International Journal
Information Technology Applications (ITA)
Volume 11, Number 2, December 2022

NEURAL-GENETIC CONTROL ALGORITHM FOR TWO-LINK ROBOT

Slavomír Kajan, Štefan Kozák

Abstract:

This paper deals with soft state control of non-linear dynamic model – robot. Soft methods based on neural networks and genetic algorithms demonstrate powerful problem solving ability. They are based on quite simple principles, but take advantage of their mathematical nature: non-linear iteration computation solutions. One of the ways of control of such non-linear systems is the use of neural networks as an effective controllers. In this paper a new methodology is proposed, where for neural controller structures and parameters are computed by the genetic algorithm (GA). The proposed approach is represents by direct neural controller using multilayer perceptron (MLP) network in feedback tracking control loop. The training method using GA allows find optimal adjustment of neural network weights so that high performance is obtained. The proposed control method is realized in Matlab/Simulink and demonstrated on typical non-linear systems (two-link robot).

Keywords:

Genetic algorithm (GA), MLP network, neural controller (NC), neural network, non-linear dynamic system, robot model.

ACM Computing Classification System:

Evolutionary algorithms, Artificial intelligence, Control systems.

Introduction

During the past decade there has been an intense interest in developing the soft computing methods (SCM) and techniques for a wide variety of scientific and engineering applications. The process control research in this area has been largely concerned with four SCM methods: knowledge-based systems, neural networks, fuzzy logic, genetic algorithms and various combinations of these techniques. In recent years, different fuzzy logic models are developed to cope with nonlinearities and uncertainties in many industrial systems. The fuzzy and neural models are mainly used to model system with complex physical structure. Soft computing methods can be used to optimize model parameters over a full range of input–output data. In recent years, genetic algorithm (GA) is widely used as an optimization method for training and adaptation of parameters in dynamical system. In many cases, the GA techniques are integrated in fuzzy logic and neural network structure as suitable optimization approach. GAs have many advantages over the conventional optimization methods. It does not require a complete system model and can be employed to globally search for the optimal solution. In literature as well as in practice applications of control systems occur, which are using artificial neural networks. The multilayer perceptron (MLP) neural network has good properties for direct control of non-linear systems (Chaozing Z., 1998).

For control of some classes of non-linear dynamical systems with advantage neural controllers (NC) are used. The neural network can be applied as a direct controller. It can emulate expert or another type of controller, it can be direct inverse controller, neuro-predictive controller or direct controller.

The main goal of this paper is to present an approach to the state control of an industrial robot using neural network and genetic algorithm. We know that the robots are characterized by a complex non-linear dynamical structure with un-modelled dynamics and unstructured uncertainties. These features make the designing of controllers for the robots a difficult task in the framework of state control. For the design of robot control are often used optimal control, linear-quadratic and neural control approaches. This article deals with last named type which is optimized by genetic algorithms.

1 Dynamic Model of Robot

We shall consider the robot with the kinematics structure by (Fig.1), the dynamic model of which is described in the state space:

$$\begin{aligned} \dot{x}_1 &= x_2, \quad \dot{x}_2 = \frac{m_b}{m_r} x_1 x_4^2 + \frac{K_1}{m_r} u_1 \\ \dot{x}_3 &= x_4, \quad \dot{x}_4 = -\frac{2 m_b x_1 x_2 x_4}{I_{23} + m_b x_1^2} + \frac{K_2}{I_{23} + m_b x_1^2} u_2 \end{aligned} \tag{1}$$

where $m_b = m_z + m_1,\ m_r = m_{rr} + m_b + m_2$

m_z = 35 kg, *is the mass of the weight,*

m_1 = 52 kg, *is the mass of the grasp head and part of the arm,*

m_{rr}= 62.5 kg, *is reduced mass of the gear,*

m_2 = 78 kg, *is the mass of the servomotors of the arm.*

K_1 = 281 Nm, K_2 = 291 Nm *are constants of the operational values.*

$$I_{23} = I_r + (m_2 + m_3 + m_4) r_0^2$$

I_r = 82.5 kgm^2 is reduced torque of the inertia of the electric servo-motor and the gear box.

m_3 = 90 kg, m_4 = 125 kg, r_0 = 250 mm

The elements of the state vector are:

$x_1 = r$[m]; $x_2 = \dot{r}$ [m.s^{-1}]

$x_3 = \varphi$ [rad]; $x_4 = \dot{\varphi}$ [rad.s^{-1}]

$u_1(t)$ and $u_2(t)$ are control action variables

where r is translation of the arm.

φ is rotation of the arm $(0,2\pi)$

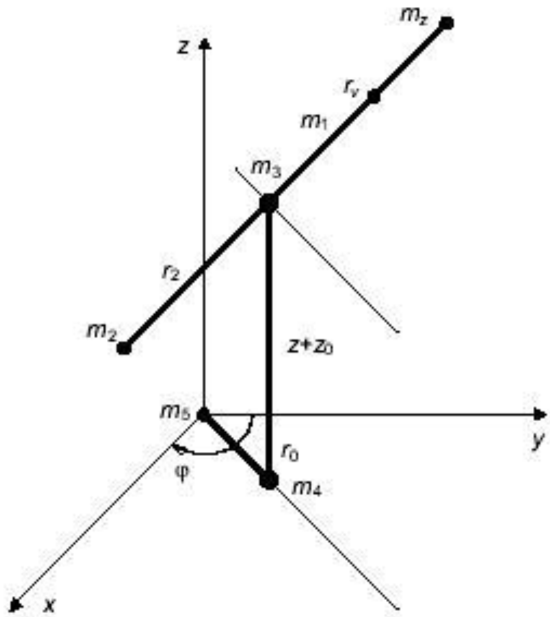


Fig.1. The kinematic scheme of the robot.

2 Neural Genetic Control Algorithm

In (Fig.2) the scheme of the neural control (NC) with optimisation of controller parameters using genetic algorithm is depicted. The main aim of euro-genetic control is computation of controller law parameters subject to the constraints and objective function defined by (3).

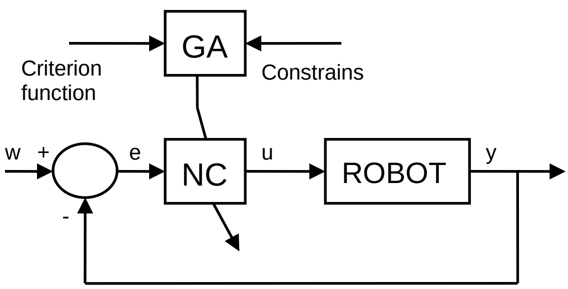


Fig.2. Block scheme of the neural – genetic control system.

Control action can be defined as nonlinear function

$$u(t) = f(e(t), y(t), y(t-1), y(t-2), W(t), t)$$

3.1 Neural Controller

Consider, the neural controller is represented by a multi-layer perceptron network (MLP) with a single hidden layer. This type of neural network is able to approximate any type of a arbitrally continuous non-linear function. The scheme of the proposed neural controller is shown in (Fig.3).

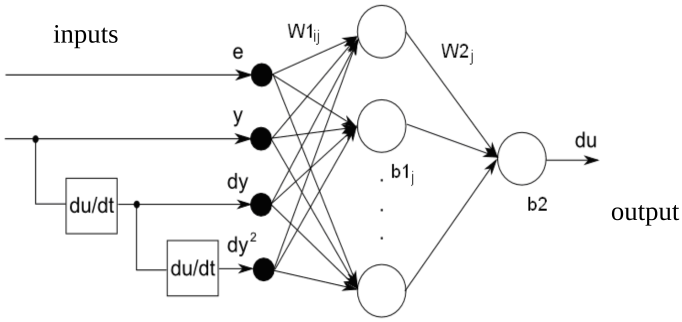


Fig.3. Scheme of the neural controller.

The inputs, states and output variables are the neural network are:

- the control errors $\boldsymbol{e_1}(t)$ and $\boldsymbol{e_2}(t)$, $\boldsymbol{e_1}(t)= w_1(t)- x_1(t)$ and $\boldsymbol{e_2}(t)= w_2(t)- x_3(t)$
- output process variable $\boldsymbol{y}(t)$,
- states of system $\boldsymbol{x}(t)=[x_1(t), x_2(t), x_3(t), x_4(t)]$.

where $w_1(t)$, $w_2(t)$ are reference values of arm translation and rotation. Outputs from the neural network are then the control values $u_1(t)$ and $u_2(t)$. The neural network with such inputs and outputs represents a non-linear state controller, where its outputs are a non-linear functions of its inputs.

In the hidden layer of the multilayer perceptron network (MLP) the hyperbolic tangent activation functions are used (tansig) in form

$$\varphi(a)=\frac{2}{1+e^{-2a}}-1 \ . \tag{2}$$

In the output layer linear activation function has been used. The optimized parameters are the weights between input and hidden layer $W1_{ij}$, weights between hidden and output layer $W2_{jl}$, biases in the hidden layer $b1_j$ and bias in output layer $b2_l$. For the initial setting of the neural controller parameters the Levenberg-Marquardt method is possible to use, where the data from a designed LQ controller as training data can be used.

3.2 Genetic Algorithm

A general scheme of the used GA can be described by following steps (Fig.4):

1. Initialization of the population of chromosomes.
2. Fitness evaluation of the population.
3. End if terminal conditions are satisfied.
4. Selection of parent chromosomes.

5. Crossover and mutation of the parents → children.
6. Completion of the new population from the new children and selected members of the old population. Jump to the step 2.

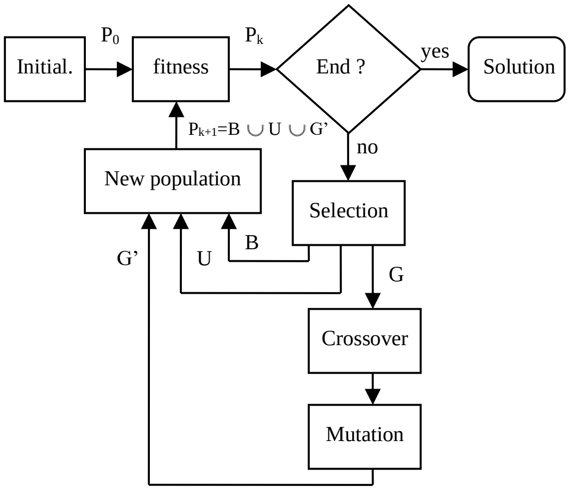


Fig.4. Block scheme of the used genetic algorithm.

The chromosome contains the set of neural network parameters - weights and biases and the optimised fitness function can use performance index in form (3), or other, where T is simulation time, e_1 and e_2 are the control errors, u_1 and u_2 are the control values and q_1, q_2, r_1, r_2 are weight constants.

$$J = \frac{1}{2}\int_0^T \left(q_1 e_1^2 + q_2 e_2^2\right)dt + \frac{1}{2}\int_0^T \left(r_1 u_1^2 + r_2 u_2^2\right)dt \,, \tag{3}$$

After the initialization of the population, fitness of each chromosome of the population is evaluated. Fitness contains closed loop simulation with the model of the non-linear system and the neural controller and the performance index evaluation. The design procedure is based on the genetic algorithm (Fig.5).

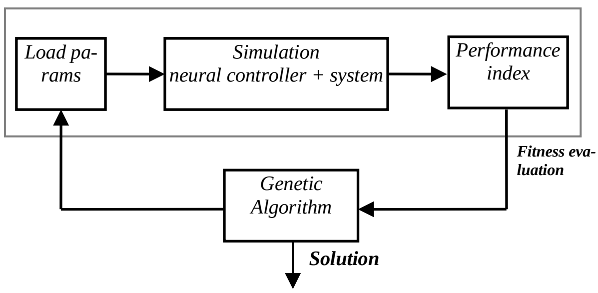


Fig.5. Block scheme of the GA-based neural controller design.

3 Simulation Results

As mentioned, for verification of the design approach NC the simulation model of robot (Fig.6) has been used. The non-linear simulation model of robot was created according to the equations (1). The simulation scheme of neural control of robot dynamic model is displayed in (Fig.7).

In case of the closed loop controller the neural controller with the MLP network with a single hidden layer is used [4]. The hidden layer contains 7 neurons for control of the robot model. The weights and biases of MLP network were optimised using GA with criterion function (3), where weights constants were setting as $q_1=1000$, $q_2=1000$, $r_1 = 0.1$, $r_2 = 0.1$.

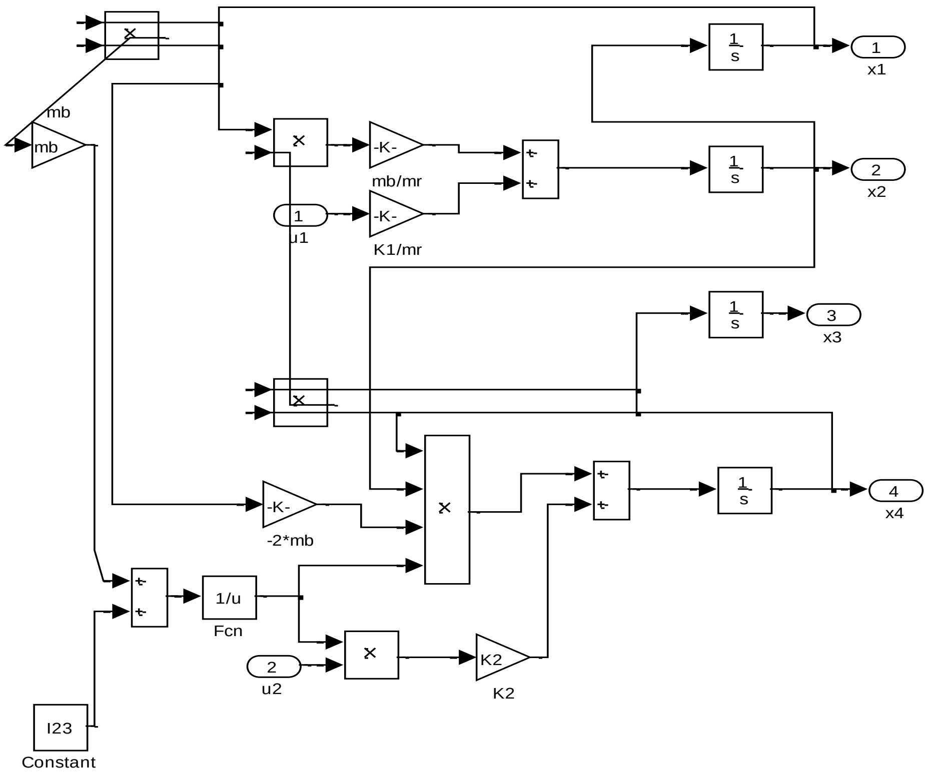


Fig.6. Simulation scheme of non-linear dynamic model of robot.

In (Fig.8) the simulation results as time-responses to arm translation x_1 and tracking of desired arm translation w_1 trajectory under the neural controller are compared. The trajectory of desired arm rotation w_2 is equal as in (Fig.8). The time-responses of system state variables in movement from state $x(0)$=[0 0 0 0] to state $x(T)$=[0.2 0 0.2 0] are displayed in (Fig.9) and (Fig.10). In these figures the simulation results as time-responses to arm translation x_1 and arm rotation x_3 are compared with desired arm translation w_1 trajectory and desired arm rotation w_2 trajectory. The time-responses of control variables u_1 and u_2 of neural controller are displayed in (Fig.11). For tracking of desired trajectory w_1 maximal value of absolute control error was 0.0027 and mean value was 0.0012. For tracking of desired trajectory w_2 maximal value of absolute control error was 0.0014 and mean value was 0.0006.

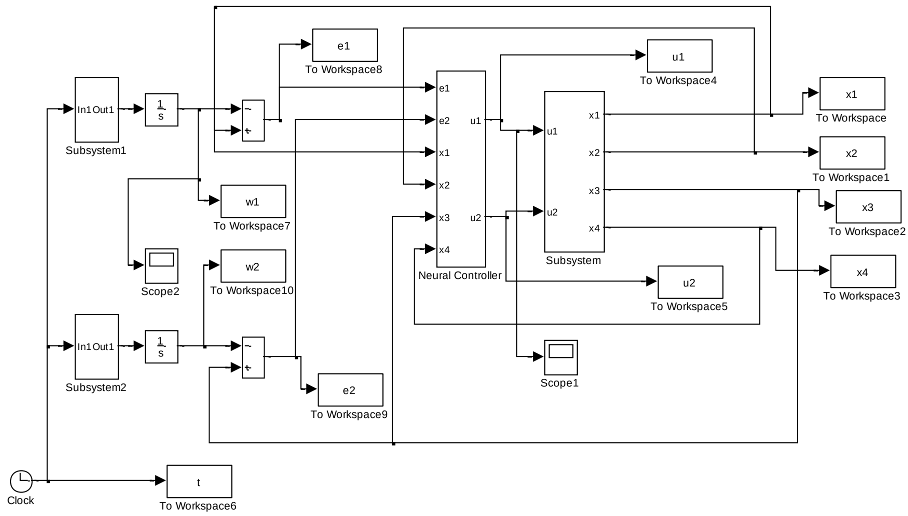


Fig.7. Simulation scheme of neural control of robot model.

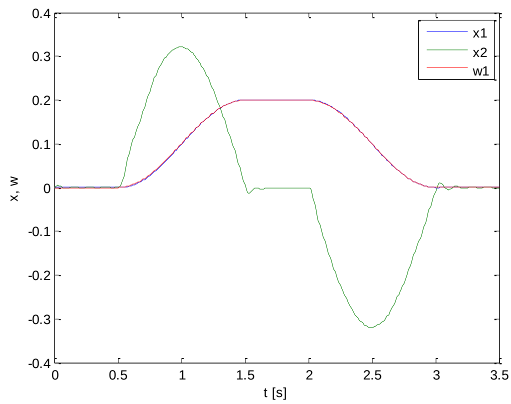


Fig.8. Time-responses of arm translation x_1 (x_2 is speed of x_1) for tracking of desired trajectory w_1

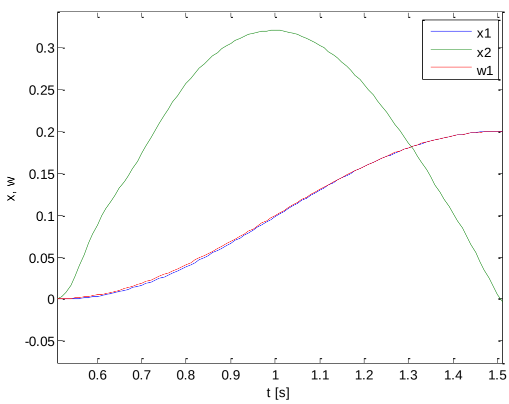


Fig.9. Detail of time-responses of arm translation x_1 (x_2 is speed of x_1) for tracking of desired trajectory w_1

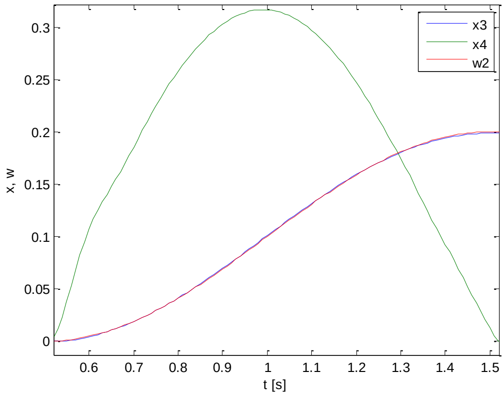


Fig.10. Detail of time-responses of arm rotation x_3 (x_4 is speed of x_3) for tracking of desired trajectory w_2

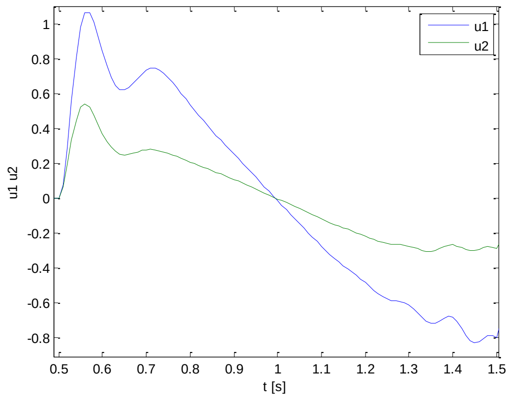


Fig.11. Detail of time-responses of control variables u_1 and u_2 under neural controller.

Conclusions

The new hybrid intelligent control methods based on neural-genetic approach presents an efficient tool for handling plants with complex dynamics as well as unstable inverse systems, time-varying time delays, occasional open-loop instability, plant model miss-matches, different uncertainties, etc. Neural controllers are able to provide high performance in control of non-linear systems. Hybrid soft computing methods based on genetic algorithms are an efficient means for neural controller parameters computation. The obtained numerical and graphical control results demonstrated in paper demonstrate that the hybrid ANN-GA control approach is well formulated and can be effectively implemented to control for robot.

Acknowledgements

The work has been supported by the grant agency VEGA no.1/0937/14.

References

[1] Chaozing, Z. (1998). Control and Dynamic System. Academic Press. New York.

[2] Z. Dideková, S. Kajan, Neural Control of Non-Linear Processes Designed by Genetic Algorithms, In: ELITECH `09 : 11th Conference of Doctoral Students. Bratislava, STU in Bratislava FEI, 2009. - ISBN 978-80-227-3091-4.

[3] R. Gašparík, Optimálne riadenie manipulátora, Diplomová práca, FEI STU Bratislava, 2007

[4] A. Jadlovská, Modelovanie a riadenie dynamických procesov s využitím neurónových sietí (Edícia vedeckých spisov. FEI TU Košice, ISBN 80-8894122-9, 2003, in Slovak language).

[5] A. Jadlovská, Using Forward and Inverse Neural Models for Solving Optimal Tracking Problem of Non-Linear System, Journal of Electrical Engineering, Vol. 55, No. 5-6, 2004, pp. 150-155

[6] I. Sekaj, Genetic Algorithm Based Controller Design, In: 2nd IFAC conference Control System Design'03, Bratislava, 2003.

[7] M. T. Hagan, M. B. Menhaj, Training Feedforward Networks with the Marquardt Algorithm, Submitted to the IEEE Proceedings on Neural Network, 1994.

[8] D. E. Goldberg, Genetic Algorithms in Search, Optimisation and Machine Learning (Addisson-Wesley, 1989).

[9] I. Sekaj, Evolučné výpočty a ich využitie v praxi (Iris, Bratislava, 2005, in Slovak language).

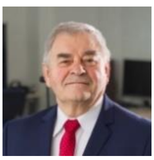
[10] I. Sekaj, Evolutionary Based Controller Design, In: Evolutionary Computation, Book edited by: Wellington Pinheiro dos Santos, ISBN 978-953-307-008-7, pp. 239-260, October 2009, In-Tech, Vienna, Austria. 2009.

[11] R. C. Dorf, Modern Control Systems (Addison-Wesley publishing Company, 5th edition, 1990).

Authors

Ing. Slavomír Kajan, PhD.
Institute of Robotics and Cybernetics,
Faculty of Electrical Engineering and Information Technology,
Slovak University of Technology in Bratislava, Slovakia
slavomir.kajan@stuba.sk

His research interests are artificial intelligence, neural networks and applications of AI methods in medicine, control systems and robotics.



prof. Ing. Štefan Kozák, PhD.
Faculty of Informatics,
Pan-European University in Bratislava, Slovakia
stefan.kozak@paneurouni.com

His research interests include system theory, linear and nonlinear control methods, numerical methods and software for modeling, control, signal processing, IoT, IIoT and embedded intelligent systems for digital factory in industry and medicine.