Design and validation of Real Time Neuro Fuzzy Controller for stabilization of Pendulum-Cart System

Tharwat O. S. Hanafy
Computers and Systems Engineering Department, Al-Azhar University, Cairo, Egypt, s_ewiss@yahoo.com

Abstract: This paper presents an application of how to design and validate a real time neuro fuzzy controller of complex a nonlinear dynamic system using the Matlab-Simulink Real-Time Workshop environment. Once the controller is obtained and validated by simulation, it's implemented to control the pendulum-cart system. Design of a neuro fuzzy controller is considered in this work because of its insensitivity to disturbances and uncertainties of model parameters. The design and optimization process of neuro fuzzy controller are based on an extended learning technique derived from adaptive neuro fuzzy inference system (ANFIS). The design and implementation of this pendulum-cart control system has been realized under MATLAB/SIMULINK environment. The experimental results demonstrate the efficiency of this design procedure and the ensured stability of the system.
[Tharwat O. S. Hanafy. **Design and validation of Real Time Neuro Fuzzy Controller for stabilization of Pendulum-Cart System**. Life Science Journal. 2011; 8(1):52–60] (ISSN: 1097–8135).

Keywords: design and validate; real time neuro fuzzy controller; nonlinear dynamic system;

I. Introduction:

Fuzzy controllers have been widely used in many control systems applications. Besides being convenient for qualitative system modeling, they are very simple conceptually [1]. They consist of an input stage, a processing stage, and an output stage. The input stage maps sensor or other inputs, such as switches, thumbwheels, and so on, to the appropriate membership functions. The processing stage invokes each appropriate rule and generates a result for each, then combines the results of the rules. Finally, the output stage converts the combined result back into a specific control output value. The most common shapes of membership functions are triangular, trapezoidal and bell curves. From three to seven curves are generally appropriate to cover the required range of an input value, or the "universe of discourse" in fuzzy jargon. The shape is generally less important than the number of curves and their placement.

Design of a fuzzy controller requires more design decisions than usual, for example regarding rule base, inference engine, defuzzification, and data pre- and post processing [2]. This paper describes the design decisions related to closed-loop neuro fuzzy controller of the pendulum-cart system. The main problems in neuro fuzzy controller design are the inference of an initial rule base and in particular the optimization of an existing rule base.

Many researchers addressed the design problem of neuro fuzzy controller. Nauck et al introduced the design of neuro fuzzy controller using backpropagation algorithm [11]. They also, presented their learning algorithm for neuro fuzzy environment NEFCON-I under matlab/simulink [22].

The pendulum-cart system is an interesting nonlinear dynamic model which has been extensively studied by control community. It represents many real world systems, such as crane at shipping port and space mission launchers. The selection of a control strategy for stabilization of such systems is a difficult design task. Optimality of the control strategy and its robustness are the main design criteria to be considered. However, due to the presence of disturbances and model parameter uncertainties, a robust behavior is more important than the optimal character of the control strategy. The efficiency of neuro fuzzy techniques to reduce disturbances makes it an excellent candidate to design a closed loop controller. With an efficient learning method, the parameters of the neuro fuzzy controller can be optimally designed. The design and validation of a neuro fuzzy controller should be assisted by a software environment that can provide the designer with functions of fuzzy logic systems and targeting a real time application.

The Fuzzy Logic Toolbox of the MATLAB technical computing environment is an efficient tool for designing systems based on fuzzy logic. The toolbox provides many functions which allow control engineers to develop and analyze fuzzy inference systems, to develop adaptive inference systems, and perform fuzzy clustering [3]. Its Graphical User Interfaces (GUIs) simplifies the steps of neuro fuzzy inference system design. Alternatively, Simulink provides fuzzy inference blocks in order to simulate the fuzzy systems within a comprehensive model of the entire dynamic system [7]. From Simulink, C code can be automatically generated for use in embedded applications that include neuro fuzzy logic [4].

This paper describes the design decisions related to closed-loop neuro fuzzy controller of the pendulum-cart model. We propose an efficient design and rule learning procedure of the neuro fuzzy controller. We also present the experimental results on the design and implementation of real time neuro fuzzy control system under matlab/simulink computation environment.

The paper is organized as follows. Sections I is this introduction. In Section II, we present an analysis of the control system development under Matlab/Simulink Environment. Section III describes the pendulum-cart set-up. Section IV presents the control algorithm, the rule base learning and its optimization. Section V evaluates the implementation and presents the experimental results of neuro fuzzy algorithm. Finally, our conclusions are drawn in Section VI.

II. Control System Development under Matlab/Simulink Environment

This work has been developed using the Mathworks tools. These tools are in varying widespread use across a number of industries for control system development [3]. Fig. 1 shows how the various elements of the MATLAB environment can be linked together to provide an integrated set of tools for control system design and experimental validation [3]. The use of these standard software tools means that, during the controller design stage, the designer only needs to model the process using the graphics tools available in Simulink without being concerned with the mechanics of communication to and from the device under test.

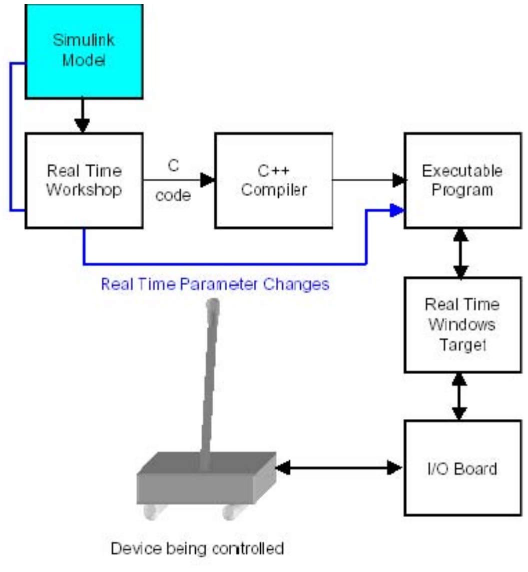

Fig. (1) Control System Development Flow Diagram.

A brief description of these tools and their use in control system development is given in next paragraphs.

Matlab acts as the application host environment in which the other mathworks products run. It provides a sophisticated set of tools for solving mathematical problems in addition there are specialized toolboxes, such as fuzzy logic toolbox which extend the Matlab functions in several different specific application areas.

Simulink is a graphics based system for modeling process, which takes the form of blocks, is fed as input into another block. Blocks perform specialized operations on the data and may be standard blocks from the Simulink library or written by the user where no suitable library blocks exists [3, 4]. Simulink model is passed to real Time Workshop.

Real-Time Workshop -RTW generates optimized, portable, and customizable code from Simulink models. Real time workshop automatically builds a C++ source program from Simulink model.

C++ Compiler compiles and links the code created by Real Time workshop to produce an executable program. The program interfaces to the outside environment via a "Target", in our case Real Time Windows Target.

Real Time Windows Target communicates with the executable program acting as the control program, and interfaces with the hardware device through an I/O board. Real Time Windows Target controls the two-way data or signal flow to and from the model, and to and from the I/O Board. When the program is running, the user may change certain of the parameters in the Simulink model, which are then passed, via Real Time Workshop, to the executable program

III-Pendulum-Cart Set-Up Description

One of the simplest problems in robotics is that of controlling the position of a single link using a steering force applied at the end. Pole-balancing systems are impressive demonstration models of missile stabilization problems [3, 4, 5, 6, 7]. The crane used at shipping ports is an example of non-linear electromechanical systems having a complex dynamic behavior and creating challenging control problems. Mathematically either is just a pendulum in a stable or unstable position. The pendulum-cart set-up consists of a pole mounted on a cart in such a way that the pole can swing free only in vertical plane. The cart is derived by DC motor. To swing and to balance the pole the cart is pushed back and forth on a rail of limited length. The vertical stationary positions of the pendulum (upright and down) are equilibrium positions when no force is being applied. In the upright position a small deviation from it results in an unstable motion. Generally the pendulum control problem is to bring the pole to one of the equilibrium positions and preferably to do so as fast as possible, with few oscillations, and without letting the angle and velocity become too large. After the desired position is reached, we would like to keep the system in this state despite random perturbations. Manual control of the cart-pole system is possible only for simple tasks e.g. for moving the cart from one place on the rail to another. For more complicated tasks (such as stabilizing the pole in an upright position) a feedback control system must be implemented Fig. (2). The purpose of the inverted pendulum control algorithm is to apply a sequence of forces of constrained magnitude to the cart, such that the pole starts to swing with increasing amplitude without the cart overriding the ends of the rail. Firstly the pole is swung up to the vicinity of its upright position and then, once this has been accomplished, the controller maintains the pole vertically and at the same time brings cart back to the center of the rail.

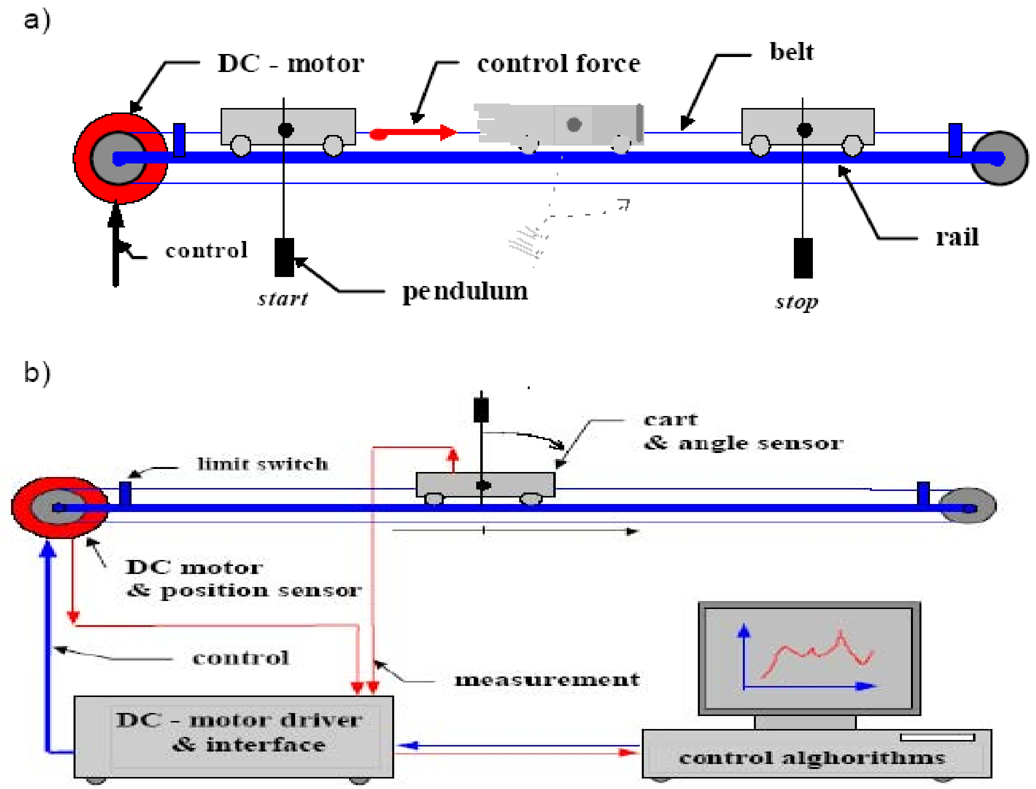

Fig. (2) Pendulum Control System

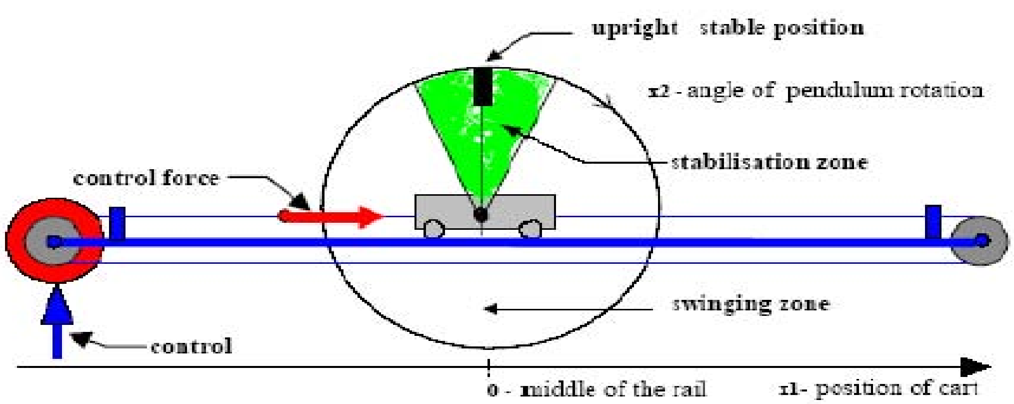

Fig. (3) Activity zones of two control algorithms

The swinging control algorithm is a heuristic one, based on energy rules. The algorithm steers the pole up thus increasing its total energy [3, 4, 19]. There is a trade-off between two tasks: to swing the pendulum to the upright position and to center the cart on the rail. Due to the presence of disturbances and parameter uncertainties, a robust behavior is more important than the optimal character of the control strategy. The switching moments are calculated according to a simple rule. The characteristic feature of control is its “bang bang” character. Swinging up the pole may result in over-reaching the upper unstable equilibrium point. To achieve a “soft” landing in the vicinity of the upright position (“stabilization zone” in Fig. (3), a routine called the “soft landing arbiter” checks whether the kinetic energy of the pole, minus the energy loss due to friction, is sufficient to raise the center of gravity of the pole to its upright position. If the condition is satisfied then the control is set to zero and the "bang-bang" character of the control is finished. After the pole has entered the stabilization zone the system can be treated as linear and the control is switched to the stabilizing algorithm. Due to the limited length of the rail a routine called “length control” is introduced, to reinforce centering of the cart and prevent over-running the edges of the rail. The rule is very simple. When the position given by the parameter "length" is reached, then the maximal force is applied to the cart steering it back away from this position.

III.I System Model

The state of the system is the vector $x = [x_1, x_2, x_3, x_4]^t$, where x_1 is the cart position (distance from the centre of the rail), x_2 is the angle between the upward vertical and the ray pointing at the centre of mass, measured counter-clockwise from the cart ($x_2 = 0$ for the upright position of the pendulum), x_3 is the cart velocity, and x_4 is the pendulum angular velocity. The pendulum rotates in a vertical plane around an axis located on a cart [23]. The cart can move along a horizontal rail, lying in the plane of rotation. A control force u, parallel to the rail, is applied to the cart. The mass of the cart is denoted by m_c and the mass of the pendulum, by m_p. l is the distance from the axis of rotation to the centre of mass of the pendulum-cart system. J is the moment of inertia of the pendulum-cart system with respect to the centre of mass. T_c denotes the friction in the motion of the cart, and D_p is the moment of friction in the angular motion of the pendulum, proportional to the angular velocity: $D_p = f_p x_4$. The force of reaction of the rail V acts vertically on the cart. As the horizontal co-ordinate of the centre of mass is equal to x_1 - lsin x_2 and the vertical to l cos x_2, the motion equations are as follows:

$$(m_c+m_p)(x_1-l \sin x_2)''=F-T_c, \quad (1)$$
$$(m_c+m_p)(l \cos x_2)''=V-(m_c+m_p) g, \quad (2)$$
$$Jx_2''=(u-T_c)\, l \cos x_2 + Vl \sin x_2 - D_p, \quad (3)$$

(.)" denotes the second derivative with respect to time t and (.)’ denotes the first derivative with respect to time t. The first two equations describe the translation of the centre of mass, while the third describes the rotation of the whole system around the centre of mass. After the elimination of V and simple calculations we obtain the state equations (for t >= 0)

$$x'_1=x_3 \quad \textbf{(4)}$$

$$\mathbf{x'_3}=\frac{a(u-Tc-\mu x_4^2\sin x_2)+l\cos x_2(\mu g\sin x_2-f_p x_4)}{J+\mu l\sin^2 x_2} \quad \textbf{(5)}$$

$$\mathbf{x'_2=x_4,} \quad \textbf{(6)}$$

$$\mathbf{x'_4}=\frac{l\cos x_2(u-Tc-\mu x_4^2\sin x_2)+\mu g\sin x_2-f_p x_4}{J+\mu l\sin^2 x_2} \quad \textbf{(7)}$$

Where

$$\mathbf{a}=l^2+\frac{J}{m_c+m_p}, \qquad \mu=(m_c+m_p)l$$

The admissible controls are bounded such that

$$|u(t)| \le M$$

The cart friction T_c in the model is a non-linear function of the cart velocity x_3. As an approximation one can assume $T_c = f_c x_3$. The rail has a finite length and hence the cart position x_1 is bounded: The typical parameters of the cart-pole set-up are given in Table 1.

Table 1. Parameters of the pendulum-cart set-up

name of parameter	value of parameter
track limits	±0.5 m
gravity g	9.81 m/s^2
Distance between mass centre and axis of rotation l	0.017 m
mass of cart m_c	1.12 kg
mass of pole m_p	0.11 kg
magnitude of control force M	17.0 N
moment of inertia of system J	0.0136 kgm^2
friction coefficient of pole rotation f_p	negligible
friction coefficient of cart f_c	0.05 Ns/ m

The model of the pendulum-cart set-up is an example of a SIMO system: a single control input and multi outputs (states) and can be used to demonstrate the advantages of closed-loop control.

III.II Real Time Computer Control

One of the main objectives of this work is the direct implementation of designed neruro fuzzy controller in a real time process. Computer control of a real time process is presented in this section. A block diagram of a computer-controlled process is given in Fig. 4

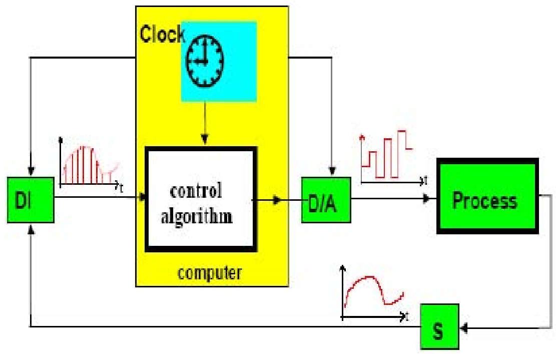

Fig. (4) Computer controlled process

The system contains six blocks: the process, sensors (S), D/A converter, control algorithm, and a clock. The software clock controls the operation of the converters and the control algorithm. The time between successive conversions of the signal to digital form is called the sampling period (T_0). The clock supplies a pulse every T_0 seconds, and the DI supplies a number to the computer every time an interrupt arrives. The control algorithm computes the value of the control variable and sends it as a number to the D/A converter. It is assumed that the D/A converter hold the signal constant over the sampling period; periodic sampling is normally used [3]. An application of the general digital control system schema for pendulum control is given in block diagram form in Fig. 5. Two process states are measured: the cart position x_1 and the pendulum angle x_2. Process states are measured as continuous signals and converted to digital by optical encoders (sensors S1 S2). The reference input (desired value of the cart position x_1) can be generated in a digital form using a desired position generator. The software timer is used to supply interrupts for the system: The basic clock activates the periodic sampling of optical decoders outputs and synchronizes the computation of controller outputs (u) and periodic digital-to-analog (D/A) conversion.

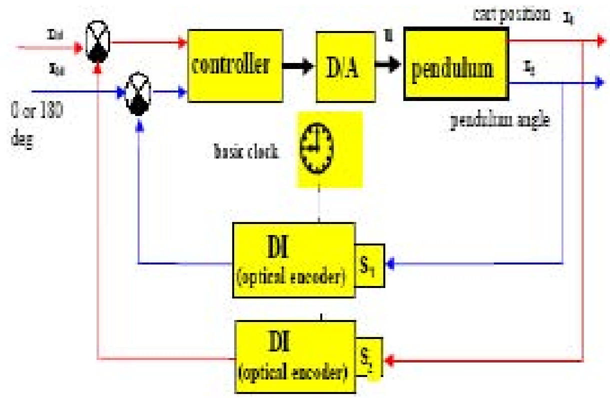

Fig. (5) Digital Control of the Pendulum-Cart System (basic block diagram).

The pendulum-cart system is controlled in real-time. The term "real-time" is often used but seldom defined. One possible definition is [4]: "Real-time is the operating mode of a computer system in which the programs for the processing of data arriving from the outside are permanently ready, so that their results will be available within predetermined periods of time; the arrival times of the data can be randomly distributed or be already determined depending on the different applications." The real-time software for pendulum control is structured around particular internal signals (events) into a set of tasks. Each task implements the processing required by a corresponding event. A task scheduler recognizes the events and activates or suspends the tasks. In the simplest case, when all tasks require processing at the same frequency, a sequential organization of the tasks can be implemented [14]. The time frame of each task is fixed. It is assumed that the longest task job takes no longer than the period of time generated by the software timer.

IV Control Algorithm

The controller in this experimental setup is based on a neuro fuzzy algorithm**.** The inputs of the neuro fuzzy system are pendulum angle, cart position, and the outputs are cart velocity, pendulum velocity as shown in Fig. (6). Fig. (7) Shows the initial membership function for inputs.

IV.I Rule base Learning

The learning process of the ANFIS model can be divided into two main phases. The first phase is designed to learn an initial rule base, if no prior knowledge about the system is available. Furthermore it can be used to complete a manually defined rule base. The second phase optimizes the rules by shifting or modifying the fuzzy sets of the rules. Both phases use a fuzzy error, E, which describes the quality of the current system state, to learn or to optimize the rule base. In this work, we used the 'ANFIS Learning`-Algorithm [9, 10]. This algorithm starts with an empty rule base. An initial fuzzy partitioning of the input and output intervals must be given. The algorithm can be divided into two parts. During the first part, the rules' antecedents are determined by classifying the input values, i.e. finding that membership function for each variable that yields the highest membership value for the respective input value [13, 14]. Then the algorithm tries to 'guesses the output value by deriving it from the current fuzzy error. During the second part, the rule base is optimized by changing the consequent to an adjacent membership function, if it is necessary [15, 16, 19]. Fig. (8) Shows the viewing rules between inputs-outputs. The relations among cart velocity, cart position, and pendulum angle are introduced as shown in Fig. (9).

IV.II Optimization of the Rule base (Implementation)

The aim of the implementation under MATLAB/SIMULINK was to develop an interactive tool for the construction and optimization of a fuzzy controller. This frees the user of programming and supports him to concentrate on controller design. It is possible to include prior knowledge into the system, to stop and to resume the learning process at any time, and to modify the rule base and the optimization parameters interactively. To optimize the rule base we choose the optimization algorithm ANFIS [10, 11, 12].

This algorithm is motivated by the back-propagation algorithm for the multilayer preceptron [8]. It optimizes the rule base by back-propagation of error. A rule is 'rewarded' by shifting its consequent to a higher value and by widening the support of the antecedents, if it's current output has the same sign as the optimal output [20, 21]. Otherwise, the rule is 'punished' by shifting its consequent to a lower value and by reducing the support of the antecedents. The inferred rule base of the system under study has 27 rules.

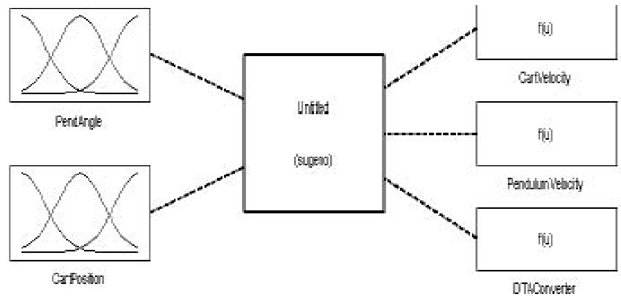

Fig. (6) Relation between Inputs-Outputs

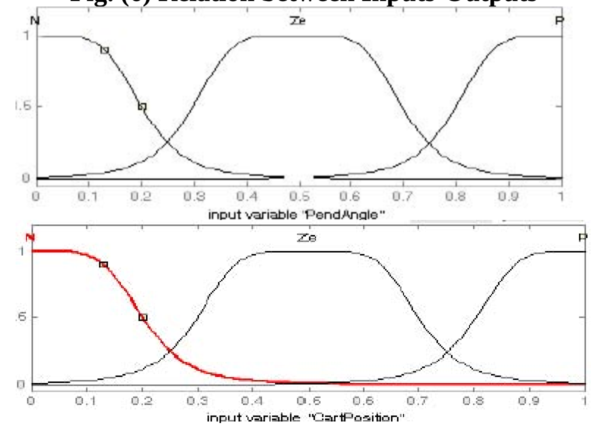

Fig. (7) Initial membership function for inputs

The reduced rule base is presented below.

1- IF (PendAngle is N) and (CartPostion is N) then (CartVelocity is N)(PendVelocity is N)(F is N)

2- IF (PendAngle is P) and (CartPostion is P) then (CartVelocity is p)(PendVelocity is p)(F is p)

3- IF (PendAngle is Ze) and (CartPostion is Ze) then (CartVelocity is Ze)(PendVelocity is Ze)(F is Ze)

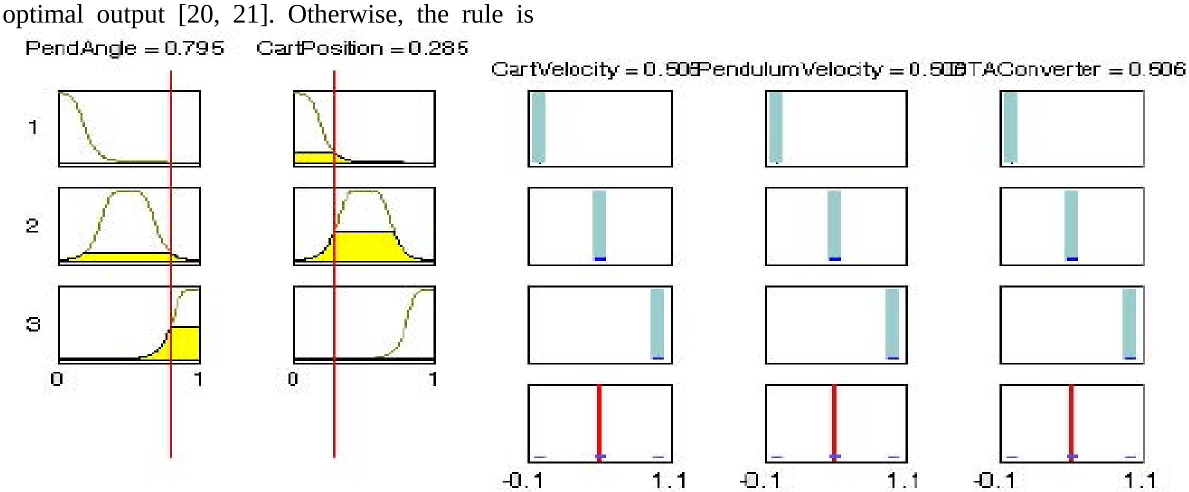

Fig.(8) Visualization of rules between inputs-outputs

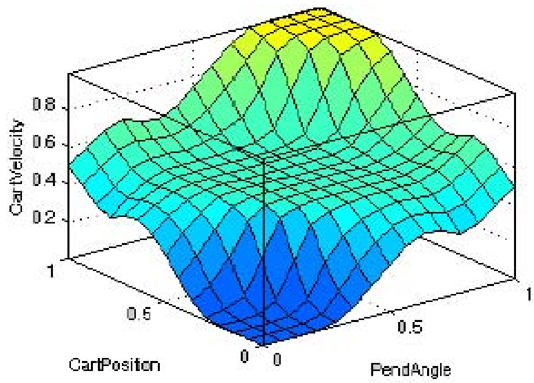

Fig. (9) The relations among cart velocity, cart position, and pendulum angle

V. Experimental Results

To test our approach we conducted experiments on a cart-pendulum hardware interfaced with a neuro fuzzy controller implemented in Matlab/Simulink environment as previously indicated in fig. (5). In these experiments, the cart is driven by DC motor. To swing and to balance the pole the cart is pushed back and forth on a rail of limited length. The neuro fuzzy system is used to stabilize the pendulum. Fig. (10) shows the structure of neuro fuzzy system used in the implementation. The result for this simulation of ANFIS controller system with real time inverted pendulum system is shown in following figures. Fig. (11) Represents the change of cart position with time, in another meaning this figure shows the inverse relationship between the force and stability" The higher the force the lower the stability". Fig. (12) Shows the change of pendulum angle with time, in another meaning this figure shows the direct relationship between the force and the angle of pendulum" The higher the force the higher the angle". The change of cart velocity and pendulum velocity with time is shown in Fig. (13) And Fig (14) respectively. Due to the high force generated from the initial movement it takes few seconds to reach the stability level.

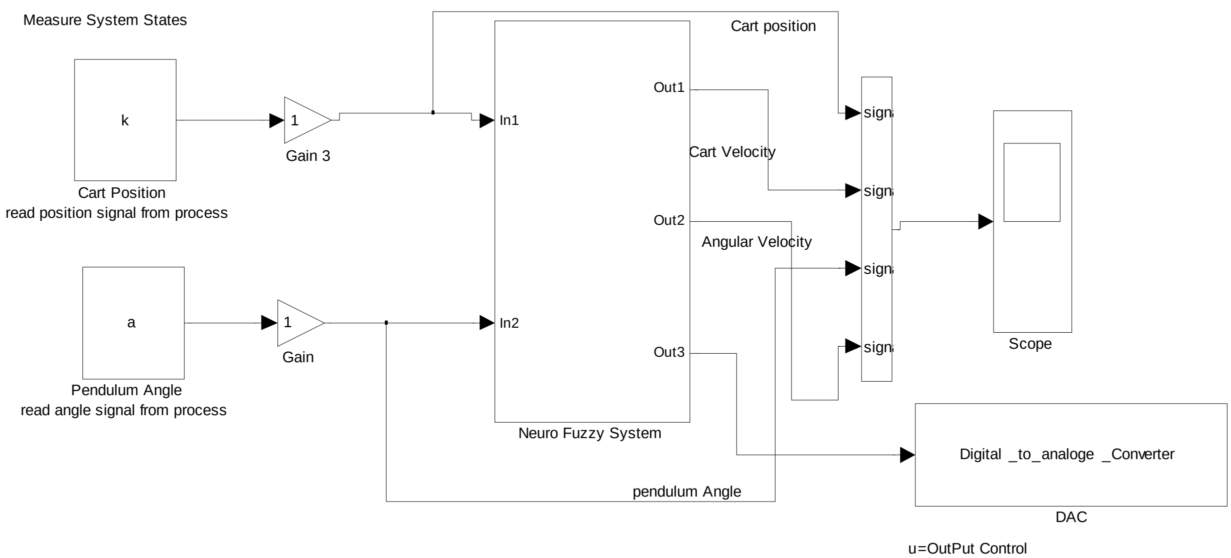

Fig.(10) The Structure of Neuro Fuzzy Simulink System used in the implementation

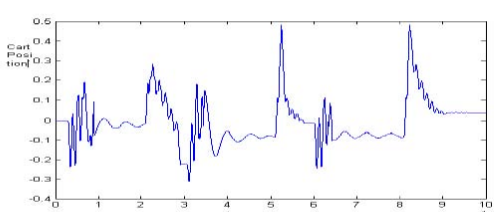

Fig. (11) The change of cart position with time.

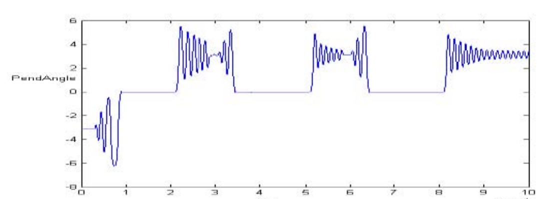

Fig. (12) The change of pendulum angle with time.

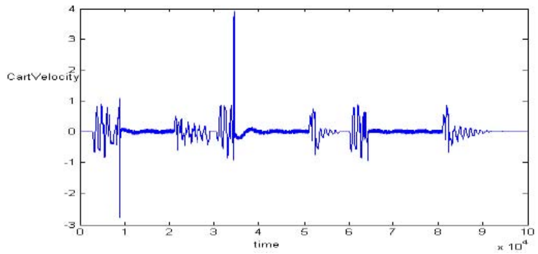

Fig. (13) The change of cart velocity with time

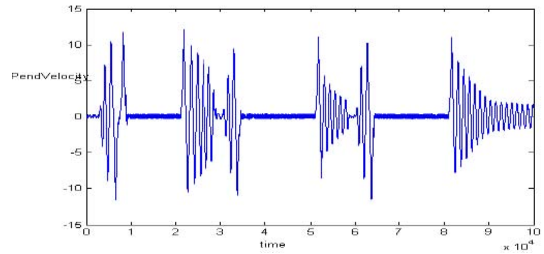

Fig. (14) The change of pendulum velocity with time

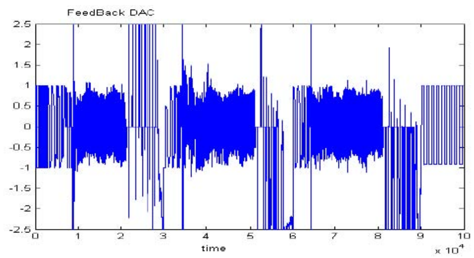

Fig. (15) The output of DTAC and inputs to process with time

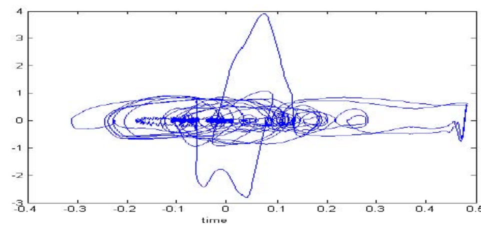

Fig. (16) The relationship between cart position and cart velocity

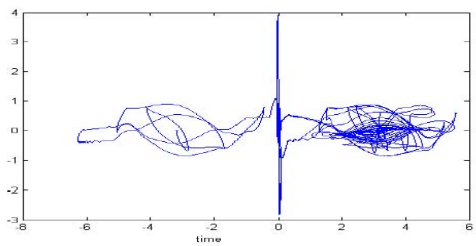

Fig. (17) The relationship between cart position and pendulum velocity

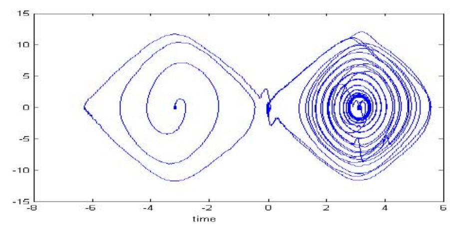

Fig. (18) The Relationship between pendulum angle and pendulum velocity

Then after few seconds the stability tends to be lost, at this moment we should give another force in order to keep the stability of pendulum at the required level. Fig. (15) displays the digital output from the computer which was converted by the digital signal to analog converter (DTAC) using data acquisition card and sending this signal as inputs to the process with time. Fig. (16) to Fig (18) show the relationship among inputs and outputs. We notice from these figures that the high velocity generated from force lead to high change in the pendulum angle and this change tend to be less when the velocity is reduced which is a positive relationship between the velocity and the pendulum angle, and force.

VI-Conclusions:

In this paper we presented the design and optimization process of neuro fuzzy controller supported by learning techniques derived from neural networks (ANFIS). The generation of rule base has been done from input output data. The implementation of this controller has been realized under the MATLAB/SIMULINK. This implementation supports the development of real time process in an easy way. One of the important conclusions in this model is that the stability of the pendulum is negatively related with the force, velocity, and angle. The design and implementation of this pendulum-cart control system has been done under MATLAB/SIMULINK environment. The experimental results demonstrated the efficiency of this design procedure and the ensured stability of the system**.**

Corresponding author
Tharwat E. Alhanafy
Computers and Systems Engineering Department, Al-Azhar University, Cairo, Egypt
s_ewiss@yahoo.com

References:

1. Barto, A.G.; Bradtke, S. J.; Singh, S. P.(1995): Learning to act using real-time dynamic programming, Artificial Intelligence, Special

Volume: Computational Research on Interaction and Agency, 72(1): 81-138, 1995
2. Riedmiller, M., and Janusz, B.(1995), Using Neural Reinforcement Controllers in Robotics, Proceedings of the 8th Australian Conference on Artificial Intelligence,Canberra, Australia, 1995
3. Nürnberger, A., Nauck, D., Kruse, R., Merz, L., A Neuro-Fuzzy Development Tool for Fuzzy Controllers under MATLAB/SIMULINK, In Proc. of the 5th European Congress on Intelligent Techniques & Soft Computing (EUFIT '97), Aachen, Germany, 1997
4. Mamdani, E. H.; Assilian S. (1973): An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller, International Journal of Man-Machine Studies, 7:1-13
5. Barto, A.G., Sutton R. S., and Anderson, C. W., Neuron like adaptive elements that can solve difficult learning control problems, IEEE Transactions on Systems, Man and Cybernetics, 13, 834-846, 1983
6. Lin, C.T., Neural Fuzzy Control Systems with structure and Parameter Learning, World Scientific Publishing, Singapore, 1994
7. Knappe, Heiko (1994): Comparison of Conventional and Fuzzy-Control of Non-Linear Systems
8. D. A., Sofge, D. A., Publ. (1992): Handbook of Intelligent Control. Neural, Fuzzy and Adaptive Approaches, Van Nostrand Reinhold, New York.
9. Nauck, Detlef (1994): A Fuzzy Perceptron as a Generic Model for Neuro-Fuzzy Approaches, In Proc. of the 2nd German GI-Workshop Fuzzy-Systeme '94, München
10. Nauck, Detlef and Kruse, Rudolf (1993): A Fuzzy Neural Network Learning Fuzzy Control Rules and Membership Functions by Fuzzy Error Backpropagation, In Proc. IEEE Int. Conf. on Neural Networks 1993, San Francisco.
11. Nauck, Detlef and Kruse, Rudolf (1996): Designing neuro-fuzzy systems through backpropagation, In Witold Pedryz, editor, Fuzzy Modeling: Paradigms and Practice, pages 203-228, Kluwer Academic Publishers, Boston, Dordrecht, London.
12. Lin, C.T. (1994): Neural Fuzzy Control Systems with structure and Parameter Learning, World Scientific Publishing, Singapore.
13. Nauck, Detlef; Kruse, Rudolf; Stellmach, Roland (1995): New Learning Algorithms for the Neuro-Fuzzy Environment NEFCON-I, In Proceedings of Neuro-Fuzzy-Systeme '95, 357-364, Darmstadt.
14. Kruse, Rudolf; Gebhardt, Jörg; Palm, Rainer (Eds.) (1994): Fuzzy Systems in Computer Science, Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, Braunschweig, Wiesbaden.
15. Riedmiller, Martin; Janusz, Barbara (1995): Using Neural Reinforcement Controllers in Robotics, In Xian Yao, editor, Proceedings of the 8th. Australian Conference on Artificial Intelligence, Singapore, 1995, World Scientific Publishing, Singapore
16. Ahmeda M. A., Saliu, M. O. and AlGhamdi, J.(2005) "Adaptive fuzzy logic-based framework for software development effort prediction", Information and Software Technology 47 (2005) 31–48, 2005
17. Tschichold-Gürman, Nadine (1995): RuleNet - A new Knowledge-based Artificial Neural Network Model with Application Examples in Robotics, Dissertational Thesis, ETH Zürich.
18. Tsukamoto, Y. (1979): An Approach to Fuzzy Reasoning Method, In M. Gupta, R. Ragade and R. Yager, Hrsg.: Advances in Fuzzy Set Theory, North-Holland, Amsterdam
19. Mori, Shozo, Nishihara, Hiroyoshi, and Furuta, Kattsuhisa, "Control of an unstable mechanical system, Control of pendulum", Int. J. Control, Vol. 23, No. 5, pp. 673-692, 1976
20. Nauck, D., 1994, A Fuzzy Perceptron as a Generic Model for Neuro-Fuzzy Approaches, Proceedings of the 2nd German GI-Workshop Fuzzy-Systems '94, München, Germany, October.
21. Nauck, D., Klawonn, F., and Kruse, R., Foundations of Neuro-Fuzzy Systems, John Wiley & Sons, Inc., New York, Chichester, 1997
22. Nauck, D., Kruse, R., and Stellmach, R., New Learning Algorithms for the Neuro-Fuzzy Environment NEFCON-I, Proceedings of the 3rd German GI-Workshop Fuzzy-Neuro-System '95, Darmstadt, Germany, November, pp. 357-364,1995
23. Tou, J. T. (1964): Modern Control Theory, McGraw Hill, New York

11/3/2010