Servo Mechanic Rhythmic System

Algeir P. Sampaio [1], Pablo J. Alsina [2] & José Homero F. Cavalcanti [1]
NEUROLAB/COPIN/DSC - UFPB
Av. Aprígio Veloso, 882 - CEP: 58109 - 970 Campina Grande - PB - BRAZIL
(1) apsam@dsc.ufpb.br, (2) pablo@leca.ufrn.br, (3) homero@dsc.ufpb.br

Abstract

This paper presents the project and implementation of an intelligent rhythmic servo mechanism, called RITMUSROB, developed for basic rhythmic sequence generation in real time. The proposed system uses Artificial Neural Network for learning and control of the movement of a drum stick that plays the rhythm. The system also includes Fuzzy Sets to determine intervals and musical figures.

1 INTRODUCTION

The rhythm notion is basic for the musical perception. The creation of the first rhythms started from the consciousness of the time element in the tribal beats of the primitive man. Since then, the man's domain over woods, rocks, leathers and ropes, allowed to improve the means of percussion and reverberation of the sounds that he extracted from its own rhythm of life. The rhythm as basic element of music, is important in the melody, affects the progression of the harmony and plays roles in the questions as texture, timbre and ornamentation (Sadie, 1994). This great importance justifies a special attention in the Computer Music research.

In this work an intelligent beat system, that is called RITMUSROB, is presented. The system is capable of composing elementary rhythms starting from a sequence of beats. RITMUSROB can be used in the teaching of rhythmic division for beginners in the musical learning. In the following sections of this work, the servo mechanic rhythmic system, its neural control system, the transcription system based on Fuzzy Sets and the rhythm generating system are presented. Experimental results are discussed.

2 RITMUSROB - THE RHYTHMIC SYSTEM

The Fig. 1 presents the simplified model of the RITMUSROB with only one drum stick. RITMUSROB uses an IBM PC (block a) connected by the parallel port to a power electronic drive (block b) for the direct- current (DC motor) (block c). A drum stick (block d), that it is coupled to the DC motor axis, beats the drum (block e). The drum stick movement can be modeled as an inverted pendulum movement. Standards instructions of input and output (I/O) of the PC are used to control the actions of the servo mechanic system in real time. The actions of each one drum stick require the use of tasks that are set in motion by the scheduler.

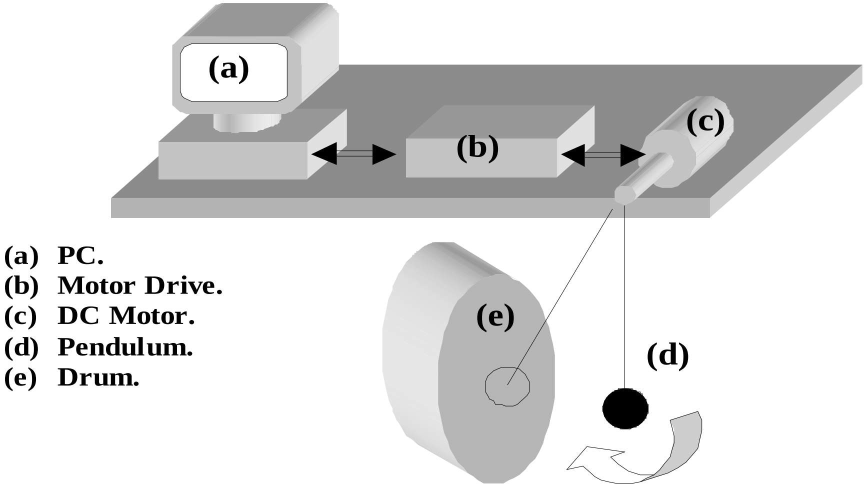


Fig. 1 – Simplified Model of RITMUSROB.

The control software of RITMUSROB was developed in C language using Borland 4.5 under the DOS operational system platform. The block diagram of the developed software is shown in Fig. 2. The MAIN MANAGER is a routine implemented through procedural programming that manages the access to other RITMUSROB routines. The neural net block implements the DC motor position control through a Multi Layer Neural Network (MLNN). It contains all the algorithms that implement the neural network and its additional routines such as those to read/write its parameters in disk, to generate random values to the parameters of the neural network, to draw the neural network in the screen, etc. The fuzzy logic block is responsible by the detection of the duration of musical figures (Semibreve, Minim, Crotchet, Quaver, Semi Quaver, Demi-Semi Quaver and sixty fourth note). It is also responsible by the generation of the parameters of the neural network controller. The real time scheduler is used to activate the percussion routines. Its main purpose is to allow that the commands, in the form of tasks, will be executed at the right moment and in the correct sequence (Liu & Layland, 1973). The block of tasks is composed for all the predefined tasks used for drive the DC motor, where the drum stick is coupled. IRQ (Interrupt Request Query) supplies the pulses for the synchronism of the scheduler.

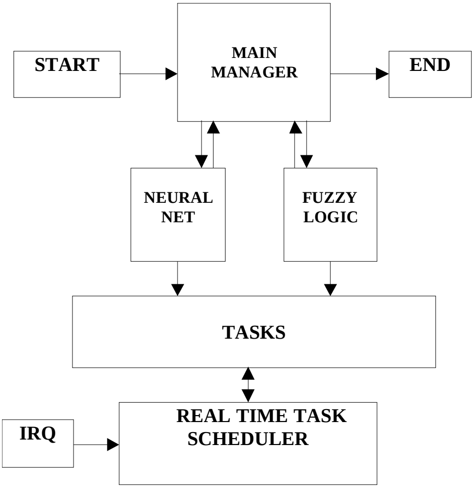


Fig. 2 – Software structure.

3 THE NEURAL CONTROLLER

Fig. 3 shows the neural network utilized to control de DC motor. In this figure, the neurons of the input layer (3 neurons) are linear (represented for L). The neurons of the hidden layer (4 neurons) are of the sigmóide type (represented by S). The neuron of the output layer is of the hyperbolic tangent type (represented for T) [CAV 94]. In the Fig. 3 APR is the Error Back Propagation Algorithm, used to train the MLNN, qr(t) and q(t) are the reference and current angular position of the drum stick, and U(t) is the normalized value of the armature voltage of the DC motor.

Equation (1) shows the performance index used for the training of the MLNN. Cavalcanti (Cavalcanti, et al, 1994)(Cavalcanti, et al, 1995) had shown that the parameters of the MLNN can be updated in real time using equation (3). The value $\Delta U(t) = (-\eta \nabla E_k)$ was calculated using the generalized delta rule (equation (2)).

$E = ½[\theta r(t+1) - \theta(t+1)]2$

(1)

$U(t+1) = U(t) + (-\eta \nabla E_k) = U(t) + \Delta U(t)$

(2)

$\Delta U(t) = \eta \partial\theta(t)/\partial U(t) e(t+1) = \mu e(t+1)$

(3)

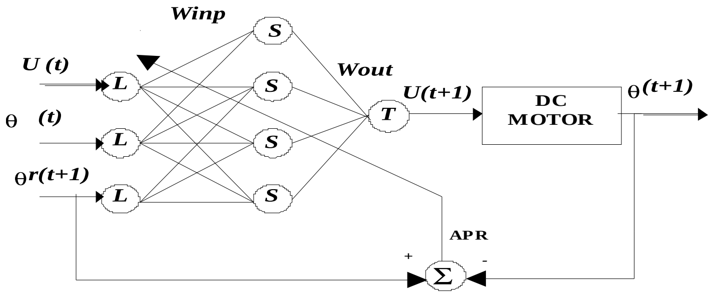


Fig. 3 – Direct Neural Controller.

4 THE FUZZY SETS

Human operators are efficient in the control of several process. This is mainly due to the fact that the human operators are capable to construct, in their minds, a model of the process with the necessary accuracy for the execution of the control task. Also, they can learn starting from the experience obtained in the control of the process.

The human operator implements the process control through a set of linguistic rules, frequently expressed in a vague and ambiguous manner. To model these linguistic rules, a form of semiquantitative calculation is necessary. Based in these considerations, Zadeh (Zadeh, 1965), (Zadeh, 1996) introduced and developed the theory of the fuzzy sets. Fuzzy sets are tools that can particularly be used in the manipulation of vague concepts and fuzzy controllers represent an efforts in the route of the emulation of the human being thinking.

RITMUSROB uses the Fuzzy Logic for adaptation of the training parameters of the MLNN controller and for the generation of the rhythm. RITMUSROB uses the definitions of the musical figures for the development of the membership functions for the generation of the rhythms. Fig. 4 shows the graph of the membership functions of the musical figures as a function of the duration of the musical notes. The membership functions of the musical figures are represented by the linguistic variable Φ figures SB (Semibreve), M (Minim), SM (Crotchet), C (Quaver), SC (Semi Quaver), F (Demi- Semi Quaver) and SF (sixty fourth note).

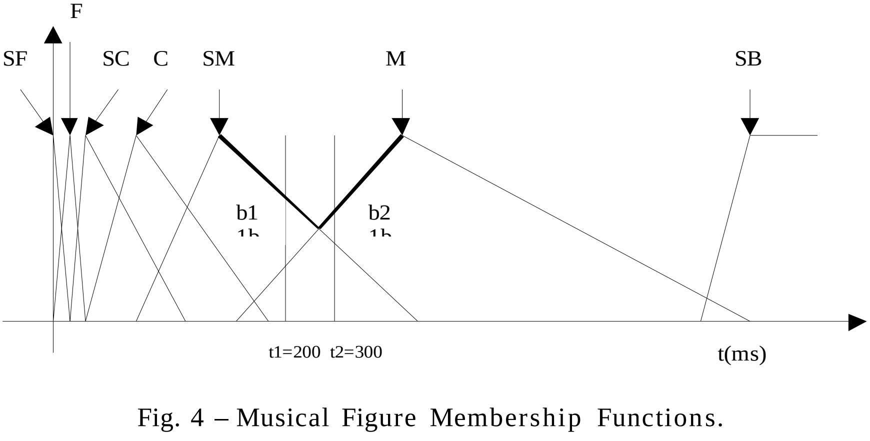


Fig. 4 – Musical Figure Membership Functions.

The time interval between beats is fuzzyfied using the pertinence functions shown in Fig. 4. Each linguistic value has its own reference standard time. Using this times, the system is able to execute any rhythmic sequence given by the user. RITMUSROB stores the obtained fuzzyfied sequence and compares the same with the sequence of beats furnished by the student, indicating when there is a difference between the corresponding time intervals.

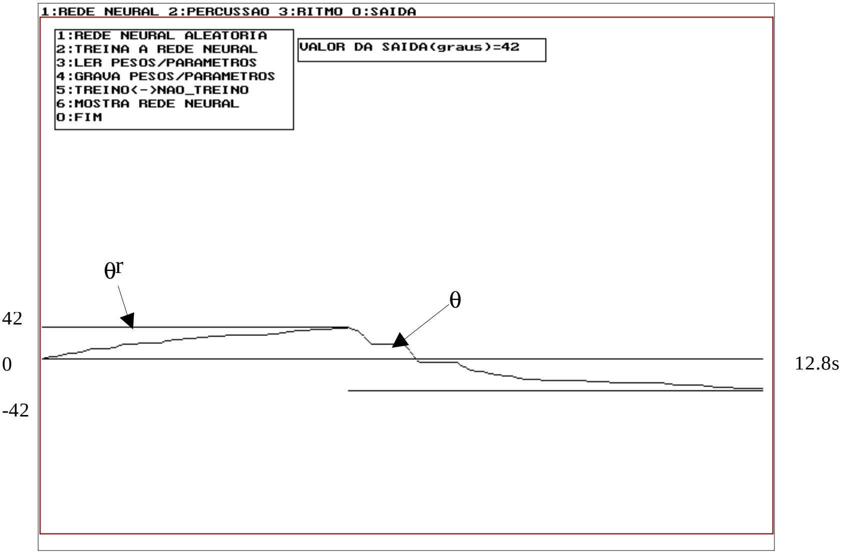
When the user inputs a sequence of beats, RITMUSROB utilizes the membership functions above and fuzzy logic inference to determine its corresponding musical figures starting from its duration. For example, Fig. 4 shows the duration T1 = 200 ms and t2 = 300 ms of two beats, b1 and b2. The problem is to determine their related musical figures. This can be made using the union operator of the Fuzzy Logic. The Union operator can be represented by $A \cup B$ and means: $\mu A \cup B(x) = \max(\mu A(x), \mu B(x)) \; \forall x \in X$, where X is the abscissa of Fig. 4. To illustrate the logical fuzzy operation, the biggest values of the memberships SM and M are presented in boldface. It is observed that for b1 the winning membership is SM, and for b2, the winning membership is M. Therefore the beat b1 will have a duration SM and beat b2 will have a duration M.

5 THE RHYTHMIC GENERATING SYSTEM

Fig. 5 shows the user interface of the RITMUSROB system when the MLNN was been trained. The main menu of the interface has four options. Option number 1 chooses the sub MENU with the functions for manipulation of the MLNN. In this sub MENU, with the sub-option 1, the user will be able to generate random values for the weights and parameters of the MLNN. Sub-option 2 is utilized to train the MLNN. The user will be able to read/write weights and parameters of the MLNN using sub-options 3 and 4. Sub-option 5 allows or not the training of the MLNN and sub-option 6 shows the drawing of the MLNN in the screen of the microcomputer. In the inferior part of Fig. 5 the experimental results during of the training of the RNMC are shown. It is considered that initially the MLNN is not trained with the dynamics of DC motor system and the drum stick. To the side of option A reference angle qr(t) = 42 degrees was utilized to train the MLNN by means of Rule 1:

Rule 1: if $\theta == \theta r$ then $\theta r = - \theta r$

Is is observed that qr(t) = 42 degrees between the time t = 0 and t = 6s and qr(t) = - 42 between t = 6s and t = 12s, qr(t) = - 42 degrees. It was considered that the MLNN approximately learned the system dynamics in 10 iterations. After the training, the weights and the parameters of the MLNN are recorded in a file named pesos.c using sub-option 4.



VALOR DA SAIDA(graus)=42

Fig. 5 - Experimental results of the training of the MLNN.

Fig. 6 shows the screen presented when MENU option 2: PERCUSSION is chosen. The user can input a beat sequence through sub-option 1. The screen used for the presentation of the curves of the references and current positions of the drum stick has 640 points of width, each one representing 20 ms in the horizontal axis. Thus, a maximum of 12800 ms are represented in the full screen. After 12800 ms, the screens if overlapped. According to the desired rhythm, the user can modify, if necessary, this time limit. The values of the reference angle, in maximum of 640 points, are recorded in file named refer.c which includes other information about the rhythm (sub-option 3: READ/WRITE REFERENCES).

When the user choose the sub-option 1: PERCUSSION, the phase of percussion is executed in real time through the keyboard keys 1 to 6 and ESC presented in sub MENU shown in the right upper part of Fig. 6. When using sub-option 1: ZERO, all the reference angles (maximum 640) are set to zero. When using sub-option 2: REFERENCE, the value of reference angle of the STICK DRUM is modified.

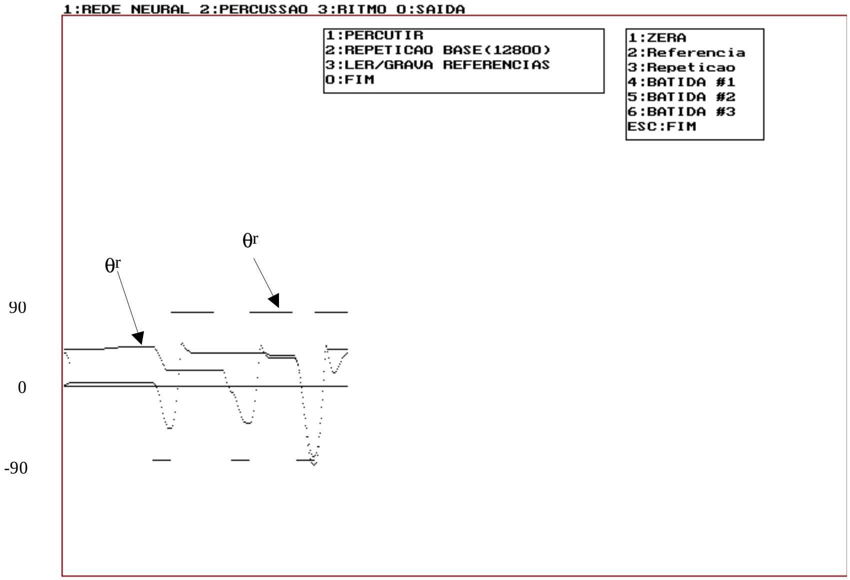


Fig. 6 - Experimental results of the execution of a rhythmic sequence.

In the inferior part of Fig. 6, reference position qr(t) and current position q(t) of the pendulum are shown. In the example of Fig. 6, the user beat the drum using the keyboard of the PC. At each percussion, a reference armature voltage PWM wave was generated for the DC motor that drives the pendulum. In the first beat, the negative reference pulse, followed by a positive reference pulse allows that the pendulum acquires more energy due to the resulting swinging. This way, the pendulum moves faster and a stronger beat is obtained. It is noticed that, although the reference presents positive pulse of +90 °, the pendulum doesn't surpass the position +30 °, because in this position the same is braked by the drummer, happening the percussion. These waveforms had been generated by the user using a keyboard sub- options 4, 5 and 6, which are utilized by the user to indicate the movement of the drum stick.

Using option 3: RHYTHM, whose corresponding screen is not shown in this work, the user will be able to choose between four sub options of the MENU. Sub-option 1: RECORD EXECUTED RHYTHM stores on disk the file with the sequence of beats. Sub option 2: EXECUTE RECORDED RHYTHM repeats the sequence of beats saved on file with the previous option. Sub option 3: SHOW INTERVALS/KEYS shows, through the fuzzy inference, the linguistic values (musical figures) and played keys corresponding to the sequence of beats typed by the user. Several keys, associated to different beat intensities, are available to the user. The key sequence, corresponding to the rhythm executed by the user, is stored in a file named ritmo.c. Finally, sub-option 0: END is used to leave the MENU.

6 CONCLUSIONS

A system for rhythms generation in real time was presented. The system offers to the possibility of construction of new rhythms from the beats executed by the user. Once that the system executes programmed rhythmic sequences, it can be easily adapted for new tasks. In future works, the developed system will be adapted to read and execute a rhythmic partitur. The proposed system can be utilized to assist music beginners, by helping them to learn the execution of a rhythmic dictation. A new version of RITMUSROB is being developed in language C++ Builder for Windows platform. For a future work, it is under studies the possibility of using Wavelet transformation to describe the associated curves to the pulses of the rhythmic beats, qr(t) and q(t).

7 REFERENCES

Cavalcanti, J. H. F.; Lima, A. M. N.; Deep, G. S. (1994), *On-line Training of Adaptative Neural Network Controllers, IEEE Industrial Electronics Society* IECON'94, Proceedings, Bologna, Italy, setembro 1994, pp. 92-114.

Cavalcanti, J. H. F.; Sales, E. F. Jr. & Ferneda, E. (1995*). Intelligent Control of an Inverted Pendulum. Proc. of IISBRN’95*, São Carlos, São Paulo, october 1995.

Liu, C. L. & Layland, J. W. (1973). Scheduling Algorithms for Multiprogramming in Hard Real Time Environments. *JACM,* Vol. 20, January, 1973.

Sadie, Stanley (1994). *Dicionário Groove de Música (Edição Concisa)*. Jorge Zahar Ed., Rio de Janeiro - Brasil.

Zadeh, L. A. (1965). Fuzzy Sets, Information and Control, 1965, 8, pp.28- 44.

Zadeh, L. A. (1996). Fuzzy Logic - Computing with Words, *IEEE Fuzzy Systems*, Vol. 4, may, 1996.