

## Sistema de Acionamento Percussivo Inteligente - RITMUSROB

Algeir P. Sampaio<sup>1</sup>, José Homero Feitosa Cavalcanti<sup>2</sup>, Pablo Javier Alsina<sup>3</sup>

UNIVERSIDADE FEDERAL DA PARAÍBA  
NEUROLAB/COPIN/DSC/CCT

Rua Aprígio Veloso, 882 - Campus II  
58109-790 - Campina Grande - PB

E-mails: {apsam,homero}@dsc.ufpb.br, pablo@leca.ufrn.br

### Abstract

*This paper presents the project and implementation of an intelligent rhythmic robot system, called RITMUSROB developed from a basic rhythmic sequence generation in real time. The proposed system uses Artificial Neural Network to 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. Introdução

A noção de ritmo é fundamental para a percepção musical. A criação dos primeiros ritmos, partiu da consciência do elemento tempo nas batidas tribais do homem primitivo. Desde então, o domínio do homem sobre paus, pedras, couros e cordas, serviu para percutir e repercutir sons extraídos do seu próprio ritmo de vida. O ritmo, como elemento fundamental da música, é importante na melodia, afeta a progressão da harmonia e desempenha papéis em questões como textura, timbre e ornamentação [SAD 94]. Tanta importância justifica a atenção especial por parte da pesquisa na área de Computação & Música.

Neste trabalho apresenta-se um acionador robótico percussivo inteligente, que é capaz de, a partir de uma seqüência de batidas percussivas, compor ritmos elementares. O RITMUSROB pode ser usado no ensino de divisão rítmica para iniciantes na aprendizagem musical. Nas demais seções deste trabalho são apresentados o sistema ritmista robótico, o sistema de controle neural, a utilização da Lógica Fuzzy no RITMUSROB e o sistema gerador de ritmos e alguns comentários sobre os resultados experimentais obtidos.

#### 1.1. O RITMUSROB - O Sistema de Acionamento Percussivo

A Fig. 1 apresenta o modelo simplificado do RITMUSROB com uma única baqueta percussiva. O RITMUSROB usa um microcomputador IBM PC (*bloco a*) ligado via interface paralela da impressora a uma placa (*bloco b*) para acionamento do motor de corrente contínua (motor CC) (*bloco c*). No eixo do

motor é acoplada a baqueta que se comporta, durante a percussão, de forma similar a um pêndulo. Utilizam-se as instruções padrões de entrada e saída do PC para controlar em tempo real os movimentos do braço robótico ou baqueta percussiva. As ações de cada uma das baquetas requerem o uso de tarefas que são acionadas por um escalonador.

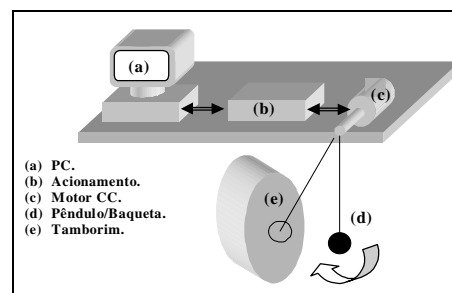


Fig. 1: Modelo Simplificado do RITMUSROB.

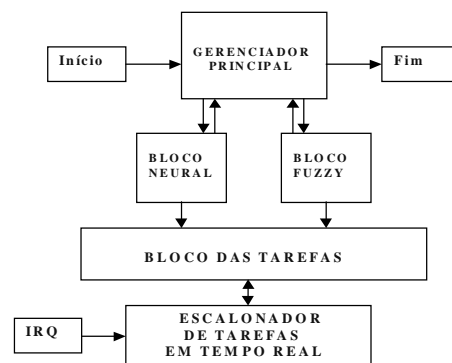


Fig. 2: Diagrama de Bloco do Software.

O software do RITMUSROB foi desenvolvido em linguagem C usando o Borland 4.5 sob a plataforma do sistema operacional DOS. O diagrama de blocos do RITMUSROB está mostrado na Fig. 2. O **GERENCIADOR PRINCIPAL** é uma rotina implementada através de programação procedural que gerencia o acesso as outras rotinas que compõem o software do RITMUSROB. O **BLOCO NEURAL** é utilizado no acionamento da rede neural multi camadas (RNMC) que controla o posicionamento do motor CC, contém todos os algoritmos que

implementam a rede neural e suas rotinas auxiliares como ler/gravar os seus pesos e parâmetros em disco, gerar valores aleatórios para os pesos e parâmetros da rede neural, desenhar a rede neural no monitor de vídeo etc. O **BLOCO FUZZY** é utilizado na parte inteligente do RITMUSROB. Ele é encarregado da detecção do tempo das figuras melódicas (semibreve, mínima, semínima, colcheia, semicolcheia, fusa e semifusa) e geração dos parâmetros da rede neural controladora. O **ESCALONADOR EM TEMPO REAL** é utilizado para decidir sobre o acionamento das rotinas de percussão e sua principal finalidade é permitir que os comandos, em forma de tarefas, sejam executados no momento exato e na seqüência correta [LIU 73]. O **BLOCO DE TAREFAS** é composto por todas as tarefas predefinidas para preparação e o acionamento do motor CC onde está acoplado a baqueta percussiva. E o **IRQ** (Relógio de Requerimento de Interrupções) fornece os pulsos para sincronismo do escalonador.

### 1.2. O Controlador Neural

A Fig. 3 mostra o esquema da rede neural multi camadas usada como controlador do motor CC. Nessa figura os neurônios da camada de entrada (3 neurônios) são lineares (representados por L), os da camada escondida (4 neurônios) são do tipo sigmóide (representados por S), e o neurônio da camada de saída é do tipo tangente hiperbólico (representado por T) [CAV 94]. Na Fig. 3 APR significa o Algoritmo de Propagação Retroativa do Erro usado para treinar a RNMC,  $\theta_r(t)$  e  $\theta(t)$  representam as posições referência e atual da baqueta percussiva, e  $U(t)$  representa o valor (por unidade) da tensão de armadura do motor CC.

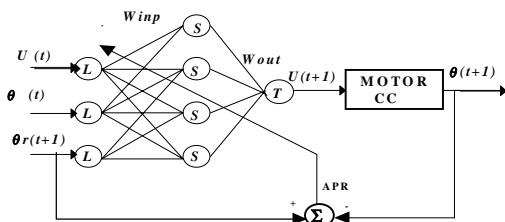


Fig. 3: Controlador Neural Direto.

Na equação (1) é mostrado o índice de desempenho utilizado para treinamento da RNMC. Em [CAV 94] e [CAV 95] foi mostrado que os parâmetros da RNMC podem ser adaptados em tempo real usando a equação (3). O valor  $\Delta U(t) = (-\kappa \nabla E_U)$  foi calculado usando a regra delta generalizada (equação 2).

$$E = \frac{1}{2} [\theta_r(t+1) - \theta(t+1)]^2 = \frac{1}{2} e(t+1)^2 \quad (1)$$

$$U(t+1) = U(t) + (-\kappa \nabla E_U) = U(t) + \Delta U(t) \quad (2)$$

$$\Delta U(t) = \kappa \frac{\partial \theta(t)}{\partial U(t)} e(t+1) = \mu e(t+1) \quad (3)$$

### 1.3. A Lógica Nebulosa

Os processos controlados com operadores humanos são geralmente adequados em algumas aplicações. Isto se deve ao fato de que os operadores humanos são capazes de construir nas suas mentes um modelo do processo com a necessária precisão para a execução à contento da tarefa de controle, podendo, inclusive, aprender com a experiência obtida no controle do processo. Para dar suporte à transformação das declarações sobre a estratégia de controle, geralmente vagas, não numéricas, dos controladores humanos, é necessária uma forma de cálculo semiquantitativa. Baseado nessas considerações é que Zadeh [ZAD 65][ZAD 96] introduziu e desenvolveu a teoria dos conjuntos nebulosos ou fuzzy e do raciocínio aproximado. Conjuntos nebulosos são ferramentas que podem ser usadas na manipulação de conceitos vagos e particularmente, controladores nebulosos representam os esforços na direção da emulação da capacidade humana.

O RITMUSROB utiliza a Lógica Nebulosa para modificação dos parâmetros de treinamento da RNMC controladora e para a geração do ritmo. Não fosse, o uso da Lógica Nebulosa, o sistema não teria como proceder na classificação das figuras musicais. O RITMUSROB utiliza as definições das figuras melódicas para o desenvolvimento das funções de pertinência para a geração dos ritmos. Na Fig. 4 é mostrado o gráfico das funções de pertinência das figuras musicais em função do tempo (abscissa). As funções de pertinência das figuras musicais são representadas pelas variáveis linguísticas: SB (semi breve), M (mínima), SM (semínima), C (colcheia), SC (semi colcheia), F (fusa) e SF (semi fusa).

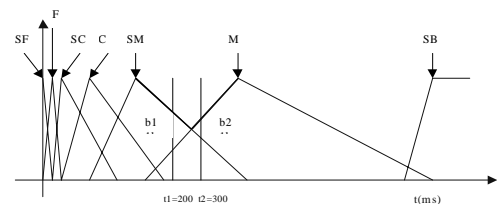


Fig. 4: Função de Pertinência das Figuras Musicais.

A idéia é, através do intervalo de tempo entre as diversas batidas, usando a Lógica Nebulosa, encontrar as figuras musicais de cada batida. Por exemplo, na Fig. 4 são mostrados os intervalos de tempo  $t_1 = 200$  ms e  $t_2 = 300$  ms de duas batidas  $b_1$  e  $b_2$ . O problema é detectar quais são as figuras musicais de cada batida. Isto pode ser feito usando-se o operador união da Lógica Nebulosa. O operador **União** pode ser representado por  $A \cup B$  e significa:  $\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x)) \forall x \in X$ . Onde  $X$  é a abscissa da Fig. 7. Para ilustrar a operação lógica fuzzy, os valores maiores das funções de pertinência SM e M são apresentados em negrito. Observe-se que para  $b_1$  a

função de pertinência vencedora é SM, e para b2, a função de pertinência vencedora é M. Portanto a batida b1 terá intervalo de tempo para execução SM e b2, M.

#### 1.4. O Software do RITMUSROB

A Fig. 5 mostra a tela gráfica do sistema RITMUSROB obtida na ocasião em que foi treinada a RNMC. Observa-se na parte superior da figura, o MENU principal. Se o usuário teclar o número 1:REDE NEURAL, ele escolhe as funções de manipulação da RNMC, disponíveis no sub MENU desta opção. No sub MENU, o usuário poderá gerar valores aleatórios para os pesos e parâmetros da RNMC com a opção 1:REDE NEURAL ALEATORIA. Teclando a opção 2:TREINA REDE NEURAL, o usuário treina a rede neural e poderá ler/gravar pesos e parâmetros da RNMC usando as opções 3:LER PESOS/PARAMETROS e 4:GRAVA PESOS/PARAMETROS. A opção 5:TREINO <-> NÃO TREINO deixa a opção para treinar ou não treinar a rede neural, caso o usuário queira. A opção 6:DESENHA REDE NEURAL desenha a rede neural na tela. A opção 7:PONTO PASSIVO REDE NEURAL coloca o valor zero na saída da rede neural e finalmente a opção 0:SAIDA, oferece a saída de 1:REDE NEURAL.

Na parte inferior da figura 5 também são mostrados os resultados experimentais obtidos durante as duas primeiras iterações do treinamento da RNMC (opção 2). Considera-se que inicialmente a RNMC não está treinada com a dinâmica do sistema motor CC acoplado à baqueta percussiva. Ao lado da opção 2 é mostrado o valor do ângulo de referência ( $\theta_r(t) = 30$  graus) usado para treinar a RNMC. Usou-se a regra 1 para o treinamento da RNMC. Observe-se que  $\theta_r(t) = 30$  graus entre os intervalos de tempo  $t = 0$  e  $t = 6$ s e entre os intervalos de tempo  $t = 6$  segundos e  $t = 12$  segundos,  $\theta_r(t) = -30$  graus. Considerou-se que a RNMC aprendeu a dinâmica do sistema em aproximadamente 10 iterações. Após o treinamento, os pesos e os parâmetros da RNMC são gravados usando a opção 4 num arquivo chamado *pesos.c*.

Regra 1) **if**  $\theta == \theta_r$  **then**  $\theta_r = -\theta_r$

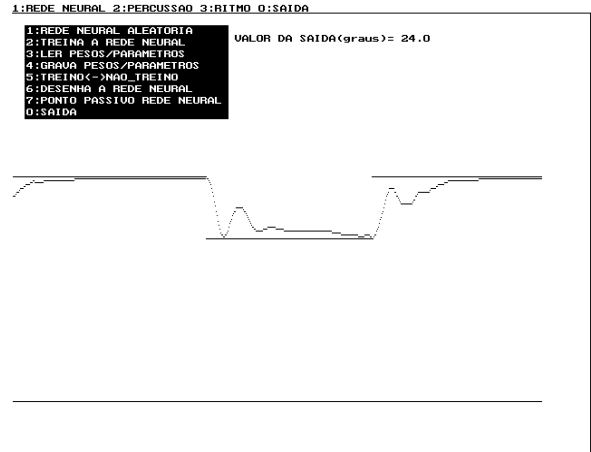


Fig. 5: Resultados Experimentais obtidos no treinamento da RNMC.

A figura 6 mostra a tela obtida com a opção 2:PERCUSSÃO do RITMUSROB. A tela usada para a apresentação das curvas das referências e das posições da baqueta possui 640 pontos, cada ponto representando 20 milissegundos, tendo um tempo total de 12800 milissegundos por tela. Após 12800 milissegundos, as telas se sobrepõem. Portanto, cada tela terá um máximo de 12.8 segundos. Após o usuário teclar a sub opção 1:PERCUTIR, o programa disponibiliza as teclas 1 a 6 e a tecla ESC para a execução das batidas em tempo real pelo robô (mostradas na figura 6). Usando a sub opção 2:REPETE INICIO E FIM, o usuário poderá modificar este tempo (o tempo total de 12,8 s) de acordo com a duração do ritmo desejado (tempo inicial e tempo final). A partir da sub opção 3:MUDA PARÂMETROS BATIDA, o usuário pode modificar o número de batidas e o números das batidas para executar um novo ritmo. A sub opção 4:LER/GRAVAR REFERÊNCIAS oferece ao usuário a possibilidade de ler e/ou gravar as referências do novo ritmo gerado. A sub opção 5:REPETE INICIO BATIDA #1, repete a sequência teclada com o início no instante oferecido pelo usuário e com a batida #1. A sub opção 0:SAIDA é usada para terminar a apresentação em tempo real, retornando ao MENU principal na opção 2:PERCUSSÃO. A figura 6 mostra ainda, o gráfico com os valores das posições de referência ( $\theta_r(t)$ ) e atual ( $\theta(t)$ ) do pêndulo para a batida #2.

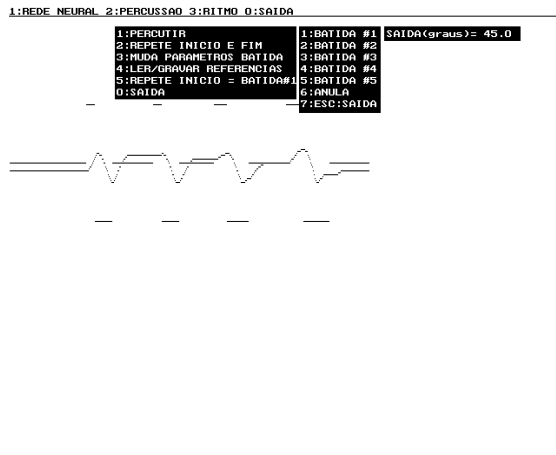


Fig. 6: Seqüência de batidas obtidas através do teclado.

Na parte inferior da Fig. 6 é mostrado o gráfico com os valores das posições referência ( $\theta_r(t)$ ) e atual ( $\theta(t)$ ) do pêndulo. No exemplo da Fig. 6, o usuário percutiu o tamborim, através do teclado do PC. Observa-se que a cada percussão foi gerada uma onda PWM de referência do pêndulo. Na primeira percussão, a forma de onda referência transportou o pêndulo na direção negativa, representado pelo sinal PWM abaixo de zero, logo seguida de uma referência positiva. Isto aumentou a velocidade do pêndulo, aumentando a intensidade da batida. Nota-se que, embora a referência apresente pulso positivos de  $90^\circ$ , o pêndulo não ultrapassa a posição  $+30^\circ$ , pois nesta posição o mesmo é freado pelo bombo, ocorrendo a percussão.

Essas formas de onda foram geradas pelo usuário teclando as teclas 2 (duas vezes); 3 e 4 que representam as batidas #2, batida #3 e batida #4. Na Fig. 6, o usuário está gerando, a partir do tempo atual, 20 valores de referência como mostrado no trecho de programa na linguagem C mostrado abaixo:

```
Prog 1
```

```
For(i =1; i<=20;i++) posref [onde+i] = saida;
```

Usou-se a variável **onde** como apontador do ponto atual de percussão e **saida** é o valor do ângulo de referência.

A opção 3:RITMO do MENU principal é responsável pela parte fuzzy ou parte nebulosa do software de controle do RITMUSROB. A sub opção 1:GRAVA TABELA DO RITMO grava um arquivo (ritmoout.c) com os dados das batidas executadas via teclado ou seja, faz com que o arquivo gerado com as saídas (execução das batidas), seja gravado no disco. A sub opção 2:LER TABELA DO RITMO é usada para ler os valores gravados no arquivo gerado na opção anterior. A sub opção 3:PERCUTIR VIA REFERENCIA é usada para a execução das batidas, uma vez que o RITMUSROB já assimilou a seqüência rítmica teclada pelo usuário. A sub opção 4:CURVA FUZZY DA PERCUSSAO exhibe na tela a curva fuzzy das batidas e calcula as variáveis lingüísticas das figuras musicais.

Os tempos entre as batidas da baqueta é “fuzzyficado” usando as funções de pertinência da Fig. 4. Desta forma, o sistema terá uma referência, a partir de um tempo padrão, representado por uma percussão e desta referência poderá executar uma seqüência rítmica qualquer composta pelo usuário. O RITMUSROB armazena a seqüência Fuzzy e a compara com a seqüência de batidas do usuário indicando quando houver uma diferença entre os intervalos de tempo.

A sub opção 5:PARAMETROS TETA/BETA/TI mostra todos os parâmetros fuzzy utilizados nas curvas. A sub opção 6:MOSTRA TABELA BATIDAS mostra uma tabela com todas as batidas, os instantes reais, o ângulo de referência de cada batida, o número das batidas (exemplo: se batida #1 ou batida #2 ou qualquer outra), o tempo das batidas (fuzzyficado) e a indicação das figuras musicais referentes aos intervalos entre as batidas, além da indicação da baqueta que efetuou a batida (se a baqueta 1 ou a baqueta 2). A figura 7 mostra a execução da sub opção 6 para a seqüência de batidas da figura 6.

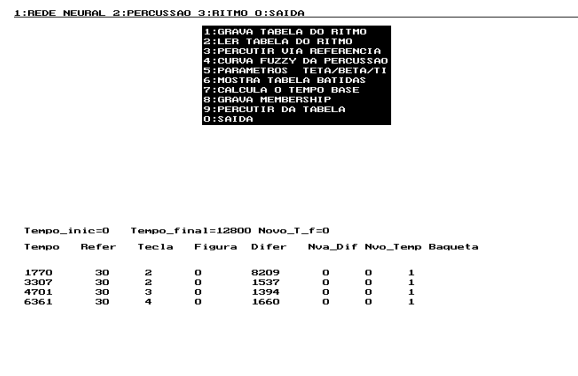


Fig. 7: Tela da execução da sub opção 6:MOSTRA TABELA BATIDAS da opção 3: RITMO do MENU.

A sub opção 7:CALCULA O TEMPO BASE mostra a relação entre os tempos reais e os tempos fuzzyficados atribuídos a cada figura musical de acordo com sua duração. Ao teclar esta opção o usuário, a pedido do programa, fornecerá o novo tempo base que servirá de referência para a fuzzyficação dos intervalos. A sub opção 8:GRAVA MEMBERSHIP grava um arquivo (members.c) com informações sobre a função de pertinência da curva fuzzy das batidas. A sub opção 9:PERCUTIR DA TABELA executa uma seqüência rítmica fornecida através da tabela gravada com a sub opção 1 que pode ser modificada pelo usuário. Nesta sub opção o usuário tem a escolha entre a seqüência rítmica executada via teclado (em tempo real) ou a seqüência rítmica corrigida (fuzzyficada) através da tabela gerada na sub opção 6. E finalmente, a sub opção 0:SAIDA fornece a saída deste MENU.

## 1.5. Conclusão

Apresentou-se um sistema para a geração de ritmos percussivos em tempo real. O sistema oferece a possibilidade da construção de ritmos a partir de batidas executadas pelo usuário, cumprindo sua tarefa de gerador de ritmos percussivos. Uma vez que este mesmo sistema executa seqüências rítmicas programadas, existe a possibilidade de uma evolução nas suas tarefas.

Observou-se também que uma vez que um dos robôs (braços robóticos/baqueta) esteja executando um ritmo, o sistema oferece condições de interferência nesta execução por parte do usuário, comandando a outra baqueta para com efeito de improvisação acrescentar batidas ao ritmo em execução, criando desta forma um ritmo mais complexo formado por tempo e contratempo. Isto sugere uma operação de sincronização entre os dois robôs cooperantes.

É pretendida uma adaptação no algoritmo do sistema ritmista robótico RITMUSROB para que este seja capaz de ler uma entrada em forma de partitura musical e execute a mesma, auxiliando alunos de iniciação musical na reprodução das batidas rítmicas durante um ditado rítmico. Uma nova versão do software do RITMUSROB está sendo desenvolvida usando a linguagem C++ Builder para Windows.

Para um trabalho futuro, sugere-se a investigação da utilização das transformadas de Wavelet nas curvas  $\theta_r(t)$  e atual  $\theta(t)$  e suas relações com as curvas PWM geradas pelo pulso das batidas rítmicas.

## Referências

- [CAV 94] CAVALCANTI, J. H. F.; LIMA, A. M. N.; DEEP, G. S., *On-line Training of Adaptive Neural Network Controllers*, IEEE Industrial Electronics Society IECON'94, Proceedings, Bologna, Italy, setembro 1994, pp. 92-114.
- [CAV 95] CAVALCANTI, J. H. F.; SALES, E. F. Jr.; FERNEDA, E., *Intelligent Control of an Inverted Pendulum*, IISBRN'95, São Carlos, São Paulo, outubro 1995.
- [LIU 73] LIU, C. L. & LAYLAND, J. W., Scheduling Algorithms for Multiprogramming in Hard Real Time Environments. JACM Vol. 20, janeiro 73.
- [SAD 94] SADIE, STANLEY, *Dicionário Groove de Música (Edição Concisa)* - Jorge Zahar Ed., Rio de Janeiro, 1994
- [SAM 97] SAMPAIO, A. P. & CAVALCANTI, H. *RitmusRob - Um Acionador Percussivo Inteligente*, Artigo referente ao trabalho da disciplina Redes Neurais no mestrado de Informática, COPIN - DSC -UFPB, julho 1997.
- [ZAD 65] ZADEH, L.A. *Fuzzy Sets, Information and Control*, 1965, 8, pp.28-44.
- [ZAD 96] ZADEH, L. A. *Fuzzy Logic = Computing with Words*, IEEE Fuzzy Systems, Vol. 4, maio 1996.