Nonlinear Model Predictive Control of Hammerstein and Wiener Models Using Genetic Algorithms

Al-Duwaish H. and Naeem, Wasif

Electrical Engineering Department/King Fahd University of Petroleum and Minerals
Dhahran, 31261, Saudi Arabia
{hduwaish, wasifn}@kfupm.edu.sa
http://www.kfupm.edu.sa

Abstract— **Model Predictive Control or MPC can provide robust control for processes with variable gain and dynamics, multivariable interaction, measured loads and unmeasured disturbances. In this paper a novel approach for the implementation of Nonlinear MPC is proposed using Genetic Algorithms (GAs). The proposed method formulates the MPC as an optimization problem and genetic algorithms is used in the optimization process. Application to two types of Nonlinear models namely Hammerstein and Wiener Models is studied and the simulation results are shown for the case of two chemical processes to demonstrate the performance of the proposed scheme.**

I. Introduction

Model Predictive Control (MPC) refers to a class of algorithms that compute a sequence of manipulated variable adjustments in order to optimize the future behavior of a plant. Originally developed to meet the specialized control needs of power plants and petroleum refineries, MPC technology can now be found in a wide variety of application areas including chemicals, food processing, automotive, aerospace, metallurgy, and pulp and paper [1].
The development of MPC can be traced back to 1978 after the publication of the paper by Richalet et al [2]. Then Cutler and Ramaker from shell oil in 1979, 1980 developed their own independent MPC technology "Dynamic Matrix Control" [3]. For more details, interested readers are referred to [4], [5], [6], [7], [8], [9] and [10].
The process output is predicted by using a model of the process to be controlled. Any model that describes the relationship between the input and the output of the process can be used. Further if the process is subject to disturbances, a disturbance or noise model can be added to the process model. In order to define how well the predicted process output tracks the reference trajectory, a criterion function is used. Typically the criterion is the difference between the predicted process output and the desired reference trajectory. A simple criterion function is

$$J = \sum_{i=1}^{H_p} [\hat{y}(k+i) - w(k+i)]^2 \qquad (1)$$

where $\hat{y}$ is the predicted process output, w is the reference trajectory, and H_p is the prediction horizon or output horizon. The structure of an MPC is shown in figure 1. Now the controller output sequence u_{opt} over the prediction horizon is obtained by minimization of J with respect to u. As a result the future tracking error is minimized. If there is no model mismatch i.e. the model is identical to the process and there are no disturbances and constraints, the process will track the reference trajectory exactly on the sampling instants.

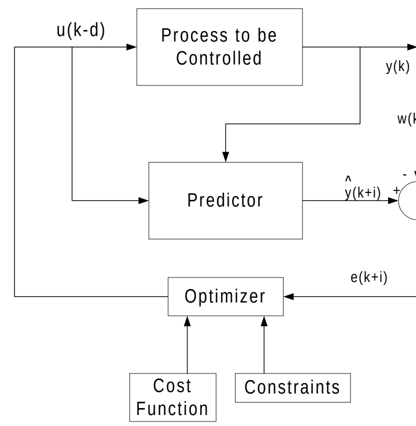

Fig. 1. Structure of MPC

Model Predictive Control algorithm, consists of the following three steps.
1. Explicit use of a model to predict the process output along a future time horizon (Prediction Horizon).
2. Calculation of a control sequence along a future time horizon (Control Horizon), to optimize a performance index.
3. A receding horizon strategy, so that at each instant the horizon is moved towards the future, which involves the application of the first control signal of the sequence calculated at each step. The strategy is illustrated as shown in figure 2.

II. Process Models

MPC allows us to use the detailed knowledge of the process, in the form of a dynamic model, as an aid to controlling that process. Typically, linear models are used for this, despite the fact that essentially all industrial processes exhibit some degree of nonlinear behavior. This is due to the significant increase in complexity of the predictive control problem resulting from the use of a nonlinear model.
Linear MPC employs models which are linearized about the operating point as an aid to predicting the response of the

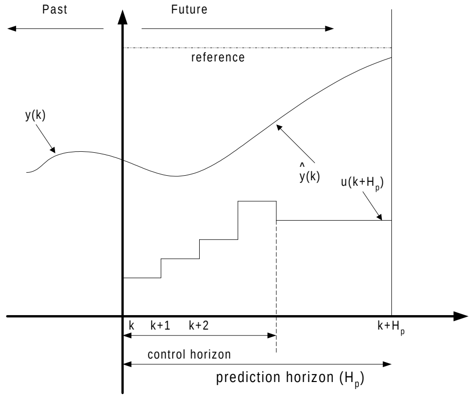

Fig. 2. Predicted output and the corresponding optimum input over a horizon H_p, where $u(k)$, optimum input, $\hat{y}(k)$, predicted output, and $y(k)$, process output

controlled process. This strategy proved to be quite successful even in controlling mildly nonlinear processes. The higher the degree of nonlinearity, however, the greater the level of mismatch between actual process and the representative model, hence resulting in a deterioration of controller performance [11].

For extending linear MPC to the control of nonlinear processes, a model is required that can represent the salient nonlinearities but possibly without the complications associated with general nonlinear models. It is in fulfilling this need that model structures which contain a static nonlinearity in series with a linear dynamic system have been developed. When the Nonlinear element precedes the linear block, it is called the *Hammerstein model* as shown in figure 3.

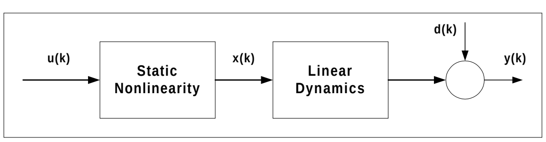

Fig. 3. The Hammerstein model structure

The Hammerstein model is represented by the following equations:

$$y(k) = \frac{B(q^{-1})}{A(q^{-1})}x(k) + d(k) \tag{2}$$

with

$$B(q^{-1}) = b_0 + b_1q^{-1} + \cdots + b_mq^{-m} \tag{3}$$
$$A(q^{-1}) = 1 + a_1q^{-1} + \cdots + a_nq^{-n} \tag{4}$$

the non-measured intermediate variable $x(k)$ is given by

$$x(k) = f(\theta, u(k)) \tag{5}$$

where q^{-1} is the unit delay operator, $u(k)$ is the input, $y(k)$ is the output, $d(k)$ is the measurement noise, (m, n) is the order of the linear part, $f(\cdot)$ is any nonlinear function and θ is a set of parameters describing the nonlinearity.

If the nonlinear element follows linear block, it is called the *Wiener Model* as shown in figure 4.

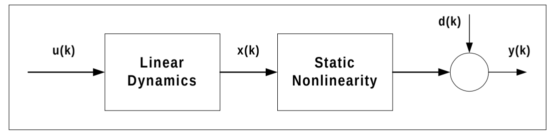

Fig. 4. The Wiener model structure

The Wiener model is represented by the following equations:

$$x(k) = \frac{B(q^{-1})}{A(q^{-1})}u(k) \tag{6}$$

where,

$$B(q^{-1}) = b_0 + b_1q^{-1} + \cdots + b_mq^{-m} \tag{7}$$
$$A(q^{-1}) = 1 + a_1q^{-1} + \cdots + a_nq^{-n} \tag{8}$$

The non-measured intermediate variable $x(k)$ is the input to the static nonlinearity given by

$$y(k) = f(\theta, x(k)) + d(k) \tag{9}$$

III. Overview of Genetic Algorithms

Genetic Algorithms (GAs) inspired by Darwinian theory, is a powerful non-deterministic iterative search heuristic. Genetic Algorithms operate on a population consists of encoded strings, each string represents a solution. Crossover operator is used on these strings to obtain the new solutions which inherits the good and bad properties of their parent solutions. Each solution has a fitness value, solutions having higher fitness values are most likely to survive for the next generation. Mutation operator is applied to produce new characteristics, which are not present in the parent solutions. The whole procedure is repeated until no further improvement is observed or run time exceeds to some threshold [12].

The flowchart of a simple genetic algorithms is presented in figure 5 and the operation of the *GA* is explained as follows.

To start the optimization, *GA* use randomly produced initial solutions. This method is preferred when a priori knowledge about the problem is not available. After randomly generating the initial population of say N solutions, the *GA* use the three genetic operators to yield N new solutions at each iteration. In the selection operation, each solution of the current population is evaluated by it's fitness normally represented by the value of some objective function and individuals with higher fitness value are selected. Different selection methods such as *roulette wheel selection* and *stochastic universal sampling* can be used.

The crossover operator works on pairs of selected solutions with certain crossover rate. The crossover rate is defined as the probability of applying crossover to a pair of selected solutions. There are many ways of defining this operator such

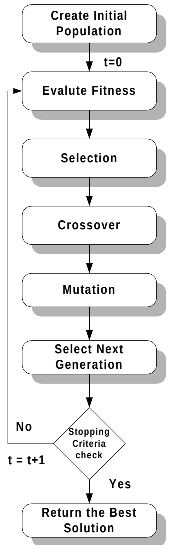

Fig. 5. Flowchart of a Simple Genetic Algorithms

as single point crossover , double point crossover, multi-point crossover etc. For example the single point crossover works on a binary string by determining a point randomly in the two strings and corresponding bits are swapped to generate two new solutions.
Mutation is a random alteration with small probability of the binary value of a string position. This operator prevents GA from being trapped in a local minima. The fitness evaluation unit in GA acts as an interface between the GA and the optimization problem. Information generated by this unit about the quality of different solutions is used by the selection operation in the GA. Next the stopping criteria must be decided. This may be the case when there is no significant improvement in maximum fitness or the maximum allowable time(number of iterations) is passed. At the end of the algorithm, we return the result as the best solution found so far.

IV. Proposed GA-Based MPC Algorithm

The proposed genetic-based control algorithm is shown in figure 6. The GA-based controller uses the process model to search for the control moves, which satisfy the process constraints and optimize some cost function. The process models used in this paper are the Hammerstein and Wiener models. The following steps describe the operation of the proposed GA-based MPC algorithm. At time step k
1. Evaluate process outputs using the process model.
2. Use GAs search to find the optimal control moves which optimize the cost function and satisfy process constraints. This can be accomplished as follows.
(a) generate a set of random possible control moves.
(b) find the corresponding process outputs for all possible control moves using the process models.
(c) Evaluate the fitness of each solution using the cost function and the process constraints.
(d) apply the genetic operators (selection, crossover and mutation)to produce new generation of possible solutions.

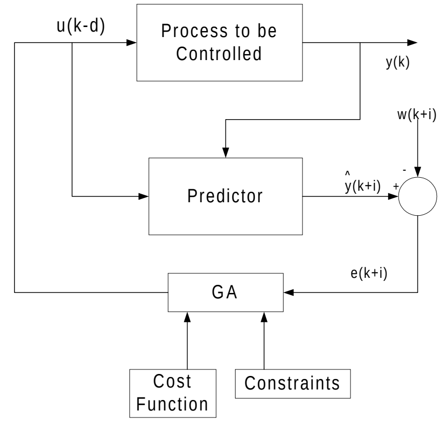

Fig. 6. Proposed GA-based MPC

(e) repeat until predefined number of generations is reached and thus the optimal control moves are determined.
3. Apply the optimal control moves generated in step 2 to the process.
4. Repeat steps 1 to 3 for time step $k+1$

V. Simulation Results

The proposed GA-based **MPC** algorithm was applied to different processes for the case of **SISO** systems and the results are found to be reasonably good. We applied the algorithm to two different nonlinear processes. The first one is the control valve and the second is the heat exchanger. Details are given in the subsequent section.

A. Control Valve

The control valve is an opening with adjustable area. Normally it consists of an actuator, a valve body and a valve plug. The actuator is a device that transforms the control signal to movement of the stem and valve plug. We picked the model from [13], in which the control valve is described by a *Wiener model*. The model is shown here for convenience.

$$x(k) = \frac{0.0616q^{-1} + 0.0543q^{-2}}{1 - 1.5714q^{-1} + 0.6873q^{-2}} u(k) \quad (10)$$

$$y(k) = \frac{x(k)}{\sqrt{0.10 + 0.90x^2(k)}} \quad (11)$$

where $u(k)$ is the control pressure, $x(k)$ is the stem position, and $y(k)$ is the flow through the valve which is the controlled variable.
The nonlinear behavior of the control valve described by equation 11 is shown in figure 9. The input to the process is constrained between $[0, 0.4]$. The prediction and control horizon are taken as 3. The control objective is to keep the process output as close as possible to the reference. The objective function taken is the sum of the square of the future tracking errors between model output and the

reference over the prediction horizon.

$$J = \sum_{i=1}^{H_p} [\hat{y}(k+i) - w(k+i)]^2 \tag{12}$$

where $w(k+i)$ is the reference trajectory.
The parameters of the GA are selected as 50, 0.7 and 0.001 for the population size, crossover probability and mutation probability respectively. The proposed genetic-based control algorithm was simulated and the results are shown in figure 7, which clearly demonstrates the successful performance of the proposed control algorithm. The performance of the proposed algorithm for step changes in the reference(set-point) at different instants of time can be seen in figure 8. We can see that the output tracks the references quite smoothly.

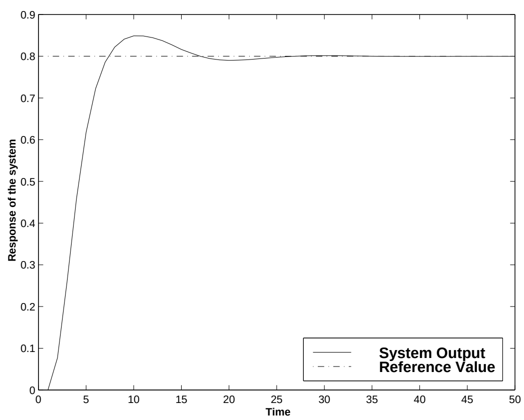

Fig. 7. Application of the proposed algorithm to Control Valve (Single Reference)

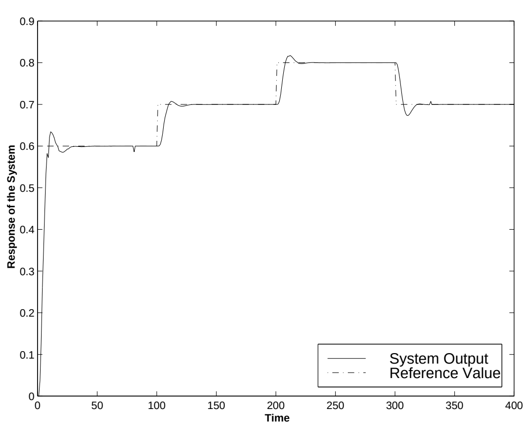

Fig. 8. Application of the proposed algorithm to Control Valve (Multiple references)

B. Heat Exchanger

The model for the heat exchanger problem was obtained from [14], which identifies the heat exchanger as a *Hammerstein Model* and is shown in equation 14.

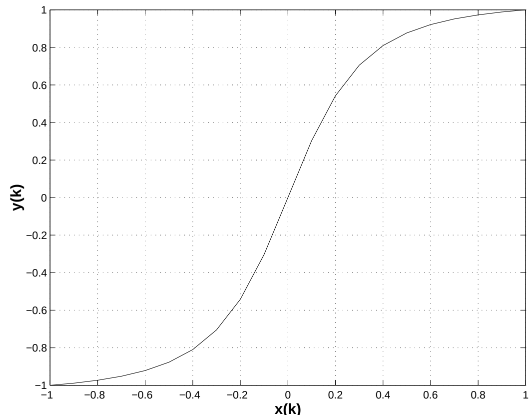

Fig. 9. Nonlinear characteristics of a Control Valve

$$\begin{aligned} x(k) &= -31.549u(k) + 41.732u^2(k) - \\ &\quad 24.201u^3(k) + 68.634u^4(k) \qquad (13) \\ y(k) &= \frac{0.207z^{-1} - 0.1764z^{-2}}{1 - 1.608z^{-1} + 0.6385z^{-2}} x(k) \qquad (14) \end{aligned}$$

where $x(k)$ is the static nonlinearity, $u(k)$ is the process flow rate and $y(k)$ is the process exit temperature.
The input to the process is constrained between $[0, 1]$. The control objective is to keep the process output as close as possible to the reference. The objective function taken is the same as equation 12.
The parameters of the GA are selected as 50, 0.7, 0.005 for the population size, crossover probability and mutation probability respectively. The proposed genetic-based control algorithm was simulated and the results are shown in figure 10, which clearly demonstrates the successful performance of the proposed control algorithm. The nonlinear characteristics of the heat exchanger as indicated by equation 13 are shown in figure 11.

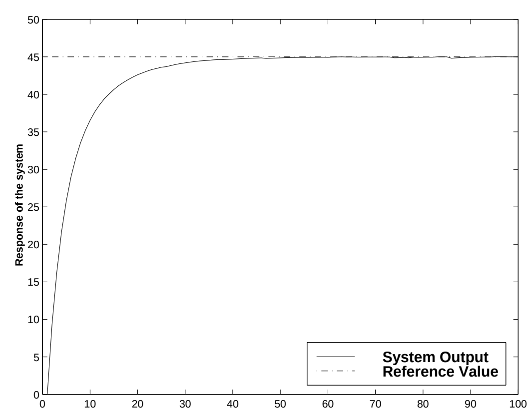

Fig. 10. Application of the proposed algorithm to a Heat Exchanger

VI. Conclusion

A novel approach to implement Nonlinear Model Predictive Control using Genetic Algorithms has been pro-

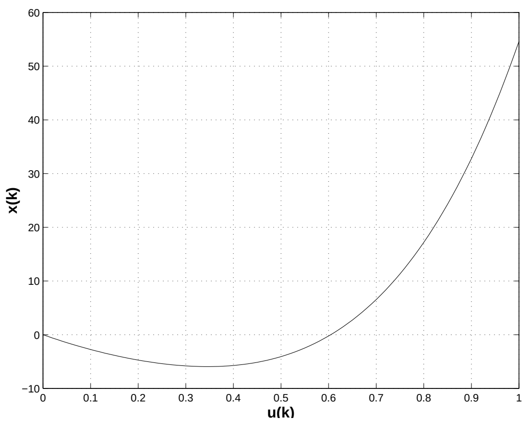

Fig. 11. Nonlinear characteristics of a Heat Exchanger

posed. Results are shown for the case of two chemical plants demonstrating the successful application of the proposed algorithm. The algorithm is applied to single-input single-output(SISO) systems. However, it can be generalized to multi-input multi-output(MIMO) systems.

Acknowledgements

This work was supported by the King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia under Grant Reference Number SABIC 99-13.

References

[1] S. J. Qin and T. A. Badgewell, "An overview of industrial model predictive control technology," *Survey Paper*.
[2] J. T. J. Richalet, A. Rault and J. Papon, "Model predictive heuristic control: Applications to industial processes," *Automatica*, vol. 14, pp. 413–428, 1978.
[3] C. Cutler and B. Ramaker, "Dynamic matrix control– a computer control algorithm," *Proceedings of the Joint Automatic Control Conference*, 1980.
[4] M. C. Clarke, D.W. and T. P.S., "Generalized predictive control. part 1: The basic algorithm," *Automatica*, vol. 23, no. 2, pp. 137–148, 1987.
[5] M. C. Clarke D.W. and T. P.S., "Generalized predictive control. part 2: Extensions and interpretations," *Automatica*, vol. 23, no. 2, pp. 149–160, 1987.
[6] D. Clarke, ed., *Advances in Model-Based Predictive Control*. Oxford Science Publications, 1994.
[7] R. Soeterboek, *Predictive Control, A Unified Approach*. Prentice Hall, 1992.
[8] J. Richalet, "Industrial applications of model based predictive control," *Automatica*, vol. 29, no. 5, pp. 1251–1274, 1993.
[9] D. M. P. Carlos E. Garcia and M. Morari, "Model predictive control: Theory and practice — a survey," *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.
[10] J. H. L. Manfred Morari, "Model predictive control: past, present and future," *Computers and Chemical Engineering*, vol. 23, pp. 667–682, 1999.
[11] A. P. Sandra J. Norquay and A. Romagnoli, "Application of Wiener Model Predictive Control(WMPC) to a pH Neutralization Experiment," *IEEE Transactions on Control Systems Technology*, vol. 7, no. 4, pp. 437–445, 1999.
[12] S. M. Sait and H. Youssef, *Iterative Computer Algorithms with Applications in Engineering, Solving Combinatorial Optimization Problems*. IEEE Computer Society, 1999.
[13] T. Wigren, *Recursive Identification Based on the Nonlinear Wiener Model*. PhD thesis, ACTA Universitatis Upsaliensis, 1990.
[14] S. H. J. Esref Eskinat and W. L. Luyben, "Use of hammerstein models in identification of nonlinear systems," *AIChE*, vol. 37, no. 2, pp. 255–268, 1991.