Neuro-Evolution of Continuous-Time Dynamic Process Controllers

Ivan Sekaj, Ivan Kenický, Filip Zúbek
Institute of Robotics and Cybernetics, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava, Bratislava, Slovak Republic
ivan.sekaj@stuba.sk, ivan.kenicky@stuba.sk, filip.zubek@stuba.sk

Abstract
Artificial neural networks are means which are, among several other approaches, effectively usable for modelling and control of non-linear dynamic systems. In case of modelling systems input and output signals are a-priori known, supervised learning methods can be used. But in case of controller design of dynamic systems the required (optimal) controller output is a-priori unknown, supervised learning cannot be used. In such case we only can define some criterion function, which represents the required control performance of the closed-loop system. We present a neuro-evolution design for control of a continuous-time controller of non-linear dynamic systems. The controller is represented by an MLP-type artificial neural network. The learning algorithm of the neural network is based on an evolutionary approach with genetic algorithm. An integral-type performance index representing control quality, which is based on closed-loop simulation, is minimised. The results are demonstrated on selected experiments with controller reference value changes as well as with noisy system outputs.

Keywords: Continuous-Time Controller, Non-linear Dynamic System, Artificial Neural Network, Genetic Algorithm-Based Learning, Control Performance.

1 Introduction

Artificial neural networks (ANN) are used in many practical application domains as means for classification, approximation and modelling. Beside these traditional implementations, they can be used also as controllers in the control engineering domain. Under controllers we understand generators of signals for controlling continuous-time dynamic systems, dynamic event systems, logical systems, etc., which are based on feedback information, with the aim to reach desired behaviour of the considered system. The advantage of using ANN in comparison to several other controller design approaches is the ability to deal with non-linear systems, systems with complex internal structure and complex behaviour. The obvious way in utilising ANN in most of the applications is the use of known input/output data which were obtained during observation of the system behaviour during its normal operation in history (supervised learning, reconstruction of the reality, imitation of other existing control systems). But obtaining input/output data of a future control process is usually not possible. The (sub)optimal behaviour of the designed controller is a-priori unknown. Known is only the desired closed-loop behaviour, or the desired behaviour of the controlled object respectively, which can be described by some cost function or the desired trajectory.

The aim of this project is to design an ANN-based controller of continuous-time dynamic processes. There exists a wide spectrum of approaches for designing continuous-time controllers and most of them do not use ANNs. Our goal is to introduce an evolutionary-based unsupervised way of designing a direct ANN controller using neuro-evolution. The controller is represented by an MLP-type ANN and the learning approach uses a Genetic Algorithm-based optimisation of the desired closed-loop performance index.

There are several authors who deal with controlling systems using ANNs. Articles [1, 6, 11] mention among several others the use of ANN for replacement of the very popular PID controllers. The neuro-evolution learning approach in general was reported by many authors for optimizing of the ANN architecture and/or the synaptic weights [12, 4, 3, 10, 16, 17]. Neuro-evolution as a learning approach has been used for solving various practical control implementations in mobile robotics and autonomous driving [5, 15].

In Section 2 of this article, we explain the architecture of the proposed ANN controller and its inputs and outputs. In Section 3 the learning approach based on genetic algorithm is described. Section 4 shows experimental results of the evolution-based design in closed-loops with controller reference value changes as well as with noisy system outputs.

2 The ANN-Based Controller Architecture

The most common type of controller used in practice to control dynamic systems is the PID controller. Its
The control algorithm of the neuro-controller contains following two steps in each sampling period: 1. measuring of the controlled output $y(k)$ in step $k$ and preprocessing of the input signal vector $X$ consisting of the 7 input signals

$$X = \{y, \Delta y, \Delta^2 y, \Delta^3 y, e, se, \Delta u\} \quad (5)$$

2. signal propagation over the network and calculation of the new controller output $u(k)$.

### 3 The Neuro-Evolution-Based Controller Design

In order the network to be able to calculate the required output and to obtain the required closed-loop dynamic behaviour it must be correctly parametrized. The goal of the learning process is to find such parameters of the ANN, which minimise the control performance index of the closed-loop. The first considered performance index is a simple integral form criterion $IAE$ – integral of the absolute control error

$$J_{IAE} = \int_{T_1}^{T_2} |e(t)| dt \quad (6)$$

where $T_1$ is the start time end $T_2$ is the stop time of closed-loop simulation. To dump oscillations of the system output an extended performance index can be considered in form

$$J = \int (\alpha |e| + \beta |e'|) dt \quad (7)$$

where $e'$ is the first derivative (or difference) of the control error. Note, that several other types of performance indices can be used for reaching various goals.
as explained in [13, 14]. The ANN parameters are the items of the: 1. weight matrix $W_1$ (weights of the fully connected connection between the input vector $X$ and first hidden layer of neurons), 2. vector of biases of the first hidden layer $B_1$, 3. weight matrix $W_2$ (weights of the fully connected connection between the first and second hidden layer of neurons), 4. vector of biases of the second hidden layer $B_2$, 5. weight matrix $W_3$ (weights of the fully connected connection between the second hidden layer of neurons and the single output neuron).

All these parameters are parts of each chromosome (individual) of the population of the genetic algorithm which is in form

$$ch = \{W_1, B_1, W_2, B_2, W_3\}$$

$$= \{w_{1,1,1}, \ldots, w_{1,N_1,N_1}, b_{1,1}, \ldots, b_{1,N_1}, w_{2,1,1}, \ldots, w_{N_1,N_2}, b_{2,1}, \ldots, b_{2,N_2}, w_{3,1}, \ldots, w_{3,N_2}\}$$

where $M_1$ is the number of inputs (length of the input vector $X$), $N_1$ is number of neurons in the first hidden layer, $N_2$ is number of neurons in the second hidden layer.

As already mentioned, while searching for the optimal solution the genetic algorithm (GA) is minimising the cost function $fitness(ch)$

$$ch_{opt} = \arg \min fitness(ch)$$

The used genetic algorithm consists of following steps:

1. Random initialization of population of $N_{pop}$
2. Fitness function evaluation of each chromosome of the population - simulation of the closed-loop process and calculation of the fitness function.
3. If the predefined number of generations is reached, or the required terminating conditions are reached then finish, otherwise continue to step 4.
4. Selection of 40% parents1 for crossover, selection of 40% parents2 for mutation. Stochastic Universal Sampling Selection is used in both cases [2, 8, 9]. Selection of 2% of best chromosomes and selection of 18% random chromosomes which all will survive without modification.
5. Modification of parents1 by 1-point crossover = children1. Modification of parents2 by the mutation = children2. Mutation rate is 0.1 (10% of genes in population are mutated).

4 Experimental Results

The controlled dynamic system $S$ is described by the differential equation

$$a_2 \Delta^2 y + a_1 \Delta y + a_0 y + a_3 y^3 - b_1 \Delta u - b_0 u = 0;$$

$$a_0 = 1; a_1 = 4; a_2 = 1; a_3 = 0.25; b_0 = 1; b_1 = 0.1$$

(10)

In the first experiment (Fig. 5) we compared two performance indices. The first is the simple IAE criterion according to equation (6) and the second is a combined criterion in form (7) with $\alpha = 1, \beta = 1$. The work was programmed in MATLAB [7]. In the training scenario several steps of reference signal were performed (black line), the neuro-controller closed-loop output with IAE criterion (6) is fast, but with oscillations and overshoot (green) and the aperiodic response, which is slower and damped was obtained using minimization of the second combined criterion (7) (blue). The detail of this experiment is in Fig. 6.
The next experiment compares the PID controller with the neuro-controller. The PID was designed using genetic algorithm with the same criterion (7) $\alpha = 1, \beta = 1$. PID controller does not achieve good performance on controlling the non-linear system. In Fig. 7 and 8 the training scenario is depicted, which was used during the learning phase of the controller design. In Fig. 9 the results of the testing scenario are depicted. The testing scenario was not known during training process. In Fig. 10 the control value of this experiment is shown. In the last experiment time-responses of the controlled system with additional noise on its output $y$ with amplitude 1% of the controlled range during test scenario using PID controller and the neuro-controller are compared (Fig. 11).

5 Conclusion

An unsupervised-type learning procedure of ANN based on genetic algorithm was used for design of a neuro-controller of a non-linear dynamic system. This approach can achieve good results which highly outperform the results of a linear PID controller. In this project the genetic algorithm was used for the design of parameters of an a-priori defined architecture of the neural network controller. Feed-forward ANNs have been considered here. According to our experience the neuro-evolution of controllers is an efficient approach for solving a wide spectrum of controller design problems in continuous-time system control for non-linear dynamic systems, systems with complex behaviour as well as systems with multiple inputs and outputs.

References

Figure 11: Time responses of the controlled system with additional noise on its output $y$ (with amplitude $1\%$ of the controlled range) during test scenario using PID controller (green) and the neuro-controller (blue).


