A RULE-BASED NEURO-OPTIMAL CONTROLLER for NONLINER MIMO SYSTEMS

Serhat Tuncay*, Kemal Leblebicioglu*, Canan Ozgen+, Ugur Halici*

*Middle East Technical University, Dept. of EE. Eng.

+Middle East Technical University, Dept. of Chem. Eng.

[1]E-mail addresses of the authors.

Abstract

In this study, we propose a new method to control multi-input multi-output (MIMO) systems optimally. The method is based on a rule-base derived optimally, which is then interpolated by neural networks. The idea is originally based on the knowledge-based artificial neural networks (KBANN) which perform interpolation in the rule space of an expert system.

Keywords: Optimal control, neural networks, rules based systems, interpolation.

I.INTRODUCTION

The design of controllers for MIMO systems has always been a hard problem even for the linear ones [1]. The only prevailing idea used in the control of linear MIMO system is decoupling, if possible at all. During the last 10-15 years there have been serious attacks on this problem by methods that are especially constructed to control nonlinear plants, such as neuro-control

[1] {tuncay,lebleb}@ec.eee.metu.edu.tr, {cozgen, halici}@metu.edu.tr

and sliding mode control techniques. Just to mention a few of those studies recently done, one may look at (Nie 1997), (Shogested and Postlethwaite 1997), (Ahmed and Tasaddug 1998), (Linker and Nyogesu 1996), (Utkin 1970). Most of these techniques are quite complicated and possibly working for a particular case only.

The fuzzy control techniques had limited application in MIMO systems control mainly because of the facts that the derivation of rules is not easy (usually not available) and the number of rules is too high, depending on the number of outputs and states.

Ours is a new attempt to this unsettled problem using a rule-base combined with neural networks. The idea is originally based on the knowledge-based artificial neural networks (KBANN) which perform interpolation in the rule space of an expert system (Towell and Shavlih 1994). On the other hand there are interesting details and generalizations (that we have developed together with an interesting case study) which have been discussed in the following sections.

1.1 PROBLEM DEFINITION

It is assumed that a MIMO plant is given with a known mathematical model as shown below

$$\begin{aligned} \dot{x}(t) &= f\left(x(t), u(t)\right) \\ y(t) &= g\left(x(t)\right) \end{aligned} \tag{1}$$

where $x(t),\ f(x(t),u(t)) \in R^n$, $u(t) \in R^m$ and $y(t), g(x(t)) \in R^p$. There is a reference signal $y_d(t) \in R^P$ and the system output $y(t)$ is supposed to track it. Thus the controller to be developed has the model the reference tracking controller structure.

2.CONTROLLER

It is based on a rule-base and the rules are developed by making use of the mathematical model of the plant in an optimal sense. That is, since model is available, by partitioning the state-space and the output-space and defining a representative for each partition, one can determine the control signals (i.e. rules) optimally, using a suitably chosen cost function.

2.1 RULE DERIVATION

Suppose that each component of the state vector has N_i,

k

components. Then there is a total of $\left(\prod_{i=1}^{n} N_i\right)\left(\prod_{k=1}^{P} O_k\right)$ rules to be derived. If the system state is initially at the i-th partition (the representative of which is x_i) and the system's initial and final or desired stated are at partitions O_v and O_k (their representatives are y_v and y_k, respectively), the associated rule can be found optimally by solving the optimal control problem of minimizing the cost function

$$J(u) = \frac{1}{2}\left(y(t_f) - y_k\right)^T H\left(y(t_f) - y_k\right) + \frac{1}{2}\int_0^{t_f}\left(y(t) - y_d(t)\right)^T Q\left(y(t) - y_d(t)\right)dt + \frac{1}{2}\int_0^{t_f} u(t)^T R u(t) \tag{2}$$

Subject to the state equation

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t)) \\ x(0) &= x_i \\ y(t) &= g(x(t)) \end{aligned} \tag{3}$$

Usually H, Q and R are diagonal matrices with suitably chosen diagonal entries. The vector function y_d (t) can be taken as any smooth function with

$$y_d(0) = y_v \ , \ y_d(t_f) = y_k$$

$$\dot{y}_d(0) = \dot{y}_d(t_f) = 0$$

In addition, the constraints on u(t), that is, $|u_i(t)| \le B_i$
can be easily incorporated in our steepest descent like optimal control problem solver (Haykin 1996) .

2.2 NEURAL NETWORK

In order to be able to generate the control inputs so that the system output trajectory follows an optimal path between arbitrarily specified initial and final output states, one has to train a multilayer perceptron-like neural network [8]. This neural network should accept present state x(0) and output y(0), and desired output $y(t_f)$ as its inputs and should generate the optimal control signal u(t) to accomplish the task. For training, input signals produced by optimal control and initial and final points of outputs should be used. It is interesting to note that, at least theoretically, the neural network is a semi-infinite dimensional one [9], [10] in the sense that it is a mapping between the finite dimensional input space and the infinite dimensional output space (i.e., control functions). In practice, the neural network can produce the samples of the control signal. After training, the neural network acts as a real-time optimal controller.

3. CASE STUDY

The dynamic response and control of the steam-jacketed kettle shown in figure 1 are to be considered. The system consists of a kettle through which water flows at a variable rate w lb/time. The inlet water temperature T_i, is 40 °F, which may vary with time. The kettle water, which is well agitated, is heated by steam condensing in the jacket at temperature T_v. This is a three-input two output nonlinear system. Flow rate of inlet water, flow rate of outlet water and flow rate of steam are the inputs for the system. Temperature and the mass of the water inside the kettle are outputs.

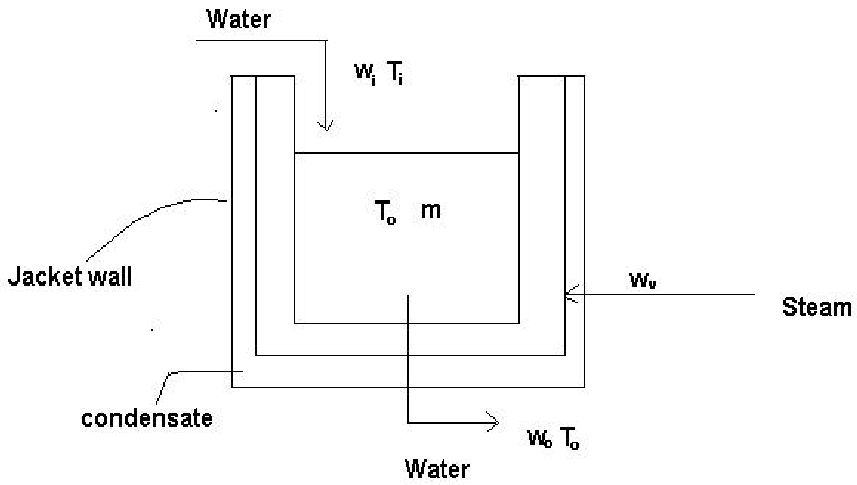

Figure 1
steam-jacketed kettle

3.1. ANALYSIS of the KETTLE

The following assumptions are made for the kettle:

1. The heat loss to the atmosphere is negligible.
2. The thermal capacity of the kettle wall, which separates stem from water, is negligible compared to that of water in the kettle.
3. The thermal capacity of the outer jacket wall, adjacent to the surroundings, is finite, and the temperature of this jacket wall is uniform and equal to the steam temperature at any instant.
4. The kettle water is sufficiently agitated to result in a uniform temperature.
5. The flow of heat from the steam to the water in the kettle is described by the expression

$$q = U\left(T_v - T_o\right)$$

where

q=flow rate of heat Btu/(hr)(ft^2)
U=overall heat transfer coefficient, Btu / (hr) (ft^2)(°F)
T_v=steam tempreture °F
T_o=water temperature °F

The mathematical model of the kettle can be obtained by employing the ideas of energy and mass balance on the water side first, and then on the steam side next. The symbols used throughout this analysis are defined below:

T_i= Temperature of inlet water. °F
T_o = Temperature of outlet water °F
w_i = flow rate of inlet water, lb/time
w_o = flow rate of outlet water, lb/time
w_v = flow rate of steam, lb/time
w_c = flow rate of condensate from kettle, lb/time
m = mass of water inside the kettle, lb
m_1 = mass of jacket wall, lb
V = volume of the jacket steam space, ft^3

C = heat capacity of water Btu / (lb)(°F)
C_1 = heat capacity of metal in jacket wall Btu / (lb)(°F)
A = cross sectional area for heat exchange
t = time
H_v = specific enthalpy of steam entering, Btu / lb
H_c = specific enthalpy of steam leaving, Btu / lb
U_v = specific internal energy of steam in jacket, Btu / lb
ρ_v = density of steam in jacket, lb / ft^3

Writing energy balance and mass balance equations for water and steam side we obtain

$$mC\frac{dT_o}{dt} = w_i C(T_i - T_o) + UA(T_v - T_o) + w_o(T_o - T_v)$$

$$\frac{dm}{dt} = w_i - w_o$$

$$m_1 C_1 \frac{dT_v}{dt} = w_v(H_v - H_c) - (U_v - H_c)V\frac{dr_v}{dt} - UA(T_v - T_o)$$

$$V\frac{dr_v}{dt} = w_v - w_o$$

So, the state, input and output vectors are, respectively

$$\bar{x} = \begin{bmatrix} T_o \\ m \\ T_v \\ r_v \end{bmatrix} \quad \bar{u} = \begin{bmatrix} w_i \\ w_v \\ w_o \end{bmatrix} \quad \bar{y} = \begin{bmatrix} T_o \\ m \end{bmatrix}$$

3.2. SIMULATION RESULTS

In our simulation, the temperature range is [40 °F, 180 °F] and mass (i.e., the level of the water inside the kettle.) range is [20 lb, 30 lb]. There is no need to partition the rest of the states because these are related with the temperature of the steam flowing into the jacket. Since the temperature of the steam flowing into is constant, single partition is enough for these states. The temperature range is divided into seven regions and mass range into two regions. Therefore, we have to produce 7x7x2x2=196 rules from the optimal control procedure and then use these 196 rules in the training of the neural networks. Since there are three inputs, three separate neural networks, each of them have four inputs, two hidden layers having 100 and 50 neurons respectively, and an output layer consisting of 25 neurons, have been constructed.

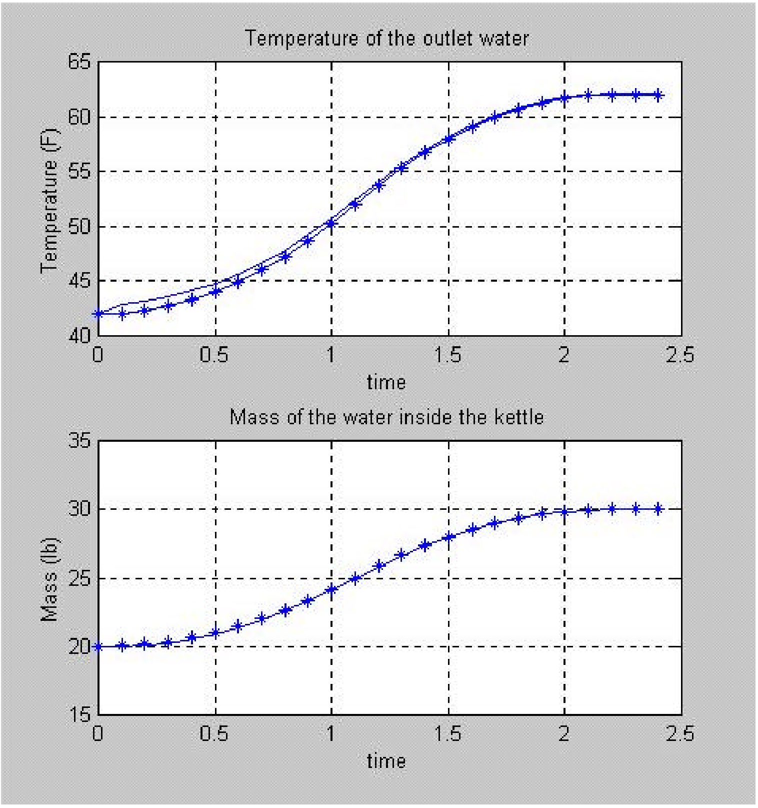

Figure 1, * desired trajectory, _ trajectory from neuro-controller

After training, neural networks work as real time controllers for the system. For example, if we take the initial values for outlet water temperature and mass of the water as [20 lb, 42 °F] and reference inputs as [30 lb, 62 °F], the results from neural networks are given in figures 2, 3, 4 and 5 together with the results from the optimal control procedure. In figure 6, water temperature in the kettle which is controlled by a neural network in real time is given with the desired trajectory.

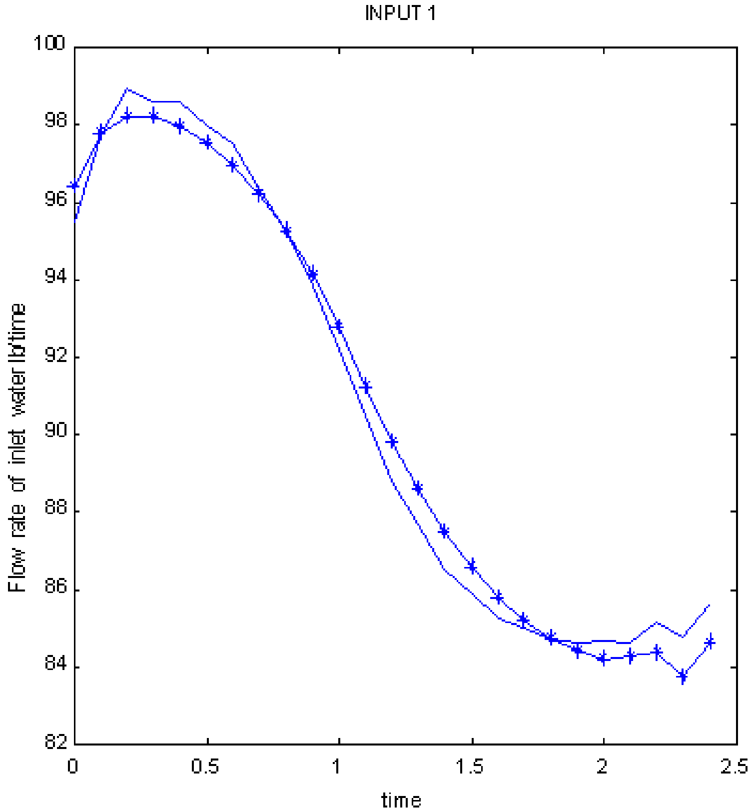

Figure 2. * output from optimal control, _ output from neural network.

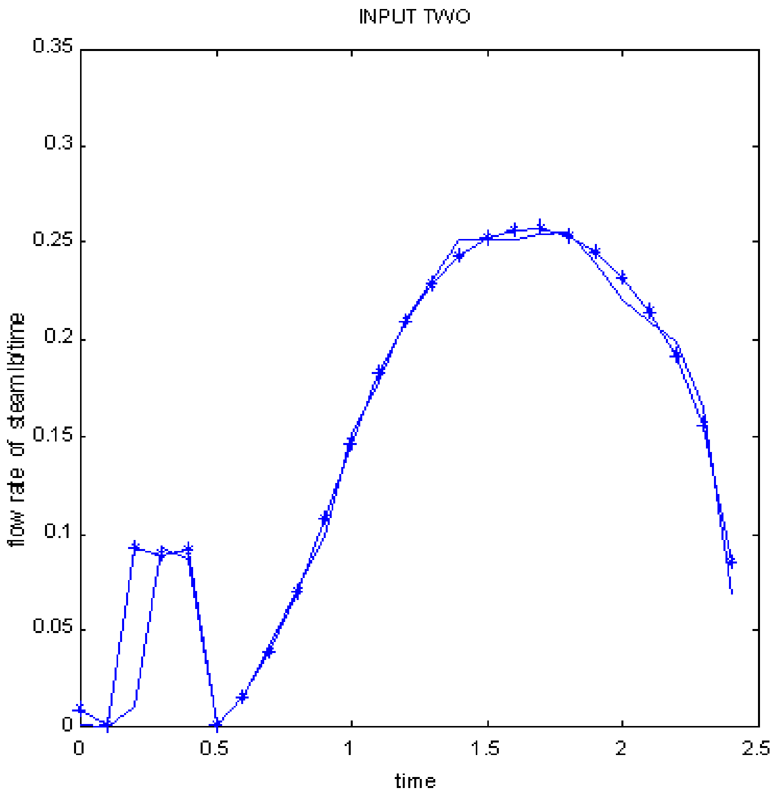

Figure 3. * Output from optimal control, _ output from neural network

It can be seen from figure 1 that the online optimal neuro-controller can bring the system into the desired output states through a desired trajectory. The desired trajectories for the output states are also given in figure 1.

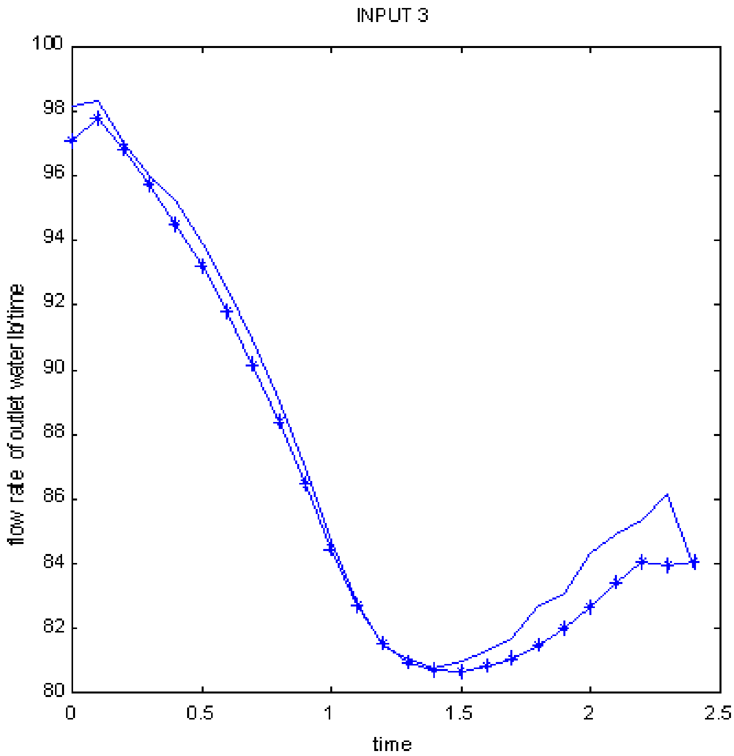

Figure 5
* output from optimal control
_ output from neural network

In figure 6, a bell-shaped output state trajectory for one of the output states, the temperature of the water inside the kettle, is given, the neuro-controller achieves to produce control inputs to follow the given trajectory which can be seen in figure 6.

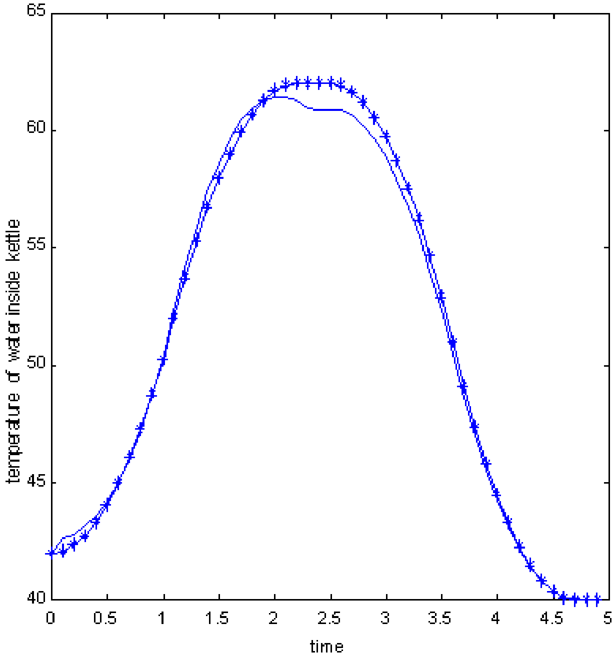

Figure 6
* Desired trajectory
_ Trajectory produced by the inputs from online neural network controller

4. CONCLUSIONS

In this work, an optimal neurocontroller has been suggested for controlling MIMO systems. The ideas presented were checked by simulation studies on a simple steam-jacketed kettle system. The preliminary results obtained so far have shown that the suggested method is worth pursuing further. The only disadvantage of the method (according to us) is that the number of rules to be derived in a complex plant control can be prohibitively high which also makes the derivation time too long. On the other hand, the method is very simple and can be made adaptive with some effort. Studies are continuing to generalize the method to cover the disturbance rejection and robustness problems as well.

REFERENCES

1) S. Shogestad and I. Postlethwaite (1997). Multivariable Feedback Control, John Wiley & Sons.
2) J.Nie (1997). "Fuzzy Control of Multivariable Nonlinear Servomechanisms with Expilicit Decoupling Scheme, " IEEE Transactions on Fuzzy Systems, Vol 5. No 2., pp. 304-311.
3) M.S. Ahmed, I. A. Tasaddug (1998). "Neural Servocontroller for Nonlinear MIMO Plant," IEEE Proceedings Control Theory Applications, Vol. 145, No 3, pp.277-291.
4) D.A. Linkens , H.O. Nyogesu (1996). " A Hierarchical Multivariable Fuzzy Controller for Learning with Genetic Algorithms, " Int. Journal of Control, Vol. 63 , No.5 ,pp. 865-883.
5) V.I. Utkin (1992). Sliding Modes in Control Optimization, Springer-Verlag, Berlin.
6) G. G. Towell, J. W. Shavlih (1994). " Knowledge-Based Artificial Neural Networks, " AI. , Vol. 70 , pp. 119-165.
7) S. Haykin (1996). A comprehensive Foundation of Neural Networks, Prentice-Hall, NJ.
8) D.E. Kirk (1970). Optimal Control, Prentice-Hall, NJ.
9) M. Kuzuoglu, K. Leblebicioglu (1996). "Infinite Dimensional Multilayer Perceptrons, IEEE Trans. Neural Networks, Vol.7, No 4., pp. 889-896.
10) K. Leblebicioglu, U. Halici (1997). "Infinite Dimensional Radial Basis Function Neural Networks for Nonlinear Transformation on Function Spaces, Nonlinear Analysis, Vol. 30, Part 3, pp. 1649-1654.
11) D.R.Coughanowr, L.W. Koppel (1965). Process Systems Analysis and Control, McGraw-Hill.