Stability of a neural predictive controller scheme on a neural model

Luther, Jim Benjamin; Sørensen, Paul Haase

Published in:
Proceedings of the International Joint Conference on Neural Networks

Link to article, DOI:
10.1109/IJCNN.1999.832708

Publication date:
2009

Document Version
Publisher's PDF, also known as Version of record

Link back to DTU Orbit

Citation (APA):
Luther, J. B., & Sørensen, P. H. (2009). Stability of a neural predictive controller scheme on a neural model. In Proceedings of the International Joint Conference on Neural Networks (Vol. 3, pp. 2087-2091). DOI: 10.1109/IJCNN.1999.832708

Stability of a Neural Predictive Controller Scheme On a Neural Model.

Jim Benjamin Luther
Email: jlu@iau.dtu.dk
Paul Haase Sørensen
Email: phs@iau.dtu.dk

Department of automation (IAU)
Building 326 Technical university of Denmark (DTU)
DK-2800 Lyngby
Denmark

8th June 1999

Abstract

In recent papers [4],[7],[8],[11], [12],[14] different forms of neural network based predictive controllers have been proposed.The main emphasis in these papers is on the implementation aspects of the controller, i.e. the development of a robust optimization algorithm for the controller, which will be able to perform in real time. However, the stability issue has not been addressed specifically for these controllers. On the other hand a number of results concerning the stability of receding horizon controllers on a non-linear system exist [2], [10] and [9].

In this paper we present a proof of stability for a predictive controller controlling a neural network model. The resulting controller is tested on a non-linear pneumatic servo system.

1 Introduction

Predictive control of non-linear systems has become increasingly interesting because good stability(see [2], [10], [9]) and robustness(see [6] and [1]) properties can be proven. These proofs are relatively general in the sense that only general properties of the non-linear system are required. The stability proofs are given both in continuous [5], [6] and [3] and discrete-time [2], [1],[10] and [9]. In particular the discrete-time stability guarantee is interesting in connection with real-time implementations.
In a number of papers the use of receding horizon control with neural network models or controllers has been investigated [3],[10] [9] and [13]. In particular, the paper [13] develop a fully implementable algorithm for a generalized predictive controller GPC, with a feed forward neural network as a model of the non-linear plant([13]). Since we are dealing with non-linear systems, the minimization of the GPC cost function has to be performed numerically. In [13] the numerical optimization is developed in detail and convergence of the numerical algorithm is emphasized. However, the stability of the resulting controller cannot be guaranteed. The present paper addresses this problem, and gives a proof of asymptotic stability.

In [13] the system model is a non-linear input-output model but for convenience we assume here that the system to be controlled is non-linear, time-invariant and described by a discrete time state equation of the form:

$$x_{k+1} = f(x_k, u_k) \ , \ \forall x_k \in X \ , \ \forall u_k \in U \qquad (1)$$

where $f \in \mathcal{C}$ (the set of continuous vector-functions on $X \times U$), $f(0,0) = 0$, X and U are compact sets on $\mathcal{R}^n$ and $\mathcal{R}^m$ respectively, and they include the origin.

If the function f is formed from a feed forward neural network with continuous activation functions the continuity properties of f are guaranteed.

The neuro GPC cost function in [13] is

$$\begin{aligned} J(k,u(k)) = \sum_{i=N_1}^{N_2} [r(k+i) - \hat{y}(k+i)]^2 \\ + \rho \sum_{i=0}^{N_u} \Delta u(k+i)^2 \end{aligned} \qquad (2)$$

0-7803-5529-6/99/$10.00 ©1999 IEEE

It is seen to be a quadratic cost function of the system error (reference minus model output) summed over the time interval from N_1 to N_2 in the future, Also the cost function includes the future control signal changes ($\Delta u_k = u_k - u_{k-1}$) over the control horizon (N_u). Paper [13] gives a detailed account of the implementation of the optimizing controller. This cost function makes it possible to reduce the number of control signals(N_u) to compute independably of the prediction horizon. Moreover the prediction does not have to begin at sample k, but can be moved beyond the systems time delay.

2 The proof

The proof of stability is based on rewriting the cost function and system model in such a form that the general proof of stability in [9] can be utilized. This can be achieved by only minor changes in the assumptions.

2.1 The cost function

First a few changes to (2) are neccesary. The outputs in (2) are replaced with the system states. This is done for the sake of simplicity. This does not impair the generality of the scheme. One can always make a state transformation to make the origin an equilibrium point of the new system. Let for instance the system

$$\bar{x}_{k+1} = \bar{f}(\bar{x}_k, \bar{u}_k) \tag{3}$$

have an equilibrium point ($\bar{f}(\tilde{x}, \tilde{u}) = \tilde{x}$) in $x = \tilde{x}, u = \tilde{u}$, then a new system that satisfies the assumptions can be found by defining a new state and a new control signal.

$$\begin{aligned} x_k &= \bar{x_k} - \tilde{x} \\ u_k &= \bar{u_k} - \tilde{u} \\ x_{k+1} &= f(x_k, u_k) = \bar{f}(x_k + \tilde{x}, u_k + \tilde{u}) - \tilde{x} \end{aligned} \tag{4}$$

The cost function (2) is supplemented with a final state penalty. This is necessary to ensure stability. Also we generalize the cost function by replacing the quadratic terms with positive functions, hereby obtaining the cost function:

$$\begin{aligned} J(x_k, N_1, N_2, N_u) = \sum_{i=N_1}^{N_2-1} h_x(x_{k+i}) + \sum_{i=0}^{N_u-1} h_u(u_{k+i}) \\ + a\|x_{k+N_2}\|_P^2 \end{aligned} \tag{5}$$

Where $h_x, h_u \in \mathcal{C}$, $h_x(0) = h_u(0) = 0$, $h_x(x) \geq 0, \forall x \in X_0 - \{0\}$ $h_u(u) \geq 0 \, \forall u \in U - \{0\}$, $a \in \mathcal{R}_+, \|x\|_P^2 = x^T P x, P > 0$

As in [13], the control signal u_{k+i} follows some predetermined sequence for $N_u \leq i \leq N_2 - d$, where d is the system time delay.

2.2 Assumptions

The assumptions needed to guarantee stability are basically the same as in [9] , but there are a few changes to accommodate the extra h-function $h_u(u)$.

Assumption 1: The linearized system
$(A, B) = \left(\frac{\partial f}{\partial x} \Big|_{x=0,u=0}, \frac{\partial f}{\partial u} \Big|_{x=0,u=0} \right)$ is stabilizable.

Assumption 2:

$$\begin{aligned} r_x(\|x\|^2) \leq h_x(x) \leq s_x(\|x\|^2) \\ r_u(\|u\|^2) \leq h_u(u) \leq s_u(\|u\|^2) \,, \; \forall x \in X \,, \; \forall u \in U \end{aligned} \tag{6}$$

where $r_x, s_x, r_u, s_u \in \mathcal{C}$ and are strictly increasing
$r_x(0) = s_x(0) = 0$, $r_u(0) = s_u(0) = 0$

Assumption 3: There exists a compact set $X_0 \subseteq X$, which includes the origin, with the property that there exists a control horizon $M \geq 1$ such that there exists a sequence of admissible control vectors
$\{u_k, \ldots, u_{k+M-1}, u_{k+M}, \ldots, u_{k+N_2-d}\}$ that yield an admissible state trajectory $\{x_k, \ldots, x_{N_2}\}$ ending in the origin. Here $\{u_{k+M}, \ldots, u_{k+N_2-d}\}$ is a predetermined sequence (Usually $u_{k+M+i} = u_{k+M+i-1}$, $0 \leq i \leq N_2 - d$) that u_k is should follow for $k \geq N_u$.

Assumption 4: The optimal control signal u_{k+i}, $0 \leq i \leq N_2$ is continuous with respect to x_{k+1}.

The trajectory in assumption 3, that u_k follows for $k \geq N_u$, is determined by the user of this algorithm. This information is needed by the minimization algorithm used, since the value of the states are used in the cost function for $k \geq N_u$. This trajectory usually reflects the stationary behavior of the system to be controlled.

2.3 Rewriting the costfuntion

Now we have made all the assumptions necessary to make use of the proof in [9] , and here we will write the cost function (5) in same form as in (9).

For given N_1, N_2 and N_u we define

$$\begin{aligned} h(x_{k+1}, u_{k+1}) = &I_x(i, N_1, N_2) h_x(x_{k+1}) \\ &+ I_u(i, N_u) h_u(u_{k+1}) \end{aligned} \tag{7}$$

where

$$I_x(i,N_1,N_2) = \begin{cases} 1 & N_1 \leq i \leq N_2 - 1 \\ 0 & Otherwise \end{cases}$$
$$I_u(i,N_u) = \begin{cases} 1 & 0 \leq i \leq N_u - 1 \\ 0 & Otherwise \end{cases} \tag{8}$$

This would make the cost function in (5) equal to

$$J(x_k, N) = \sum_{i=0}^{N-1} h(x_{k+i}, u_{k+i}) + a||x_{k+N}||_P^2 \tag{9}$$

for $N = N_2$. By the new assumptions.

$$\begin{aligned} h(0,0) &= I_x(i,N_1,N_2)h_x(0) + I_u(i,N_u)h_u(0) \\ &= I_x(i,N_1,N_2)0 + I_u(i,N_u)0 \\ &= 0 \end{aligned} \tag{10}$$

$h(x,u)$ is continuous in x and u since $h_x(x)$ and $h_u(u)$ are.

For all i, (6) gives:

$$\begin{aligned} h(x,u) \leq & I_x(i,N_1,N_2)s_x(||x||^2) \\ & + I_u(i,N_u)s_u(||u||^2) \\ \leq & s_x(||x||^2) + s_u(||u||^2) \\ \leq & s_x(||(x,u)||^2) + s_u(||(x,u)||^2) \\ = & s(||(x,u)||^2) \end{aligned} \tag{11}$$

$s(||(x,u)||^2)$ is continuous since $s_x(||x||^2)$ and $s_u(||u||^2)$ are, and $s(0) = s_x(0) + s_u(0) = 0$.

Noting that the cost $h(x,u) > 0$, $\forall x, u \neq 0$, the assumptions in [9] are fulfilled and the predictive control strategy minimizing the cost function in (5) are guaranteed stable.

2.4 A neural model

The function f in (1) could be a neural network since a neural network is continuous.

$$x_{k+1} = f(x_k, u_k) = NN(x_k, u_k) \tag{12}$$

Where $NN(x_k, u_k)$ is a neural network function trained to describe the system being considered. So if the system can be described by a neural network, this control is a stable and very useful strategy for nonlinear systems with significant time delays.

If one makes use of the cost function (5) then the computational burden can be drastically reduced since the control signal usually becomes constant after a short while, and therefore does not need to be calculated explicitly by the minimization algorithm.

3 Example

In figure 1 and 2 is a simulation of a GPC controller in action on pneumatic servo system. Figure 1 shows the response when using the original cost function as in [13]. Figure 2 shows the response using the cost function (5). Both controllers are tuned as accurately as possible. This to see if there is any significant loss in performance or change in behaviour by adding the final term cost. In both simulations, the following parameters were used. $N_1 = 1, N_2 = 10, N_u = 2, \rho = 0.05$

As is seen in the figures, the performance is almost fully preserved for the present system. This means that the addition of the stabilizing term in the cost function does not significantly degrade the performance fo the system.

4 Conclusion

We have found that the costfunction (5) leads to a stabilizing control signal for a system that can be acurately modelled by a neural network. The system performance was only slightly reduced when using the cost function that guarantees stability.

The next step will be to examine what happens if you have a neural network model that does not model the system accurately.

References

[1] Alamir, M. and Bornard, G. Optimization based stabilizing strategy for nonlinear discrete time systems with unmatched uncertain. *Second International Symposium on Methods and Models in Automation and Robotics*, 1995.

[2] Alamir, M. and Bornard, G. Stability of a Truncated Infinite Constrained Receding Horizon Scheme: the General Discrete Nonlinear Case. *Automatica, No. 9*, 31, 1995.

[3] Mason, J.D. Kambhampati, C., Delgado, A. and Warwick,K. Stable receding horizon control based on recurrent networks. *IEE Proc.-Control Theory Appl., No. 3*, 144, 1997.

[4] Lightbody, G. and Irwin, G. A novel neural internal model control structure. *Proceedings of the American Control Conference, pages 350-354*, 1995.

[5] Michalska, H. Mayne, D.Q. Receding Horizon Control of Nonlinear Systems. *IEEE Transactions on automatic control, No. 7*, 35, 1990.

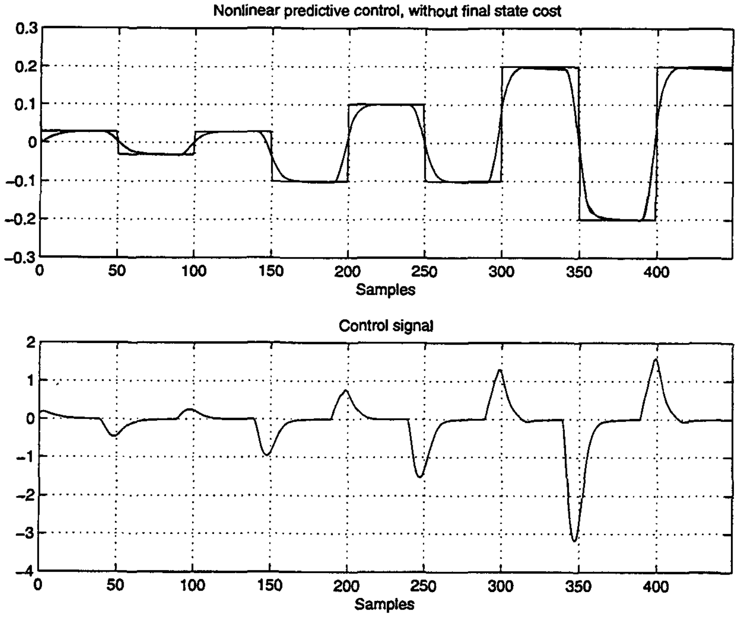

Figure 1: A simulation using the cost function without a final state cost added.

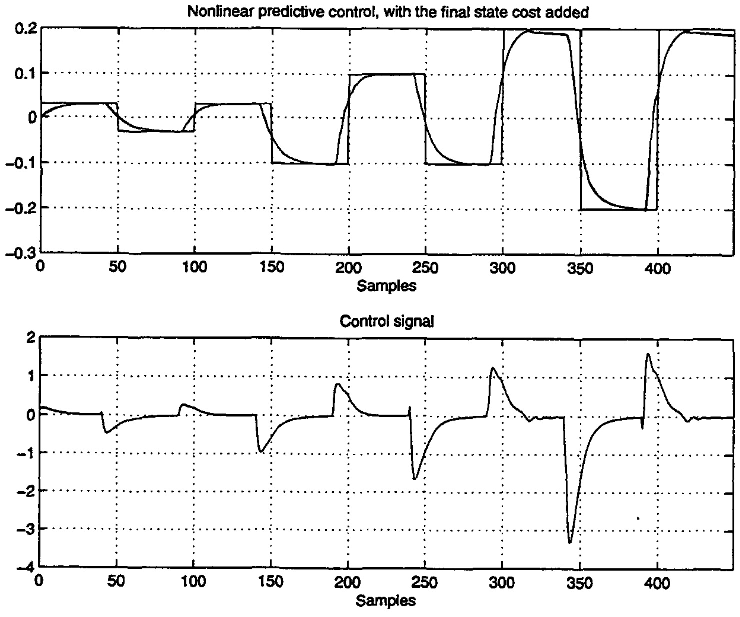

Figure 2: A simulation using the cost function with the final state cost added.

[6] Michalska, H. Mayne, D.Q. Robust Receding Horizon Control of Constrained Nonlinear Systems. *IEEE Transactions on automatic control, No. 11*, 38, 1993.

[7] Nørgaard, M. and Sørensen, P.H. Generalized predictive control of a nonlinear system using neural networks. *Prep. of the 1995 Int. Symposium on Artificial Neural Networks, pages pp. B1-33-40, Hsinchu, Taiwan, ROC*, 1995.

[8] Nørgaard, M., Sørensen, P.H., Poulsen, N., O. Ravn, and Hansen, L. Intelligent predictive control of nonlinear processes using neural networks. *Proc. of the 1996 IEEE Int. Symposium on Intelligent Control (ISIC), pages 374-379, Dearborn, Michigan, USA*, 1996.

[9] Parisini, T., Sanguineti, M., and Zoppoli, R. Nonlinear stabilization by receding-horizon neural regulators. *Int. J. Control, No. 3*, 70, 1998.

[10] Parisini, T. and Zoppoli, R. A Receding-horizon Regulator for Nonlinear Systems and a Neural Approximation. *Automatica, No. 10*, 31, 1995.

[11] Psaltis, D., Sideris, A., and Yamamura,A.A. A multilayer neural network controller. *IEEE Control Systems Magazine, 8(2):17-21*, 1988.

[12] Soloway, D. and Haley, P. Neural/generalized predictive control, a newton-raphson implementation. *Proceedings of the 11th IEEE international Symposium on Intelligent Control, pages 277-282*, 1996.

[13] Sørensen, P.H., Nørgaard, M., Ravn, Ole, and Poulsen, Niels Kjølstad. Implementation of neural network based non-linear predictive control. *Proc. of the int. conf. Neurap 11-13 March 1998, Marseilles, France*, 3, 1998.

[14] Willis, M., Montague, G., Massimo, C., Tham, M., and Morris, A. Artificial neural networks in process estimation and control. *Automatica, 28(6):1181-1187*, 1992.