Proceedings of the 7^th International Conference on Applied Informatics
Eger, Hungary, January 28–31, 2007. Vol. 1. pp. 247–254.

RBF-MRAC neurocontroller for industrial applications

István Pintér, Iván Lajtai

College of Kecskemét GAMF Faculty
Automation and Applied Informatics Group
e-mail: {pinter.istvan,lajtai.ivan}@gamf.kefo.hu

Abstract

In the case of model-reference adaptive control (MRAC) the output of an ideal reference model is approximated by the controlled plant using an appropriate control rule. Our former results in MRAC of first order linear systems have been extended to non-linear systems with static non-linearity. In the latter case there is a problem of approximating non-linearity during the adaptation process, therefore a radial basis function neural network (RBF) has been applied as an approximator. An algorithm, published in the relevant literature, has been investigated and implemented for simulation purposes. The purpose of the simulations is to study the properties of the RBF-MRAC neurocontroller in order to apply it for industrial applications. The reference model is given by a first order linear differential equation and the plant is given by a first order non-linear one. The computations and implementations are based on the fourth-order Runge-Kutta method. In the paper the details of implementations and the simulation results are presented. From the point of view of possible industrial applications it is worth-while mentioning that the IEEE-754-compatible floating point arithmetic has become a standard component of the instruction set of recent programmable logic controllers (PLCs). This fact - together with the ST-language standard - makes it possible to realize much more difficult algorithms in PLC programs, so such way of the realisation of the neurocontroller is our recent task.

Keywords: model-reference adaptive control, radial basis function network, neurocontroller

MSC: 93C40

1. Introduction

In the practice of industrial informatics programmable logic controllers (PLCs) are often used to solve control problems. Because of the demand for leading-edge industrial applications the IEEE-754-compliant floating-point arithmetic has become a standard component of the recent PLC's instruction set, which is obviously makes it possible to implement novel algorithms on PLC. Recently a HHL-language, the so called ST-language has been defined and standardized in order to increase the speed and accuracy of the PLC-based software development. There is a PLC with floating-point arithmetic and with ST-language support in our laboratory, so we decided to implement a PLC-based neurocontroller in order to regulate a first-order non-linear plant. The recent paper summarizes our results in the implementation and simulation of the so-called RBF-MRAC system. The structure of the article is the following. After the introduction in the second section the concept of model reference adaptive control is introduced by a SISO-LTI (single input single output, linear time invariant) system, followed by the detailed mathematical description of the RBF-MRAC neurocontroller system in the third part. In the fourth section the results of the simulations are presented with figures. The paper ends with summary, acknowledgements and references.

2. The model reference adaptive control

Adaptive controllers are able to change their parameters according to specifications even in varying operation conditions. During the past decades several methods have been elaborated for the adaptive control task – one of these, the so called model reference adaptive control (see [1, 2]), is the subject of this paper. Our goal is to control a first order non-linear plant, it is reasonable to introduce the concept of MRAC in the case of a first order linear system. Let the differential equation below give the mathematical model of the plant:

$$\frac{dy(t)}{dt} = -ay(t) + bu(t), \tag{2.1}$$

where $y(t)$ and $u(t)$ in (2.1) denote the output and input signals of the plant, respectively, with fixed, but unknown parameters of $a \in \mathbf{R}^+$ and $b \in \mathbf{R}$. Let the first order SISO-LTI reference model be described by

$$\frac{dy_m(t)}{dt} = -a_m y_m(t) + b_m r(t), \tag{2.2}$$

where $y_m(t)$ in (2.2) denotes the output signal of the model, $r(t)$ denotes the reference signal of the system, $a_m \in \mathbf{R}^+$ and $b_m \in \mathbf{R}$ are the prescribed model parameters. Our goal is to give a control rule $u(t)$ and by using this rule and the reference signal we have to minimize the criterion function

$$E(t) = \frac{1}{2}e^2(t), \tag{2.3}$$

where

$$e(t) = y(t) - y_m(t). \tag{2.4}$$

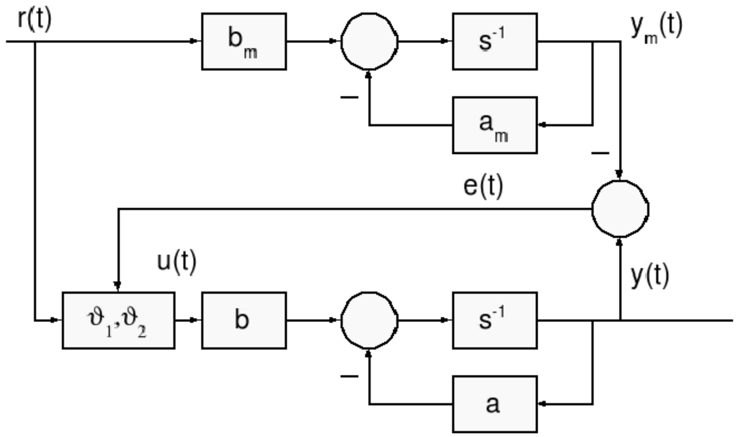

Figure 1: Block diagram of the first order MRAC system

By settling the control rule as

$$u(t) = \vartheta_1(t)r(t) - \vartheta_2(t)y(t), \tag{2.5}$$

and by following Whittaker's original arguments on applying the steepest descent method [1], for the parameter change we get (given here only for the parameter ϑ_1):

$$\frac{d\vartheta_1(t)}{dt} = -\gamma \frac{\partial E(t)}{\partial t}, \tag{2.6}$$

where $\gamma \in \mathbf{R}$ is an adequately chosen constant. The task in this case is to determine the partial derivative(s) in equation (2.6) – the details can be found at [3]. Figure 1 shows the structure of the first order MRAC system.

3. RBF-MRAC neurocontroller

Following the method presented in section 2 it is not an easy task to derive a suitable control rule $u(t)$ even in a first oder non-linear plant, where component $ay(t)$ is replaced by $f[y(t)]$, a real-value, continuously differentiable static non-linear function. Such systems are desrcibed by the equation below:

$$\frac{dy(t)}{dt} = -f[y(t)] + u(t). \tag{3.1}$$

The linear model in this case is the same as in the prevoius section (see (2.2)). By choosing the control rule $u(t)$ as:

$$u(t) = -a_m y(t) + b_m r(t) + h[y(t), \mathbf{p}(t)], \tag{3.2}$$

where $h(.)$ is a suitable function and $\mathbf{p}(t)$ is the parameter vector of the system, then after substituting (3.2) into (3.1) we get:

$$\frac{dy(t)}{dt} = -a_m y(t) + b_m r(t) + \varepsilon(t), \tag{3.3}$$

where

$$\varepsilon(t) = h[y(t), \mathbf{p}(t)] - f[y(t)]. \tag{3.4}$$

That is, it is necessary to decrease the difference given in (3.4) in order to approximate the model output with the regulated output of the plant. For solving this approximation problem [4] suggests the application of an online-trained RBF network, because it can be considered as universal approximator [5]. The block

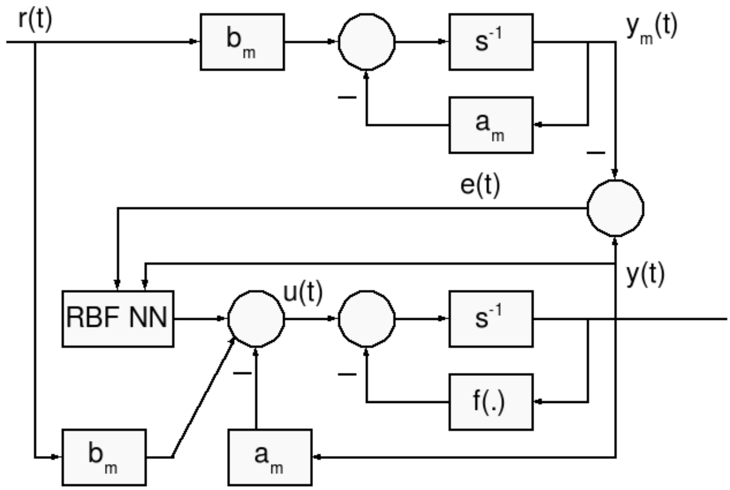

Figure 2: Block diagram of the RBF-MRAC system

diagram of the neurocontroller can be seen in Figure 2. The parameter vector can be related to number N of neurons and to the weight-vector $\mathbf{w}(t)$ of the RBF network, and the function $h(.)$ above is the output of the network. Because there is no estimation given in [4] for the determination of the number of neurons, it is left for numerical experiments. For the function $h(.)$ and for the weight-vector update the propositions are the following in [4]:

$$h([y(t), \mathbf{w}(t)] = \sum_{i=0}^{N-1} w_i(t) exp \left\{ \frac{[y(t) - c_i]^2}{2\sigma_i^2} \right\}, \tag{3.5}$$

and

$$\frac{dw_i(t)}{dt} = -\alpha e(t) exp \frac{[y(t) - c_i]^2}{2\sigma_i^2} - \beta w_i(t), \tag{3.6}$$

where $w_i(t)$ denotes the ith component of the weight vector, c_i and σ_i denote the fixed centers and widths of the radial basis functions (which are Gaussian-functions in our case), N is the number of RBF-neurons and $\alpha, \beta \in \mathbf{R}^+$. The RBF-MRAC system has been simulated using equations (2.2), (3.1), (3.2), (3.5), (3.6). As our basic goal is an ST-language program written for a suitable PLC, the fourth-order Runge-Kutta method has been used for solving the differential equations above.

4. Simulation results

Our simulation results in the case of non-linearity of [4] (a third-order polynomial) and in the case of exponetial-type non-linearity can be found in this section. The linear model was the same in both cases.

4.1. Simulation in the case of a third-order polynomial

According to [4], the parameters of the linear model have been chosen as $b_m = 2$, $a_m = 2$, and the non-linear function was the third-order polynomial $f(x) = 2x + 0.8x^3$. The simulation parameters are summarized in Table 1. The initial value of

$r(t)$	N	α	β	centers	widths
$sin(2\pi f_0 t),\ f_0 = 1$	10	0.999	0.001	$c_i = -1 + i\frac{2}{N-1}$	0.3

Table 1: Simulation parameters in the case of a third-order non-linear function

the weight-vector has been chosen according to equation $w_i(0) = c_i + rnd(1) - 0.5$, where rnd(1) denotes a pseudo-random number in [0,1]. Figure 3 shows the

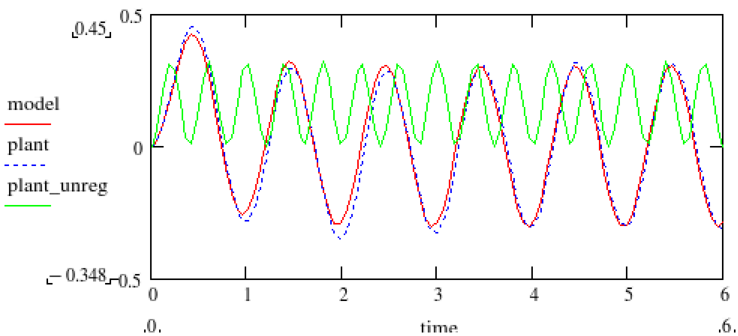

Figure 3: Regulated plant output in the case of a third-order non-linear function

regulated plant output, the output of the model and the output of the unregulated

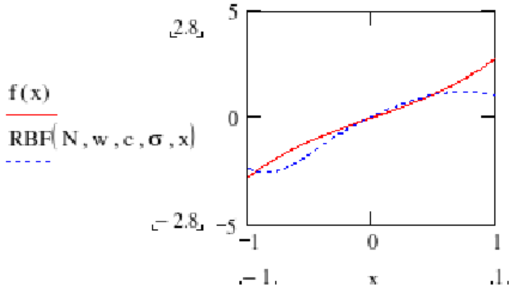

Figure 4: Approximation of the non-linear function by RBF-network after 100 iteration steps

plant in the case of $r(t) = u(t)$, while in Figure 4 the approximation of the non-linear function can be seen after 100 iterations with 0.06 step-size.

4.2. Simulation in the case of exponential-type non-linearity

In this case the parameters of the linear model have also been chosen as $b_m = 2$, $a_m = 2$, and the general form of the non-linear function was $f(a, b, c, x) = aexp(bx) + c$, with $a = 6.13 \times 10^3$, $b = -9.787 \times 10^{-4}$, $c = 2.097 \times 10^3$. These parameters have been obtained by measurement in the realised prototype-system (see Figure 5). The current-resistor characteristic of the lamp-photoresistor system

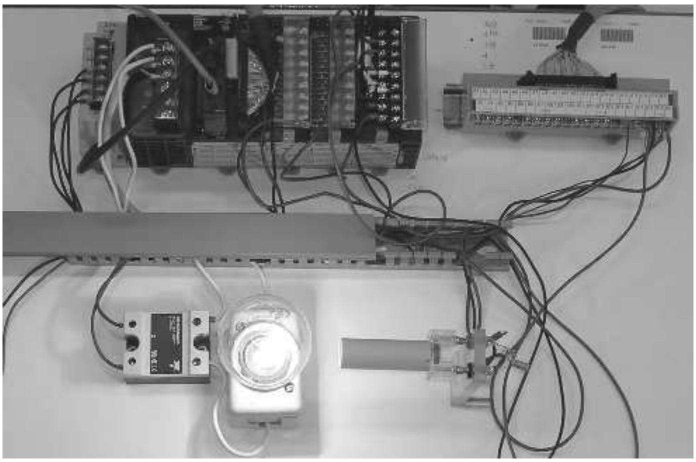

Figure 5: The realised prototype system with exponential-type non-linearity

has been measured using the PLC's analogue I/O modules and the mathematical form of the non-linear function has been determined by curve-fitting. The simulation parameters in this case can be seen in Table 2. The initial value of the

$r(t)$	N	α	β	centers	widths
$sin(2\pi f_0 t),\ f_0 = 1$	25	0.999	0.001	$c_i = 2500\frac{i-1}{N-1}$	$\frac{2500}{N}$

Table 2: Simulation parameters in the case of exponential-type non-linear function

weight-vector has been chosen according to equation $w_i(0) = 5000 - c_i$. Figure 6

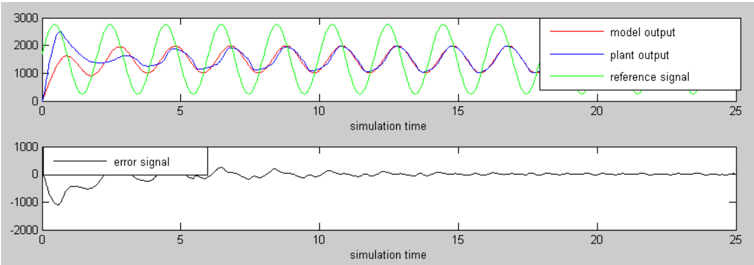

Figure 6: Regulated plant output and output error in the case of exponential-type non-linearity

shows the regulated plant output, the output of the model and the output of the unregulated plant in the case of $r(t) = u(t)$. The difference signal of (2.4) can also be seen (lower trace).

5. Summary

The basic aim of our work is to realise a RBF-MRAC neurocontroller in a PLC. In this paper our simulation results have been demonstrated. Using the fourth-order Runge-Kutta method the linear and non-linear differential equations have been solved. The output error with acceptable convergence speed has been achieved both in the case of third-order non-linearity and in the case of exponential-type one. The parameters of the RBF-network have been determined experimentally. Our ongoing task is the realisation of the algorithm described in this paper using PLC's ST-language.

Acknowledgements. We would like to thank to Prof. László Dávid (Sapientia University) for his kind help and encouragement in our work in model reference adaptive control systems.

References

[1] Osburn, P. V., Whitaker, H. P., Kezer, A., Comparative studies of model-reference adaptive control systems, *Institute of Aeronautical Sciences*, Paper No. 61-39., (1961).

[2] Sastry, S., Bodson, M., Adaptive Control - Stability, Convergence and Robustness, *Prentice Hall*, (1989).

[3] Lajtai I., Pintér I., Controlling of a MRAC-based servodrive using digital PLC-outputs (in Hungarian), *A GAMF Közleményei*, Vol. XIX. (2004), 45–59.

[4] Patino, H. G., Liu, D., Neural Network-Based Model Reference Adaptive Control System, *IEEE Trans. on Systems, Man and Cybernetics Part B: Cybernetics*, Vol. 30 No.1. February, (2000), 198–204.

[5] Haykin, S., Neural Networks, *Prentice Hall*, (1999).