



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

INPE-11484-RPQ/778

**INTEGRAÇÃO NEURAL EM UMA ESTRUTURA DE
CONTROLE PREDITIVO**

Paulo Marcelo Tasinaffo
Atair Rios Neto

RESUMO

A aplicação de redes neurais em controle de sistemas dinâmicos sempre implica na necessidade de treinar uma rede, para se obter um modelo interno do sistema. Uma das maneiras de se representar o modelo interno da dinâmica do sistema é projetar a rede neural como um modelo discretizado com entradas atrasadas. A rede treinada desta forma possui a desvantagem de exigir vários neurônios nas camadas internas e de entrada. Neste trabalho uma nova metodologia para representar a dinâmica do sistema na rede *Feedforward* é preliminarmente testada. Nesta metodologia, através da utilização da estrutura de um integrador numérico de múltiplos passos do tipo Adams-Bashforth de quarta ordem pode-se projetar a arquitetura da rede apenas treinando-a para aprender a função de derivadas do modelo de equações diferenciais ordinárias do sistema. Este tipo de abordagem evita uma complexidade desnecessária na arquitetura da rede, e transforma o problema de treinamento de dinâmico para estático. É desenvolvida a aplicação desta nova metodologia numa estrutura de controle preditivo não-linear. Um exemplo prático do problema de transferência de órbita Terra/Marte é considerado para os testes desta nova metodologia.

Palavras-chaves

Redes Neurais, Modelagem de Sistemas Dinâmicos, Controle Preditivo Não-linear, Integrador de Múltiplos-Passos de Adams-Bashforth.

NEURAL INTEGRATION APPLIED TO A PREDICTIVE CONTROL STRUCTURE

Abstract

The use of neural networks in the control of dynamic systems always implies in the need of training the neural net in order to get an internal model of the system. One of the ways of representing the internal model of the dynamics of the system is to design the neural network to learn a system approximation in the form of a discrete model with delayed inputs. The neural net designed and trained in this way has the disadvantage of needing too many neurons in the input and hidden layers. In this work a new approach to represent the dynamics of the system in a feedforward neural network is preliminarily tested. In this approach, using the structure of a multistep Adams-Bashforth ordinary differential equation numerical integrator it is possible to have the neural network designed to only learn the dynamic system derivative function. As a consequence, an unnecessary complexity in the design of the neural network is avoided and it has to only learn an algebraic static function. This new approach is applied to a Nonlinear Predictive Control (NPC) structure. A simple but practical problem of orbit transfer between Earth and Mars is considered for the testing of the scheme.

Keywords

Neural Networks, Dynamic Systems Modeling, Nonlinear Predictive Control Framing Adams-Bashforth Integrator.

SUMÁRIO

1 - INTRODUÇÃO	13
2 - IDENTIFICAÇÃO DE SISTEMAS DINÂMICOS	14
3 - ESTRUTURA DE CONTROLE PREDITIVO NEURAL	16
4 - REPRESENTAÇÃO DE SISTEMAS DINÂMICOS ATRAVÉS DE REDES NEURAIS NA ESTRUTURA DE INTEGRADORES NUMÉRICOS	23
5 - O ALGORITMO COMPUTACIONAL COMBINANDO INTEGRA- DORES NEURAIS E ESTRUTURA DE CONTROLE PREDITIVO	32
6 - RESULTADOS E SIMULAÇÕES	38
7 - COMENTÁRIOS FINAIS E CONCLUSÕES	42
REFERÊNCIAS BIBLIOGRÁFICAS	45
APÊNCIDE A – O CÁLCULO DA RETRO-PROPAGAÇÃO	47
APÊNDICE B - INTEGRADORES NUMÉRICOS	50

LISTA DE FIGURAS

01 - Identificação de sistemas dinâmicos através de Redes Neurais	15
02 - Esquema de otimização neural para a determinação de controle $u(t)$ que rastreará a trajetória de referência $r(t)$	16
03 - Distinção entre a metodologia tradicional (MT) e a nova metodologia (NM)	25
04 - Esquema ilustrativo: vantagens de se representar à função de derivadas através de uma rede neural artificial na estrutura de integrador numérico	27
05 - Maneiras distintas de se treinar uma rede neural na estrutura de um integrador numérico	31
06 - Esquema gráfico das condições iniciais das variáveis de controle no esquema de controle preditivo	33
07 - Esquema ilustrativo da transferência de órbita Terra/Marte	39
08 - Simulação com integrador neural para a dinâmica do problema Terra/Marte	40
09 - Simulação de controle preditivo da transferência Terra/Marte ($\Delta t = 0.05$) ..	40
10 - Simulação de controle preditivo da transferência Terra/Marte ($\Delta t = 0.01$) ..	41
11 - Controle estimado pela estrutura de controle preditivo ($\Delta t = 0.01$)	41
12 - Entradas e saídas de uma camada genérica k em rede <i>Feedforward</i>	47
13 - Esquema Simplificado de uma Estrutura de Integração Neural	49

LISTA DE TABELAS

- 1- Derivadas parciais $\frac{\partial \hat{y}(t_j)}{\partial u(t_k)}$ necessárias para estimar os controles $u(t)$ dentro de um horizonte de controle com $n=1$ e entradas atrasadas da rede neural iguais a $n_y=n_u=4$ 34

1-INTRODUÇÃO.

As propriedades de mapeamento de funções não-lineares são o ponto central para a utilização de redes neurais em controle. Treinar uma rede neural utilizando dados de entrada/saída de uma planta não-linear pode ser considerado como um problema de aproximação de uma função não-linear. Já foi demonstrado que as redes *Feedforward* multicamadas do tipo *Perceptron* podem aproximar arbitrariamente bem uma função contínua (Cybenko,1988, Hornik et al, 1989). Uma rede *Feedforward* com somente uma simples camada interna, onde cada unidade da camada interna é representada por uma função sigmoideal contínua, é suficiente para representar qualquer sistema dinâmico. Modelos de sistemas dinâmicos e suas inversas têm utilidade imediata em controle. Na literatura de controle um número grande de trabalhos bem estabelecidos já existe (Hunt et al, 1992), com destaque para controle preditivo neural, por sua eficiência e desempenho (Mills et al, 1994, Liu et al, 1998).

Neste trabalho será abordada a aplicação de integradores numéricos neurais sobre uma estrutura de controle preditivo (Rios Neto, 2001). Serão analisadas as possibilidades de se utilizar em conjunto os integradores numéricos e as redes neurais artificiais diretamente em estratégias de controle preditivo não-linear. Este tipo de abordagem tem a vantagem de reduzir a dimensão da rede neural, e, portanto, facilitar seu treinamento uma vez que a rede neural só precisará aprender a função de derivadas do sistema dinâmico original (Wang e Lin (1998), Rios Neto (2001)). É importante observar que a implementação, teste e ajuste desta metodologia sobre uma estrutura de integração de múltiplos passos do tipo Adams-Bashforth de quarta ordem é que torna este trabalho original.

Algoritmos de filtragem de Kalman com ou sem processamento paralelo podem ser utilizados não só para treinar as redes *Feedforward* (Chandran (1994), Rios Neto (1997)) para representar a dinâmica de sistemas físicos, mas também para determinar a política de controle suave numa estrutura de controle preditivo em malha fechada (Rios Neto,2000).

Deste modo, inicialmente na seção 2 será vista a representação convencional do modelo direto da planta dinâmica. Na seção 3 será vista a estrutura de controle preditivo neural. Na seção 4 introduzir-se-ão as funções de derivadas neurais sobre os esquemas de integração numérica. Na seção 5 será apresentado um algoritmo computacional combinando integradores neurais e estrutura de controle preditivo cuja solução da estimativa numérica da política de controle é obtida pelo filtro de Kalman. Na seção 6 um resultado numérico obtido a partir do problema não-linear de transferência de órbita Terra/Marte será apresentado. Por fim, na seção 7 serão realizados alguns comentários finais e conclusões. No apêndice A dar-se-á uma pequena introdução ao conceito de retro-propagação e no apêndice B um breve comentário sobre a teoria de integração numérica.

2 - IDENTIFICAÇÃO DE SISTEMAS DINÂMICOS.

É importante identificar o modelo direto da planta do sistema físico a ser estudado. A identificação deste modelo é o ponto chave de partida para se querer utilizar qualquer uma das possíveis estruturas de controle não-linear encontradas na literatura.

O procedimento de treinar uma rede neural para representar a dinâmica de um sistema (Chen e Billings, 1992) será referido aqui como modelagem direta. Uma estrutura neural (Narendra e Parthasarathy, 1990) para alcançar isso é mostrada na figura 1. Como pode ser visto, o modelo da rede neural é colocado em paralelo com o sistema e o erro entre o sistema e as saídas da rede é utilizado como o sinal de treinamento da rede. Esta estrutura de aprendizagem é um clássico problema de aprendizado supervisionado (Zurada, 1992) onde a necessidade de um professor (padrões de treinamento) é imprescindível.

Uma questão no contexto de controle bastante importante é a natureza dinâmica dos sistemas a serem estudados. É assumido que o sistema é governado (Norgaard et al, 2000) pela seguinte equação de diferenças discretas no tempo não-linear.

$$y^P(t+1) = f[y^P(t), \dots, y^P(t-n+1); u(t), \dots, u(t-m+1)] \quad (1)$$

Desta forma, a saída do sistema y^P no tempo $t+1$ depende dos n valores passados e dos m valores de entrada u . Uma abordagem óbvia para a modelagem do sistema é escolher as estruturas de entrada/saída da rede neural como sendo as mesmas daquela do sistema. Denotando a saída da rede como y^m tem-se então,

$$y^m(t+1) = \hat{f}[y^P(t), \dots, y^P(t-n+1); u(t), \dots, u(t-m+1), w] \quad (2)$$

Aqui, \hat{f} representa o mapeamento não-linear de entrada/saída da rede. Note que a entrada para a rede inclui os valores passados do sistema de saída real. Esta dependência sobre o sistema de saída não está incluída explicitamente no esquema da figura 1.

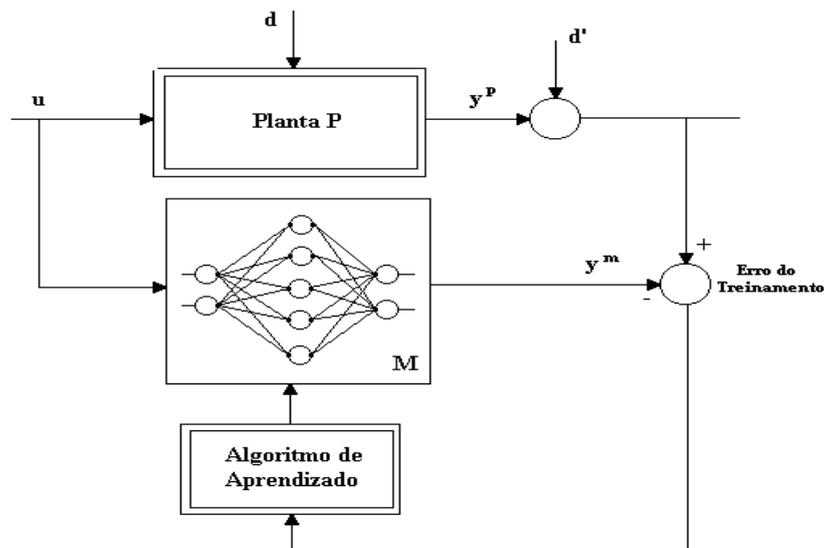


FIGURA 1 – Identificação de sistemas dinâmicos através de redes neurais.

Se é assumido que após um período de treinamento a rede fornece uma boa representação da planta (isto é $y^m \cong y^P$) então, a rede poderá ser utilizada independentemente da planta. Tal modelo neural poderá ser descrito por:

$$y^m(t+1) = \hat{f}[y^m(t), \dots, y^m(t-n+1); u(t); \dots, u(t-m+1), w] \quad (3)$$

3 - ESTRUTURA DE CONTROLE PREDITIVO NEURAL.

Nesta seção, adotando-se uma abordagem heurística e teórica, apresentar-se-á o projeto e análise de convergência de um método de controle preditivo neural. Este método consiste em resolver um problema de otimização de um índice quadrático de desempenho, cujo vínculo é a rede já treinada com a dinâmica do sistema desejado (Mills et al, 1994). Por se tratar de um problema de otimização não-linear, a solução é obtida iterativamente para as ações discretas de controle através de sucessivas linearizações. Para o método se tornar viável na prática a estimação dos controles deverá ser obtida necessariamente em tempo real. Com a evolução dos processadores atuais isto não constitui nenhum problema que impeça sua utilização.

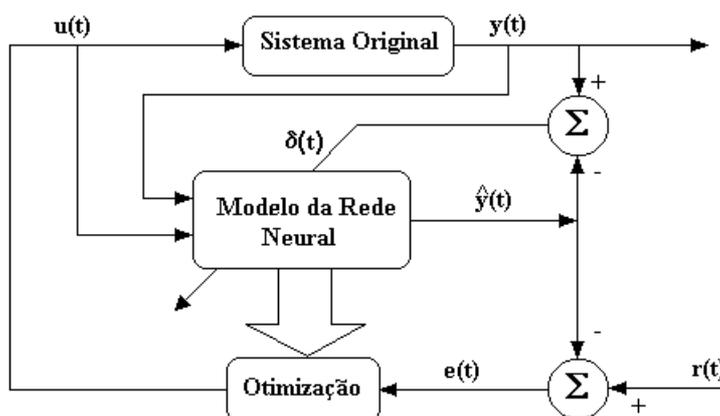


FIGURA 2 – Esquema de otimização neural para a determinação de controle $u(t)$ que rastreará a trajetória de referência $r(t)$.

Pode ser demonstrado (Rios Neto, 2000) que os algoritmos de filtragem de Kalman fornecem soluções que convergem para soluções suaves para as variáveis de controle e que rastreiam a trajetória de referência.

Ao contrário de uma estrutura de controle do tipo “Internal Model Control” (IMC), num esquema de controle preditivo não é necessário empregar a dinâmica do *modelo inverso* da planta, e desta forma, evita-se um treinamento neural. Por outro lado, no

controle preditivo neural deve-se a todo instante resolver um problema de otimização envolvendo o modelo neural da dinâmica da planta.

A figura 2 (Rios Neto, 2000) apresenta um esquema simplificado da estrutura de controle preditivo. Como pode ser visto, uma rede neural é colocada em paralelo com a planta na tentativa de aprendê-la. Quando o aprendizado alcançado estiver dentro de um erro ou tolerância aceitáveis, então a determinação dos controles suaves poderá ser obtida pelo algoritmo da filtragem de Kalman como a solução de um problema de otimização vinculado a rede e que rastreará também a trajetória de referência $\mathbf{r}(\mathbf{t})$.

No esquema da figura 2, os problemas associados ao treinamento da rede neural *Feedforward* e da determinação da política de controle suave são ambos vistos e tratados de uma maneira integrada como problemas de estimação linear estocástica de parâmetros. O tipo de abordagem elaborada aqui permite ver o problema de controle ótimo em uma estrutura estocástica mais geral e derivar versões de algoritmos de controle com processamento paralelo ou não (Rios Neto, 2000) que são formalmente equivalentes às versões do filtro de Kalman derivadas e utilizadas para o problema de treinamento da rede neural *Feedforward* (Rios Neto, 1997). O problema que se deseja resolver é o de *controlar* o sistema dinâmico dado por,

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \quad (4)$$

Onde um modelo de entrada/saída não-linear discretizado no tempo é utilizado para prever respostas aproximadas do modelo dado em (4):

$$\hat{\mathbf{y}}(\mathbf{t}_j) = \mathbf{f}[\mathbf{y}(\mathbf{t}_{j-1}), \dots, \mathbf{y}(\mathbf{t}_{j-n_y}); \mathbf{u}(\mathbf{t}_{j-1}), \dots, \mathbf{u}(\mathbf{t}_{j-n_u}), \mathbf{w}] \quad (5.a)$$

Onde,

$$\mathbf{t}_j = \mathbf{t} + \mathbf{j} \cdot \Delta \mathbf{t} \quad (5.b)$$

O esquema de controle preditivo neural utiliza-se de uma rede *Feedforward* que possui a habilidade de aprender com precisão desejada um mapeamento como aquele representado por (5.a) para modelar o sistema dinâmico da equação (4). O modelo

interno representado pela rede será então o modelo de resposta que poderá ser utilizado para determinar as ações de controle suaves que rastrearão a trajetória de referência, por minimizar um índice de performance preditivo quadrático (Hunt et al, 1992) ou (Norgaard, 2000) em uma estrutura de controle preditivo. O índice de performance ou funcional desta estrutura de controle é dado por:

$$\mathbf{J}(\mathbf{t}) = \left[\sum_{j=1}^n [\mathbf{y}_r(\mathbf{t}_j) - \hat{\mathbf{y}}(\mathbf{t}_j)]^T \cdot \mathbf{r}_y^{-1}(\mathbf{t}) \cdot [\mathbf{y}_r(\mathbf{t}_j) - \hat{\mathbf{y}}(\mathbf{t}_j)] + \sum_{j=0}^{n-1} [\mathbf{u}(\mathbf{t}_j) - \mathbf{u}(\mathbf{t}_{j-1})]^T \cdot \mathbf{r}_u^{-1}(\mathbf{t}) \cdot [\mathbf{u}(\mathbf{t}_j) - \mathbf{u}(\mathbf{t}_{j-1})] \right] / 2 \quad (6)$$

Onde,

$\mathbf{y}_r(\mathbf{t}_j)$... trajetória de referência no instante \mathbf{t}_j ;

n ... horizonte em que as ações de controle e trajetórias de referência são consideradas;

$\mathbf{r}_r^{-1}(\mathbf{t}_j)$ e $\mathbf{r}_u^{-1}(\mathbf{t}_j)$... matrizes de pesos, definidas positivas;

$\hat{\mathbf{y}}(\mathbf{t}_j)$... saída da rede *Feedforward* treinada para aproximar o modelo do sistema dinâmico.

Para maiores detalhes sobre a construção do funcional da equação (6) ver Clarke et al (1987a,b). A saída da rede *Feedforward* $\hat{\mathbf{y}}(\mathbf{t}_j)$ é representada pela expressão (5.a). Os parâmetros ou pesos \mathbf{w} desta rede já devem ter passados por um treinamento que produza uma saída na rede já com um erro dentro de uma tolerância aceitável. O primeiro termo do funcional da equação (6) está associado ao rastreo da trajetória de referência e o segundo termo na determinação de uma política de controle suave. Quando este funcional é minimizado espera-se que estas duas condições sejam satisfeitas simultaneamente. Note-se que no esquema de controle preditivo, em um ciclo típico, quando este funcional $\mathbf{J}(\mathbf{t})$ é otimizado, apenas o controle determinado em

t é implementado; e assim por diante, repetindo-se o ciclo para cada instante discreto posterior de tempo.

O problema de determinação das ações de controle preditivo pode ser tratado também como uma estimação linear ótima de parâmetros permitindo assim, a derivação e utilização de um algoritmo do tipo Filtro de Kalman. Este método estocástico assume inicialmente que o problema de determinação do controle sobre o funcional da equação (6) pode ser visto como o seguinte problema de estimação de parâmetros estocásticos (Rios Neto, 2000):

$$\mathbf{y}_r(t_j) = \hat{\mathbf{y}}(t_j) + \mathbf{v}_y(t_j) \quad (7.a)$$

$$\mathbf{0} = \mathbf{u}(t_{j-1}) - \mathbf{u}(t_{j-2}) + \mathbf{v}_u(t_{j-1}) \quad (7.b)$$

$$\mathbf{E}[\mathbf{v}_y(t_j)] = \mathbf{0}, \mathbf{E}[\mathbf{v}_y(t_j) \cdot \mathbf{v}_y^T(t_j)] = \mathbf{r}_y(t_j) \quad (7.c)$$

$$\mathbf{E}[\mathbf{v}_u(t_j)] = \mathbf{0}, \mathbf{E}[\mathbf{v}_u(t_j) \cdot \mathbf{v}_u^T(t_j)] = \mathbf{r}_u(t_j) \quad (7.d)$$

Para,

$$\mathbf{j} = 1, 2, \dots, \mathbf{n} \quad (7.e)$$

Onde,

$\hat{\mathbf{y}}(t_j) = \mathbf{f}[\mathbf{y}(t_{j-1}), \dots, \mathbf{y}(t_{j-n_y}); \mathbf{u}(t_{j-1}), \dots, \mathbf{u}(t_{j-n_u}), \mathbf{w}]$ é a saída da rede neural;

$\hat{\mathbf{y}}(t), \dots, \hat{\mathbf{y}}(t_{1-n_y})$ e $\mathbf{u}(t_{-1}), \dots, \mathbf{u}(t_{1-n_u})$ são, respectivamente, as respostas do sistema e as ações de controle já ocorridas e conhecidas;

$\mathbf{v}_y(t_j)$ e $\mathbf{v}_u(t_j)$ são as componentes não-correlacionadas de ruído para diferentes valores de t_j .

A equação (7.a) estabelece que a trajetória de referência do estado do sistema $\mathbf{y}_r(\mathbf{t}_j)$ no instante futuro é igual a estimação realizada pela *rede* em relação aos instantes atrasados mais uma incerteza $\mathbf{v}_y(\mathbf{t}_j)$. A equação (7.b) traduz a característica suave das ações de controle, ou seja, duas atuações sucessivas devem ser estimadas de tal forma que a diferença entre elas esteja o mais próximo possível da média nula do erro $\mathbf{v}_u(\mathbf{t}_{j-1})$.

A equação (7.b) é uma relação recursiva que pode ser expressa na forma de uma informação a priori como segue:

$$\hat{\mathbf{u}}(\mathbf{t}_{-1}) = \mathbf{u}(\mathbf{t}_{j-1}) + \sum_{k=0}^{j-1} \mathbf{v}_u(\mathbf{t}_k) \quad (8)$$

Onde o valor a priori $\hat{\mathbf{u}}(\mathbf{t}_{-1})$ é o valor estimado de um controle no instante $\mathbf{t}_{-1} = \mathbf{t} - \Delta\mathbf{t}$. Desta forma, a minimização do funcional dado pela equação (6) é modelada como satisfazendo as observações da equação (7.a) sujeita a informação a priori da equação (8). A primeira consequência do tratamento deste problema numa estrutura estocástica mais geral é que as matrizes de pesos presentes na função objetivo da equação (6), ou seja, $\mathbf{r}_y^{-1}(\mathbf{t}_j)$ e $\mathbf{r}_u^{-1}(\mathbf{t}_j)$ têm agora o significado de matrizes de covariância associadas as variáveis aleatórias cujos os desvios padrões modelam, respectivamente, a precisão no rastreamento da trajetória de referência e a dispersão do incremento do controle suave. Isto facilita bastante o entendimento de suas definições.

Para tratar reiterativamente o problema representado pelas equações (7.a) e (7.b) como um de estimação linear de parâmetros é necessário tomar uma aproximação linearizada da equação (7.a) como segue:

$$\alpha(\mathbf{i}) \cdot [\mathbf{y}_r(\mathbf{t}_j) - \bar{\mathbf{y}}(\mathbf{t}_j, \mathbf{i})] = \sum_{k=0}^{j-1} [\partial \hat{\mathbf{y}}(\mathbf{t}_j) / \partial \mathbf{u}(\mathbf{t}_k)]_{\{\bar{\mathbf{u}}(\mathbf{t}_k; \mathbf{i})\}} \cdot [\mathbf{u}(\mathbf{t}_k, \mathbf{i}) - \bar{\mathbf{u}}(\mathbf{t}_k, \mathbf{i})] + \mathbf{v}_y(\mathbf{t}_j) \quad (9)$$

Onde,

k é a partir de zero, uma vez que $\hat{y}(t_j)$ é também uma função de $u(t_{j-2}), \dots, u(t)$ através de sucessivas recursões, começando com $\hat{y}(t_{j-1}), \dots, \hat{y}(t_{j-n_y})$;

$0 < \alpha(i) \leq 1$ é uma constante que deve ser ajustada para garantir a hipótese de aproximação da perturbação linear.

As derivadas parciais que aparecem na equação (9) são calculadas utilizando-se a regra da retro-propagação em relação às saídas dos neurônios da rede *Feedforward*. No apêndice A são desenvolvidas em detalhes as expressões matemáticas necessárias para a determinação destas derivadas parciais. Tem-se também que:

$$\alpha(i) \cdot [\hat{u}(t_{-1}) - \bar{u}(t_1, i)] = [u(t_1, i) - \bar{u}(t_1, i)] + \sum_{k=0}^1 v_u(t_k) \quad (10.a)$$

Para,

$$l = 0, 1, \dots, n-1 \text{ e } i = 1, 2, \dots, I \quad (10.b)$$

Onde,

$\hat{u}(t_{-1})$... é a solução estimada do último passo de controle;

$$\alpha(i) \leftarrow \alpha(i+1);$$

$\bar{u}(t, i+1) = \hat{u}(t_1, i)$... é o valor estimado aproximado de $u(t_1)$ na i -ésima iteração e para $i=1$ são utilizados os valores estimados ou extrapolados do último passo de controle.

Para $j = 1, 2, \dots, n$ e $l = 0, 1, \dots, n-1$ o problema das equações (9) e (10.a) é um de estimação linear estocástica de parâmetros. Este problema pode ser representado numa notação mais compacta e de mais fácil entendimento. Seja então, feitas as seguintes definições:

$$\mathbf{U}_1(\mathbf{t}, \mathbf{i}) \equiv \mathbf{u}(\mathbf{t}_1, \mathbf{i}) \quad (11.a)$$

$$\hat{\mathbf{U}}_1(\mathbf{t}_{-1}) \equiv \hat{\mathbf{u}}(\mathbf{t}_{-1}) \quad (11.b)$$

Assim, o problema pode ser equivalentemente expresso como:

$$\alpha(\mathbf{i}) \cdot [\hat{\mathbf{U}}(\mathbf{t}_{-1}) - \bar{\mathbf{U}}(\mathbf{t}, \mathbf{i})] = \mathbf{U}(\mathbf{t}, \mathbf{i}) - \bar{\mathbf{U}}(\mathbf{t}, \mathbf{i}) + \mathbf{V}_u(\mathbf{t}) \quad (12.a)$$

$$\alpha(\mathbf{i}) \cdot \bar{\mathbf{Z}}^u(\mathbf{t}, \mathbf{i}) = \mathbf{H}^u(\mathbf{t}, \mathbf{i}) \cdot [\mathbf{U}(\mathbf{t}, \mathbf{i}) - \bar{\mathbf{U}}(\mathbf{t}, \mathbf{i})] + \mathbf{V}_y(\mathbf{t}) \quad (12.b)$$

O significado das variáveis compactas nas equações acima são obtidas por comparação direta das equações (12.a) e (12.b) com as equações (9) e (10.a), respectivamente. A seguir é apresentada uma solução heurística e sem demonstração das expressões do filtro de Kalman como solução típica de uma i -ésima iteração deste problema:

$$\begin{aligned} \hat{\mathbf{U}}(\mathbf{t}, \mathbf{i}) &= \bar{\mathbf{U}}(\mathbf{t}, \mathbf{i}) + \alpha(\mathbf{i}) \cdot [\hat{\mathbf{U}}(\mathbf{t}_{-1}) - \bar{\mathbf{U}}(\mathbf{t}, \mathbf{i})] + \\ &\mathbf{k}(\mathbf{t}, \mathbf{i}) \cdot \alpha(\mathbf{i}) \cdot [\bar{\mathbf{Z}}^u(\mathbf{t}, \mathbf{i}) - \mathbf{H}^u(\mathbf{t}, \mathbf{i}) \cdot [\hat{\mathbf{U}}(\mathbf{t}_{-1}) - \bar{\mathbf{U}}(\mathbf{t}, \mathbf{i})]] \end{aligned} \quad (13.a)$$

$$\begin{aligned} \mathbf{k}(\mathbf{t}, \mathbf{i}) &= \mathbf{R}_u(\mathbf{t}) \cdot \mathbf{H}^{uT}(\mathbf{t}, \mathbf{i}) \cdot [\mathbf{H}^u(\mathbf{t}, \mathbf{i}) \cdot \mathbf{R}_u(\mathbf{t}) \cdot \mathbf{H}^{uT}(\mathbf{t}, \mathbf{i}) + \mathbf{R}_y(\mathbf{t})]^{-1} \equiv \\ &[\mathbf{R}_u^{-1}(\mathbf{t}) + \mathbf{H}^{uT}(\mathbf{t}, \mathbf{i}) \cdot \mathbf{R}_y^{-1}(\mathbf{t}) \cdot \mathbf{H}^u(\mathbf{t}, \mathbf{i})]^{-1} \cdot \mathbf{H}^{uT}(\mathbf{t}, \mathbf{i}) \cdot \mathbf{R}_y^{-1}(\mathbf{t}) \end{aligned} \quad (13.b)$$

$$\bar{\mathbf{U}}(\mathbf{t}, \mathbf{i} + 1) = \hat{\mathbf{U}}(\mathbf{t}, \mathbf{i}); \alpha(\mathbf{i}) \leftarrow \alpha(\mathbf{i} + 1); \hat{\mathbf{U}}(\mathbf{t}) = \hat{\mathbf{U}}(\mathbf{t}, \mathbf{I}) \quad (13.c)$$

$$\hat{\mathbf{R}}_u(\mathbf{t}, \mathbf{I}) = [\mathbf{I}_u - \mathbf{K}(\mathbf{t}, \mathbf{I}) \cdot \mathbf{H}^u(\mathbf{t}, \mathbf{I})] \cdot \mathbf{R}_u(\mathbf{t}) \quad (13.d)$$

Para, $\mathbf{i} = 1, 2, \dots, \mathbf{I}$.

Onde,

$\mathbf{R}_u(\mathbf{t})$, $\mathbf{R}_y(\mathbf{t})$ e $\hat{\mathbf{R}}(\mathbf{t}, \mathbf{I})$... são, respectivamente, as matrizes de covariância de $\mathbf{V}_u(\mathbf{t})$, $\mathbf{V}_y(\mathbf{t})$ e $(\hat{\mathbf{U}}(\mathbf{t}, \mathbf{I}) - \mathbf{U}(\mathbf{t}))$;

\mathbf{I}_u ... matriz identidade.

O controle calculado com este algoritmo é o mínimo do funcional:

$$J(\alpha, i) = [[\alpha(i) \cdot \bar{\mathbf{Z}}^u(\mathbf{t}, i) - \mathbf{H}^u(\mathbf{t}, i) \cdot [\mathbf{U}(\mathbf{t}, i) - \bar{\mathbf{U}}(\mathbf{t}, i)]]^T \cdot \mathbf{R}_y^{-1}(\mathbf{t}) \cdot [\alpha(i) \cdot \bar{\mathbf{Z}}^u(\mathbf{t}, i) - \mathbf{H}^u(\mathbf{t}, i) \cdot [\mathbf{U}(\mathbf{t}, i) - \bar{\mathbf{U}}(\mathbf{t}, i)]] + [\mathbf{U}(\mathbf{t}, i) - \bar{\mathbf{U}}(\mathbf{t}, i) - \alpha(i) \cdot [\hat{\mathbf{U}}(\mathbf{t}_{-1}) - \bar{\mathbf{U}}(\mathbf{t}, i)]]^T \cdot \mathbf{R}_u^{-1}(\mathbf{t}) \cdot [\mathbf{U}(\mathbf{t}, i) - \bar{\mathbf{U}}(\mathbf{t}, i) - \alpha(i) \cdot [\hat{\mathbf{U}}(\mathbf{t}_{-1}) - \bar{\mathbf{U}}(\mathbf{t}, i)]]] \quad (14)$$

Desta forma, a convergência para um controle suave $\hat{\mathbf{U}}(\mathbf{t})$ que rastreará a trajetória de referência $\mathbf{y}_r(\mathbf{t})$ é garantida uma vez que a rede neural *Feedforward* $\hat{\mathbf{y}}(\mathbf{t})$ tem a capacidade de representar a dinâmica do sistema da equação (4) e de permitir uma aproximação linearizada (Chen and Billings, 1992) em uma i-ésima típica iteração, quando se emprega um valor suficientemente pequeno para $\alpha(i)$.

4 - REPRESENTAÇÃO DE SISTEMAS DINÂMICOS ATRAVÉS DE REDES NEURAI NA ESTRUTURA DE INTEGRADORES NUMÉRICOS.

Nesta seção uma abordagem teórica e heurística é tomada com o propósito de mostrar explicitamente como utilizar integradores numéricos de sistemas de equações diferenciais ordinárias em esquemas de controle onde um modelo interno discreto para frente é necessário (Wang e Lin, 1998; Rios Neto, 2001). A estrutura interna dos algoritmos de integração numérica é aproveitada para obter um modelo de rede neural com arquitetura *Feedforward* que necessite somente aprender a função de derivadas do modelo original representado por sistemas de equações diferenciais ordinárias. Nesta metodologia, os integradores numéricos e a rede neural *Feedforward* trabalharão em conjunto para representar a dinâmica discretizada do sistema original. Feito isto, então será ilustrado como utilizar um esquema de controle preditivo para a obtenção de uma política de controle suave que também rastreie a trajetória de referência.

Em esquemas de controle empregando redes neurais do tipo *Feedforward*, a abordagem usual é utilizar as estruturas dos integradores numéricos para gerar os padrões de treinamento da rede e, só depois, então, treinar a rede neural através de um aprendizado supervisionado. Há no mínimo duas dificuldades com esta abordagem: a principal é ter que lidar com muitas entradas na rede neural e, portanto, muitos parâmetros a serem ajustados, dificultando o treinamento da rede; e a secunda é a impossibilidade de variar o passo de integração. Na nova abordagem que será adotada aqui poder-se-á diminuir o número de entradas da rede e variar constantemente o passo de integração Δt . Na abordagem usual os integradores numéricos são utilizados somente para gerar os padrões de treinamento e na abordagem usada ele trabalhará em conjunto com a rede como ilustra a figura 3.

O modelo de simulação computacional provido por um integrador numérico de sistemas de equações diferenciais ordinárias é naturalmente um modelo discreto para frente de sistemas dinâmicos, que por si só pode ser utilizado como um modelo interno em esquemas de controle. Estes integradores numéricos possuem características que são bastante relevantes para um modelo de sistemas dinâmicos a ser utilizado em controle, uma vez que eles permitem: processamento paralelo de todas as componentes dos estados do sistema dinâmico, o erro local e a precisão do integrador podem ser avaliados aplicando métodos que automaticamente variam a ordem e o *tamanho do passo* de integração do integrador e a estimação dos erros globais acumulados podem também ser feita (e.g., Rios Neto e Rama Rao, 1990).

Quando uma estrutura de um modelo de integrador numérico é utilizada, é possível ter uma rede neural com arquitetura *Feedforward* para aproximar a função de derivadas de um modelo matemático de equações diferenciais da dinâmica do sistema. Neste tipo de abordagem, a dificuldade em lidar com muitas entradas no treinamento da rede neural é aliviada, uma vez que é somente necessário aprender uma função algébrica e estática onde as entradas da rede são somente as ocorrências das variáveis de estado e de controle em seus respectivos domínios de atuação. Lembre-se que na abordagem convencional às vezes é necessário lidar com muitas entradas atrasadas das variáveis de estado e controle, mas nesta nova metodologia isso não será mais necessário.

Considere-se, pois, um sistema dinâmico e suponha que o modelo matemático ou o modelo de uma rede neural artificial representando a função de derivadas deste sistema são conhecidos. Assim, tem-se:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \cong \hat{\mathbf{f}}(\mathbf{x}, \mathbf{u}, \hat{\mathbf{w}}) \quad (15)$$

Onde,

$\mathbf{x} \in \mathbf{R}^n$... é o vetor das variáveis de estados;

$\mathbf{u} \in \mathbf{R}^n$... é o vetor das variáveis de controle;

$\hat{\mathbf{w}}$... são os pesos aprendidos pela rede neural;

$\mathbf{f}(\mathbf{x}, \mathbf{u})$... é a função de derivadas;

$\hat{\mathbf{f}}(\mathbf{x}, \mathbf{u}, \hat{\mathbf{w}})$... representa o treinamento da rede neural para aproximar a função de derivadas.

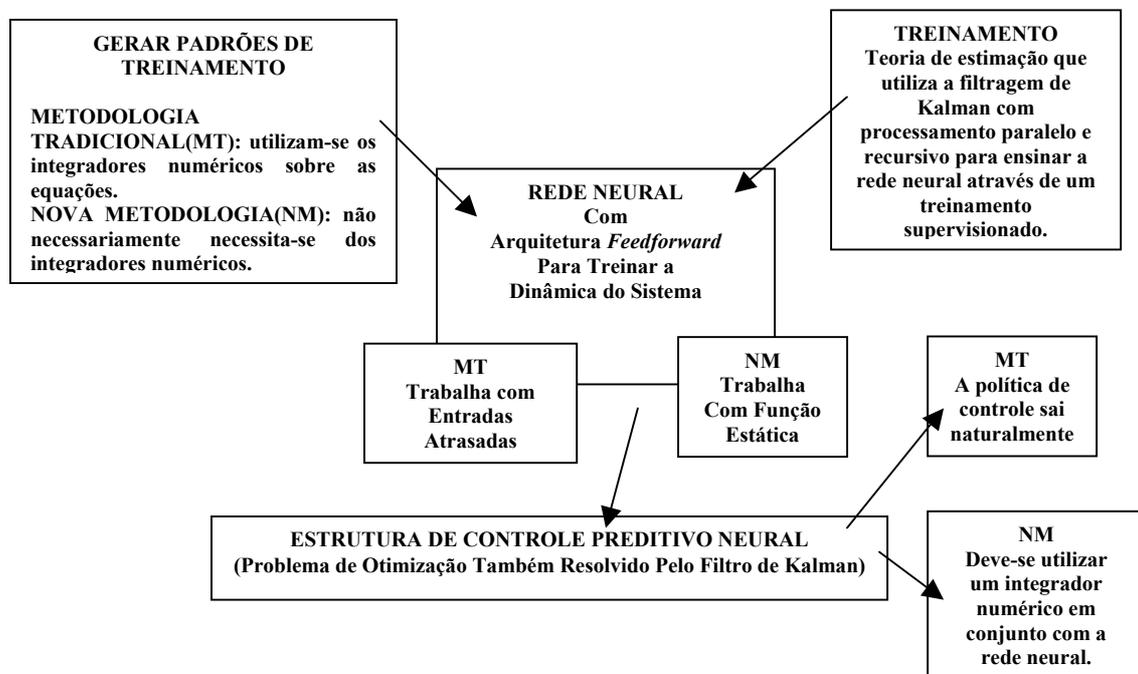


FIGURA 3 – Distinção entre a metodologia tradicional (MT) e a nova metodologia (NM).

Considere agora um integrador numérico para obter uma aproximação discreta do sistema da equação (15):

$$\mathbf{x}(t + \Delta t) \cong \mathbf{f}_n(\mathbf{x}(t), \mathbf{x}(t - \Delta t), \dots, \mathbf{x}(t - n_0 \Delta t); \mathbf{u}(t), \dots, \mathbf{u}(t - n_0 \Delta t); \Delta t) \quad (16)$$

Onde,

n_0 ... é o número de entradas atrasadas do integrador numérico ou a ordem de aproximação.

No apêndice B há uma classificação dos tipos principais de integradores numéricos segundo Rao, (1984) em função do valor numérico de n_0 . Se n_0 é igual a zero tem-se os integradores de passo simples, por exemplo, os integradores de Euler ou de Runge-Kutta. Se n_0 maior que zero tem-se os integradores de múltiplos passos ou de diferenças finitas, como por exemplo, os integradores de Adams-Bashforth.

Um integrador numérico na equação (16) pode ser utilizado como o modelo preditivo discreto do sistema dinâmico dado pela equação (15) em um esquema de controle de modelo interno. O erro em cada passo pode ser controlado por variar o tamanho do passo ou a ordem do integrador numérico. O resultado numérico do algoritmo pode ser processado em paralelo para cada componente do estado do sistema dinâmico.

Sendo possível aproximar um sistema dinâmico, através de uma estrutura de integrador numérico de equações diferenciais ordinárias trabalhando em conjunto com uma rede neural artificial que necessita somente aprender a função de derivadas do sistema dinâmico, então resta saber agora qual a vantagem de se representar a função de derivadas através de uma rede neural artificial, sendo que a própria função de derivadas, em geral, já possui uma forma analítica conhecida que poderá ser utilizada diretamente nas estruturas do integrador numérico. Não se estaria fazendo um trabalho desnecessário? Para responder a esta pergunta veja o esquema proposto na figura 4. Como se sabe, todo modelo teórico representado, em geral, através de sistemas de equações diferenciais possui uma margem de erro em relação ao sistema real que ele pretende representar matematicamente, pois quase sempre existirá pelo menos uma

variável de estado de difícil modelagem matemática em sistemas muito complexos. Tendo o modelo teórico em mãos, ainda assim, é necessário resolvê-lo através de algum método numérico quando este é não-linear. Em teoria de controle é muito comum obter a solução teórica do sistema dinâmico através da utilização de integradores numéricos. A solução assim obtida será representada pelo vetor de estados x^m como ilustra a figura 4.

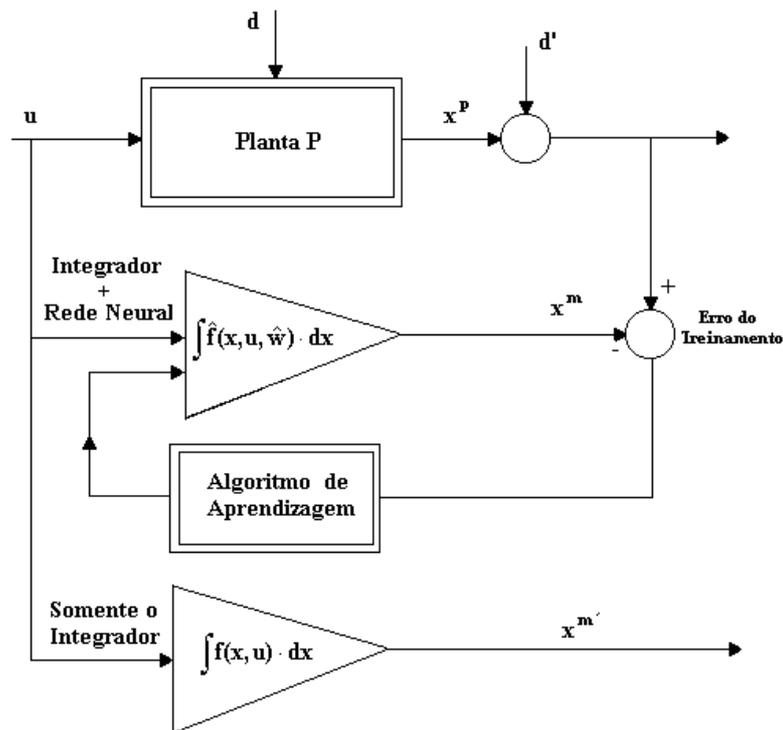


FIGURA 4 - Esquema ilustrativo: vantagens de se representar à função de derivadas através de uma rede neural artificial na estrutura de integrador numérico.

Entretanto, como se vê na figura 4 não é, em geral, possível corrigir o modelo teórico em relação ao sistema real quando se utiliza uma estrutura constituída puramente por um integrador numérico para resolver numericamente o modelo teórico original (com exceção para o caso em que se conhece uma estrutura parametrizada da dinâmica). Por outro lado, quando se utiliza uma rede neural para representar a função de derivadas do modelo teórico do sistema dinâmico ainda será possível continuar o treinamento da rede neural que modela a função de derivadas em tempo real, para que o modelo

teórico se aproxime cada vez mais do real. A solução obtida por esse último método é representada pelo vetor de estados \mathbf{x}^m da figura 5. Nesta última metodologia a diferença do erro de $(\mathbf{x}^p - \mathbf{x}^m)$ tende a diminuir com o passar do tempo de funcionamento da planta. No primeiro modelo a diferença do erro $(\mathbf{x}^p - \mathbf{x}^{m'})$ tende a crescer.

Por outro lado, se for adicionado e ajustado um processo de Gauss-Markov ao modelo dinâmico original para se compensar constantemente em tempo real os erros de modelagem, isso também poderia ser suficiente para corrigir o modelo teórico em relação ao real da planta. Entretanto, neste caso, certamente não se teria a capacidade de aprendizagem da solução proposta com redes neurais e integradores.

Uma vantagem que pode ser tirada do esquema da figura 4 é que não é necessário conhecer previamente a função de derivadas do sistema dinâmico para utilizar a estrutura de treinamento composta por um integrador numérico e uma rede neural. Acaso se conheça somente um número suficientemente grande de entradas/estados da planta, por exemplo, obtidas através de sensores, então através de um treinamento supervisionado a estrutura composta pelo conjunto integrador numérico e rede neural poderá ser treinada para aprender a dinâmica do sistema. É interessante perceber que, mesmo neste caso, a rede continuará fazendo um papel semelhante ao da função de derivadas e, portanto ela poderá ser projetada com uma arquitetura que, em geral, exigirá menos camadas e menos neurônios. O inconveniente desta metodologia é que dever-se-á adaptar a retro-propagação dentro da estrutura de integração numérica com a aplicação da regra da cadeia (veja-se, como exemplo, Wang e Lin (1998) que desenvolvem essa metodologia para o Runge-Kutta de quarta ordem).

Sendo possível representar sistemas dinâmicos através do conjunto integrador numérico e redes neurais, resta saber como projetar este esquema sobre uma estrutura de controle preditivo. No caso presente, deverá ser empregada a regra da cadeia sobre a estrutura do integrador combinada com a retro-propagação da rede neural. Desta forma, quanto maior a ordem do integrador mais complexo se tornará o cálculo desta derivadas. Por

ora, é importante salientar que o problema continua sendo determinar um controle suave que rastreie a trajetória de referência, por minimizar o índice de performance quadrático do tipo:

$$\mathbf{J} = \left[\sum_{j=1}^n [\mathbf{y}_r(t_j) - \mathbf{y}_n(t_j)]^T \cdot \mathbf{r}_y^{-1}(t) \cdot [\mathbf{y}_r(t_j) - \mathbf{y}_n(t_j)] + \sum_{j=0}^{n-1} [\mathbf{u}(t_j) - \mathbf{u}(t_{j-1})]^T \cdot \mathbf{r}_u^{-1}(t) \cdot [\mathbf{u}(t_j) - \mathbf{u}(t_{j-1})] \right] / 2 \quad (17.a)$$

Para,

$$\mathbf{t}_j = \mathbf{t} + j \cdot \Delta t \quad (17.b)$$

Onde,

$\mathbf{y}_r(t_j)$... é a trajetória de referência;

n ... horizonte de atuação do controle suave;

$\mathbf{r}_y^{-1}(t_j)$ e $\mathbf{r}_u^{-1}(t_j)$... matrizes de pesos positiva definida;

$\hat{\mathbf{y}}(t_j)$... é a saída aproximada do sistema dinâmico da equação (15).

A saída aproximada para o sistema dinâmico da equação (15) pode ser formalmente representada por:

$$\mathbf{y}_n(t_j) \cong \mathbf{g}[\mathbf{x}(t_j)] \cong \mathbf{g}[\mathbf{f}_n(\mathbf{x}(t_{j-1}), \dots, \mathbf{x}(t_{j-1-n_0}); \mathbf{u}(t_{j-1}), \dots, \mathbf{u}(t_{j-1-n_0}); \Delta t; \hat{\mathbf{w}})] = \mathbf{g}_n(\mathbf{x}(t_{j-1}), \dots, \mathbf{x}(t_{j-1-n_0}); \mathbf{u}(t_{j-1}), \dots, \mathbf{u}(t_{j-1-n_0}); \hat{\mathbf{w}}) \quad (18)$$

Onde,

$\hat{\mathbf{w}}$... pesos de treinamento da rede neural que representa a função de derivadas do sistema dinâmico;

$y_n(t_j)$... resposta do sistema dinâmico discretizado pelo conjunto integrador e rede.

A possibilidade de ajustar o nível de aproximação do integrador numérico e da rede neural garante a precisão necessária para $y_n(t_j)$ ao longo do horizonte preditivo. A solução do problema de programação não-linear do índice de desempenho dado pela equação (17.a), por qualquer que seja o método, envolverá a necessidade de calcular uma aproximação do gradiente de saída do sistema dinâmico, para obter uma aproximação linearizada em cada passo de busca:

$$y_n(t_j) = \bar{y}(t_j) + \sum_{k=0}^{j-1} [\partial y_n(t_j) / \partial u(t_k)]_{\{\bar{u}(t_k)\}} \cdot [u(t_k) - \bar{u}(t_k)] \quad (19.a)$$

Onde k é a partir de zero, uma vez que $y_n(t_j)$ é também uma função de $u(t_{j-2}), \dots, u(t)$ através de sucessivas recursões no horizonte de controle, começando com $x_n(t_{j-1}), \dots, x_n(t_{j-n_0})$. As derivadas parciais são calculadas, como mencionado já anteriormente, utilizando-se a regra da cadeia.

Note que em vez de se utilizar uma rede neural para prover um modelo discreto para frente para a dinâmica do sistema, pode-se utilizar diretamente o algoritmo do integrador numérico como o modelo discreto para frente para aproximar o modelo matemático da dinâmica do sistema da equação (15). No caso do esquema de controle preditivo, o modelo do integrador numérico da equação (16) pode ser diretamente utilizado para calcular as derivadas parciais da equação (19.a).

Há duas formas de treinamento supervisionado que a metodologia dos integradores numéricos permite realizar quando se conhece a priori a função de derivadas do modelo teórico. Como ilustra a figura 5, as duas maneiras distintas de realizar este treinamento são:

- a) a primeira metodologia é simplesmente gerar aleatoriamente valores suficientes da função de derivadas dentro de um domínio de interesse e treinar diretamente

a rede neural para aprender a função de derivadas (Rios Neto, 2001) do modelo teórico. Terminado o treinamento, a simulação dos resultados deverá ser feita aplicando-se a função de derivadas treinada pela rede no lugar da teórica sobre a estrutura do integrador. Para a análise da compatibilidade dos erros obtidos pela rede e pelo integrador, a solução numérica do integrador deverá ser obtida pela aplicação das duas funções de derivadas, a treinada e a original, para que os resultados da propagação da solução sejam comparados. Nesta metodologia, o integrador numérico só será utilizado na simulação dos resultados;

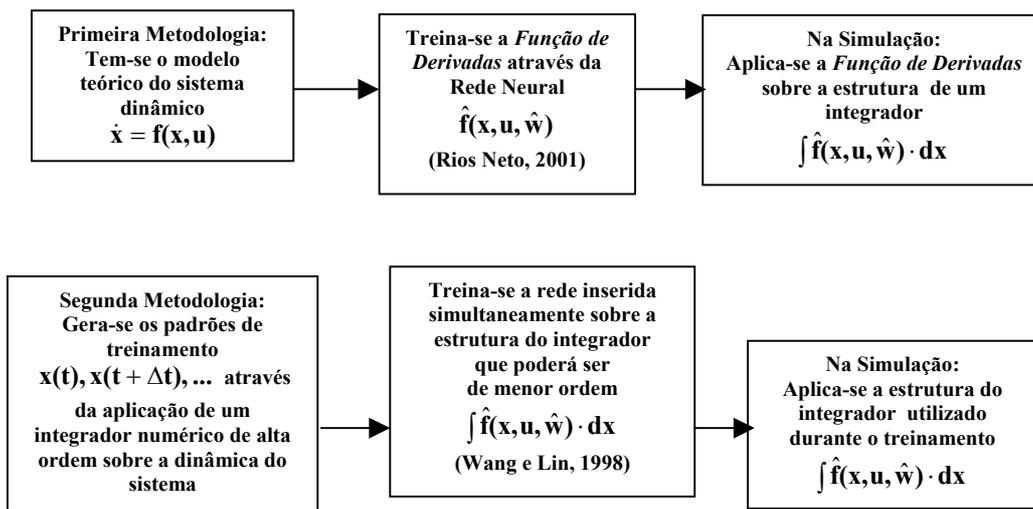


FIGURA 5 – Maneiras distintas de se treinar uma rede neural na estrutura de um integrador numérico.

- b) na segunda metodologia gera-se as soluções temporais do sistema dinâmico $x(t), x(t + \Delta t), \dots, x(t + n \cdot \Delta t)$ para um número suficiente de condições iniciais e valores de controle através da estrutura de um integrador, de preferência, de alta ordem para garantir a alta precisão dos padrões de treinamento. O aprendizado da função de derivadas pela rede neural só será possível se a rede for treinada amarrada à estrutura do integrador (Wang e Lin, 1998). É interessante observar que durante o treinamento poder-se-á diminuir a ordem do integrador em relação àquele utilizado para gerar os padrões de treinamento. A análise do erro obtido pelo conjunto integrador e rede neural é realizada também pela comparação das soluções numéricas propagadas pelas funções de

derivadas original e treinada. Nesta metodologia, o integrador numérico é utilizado na geração dos padrões de treinamento e é requerido durante o treinamento e a simulação da rede.

O erro de treinamento da rede neural, em geral, é medido pela média do erro quadrático (**meq**) dos padrões de treinamento e o erro do integrador (**EI**) numérico é dividido em quatro partes: erro de truncamentolocal (τ), erro global (ν), erro local (λ) e *round-off error* (**re**). Na primeira metodologia de treinamento, apresentada na figura 5, o erro cometido pela rede (**meq**) será propagado pelo integrador. Na segunda metodologia tem que se explorar melhor se este erro será totalmente propagado pelo integrador.

5 - O ALGORITMO COMPUTACIONAL COMBINANDO INTEGRADORES NEURAI E ESTRUTURA DE CONTROLE PREDITIVO.

Numa estrutura de controle preditivo os parâmetros a serem estimados são os próprios controles que manterão o sistema dinâmico em torno de uma trajetória de referência dentro de um horizonte de controle desejado. Nesta seção apresentar-se-á um algoritmo em linguagem humana do filtro de Kalman com o objetivo de facilitar a implementação computacional deste problema, que é um de programação dinâmica não-linear. Como é difícil construir um algoritmo genérico então, serão assumidos valores particulares para n , n_y e n_u . Neste caso, se adotará $n=1$ (horizonte de estimação das variáveis de controle) e $n_y=n_u=4$ (número de entradas atrasadas das variáveis de estado e de controle da estrutura de integração neural).

Como o problema de estimação proposto é não linear, então o algoritmo possuirá uma característica iterativa, ou seja, parte-se de uma informação a priori das variáveis de controle, e em seguida, estima-se novos valores destas variáveis até que a saída dos estados do sistema obtida pela rede neural convirja para a trajetória de referência conhecida, dentro de um erro aceitável para esse horizonte, como ilustra a Figura 6.

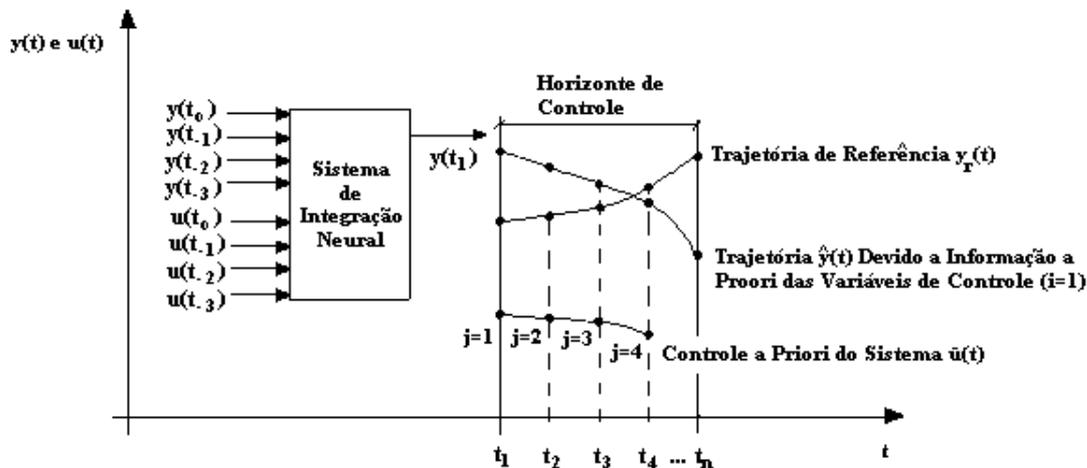


FIGURA 6 - Esquema gráfico das condições iniciais das variáveis de controle no esquema de controle preditivo.

Ainda com relação a Figura 6 observe que os valores das variáveis de estado $y(t_3)$, $y(t_2)$, $y(t_1)$ e $y(t_0)$ e das variáveis de controle $u(t_3)$, $u(t_2)$ e $u(t_1)$ devem ser conhecidos assim como a trajetória de referência e a informação a priori dos controles a serem atuados dentro do horizonte de previsão desejado. Deste modo, o algoritmo para a solução do problema proposto pode ser sumarizado nos seguintes passos essenciais:

- 1) Gerar uma informação a priori dos controles $u(t)$ dentro do horizonte desejado. Como a política de controle desejada deverá ser suave como imposta pelo funcional J da equação (6) nada mais natural do que iniciá-la com o vetor nulo (só no início do controle, no instante inicial), ou seja,

$$\bar{U}(t, i) = [\bar{u}(t_0, i)]^T \equiv \mathbf{0} \quad \text{para } i=1 \quad (20)$$

Onde,

$\bar{u}(t_0, i)$... vetor das variáveis de controle no instante t_0 na i -ésima iteração.

2) Calcular as derivadas parciais $\frac{\partial \hat{y}(t_j)}{\partial \mathbf{u}(t_k)}$ presentes na equação (9) através das expressões de (33.a) à (33.b) da retro-propagação. A tabela 1 ilustra o caso particular para $n=1$ e $n_y=n_u=4$.

$$\alpha(\mathbf{i}) \cdot [y_r(t_j) - \bar{y}(t_j, \mathbf{i})] = \sum_{k=0}^{j-1} [\partial \hat{y}(t_j) / \partial \mathbf{u}(t_k)]_{\{\bar{\mathbf{u}}(t_k; \mathbf{i})\}} \cdot [\mathbf{u}(t_k, \mathbf{i}) - \bar{\mathbf{u}}(t_k, \mathbf{i})] + \mathbf{v}_y(t_j) \quad (21)$$

A derivada vetorial presente na tabela 1 poderá apresentar uma grande dificuldade para ser determinada: quando o instante de tempo associado às variáveis de controle \mathbf{u} for muito atrasado, isso ocorre quando se adota valores muito grandes para \mathbf{n} , a ponto do número de entradas atrasadas presentes no sistema de integração neural não ser suficiente para permitir sua determinação a partir das equações básicas da retro-propagação, então dever-se-á empregar a regra da cadeia combinada com a retro-propagação tornando o método muito complicado, principalmente se o sistema de integração adotado for do tipo Runge-Kutta de alta ordem. Pode-se evitar isso facilmente empregando um valor pequeno para \mathbf{n} .

TABELA 1- Derivadas parciais $\frac{\partial \hat{y}(t_j)}{\partial \mathbf{u}(t_k)}$ necessárias para estimar os controles $\mathbf{u}(t)$ dentro de um horizonte de controle com $n=1$ e entradas atrasadas da rede neural iguais a $n_y=n_u=4$.

J	k	j-1	Derivadas Parciais (Retro-propagação)
1	0	0	$\frac{\partial \hat{y}(t_1)}{\partial \mathbf{u}(t_0)} \Big _{\bar{\mathbf{u}}(t_0, \mathbf{i})}$

Para que a derivada vetorial parcial presente na Tabela 1 seja computada corretamente é necessário a priori estabelecer o tipo de integrador numérico no qual se introduzirá a função de derivadas neurais do sistema dinâmico a ser estudado. No presente caso será

adotado o integrador de múltiplos passos Adams-Bashforth de ordem quatro (ver apêndice B) dado por:

$$\begin{aligned} y(t_1) = y(t_0) + \frac{h}{24} \cdot \{ & 55 \cdot \hat{f}[y(t_0), u(t_0)] - 59 \cdot \hat{f}[y(t_{-1}), u(t_{-1})] \\ & + 37 \cdot \hat{f}[y(t_{-2}), u(t_{-2})] - 9 \cdot \hat{f}[y(t_{-3}), u(t_{-3})] \} \end{aligned} \quad (22)$$

Assim, o cálculo da derivada $\left. \frac{\partial \hat{y}(t_1)}{\partial u(t_0)} \right|_{\bar{u}(t_0, i)}$, para o caso particular do integrador adotado, toma a seguinte forma:

$$\left. \frac{\partial \hat{y}(t_1)}{\partial u(t_0)} \right|_{\bar{u}(t_0, i)} = \frac{55}{24} \cdot h \cdot \left. \frac{\partial \hat{f}[y(t_0), u(t_0)]}{\partial u(t_0)} \right|_{\bar{u}(t_0, i)} \quad (23)$$

A equação anterior pode ser computada diretamente através da Retro-propagação. Para maiores detalhes sobre o cálculo destas derivadas ver o apêndice A.

3) Montar a matriz $\mathbf{H}^u(t, i)$ e o vetor $\bar{\mathbf{Z}}^u(t, i)$ presentes na equação $\alpha(i) \cdot \bar{\mathbf{Z}}^u(t, i) = \mathbf{H}^u(t, i) \cdot [\mathbf{U}(t, i) - \bar{\mathbf{U}}(t, i)] + \mathbf{V}_y(t)$. Deve-se construir a matriz $\mathbf{H}^u(t, i)$ à partir da Tabela 1. Para simplificar a notação que está sendo utilizada é conveniente adotar a seguinte definição:

$$\mathbf{A}_{k,j} = \left. \frac{\partial \hat{y}(t_k)}{\partial u(t_j)} \right|_{\bar{u}(t_j, i)} = \begin{bmatrix} \frac{\partial \hat{y}_1(t_k)}{\partial u_1(t_j)} & \frac{\partial \hat{y}_1(t_k)}{\partial u_2(t_j)} & \dots & \frac{\partial \hat{y}_1(t_k)}{\partial u_{n_u}(t_j)} \\ \frac{\partial \hat{y}_2(t_k)}{\partial u_1(t_j)} & \frac{\partial \hat{y}_2(t_k)}{\partial u_2(t_j)} & \dots & \frac{\partial \hat{y}_2(t_k)}{\partial u_{n_u}(t_j)} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \hat{y}_{n_y}(t_k)}{\partial u_1(t_j)} & \frac{\partial \hat{y}_{n_y}(t_k)}{\partial u_2(t_j)} & \dots & \frac{\partial \hat{y}_{n_y}(t_k)}{\partial u_{n_u}(t_j)} \end{bmatrix} \quad (24)$$

Montar a matriz $\mathbf{H}^u(t, i)$ seria uma tarefa bastante difícil se os valores de \mathbf{n} , n_u e n_y fossem grandes, como exemplo, $\mathbf{n}=5$ e $n_u=n_y=3$. Neste caso, apenas como efeito ilustrativo, ter-se-ia o seguinte:

$$\mathbf{H}^u(t, \mathbf{i}) = \begin{bmatrix} \mathbf{A}_{1,0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_{2,0} & \mathbf{A}_{2,1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_{3,0} & \mathbf{A}_{3,1} & \mathbf{A}_{3,2} & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_{4,0} & \mathbf{A}_{4,1} & \mathbf{A}_{4,2} & \mathbf{A}_{4,3} & \mathbf{0} \\ \mathbf{A}_{5,0} & \mathbf{A}_{5,1} & \mathbf{A}_{5,2} & \mathbf{A}_{5,3} & \mathbf{A}_{5,4} \end{bmatrix} \quad (25)$$

onde cada elemento genérico $\mathbf{A}_{k,j}$ presentes na matriz anterior seria outra matriz de dimensão $\mathbf{n}_y \times \mathbf{n}_u$ como indica a equação (24). Entretanto, como tem-se somente $\mathbf{n}=1$ então, a matriz $\mathbf{H}^u(t, \mathbf{i})$ se reduz a:

$$\mathbf{H}^u(t, \mathbf{i}) = \mathbf{A}_{1,0} = \frac{\partial \hat{y}_j(t_1)}{\partial \mathbf{u}(t_0)} \Big|_{\bar{\mathbf{u}}(t_0, \mathbf{i})} = \left[\frac{\partial \hat{y}_j(t_1)}{\partial \mathbf{u}_k(t_0)} \Big|_{\bar{\mathbf{u}}(t_0, \mathbf{i})} \right]_{\mathbf{n}_y \times \mathbf{n}_u} \quad (26)$$

Deste modo expandindo-se a expressão $\alpha(\mathbf{i}) \cdot \bar{\mathbf{Z}}^u(t, \mathbf{i}) = \mathbf{H}^u(t, \mathbf{i}) \cdot [\mathbf{U}(t, \mathbf{i}) - \bar{\mathbf{U}}(t, \mathbf{i})] + \mathbf{V}_y(t)$, tem-se:

$$\underbrace{\begin{bmatrix} y_{r_1}(t_1) - \bar{y}_1(t_1, \mathbf{i}) \\ y_{r_2}(t_1) - \bar{y}_2(t_1, \mathbf{i}) \\ \vdots \\ y_{r_{n_y}}(t_1) - \bar{y}_{n_y}(t_1, \mathbf{i}) \end{bmatrix}}_{\bar{\mathbf{Z}}^u(t, \mathbf{i})} \cdot \alpha(\mathbf{i}) = \underbrace{\begin{bmatrix} \frac{\partial \hat{y}_1(t_1)}{\partial \mathbf{u}_1(t_0)} & \frac{\partial \hat{y}_1(t_1)}{\partial \mathbf{u}_2(t_0)} & \dots & \frac{\partial \hat{y}_1(t_1)}{\partial \mathbf{u}_{n_u}(t_0)} \\ \frac{\partial \hat{y}_2(t_1)}{\partial \mathbf{u}_1(t_0)} & \frac{\partial \hat{y}_2(t_1)}{\partial \mathbf{u}_2(t_0)} & \dots & \frac{\partial \hat{y}_2(t_1)}{\partial \mathbf{u}_{n_u}(t_0)} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \hat{y}_{n_y}(t_1)}{\partial \mathbf{u}_1(t_0)} & \frac{\partial \hat{y}_{n_y}(t_1)}{\partial \mathbf{u}_2(t_0)} & \dots & \frac{\partial \hat{y}_{n_y}(t_1)}{\partial \mathbf{u}_{n_u}(t_0)} \end{bmatrix}}_{\mathbf{H}^u(t, \mathbf{i})} \Big|_{\bar{\mathbf{u}}(t_0, \mathbf{i})} \cdot \underbrace{\begin{bmatrix} \mathbf{u}_1(t_0, \mathbf{i}) - \bar{\mathbf{u}}_1(t_0, \mathbf{i}) \\ \mathbf{u}_2(t_0, \mathbf{i}) - \bar{\mathbf{u}}_2(t_0, \mathbf{i}) \\ \vdots \\ \mathbf{u}_{n_u}(t_0, \mathbf{i}) - \bar{\mathbf{u}}_{n_u}(t_0, \mathbf{i}) \end{bmatrix}}_{\mathbf{U}(t, \mathbf{i}) - \bar{\mathbf{U}}(t, \mathbf{i})} + \underbrace{\begin{bmatrix} \mathbf{v}_1(t_1) \\ \mathbf{v}_2(t_1) \\ \vdots \\ \mathbf{v}_{n_y}(t_1) \end{bmatrix}}_{\mathbf{V}_y(t)}$$

(27)

Onde,

$\bar{\mathbf{U}}(\mathbf{t}, \mathbf{i})$... vetor de informações a priori das variáveis de controle adotado no passo 1;

$\mathbf{U}(\mathbf{t}, \mathbf{i})$... Vetor que deverá ser estimado;

$\mathbf{H}^u(\mathbf{t}, \mathbf{i})$... Matriz de derivadas parciais obtida no passo 2 através da retro-propagação e montada no passo 3;

$0 < \alpha(\mathbf{i}) \leq 1$... Valor que deve ser adotado empiricamente para garantir a hipótese de linearização;

$\mathbf{V}_y(\mathbf{t})$... Vetor de ruídos de média nula e matriz de covariância $\mathbf{r}_y(\mathbf{t})$.

- 4) Estimar $\hat{\mathbf{U}}(\mathbf{t}, \mathbf{i})$ através das seguintes expressões iterativas do filtro de Kalman (Rios Neto, 2000):

$$\mathbf{k}(\mathbf{t}, \mathbf{i}) = \mathbf{R}_u(\mathbf{t}) \cdot \mathbf{H}^{uT}(\mathbf{t}, \mathbf{i}) \cdot [\mathbf{H}^u(\mathbf{t}, \mathbf{i}) \cdot \mathbf{R}_u(\mathbf{t}) \cdot \mathbf{H}^{uT}(\mathbf{t}, \mathbf{i}) + \mathbf{R}_y(\mathbf{t})]^{-1} \equiv$$
$$[\mathbf{R}_u^{-1}(\mathbf{t}) + \mathbf{H}^{uT}(\mathbf{t}, \mathbf{i}) \cdot \mathbf{R}_y^{-1}(\mathbf{t}) \cdot \mathbf{H}^u(\mathbf{t}, \mathbf{i})]^{-1} \cdot \mathbf{H}^{uT}(\mathbf{t}, \mathbf{i}) \cdot \mathbf{R}_y^{-1}(\mathbf{t}) \quad (28.a)$$

$$\hat{\mathbf{U}}(\mathbf{t}, \mathbf{i}) = \bar{\mathbf{U}}(\mathbf{t}, \mathbf{i}) + \alpha(\mathbf{i}) \cdot [\hat{\mathbf{U}}(\mathbf{t}_{-1}) - \bar{\mathbf{U}}(\mathbf{t}, \mathbf{i})] +$$
$$\mathbf{k}(\mathbf{t}, \mathbf{i}) \cdot \alpha(\mathbf{i}) \cdot [\bar{\mathbf{Z}}^u(\mathbf{t}, \mathbf{i}) - \mathbf{H}^u(\mathbf{t}, \mathbf{i}) \cdot [\hat{\mathbf{U}}(\mathbf{t}_{-1}) - \bar{\mathbf{U}}(\mathbf{t}, \mathbf{i})]] \quad (28.b)$$

Onde,

$\mathbf{R}_u(\mathbf{t})$ e $\mathbf{R}_y(\mathbf{t})$... matrizes de covariâncias, respectivamente, de $\mathbf{V}_u(\mathbf{t})$ e $\mathbf{R}_y(\mathbf{t})$;

$\hat{\mathbf{U}}(\mathbf{t}_{-1})$... estimativa anterior das variáveis de controle dentro de uma precisão aceitável.

Observe que o processamento das equações (28.a) e (28.b) pode ser efetuado tanto em lotes como recursivamente.

- 5) Implementar a variável i de $i=i+1$, fazer $\bar{U}(t, i+1) = \hat{U}(t, i)$ e repetir os passos (2),(3) e (4) até que $y(t)$ esteja suficientemente próximo de $y_r(t)$ dentro de um erro desejado, em geral, $3 \cdot \sigma$ de v_y . Pode-se utilizar o critério do erro quadrático médio para determinar a última iteração I para a estimação presente. Quando isso ocorrer fazer:

$$\hat{R}_u(t, I) = [I_u - k(t, I) \cdot H^u(t, I)] \cdot R_u(t) \quad (29)$$

Onde,

$\hat{R}_u(t, I)$... estimação da matriz de covariâncias de $[\hat{U}(t, I) - U(t)]$.

- 6) Quando a estimação atual convergir para o valor desejado $y_r(t)$ *avançar apenas um instante de tempo para frente*, mesmo que o horizonte de previsão adotado n seja maior que um, e repetir todos os passos anteriores para a nova estimação. não esquecer que agora ter-se-á $\hat{U}(t_{-1}) = \hat{U}(t, I)$ e $\bar{U}(t, 1) = \hat{U}(t, I)$.

6 - RESULTADOS E SIMULAÇÕES.

Os resultados práticos, da metodologia desenvolvida neste trabalho, serão apresentados para o problema de transferência de órbita Terra/Marte como ilustra a Figura 7. As variáveis de estado deste problema são: a massa do foguete m , o raio da órbita r , a velocidade radial w e a velocidade transversal v . A única variável de controle é o ângulo guiado do empuxo θ . As equações de *estado* (Sage, 1968) da dinâmica da transferência de órbita Terra/Marte do foguete de massa m são:

$$\dot{m} = -0.0749 \quad (30.a)$$

$$\dot{r} = w \quad (30.b)$$

$$\dot{w} = \frac{v^2}{r} - \frac{\mu}{r^2} + \frac{T \cdot \sin\theta}{m} \quad (30.c)$$

$$\dot{v} = \frac{-w \cdot v}{r} + \frac{T \cdot \cos\theta}{m} \quad (30.d)$$

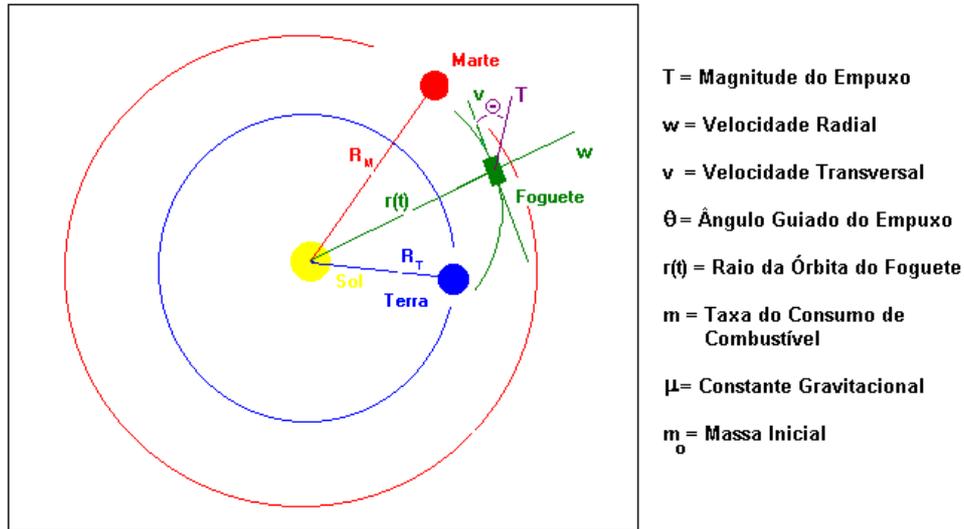


FIGURA 7 – Esquema ilustrativo da transferência de órbita Terra/Marte.

As constantes normalizadas deste problema são: $\mu=1.0$ (constante gravitacional), $T=0.1405$ (empuxo do foguete), $t_0=0$ e $t_f=3.3$, onde cada unidade de tempo é equivalente a **58.2** dias. Uma solução numérica da *dinâmica* obtida pelo integrador neural é apresentada na Figura 8, com o emprego de uma política de controle aleatória entre $-\pi$ e $+\pi$ para o controle θ . A curva contínua representa a solução numérica obtida pelo integrador Runge-Kutta de quarta ordem que utilizou a função de derivadas original, os círculos representam a solução numérica do Runge-Kutta de quarta ordem que utilizou a função de derivadas neural e os triângulos são as soluções obtidas pelo integrador de Adams-Bashforth de quarta ordem que também utilizou a função de derivadas neural. Neste gráfico é importante observar que há uma atualização das condições iniciais, em relação ao modelo teórico original, a cada 100 passos de integração. Na Figura 9 tem-se o resultado de controle preditivo neural, onde o sistema parte com erro desprezível em relação ao início da trajetória de referência, mas por possuir um passo de integração e estimação iguais entre si e com o valor relativamente alto de 0.2 os resultados finais da simulação não ficaram muito precisos.

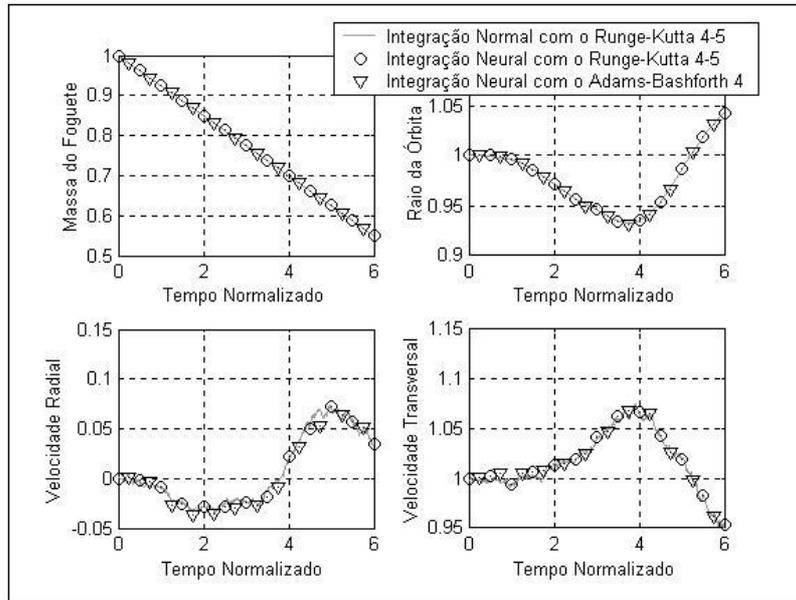


FIGURA 08 - Simulação com integrador neural para a dinâmica do problema Terra/Marte.

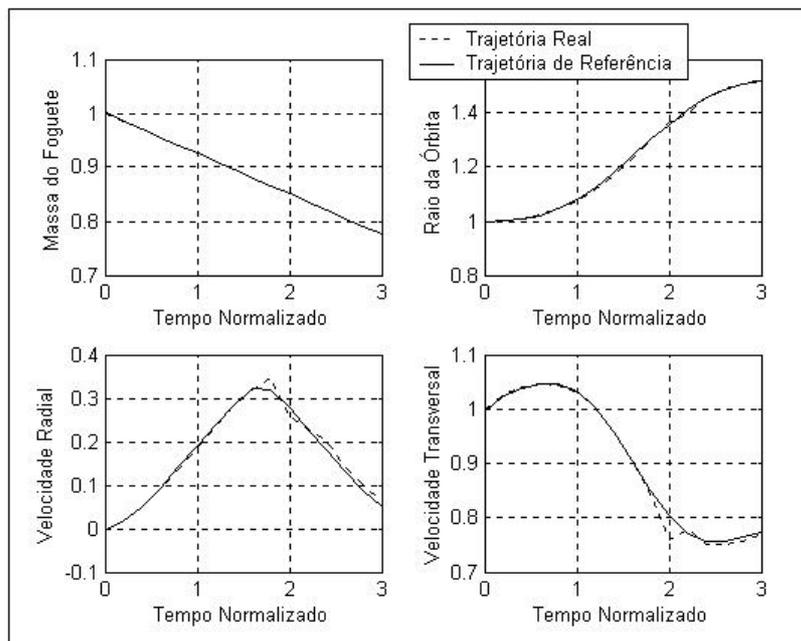


FIGURA 09- Simulação de controle preditivo da transferência Terra/Marte ($\Delta t = 0.2$).

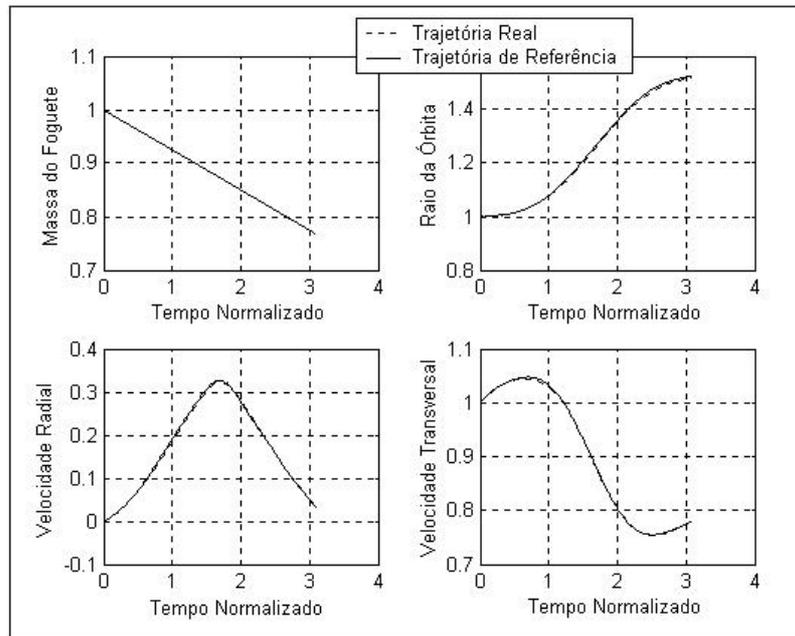


FIGURA 10 - Simulação de controle preditivo da transferência Terra/Marte ($\Delta t = 0.01$).

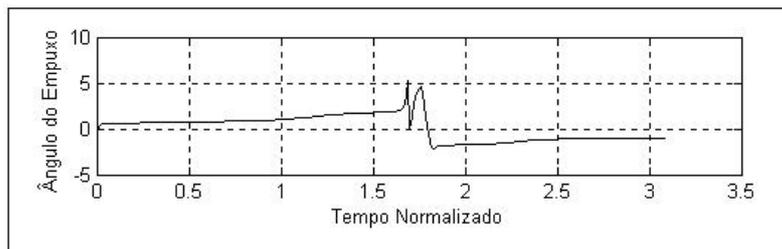


FIGURA 11 – Controle estimado pela estrutura de controle preditivo($\Delta t = 0.01$).

Os resultados da Figura 10 mostram que para um passo de integração e estimação menor que o utilizado na simulação da Figura 9 e no valor de 0.01 unidades de tempo normalizado, consegue-se rastrear com precisão a trajetória de referência. . A figura 11 mostra que os controles estimados para este caso são suaves.

7 - COMENTÁRIOS FINAIS E CONCLUSÕES.

Neste trabalho uma nova abordagem para obter os modelos discretos de sistemas dinâmicos, onde um modelo interno é necessário, foi apresentada e testada, em um esquema de controle preditivo. A possibilidade de se utilizar integradores numéricos de sistemas de equações diferenciais como modelos internos foi considerada. Foi mostrado que a estrutura destes integradores numéricos pode ser explorada para obter modelos neurais discretos para frente onde a rede neural tem que somente aprender e aproximar as funções de derivadas que são simplesmente de natureza algébrica estática. Algumas conclusões, já percebidas anteriormente pelos proponentes da abordagem, puderam ser confirmadas:

- a) é uma tarefa mais simples treinar uma rede neural com arquitetura *Feedforward* para aprender uma função algébrica e estática do que treiná-la para aprender o modelo dinâmico discreto como explicado na seção 2 através da equação (2);
- b) a arquitetura da rede neural resultante é mais simples no que concerne ao número de camadas e número de neurônios uma vez que ela não tem que aprender a lei da dinâmica, mas apenas a função de derivadas;
- c) a utilização do integrador numérico para sistemas de equações diferenciais ordinárias, como um modelo aproximado de tempo discreto, não destrói a característica de processamento paralelo, uma vez que o algoritmo de integração numérica somente envolve cálculos e avaliações de combinações lineares da rede neural treinada;
- d) a flexibilidade de se poder variar o tamanho do passo e a ordem do integrador numérico pode ser utilizada para controlar a precisão desejada para a saída de resposta do sistema, desde que se treine a rede neural com uma tolerância compatível com aquela que se deseja no integrador;
- e) quando os tempos dinâmicos de resposta não são muitos pequenos e o modelo matemático de equações diferenciais ordinárias avaliado é razoavelmente bom, então a estrutura do integrador numérico pode ser diretamente utilizada como um

modelo interno discreto;

- f) mesmo em situações onde um modelo matemático teórico não pode ser avaliado e há somente pares de informação dinâmica de entrada/saída do sistema obtidos experimentalmente, ainda assim, a estrutura do integrador numérico com uma rede *Feedforward* no lugar da função de derivadas pode ser treinada para obter um modelo interno discreto em esquemas de controle;
- g) finalmente, é importante considerar que a utilização de uma rede neural em um modelo discreto do sistema dinâmico naturalmente permitirá a implementação de esquemas de controle adaptativos, devido a capacidade de aprendizado das redes neurais.

Pode-se portanto concluir existirem três maneiras distintas de se resolver o problema de programação não-linear exigido pela estrutura de controle preditivo:

- a) utilizar somente a rede neural para aproximar a dinâmica do sistema utilizando o critério de entradas atrasadas;
- b) utilizar somente o integrador numérico como modelo discretizado da dinâmica do sistema;
- c) utilizar em conjunto o integrador numérico e uma rede neural que fará o papel de representar as funções de derivadas para representar o sistema dinâmico.

Utilizar somente a rede neural leva a uma rede mais robusta e, portanto mais difícil de ser treinada, além de não possibilitar a variação do passo de integração. Utilizar somente a estrutura do integrador numérico com a função de derivadas dificulta adaptar o modelo teórico do sistema dinâmico com o real da planta. Utilizar em conjunto a estrutura do integrador numérico com a função de derivadas aproximada pela rede neural permite construir uma rede menos robusta e, portanto de mais fácil treinamento, além de se poder projetar, neste caso, um modelo dinâmico com *passo de integração variável* e que pode ser adaptado em tempo real com relação ao sistema original da planta.

A maior desvantagem do último método é que o cálculo das derivadas parciais torna-se mais complexo e complicado, sendo que para cada tipo de integrador a ser utilizado ter-

se-á uma expressão diferente para as derivadas parciais e, em geral, estas expressões serão tão mais complexas e difíceis de serem obtidas quanto maior a ordem do integrador.

O uso do integrador neural de múltiplos passos de Adams Bashfort como modelo interno, no esquema de controle preditivo baseado no algoritmo de filtragem de Kalman, aplicado ao problema de transferência de órbita Terra Marte, levou a resultados em que:

- a) a interpretação estocástica dos erros deu mais realismo ao tratamento do problema e facilitou o ajuste das matrizes peso no funcional do problema de controle preditivo;
- b) a versão com processamento paralelo do algoritmo de Kalman para cálculo das ações de controle teve um desempenho equivalente ao do correspondente algoritmo de Kalman para treinamento de redes neurais, como esperado, já que são algoritmos completamente similares, aplicados para resolver problemas numericamente equivalentes de estimação de parâmetros;
- c) apenas um passo à frente foi suficiente como horizonte de controle; esta característica, juntamente com a eficiência e desempenho dos algoritmos de processamento paralelo de Kalman, combinada com a capacidade atual de processamento a bordo, garante a viabilidade de aplicações adaptativas em tempo real.

REFERÊNCIAS BIBLIOGRÁFICAS

- Carrara, V. *Redes neurais aplicadas ao controle de atitude de satélites com geometria variável*. 1997. 202 p. INPE-6384-TDI/603. Tese (Doutorado em Ciências Espaciais) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 1997.
- Chen, S.; Billings, S. A. Neural networks for nonlinear dynamic system modelling and identification. *Int. J. Control*, v. 56, n. 2, p. 319-346, 1992.
- Clarke, D.W.; Mohtadi, C.; Tuffs, P. S. Generalized predictive control-part I. The basic algorithm. *The Journal of IFAC the International Federation of Automatic Control. Automatica*, v. 23, n. 2, p. 137-148, Mar 1987a.
- Clarke, D.W.; Mohtadi, C.; Tuffs, P. S. Generalized predictive control-part II. Extensions and interpretations. *The Journal of IFAC the International Federation of Automatic Control. Automatica*, v. 23, n. 2, p. 149-160, Mar 1987b.
- Cybenko, G. *Continuous valued networks with two hidden layers are sufficient*. Local: Tufts University, Department of Computer Science, Tufts University, 1988, Technical Report.
- Henrici, P. *Elements of Numerical Analysis*. New York: John Wiley & Sons, 1964.
- Huang, D.; Van Cauwenberghe, A. R. Neural-network-based multiple feedback long-range predictive control. *Neurocomputing*, v. 18, n. 1-3, p. 127-139, Jan. 1998.
- Hornik, K.; Stinchcombe, M.; White, H. Multilayer feedforward networks are universal approximators. *Neural Networks*, v. 2, n. 5, p. 359-366, 1989.
- Hunt, K. J.; Sbarbaro, D. Neural networks for nonlinear internal model control. *IEE Proceedings – D*, v. 138, n. 5, p. 431-438, Sep. 1991.
- Hunt, K. J.; Sbarbaro, D.; Zbikowski, R.; Gawthrop, P. J. Neural networks for control systems – A survey. *Automatica*, v. 28, n. 6, p. 1083-1112, Nov. 1992.
- Lambert, J. D. *Computational methods in ordinary differential equations*. New York: John Wiley & Sons, 1973.
- Liu, G. P.; Kadiramanathan, V.; Billings, S. A. Predictive control for non-linear systems using neural networks. *Int. J. Control*, v. 71, n. 6, p. 1119-1132, 1998.
- Mills, P. M.; Zomaya, A. Y.; Tadó, M. O. Adaptive model-based control using neural networks. *Int. J. Control*, v. 60, n. 6, p. 1163-1192, 1994.

Narendra, K. S.; Parthasarathy, K. Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, v. 1, n. 1, p. 4-27, Mar 1990.
Rao, K. R. *A Review on numerical methods for inicial value problems*. São José dos Campos: INPE, Feb. 1984, 173 p. (INPE-3011-RPI/088).

Rios Neto, A. Stochastic optimal linear parameter estimation and neural nets training in systems modeling. *RBCM – J. of the Braz. Soc. Mechanical Sciences*, v. 19, n. 2, p. 138-146, 1997.

Rios Neto, A. Design of a Kalman filtering based neural predictive control method. In: CONGRESSO BRASILEIRO DE AUTOMÁTICA – CBA, 13., 2000, Florianópolis, código 543/00. *Anais ...* Florianópolis: UFSC (Universidade Federal de Santa Catarina), 2000, p. 2130-2134. 1 CD-ROM.

Rios Neto, A. Dynamic systems numerical integrators in neural control schemes. In: CONGRESSO BRASILEIRO DE REDES NEURAIAS, 5., 2001, Rio de Janeiro, RJ, Brasil. *Anais ...* Rio de Janeiro: Conselho Nacional de Redes Neurais, 2001, p. 85-88. 1 CD-ROM.

Sage A. P. *Optimum systems control*. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1968.

Vidyasagar, M. *Nonlinear systems analysis*. New Jersey: Prentice-Hall, Networks Series, Electrical Engineering Series, 1978.

Wang, Y.-J.; Lin, C.-T. Runge-Kutta neural network for identification of dynamical systems in high accuracy. *IEEE Transactions On Neural Networks*, v. 9, n. 2, p. 294-307, Mar 1998b.

Zurada, J. M. *Introduction to Artificial Neural System*. St. Paul, MN, USA: West Pub. Co., 1992.

APÊNDICE A

O CÁLCULO DA RETRO-PROPAGAÇÃO

A Figura 12 apresenta um esquema gráfico simplificado de uma camada genérica k em uma rede *Feedforward* (Carrara, 1997). A representação matemática formal para o vetor de saída y^k da camada k será então dada pela seguinte expressão:

$$y_i^k = f^k \left(\sum_{j=1}^{n_{k-1}} w_{ij}^k \cdot y_j^{k-1} - b_i^k \right) \quad p/ \quad i = 1, 2, \dots, n_k \quad (31)$$

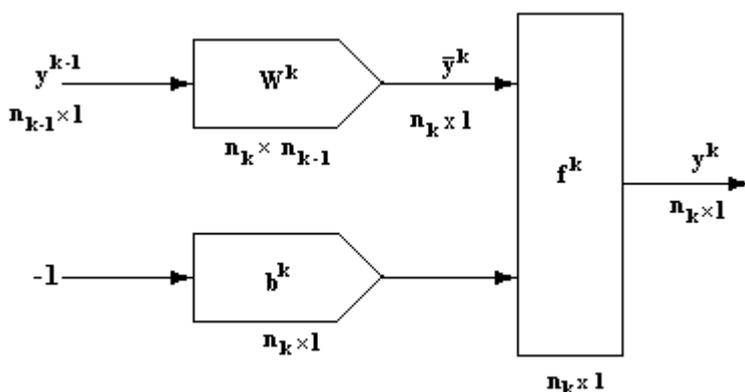


FIGURA 12 – Entradas e saídas de uma camada genérica k em rede *Feedforward*.

O processamento analítico dado pela equação (31) pode ser colocado na forma matricial, demonstrando a característica de paralelismo apresentada pelas redes *feedforward*. Desta forma, sendo W^k a matriz de pesos da camada k ($k=1, 2, \dots, L$), o vetor de saída desta camada será:

$$y^k = f^k \left(W^k \cdot y^{k-1} \right) = f^k(\bar{y}^k) \quad (32.a)$$

Onde,

$$y^k = [f^k(\bar{y}_1^k) \dots f^k(\bar{y}_{n_k}^k) - 1]^T \quad (32.b)$$

$$\mathbf{W}^k = \begin{bmatrix} w_{11}^k & \cdots & w_{1n_{k-1}}^k & b_1^k \\ w_{21}^k & \cdots & w_{2n_{k-1}}^k & b_2^k \\ \vdots & \ddots & \vdots & \vdots \\ w_{n_k 1}^k & \cdots & w_{n_k n_{k-1}}^k & b_{n_k}^k \end{bmatrix}_{n_k, n_{k-1}+1} \quad (32.c)$$

Deve ser observado que na equação (32.c) foi acrescentado o vetor linha \mathbf{b}^k à matriz de pesos \mathbf{W}^k , mas desde que não se esqueça de acrescentar o valor unitário negativo na última linha do vetor \mathbf{y}^k representado pela equação (32.b). O cálculo da Retropropagação pode ser determinado (Carrara, 1997) da seguinte forma:

$$\frac{\partial y^j}{\partial \bar{y}^k} = \frac{\partial y^j}{\partial \bar{y}^{k+1}} \cdot \mathbf{W}^{k+1} \cdot \mathbf{I}'_{f'(\bar{y}^k)} \quad \text{para } k=L-1, L-2, \dots, 1 \quad (33.a)$$

Onde,

$$\frac{\partial y^L}{\partial \bar{y}^L} = \mathbf{I}'_{f'(\bar{y}^L)} \quad \text{para } j=k=L \text{ (camada de saída da rede)} \quad (33.b)$$

$$\mathbf{I}'_{f'(\bar{y}^1)} = \begin{bmatrix} f'(\bar{y}_1^L) & 0 & 0 & 0 \\ 0 & f'(\bar{y}_2^L) & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & f'(\bar{y}_{n_k}^L) \end{bmatrix} \quad (33.c)$$

$$\frac{\partial \hat{f}[y(t_0), u(t_0)]}{\partial \mathbf{u}(t_0)} \Big|_{\bar{\mathbf{u}}(t_0, i)} = \mathbf{A}_u \quad (33.d)$$

$$\frac{\partial y^L}{\partial \bar{y}^1} = \left[\frac{\partial \hat{f}[y(t_0), y(t_0)]}{\partial \mathbf{y}(t_0)} ; \frac{\partial \hat{f}[y(t_0), u(t_0)]}{\partial \mathbf{u}(t_0)} \right] = [\mathbf{A}_y : \mathbf{A}_u] \quad (33.e)$$

$$\frac{\partial \mathbf{y}^L}{\partial \bar{\mathbf{y}}^1} = \frac{\partial \hat{\mathbf{y}}^L}{\partial \bar{\mathbf{y}}^1} = \underbrace{\begin{bmatrix} \frac{\partial \hat{y}_1^L}{\partial \bar{y}_1^1} & \frac{\partial \hat{y}_1^L}{\partial \bar{y}_2^1} & \dots & \frac{\partial \hat{y}_1^L}{\partial \bar{y}_{n_y}^1} \\ \frac{\partial \hat{y}_2^L}{\partial \bar{y}_1^1} & \frac{\partial \hat{y}_2^L}{\partial \bar{y}_2^1} & \dots & \frac{\partial \hat{y}_2^L}{\partial \bar{y}_{n_y}^1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \hat{y}_{n_y}^L}{\partial \bar{y}_1^1} & \frac{\partial \hat{y}_{n_y}^L}{\partial \bar{y}_2^1} & \dots & \frac{\partial \hat{y}_{n_y}^L}{\partial \bar{y}_{n_y}^1} \end{bmatrix}}_{\mathbf{A}_y = \frac{\partial \hat{\mathbf{f}}[\mathbf{y}(t_0), \mathbf{u}(t_0)]}{\partial \mathbf{y}(t_0)}}} : \underbrace{\begin{bmatrix} \frac{\partial \hat{y}_1^L}{\partial \bar{y}_{n_y+1}^1} & \frac{\partial \hat{y}_1^L}{\partial \bar{y}_{n_y+2}^1} & \dots & \frac{\partial \hat{y}_1^L}{\partial \bar{y}_{n_y+n_u}^1} \\ \frac{\partial \hat{y}_2^L}{\partial \bar{y}_{n_y+1}^1} & \frac{\partial \hat{y}_2^L}{\partial \bar{y}_{n_y+2}^1} & \dots & \frac{\partial \hat{y}_2^L}{\partial \bar{y}_{n_y+n_u}^1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \hat{y}_{n_y}^L}{\partial \bar{y}_{n_y+1}^1} & \frac{\partial \hat{y}_{n_y}^L}{\partial \bar{y}_{n_y+2}^1} & \dots & \frac{\partial \hat{y}_{n_y}^L}{\partial \bar{y}_{n_y+n_u}^1} \end{bmatrix}}_{\mathbf{A}_u = \frac{\partial \hat{\mathbf{f}}[\mathbf{y}(t_0), \mathbf{u}(t_0)]}{\partial \mathbf{u}(t_0)}}$$

(33.f)

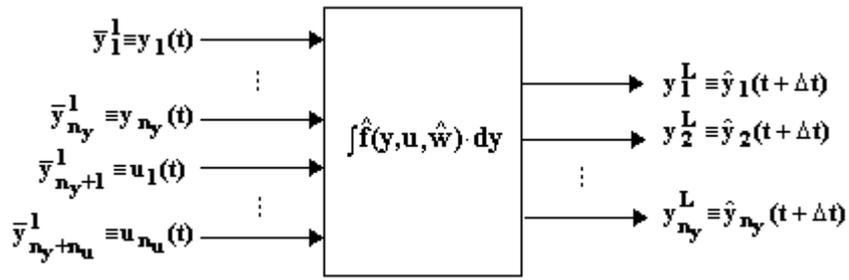


FIGURA 13 – Esquema Simplificado de uma Estrutura de Integração Neural.

A Figura 13 ilustra o significado da simbologia utilizada para representar as derivadas da retro-propagação como representada nas equações (33.e), (33.f) e (23). As equações de (33.a) à (33.f) são utilizadas para calcular a matriz de derivadas parciais presente na equação (23) e referente ao algoritmo computacional da estrutura de controle preditivo neural detalhado na seção 5.

APÊNDICE B

INTEGRADORES NUMÉRICOS

Neste apêndice são sugeridos alguns algoritmos computacionais utilizados para resolver *numericamente* equações diferenciais ordinárias ou sistema de equações diferenciais. Admita $\mathbf{f}=\mathbf{f}(\mathbf{y},t)$ ser uma função real de duas variáveis reais definidas para $\mathbf{a} \leq t \leq \mathbf{b}$ onde \mathbf{a} e \mathbf{b} são finitos, e para todo valor real de \mathbf{y} . A equação

$$\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y}, t) \quad (35)$$

é chamada uma equação diferencial ordinária de primeira ordem; ela simboliza o seguinte problema: achar uma função $\mathbf{y} = \mathbf{y}(t)$, contínua e diferenciável para $t \in [\mathbf{a}, \mathbf{b}]$, tal que:

$$\dot{\mathbf{y}}(t) = \mathbf{f}(\mathbf{y}(t), t) \quad (36)$$

para todo $t \in [\mathbf{a}, \mathbf{b}]$. A função \mathbf{y} com esta propriedade é chamada a solução da equação diferencial (35). Em geral uma equação diferencial pode ter muitas soluções. Para evitar este tipo de problema deve-se especificar o valor da função $\mathbf{y}(t)$ em um dado ponto, por exemplo, $\mathbf{y}(t) = \boldsymbol{\eta}$ em $t=\mathbf{a}$. Dada então, a condição inicial do sistema dinâmico pode-se demonstrar matematicamente que a solução será única quando o sistema for autônomo, ou seja, quando $\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y})$. Em muitos problemas de aplicação pode-se deparar não com uma simples equação diferencial, mas com um sistema de \mathbf{m} equações simultâneas de primeira ordem com variáveis dependentes ${}^1\mathbf{y}, {}^2\mathbf{y}, \dots, {}^m\mathbf{y}$. Se cada uma dessas variáveis satisfaz uma dada condição inicial para o mesmo valor \mathbf{a} de t , então nós temos um problema de valor inicial para um *sistema* de primeira ordem, que se pode escrever:

$$\begin{aligned} {}^1\dot{\mathbf{y}} &= {}^1\mathbf{f}(t, {}^1\mathbf{y}, {}^2\mathbf{y}, \dots, {}^m\mathbf{y}), & {}^1\mathbf{y}(\mathbf{a}) &= {}^1\boldsymbol{\eta} \\ {}^2\dot{\mathbf{y}} &= {}^2\mathbf{f}(t, {}^1\mathbf{y}, {}^2\mathbf{y}, \dots, {}^m\mathbf{y}), & {}^2\mathbf{y}(\mathbf{a}) &= {}^2\boldsymbol{\eta} \\ & \vdots & & \vdots \\ {}^m\dot{\mathbf{y}} &= {}^m\mathbf{f}(t, {}^1\mathbf{y}, {}^2\mathbf{y}, \dots, {}^m\mathbf{y}), & {}^m\mathbf{y}(\mathbf{a}) &= {}^m\boldsymbol{\eta} \end{aligned} \quad (37)$$

Introduzindo a notação vetorial:

$$\bar{y} = [{}^1y, {}^2y, \dots, {}^m y]^T \quad (38.a)$$

$$\bar{f} = [{}^1f, {}^2f, \dots, {}^m f]^T = \bar{f}(t, \bar{y}) \quad (38.b)$$

$$\bar{\eta} = [{}^1n, {}^2n, \dots, {}^m n]^T \quad (38.c)$$

pode-se então escrever o problema de valor inicial (37) na forma:

$$\dot{\bar{y}} = \bar{f}(t, \bar{y}), \quad \bar{y}(a) = \bar{\eta} \quad (39)$$

Pode-se demonstrar que uma equação diferencial ordinária de segunda ordem pode ser transformada num sistema de duas equações diferenciais de primeira ordem, uma equação de terceira ordem em um sistema de duas equações diferenciais de segunda ordem e assim por diante. Desta forma, qualquer sistema de alta ordem pode ser transformado num sistema de primeira ordem e recair na situação dada pelas equações (37) ou (39).

Garantida as unicidades da equação diferencial (35) para sistemas autônomos e do sistema de equações diferenciais de primeira ordem (36) ou (39), então pode-se afirmar que nem sempre eles possuirão uma solução analítica, principalmente se forem do tipo não-linear. Nestes casos somente soluções numéricas poderão ser obtidas. Desta forma, basicamente há três classes de métodos numéricos para resolver um sistema de equações diferenciais de primeira ordem ordinário:

- a) método de passo simples, que computa o valor de y em apenas um passo $\Delta t (= h)$ na variável independente t por utilizar somente a informação vinda das funções feitas entre t e $t + \Delta t$ (ponto de vista local);
- b) método de múltiplos passos, que computa um passo por utilizar a informação ao longo de vários passos precedentes (ponto de vista global);

c) método de extrapolação, que primeiramente utiliza-se de um método de passo simples para integrar a equação diferencial para vários tamanhos de passos h_j , e então, uma interpolação polinomial é construída através dos valores calculados para gerar os valores desejados de y .

A aproximação geral para uma solução numérica de uma equação diferencial envolve uma seqüência de, igualmente espaçados ou não, pontos discretos $t_0, t_1, t_2 \dots$. Em cada ponto t_n , a solução $y(t_n)$ é aproximada por um número y_n que é computado de valores mais anteriores. Isto é, em cada passo, uma aproximação é feita utilizando os valores das aproximações realizadas anteriormente. Em outras palavras, um erro em um dado ponto propaga-se com progresso da integração. Assim, $h_n = t_{n+1} - t_n$ deve ser ajustado e o erro em $y(t_n)$ deve ser *estimado*.

Existem basicamente quatro tipos de erros a serem considerados na utilização de integradores numéricos: erro de truncamento local, erro global, erro local e os *round-off* erros, este último relacionado com a precisão alcançada com os computadores digitais.

Métodos de Passo Simples

O método de passo simples é um algoritmo que descreve uma técnica numérica para calcular a aproximação para a solução em y_{n+1} , em termos do valor em um passo anterior (y_n). Os métodos de passo simples são principalmente divididos em duas categorias: métodos em séries de Taylor e métodos de Runge-Kutta.

Os Métodos de Runge-Kutta: é muito mais fácil ter um polinômio que concorda com y e \dot{y} em t_n e com \dot{y} (ou f) em vários outros argumentos próximos de t_n do que um polinômio que concorda com y e suas k derivadas em t_n (séries de Taylor). Os métodos de R-K são baseados neste princípio. As equações de segunda ordem do método de Runge-Kutta são enunciadas a seguir sem demonstração:

$$k_1 = h \cdot f(y_n, t_n) \quad (40.a)$$

$$k_2 = h \cdot f(t_n + h/2, y_n + k_1/2) \quad (40.b)$$

$$y_{n+1} \cong y_n + k_2 \quad (40.c)$$

As equações (40.a), (40.b) e (40.c) formam, desta maneira, um esquema para obter x_{n+1} conhecidos os valores de t_n e y_n , para uma precisão $O(h^2)$, e por envolver $f(y,t)$ em dois pontos distintos (y_n, t_n) e $(t_n + h/2, y_n + k_1/2)$. A equação descrita é o processo de Runge-Kutta de segunda ordem. O método geral de Runge-Kutta é definido como (Rao, 1984):

$$y_{n+1} = y_n + h \cdot \Phi(t_n, y_n, h) \quad (41.a)$$

Onde,

$$\Phi(t_n, y_n, h) = \sum_{r=1}^R C_r \cdot k_r \quad (41.b)$$

$$k_1 = f(y_n, t_n) \quad (41.c)$$

$$k_r = f(t_n + h \cdot a_r, y_n + h \cdot \sum_{s=1}^{r-1} b_{rs} \cdot k_s) \quad (41.d)$$

$$a_r = \sum_{s=1}^{r-1} b_{rs} \quad r = 2, 3, \dots, R \quad (41.e)$$

Os parâmetros C_r , a_r e b_{rs} são determinados por comparar os termos das potências sucessivas de h entre (41.a) e o algoritmo de Taylor. Deve ser observado que, em geral, o número de equações não-lineares das condições que surgem de (41.a) à (41.e) em comparação ao algoritmo de Taylor é diferente do número de parâmetros a serem determinados, para uma dada ordem pré-fixada. Com isto, há muitos conjuntos de equações de R-K de mesma ordem propostos por vários autores. Por exemplo, a fórmula de Heun (e.g., Lambert, 1973) de terceira ordem ou a fórmula de Kutta (e.g., Rao, 1984) também de terceira ordem. Por fim, são listadas a seguir as equações de Runge-Kutta de ordem quatro (e.g., Lambert, 1973, Henrici, 1964) que é uma das mais

utilizadas nos algoritmos numéricos para a resolução de sistemas de equações diferenciais ordinárias devido a sua alta precisão e elegante simplicidade:

$$y_{n+1} = y_n + \frac{h}{6} \cdot (k_1 + 2 \cdot k_2 + 2 \cdot k_3 + k_4) \quad (42.a)$$

$$k_1 = f(y_n, t_n) \quad (42.b)$$

$$k_2 = f\left(t_n + \frac{1}{2} \cdot h, y_n + \frac{1}{2} \cdot h \cdot k_1\right) \quad (42.c)$$

$$k_3 = f\left(t_n + \frac{1}{2} \cdot h, y_n + \frac{1}{2} \cdot h \cdot k_2\right) \quad (42.d)$$

$$k_4 = f(t_n + h, y_n + h \cdot k_3) \quad (42.e)$$

Há muitos esquemas de integração que dependem de valores em mais do que um ponto prévio para estimar o próximo ponto sobre a curva de solução. Entre estes métodos estão os de Adams-Bashforth, Adams-Moulton, regra de Simpson e Preditivo-Corretor.

Existem basicamente três maneiras de se derivar essas expressões numéricas (e.g., Lambert, 1973): através de uma expansão em série de Taylor, por integração numérica e através de interpolações.

Neste apêndice serão considerados somente os integradores obtidos por integração numérica, pois esta é a metodologia mais empregada. Suponha que para valores igualmente espaçados t_1, t_2, \dots, t_n obteve-se as soluções numéricas y_1, y_2, \dots, y_n por algum método de passo simples e agora deseja-se obter a solução em y_{n+1} utilizando-se um ou mais valores anteriores de $y(t)$. A maneira de se resolver este problema através da *integração numérica* é representar a equação (35) por:

$$y(t_{n+p}) = y(t_n) + \int_{t_n}^{t_{n+p}} f(y(t), t) \cdot dt \quad (43)$$

onde, $f(y(t),t)$ é a equação de derivadas da expressão (35).

A seguir, sem entrar no mérito das demonstrações, são enunciadas algumas expressões bastante interessantes com relação aos métodos numéricos de múltiplos passos para resolver sistemas dinâmicos de primeira ordem. As principais expressões correspondentes aos algoritmos de Adams-Moulton (Vidyasagar, 1978) em termos de $y(t_{n+1})$ com ordens de 1 a 4 são, respectivamente, dadas por:

$$y(t_{n+1}) = y(t_n) + h \cdot f_n \quad (44.a)$$

$$y(t_{n+1}) = y(t_n) + \frac{h}{2} \cdot [f_{n+1} + f_n] \quad (44.b)$$

$$y(t_{n+1}) = y(t_n) + \frac{h}{12} \cdot [5 \cdot f_{n+1} + 8 \cdot f_n - f_{n-1}] \quad (44.c)$$

$$y(t_{n+1}) = y(t_n) + \frac{h}{24} \cdot [9 \cdot f_{n+1} + 19 \cdot f_n - 5 \cdot f_{n-1} + f_{n-2}] \quad (44.d)$$

Ainda em (Vidyasagar, 1978) podem ser encontradas as fórmulas dos algoritmos de Adams-Bashforth de ordens de 1 a 4 como listadas abaixo:

$$y(t_{n+1}) = y(t_n) + h \cdot f_n \quad (45.a)$$

$$y(t_{n+1}) = y(t_n) + \frac{h}{2} \cdot [3 \cdot f_n - f_{n-1}] \quad (45.b)$$

$$y(t_{n+1}) = y(t_n) + \frac{h}{12} \cdot [21 \cdot f_n - 16 \cdot f_{n-1} + 5 \cdot f_{n-2}] \quad (45.c)$$

$$y(t_{n+1}) = y(t_n) + \frac{h}{24} \cdot [55 \cdot f_n - 59 \cdot f_{n-1} + 37 \cdot f_{n-2} - 9 \cdot f_{n-3}] \quad (45.d)$$

Embora a equação $\dot{y} = f(y)$ não contenha explicitamente as entradas do controle u como uma variável independente, esta equação ainda pode ser diretamente aplicada para sistemas de controle em *Feedback* (Li et al, 1991). A razão disso é que a equação

$\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y})$ pode representar a dinâmica do sistema de controle em *Feedback*, com a entrada do controle sendo uma função do estado \mathbf{y} , e portanto, não necessitando-se da variável \mathbf{u} na equação diferencial em questão. Especificamente, se a planta dinâmica é dada por,

$$\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y}, \mathbf{u}, \mathbf{t}) \quad (46)$$

Alguma lei de controle pode ser selecionada como,

$$\mathbf{u} = \mathbf{g}(\mathbf{y}, \mathbf{t}) \quad (47)$$

Assim, o sistema dinâmico final será dado por:

$$\dot{\mathbf{y}} = \mathbf{f}[\mathbf{y}, \mathbf{g}(\mathbf{y}, \mathbf{t})] \quad (48)$$

A equação (48) pode naturalmente ser escrita na forma $\dot{\mathbf{y}} = \mathbf{h}(\mathbf{y}, \mathbf{t})$ recaindo-se novamente na situação original dada pela equação (35).