

INTEGRAÇÃO NEURAL UTILIZANDO DERIVADAS MÉDIAS EM ESTRUTURAS DE CONTROLE PREDITIVO PARA O CONTROLE DO PÊNDULO INVERTIDO NÃO LINEAR

Rahyan Azin Gondim Paiva

Instituto Tecnológico de Aeronáutica
Bolsista PIBIC-CNPq
rahyan.azin@gmail.com

Paulo Marcelo Tasinaffo

Instituto Tecnológico de Aeronáutica
tasinaffo@ita.br

Resumo. A inteligência artificial permitiu grandes avanços na área da computação. Uma das abordagens com maior sucesso são as redes neurais artificiais, que permitem, por exemplo, o reconhecimento da escrita, processamento de imagens, modelagem de sistemas dinâmicos não lineares, aplicações em teoria de controle e etc. Nesse trabalho pretende-se utilizar as redes neurais com arquitetura MLP (Multilayer Perceptron) numa estrutura de controle preditivo para resolver o problema de equilíbrio do pêndulo invertido não linear. A modelagem da dinâmica não linear será realizada utilizando-se o método das derivadas média

Palavras chave: Redes Neurais Artificiais, Modelagem de Sistemas Dinâmicos Não Lineares, Método das Derivadas Médias, Controle Preditivo, Pêndulo Invertido Não Linear.

1. Introdução

Uma boa introdução à teoria das redes neurais artificiais pode ser encontrada em Zurada, 1992, em Haykin, 1999 ou em Russell e Norvig, 2003. Um ponto de partida bastante importante no estudo das redes neurais artificiais é que elas são consideradas aproximadores universais de funções (e.g., Hornik et al, 1989; Cotter, 1990; Jang et al, 1997; Spooner et al, 2002). Assim, as redes neurais artificiais (RNA) têm sido muito utilizadas na modelagem de sistemas dinâmicos não lineares nas últimas décadas, pois elas possuem uma alta capacidade de aproximar mapeamentos não lineares. Vários estudos foram desenvolvidos nessa área utilizando a metodologia NARMAX (*Non Linear AutoRegressive Moving Average with eXogenous inputs*) e posterior aplicação em controle (e.g., Narendra and Parthasarathy, 1990; Hunt et al., 1992; Narendra, 1996). Além desta, outras metodologias também foram desenvolvidas nos últimos anos para representação de sistemas dinâmicos não lineares, entre elas: a metodologia das derivadas instantâneas (e.g., Wang e Lin, 1998; Rios Neto, 2001; Melo, 2008; Melo e Tasinaffo, 2010) e a metodologia das derivadas médias (Tasinaffo, 2003; Tasinaffo e Rios Neto, 2005 e 2006).

Embora existam essas três metodologias para representação de sistemas dinâmicos não lineares, através de redes neurais artificiais com arquitetura *feedforward*, esse trabalho se concentrará somente na metodologia das derivadas médias. A aplicação original do método das derivadas médias em uma estrutura de controle preditivo pode ser encontrada em Tasinaffo e Rios Neto, 2006. Portanto nesse trabalho iremos utilizar as redes neurais com arquitetura MLP (Multiplayer Perceptron) numa estrutura de controle preditivo para resolver o problema de equilíbrio do pêndulo invertido não linear utilizando o método das derivadas médias para modelagem da dinâmica não linear.

2. Descrição do Problema

O objetivo desse trabalho será resolver o problema de equilíbrio do pêndulo invertido não linear utilizando as redes neurais com arquitetura MLP. Será utilizado o método das derivadas médias para realizar a modelagem dinâmica não linear em uma estrutura de controle preditivo.

3. Representação de Sistemas Dinâmicos Através de Redes Neurais Artificiais

Um sistema é dinâmico quando possui memória, ou seja, quando seu estado atual depende de condições passadas. As influências externas ao sistema e que não são afetadas por ele, são chamadas de *entradas* do sistema. O resultado da ação das influências externas e internas são descritos por indicadores chamados de *saídas* do sistema.

Para compreender os relacionamentos do tipo causa-efeito, é necessário então definir o sistema, suas entradas e saídas e realizar um processo de análise formal. É estabelecido um modelo que representa adequadamente as iterações de sinais e os relacionamentos de causa e efeito para o sistema a fim de que o processo de análise possa ser efetuado. Na

modelagem teórica, um modelo é obtido através da aplicação de leis conhecidas que regem os eventos essenciais. O modelo matemático que resulta disso é um conjunto de equações diferenciais ordinárias relacionando as variáveis representativas do sistema.

3.1 Redes Neurais Artificiais

Redes neurais artificiais são sistemas computacionais baseados em ligações onde são usados nós simples ou neurônios para formar uma rede de nós. Essa técnica é inspirada em estruturas do cérebro como os neurônios. Redes Neurais Artificiais (RNA), no campo da inteligência artificial, são sistemas não lineares que imitam o mecanismo de processamento do cérebro humano. As redes neurais possuem nós ou unidades de processamento, estes nós estão conectados através de pesos. Cada unidade possui links para outras unidades, no qual recebem e enviam sinais. Cada unidade pode possuir memória local. Estas unidades são a simulação dos neurônios, recebendo e retransmitindo informações.

Uma rede neural pode possuir múltiplas camadas. Quanto maior o número de camadas, melhor a capacidade de aprendizado. O número de camadas define a capacidade de representação das relações entre o espaço de entrada e de saída. O modelo *perceptron* de múltipla camada (MLP) é caracterizado pela existência de uma camada intermediária, além das camadas de saída e entrada. Se houver apenas uma camada intermediária, a MLP pode aproximar com qualquer grau de aproximação qualquer função contínua (Hornik et. al, 1989). Duas ou mais camadas possibilitam a aproximação de qualquer função contínua ou não.

As aplicações de redes neurais artificiais se baseiam principalmente em sua característica de aprendizado. Atualmente, as aplicações são bastante abrangentes como utilização para *data mining*, OCR (*Optical Character Recognition*), mercado financeiro, sistemas de controle, previsões, reconhecimentos de voz e imagem entre outras.

As redes neurais podem ser classificadas como *feedforward* e *feedback*. O primeiro caso consiste de entradas e saídas apresentadas para treinamento e problemas de instabilidade não ocorrem. As entradas de uma rede deste tipo são conectadas diretamente às saídas para efetuar o processamento em cascata e o treinamento é caracterizado por ser supervisionado. O segundo caso é caracterizado por possuir pelo menos um laço de realimentação. As redes *feedback* são caracterizadas por possuírem um aprendizado não-supervisionado e apresentam problemas de instabilidade.

Para redes *feedforward* multicamadas os nós da camada de entrada servem de entrada para a primeira camada interna e assim por diante até a obtenção da saída. Ainda, pode-se dizer que uma RNA cuja camada interna contém neurônios que efetuam todas as ligações com a camada anterior é chamada de rede completamente conectada.

Existem vários algoritmos de treinamento para uma RNA *feedforward* com arquitetura MLP como o método do Gradiente e o filtro de Kalman. O método do Gradiente é um método determinístico que se baseia na retro-propagação do erro de camadas subsequentes. Cada vez que o padrão atinge uma camada de saída o erro quadrático é computado e comparado a um erro arbitrado, o erro obtido é utilizado para alterar os pesos sinápticos que ligam as camadas de saída às anteriores. O método do filtro de Kalman é um método estocástico e sua operação se dá pela minimização do erro quadrático médio e parte-se do pressuposto que o sistema deva ser pelo menos linearizável em torno de um ponto nominal. O ruído do sistema é do tipo gaussiano de média nula e a variância constante. Devido às características de retro-propagação é possível o processamento paralelo dos neurônios. O processamento paralelo permite o uso da retro-propagação e da filtragem de Kalman em nível neural, portanto o processamento e aprendizado da rede podem ser distribuídos paralelamente, diminuindo, assim, a memória utilizada para armazenar a matriz de covariância dos parâmetros.

3.2 Estruturas de Integração Numéricas

Nem todos os sistemas de equações diferenciais possuem soluções analíticas, principalmente se tratando de equações do tipo não-linear, neste caso pode-se obter somente soluções numéricas. Integradores numéricos são utilizados para resolver tais sistemas diferenciais. São conhecidos dois métodos principais para resolver um sistema de equações diferenciais de primeira ordem ordinárias, o método de passo simples e o método de múltiplos passos. O método de passo simples utiliza e computa o valor de apenas um passo h para propagação da solução. Temos como exemplo desse método os integradores do tipo Runge-Kutta de mais alta ordem. Já o método dos múltiplos passos usam os valores da função e suas derivadas para vários pontos anteriores. Alguns exemplos são Adams-Bashforth, Adam-Moulton e Preditivo-Corretor.

3.3 Representação de Sistemas Dinâmicos

Serão abordadas algumas representações para sistemas dinâmicos com metodologias neurais. São elas: metodologia NARMAX, metodologia das derivadas instantâneas e metodologia das derivadas médias. Como a metodologia das derivadas médias será a utilizada no projeto, sua abordagem será mais aprofundada.

Metodologia NARMAX: Essa metodologia se baseia na utilização de entradas atrasadas, a solução é propagada por meio de uma técnica que usa informações de estados e tempos anteriores. O passo de integração é fixo e a rede aprende e modela um discretizador da dinâmica do sistema.

Metodologia das Derivadas Instantâneas: Esse método utiliza para treinamento da rede a função de derivadas instantâneas. A obtenção dessa função foi proposta por Wang e Lin (1998) e o método se resume na construção destas redes de acordo com o método de aproximação de Runge-Kutta que estima precisamente a mudança nas derivadas instantâneas dos estados do sistema.

Metodologia das Derivadas Médias: Essa metodologia utiliza uma rede *feedforward* em conjunto com um integrador numérico de primeira ordem do tipo Euler, obtendo precisão semelhante àquela obtida por integradores de mais alta ordem. Nesse método ao invés de utilizar as derivadas instantâneas são inseridas internamente, na estrutura de integração do tipo Euler, as funções de derivadas médias. As principais diferenças que surgem são que no método das derivadas médias o passo de integração é fixo e a precisão da modelagem é garantida através da aproximação com que a rede neural é treinada, já o método das derivadas instantâneas o passo de integração pode ser variado para auxiliar no ajuste do erro de propagação do sistema dinâmico em cada passo. No método das derivadas médias consegue-se ter uma elevada precisão com o uso de um método de integração de primeira ordem de Euler, e o método das derivadas instantâneas só consegue ter uma boa precisão se a rede neural for utilizada dentro de um integrador de alta ordem. Essa última diferença traz um fato muito importante de que o método das derivadas médias possuirá expressões analíticas para a retro propagação bem mais simples do que o método das derivadas instantâneas. O algoritmo para obter o integrador neural baseado em derivadas médias, através de uma aprendizagem supervisionada, segue abaixo.

1) Dados os domínios finitos de interesse $[y_j^{\min}(t_0), y_j^{\max}(t_0)]^{n_1}$ em t_0 para $j=1, 2, \dots, n_1$ das variáveis de estado e os domínios também finitos de interesse $[u_j^{\min}, u_j^{\max}]^{n_2}$ para $j=1, 2, \dots, n_2$ das variáveis de controle geram-se m vetores aleatórios, segundo uma distribuição uniforme, dentro destes intervalos na forma:

$$p_i = [y_1^i(t_0) \ y_2^i(t_0) \ \dots \ y_{n_1}^i(t_0); u_1^i(t_0) u_2^i(t_0) \dots u_{n_2}^i(t_0)]^T \quad (1.a)$$

e

$$P = [p_1 : p_2 : \dots : p_m]_{(n_1+n_2) \times m} \quad (1.b)$$

que serão os vetores de entrada ou padrões de treinamento da entrada da rede *feedforward* no instante t_0 .

2) Utilizando-se um integrador de alta ordem propagam-se todas as condições iniciais p_i para $i=1, 2, \dots, m$ com um passo de integração *fixo* de Δt para todas as condições iniciais p_i . Desta forma, geram-se os seguintes vetores de estado no instante $t_0 + \Delta t$:

$$p_i^{\Delta t} = [y_1^i(t_0 + \Delta t) \ y_2^i(t_0 + \Delta t) \ \dots \ y_{n_1}^i(t_0 + \Delta t)]^T \quad (2.a)$$

e

$$P^{\Delta t} = [p_1^{\Delta t} : p_2^{\Delta t} : \dots : p_m^{\Delta t}]_{n_1 \times m} \quad (2.b)$$

3) Determinam-se os vetores de saída T_i ou padrões de treinamento de saída da rede neural na forma:

$$\begin{aligned} T_i &= \frac{1}{\Delta t} \cdot [y_1^i(t_0 + \Delta t) - y_1^i(t_0) \ y_2^i(t_0 + \Delta t) - y_2^i(t_0) \ \dots \ y_{n_1}^i(t_0 + \Delta t) - y_{n_1}^i(t_0)]^T \\ &= [\tan_{\Delta t}^k \alpha_1^i \ \tan_{\Delta t}^k \alpha_2^i \ \dots \ \tan_{\Delta t}^k \alpha_{n_1}^i]^T = (\tan_{\Delta t}^k \alpha^i)^T \end{aligned} \quad (3.a)$$

e

$$T = [T_1 : T_2 : \dots : T_m]_{n_1 \times m} \quad (3.b)$$

Como a função $\tan_{\Delta t}^k \alpha^i$ também é autônoma, então basta apenas uma propagação de Δt sobre todas as condições iniciais dos vetores de entrada p_i para $i=1, 2, \dots, m$ para montar os padrões de treinamento P e T da rede neural.

- 4) Tendo-se os vetores de entrada P e os vetores de saída T da rede *feedforward* ela poderá, então, ser treinada através de um aprendizado supervisionado utilizando-se os algoritmos convencionais de retro-propagação.
- 5) Após o treinamento da rede ter sido concluído com um erro quadrático médio (meq) desejado, então poder-se-á simular a dinâmica a partir da seguinte relação:

$$^{k+1}y^i = \tan_{\Delta t}^k \alpha^i \cdot \Delta t + ^k y^i \quad (4)$$

4. Estruturas de Controle Preditivo Neural

O controle de sistemas dinâmicos não-lineares utilizando-se de redes neurais com arquitetura *feedforward* foi proposto no final da década de 80 (Hunt et al., 1992). Quando se trata de sistemas lineares, várias técnicas de projetos de sistemas de controles são conhecidas, mas quando se trata de sistemas não-lineares são necessárias várias conjecturas de modo a ter-se um embasamento teórico consistente. Inicialmente as redes neurais foram trabalhadas empiricamente e consideradas como sendo um método experimental para resolução de sistemas dinâmicos não-lineares. Nas últimas décadas têm surgido avanços em relação aos resultados teóricos, porém os modelos matemáticos envolvidos partem do princípio do conhecimento fenomenológico do sistema a ser modelado. Portanto, quando o problema envolve sistemas não-lineares, sistemas com complexidade elevada a busca de soluções caminha em direção de sistemas heurísticos, em decisões baseadas em lógicas nebulosa, em redes neurais, etc. Estes algoritmos são baseados em dados pré-existentes, onde a memória histórica é a principal fonte de informação.

O controle preditivo é uma abordagem adequada para utilizar redes neurais (Phan e Xing, 1995). As etapas para a utilização constituem da identificação do sistema dinâmico, que corresponde ao treinamento de um modelo e da determinação da lei de controle. O esquema de controle preditivo neural apresentado no que se segue, foi proposto e demonstrado por Rios Neto (2000). Em um problema de controle preditivo neural de um sistema dinâmico (Mills et al, 1994), ele adota uma abordagem heurística e teórica para resolver o problema de minimizar um objeto funcional quadrático para a restrição de uma rede neural preditiva, representando a dinâmica do sistema a ser controlado. No esquema proposto, os problemas de treinamento da rede neural e de determinar as ações de controle preditivo são vistas e tratadas de forma integrada, como estimativa linear ótima estocástica de parâmetros. O problema a ser resolvido é o de controlar um sistema dinâmico modelado por uma EDO:

$$\dot{y} = f(y) \quad (5)$$

Supõe-se que o sistema a ser controlado pode ser aproximado por um modelo discreto. Ou seja, por $t_j = t + j \cdot \Delta t$:

$$y(t_j) \cong f[y(t_{j-1}), \dots, y(t_{j-n_y}); u(t_{j-1}), \dots, u(t_{j-n_u})] \quad (6)$$

onde, $y(t), \dots, y(t_{1-n_y})$ e $u(t_{j-1}), \dots, u(t_{1-n_u})$ são as respostas e ações passadas do sistema, respectivamente. No esquema de controle preditivo neural usual, uma rede neural *feedforward* é treinada para aprender um modelo discreto como na Eq. (2). Esse modelo é então usado como um modelo de resposta o sistema interno para conseguir as ações de controle suaves que irão acompanhar uma trajetória resposta de referência pela minimização (e.g., Clarke et al, 1987a; Clarke et al, 1987b; Liu et al, 1998) do horizonte finito funcional:

$$J = \left[\sum_{j=1}^{n_h} y_r(t_j) - \hat{y}(t_j) \right]^T r_y^{-1}(t) [y_r(t_j) - \hat{y}(t_j)] + \sum_{j=0}^{n_h-1} [u(t_j) - u(t_{j-1})]^T r_u^{-1}(t) [u(t_j) - u(t_{j-1})] / 2 \quad (7)$$

onde, $y_r(t_j)$ é a trajetória resposta de referência no instante t_j ; n_h é o número de passos no horizonte finito de otimização; $r_r^{-1}(t_j)$ e $r_u^{-1}(t_j)$ são definidas como matrizes positivas de ponderação; $\hat{y}(t_j)$ é a saída da rede neural *feedforward* previamente treinada para aproximar um modelo discreto do sistema dinâmico de resposta. A determinação das ações do controle preditivo podem ser tratadas como um problema de estimação de parâmetro, se a minimização do funcional da Eq. (7) é vista como o seguinte problema estocástico:

$$y_r(t_j) = \hat{y}(t_j) + v_y(t_j) \quad (8.a)$$

$$0 = u(t_{j-1}) - u(t_{j-2}) + v_u(t_{j-1}) \quad (8.b)$$

$$E[v_y(t_j)] = 0, E[v_y(t_j) \cdot v_y^T(t_j)] = r_y(t_j) \quad (8.c)$$

$$E[v_u(t_j)] = 0, E[v_u(t_j) \cdot v_u^T(t_j)] = r_u(t_j) \quad (8.d)$$

com, $j = 1, 2, \dots, n_h$; onde $\hat{y}(t_j) = f[\hat{y}(t_{j-1}), \dots, \hat{y}(t_{j-n_y})]$; $u(t_{j-1}), \dots, u(t_{j-n_u}), w$ são as saídas da rede neural que é recursivamente usada como preditora das respostas do sistema dinâmico no horizonte de otimização e é entendido que para $t_{j-k} \leq t$, $\hat{y}(t_{j-k})$ são estimativas ou medições de valores de saída já ocorridos, no *loop* de controle de *feedback*; $v_y(t_j)$ e $v_u(t_j)$ são ruídos não correlacionados para diferentes valores de t_j . Para resolver o problema das Eqs de (8.a) até (8.d) é necessária uma abordagem iterativa, onde em cada iteração é feita uma perturbação para conseguir uma aproximação linear da Eq. (8.a):

$$\alpha(i) \cdot [y_r(t_j) - \bar{y}(t_j, i)] = \sum_{k=0}^{j-1} [\partial \hat{y}(t_j) / \partial u(t_k)]_{\{\bar{u}(t_k; i)\}} \cdot [u(t_k, i) - \bar{u}(t_k, i)] + v_y(t_j) \quad (9)$$

onde k começa em zero, mesmo para $j > n_u$, como consequência de $\hat{y}(t_j)$ ser recursivamente uma função de $u(t_{j-2}), \dots, u(t)$ através das sucessivas recursões começando com $\hat{y}(t_{j-1}), \dots, \hat{y}(t_{j-n_y})$; $0 < \alpha(i) \leq 1$ é um parâmetro a ser ajustado para garantir a hipótese de perturbação linear; e as derivadas parciais são calculadas tanto pela diferenciação numérica ou usando a regra da cadeia para resolver a situação de uma função composta incluindo a regra *back propagation* (ver, e.g., Chandran (1994)) na rede neural *feedforward* que aproxima a função derivada do sistema dinâmico. A formulação como um problema de estimação estocástica linear em cada i -ésima iteração está completa se a recursão da Eq. (8.b) é levada em conta:

$$\alpha(i) \cdot [\hat{u}(t_{j-1}) - \bar{u}(t_{j-1}, i)] = [u(t_{j-1}, i) - \bar{u}(t_{j-1}, i)] + \sum_{k=0}^{j-1} v_u(t_k) \quad (10.a)$$

$$l = 0, 1, \dots, n_h - 1; i = 1, 2, \dots, I \quad (10.b)$$

Uma notação mais compacta:

$$U_l(t, i) \equiv u(t_l, i) \quad (11.a)$$

$$\hat{U}_l(t_{j-1}) \equiv \hat{u}(t_{j-1}) \quad (11.b)$$

$$\alpha(i) [\hat{U}(t_{j-1}) - \bar{U}(t, i)] = U(t, i) - \bar{U}(t, i) + V_u(t) \quad (11.c)$$

$$\alpha(i) \bar{Z}^u(t, i) = H^u(t, i) [U(t, i) - \bar{U}(t, i)] + V_y(t) \quad (11.d)$$

onde o significado das variáveis compactas se torna óbvio pela comparação com as Eqs. (9) e (10). Aplicando o algoritmo do filtro de Kalman, a seguinte solução resulta numa iteração simples (Rios Neto, 2000):

$$\begin{aligned} \hat{U}(t, i) &= \bar{U}(t, i) + \alpha(i) \cdot [\hat{U}(t_{j-1}) - \bar{U}(t, i)] + \\ k(t, i) \cdot \alpha(i) \cdot [\bar{Z}^u(t, i) - H^u(t, i) \cdot [\hat{U}(t_{j-1}) - \bar{U}(t, i)]] \end{aligned} \quad (12.a)$$

$$k(t,i) = R_u(t) \cdot H^u{}^T(t,i) \cdot [H^u(t,i) \cdot R_u(t) \cdot H^u{}^T(t,i) + R_y(t)]^{-1} \equiv \quad (12.b)$$

$$[R_u^{-1}(t) + H^u{}^T(t,i) \cdot R_y^{-1}(t) \cdot H^u(t,i)]^{-1} \cdot H^u{}^T(t,i) \cdot R_y^{-1}(t)$$

$$\bar{U}(t,i+1) = \hat{U}(t,i); \alpha(i) \leftarrow \alpha(i+1); \hat{U}(t) = \hat{U}(t,I) \quad (12.c)$$

$$\hat{R}_u(t,I) = [I_u - K(t,I) \cdot H^u(t,I)] \cdot R_u(t) \quad (12.d)$$

onde, $i=1,2,\dots,I$; $R_u(t)$, $R_y(t)$ e $\hat{R}(t,I)$ são as matrizes de covariância de $V_u(t)$, $V_y(t)$ e $(\hat{U}(t,I) - U(t))$, respectivamente; e I_u é uma matriz identidade. Um algoritmo correspondente para esse esquema de controle preditivo em um intervalo de tempo típico seria então o seguinte:

1) O controle $\hat{u}(t_{-1})$ (ver Eq. (11.b)) é a solução estimada do último passo de controle. Na i -ésima iteração o valor estimado de controle aproximado é $\bar{U}(t,i) = \hat{U}(t,i-1)$; $\alpha(i) \leftarrow \alpha(i-1)$; e para $i=1$ estimativas ou extrapolações de estimativas do ultimo passo de controle são usadas.

2) Calcula-se as derivadas parciais $\frac{\partial \hat{y}(t_j)}{\partial u(t_k)}$ da Eq. (9), obtém-se $H^u(t,i)$ e $\bar{Z}^u(t,i)$, da Eq. (11.c).

3) Estimar $\hat{U}(t,i)$ com o filtro de Kalman das Eqs. (12.a) até (12.d). Perceba que o filtro de Kalman pode ser feito tanto em lote ou sequencialmente, pelo processamento recursivo de componente a componente, em sequência. Incrementa-se i , e repete os passos, até o $\hat{y}(t_j)$ estar suficientemente perto $y_r(t_j)$ de acordo com um erro específico, normalmente $3 \cdot \sigma$ de v_y , e quando isso ocorre, tome:

$$\hat{U}(t) = \hat{U}(t,I) \quad (13)$$

5. Resultados Obtidos

Nesta seção são apresentados os resultados obtidos na resolução do problema de controle do pêndulo invertido. Neste problema, a variável de controle é a força externa aplicada ao pêndulo (haste) de modo a equilibrar o pêndulo em torno da trajetória de referência. Para que a determinação destas variáveis de controle seja possível, deve-se utilizar trajetórias de referência *controláveis* e efetuar o controle de modo que o mesmo não extrapole o intervalo de treinamento. A variável de controle, além de se encontrar dentro da faixa de treinamento, deve possuir função suave e aplicável em uma situação real.

Prossegue-se o estudo de controle preditivo introduzindo-se o sistema não linear do pêndulo invertido, este sistema consiste de uma haste (pêndulo) de massa m e coeficiente de inércia I localizada sobre um corpo de massa M e coeficiente de fricção b . A Figura 01 abaixo ilustra esquematicamente o problema de controle a se considerar.

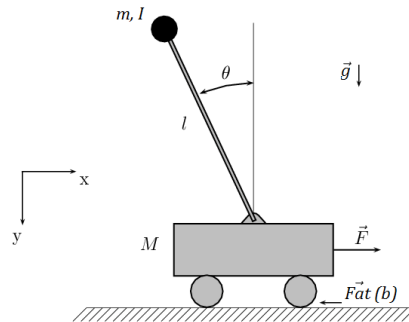


Figura 01 - Esquema físico do pêndulo invertido não linear.

Aplicando as Leis de Newton ao sistema proposto da Figura 01, obtemos o sistema (14) de equações diferenciais ordinárias não linear de segunda ordem dadas por:

$$\begin{cases} (M + m)\ddot{x} + b\dot{x} + ml\ddot{\theta} \cos \theta - ml\dot{\theta}^2 \sin \theta = F \\ (I + ml^2)\ddot{\theta} + mgl \sin \theta = -ml\ddot{x} \cos \theta \end{cases} \quad (14)$$

Com um pouco de manipulação matemática conseguimos transformar esse sistema anterior em um sistema de equações ordinárias não linear de primeira ordem. Esse novo sistema é o que aplicaremos no *software*. Sendo $x = x_1$, $\dot{x} = x_2$, $\theta = x_3$, $\dot{\theta} = x_4$. Sendo, x_1 a posição do centro do carrinho em relação a origem do sistema cartesiano, x_2 a velocidade horizontal do carrinho, x_3 o ângulo que a haste do pêndulo faz com a vertical, x_4 a velocidade angular da haste e u a força externa aplicada ao carrinho. Temos o sistema (15):

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \frac{[u - bx_2 + mlx_4^2 \sin(x_3)](I + ml^2) + (ml)^2 g \sin(x_3) \cos(x_3)}{(I + ml^2)(M + m) - [ml \cos(x_3)]^2} \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = \frac{\{[u - bx_2 + mlx_4^2 \sin(x_3)] \cos(x_3) + (M + m)g \sin(x_3)\}ml}{[ml \cos(x_3)]^2 - (I + ml^2)(M + m)} \end{cases} \quad (15)$$

O sistema (15) é de primeira ordem e pode ser aplicado no *software*. O conjunto de equações acima permite explicitar, para o esquema da Figura 01, a existência de quatro variáveis de estado: x_1 , x_2 , x_3 e x_4 e apenas uma variável de controle u . A variável de controle, no caso deste problema, tem como objetivo fazer com que as variáveis de estado sigam as trajetórias de referência dadas, de modo que seja possível equilibrar o pêndulo sem que haja oscilação angular.

5.1 Treinamento e Simulação da Rede Neural

Treinou-se uma rede neural para que a mesma aprenda a dinâmica do sistema em termos das variáveis de estados representadas pelo sistema (15) e simulou-se a resposta no tempo desta rede à diferentes variações dos parâmetros de entrada, desde que estes se encontrem dentro da faixa de treinamento, para verificação da consistência numérica obtida no treinamento neural. Para este treinamento neural optou-se pela escolha de uma rede do tipo *feedforward* com arquitetura *MLP* e definiu-se como método para modelagem neural o das derivadas médias e o algoritmo de treinamento como sendo o do método do gradiente.

Para o sistema dinâmico representado analiticamente pela sistema (15) foram utilizados os seguintes parâmetros durante a fase de treinamento: massa do carrinho $M = 0.1 \text{ kg}$, massa do pêndulo $m = 4.0 \text{ kg}$, comprimento da haste do pêndulo $l = 5.0 \text{ m}$, momento de inércia do pêndulo $I = 0.05 \text{ kg.m}^2$, coeficiente de fricção $b = 1.0 \text{ N/(}\frac{\text{m}}{\text{s}}\text{)}$, e aceleração da gravidade $g = 10 \text{ m/s}^2$.

Para o treinamento da rede neural utilizou-se os seguintes parâmetros: número de variáveis de estado = 4, número de variáveis de controle = 1, o intervalo de variação da variável de estado x_1 entre $[-5.0, +5.0]$, intervalo de variação da variável de estado x_2 entre $[-4.0, +4.0]$, intervalo de variação da variável de estado x_3 entre $[-1.2, +1.2]$, intervalo de variação da variável de estado x_4 entre $[-5.0, +5.0]$, intervalo de variação da variável de controle u entre $[-5.0, +5.0]$, 1200 padrões de treinamento, 40 neurônios na camada interna, função *tansig* de ativação da camada interna, função *puramente linear* da camada de saída, $(\lambda = 2)$ para as funções de ativação, taxa de aprendizado da rede neural de 0.01, erro máximo esperado de 10^{-7} , épocas de treinamento = 100.000 e porcentagem destinada a padrões de teste de 20%.

Depois de concluído o treinamento, foi obtido o erro máximo MEQ (média dos erros quadráticos) para os padrões de treinamento igual a $3.24 \cdot 10^{-6}$ e para os padrões de teste estimado em $1.34 \cdot 10^{-6}$. Durante todo o treinamento foi utilizado o gráfico em escala *di-log* para acompanhar o desempenho do aprendizado. Para a simulação da rede neural aplicada ao sistema do pêndulo invertido foram adotados os seguintes parâmetros: condição inicial da variável de estado $x_1 = 0.0$, condição inicial da variável de estado $x_2 = 1.0$, condição inicial da variável de estado $x_3 = 0.3$, condição inicial da variável de estado $x_4 = -1.0$, intervalo de variação aleatória da variável de controle u em $[-3.0, 3.0]$, passo de integração = 0.05, intervalo de tempo da simulação dinâmica em $[0, 20]$ segundos e

número de iterações para atualização das condições iniciais = **20**. Os gráficos das Figuras 02 correspondem a simulação neural do sistema dinâmico do pêndulo invertido comparando a função de derivadas médias neural integrada por Euler com a função de derivadas instantâneas original integrada pelo Runge-Kutta 4-5. Analisando-se os gráficos da Figuras 02 nota-se que mesmo para um intervalo de simulação razoavelmente longo, no caso 20 segundos, os gráficos decorrentes do treinamento neural se mostram coerentes aos obtidos pelo método de integração numérica. Assim, pode-se considerar que o aprendizado da rede neural foi efetivo para o sistema (15) de equações diferenciais.

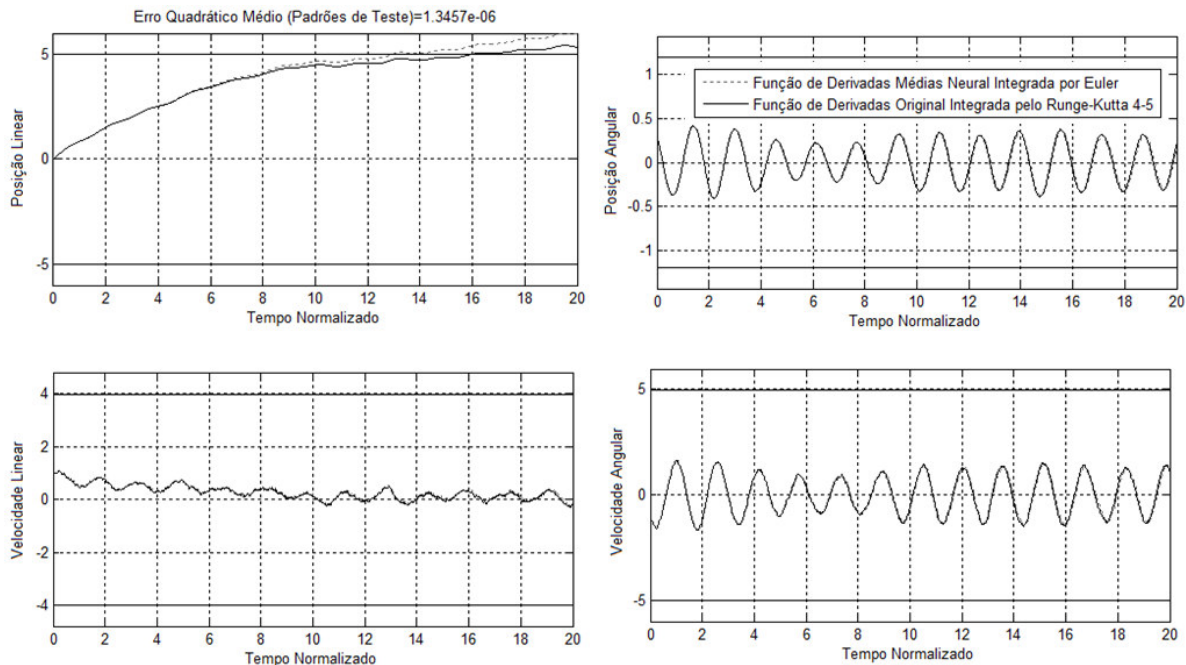


Figura 02 - Simulação neural da dinâmica do pêndulo invertido não linear.

5.2 Controle do Pêndulo Invertido

Depois de realizado o treinamento e simulação da rede neural com a dinâmica do sistema pêndulo invertido não linear foram definidas as trajetórias de referência para que se pudesse realizar o controle preditivo. Optou-se pelo uso de trajetórias de referência, para as variáveis de estados (x_1, x_2, x_3, x_4) na forma $y = 0$. As trajetórias de referências nulas para todas as quatro variáveis de estado se perseguidas pelo sistema de controle garantirão o equilíbrio do pêndulo na vertical. A estrutura de controle preditivo neural resolveu um problema de estimação estocástica utilizando o Filtro de Kalman não linear para rastrear uma trajetória de referência com controle suave.

Encontram-se apresentados a seguir, os parâmetros utilizados para a aplicação do controle preditivo: horizonte de controle = **1**, valor constantes dos controles a priori = **0**, intervalo de estimação entre **[0,500]** segundos, valor da constante que garante a hipótese de linearização = **0.15**, número máximo de iterações do filtro de Kalman de cada previsão = **15**, ponderação da Matriz de Covariância $P_u = \mathbf{1}$, ponderação da Matriz de Covariância $P_y = 10^{-12}$. Os gráficos da Figuras 03 correspondem à aplicação do controle preditivo ao sistema dinâmico não linear do pêndulo invertido. Para as trajetórias percorridas pelas variáveis de estado foi obtida a variável de controle força, que representa a força externa aplicada ao pêndulo (haste). Esta força é representada graficamente na Figura 04. O intervalo do gráfico da Figura 04 foi [30, 33] segundos, pois o gráfico com o intervalo todo de [0,500] segundos iria aparecer apenas um borrão incompreensível. Vale salientar que durante os testes a variável x_1 (posição linear) saiu um pouco do intervalo de confiança utilizado no treinamento e isso talvez tenha comprometido levemente os resultados. Entretanto, os resultados obtidos mostraram-se bastante coerentes com os esperados pela análise física do problema, desde modo pode-se dizer que a rede neural aprendeu de modo satisfatório a dinâmica do sistema de modo que o controle preditivo pudesse ser aplicado.

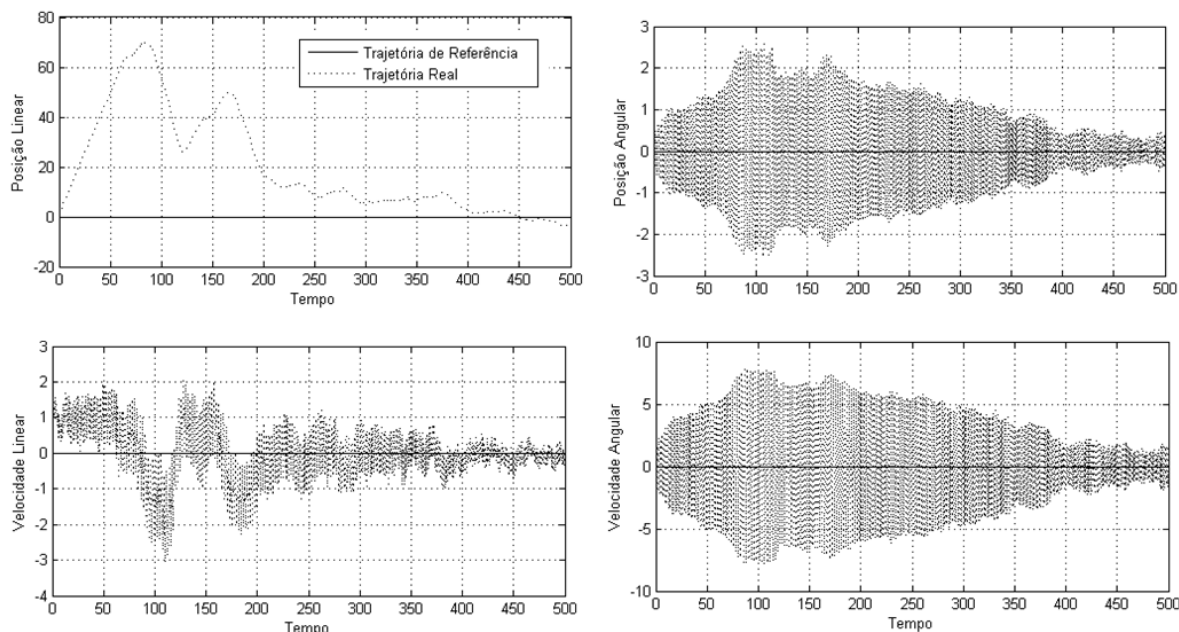


Figura 03 - Trajetórias de referência perseguidas pela estrutura de controle preditivo neural.

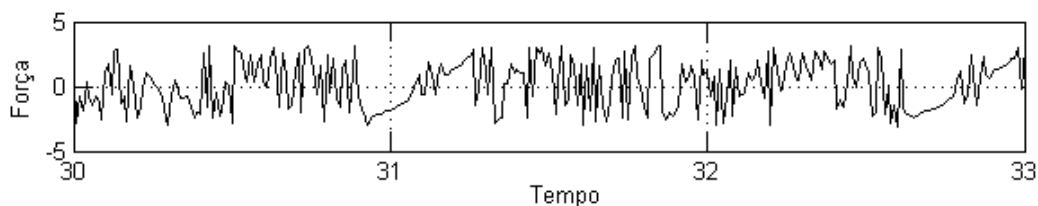


Figura 04 - Força estimada no tempo para equilibrar o pêndulo invertido.

5. Conclusão

O projeto ratificou o método de utilização das derivadas médias. Foram feitas explicações a respeito da representação de sistemas dinâmicos de redes neurais artificiais de um modo superficial, porém bem explicativo, com um foco maior no método das derivadas médias, pois este foi o método efetivamente utilizado no projeto. O treinamento da rede neural do tipo *feedforward* com arquitetura *MLP* foi realizado com sucesso. Também houve êxito na determinação do controle do pêndulo invertido. O resultado do projeto sugere que o procedimento adotado pode ser aplicado numa situação física real, que era a conclusão esperada no início do projeto.

5. Referências

- [1] Cotter, N. E. (1990). The Stone-Weierstrass and its application to neural networks. *IEEE Transactions on Neural Networks*, v. 1, n. 4, pp. 290-295.
- [2] Haykin, S. (1999). *Neural networks: a comprehensive foundation*. 2nd ed., New Jersey: Prentice-Hall, Inc..
- [3] Hornik, K., Stinchcombe, M. and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, v. 2, n. 5, pp. 359-366.
- [4] Hunt, J. K., Sbarbaro, D., Zbikowski, R. and Gawthrop, P. (1992). Neural networks for control system-a survey, *Automatica* **28**(6): 1083-1112.
- [5] Jang, J.-S. R., Sun, C.-T and Mizutani, E. (1997). *Neuro-Fuzzy and soft computing: a computational approach to learning and machine intelligence*. Prentice-Hall, Inc.
- [6] Melo, R. P. (2008). *Metodologia de modelagem empírica utilizando integradores neurais aplicada a sistemas dinâmicos não-lineares*. 136f. Tese de Mestrado. Instituto Tecnológico de Aeronáutica, São José dos Campos.
- [7] Melo, R. P. and Tasinaffo, P. M. (2010). Uma metodologia de modelagem empírica utilizando o integrador neural de múltiplos passos do tipo Adams-Bashforth. *SBA. Sociedade Brasileira de Automática*, v. 21, n. 5, pp. 487-509.

- [8] Mitchell, M. (1998). *An Introduction to genetic algorithms*. First MIT Press paperback edition.
- [9] Narendra, K. S. and Parthasarathy, K. (1990). Identification and control of dynamical systems using neural networks, *IEEE Transactions on Neural Networks* 1: 4–27.
- [10] Narendra, K. S. (1996). Neural networks for control, *Proceedings of the IEEE* 84: 1385–1406.
- [11] Oliveira, D. G.; Tasinaffo, P. M. Controle Preditivo com Rede Neural Artificial Aplicado ao Problema de Equilíbrio do Pêndulo Invertido Não-Linear. Santos, N.18, ano 5.
- [12] Rios Neto, A. (2001). Dynamic systems numerical integrators in neural control schemes. *V Congresso Brasileiro de Redes Neurais*, Rio de Janeiro-RJ, pp. 85–88.
- [13] Russell, S. and Norvig, P. (2003). *Artificial Intelligence*. 2nd ed., New Jersey: Prentice-Hall, Inc.
- [14] Spooner, J. T., Maggiore, M., Ordóñez, R. and Passino, K. M. (2002). *Stable adaptive control and estimation for nonlinear systems neural and fuzzy approximator techniques*. New York: Wiley-Interscience.
- [15] Tasinaffo, P. M. (2003). *Estruturas de integração neural feedforward testadas em problemas de controle preditivo*. 230 p. INPE-10475-TDI/945. Doctoral Thesis, Instituto Nacional de Pesquisas Espaciais -INPE, São José dos Campos, Brasil.
- [16] Tasinaffo, P. M.; Neto, A. R. Modelagem de Sistemas Dinâmicos com Integrador Neural de Euler Baseado em Derivadas Médias.
- [17] Tasinaffo, P. M. and Rios Neto, A. (2005). Mean derivatives based neural euler integrator for nonlinear dynamic systems modeling. *Learning and Nonlinear Models*, v. 3, pp. 98–109.
- [18] Tasinaffo, P. M. and Rios Neto, A (2006). Predictive control with mean derivative based neural euler integrator dynamic model. *SBA. Sociedade Brasileira de Automática*, v. 18, n. 1, pp. 94-105.
- [19] Wang, Y.-J. and Lin, C.-T. (1998). Runge-Kutta neural network for identification of dynamical systems in high accuracy. *IEEE Transactions On Neural Networks*, v. 9, n. 2, p. 294-307.
- [20] Zurada, J. M. (1992). *Introduction to artificial neural system*. St. Paul, MN, USA: West Pub. Co.