

# Aplicação de Processamento Paralelo e Distribuído na Robótica

Vinícius M. de Oliveira, Rodrigo Neves Calheiros,  
Odorico Mendizabal, Cristina Meinhardt

Fundação Universidade Federal do Rio Grande  
vinicius@ieee.org, {rcalheiros,odo,crisina}@ecomf.furg.br

## Introdução

Há algum tempo se estuda a aplicação de redes neurais artificiais no controle de robôs, sejam manipuladores ou móveis, mas devido à complexidade inerente às redes neurais, em termos de poder computacional exigido, faz com que se inviabilize a aplicação prática desses sistemas, uma vez que se trabalha sob condições restritas de tempo. Em [OLI 01, FIE 98] tem-se a utilização de uma RNA com aprendizado *online*, ou seja, a rede adquire o conhecimento à medida que se executa o movimento do robô.

Pretende-se nesse trabalho a aplicação de técnicas de paralelização e distribuição de código, a fim de se estudar a utilização de redes neurais mais complexas e com melhores leis de aprendizado. A proposta é lançar mão do processamento paralelo entre o cálculo do modelo matemático do robô e a atualização dos pesos da rede neural.

Nas seções que seguem apresenta-se o robô no qual se baseou o trabalho, bem como alguns aspectos sobre o controle aplicado a esse robô. Por fim a descrição da parte de implementação de código para execução paralela e alguns resultados obtidos por simulação.

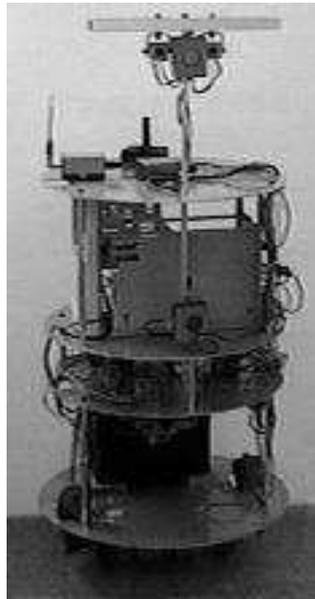


Figura 1: Robô móvel Twil.

## O Robô Móvel

Nesse trabalho utilizou-se um robô móvel com acionamento diferencial nas rodas, apresentado na figura 1.

O **Twil** é controlado por um computador baseado no microprocessador *Pentium MMX 233MHz*. Os motores são acionados por *Pulse Width Modulation* utilizando-se o *timer* programável 82C54. O robô está equipado com duas câmeras de vídeo, dotando o **Twil** de visão. Na mesma torre de sustentação das câmeras de vídeo estão montados, em alturas diferentes, dois sonares de ultra-som. A instrumentação do Twil possui ainda uma bússola digital.

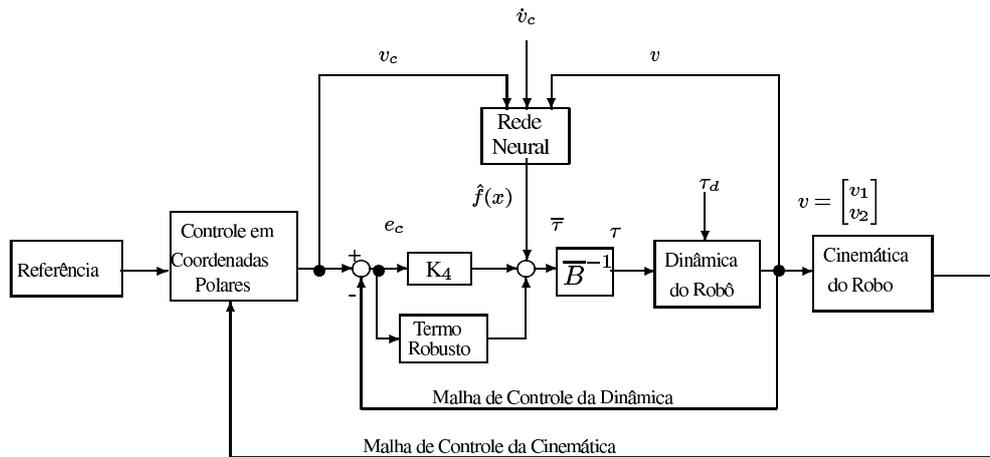


Figura 2: Diagrama de blocos do controlador.

## Estrutura de Controle

A estrutura de controle proposta neste artigo é apresentada na figura 2. Observa-se que não é necessário nenhum conhecimento prévio a respeito da dinâmica do robô, uma vez que o papel da rede neural na estrutura de controle é exatamente o de adquirir tal conhecimento de forma *online*, isto é, sem a fase de treinamento prévio da rede neural.

## Rede Neural

Neste artigo utilizou-se uma rede neural artificial multi-camada com atualização *online* dos pesos. Essa rede neural possui 6 neurônios na camada de entrada, 8 neurônios na camada intermediária (camada escondida) e 2 neurônios na camada de saída.

Como função de ativação escolheu-se a função de ativação sigmoideal, por ser simples e amplamente utilizada na literatura. A saída da rede neural é um vetor  $y \in \mathcal{R}^{2 \times 1}$ , expresso por:

$$y(x) = \mathbf{W}^T \sigma(\mathbf{V}^T x) \quad (1)$$

Uma das características mais importantes da rede neural artificial é a capacidade de aproximar funções contínuas não-lineares multivariáveis.

Os pesos são ajustados de modo *online* por meio das seguintes equações:

$$\Delta \hat{\mathbf{W}} = \mathbf{F} \sigma(\hat{\mathbf{V}}^T x) e_c^T - \mathbf{F} \sigma'(\hat{\mathbf{V}}^T x) \hat{\mathbf{V}}^T x e_c^T - k \mathbf{F} \| e_c \| \hat{\mathbf{W}} \quad (2)$$

$$\Delta \hat{\mathbf{V}} = \mathbf{G} x (\sigma'(\hat{\mathbf{V}}^T x) \hat{\mathbf{W}} e_c)^T - k \mathbf{G} \| e_c \| \hat{\mathbf{V}} \quad (3)$$

onde os parâmetros de projeto  $\mathbf{F}$  e  $\mathbf{G}$  são matrizes positivas definidas e  $k > 0$ .

## Implementação Distribuída

O problema anteriormente descrito pode-se valer de otimização através de programação paralela e distribuída pelo fato de que tanto a atualização dos pesos da rede neural quanto a simulação do modelo matemático do robô móvel exigirem um volume considerável de processamento e terem apenas a rede neural compartilhada.

Desta forma, enquanto o processo de simulação é realizado com um valor da rede neural, o processo de atualização da mesma pode ser executado simultaneamente, sendo necessário apenas o controle do acesso concorrente às matrizes de peso da rede.

Na implementação distribuída realizada nesse trabalho, foram utilizados dois computadores Pentium II 200MHz, 32MB de RAM, pertencentes ao laboratório do curso de Engenharia de Computação da FURG. As máquinas utilizadas estão conectadas por meio de uma rede Ethernet de 10Mbps. Para produzir a aplicação, foi utilizada a linguagem de programação C++ juntamente com a biblioteca LAM/MPI.

A aplicação foi distribuída de forma que um dos computadores executasse os algoritmos para a atualização *online* dos pesos da rede neural, e a outra máquina ficasse com a execução dos algoritmos referentes à simulação do modelo do robô (algoritmos de integração numérica). A comunicação existente entre os processos era o envio dos pesos da rede neural atualizados do primeiro computador para o segundo, a fim de que sejam calculados os novos valores de controle.

## Implementação Paralela

Para a execução paralela, foi utilizado um computador Athlon 1.7 GHz dual com 1GB de RAM<sup>1</sup>. A solução foi implementada utilizando-se linguagem de programação C++ e *threads* POSIX (pthread.h). Nessa implementação, eram criadas exatamente duas linhas de execução (*threads*): uma delas responsável pela atualização dos pesos da rede neural e a outra encarregada da simulação do robô (integração numérica).

Uma vez que a primeira atualiza a rede neural, a segunda poderá utilizar este valor para a simulação, e ao mesmo tempo a outra pode começar o processo de atualização dos pesos. Para garantir que os pesos não serão atualizados antes de realizada a simulação com os valores atuais, é necessária a proteção da região da memória utilizada para representação da rede. Para tal tarefa foi utilizada uma implementação de semáforos apresentada em [CAV 01] (convenientemente adaptada ao problema), que utiliza recursos da biblioteca *pthread*.

<sup>1</sup>Agradecemos ao Prof. Dr. Sebastião Gomes pela disponibilização do computador para o trabalho.

## Resultados e Conclusões

Para comparação do tempo de execução das duas implementações acima descritas, foram feitas 10 execuções do algoritmo para um processo e 10 para o distribuído, sendo utilizadas a média de cada implementação como critério de análise e comparação. Valendo-se da implementação distribuída, se obteve um melhor desempenho na execução do algoritmo, sinalizando que a utilização do tempo ocioso de computadores de baixo poder computacional para tal tarefa pode ser satisfatória.

Já os resultados obtidos com a implementação paralela em uma máquina de alto desempenho não corresponderam às expectativas com relação ao desempenho obtido pela aplicação rodada sequencialmente nesta mesma máquina. O motivo provável para tal fato é que o tempo que cada *thread* espera para acessar a região crítica é maior do que o tempo que a aplicação sequencial necessita para efetuar as operações de atualização da rede neural, devido a grande capacidade de processamento da máquina em questão. Portanto, para a implementação paralela atual do programa, não é vantajoso o uso de máquinas de alto desempenho, não tendo sido porém testado o programa em máquinas dual com menor capacidade.

O objetivo imediato é realizar testes com redes neurais mais complexas, ou seja, com um maior número de neurônios na camada intermediária ou com mais de uma camada intermediária, a fim de necessitarmos de um maior tempo de execução para os cálculos de atualização dessa rede neural.

Ainda na área da Robótica, vê-se outra possível aplicação de execução paralela e distribuída de código na visão computacional. A idéia é estudar a paralelização/distribuição de códigos referentes à manipulação de imagens captadas, para posterior análise e extração de informações úteis à lei de controle projetada.

## Referências

- [OLI 01] OLIVEIRA, V.M. et al. **Controle em Malha Fechada de Robôs Móveis Utilizando Redes Neurais e Transformação Descontínua**, Anais do V Simpósio Brasileiro de Automação Inteligente. Porto Alegre, 2001.
- [FIE 98] FIERRO, R.; LEWIS, F.L. **Control of Nonholonomic Mobile Robot Using Neural Networks**, IEEE Transactions on Neural Networks, v.9, p.589–601, 1998.
- [LAG 98] LAGES, W.F.; HEMERLY, E.M. **Smooth Time-invariant Control of Wheeled Mobile Robots**, Proceedings of XIII International Conference of System Science, Poland, 1998.
- [YAM 94] YAMAMOTO, Y.; YUN, X. **Coordinating Locomotion and Manipulation of a Mobile Platform**, IEEE Transactions on Automatic Control. v.39, p.1326–1332, 1994.
- [CAV 01] CAVALHEIRO, G.G.H. **Introdução à Programação Paralela e Distribuída**, Anais 1 Escola Regional de Alto Desempenho. p.35–74. Gramado, 2001.