International Journal of Soft Computing 2 (1): 69-74, 2007

Constrained Nonlinear Neural Model Based Predictive Control Using Genetic Algorithms

Mohamed Boumehraz
Automatic Control Department, University of Biskra,
B. P. 145, Biskra 07000, Algeria

Abstract: Nonlinear Model Based Predictive Control (MBPC) is one of the most powerful techniques in process control, however, two main problems need to be considered; obtaining a suitable nonlinear model and using an efficient optimization procedure. In this study, a neural network is used as a non-linear prediction model of the plant. The optimization routine is based on Genetic Algorithms (GAs). First a neural model of the non-linear system is derived from input-output data. Next, the neural model is used in an MBPC structure where the critical element is the constrained optimization routine which is no convex and thus difficult to solve. A genetic algorithm based approach is proposed to deal with this problem. The efficiency of this approach had been demonstrated with simulation examples.

Key words: Nonlinear system, model, predictive control, neural network, genetic algorithm

INTRODUCTION

Model based predictive control MBPC was developed in the process industries in the 1960's and 70's, based primarily on heuristic ideas and input-output step and impulse response models[1-3]. The basic principle is to solve an open-loop optimal control problem at each time step. The decision variables are a set of future manipulated variables and the objective function is to minimize deviations from a desired trajectory; constraints on manipulated, state and output variables are naturally providing a model update at each time and performing the optimization again[3,4].

The classical MBPC algorithm use linear models of the process to predict its output over the prediction horizon. When no model of the system is available, the classical system identification theory provides possible solutions to the problem, but when the process is non-linear and it is driven over a wide dynamic operating range, the use of linear models becomes impractical and the use of non-linear models becomes a necessity[4].

The use of neural networks for non-linear system modelling has proved to be extremely successful[5,6]. In this study we propose to use neural networks to model non-linear systems in an MBPC structure. Using such nonlinear prediction model, in the predictive control scheme, results in a non-linear and non convex optimization problem which must be solved at each control sample. The optimization problems to be solved on line are generally nonlinear programs without any redeeming features, which imply that convergence to global optimum cannot be assured[2]. Often the nonlinear optimization problem is solved by iterative methods such as Sequential Quadratic Programming (SQP), which is computationally very expensive with no guarantee of convergence to a global optimum. Genetic Algorithms (GAs)[7] are potential methods as optimisation techniques for complex problems. The aim of this study is to use neural networks as models for the plant in an MBPC strategy and to solve the non-linear constrained optimization problem by genetic algorithms.

BASIC ELEMENTS OF MODEL BASED PREDICTIVE CONTROL

MBPC also known as Receding Horizon Control (RHC) is a general methodology for solving control problems in the time domain. It is based on three main concepts[3,4,8,9]:

- Explicit use of a model to predict the process output.
- Computation of a sequence of future control actions by minimizing a given objective function.
- The use of the receding horizon strategy: only the first control action in the sequence is applied, the horizons are moved one sample period towards the future and optimization is repeated.

Because of the optimization approach and the explicit use of the process model, MBPC can realize multivariable

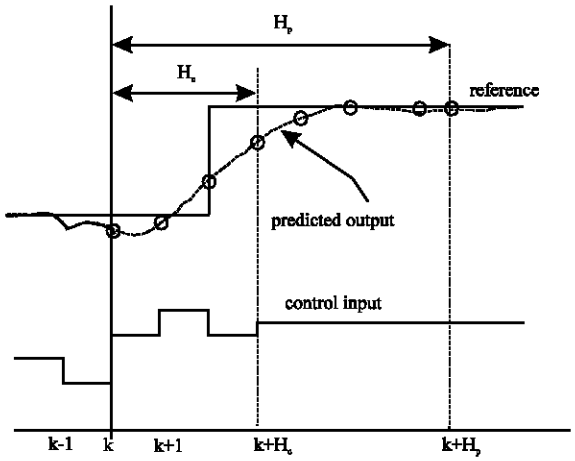

Fig. 1: The basic principle of model based predictive control

optimal control, deals with nonlinear processes and handle constraints efficiently. The three basic elements of MBPC:

- a model which describes the process,
- a goal, defined by an objective function and constraints (optional) and
- an optimization procedure.

The future process outputs are predicted over the prediction horizon H_p using the model of the process: $\hat{y}(k+i)$ for $i=1,\ldots,H_p$. These values depend on the current process state and the future control signal $u(k+i)$ for $I=0,\ldots,H_c-1$, where $H_c \leq H_p$ is the control horizon. The control variable is manipulated only within the control horizon and remains constant afterwards.

Process model: The model must describe the system well and it does not matter what type of model is used to this end: A black-box, a gray-box, or a white-box[9,10]. The process future outputs $\hat{y}(k+i)$ for $I=1,\ldots, H_p$, are predicted over the prediction horizon H_p using a model of the process.

Objective function: The objective function mathematically describes the control goal. In general, good tracking of the reference trajectory is required, with low control energy consumption. These requirements can be expressed by the general form[4]:

$$\begin{aligned} J = &\sum_{i=0}^{H_p}\left[r(k+i)-y(k+i)\right]Q\left[r(k+i)-y(k+i)\right]^T \\ &+\sum_{i=1}^{H_c}u(k+i)Pu^T(k+i)+ \\ &\sum_{i=0}^{H_p}\left[\Delta r(k+i)-\Delta y(k+i)\right]\Delta Q\left[\Delta r(k+i)-\Delta y(k+i)\right]^T \\ &+\sum_{i=1}^{H_c}\Delta u(k+i)\Delta P\Delta u^T(k+i) \end{aligned} \quad (1)$$

Where r(k) is the reference, P, ΔP, Q and ΔQ are positive definite weight matrices. Level and rate constraints of the control input and/or other process variables can be specified as a part of the optimization problem.

In MBPC, Eq. 1 is usually used in combination with input and output constraints:

$$\begin{aligned} u_{min} \leq u \leq u_{max} \\ \Delta u_{min} \leq \Delta u \leq \Delta u_{max} \\ y_{min} \leq y \leq y_{max} \\ \Delta y_{min} \leq \Delta y \leq \Delta y_{max} \end{aligned} \quad (2)$$

Other constraints can be implemented in a straightforward way, e.g. state constraints for state space models[2].

Optimisation: Model predictive control requires an optimization procedure by which a sequence of optimal control signals can be found at each step. Linear MBPC problem with constraints form a convex optimization problem that can be efficiently solved by numerical methods[2]. In the presence of nonlinearities and constraints, a non convex optimization problem must be solved at each sampling period. This hampers the application of nonlinear MBPC to fast systems where iterative optimization techniques cannot be properly used, due to short sampling periods and extensive computation times[9].

Moreover, iterative optimization algorithms, such as the Nelder-Mead method, the multi-step Newton-type algorithm[11], or Sequential Quadratic Programming (SQP)[12], usually converge to local minima, which results in poor solutions of the optimization problem. For efficiency many vendors use heuristic methods, for example, by using dynamic matrices[2].

In this study, a genetic algorithm based approach is used to solve the MBPC constrained optimisation problem.

NONLINEAR MODELLING USING NEURAL NETWORKS

Because of their ability to approximate virtually any arbitrary mapping between two sets of data, neural networks have been extensively studied for their use in the identification of dynamical systems[5,6].

Multilayer feedforward neural networks are the most used neural structures in system modelling. A multilayer feedforward neural network with one hidden layer and linear activation function for the output nodes can be described as[6]:

$$Y = W\sigma(Vu + \theta) \tag{3}$$

Here $u \in \Re^n$ is the input vector and $y \in \Re^r$ is the output vector and the nonlinear element $\sigma(.)$ is taken element wise. The interconnection matrices are $W \in \Re^{rxh}$ for the output layer, $V \in \Re^{hxn}$ for the hidden layer, $\theta \in \Re^h$ is the bias vector (thresholds of hidden neurons) with h the number of hidden neurons. Given a training set of input/output data, the original learning rule is the backpropagation algorithm.

A non-linear dynamic system with sampled input and output data can be expressed as:

$$Y(k) = \phi\begin{pmatrix} Y(k-1),...,Y(k-N_y),U \\ (k-1),...,U(k-N_u) \end{pmatrix} \tag{4}$$

Where Y(k) is the system output vector at time k, ϕ is a nonlinear function, U is the input vector, N_y and N_u are model orders.

The basic idea of non linear modelling with neural networks is to approximate the function ϕ by a neural network.

The input-output measurements are used to determine the appropriate weight values. There are many variations of the backpropagation algorithm. The simplest implementation of backpropagation learning updates the network weights and biases in the direction in which the performance function decreases most rapidly the negative of the gradient. One iteration of this algorithm can be written as:

$$W(k+1) = W(k) - \alpha(k)g(k) \tag{5}$$

Where W(k) is a vector of current weights and biases, g(k) is the current gradient and a(k) is the learning rate. Backpropagation is known by its slow convergence. Several high performance algorithms which can converge faster were proposed. Fletcher-Reeves, Polak-Ribiére, BFGS and Levenberg-Marquardt algorithms can converge from ten to one hundred times faster than the original backpropagation[13].

OPTIMIZATION

Genetic Algorithms (GAs) as an optimization method have been widely applied as an alternative to classical optimization methods. Their ability to find the optimum of functions where classical methods have difficulties (e.g. non derivative functions), is one of the most properties of this technique. In this study, a genetic algorithm is used to solve the MBPC optimization problem. The algorithm is derived from the steady-state GA and utilizes floating point encoding. The fitness function of the optimizer is defined by the objective function of the model predictive control formulation.

Encoding: Every individual $\{p_i ; i = 1,\ldots,N_{pop}\}$ in the population of a genetic algorithm determines a control sequence:

$$p_i = \{u_i(k), u_i(k+1),...,u_i(k+H_c-1)\} \tag{6}$$

the elements of which are represented as floating point numbers. An individual p_i is described by a set of H_c numbers which are selected within the admissible interval $[u_{min}, u_{max}]$ with absolute differences $\{\Delta u_i(k+j); j = 1,\ldots, H_c-1\}$ not exceeding the prescribed value Δu_{max}.

Initialization: In order to provide faster convergence of the genetic algorithm, suitable initialization procedure should be specified. In this study we combine random initialisation with the interevolution steady-state principle:

Randomly initialization: Random control sequences are generated in accordance with the constraints presented in Eq. 2.

Inter-evolution exchange: The best solutions of the last optimization cycle are used in the next period.

Termination conditions: The termination function is used to determine when the optimization loop should be finished. Selection of a fixed number of generations is not very suitable because evolution may converge earlier. Therefore we introduce a new convergence measure to determine the termination condition. Deviations of all signals of the best individual in the population are scanned for the last N_{conv} generations. The termination condition is fulfilled when either the relative maximum

deviation becomes smaller than a prescribed value or the maximum number of generations N_{gen} is exceeded.

Constraints handling: Manipulated Variables (MVs) Constraints are directly handled in the AG reproduction procedure. Each individual p_i is described by a set of H_c numbers which are selected within the admissible interval $[u_{min}, u_{max}]$ with absolute differences $\{\Delta u_i (k+j); j = 1,\ldots, H_c-1\}$ not exceeding the prescribed value Δu_{min} and Δu_{max}. Controlled Variables (CVs) constraints are handled by penalizing infeasible individuals[14]. The fitness function is modified and the violation of constraints is specified by penalties. The modified fitness function for an individual p is evaluated by:

$$\text{eval}(p) = \begin{cases} \dfrac{1}{J(p)+\varepsilon} & \text{no constraint vioaltion} \\ \dfrac{1}{J(p)+\varepsilon} - Q(p) & \text{otherwize} \end{cases} \quad (7)$$

Where J(p) is the function value given by (2) and ϵ is a small positive number to avoid the division by zero and Q(p) is a penalty function corresponding to constraints violation. The value of Q(p) is a function of the amplitude and the time of the constraint violation.

SIMULATION

Example 1: Consider the non-linear discrete system described by the Eq:

$$y(k+1) = \frac{y(k)}{1+y^2(k)} + u(k) \quad (8)$$

A neural model is obtained using input/output data sets generated by random values of u(k)ϵ[-1.0, 1.0]. The model is a feedforward neural network with three layers: one input layer, one hidden layer and one output layer. The activation function of the three hidden units is the sigmoid. The activation function of the output node is linear. The model has two inputs y(k) and u(k) and one output y(k+1).

Levenberg-marquardt algorithm is used to train the neural model using the input output data generated randomly. The structure of the neural model is represented in Fig. 2.

The goal of the predictive control is to generate suitable sequence of actions u(k)ϵ[-1.0, 1.0] so to minimize the objective function given by Eq. 1 where the reference signal is: r(k)=0.5 for k=1,…, 50; r(k)=-0.2 for k=51,…, 100 and r(k) = 0.2 for k=101,…, 200.

The constraints are:

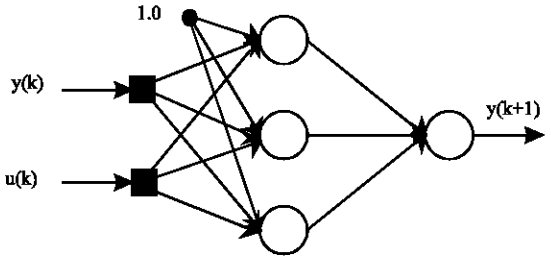

Fig. 2: The neural model

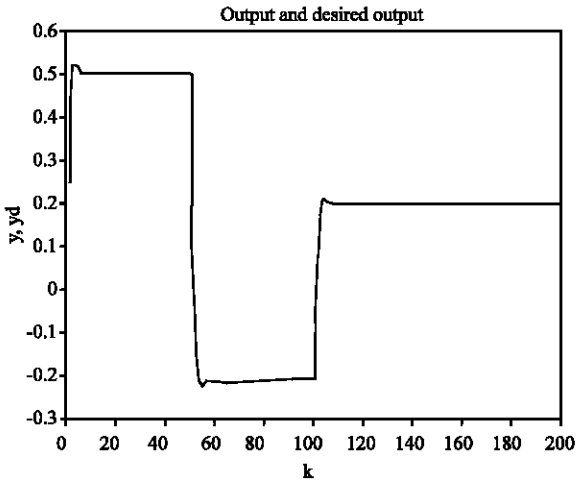

Fig. 3: System output (solid line) and the desired response (dashed line)

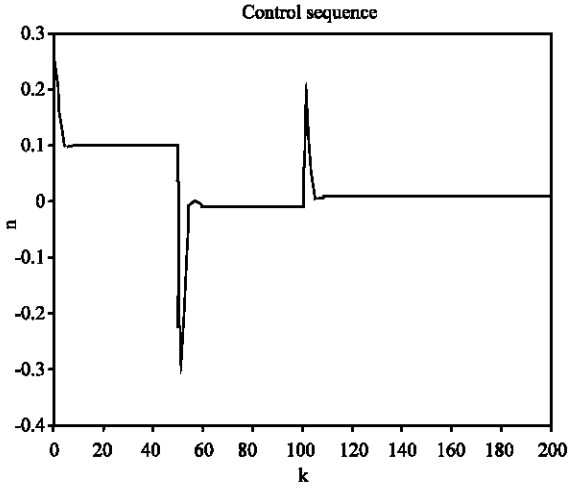

Fig. 4: Control sequence

$$-1.0 \le u(k) \le 1.0$$
$$-1.0 \le y(k) \le 1.0 \quad (9)$$

The prediction horizon is Hp=4 and the control horizon is H_c =2. The weight matrices in Eq. 1 are P =1.0 , Q =1.0, ΔP=0 and ΔQ =0.

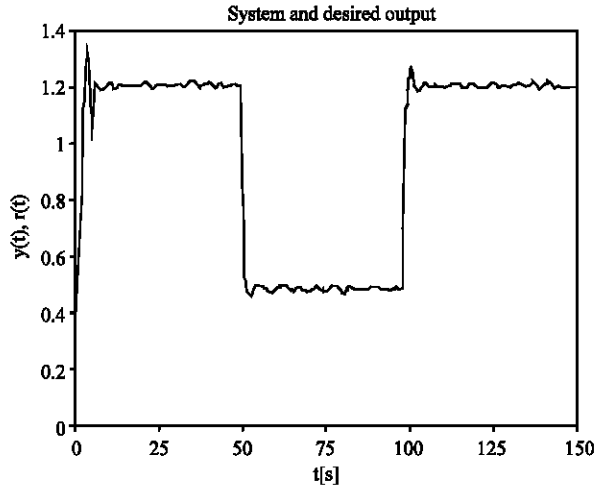

Fig. 5: System output (solid line) and the desired response (dashed line)

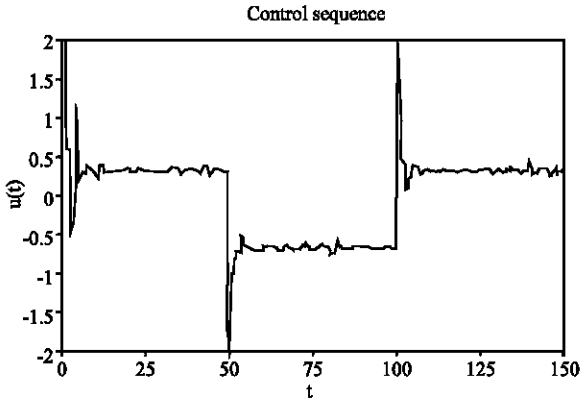

Fig. 6: Control sequence

Figure 3 represents the system output and the reference and the corresponding control input is represented in Fig. 4. As shown in Fig. 3. the process output follow closely the reference

Example 2: Let us consider an exothermic Continuous Stirred Tank Reactor (CSTR) described by the following differential Eq.[15]:

$$\begin{cases} \dfrac{dx_1(t)}{dt} = -x_1(t) + D_a\left(1-x_1(t)\right)e^{\left(\frac{x_2(t)}{1+\frac{x_2(t)}{\phi}}\right)} \\ \dfrac{dx_2(t)}{dt} = -x_2(t) + B.D_a\left(1-x_1(t)\right)e^{\left(\frac{x_2(t)}{1+\frac{x_2(t)}{\phi}}\right)} + \\ \qquad\qquad \beta\left(u(t-\tau)-x_2(t)\right) \end{cases} \tag{10}$$

Where x_1 and x_2 represent dimensionless reactant conversion and temperature, u is the coolant temperature which is used as a manipulated variable. The parameters in the process are: B=8, Da=0.072, ϕ=20, β=0.3 and τ=0.4. The sampling time is 0.1 s. The process output is the reactor temperature x_2. The purpose of the control is to keep the temperature to track the reference set point.

A sequence of random steps with amplitude between [-1, 1] is used to excite the process. Then the produced data are employed for identification. A feedforward neural network with four inputs (y(k-1),y(k-2),u(k-5),u(k-6)), five hidden nodes and one linear output y(k) is constructed to model the process. The neural model was trained by Levenberg-Marquardt algorithm.

The neural model is used in an MBPC structure, the plant response is represented in Fig. 5 and the control sequence in Fig. 6.

CONCLUSION

A non-linear model based predictive control strategy based on neural models and genetic algorithms had been presented. This strategy is a very efficient non-linear model based predictive control approach.

Future study should be done to improve the computation time of the optimiser by choosing special operators to enhance the convergence of the genetic algorithm. A combination with iterative methods may decrease the computational time and avoid the convergence to local minima.

REFERENCES

1. Camacho, E.S. and C. Bordons, 1995. Model predictive control in process industry. Springer, London, U.K.
2. Morari, M. and J.H. Lee, 1999. Model predictive control: past, present and future. Computers and Chemical Engin., 23: 667-682.
3. Richalet, J., 1993. Industrial applications of model based predictive control. Automatica, 29: 1251-1274.
4. Garcia, C.E. *et al.*, 1989. Model predictive control theory and practice: A survey. Automatica, 25: 335-348.
5. Narendra, K.S. and K. Parthasarathy, 1990. Identification and control of dynamical systems using neural networks. IEEE Transactions on Neural Networks, 1: 4-27.
6. Goldberg, D.E., 1989. Genetic algorithms in search, optimisation and machine learning. Addison Wesley, MA, U.S.A.

7. Suykens, J.A.K. *et al*., 1995. Artificial neural networks for model and control of non-linear systems. Kluwer Academic Publishers, Boston, U.S.A.
8. Magni, L., 1998. Non-linear receding horizon control: Theory and applications, PhD thesis, University Degli Studi di Paria, Italy.
9. Roubos, J.A.S. *et al*., 1999. Fuzzy model predictive control using Takagi-Sugeno models. Intl. J. Approximate Reasoning, 22: 3-30.
10. Sjoberg, J., *et al*., 1995. Non-linear black-box modeling in system identification: A unified approach. Automatica, 31: 1691-1724.
11. Oleviera, M.M.C. and L.T. Biegler, 1995. An extension of newton-type algorithms for non-linear process control. Automatica, 31: 281-285.
12. Biegler, L.T., 1997. Advances in Non-linear Programming, Concepts for Process Control. Proceedings of IFAC Adchem Conference, Banff, Canada, pp: 857-598.
13. Demuth, H. and M. Beale, 1998. Neural network toolbox user's guide. MathWorks.
14. Schoenauer, M. and S. Xanthakis, 1993. Constrained GA optimisation. Proceedings of the 5th ICGA, Morgan Kaufman, pp: 573-580.
15. Tan, Y. and A.R. Van Cauwenberghe, 1996. Direct and indirect optimizing neural predictive control J., pp: 37.