Proceedings of V Brazilian Conference on Neural Networks - V Congresso Brasileiro de Redes Neurais
pp. 85-88, April, 02-05-2001, Rio de Janeiro, RJ, Brazil

Dynamic Systems Numerical Integrators in Neural Control Schemes

Atair Rios Neto

CDT: Centro de Desenvolvimento de Tecnologia e Recurso Humanos
Escola de Engenharia Industrial-EEI
12242-800 São José dos Campos, SP
e-mail: atairrn@uol.com.br

Abstract

In this short paper, a not yet explored way of getting dynamic systems discrete forward models, to be used as internal models in control schemes, is proposed. A mixed heuristic and theoretical approach is taken to propose and explicitly show how to use dynamic systems ordinary differential equations (ODE) numerical integrators in control schemes where a discrete forward internal model is needed. Use is made of the structure of numerical integrators algorithms to make it possible to get a neural feedforward model to approximate the dynamic system by only learning the derivative function in the system ordinary differential equations model. It is then illustrated how to use this kind of discrete forward model in a predictive control scheme and in an internal model control scheme where a least control action criterion is used. Independently of any numerical experiment, conclusions are drawn concerning the peculiar and advantageous aspects of the proposed method.

1. Introdution

In control schemes employing feedforward neural networks, the usual approach has been that of using a discrete nonlinear input-output NARMAX type of model to approximate the dynamic system, to then train the neural network to learn this model (e.g.: Chen and Billings (1992); Liu et al (1998)). There are at least two difficulties with this approach. The first is the adjustment of the order of the input-output model, in terms of the number of delayed responses and of delayed control actions to account for. The second is the situation of usually having to deal with too many inputs, and thus too many parameters, in the training of the neural network.

An opportunity yet to be explored is offered by the existing knowledge and results to deal with the numerical integration of ordinary differential equations(ODE) (e.g., Stoer and Bulirsch (1980)). The computer simulation model provided by an ODE numerical integrator is a discrete forward model of the dynamic system which by itself can be used as an internal model in control schemes. These numerical integrators have characteristics which are quite relevant for a dynamical system model to be used in control, since they allow: (i) parallel processing, component by component of the dynamic system state; (ii) the demanded prediction local accuracy to be adjusted by available methods of automatically varying the order or the step size of the integrator(e.g., Fehlberg (1968); Prothero (1980)); and (iii) the estimation of accumulated global prediction errors to be also made (e.g., Rios Neto and Kondapalli (1990)).

If the structure of the numerical integrator model is used, it is possible to have a feedforward neural network to approximate the derivative function in the differential equations mathematical model of the dynamic system. With this approach, the difficulty with too many inputs in the training of the neural network is alleviated, since it is only necessary to learn an algebraic and static function, and the inputs are occurrences of the state and control variables in their envelope of variation.

In what follows, in Section 2, the basic idea of taking a feedforward neural network to learn the derivative function of a dynamic system, to then use the structure of an ODE numerical integrator to get a discrete forward internal model, is proposed. In Sections 3 and 4, it is shown how to use this numerical integrator based internal model in two control schemes: a predictive one and a least control action internal model one. In Section 5, conclusions about expected use and results are drawn.

2. Proposed Approach Basic Idea

Consider a dynamic system with state vector $x \in R^n$ and control vector $u \in R^m$. Suppose that an artificial neural network model is available as an approximation to the derivative function $f(x,u)$ coming from physical laws governing this dynamic system:

$$\dot{x} = f(x,u) \cong \hat{f}(x,u,\hat{w}) \qquad (1)$$

That is, $\hat{f}(x,u,\hat{w})$ with the learned weights $\hat{w}$ represents a neural network (see e.g., Zurada (1992)) which as an universal approximator (Hornik et al (1989)) can model and approximate the derivative function $f(x,u)$.

Consider now an ordinary differential equation (ODE) numerical integrator (e.g., Stoer and Bulirsch (1980)) to get a discrete approximation of the system of Eq.(1):

$$x(t+\Delta t) \cong f_n(x(t), x(t-\Delta t), \ldots, x(t-n_o\Delta t); u(t), \ldots, u(t-n_o\Delta t); \Delta t; \hat{w}) \quad (2)$$

where the right hand side discrete approximation is a known function of $\hat{f}(x,u,\hat{w})$, once a choice of the numerical integrator is done; n_o is related to the order of the approximation; if its value is greater than zero, one has the situation where a finite difference type of integrator is used (for example, an Adams-Bashforth method); if it is zero, a single step type of integrator is used (for example, a Runge-Kutta method); and Δt, the step size, is assumed sufficiently small to assure *u(t)* constant along the discretization interval.

The numerical integrator in Eq.(2) can be used recursively as an approximate discrete predictive model of the dynamic system of Eq.(1) in internal model control schemes, and the resulting numerical algorithm can be processed in parallel for each component of the state of the dynamic system. The error in each step can be controlled by varying step size and or the order of numerical integrator.

3. Neural Control Schemes

3.1 Predictive Control

In this scheme, the problem is to determine a smooth and reference trajectory tracking control, by minimizing a predictive quadratic index of performance , which is usually of the type (see, e.g.: Hunt et al (1992); Su and McAvoy (1993)):

$$J = [\sum_{j=1}^{n}[y_r(t_j) - y_n(t_j)]^T r_y^{-1}(t_j)[y_r(t_j) - y_n(t_j)] + \sum_{j=0}^{n-1}[u(t_j) - u(t_{j-1})]^T r_u^{-1}(t_j)[u(t_j) - u(t_{j-1})]]/2 \quad (3)$$

where $t_j = t + j\Delta t$; $y_r(t_j)$ is the reference response; n defines the horizon over which the tracking errors and control increments are considered; $r_y(t_j), r_u(t_j)$ are positive definite weight matrices; $y_n(t_j)$ is the approximated output of the dynamic system of Eq. (1), which can be formally represented by:

$$y_n(t_j) \cong g(x(t_j)) \cong g(f_n(x(t_{j-1}), \ldots, x(t_{j-1-n_o}); u(t_{j-1}), \ldots, u(t_{j-1-n_o}); \Delta t; \hat{w})) = g_n(x(t_{j-1}), \ldots, x(t_{j-1-n_o}); u(t_{j-1}), \ldots, u(t_{j-1-n_o}); \hat{w}) \quad (4)$$

where the discrete approximation given by the numerical integrator (Eq.(2)) was used , with the ODE derivative function approximated by a trained feedforward neural network with estimated weights $\hat{w}$. The possibility of adjusting the level of approximation of the numerical integrator and of the neural network guarantees the necessary prediction accuracy of $y_n(t_j)$ along the predictive horizon.

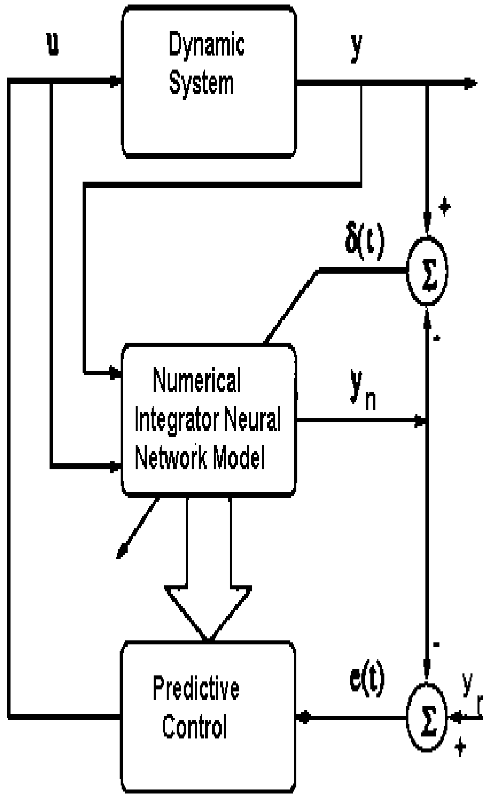

Figure 2: Predictive Control Scheme

The solution of the nonlinear programming problem of Eq.(3), by any chosen method will involve the need of calculating an approximation of the gradient of the output of the dynamic system, to get a linearized approximation in each search step:

$$y_n(t_j) = \bar{y}_n(t_j) + \sum_{k=k_0}^{j-1}[\partial y_n(t_j) / \partial u(t_k)]_{\{\bar{u}(t_k)\}}[u(t_k) - \bar{u}(t_k)] \quad (5)$$

where the over bar is to indicate the nominal last step values, around which the expansion was done; $k_0 = max[0, (j-1-2n_o)]$ since $y_n(t_j)$ is also a function of $u(t_{j-2}), \ldots, u(t_{j-n_o+1})$ through $x_n(t_{j-1}), \ldots, x_n(t_{j-n_o})$; and the partial derivatives are calculated using the chain rule to account for the composed function situation explicit in Eq.(4), including the backpropagation rule (see, e.g., Chandran (1994)) in the feedforward neural network that approximates the derivative function of the dynamic system.

3.2 Least Action Internal Model

In the Internal Model Neural Control (IMNC) scheme (Hunt and Sbarbaro (1991)), besides the internal forward model of the plant, it is needed to have a neural network for the inverse of the plant, as the controller. In what follows, a specialized model is adopted for the identification of the inverse of the plant dynamics.

In a usual IMNC method, the controller neural network is trained by, for example, minimizing the functional:

$$J = 1/2\sum_{j=1}^{L}(y_s(j) - y_c(j,w_c))^T R_y^{-1}(j)(y_s - y_c(j,w_c)) \quad (6)$$

where, $y_s(j), j = 1,2,...L$ is a training sample of values of the desired specialized response; $y_c(j,w_c)$ is the controlled response, at time j, and the result of the control action $u(j-1,w_c)$, at time *j-1*, on the plant.

In a more elaborated approach, instead of the functional of Eq.(6) one that weights previous knowledge of the weights w_c and that constraints least control actions can be considered, as follows:

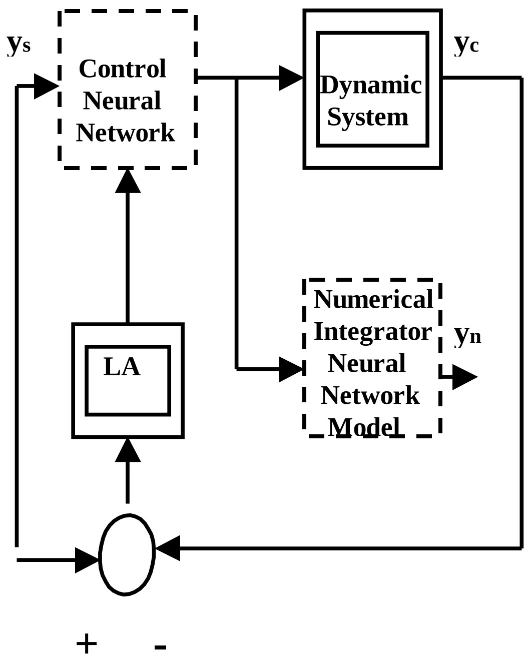

Figure 1: Specialized Controller Training

$$\begin{aligned} J = 1/2[&(w_c - \overline{w}_c)^T \overline{P}_w^{-1}(w_c - \overline{w}_c) + \\ &\sum_{j=1}^{L}(y_s(j) - y_c(j,w_c))^T R_y^{-1}(j)(y_s(j) - y_c(j,w_c)) + \\ &(u(j-1,w_c))^T R_u^{-1}(u(j-1,w_c))] \end{aligned} \quad (7)$$

where $\overline{P}_w^{-1}$, R_y^{-1} and R_u^{-1} are appropriate weight matrices.

To calculate the gradient of $y_c(j,w_c)$ with respect to the controller neural network weights w_c, one has to consider the controller neural network in series with the plant forward model $y_n(t_j)$. The chain rule is used to first calculate the approximate gradient of the controlled response with respect to $u(j-1,w_c)$, to then use the backpropagation rule to calculate the gradient of this control with respect to the controller neural network weights.

Notice that the training minimizing the functional of Eq.(7) is in batch and intended to be done off line. For sequential, on line training, only one prediction pattern $y_s(j)$, is considered at a given time t, and thus $L=1$ is taken in Eq.(7).

4 Numerical Integrator Totally Based Schemes

Notice that instead of using a neural network to provide a discrete forward model for the dynamics, one could directly use $f(x,u)$ in the numerical integrator algorithm to get the discrete forward model approximating the mathematical model of the dynamic system (Eq.(1)).

In the case of the predictive control scheme, this numerical integrator model can be directly used to calculate the partial derivatives in Eq.(5). In the case of the internal model control scheme, the numerical integrator is taken as an approximate plant forward model, in series with the controller neural network, allowing the use of the chain rule to calculate the gradient of the controlled response.

5. Conclusions

A new approach to get dynamic systems discrete forward models, to be used in control schemes where an internal model is needed, was proposed. The possibility of using ODE numerical integrators, as such internal models, was made explicit.

It was shown that the structure of these numerical integrators can be exploited to get neural discrete forward models where the neural network has only to learn and approximate the algebraic and static derivative function in the dynamic system ODE.

Independently of any numerical experiment, the following conclusions can be drawn, concerning the expected use and results:

(i) it is a simpler task to train a feedforward neural network to learn the algebraic , static function of the dynamic system ODE derivatives (where the inputs are samples of state and control variables), than to train it to learn a NARMAX type of discrete model (where the inputs are samples of delayed responses and controls (e.g., Carrara et al (1998)));

(ii) the neural network certainly will be simpler, in terms of the necessary number of layers and number of neurons, since it does not have to learn the dynamic law , but only the derivative function;

(iii) the use of the ODE numerical integrator, as a discrete time approximate model , does not destroy the parallel processing characteristic , since the numerical integrator algorithm will only involve calculations and evaluations of linear combinations of the trained neural network;

(iv) the existing knowledge about step size and order adjustment in numerical integration can be used to control expected prediction accuracy;

(v) when the dynamic time responses are not too small, and a reasonably good ODE mathematical model is available, the numerical integrator can directly be used as a discrete internal model;

(vi) even in the situation where an ODE mathematical model is not available, as long as dynamic system input output pairs are available to be used as training information, the structure of the numerical integrator with a feedforward network in place of the derivative function can be trained to get a discrete internal model in control schemes;

(vii) finally, it is important to consider that the use of a neural network in the dynamic system discrete model will naturally allow the implementation of adaptive control schemes, due to the learning capacity of the neural network.

Acknowledgement

The project, under which this research was done, has been partially supported by a FAPESP grant. The author is grateful to Jaime Augusto da Silva, for the help in the paper first revision.

References

[1] Carrara, V., Varotto, S. E. C., Rios Neto, A.: "Satellite Attitude Control Using Multilayer Perceptron Neural Networks", American Astronautical Society - AAS, Advances in the Astronautical Sciences, Volume 100, Part 1, AAS 98-345, pp. 565 – 579, 1998

[2] Chandran, P. S., " Comments on Comparative Analisys of Backpropagation and the Extended Kalman Filter for Training Multilayer Perceptron", *IEEE Transactions on Pattern Analysis and Machine Inteligence*, 16(8), pp. 862-863, 1994.

[3] Chen, S. and Billings, S. A., " Neural Networks for Nonlinear Dynamic System Modelling and Identification", *Int. J.Control*,56(2), pp. 319-346, 1992.

[4] Felhlberg, E., "Classical Fifth-,Sixth-,Seventh-, and Height-Order Runge Kutta Formulas With Step Size Control", Technical Report R-287, NASA, Washington, DC, 1968.

[5] Hornik, K., Stinchcombe, M. and White, H., "Multilayer Feedforward Networks are Universal Approximators", *Neural Networks*, 2(5), pp. 359-366, 1989.

[6] Hunt, K. J., Sbarbaro, D., Zbikowski, R. and Gawthrop, P. J., "Neural Networks for Control Systems- A Survey", *Automatica*, 28(6), pp.1083-1112, 1992.

[7] Hunt, K.J. and Sbarbaro, D., "Neural Networks for Nonlinear Internal Model Control", *Proc. IEE Pt. D*, 138, pp. 431-438, 1991.

[8] Liu, G. P., Kadirkamanathan, V. and Billings, S. A., "Predictive Control for Non-linear Systems Using Neural Networks", Int. Journal of Control, vol. 71, no. 6, pp. 1119-1132, 1998.

[9] Prothero, A.,"Estimating the Accuracy of Numerical Solution to Ordinary Differential Equations", in : Gladwell, I. and Sayers, D.K., Eds., *Computational Techniques for Ordinary Differential Equations*, Academic Press, London, 1980.

[10] Rios Neto, A.; Rama Rao, K. "A Stochastic Approach to Global Error Estimation in ODE Multistep Numerical Integration"; Journal of Computational and Applied Mathematics, Vol. 30, pp. 257 - 28l, l990.

[11] Stoer, J. and Bulirsch, R., *Introduction to Numerical Analisys*, Springer Verlag, N.Y. 1980.

[12] Su, H. T. and McAvoy, T. J., "Neural Model Predictive Control of Nonlinear Chemical Procesesses", in *Proc. of the 1993 International Symposium on Inteligent Control*,Chicago, U.S.A., pp. 358-363, 1993.

[13] Zurada, J.M., *Introduction to Artificial Neural Systems*, West Pub. Co., 1992.