro de truncamento admissível, encontra-se a seguinte expressão:

$$v(t) = v(0) + \dot{v}(0) \cdot \left(\frac{t}{1!}\right) + \ddot{v}(0) \left(\frac{t^2}{2!}\right) + E_t \quad (12)$$

Onde E_t é o erro de truncamento considerado como admissível.

para $t = \Delta$,

$$v(\Delta) = v_0 + (p^2 - |v_0| \cdot v_0) \cdot \left(\frac{b}{m}\right) \cdot \Delta - v_0 (p^2 - |v_0| \cdot v_0) \cdot \left(\frac{b}{m}\right)^2 \cdot \left(\frac{\Delta^2}{2}\right)$$

como : $p = \sqrt{\frac{|M \cdot g - A + f|}{b}}$ a equação se torna:

$$v(\Delta) = v_0 + \left(\frac{|M \cdot g - A + f|}{b} - |v_0| \cdot v_0\right) \cdot \left(\frac{b}{m}\right) \cdot \left(\frac{\Delta}{1!}\right) - v_0 \left(\frac{|M \cdot g - A + f|}{b} - |v_0| \cdot v_0\right) \cdot \left(\frac{b}{m}\right)^2 \cdot \left(\frac{\Delta^2}{2!}\right)$$

organizando tudo.

$$v(\Delta) - v_0 = \frac{\Delta}{1!} \cdot \left(\frac{M \cdot g - A}{M} + \frac{1}{M} \cdot f - \frac{b}{M} \cdot |v_0| \cdot v_0\right) + \left(\frac{\Delta^2}{2!}\right) \cdot \left(\frac{-(M \cdot g - A) \cdot b}{M^2} \cdot v_0 - \frac{b}{M^2} \cdot f \cdot v_0 + \frac{b^2}{M^2} \cdot v_0^2 \cdot |v_0|\right)$$

Substituindo todas as variáveis desconhecidas por W's.

chegamos a :

$$v(\Delta) - v_0 = \frac{\Delta}{1!} \cdot (W_1 + W_2 \cdot f + W_3 \cdot |v_0| \cdot v_0) + \left(\frac{\Delta^2}{2!}\right) \cdot (W_4 \cdot v_0 + W_5 \cdot |v_0| \cdot v_0^2 + W_6 \cdot f \cdot v_0) \quad (13)$$

$$x(\Delta) - x_0 = \frac{\Delta}{1!} \cdot v_0 + \left(\frac{\Delta^2}{2!}\right) \cdot (W_1 + W_2 \cdot f + W_3 \cdot |v_0| \cdot v_0) + \left(\frac{\Delta^3}{3!}\right) \cdot (W_4 \cdot v_0 + W_5 \cdot |v_0| \cdot v_0^2 + W_6 \cdot f \cdot v_0) \quad (14)$$

Onde os “W” são pesos e eles podem ser inicializados com qualquer valor, pois serão atualizados à medida que o programa é executado, através de Adaline, equação (3), e através de mínimos quadrados, equação (7), apenas com a ressalva de que os pesos W_2 e W_6 não podem ser inicializados com valor iguais a zero, como será demonstrado posteriormente. Para

realizar o controle efetivo do sistema, utiliza-se a minimização da distância entre o que se pretende a cada instante e onde possivelmente se estará no instante seguinte da seguinte forma :

$$E = k_1 \cdot (X_p - X_{t+2})^2 + k_2 \cdot (V_p - V_{t+2})^2 \quad (15)$$

Derivando a equação (15) e igualando a zero, encontra-se à seguinte expressão para a força que o controlado “recomenda” que seja utilizada para chegar ao ponto desejado.

$$f_{t+1} = \frac{k_1 \cdot C_1 \cdot ((X_p - X_{[t+1]}) - (\frac{\Delta^2}{2!}) \cdot G_1 - (\frac{\Delta^3}{3!}) \cdot G_2) + k_2 \cdot C_2 \cdot ((V_p - V_{[t+1]}) - (\frac{\Delta}{1!}) \cdot G_1 - (\frac{\Delta^2}{2!}) \cdot G_2)}{(k_1 \cdot C_1^2) + (k_2 \cdot C_2^2)} \quad (16)$$

$$C_1 = (\frac{\Delta^2}{2!}) \cdot W_2 + (\frac{\Delta^3}{3!}) \cdot W_6; C_2 = (\frac{\Delta}{1!}) \cdot W_2 + (\frac{\Delta^2}{2!}) \cdot W_6$$

$$G_1 = W_1 + W_3 \cdot |v_{t+1}| \cdot v_{t+1}; G_2 = W_4 \cdot v_{t+1} + W_5 \cdot |v_{t+1}| \cdot v_{(t+1)}^2;$$

Os termos k_1 e k_2 são constantes que dependem do tempo de interação adotado e do erro admissível desejado e que adimensionaliza a equação (15).

A equação para o ajuste dos pesos através de Adaline será:

$$e = x(\Delta) - (x_0 + \frac{\Delta}{1!} \cdot v_0 + (\frac{\Delta^2}{2!}) \cdot (W_1 + W_2 \cdot f + W_3 \cdot |v_0| \cdot v_0) + (\frac{\Delta^3}{3!}) \cdot (W_4 \cdot v_0 + W_5 \cdot |v_0| \cdot v_0^2 + W_6) \cdot f \cdot v_0)$$

$$W_1 = W_1 + (\frac{\Delta^2}{2!}) \cdot \eta \cdot e \quad W_2 = W_2 + (\frac{\Delta^2}{2!}) \cdot \eta \cdot e \cdot f \quad W_3 = W_3 + (\frac{\Delta^2}{2!}) \cdot \eta \cdot e \cdot |v_0| \cdot v_0$$

$$W_4 = W_4 + \frac{\Delta^3}{3!} \cdot \eta \cdot e \cdot v_0 \quad W_5 = W_5 + \frac{\Delta^3}{3!} \cdot \eta \cdot e \cdot |v_0| \cdot v_0^2 \quad W_6 = W_6 + \frac{\Delta^3}{3!} \cdot \eta \cdot e \cdot f \cdot v_0 \quad (17)$$

E a matriz A da equação (7) se torna :

$$A = \begin{bmatrix} 1 & f_1 & |v_1| \cdot v_1 & \frac{\Delta}{3} \cdot v_1 & \frac{\Delta}{3} |v_1| \cdot v_1^2 & \frac{\Delta}{3} \cdot f_1 \cdot v_1 \\ 1 & f_2 & |v_2| \cdot v_2 & \frac{\Delta}{3} \cdot v_2 & \frac{\Delta}{3} |v_2| \cdot v_2^2 & \frac{\Delta}{3} \cdot f_2 \cdot v_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & f_{n-1} & |v_{n-1}| \cdot v_{n-1} & \frac{\Delta}{3} \cdot v_{n-1} & \frac{\Delta}{3} |v_{n-1}| \cdot v_{n-1}^2 & \frac{\Delta}{3} \cdot f_{n-1} \cdot v_{n-1} \end{bmatrix} \quad (18)$$

Pela equação (18) pode-se constatar que só a quarta coluna de A pode ser linearmente dependente a primeira coluna de A se o sistema apresentar velocidade rigorosamente constante, o que devido as condições do sistema só acontecerá se o erro de rastreamento tender

a zero em trajetórias onde a velocidade pretendida for constante. E, como será demonstrado a seguir, devido as condições do problema, não acontece. Portanto, o método de mínimos quadrados pode ser usado para o ajuste dos pesos uma vez que a matriz $A^T A$ é inversível.

Devido as limitações mecânicas do motor a força prevista pelo controlador nem sempre será factível, assim será necessário adicionar ao programa uma análise dos limites da força.

No princípio de funcionamento do programa, o controlador em processo de adaptação de como deve proceder o controle, emite ordens que são impossíveis para o motor. A função de limitação corrige essa ordem, para a ordem mais próxima ao qual o motor consiga obedecer. Após um dado período as ordens do controlador passam a controlar esse sistema de forma adequada.

Para o sistema descrito o simulador adotado foi implementado na linguagem C++. A solução numérica para equação diferencial de segunda ordem foi obtida através do método Runge-Kutta de quarta ordem, equação (10), para um sistema de duas equações diferenciais, sendo considerado como parâmetros: $M = 400\text{kg}$, $g=9.81 \text{ m/s}$, $A=M.g +5 \text{ N}$, $b=200\text{kg/m}$, a capacidade máxima do propulsor é de $\pm 60\text{N}$ e a frequência de amostragem adotada foi de 1kHz para o simulador, e de 500Hz para o controlador. Para ratificar a viabilidade do controlador foi utilizado uma perturbação externa de 30% da capacidade do motor. Testando-o para trajetórias dos tipos senoidal, triangulares e em patamares, onde foi considerado, para o AUV um erro admissível da ordem de 5cm após a estabilização do sistema.

Neste exemplo foi analisado como o controlador lida com não-linearidades, com o retardo na resposta do motor devido sua inércia, com parâmetros desconhecidos, no caso a intensidade da constante de arrasto, e com as perturbações externas que simulam os efeitos das correntes marítimas. Com o intuito de comparar os métodos de ajuste de funções foi analisado como o controlador funciona sem ajuste, com ajustes através de mínimos quadrados e através de Adaline. Nos gráficos seguintes “z” representa a trajetória obtida, e “z-zp” o erro de rastreamento da trajetória.

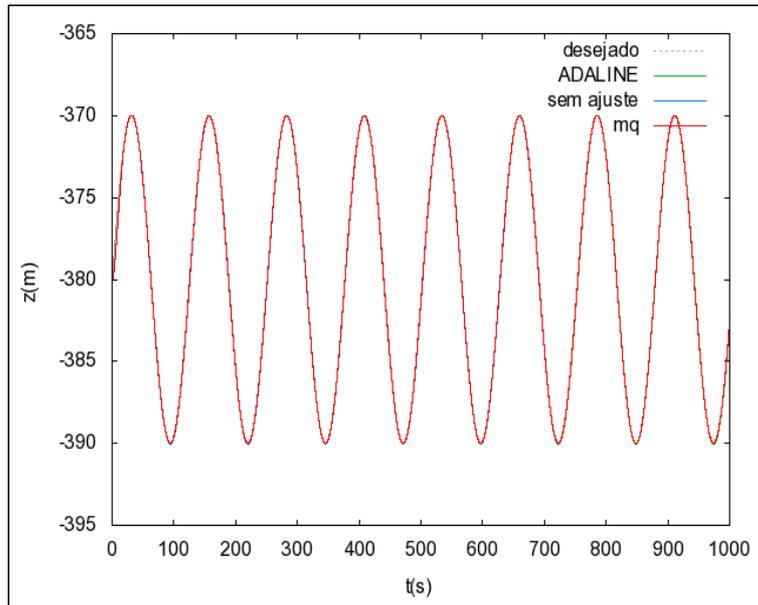


Figura 1: Trajetória senoidal para movimentação unidirecional em “z” de um submarino robô, com 30% de perturbação comparando o ajuste através de Adaline, de mínimos quadrados e sem ajuste dos pesos.

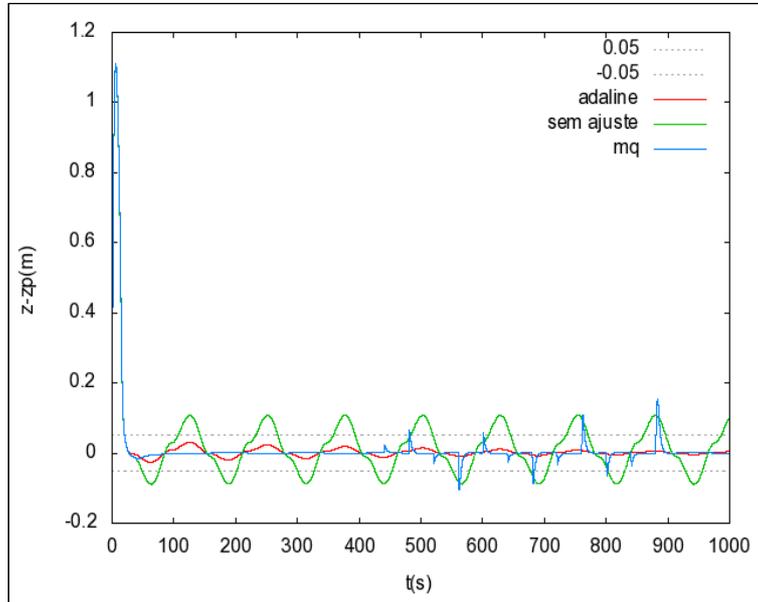


Figura 2: Comparação das diferenças entre a movimentação desejada e a obtida para a trajetória senoidal para movimento unidirecional em z de um submarino robô, com 30% de perturbação .

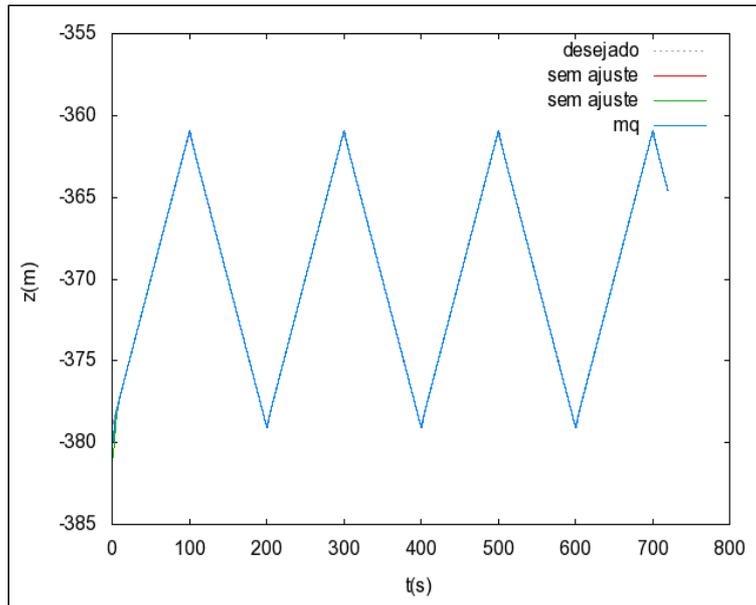


Figura 3: Trajetória triangular para movimentação unidirecional em “z” de um submarino robô, com 30% de perturbação comparando o ajuste através de Adaline, de mínimos quadrados e sem ajuste dos pesos.

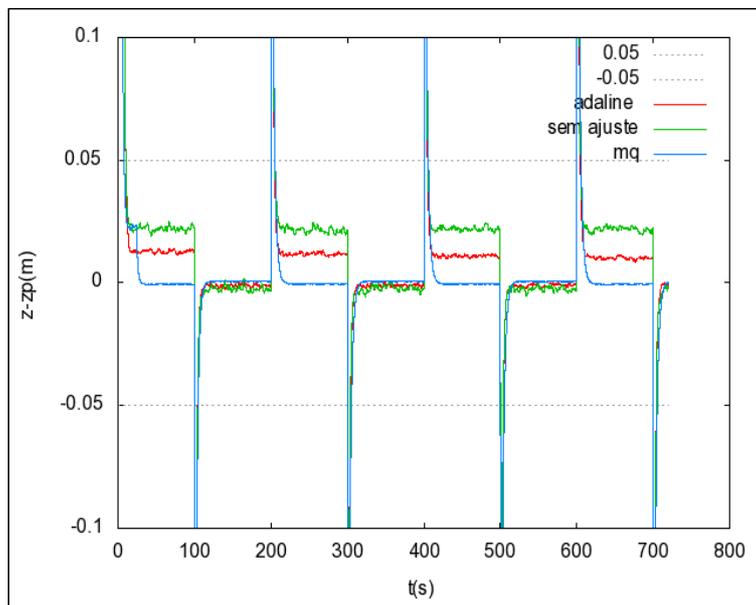


Figura 4: Comparação das diferenças entre a movimentação desejada é a obtida para a trajetória triangular para movimento unidirecional em z de um submarino robô, com 30% de perturbação externa.

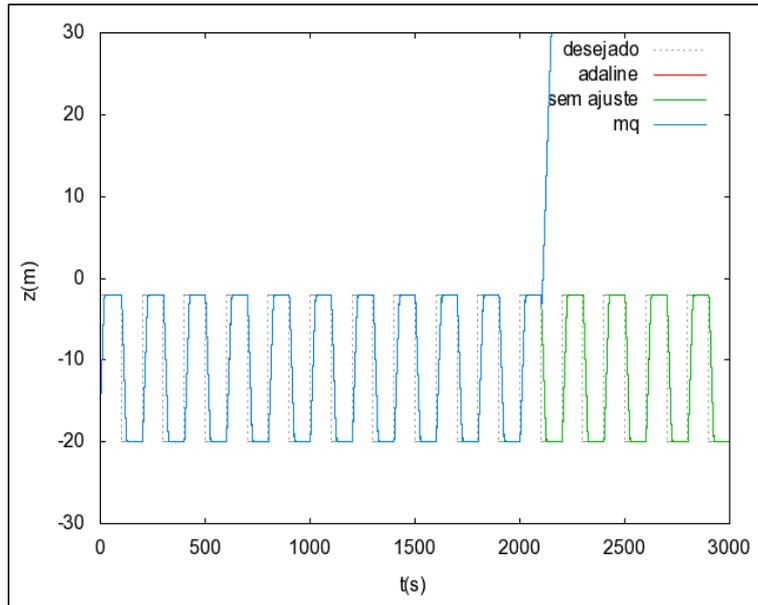


Figura 5: Trajetória quadrada para movimentação unidirecional em “z” de um submarino robô, com 30% de perturbação comparando o ajuste através de Adaline, de mínimos quadrados e sem ajuste dos pesos.

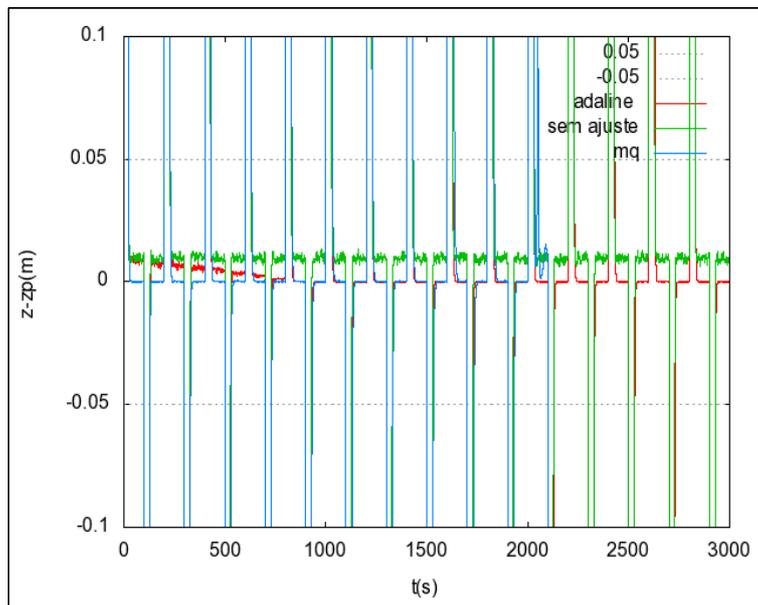


Figura 6: Comparação das diferenças entre a movimentação desejada e a obtida para a trajetória em quadrada para movimento unidirecional em z de um submarino robô, com 30% de perturbação externa.

Apesar de utilizar metodologia semelhante a usada por LABONTE (2002), os resultados obtidos com as mesmas condições operacionais, são distintos pois a metodologia usada neste trabalho não considera como parâmetro desconhecido o tempo de amostragem. O que facilita a convergência dos pesos permitindo um resultado satisfatório mais rápido, é que os parâmetros representados por W_2 e W_6 não são inicializados como zeros e sim como um valor da ordem de grandeza do valor real dessas variáveis, pois estes pesos quando inicializados como zeros resultam em divisão por zero, o que impossibilita o funcionamento do controlador.

Para a trajetória senoidal como mostrado na figura (4) foi possível constatar que o sistema de controle consegue atingir a trajetória desejada dentro dos limites de tolerância estabelecidos de ± 5 cm. Tanto com o ajuste através de Adaline, que resultou em um decaimento gradativo e constante do erro, chegando a reduzir o erro em 90% da comparação com erro sem ajuste, como também através do ajuste por mínimos quadrados. Porém, como o ajuste por mínimos quadrados não permite a contínua alteração dos pesos e, como o sistema não é constante, eventualmente, o valor aproximado por mínimos quadrados necessita de novo ajuste, resultando em pequenos picos de erro. Entretanto a redução do erro foi mais rápida do que através de Adaline.

Na trajetória triangular na figura (8) pode-se perceber que durante a inversão de movimento o fato de que o motor não consegue imediatamente mudar a direção de atuação da força resultou no desvio da trajetória obtida em relação a trajetória desejada além da tolerância pretendida mesmo quando foi usado os dois algoritmos de ajuste depois os picos de erro. O sistema descreve o erro rapidamente mesmo sem a utilização de ajuste, mas com o ajuste esse erro atenuado de forma ainda mais eficaz.

E na trajetória em patamares na figura (12) pode se perceber que após atingir o patamar seguinte, é possível manter o erro da trajetória dentro do limite tolerado mesmo sem ajuste dos pesos, mas através do ajuste por Adaline foi possível obter uma atenuação gradativa do erro, e através de mínimos quadrados uma diminuição brusca do erro.

Desta forma é possível utilizar esse método para qualquer tipo de trajetória uma vez que pode-se descrever trajetórias contínuas e diferenciável, contínuas e diferenciável por partes e descontínuas respeitando o critério de tolerância estabelecido .

5.2 PARA SISTEMA DE TERCEIRA ORDEM

Para exemplificar a aplicação do método a um sistema mecânico incerto de terceira ordem adotou-se como exemplo o sistema eletro-hidráulico. Dessa forma, a princípio é necessário desenvolver o modelo matemático do sistema eletro-hidráulico que representa o comportamento dinâmico do sistema.

Modelos dinâmicos para o sistema eletro-hidráulico podem ser encontrados na literatura como em MERRITT, (1967), BESSA, DUTRA e KREUZER, (2010b). O sistema eletro-hidráulico considerado neste presente trabalho consiste em uma válvula proporcional de quatro vias e um cilindro hidráulico submetido a um carregamento variável. O carregamento variável pode ser representado por sistema massa-mola-amortecedor, conforme na figura a baixo.

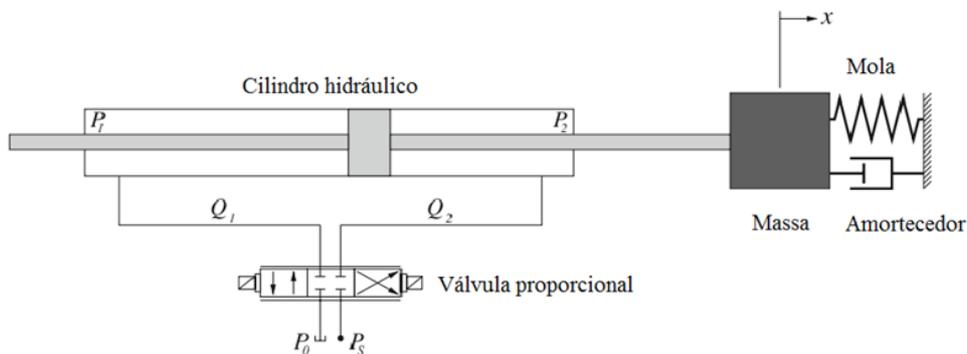


Figura 7: sistema eletrohidráulico.

Desta forma a soma das forças atuantes no sistema:

$$F_g = A_1 P_1 - A_2 P_2 = M_t \ddot{x} + B_t \dot{x} + K_s x \quad (19)$$

Onde a força realizada pelo pistão, F_g , é a diferença das forças realizadas em cada uma das faces do pistão, A_1 e A_2 são as áreas de cada face do pistão, P_1 e P_2 são as pressões atuantes nas faces do pistão, a massa total do sistema que é a soma da massa do carregamento e a massa do pistão é representada por M_t , B_t é o coeficiente de atrito viscoso, K_s é constante elástica da mola e x representa o deslocamento do pistão.

Assumindo que o cilindro é simétrico, portanto $A_p = A_1 = A_2$, e assumindo $P_d = P_1 - P_2$ a equação (19) se torna:

$$M_t \ddot{x} + B_t \dot{x} + K_s x = A_p \cdot P_d \quad (20)$$

A vazão de fluido hidráulico que entra no cilindro, $Q_l = (Q_1 + Q_2)/2$, pela equação da continuidade se divide em: volume de fluido que ocupa o espaço do cilindro durante sua movimentação $A_p \dot{x}$, pelo vazamento do pistão C_{ip} , e pela variação do volume do fluido sobre compressão $(V_t/4\beta_e) \dot{P}_l$ onde V_t é o volume total sob compressão e β_e é o módulo de elasticidade volumétrico do fluido.

$$Q_l = A_p \dot{x} + C_{ip} + \frac{V_t}{4\beta_e} \dot{P}_l \quad (21)$$

Como a pressão de retorno normalmente é muito menor que as outras pressões envolvidas no sistema pode-se considerá-la aproximadamente igual a zero, $P_0 \approx 0$, e assumindo uma válvula com orifícios simétricos, a equação para a vazão do fluido hidráulico pode ser escrita da seguinte forma :

$$Q_l = C_d w \bar{x}_{sp} \sqrt{\frac{1}{\rho} (P_s - \text{sgn}(\bar{x}_{sp}) P_l)} \quad (22)$$

Nesta equação tem-se a pressão fornecida ao sistema, P_s , a densidade do fluido hidráulico, ρ , o coeficiente de descarga, C_d , o gradiente de área do orifício da válvula, w , o deslocamento real do cilindro da válvula x_{sp} e o deslocamento efetivo que permite a passagem do fluido \bar{x}_{sp} , e a função sinal que $\text{sgn}(\cdot)$ que é definida como:

$$\text{sgn}(z) = \begin{cases} -1 & \text{se } z < 0 \\ 0 & \text{se } z = 0 \\ 1 & \text{se } z > 0 \end{cases} \quad (23)$$

Se for assumido que a dinâmica da válvula é rápida suficiente para sua influência no sistema pode ser considerada desprezível, pode-se admitir que o deslocamento do miolo do eixo pode ser considerado proporcional ao controle da tensão (u). Com o intuito de evitar vazamento algumas válvulas tem o centro fechado, ou chamadas válvulas críticas, o miolo apresenta um pouco de sobreposição ao orifício de passagem do fluido, mas leva a uma não linearidade de zona morta dentro dos limites do controle de tensão, como pode ser visto na Figura (14).

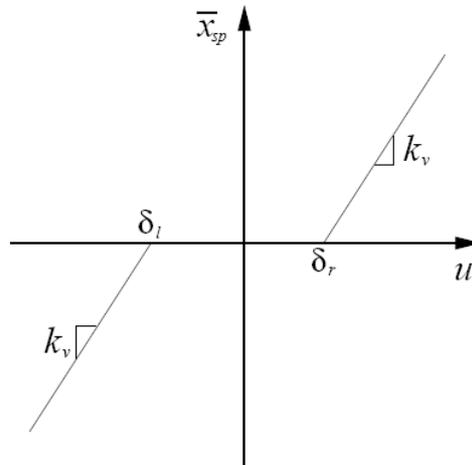


Figura 8: Não linearidade de zona morta.

A não linearidade de zona morta apresentada na figura (14) pode ser representada matematicamente por:

$$\bar{x}_{sp}(t) = \begin{cases} k_v(u(t) - \delta_l) & \text{se } u(t) \leq \delta_l \\ 0 & \text{se } \delta_l < u(t) < \delta_r \\ k_v(u(t) - \delta_r) & \text{se } u(t) \geq \delta_r \end{cases} \quad (24)$$

onde k_v é o ganho da válvula e os parâmetros δ_l e δ_r dependem do tamanho da região de sobreposição.

Para efeito de controle, como mostrado por BESSA (2010), a Equação (24) pode ser reescrita de uma forma mais adequada:

$$\bar{x}_{sp}(t) = k_v[u(t) - d] \quad (25)$$

e $d(u)$ pode ser obtido a partir das Equações (24) e (25):

$$d(u) = \begin{cases} \delta_l & \text{se } u(t) \leq \delta_l \\ u(t) & \text{se } \delta_l < u(t) < \delta_r \\ \delta_r & \text{se } u(t) \geq \delta_r \end{cases} \quad (26)$$

Através das equações (20), (21), (22), (25) e (26) é obtido uma equação diferencial de terceira ordem, representativa do comportamento dinâmico do sistema eletro-hidráulico:

$$\ddot{x} = -(a_0 \cdot x + a_1 \cdot \dot{x} + a_2 \cdot \ddot{x}) + bu - bd \quad (27)$$

os coeficientes da equação anterior são:

$$\begin{aligned}
a_0 &= \frac{4\beta_e C_{tp} K_s}{V_t M_t} \\
a_1 &= \frac{K_s}{M_t} + \frac{4\beta_e A_p^2}{V_t M_t} + \frac{4\beta_e C_{tp} B_t}{V_t M_t} \\
a_2 &= \frac{K_s}{M_t} + \frac{4\beta_e C_{tp}}{V_t} \\
b &= \frac{4\beta_e A_p}{V_t M_t} C_d w k_v \sqrt{\frac{1}{\rho} [P_s - \text{sgn}(u)(M_t \ddot{x} + B_t \dot{x} + K_s x) / A_p]} \quad (28)
\end{aligned}$$

Utilizando a equação (27) para expandir em série de Taylor até o segundo termo e aceitando o erro de truncamento admissível, encontra-se a seguinte expressão para a aceleração:

$$x^{(2)}(t) = x^{(2)}(0) + x^{(3)}(0)\left(\frac{t}{1!}\right) + x^{(4)}(0)\left(\frac{t^2}{2!}\right) + Et$$

para $t=\Delta$,
$$x_0^{(3)} = a_0 \cdot x_0 + a_1 \cdot x_0^{(1)} + a_2 \cdot x_0^{(2)} + b \cdot U - d \cdot U$$

$$x_0^{(4)} = (a_0 \cdot a_{12}) \cdot x_0 + (a_0 + a_1 \cdot a_2) \cdot x_0^{(1)} + (a_1 + a_2 \cdot a_2) \cdot x_0^{(2)} + \left(\frac{db}{dt} + b \cdot a_2\right) \cdot U - b \cdot \left(\frac{dU}{dt} + D \cdot a_2\right)$$

Fazendo:

$$G_1 = W_0 + W_1 \cdot x_0 + W_2 \cdot x_0^{(1)} + W_3 \cdot x_0^{(2)}$$

$$G_2 = W_5 + W_6 \cdot x_0 + W_7 \cdot x_0^{(1)} + W_8 \cdot x_0^{(2)}$$

$$x_\Delta^{(2)} - x_0^{(2)} = \frac{\Delta}{1!} \cdot (G_1 + W_4 \cdot U_0) + \left(\frac{\Delta^2}{2!}\right) \cdot (G_2 + W_9 \cdot U_0) \quad (29)$$

$$x_\Delta^{(1)} + x_0^{(1)} = \frac{\Delta}{1!} \cdot x_0^{(2)} + \frac{\Delta^2}{2!} \cdot (G_1 + W_4 \cdot U_0) + \left(\frac{\Delta^3}{3!}\right) \cdot (G_2 + W_9 \cdot U_0) \quad (30)$$

$$x_\Delta - x_0 = \frac{\Delta}{1!} \cdot x_0^{(1)} + \frac{\Delta^2}{2!} \cdot x_0^{(2)} + \frac{\Delta^3}{3!} \cdot (G_1 + W_4 \cdot U_0) + \left(\frac{\Delta^4}{4!}\right) \cdot (G_2 + W_9 \cdot U_0) \quad (31)$$

Onde os termos W 's são aproximados através de Adaline, mas também poderiam ser aproximados através de mínimos quadrados.

Para determinar a atuação do controle sobre o sistema, utilizando a minimização da distância entre o que se pretende a cada instante e onde se está a cada instante da seguinte forma :

$$E = k1 \cdot (Xp - X_{t+2})^2 + k2 \cdot (Vp - X_{t+2}^{(1)})^2 + k3 \cdot (Ap - X_{t+2}^{(2)})^2 \quad (32)$$

derivando a equação anterior é igualando a zero, encontra-se à seguinte expressão para a atuação do controlador.

$$H_1 = k1 \cdot C_1 \cdot ((Xp - x_{[t+1]}) - (\frac{\Delta^3}{3!}) \cdot G_1 - (\frac{\Delta^4}{4!}) \cdot G_2)$$

$$H_2 = k2 \cdot C_2 \cdot ((Vp - x_{[t+1]}^{(1)}) - (\frac{\Delta^2}{2!}) \cdot G_1 - (\frac{\Delta^3}{3!}) \cdot G_2)$$

$$H_3 = k3 \cdot C_3 \cdot ((Ap - x_{[t+1]}^{(2)}) - (\frac{\Delta}{1!}) \cdot G_1 - (\frac{\Delta^2}{2!}) \cdot G_2)$$

$$U_{t+1} = \frac{H_1 + H_2 + H_3}{(k1 \cdot C_1^2) + (k2 \cdot C_2^2) + (k3 \cdot C_3^2)} \quad (33)$$

$$C_1 = (\frac{\Delta^3}{3!}) \cdot W_4 + (\frac{\Delta^4}{4!}) \cdot W_9; C_2 = (\frac{\Delta^2}{2!}) \cdot W_4 + (\frac{\Delta^3}{3!}) \cdot W_9; C_3 = (\frac{\Delta}{1!}) \cdot W_4 + (\frac{\Delta^2}{2!}) \cdot W_9$$

Os termos k1, k2 e k3 são constantes que depende do tempo de interação adotado e do erro admissível desejado e que adimensionaliza a equação (32).

A função de ajuste através de Adaline para o sistema eletro-hidráulico se torna:

$$e = x_\Delta - (x_0 + \frac{\Delta}{1!} \cdot x_0^{(1)} + \frac{\Delta^2}{2!} \cdot x_0^{(2)} + \frac{\Delta^3}{3!} \cdot (G_1 + W_4 \cdot U_0) + (\frac{\Delta^4}{4!}) \cdot (G_2 + W_9 \cdot U_0))$$

$$W_0 = W_0 + (\frac{\Delta^3}{3!}) \cdot \eta e \quad W_1 = W_1 + (\frac{\Delta^3}{3!}) \cdot \eta e x_0 \quad W_2 = W_2 + (\frac{\Delta^3}{3!}) \cdot \eta e \cdot x_0^{(1)}$$

$$W_3 = W_3 + (\frac{\Delta^3}{3!}) \cdot \eta e x_0^{(2)} \quad W_4 = W_4 + \frac{\Delta^3}{3!} \cdot \eta \cdot e \cdot U_0 \quad W_5 = W_5 + \frac{\Delta^4}{4!} \cdot \eta \cdot e$$

$$W_6 = W_6 + (\frac{\Delta^4}{4!}) \cdot \eta e x_0 \quad W_7 = W_7 + (\frac{\Delta^4}{4!}) \cdot \eta e \cdot x_0^{(1)} \quad W_8 = W_8 + (\frac{\Delta^4}{4!}) \cdot \eta e x_0^{(2)}$$

$$W_9 = W_9 + \frac{\Delta^4}{4!} \cdot \eta \cdot e \cdot U_0 \quad (34)$$

E a matriz A na equação (7) será:

$$A = \begin{bmatrix} 1 & x_1 & x_1^{(1)} & x_1^{(2)} & U_1 & \frac{\Delta}{4} & \frac{\Delta}{4}x_1 & \frac{\Delta}{4}x_1^{(1)} & \frac{\Delta}{4}x_1^{(2)} & \frac{\Delta}{4}U_1 \\ 1 & x_2 & x_2^{(1)} & x_2^{(2)} & U_2 & \frac{\Delta}{4} & \frac{\Delta}{4}x_2 & \frac{\Delta}{4}x_2^{(1)} & \frac{\Delta}{4}x_2^{(2)} & \frac{\Delta}{4}U_2 \\ \vdots & \vdots \\ 1 & x_n & x_n^{(1)} & x_n^{(2)} & U_n & \frac{\Delta}{4} & \frac{\Delta}{4}x_n & \frac{\Delta}{4}x_n^{(1)} & \frac{\Delta}{4}x_n^{(2)} & \frac{\Delta}{4}U_n \end{bmatrix} \quad (35)$$

Para esse problema a matriz A (35) apresenta as 5 últimas colunas linearmente dependentes às 5 primeiras colunas. Dessa forma, não é possível usar o truncamento da série de Taylor até o terceiro termo. Para trajetórias contínuas e simétricas, como exemplo a senoidal, quando o erro de rastreamento se torna muito pequeno a quarta coluna tende a se tornar um múltiplo da segunda coluna, fazendo com que o determinante da matriz $A^T A$ no limite se aproxime de zero impedindo seu funcionamento. Para as trajetórias lineares, quando o erro tende a zero a velocidade tende a uma constante, conseqüentemente a terceira coluna passa a ser linearmente dependente a primeira coluna, não sendo linearmente dependente apenas nas mudanças de trajetória. Como o método de mínimos quadrados necessita realizar o ajuste dos pesos numa frequência de ajuste predefinida, se no intervalo entre dois ajustes não houver mudanças de trajetória o determinante da matriz $A^T A$ tenderá a zero.

Para avaliar o desempenho do controlador foi realizada através de implementação computacional na linguagem C++ de um simulador, usando uma taxa de amostragem de 1 kHz para o simulador e 500 Hz para o controlador, com a solução da equação diferencial de 3ª ordem do modelo do sistema eletro-hidráulico, equação (27), que foi convertida em um sistema de três equações de 1ª ordem, de modo que pudessem ser simultaneamente resolvidas pelo método de Runge-Kutta de 4ª ordem, equação(10).

Os parâmetros adotados para o sistema eletro-hidráulico foram $P_s=7$ MPa, $\rho=850$ kg/m³, $C_d=0,6$, $w=2,5 \times 10^{-2}$ m, $A_p=3 \times 10^{-4}$ m², $C_{ip}=2 \times 10^{-12}$ m³/(s Pa), $\beta_e=700$ Mpa, $V_t=6 \times 10^{-5}$ m³, $M_t=250$ kg, $B_t=100$ Ns/m, $K_s=75$ N/m, $\delta_l=-0,5$ V e $\delta_r=0,5$ V. Considerado os efeitos da zona morta desconhecida além de uma perturbação externa de 30% da capacidade do sistema. Simulando-o para a trajetórias dos tipos senoidal, triangulares em patamares, onde foi considerado, para o sistema eletro-hidráulico um erro admissível da ordem de 1mm após a estabilização. E para comparar o seu funcionamento o algoritmo foi testado sem ajuste dos pesos e com ajuste através de Adaline e mínimos quadrados.

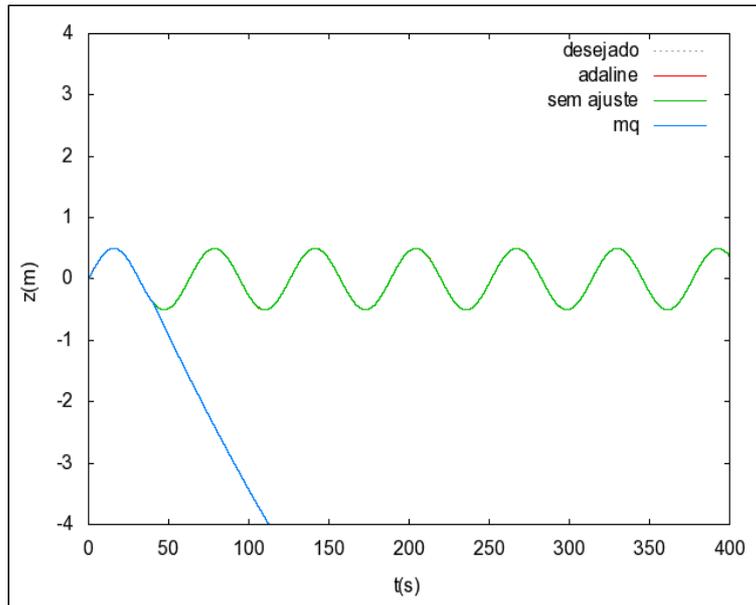


Figura 9: Trajetória senoidal para movimentação unidirecional em “z” de um atuador eletrohidráulico , com 30% de perturbação comparando o ajuste através de Adaline, de mínimos quadrados e sem ajuste dos pesos.

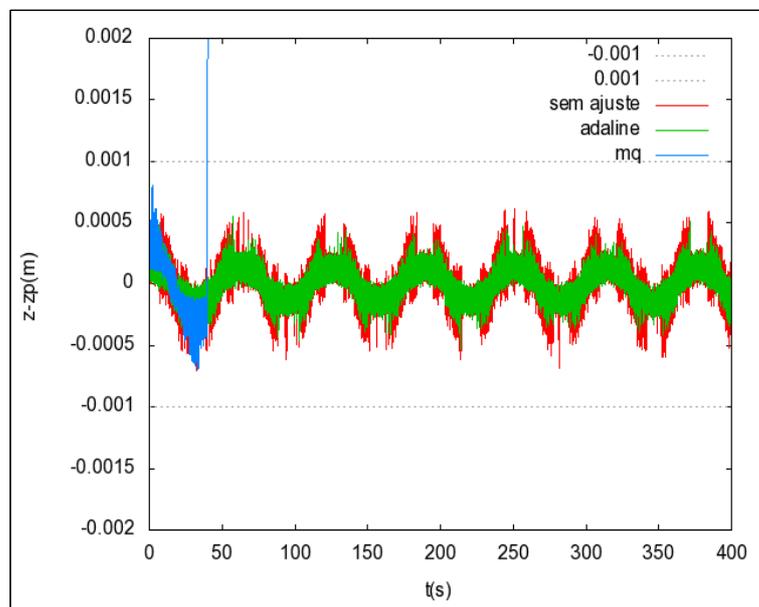


Figura 10: Comparação das diferenças entre a movimentação desejada e a obtida para a trajetória senoidal para movimento de um sistema eletro-hidráulico, com 30% de perturbação externa .

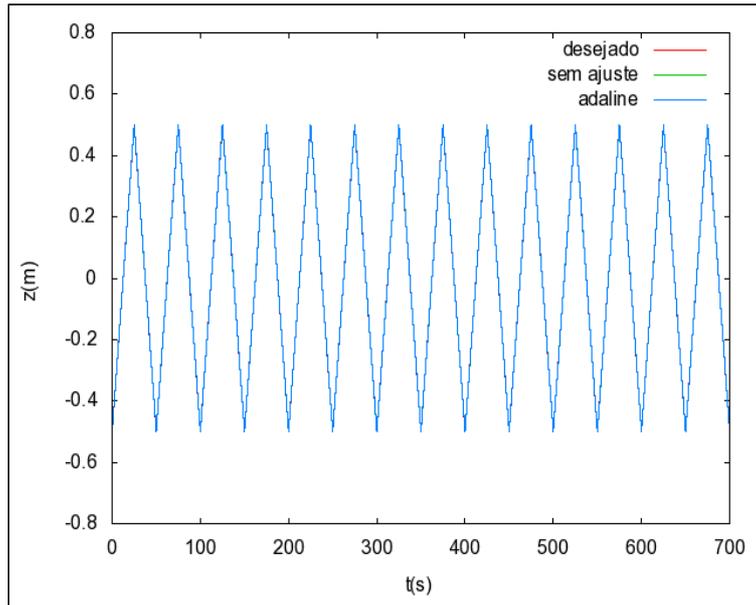


Figura 11: Trajetória triangular para movimentação unidirecional em “z” de um submarino robô, com 30% de perturbação externa, comparando ajustes através de Adaline e sem ajuste.

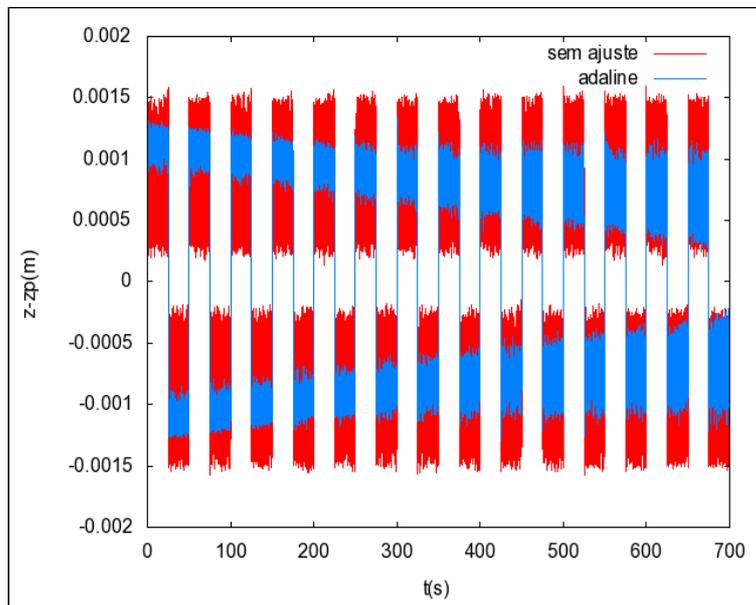


Figura 12: Comparação das diferenças entre a movimentação desejada e a obtida para a trajetória triangular para movimento de um sistema eletro-hidráulico, com 30% de perturbação externa.

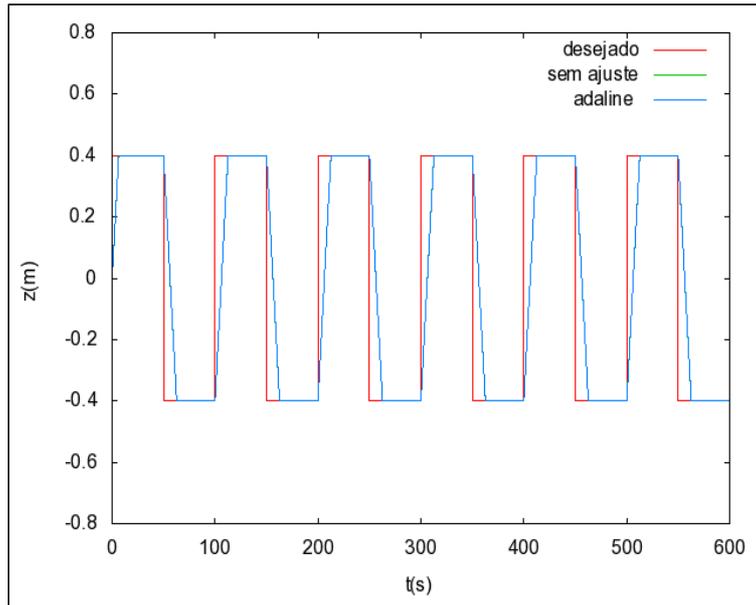


Figura 13: Trajetória triangular para movimentação unidirecional em “z” de um submarino robô, com 30% de perturbação externa, comparando ajustes através de Adaline e sem ajuste.

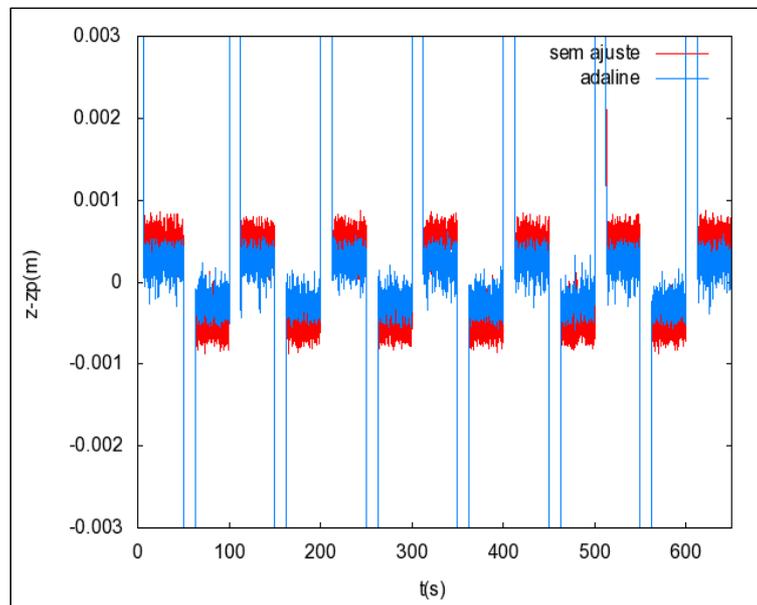


Figura 14: Comparação das diferenças entre a movimentação desejada e a obtida para a trajetória em patamares para movimento unidirecional em z de um submarino robô, com 30% de perturbação externa.

A metodologia de controle proposta foi testada para um sistema de terceira ordem, sistema eletro-hidráulico, onde se pretendia verificar se era possível descrever trajetórias contínuas, contínuas e diferenciáveis em partes, e descontínuas, respeitando o critério de que após a estabilização do sistema a diferença entre a trajetória pretendida é a obtida se reduzisse a um valor de erro tolerado. Além disso foi considerado que se desconhecia a princípio parte dos parâmetros do sistema, se conhecendo apenas a ordem de grandeza do ganho do controlador. Nesse sistema foi considerado o efeito da não linearidade de zona morta, além das perturbações externas.

Para a trajetória contínua devido ao efeito da zona morta no momento da inflexão de movimento ocorre um pico na diferença entre a trajetória obtida e a trajetória desejada. Porém mesmo na trajetória sem o ajuste através de Adaline o erro permanece dentro dos limites tolerados, e com o ajuste o erro decresceu em até 30% principalmente nas inflexões de movimento. Contudo, com o ajuste através de mínimos quadrados, uma vez que o erro decresce, acontece de que uma das colunas da matriz A se aproxime de se tornar linearmente dependente a outra, resultando na má aproximação dos pesos e consequentemente no desvio da trajetória desejada.

Na trajetória contínua e diferenciável em partes o efeito da zona morta também é notado. Entretanto o sistema retorna rapidamente para a trajetória pretendida mas apenas com o ajuste é possível manter a diferença entre a trajetória desejada é a obtida dentro dos valores tolerados .

E para a trajetória descontínua após atingir o próximo patamar desejado o sistema consegue manter a diferença entre a trajetória pretendida e a obtida dentro do valor de erro tolerado mesmo sem o ajuste através de Adaline. Mas com o ajuste é possível reduzir o erro em até 30%. Uma vez atendendo esses três tipos de trajetória pode-se afirmar que o método pode ser usado a qualquer tipo de trajetória.

6 CONCLUSÃO

Neste trabalho buscou-se compreender a metodologia de aproximação de funções através de Adaline comparando-a com a metodologia de mínimos quadrados para utilizá-lo na identificação de parâmetros para o controle de sistemas mecânicos incertos como os veículos robôs submarinos, “AUV's” que apresentam função de segunda ordem e os atuadores eletro-hidráulicos que apresentam função de terceira ordem.

Comparando os resultados obtidos com os encontrados na literatura foi possível verificar que apesar de usar metodologia semelhante, os resultados obtidos apresentaram tempo de estabilização menor, em virtude de concentrar a aproximação dos pesos nos parâmetros que apresentam maior dificuldade de se estimar com precisão.

As simulações computacionais para os algoritmos de controle propostos apontaram que é possível rastrear a trajetória desejada dentro de uma tolerância preestabelecida, seja para um sistema de segunda ordem ou de terceira ordem, desde que sua dinâmica de funcionamento seja parcialmente conhecida de forma a permitir a elaboração da lei de controle. Permitindo descrever trajetórias contínuas, contínuas e diferenciáveis por partes ou descontínuas, o que engloba todos os tipos de trajetórias.

Desta maneira, uma vez que o algoritmo de controle foi simulado computacionalmente com sucesso ele pode ser testado experimentalmente, para enfim poder ser usado em aplicações futuras onde se tenha um sistema que apresente não-linearidades ou dificuldade de se estimar os parâmetros.

REFERÊNCIAS

BESSA, W. M.; DUTRA, M. S.; KREUSER, E. *Adaptive fuzzy dead-zone compensation scheme and its application to electro-hydraulic systems*. Journal of Brazilian society of Mechanical sciences and engineering, vol 32, No 1, p. 1-7. 2010a.

BESSA, W. M.; DUTRA, M. S.; KREUZER, E. *Sliding mode control with adaptive fuzzy dead-zone compensation of an electro-hydraulic servo-system*. Journal of intelligent and robotic systems, Vol.58, No.1, p. 3-16. 2010b.

BRAGA, A.P.; CARVALHO, A. C. P. L. F. DE; LUDERMIR, T. B. *Redes neurais artificiais: teoria e aplicações*. LTC, 2007.

CASTRO, F.C.C; CASTRO, M. C . F. *Redes neurais artificiais*. PUCRS-FENG-DEE, 2000.

JANG, J.-S. R.; SUN, C.-T; MIZUTANI, E. *Neuro fuzzy and soft computing: a computational approach to learning and machine intelligence*. Prentice Hall, New Jersey, 1997.

MERRITT, H.E. *Hydraulic control systems*. John Wiley & Sons, New York. 1967.

LABONTÉ, G. *Fast adaptive control of a non-linear system by an adaline: motion in a fluid*. International joint conference on neural network, vol 3, p. 1837 – 1841, 2002.

LAWSON, T. *Álgebra linear*. Edgard Blucher, Rio de Janeiro, 1997

SENNE, E. L. F. *Cálculo numérico*. Departamento de matemática. Universidade Estadual Paulista – Unesp: Campus de Guaratinguetá. Faculdade de Engenharia, 2000.

STRANG, G. *Linear algebra and its applications third edition*. Thomson learning, 1988.

APÊNDICE

```

#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <iostream.h>
#include <math.h>
#include <fstream.h>

void rk4_2(double ,double ,double,double, double *,double *);
double Fa(double ,double,double ,double);
double adaline(double ,double , double ,double,double ,double *,double *,double *,double *,double
*,double *);
void trajetoria(double ,double *, double *) ;
void corrigir(double ,double ,double *);
void inverter(int ,double [][][6],double [][][6]);

double Fdt(double ,double ,double ,double ,double);

const double dt=1.0/500.0;// tempo de interações
const double erro=1.0/1000000.0;//erro tolerado
const double K1=1.0/(erro*erro);
const double K2=dt/(erro*erro);

int main()
{

int k;
FILE*resut, *saida0,*saida1, *saida2;
//

```

```

//posições desejadas
k=250000;
double
Xp,Vp,W0,W1,W2,W3,W4,W5,bw0,bw1,bw2,bw3,bw4,bw5,Xt,Vt,vta,xta,t,Vb,Xb,c1,c2,G1,G2;
double F,fant, FMax, Fmin, df,I,N,periodo,e1,e2,om;

double Identidade[6][6],a[6],b,Bi[6],dw[6];
double AxA[6][6];

for (int ii=0;ii<6;ii++)
{ dw[ii]=0.0;
  a[ii]=0.0;
  Bi[ii]=0.0;
  for(int jj=0;jj<6;jj++)
  {
    AxA[ii][jj]=0.0;
  }
}
if ((resut= fopen("resultadomq.dat", "w")) == NULL)
    printf("Nao foi possivel abrir o arquivo teto.dat!\n");

if ((saida0 = fopen("desejado.dat", "w")) == NULL)
    printf("Nao foi possivel abrir o arquivo teto.dat!\n");

if ((saida1 = fopen("erromq.dat", "w")) == NULL)
    printf("Nao foi possivel abrir o arquivo w1.dat!\n");

if ((saida2 = fopen("estado.dat", "w")) == NULL)
    printf("Nao foi possivel abrir o arquivo w2.dat!\n");

Xt=-380.0;
Vt=0.0;
t=0.0;
N=1.0;

```

```

F=0.00;
FMax=60.0;
Fmin=-60.0;
df=10.0;
periodo=1.0/50.0;
bw0= W0= 0.0;
bw1= W1= 0.0;
bw2= W2= 100.0/200.0;
bw3= W3= 0.0;
bw4= W4= 0.0;
bw5= W5= 0.25;
srand ( time(NULL) );

for(int j = 1; j < k; j++)
{

if( (t/N)> periodo)
{
N++;
}

trajetoria(t, &Vp , &Xp);
// correção da trajetoria

//previsão de Vk+1, Xk+1

Xb = Xt +
dt*Vt +
(dt*dt/2.0)*(W0 + W1*fabs(Vt)*Vt+ W2*F)+
(dt*dt*dt/6.0)*(W3*fabs(Vt) + W4*Vt*Vt*Vt+ W5*F*fabs(Vt));

Vb = Vt +
(dt)*(W0 + W1*fabs(Vt)*Vt+ W2*F)+
(dt*dt/2.0)*(W3*fabs(Vt) + W4*Vt*Vt*Vt+ W5*F*fabs(Vt));

```

```

c1= (dt*dt/2.0)*W2+(dt*dt*dt/6.0)*W5;
c2= dt*W2+(dt*dt/2.0)*W5;
G1= W0 + W1*fabs(Vt)*Vt;
G2= W3*fabs(Vt) + W4*Vt*Vt*Vt;
vta=Vt;
xta=Xt;

//previsão da força.

fant=F;

F= (K1*c1*((Xp-Xt)- (dt*dt/2.0)*G1 -(dt*dt*dt/6.0)*G2)+ K2*c2*((Vp-Vt)- (dt)*G1 -
(dt*dt/2.0)*G2))/(K1*c1*c1+K2*c2*c2);

// limitação da força
F = Fdt( F,fant,FMax ,Fmin ,df );

I=0.3*FMax*double(100 - rand()%200)/100.0;// perturbação

//falta o runge-kutta

for(int l=0; l<2 ; l++)
{

rk4_2(t ,t+dt ,F,I,&Vt ,&Xt);
t= t +dt ;

}

// por adaline
// adaline(F,dt, Xp,xta,vta,&w0,&w1,&w2,&w3,&w4,&w5);

```

```

// por mínimos quadrados
b=(Xt -(xta + 2.0*dt*vta));

a[0]=1.0;
a[1]=vta;
a[2]=F;
a[3]=(2.0*dt/3.0)*fabs(vta);
a[4]=(2.0*dt/3.0)*vta*vta*vta;
a[5]=(2.0*dt/3.0)*fabs(vta)*F;

for(int m=0; m<6 ; m++)
{
    Bi[m]= Bi[m]+ a[m]*b;
    for(int n=0; n<6 ; n++)
    {
        AxA[n][m]=AxA[m][n]+ a[m]*a[n];
// multiplicação de A por A transposta
    }
}

if((j%10000 ==0))
{
for(int m=0 ; m<6 ; m++)
{
for(int n=0 ; n<6 ; n++)
{
    if (m==n)
    {
        Identidade[m][n]=1.0;
    }
}
}
}

```

```

if ((m<n)||m>n)
{
  Identidade[m][n]=0.0;
}
}
}

```

```

for(int n=0;n<6;n++)
{

}
inverter(6,AxA,Identidade);

```

```

for(int n=0; n<6 ; n++)
{

dw[0]= dw[0]+ Bi[n]*Identidade[n][0];
dw[1]= dw[1]+ Bi[n]*Identidade[n][1];
dw[2]= dw[2]+ Bi[n]*Identidade[n][2];
dw[3]= dw[3]+ Bi[n]*Identidade[n][3];
dw[4]= dw[4]+ Bi[n]*Identidade[n][4];
dw[5]= dw[5]+ Bi[n]*Identidade[n][5];

}
bw0=W0;
bw1=W1;
bw2=W2;
bw3=W3;
bw4=W4;
bw5=W5;

W0=dw[0];

```

```
W1=dw[1];
```

```
W2=dw[2];
```

```
W3=dw[3];
```

```
W4=dw[4];
```

```
W5=dw[5];
```

```
}
```

```
e1=Xt -(xta + (2.0*dt)*vta + (4.0*dt*dt/2.0)*(W0 + W1*fabs(vta)*vta+ W2*F)
+(8.0*dt*dt*dt/6.0)*(W3*fabs(vta) + W4*vta*vta*vta+ W5*F*fabs(vta)));
```

```
if (fabs(Xp-Xt)>0.02)
```

```
{
```

```
W0=bw0;
```

```
W1=bw1;
```

```
W2=bw2;
```

```
W3=bw3;
```

```
W4=bw4;
```

```
W5=bw5;
```

```
}
```

```
fprintf(resut, "%.5e %.5e\n",t,Xt);
```

```
fprintf(saida0, "%.5e %.5e\n",t,Xp);
```

```
fprintf(saida1, "%.5e %.5e\n",t,Xp-Xt);
```

```
fprintf(saida2, "%.5e %.5e\n",Vp-Vt,Xp-Xt);
```

```
}
```

```

system("pause");

fclose(resut);
fclose(saida0);
fclose(saida1);
fclose(saida2);

}

void trajetoria(double n,double *V , double *X)
{

    double T,p1;
    p1 = 1.0/50.0;
    T=(n);

    double a, o;
    a=-10.0;
    o=-0.05;

    *V = a*o*cos(o*T);
    *X = a*sin(o*T)- 380.0 ;

}

double Fdt(double TF,double Fant,double Fmax ,double Fmin ,double df)
{
    double F;
    F = TF;

    if(F>Fant)
    {
        if((F < Fmax)&&(F <(Fant+df)))

```

```

{
F=F;
}
if(F > Fant+df)
{
F=Fant+df;
}
if(F > Fmax)
{
F=Fmax;
}
}
if(F< Fant)
{
if((F > Fmin)&&(F>(Fant-df)))
{
F=F;
}
if(F< Fant-df)
{
F=Fant-df;
}
if(F < Fmin)
{
F=Fmin;
}
}
return (F);
}
// Runge-Kutta de 4a ordem para EDOs de 1a ordem

void rk4_2(double t, double tf,double f,double i, double *pty0,double *pty1)
{
double ya,yb, h,k1a, k2a, k3a, k4a,k1b, k2b, k3b, k4b;
ya = *pty0;

```

```

yb = *pty1;
h = tf - t;
k1a = h * Fa(t, ya, f,i);
k1b = h * ya;
    k2a = h * Fa(t + h/2.0, ya + k1a/2.0,f,i);
    k2b = h * (ya + k1a/2.0);
        k3a = h * Fa(t + h/2.0, ya+ k2a/2.0,f,i);
        k3b = h * (ya+ k2a/2.0);
            k4a = h * Fa(t + h, ya +k3a,f,i);
            k4b = h * (ya +k3a);
    ya += (k1a + 2.0 * k2a + 2.0 * k3a + k4a) / 6.0;
yb += (k1b + 2.0 * k2b + 2.0 * k3b + k4b) / 6.0;

*pty0 = ya;

    *pty1 = yb;

}
// Funcao V'=(mg-A +f+ b|v|*v)/m
// X'=V

double Fa(double t, double v ,double f,double i)
{
    double Y,M,g,A,b;
    M= 400.0;
    g=9.81;
    A=M*g+5.0;
    b= 200.0;
    Y = (M*g - A+ f- b*fabs(v)*v+ i)/M;

    return Y;
}

```

```

double adaline(double f,double t, double x1,double x0,double v0,double *pw0,double *pw1,double
*pw2,double *pw3,double *pw4,double *pw5)
{

    double eta,e,w0,w1,w2,w3,w4,w5;

    eta=0.9;

    w0=*pw0;
    w1=*pw1;
    w2=*pw2;
    w3=*pw3;
    w4=*pw4;
    w5=*pw5;
// substituir por função de aquisição de dados.
    //calculo da aproximação por uma reta através de adaline
    e = x1 -( x0 + t*v0 + (t*t/2.0)*(w0 + w1*fabs(v0)*v0+ w2*f)+(t*t*t/6.0)*(w3*fabs(v0) +
w4*v0*v0*v0+ w5*f*fabs(v0) ) );

    w0 =w0 - eta*e*(t*t/2.0);
    w1 =w1 - eta*e*fabs(v0)*v0*(t*t/2.0);
    w2 =w2 - eta*e*f*(t*t/2.0);
    w3 =w3 - eta*e*fabs(v0)*(t*t*t/6.0);
    w4 =w4 - eta*e*v0*v0*v0*(t*t*t/6.0);
    w5 =w5 - eta*e*fabs(v0)*f*(t*t*t/6.0);

    *pw0 = w0;
    *pw1 = w1;
    *pw2 = w2;
    *pw3 = w3;
    *pw4 = w4;
    *pw5 = w5;
}

```

```

void inverter(int o,double C[][6],double I[][6])
{
double p;
//inverção da matriz(at*A)

for(int i=o-1 ; i>-1 ; i--)
{
for(int j=0; j< o ; j++)
{
if((i<j)||(i>j))
{
p=C[i][j]/C[j][j];
for(int k=0 ; k<o ; k++)
{
C[i][k]= C[i][k]- (p*C[j][k]);
I[i][k]= I[i][k]- (p*I[j][k]);
}

}

if (i==j)
{
p=C[i][i];
for(int k=0 ; k<o ; k++)
{
C[i][k]=C[i][k]/p;
I[i][k]=I[i][k]/p;
}
}
}
}
}

```