POSMEC 2016 – Postgraduate Symposium in Mechanical Engineering
Department of Mechanical Engineering – Federal University of Uberlandia
December, 1st to 2nd, Uberlandia, Minas Gerais, Brazil

TWO-WHEELED VEHICLE CONTROL USING NEURAL NETWORK AND DIFFERENTIAL EVOLUTION

Bruno Luiz Pereira, Universidade Federal de Uberlândia, brunolp.meca@gmail.com
Marcus Romano S. B. Souza, Universidade de Patos de Minas, mrsbs.mecatronica@gmail.com
Rosiane R. Rocha, Instituto Federal do Espírito Santo, rosiane.rocha@ifes.edu.br
José J. P. Z. S. Tavares, Universidade Federal de Uberlândia, jean.tavares@ufu.br

Abstract: *This paper aims to present the obtainment of a neural network model of a two-wheeled vehicle. It is trained given a data set extracted during the movement of the robot controlled by an experimentally tuned proportional integral derivative controller. This work also shows the simulation of the predictive control to be applied on it based on the model and the modified differential evolution algorithm developed.*

Keywords*: Modified Differential Evolution; Neural Network; Two-Wheeled Vehicle; Predictive Control.*

1. INTRODUCTION

The study object of this paper is a two-wheeled vehicle that can be modeled as an inverted pendulum, for being an inherently unstable and nonlinear system that requires a control action to keep it upright.

The study of control methodologies applied to inverted pendulum is widely performed due to the characteristic of system nonlinearities, and can therefore be applied in lifting benchmarks relating to the efficiency in solve nonlinear problems from the use of different types of controllers.

With the increase in processing capacity of microcontrollers and prototyping platforms occurred in recent years, it becomes feasible to make use of techniques such as optimization theory and intelligent control to solve control problems. These methods replace the classic ones, which often become ineffective as there is the increase of processes complexity, like for example, in multivariable control chains with coupling between the variables, presence of strong nonlinearities and fast variations of process parameters (Silva, 2006).

Among the known techniques of optimization, it´s chosen the differential evolution (Chen et al., 2009) due to its relative simplicity of implementation and speed in achieving the overall optimum value compared to other evolutionary methods. As for the intelligent control techniques, based on observation of how the living beings, in particular humans, behave towards certain type of problem and how they solve them (Silva, 2006), it opts for the neural network given its ability to model nonlinear systems with time variant parameters.

The two-wheeled vehicle, which techniques are applied, was designed in Souza et al. (2015), in which details the structure of the robot, as well as experimental tuning methods of the proportional integral derivative controller (PID) used for stabilize the system and allow the obtaining of the experimental data used as training patterns for the developed artificial neural network.

2. THEORETICAL FRAMEWORK

2.1. Differential Evolution

The differential evolution is an algorithm that, although evolutionary, has no inspiration in natural processes. This method has a search engine that is based on differential mutation operator (vector sum), Eq. (1):

$$\vec{v}_i = \vec{x}_{r1} + F(\vec{x}_{r2} - \vec{x}_{r3}) \tag{1}$$

where is an subject (or vector) of the population, and are subjects chosen at random within the same, and F is perturbation factor (which usually has values within the range [0,2]).

The subject generated through this scheme (Eq. 1) is evaluated by the objective function and may even replace a subject with worst value of objective function in the current generation or in subsequent generations (Malagoli et al., 2014).

2.2. Neural Network

An artificial neural network (ANN), Fig. (1), is a mathematical structure consisting of a finite number of individual units, also known as neurons, arranged in layers (Silva, 2006).

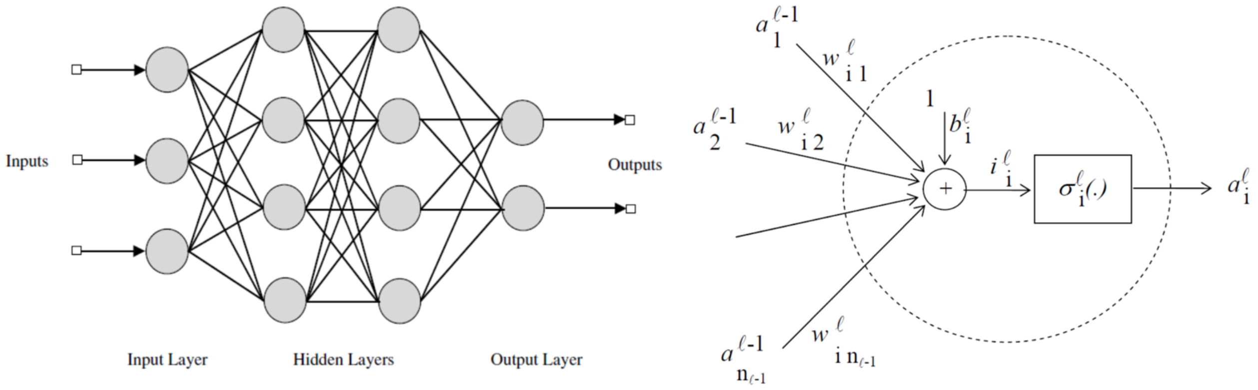

Figure 1. Architecture representation of an artificial multilayer neural network and a scheme of a neuron *i* of layer *l* (MHRD, 2016; Silva, 2006).

The output a_i^l of each neuron i of layer l is given by Eq. (2), and σ the neural network activation function, whose output is generally limited to the range [-1,1] and is responsible for providing the mathematical structure the desired characteristic of nonlinearity (Haykin, 1994).

$$a_i^l = \sigma_i^l(i_i^l) \tag{2}$$

where

$$i_i^l = \sum_{j=1}^{n_{l-1}} w_{ij}^l \, a_j^{l-1} + b_i^l \tag{3}$$

The parameters w_{ij}^l and b_i^l of Eq. (3) are respectively one of the weights and the bias value associated with the neuron i. Both are determined by a method of optimization that seeks to minimize the squared error, defined as the square of the difference between the values shown by the neural network during training, and estimated by the same. In this work, the method used for ANN training is the differential evolution.

3. MATERIALS AND METHODS

The vehicle in this study was the same as in Souza, et al. (2015) and has a MPU 6050 accelerometer, which measures the angular position (pitch angle) of the vehicle relative to its equilibrium position. Figure (2) shows the vehicle used and a simplified geometric model where the pitch angle (θ) and the length (l) of the vehicle are shown.

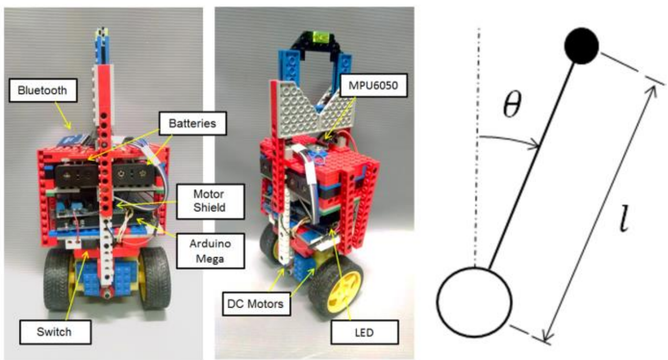

Figure 2. Vehicle mounted with the position of each electronic module and its simplified geometric model (Souza, 2015).

To make it possible to obtain data for training the neural network used in the inverted pendulum, it is necessary that the robot has a control system to ensure that this oscillate within its operating range. For this, we used the PID controller generated by the experimental tuning method which obtained the best results in Souza, et al. (2015), the AZNPID method. The data collected for this work were the control action generated by the PID controller and the pitch angle corresponding to this control action.

The execution of the simulations was performed on a Avell brand computer, model B154, Windows 10 Home Single Language 64-bit operating system, Intel (R) Core (TM) i7-3630QM CPU at 2.40GHz, 8.00 GB of memory RAM, MATLAB R2014b software.

For the realization of the hypothesis tests, the t Student's distribution with 8 degrees of freedom was used, it was considered the normally distributed and homocedastic populations.

4. DEVELOPMENT

The development of this work occurred in three stages, namely: obtaining and training artificial neural network to be used as the identification model; the design and implementation of a predictive controller based on the identification model and the modified differential evolution algorithm, MDE; and, finally, the comparison of the response of the controller with a PID controller whose parameters have been obtained in Souza, et al. (2015).

4.1. Development of the Neural Network

The developed neural network is characterized by having feedback input and being second order, therefore, it follows that the output network $\hat{y}(n+1)$ is given by:

$$\hat{y}(n+1) = f(y(n)), y(n-1), u(n+1), u(n), u(n-1) \tag{4}$$

Where the output variable y is the pitch angle of the inverted pendulum, and the control action u, belonging to the range [-1,1], is related to the voltage to be supplied to the motors of the robot.

The network developed features 2 hidden layers with 4 neurons each, plus 1 output layer with 1 neuron, thus totaling 9 neurons with weights and bias to be determined.

The activation function $\sigma(i)$ of the neural network is given by the expression:

$$\sigma(i) = \frac{1-e^{-i}}{1+e^{-i}} \tag{5}$$

The training of the neural network was carried out using the method of differential evolution, whose parameters are shown in Tab. (1).

Table 1: Algorithm Parameters

Parameter	**Value**
Maximum number of iterations	1000
Individuals number	60
Probability of occurrence of vector sum	0.95
Probability of mutation of vector	0.01
F	0.4
Lower limit of w and b	-2.5
Upper limit w and b	2.5
Initial population	Evenly distributed between -2.5 and 2.5

4.2. Development of the Predictive Controller

The predictive controller used at work, predictive evolutionary differential search controller with model based on neural networks - NDESP, is based on the estimation of the future state through the designed neural network, then the estimation of control action required to minimize the process error, ie, approach to reference value for the pitch angle.

The minimization of the error is given each time step by a differential evolution algorithm where the initial population of individuals presents the limits of the range control action u ([1,1]). The value of u calculated in last iteration, and the remaining individuals are valued following a normal distribution with standard deviation of 0.5 and mean also equal to the value u calculated in last iteration. The parameters of the algorithm are shown in Tab. (2).

Table 2. Parameters Algorithm

Parameter	Value
Maximum number of iterations	10
Individuals number	10
Probability of occurrence of vector sum	0.95
Probability of mutation of vector	0
F	0.4
Lower limit of w and b	-1
Upper limit w and b	1
Error tolerance	0.1

4.3. Simulation

In order to check the developed controller's ability to solve the proposed control problem, a simulation is idealized in that compares the system performance when using the NDESP controller and when using the PID controller with certain parameters in Souza, et al. (2015).

The simulation is based on assumptions and considers the following characteristics:

- Simulate up to 20 seconds of operation of the inverted pendulum;
- Sampling time equal to 0.02 s;
- The values virtually extracted of the sensor, $Z(n+1)$, are given by the equation:

$$Z(n+1) = \hat{y}(n+1) + \sigma_{st} N_1(0,1) + \sigma_{se} N_2(0,1) \tag{6}$$

where $\hat{y}(n+1)$ is given by the designed neural network, σ_{st} is the standard deviation value associated with the estimation error of the state, σ_{se} with respect to the standard deviation associated with the measurement error, and $N_i(0,1)$, $\mathrm{i} = 1,2$ are random variables normally distributed with zero mean and unit standard deviation;

- The imposed value of σ_{se} is 0.1;
- The parameter σ_{st} is assumed to be equal to 0.5 the simulation time t between 6 s to 12 s, and equal to 0.1 for the remainder of the simulation.
- The initial value of the pitch angle of the inverted pendulum is 4 degrees.

The variables to be considered in the simulation for each controller are the simulation runtime and the mean square error value associated with the pitch angle.

5. RESULTS AND DISCUSSIONS

5.1. Neural Network Training

The developed neural network presented the percentage value of 90.52%, thus indicating that adjusted satisfactorily to the experimental data extracted from the inverted pendulum. Figure 3 is a graph comparing the experimental data (in red) to the values estimated by the ANN (in blue) and Fig. (4) shows the evolution of the mean square error associated with the fit of the model to the experimental data.

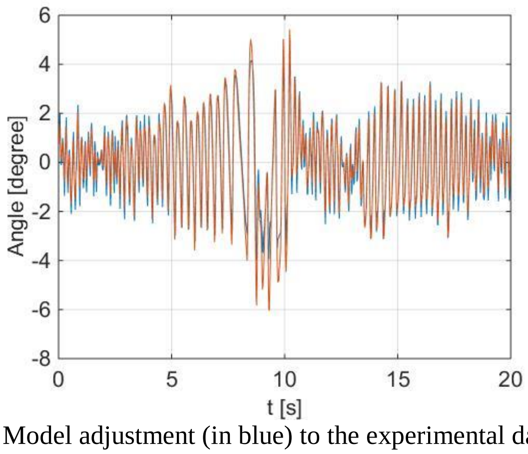

Figure 3. Model adjustment (in blue) to the experimental data (in red).

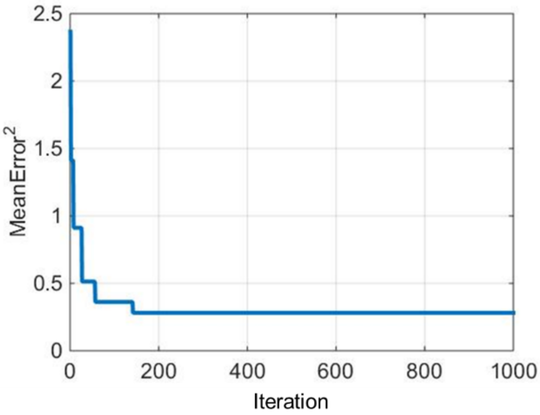

Figure 4: Evolution of mean square error due to the iteration of differential evolution.

5.2. Results of the Simulation

The simulation was replicated 10 times, 5 to each controller, and results of simulation runtime and the mean square error associated with the pitch angle are indicated respectively in Tab. (3) and Tab. (4).

Table 3. The run time of simulation.

Replication	NDESP Controller [s]	PID Controller [s]
1	6.28	0.71
2	6.84	0.72
3	5.87	0.73
4	6.57	0.75
5	6.07	0.73
$\bar{X}$	6.33	0.73
s	0.39	0.02

From the assumptions below:

- Null Hypothesis: The simulation runtime using the NDESP controller is equal to 8 times the time for its execution using the PID controller.
- Alternative Hypothesis: The simulation runtime using the NDESP controller is greater than 8 times the time for its execution using the PID controller.

With 95% confidence, the hypothesis test rejects the null hypothesis.

Table 4. Mean Square Error

Replication	NDESP Controller [º]	PID Controller [º]
1	0.13	1.70
2	0.13	1.75
3	0.14	1.66
4	0.12	1.72
5	0.12	1.65
$\bar{X}$	0.13	1.70
s	0.01	0.04

Figure 5, related to NDESP controller, indicates the control action applied virtually to the robot at each iteration; and the value of the pitch angle (in blue), and its reference (in red).

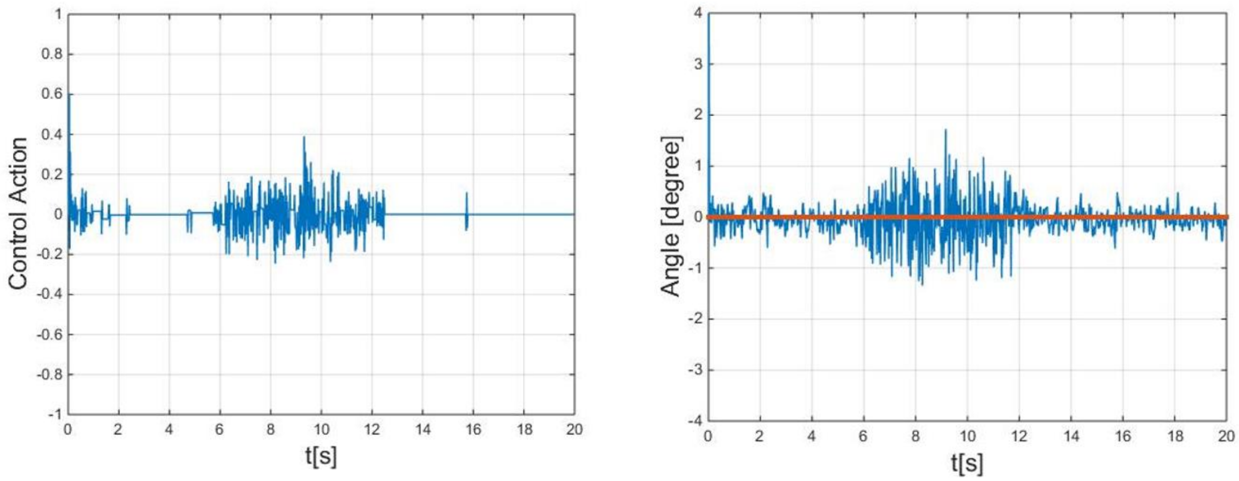

Figure 5: Robot control action and pitch angle as a function of time using the NDESP controller.

Figure 6 indicates the control action, and the value of the pitch angle (in blue), and its reference (in red) when using the PID controller with the same parameters used in the experimental test. It can be seen that the simulation could get qualitatively similar behavior to that observed during the actual movement of the robot with respect to the pitch angle, Fig. (3).

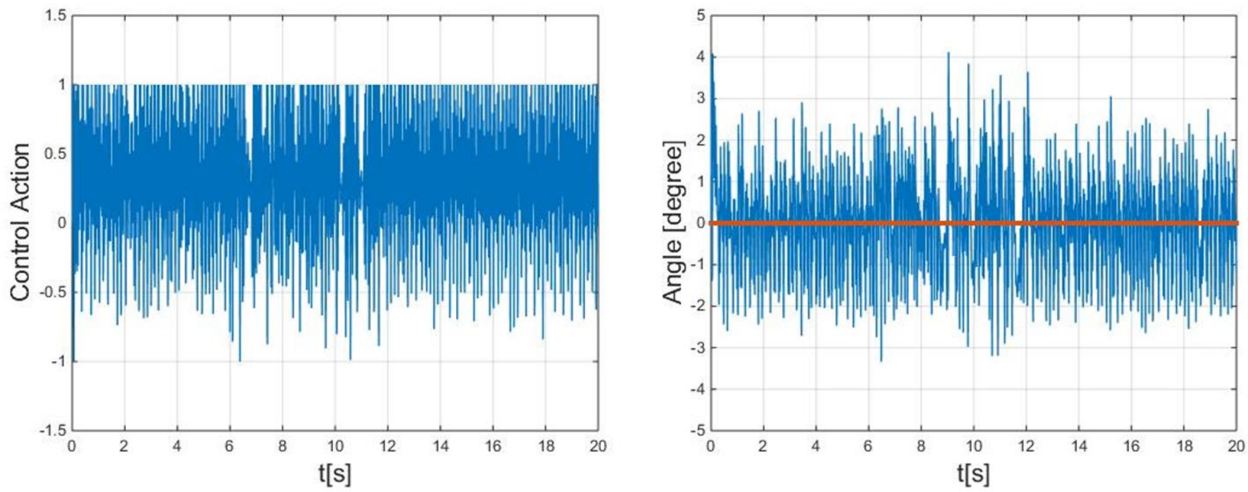

Figure 6: Robot control action and pitch angle as a function of time using the PID controller.

From the assumptions below:

- Null Hypothesis: The root mean square error associated with the pitch angle in the simulation multiplied by 12 using the NDESP controller is equal to the mean squared error when using the PID controller.
- Alternative Hypothesis: The root mean square error associated with the pitch angle in the simulation multiplied by 12 using the NDESP controller is less than the mean square error when the PID controller is used.

With 95% confidence, the hypothesis test rejects the null hypothesis.

6. CONCLUSIONS

The developed neural network can have their weights adjusted in order to satisfactorily predict the behavior of the pitch angle of the inverted pendulum, given that the model based on neural networks has fit to the data extracted during the movement of the robot with explainable percentage of 90.52%.

The ANN adjustment capability to the data also suggests the effectiveness of the differential evolution algorithm to minimize the mean square error value associated with the fit of the model to the experimental data.

The simulation results indicate the efficiency of NDESP controller developed, as in the simulation the same obtained with 95% confidence a root mean square value of the pitch angle of at least 12 times lower than that obtained by the PID controller with same parameters as those used in the actual movement of the robot.

The control algorithm runtime however, must be checked for hardware specification capable of processing in a timely manner itself, as was verified with 95% confidence that the NDESP controller is at least 8 times computationally expensive to perform the simulation of the PID controller.

7. FURTHER WORKS

Implementation of NDESP controller in inverted pendulum, obtaining the code processing time and the value of the mean square error related to the pitch angle, and comparative analysis of these data with those obtained by simulation.

8. ACKNOWLEDGMENT

The authors gratefully acknowledge the financial support of CNPq, CAPES and FAPEMIG, and also FEMEC and MAPL – Manufacturing Automated Planning Lab – at technical support.

9. REFERENCES

Chen, C. H., Lin, C. J. and Lin, C. T. “Nonlinear System Control Using Adaptive Neural Fuzzy Networks Based on a Modified Differential Evolution”. IEEE Transactions on Systems, Vol. 39, 2009, pp.459-473.

Haykin, S. “Neural Networks, a Comprehensive Foundation”. Pearson Prentice Hall, 2nd ed., 1994.

Malagoli, J. A., Camacho, J. R. and Luz, M. V. F. “Projeto de Rotor de Máquina de Indução Trifásica Usando o Algoritmo de Evolução Diferencial”. 11º Congresso Brasileiro de Eletromagnetismo, 2014, 6 p.

Ministry of Human Resource Development. “Artificial Neural Networks”. 15 September 2016. < http://cse22-iiith.vlabs.ac.in/exp4/index.html>.

Silva, G. V. M. “Controlo Não Linear”. Lisboa, 2006.

Souza, M. R. S. B., Murofushi, R. H., Tavares, J. J. P. Z. and Ribeiro, J. F. “PID Tuning for the Pitch Angle of a Two-Wheeled Vehicle”. 12th Latin American Robotics Symposium and 3rd Brazilian Symposium on Robotics, 2015.

Souza, M. R. S. B. “Projeto, Construção e Sintonia Experimental de Controlador PID Aplicado a um Robô de Duas Rodas”. Universidade Federal de Uberlândia, 2015.

10. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.