

# Um Modelo de Redes Neurais para Controle de Sistemas Robóticos

Marcio Luiz Bunte de Carvalho\*      Wagner Meira Junior†

*Departamento de Ciência da Computação  
Universidade Federal de Minas Gerais  
Caixa Postal 702  
30161 - Belo Horizonte - MG  
Fone: (031) 443-4088*

## Abstract

The field of robotics has much of its complexity because of a high dependence upon its environment which can change frequently. The use of neural networks to solve problems in robotics is justified by their flexibility and learning ability.

We present a short conceptual study in both fields. Neural models are proposed and the problems in robotics to be verified are defined. The investigations made to specify the best training conditions are described. The validity of the work is verified through practical examples.

## Resumo

A área de robótica tem muito da sua complexidade devido a alta dependência com o ambiente de atuação que pode se alterar com grande frequência. A utilização de redes neurais para a solução de problemas de robótica é justificada pela flexibilidade e capacidade de aprendizagem daquelas.

É feito um breve estudo conceitual em ambas as áreas. São propostos modelos neurais e definidos os problemas de robótica a serem tratados. Descreve-se as investigações realizadas no sentido de determinar as melhores condições de treinamento. A validade do trabalho é verificada através de exemplos práticos.

---

\*Professor do Departamento de Ciência da Computação; Doutorando em Pesquisa Operacional - UC Berkeley  
Áreas de Interesse: Algoritmos Paralelos; Programação Matemática; Álgebra Linear Numérica

†Bacharelado em Ciência da Computação - UFMG; Áreas de Interesse: Redes Neurais

## 1 Introdução

Podemos notar uma grande diferença entre os computadores tradicionais e os sistemas vivos, desta forma, enquanto os primeiros são extremamente rápidos e precisos, os segundos se destacam pela sua flexibilidade e capacidade de aprendizado. Isto pode ser exemplificado da seguinte forma: enquanto um homem consome um tempo razoável em operações com números reais, o computador faz o mesmo em frações de segundo, por outro lado, enquanto precisamos de apenas alguns segundos ou até menos para reconhecer objetos chegando a detalhes como textura, material, natureza e finalidade; um supercomputador pode demandar milhões de operações, muitas vezes apresentando resultados um tanto decepcionantes.

Em muitos problemas com os quais nos deparamos, necessitamos de algum recurso dotado de certa flexibilidade e tolerância a ruídos. Além disso, este recurso precisaria possuir uma capacidade de aprendizado, o que permitiria adaptá-lo a diferentes ambientes sem que se fizessem necessárias grandes modificações.

Uma área que tem muito da sua complexidade devido ao ambiente onde atua é a de sistemas de controle de robôs, pois aquele se altera com muita frequência e de uma maneira tal que não temos como determinar previamente o que pode acontecer.

Verificamos que um recurso que atende a esses requisitos são as chamadas redes neurais. Estas consistem de uma coleção de processadores interconectados que trabalham em paralelo. Este paralelismo permite que a rede examine diversos fatores simultaneamente e possibilita reduções no tempo computacional, razões muito atraentes em sistemas de tempo real.

Desta forma, consideramos uma boa opção estudarmos a utilização de redes neurais em aplicações de controle de robôs, o que significaria a junção de duas áreas. Este texto se divide basicamente em três partes: uma primeira onde fornecemos conceitos a respeito de redes neurais e robótica, a segunda onde definimos os problemas a serem tratados e a última versando sobre as investigações realizadas e resultados conseguidos.

## 2 Redes Neurais Computacionais

Uma razoável gama de modelos de redes neurais já foram desenvolvidos, entre eles podemos citar Hopfield, Hamming e o modelo por nós estudado e implementado, chamado de "feed-forward".

Já citamos que a rede consiste de uma coleção de processadores interconectados que trabalham em paralelo, cada um destes processadores denominaremos nodos ou neurônios.

Em cada um destes nodos temos um certo número de conexões de entrada que realizam a conexão do nodo com outros da rede, estas não são em número pré-definido e possuem um valor associado que é usado para ponderar a respectiva entrada, chamado peso da conexão. Todas estas entradas ponderadas são somadas e aplica-se uma função normalizadora sobre o resultado (no nosso caso a função sigmóide), este resultado normalizado é propagado para os neurônios que se ligam ao nodo pela saída.

Denominamos topologia à forma de disposição dos nodos e das conexões, no tipo de rede em questão, os nodos são organizados em camadas ou níveis, cada uma destas possui um número pré-determinado de nodos que recebem sinais de entrada de todos os nodos do nível anterior, sendo que o seu sinal se propaga para todos os nodos do nível superior.

Podemos entender como algoritmos a série de rotinas de manipulação da rede. Poderíamos dividi-los em dois grupos:

**Avaliação** consiste da simples propagação de valores de uma entrada através da rede até o nível de saída. Os resultados obtidos correspondem à resposta da rede àquele padrão de entrada.

**Treinamento** nestes algoritmos está contida a “mágica” das redes neurais, pois são eles que possibilitam o ajuste dos pesos das conexões da rede com vistas a obter as saídas desejadas a partir de determinadas entradas. Existem vários algoritmos, no nosso caso trabalhamos com o de retro-propagação, o qual pode ser sintetizado da seguinte forma:

- i) São conhecidos alguns valores de entrada e os respectivos padrões de saída esperados, aos quais podemos chamar amostra.
- ii) Para cada um dos valores de entrada da amostra:
  - ii.i) Realiza-se a avaliação obtendo-se as respostas da rede no nível de saída;
  - ii.ii) Com base nas saídas esperadas para cada amostra, a diferença entre estas e as saídas obtidas é usada para alterar os valores dos pesos de toda a rede utilizando alguma regra de aprendizagem, buscando a saída esperada, ou seja, a redução do erro.
- iii) Repete-se o passo ii) um certo número de vezes para cada uma das amostras, até que se atinja um resultado com nível de erro adequado.

Existem diversas variações do algoritmo de retro-propagação, em Josin[8] há aquela que utilizamos em nossos experimentos, não nos deteremos na dedução matemática da regra, mostrando somente a sua forma final:

A cada iteração de treinamento os valores dos novos pesos são calculados da seguinte forma (a notação é dada a seguir):

$$W_{pj}(t) = W_{pj}(t-1) + r\partial W_{pj}(t)$$

$$\partial W_{pj}(t) = a\partial W_{pj}(t-1) + (1-a)E_{pj}f'_j(S_{pj})$$

A grande diferença do algoritmo tradicional para o apresentado reside no cálculo do erro. Para pesos associados ao nível de saída, o cálculo é semelhante a regra delta (vide Josin[6]):

$$E_{pj} = (T_{pj} - O_{pj})f'_j(S_{pj})$$

Já para os pesos associados aos níveis intermediários, o erro é determinado de maneira diversa:

$$E_{pj} = f'_j(S_{pj}) \sum_k (E_{(p+1)k} W_{kj})$$

Índices:  $p$  - especifica o nível em questão da rede,  $(p+1)$  é a camada imediatamente superior;  $j$  - especifica o nodo enfocado no nível  $p$ ;  $k$  - especifica o nodo enfocado no nível  $p+1$ ;  $t$  - iteração ( $t-1$  é a iteração anterior).

Variáveis:  $W_{kj}$  - peso associado à conexão do  $k$ -ésimo nodo de  $(p+1)$  ao nodo em questão  $j$  em  $p$ ;  $W_{pj}$  - peso das conexões do nodo;  $\partial W_{pj}$  - valor que irá alterar o peso (delta);  $a$  - taxa de amortecimento;  $r$  - taxa de aprendizagem;  $E_{pj}$  - erro do  $j$ -ésimo nodo do nível  $p$ ;  $f_j(x)$  - função de ativação do nodo  $j$ ;  $f'_j(x)$  - derivada da função de ativação;  $T_{pj}$  - valor correto para a saída;  $O_{pj}$  - valor obtido através da avaliação;  $S_{pj}$  - resultado da função somatório do nodo.

O treinamento tem associado alguns valores que condicionam, de certa forma, o seu sucesso a que chamamos parâmetros de execução. Veremos que estes afetam diretamente a convergência do processo, assim:

-Aprendizagem: ( $r$ ) poderíamos interpretar como sendo quanto da informação atualmente apresentada será absorvida, via alteração de pesos da rede.

-Amortecimento: ( $a$ ) especifica a amplitude da mudança de pesos a ser feita. Seria como a confiança na exatidão e o grau de importância da informação em questão.

Estes dois valores são limitados ao intervalo [0..1].

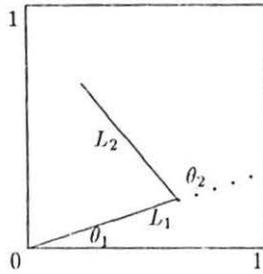


Figura 1: Braço mecânico e plano de atuação

### 3 Alguns Conceitos de Robótica

Para melhor entendimento das investigações realizadas, julgamos ser válido fornecer alguns conceitos de robótica.

Chamaremos de sistema robótico ao conjunto de elementos que interagindo entre si possibilitam a automação de alguma tarefa, são eles:

- A máquina propriamente dita, denominada robô;
- O ambiente de atuação desta máquina;
- A tarefa a ser realizada;
- O controlador, responsável por garantir a execução correta e adequada da tarefa, feita através do comando eletro-mecânico do robô.

Resumindo, o robô deve cumprir tarefas sobre o seu ambiente usando as ferramentas que possua (ou executando operações que seja capaz), sob o comando de um dispositivo controlador.

Um conceito que deveremos abordar é o referente a graus de liberdade, pois são os elementos básicos sobre os quais é feito o controle, estes podem ser de duas espécies:

- Translacionais: possibilidade do robô executar um movimento retilíneo em uma direção.
- Rotacionais: possibilidade do robô executar um movimento angular em torno de um eixo, normalmente constituindo uma junta.

Até agora vimos conceitos que se aplicam a robôs em geral, estes podem ser de variadas naturezas, desde unidades autônomas até dispositivos fixos com pouca capacidade de movimentação. Um exemplo poderiam ser os chamados braços mecânicos, que são uma variação utilizada para a consecução de atividades em um espaço limitado; assim, geralmente estes têm uma base fixa sobre a qual se assenta o dispositivo, é formado por um número de hastes ligadas por junções articuladas, na extremidade livre é fixado um órgão terminal que varia conforme a natureza da aplicação. Um exemplo está na Figura 1.

### 4 Aplicação

Nosso intuito no presente trabalho é a solução de problemas de robótica utilizando recursos neurais. A escolha da área robótica se deve ao fato de ser uma aplicação de tempo real e extremamente dependente do ambiente de execução, como já vimos anteriormente. Já que o nosso objetivo é centrado em redes neurais e não em robótica, constatamos a necessidade de definirmos explicitamente as aplicações a serem implementadas para que não nos percamos em questões que não são o objetivo do trabalho.

Inicialmente vamos definir o braço mecânico e seu espaço de atuação:

**Espaço de atuação** foi definido como espaço de atuação do braço mecânico um quadrante limitado por um plano com dimensões absolutas variando de zero a um conforme mostra a Figura 1.

**Braço mecânico** são dois segmentos de comprimento 0.75 ligados por uma junção articulada. Um primeiro segmento (que chamaremos  $L_1$ ) é articulado na origem do quadrante  $(0,0)$ , enquanto que a extremidade livre do outro segmento ( que chamaremos  $L_2$ ) indica o ponto de atuação do braço. No que tange a ângulos, chamaremos  $\theta_1$  ao ângulo formado pela reta dos pontos de ordenada zero (eixo das abscissas) e  $L_1$ , o qual varia entre noventa graus negativos e noventa graus positivos; chamaremos  $\theta_2$  ao ângulo formado pela direção indicada por  $L_1$  a partir da origem e  $L_2$ , tomado na junção entre  $L_1$  e  $L_2$ , este ângulo varia entre zero e cento e oitenta graus. Podemos visualizar o braço na Figura 1.

Definiremos agora três problemas distintos:

#### 4.1 Problema do Posicionamento

Podemos sintetizar o problema de posicionamento do braço mecânico acima definido através da seguinte indagação: “Quais os ângulos a serem aplicados às juntas do braço para que o mesmo atinja um ponto desejado dadas as suas coordenadas  $X$  e  $Y$ ?”

Consideraremos como premissa para a definição a especificação do braço feita acima.

Pela própria especificação dos ângulos das juntas ( $\theta_1$  e  $\theta_2$ ), evitamos a possibilidade de haver duas configurações do braço para atingir um mesmo ponto, o que ocorreria, por exemplo, se  $\theta_2$  variasse de zero a trezentos e sessenta graus.

Um tipo de operação muito comum em robótica é o cálculo da inversa, que nada mais é que a resolução matemática do nosso problema de posicionamento. Assim  $\theta_1$  e  $\theta_2$  seriam calculados da seguinte forma:

$$\theta_1 = \arctan(y/x) - \arctan(L_2 \sin \theta_2 / (L_1 + L_2 \cos \theta_2))$$

$$\theta_2 = \arccos(x + y - L_1 - L_2) / (2L_1L_2)$$

Ressaltamos que para calcular  $\theta_1$  precisamos saber o valor de  $\theta_2$ .

Um problema bastante sério associado a utilização deste método é o grande aumento de complexidade associado ao aumento de um ângulo ou dimensão (Coiffet[1], Craig[12]).

Vem daí a utilidade de estudarmos um meio de determinar a solução do problema de posicionamento que não implicasse em grande aumento da carga computacional pelo acréscimo de variáveis.

Este problema foi tratado em Josin[8], permitindo-nos fazer novas considerações:

- A nossa rede neural deveria receber como parâmetros de entrada as coordenadas do ponto a ser atingido e calcular como saída os ângulos  $\theta_1$  e  $\theta_2$  a serem aplicados;
- É utilizada uma versão modificada do “back-propagation” já vista anteriormente.

#### 4.2 Problema da Trajetória

Para podermos especificar uma arquitetura neural adequada para o problema em questão necessitaríamos determinar algumas premissas básicas inerentes ao sistema:

-Vamos supor que o braço não está habilitado a realizar movimentos amplos de uma única vez, assim cada movimento seria composto de uma série de pequenos passos; estimaremos como tamanho médio deste passo um décimo;

-Outra suposição que faremos é a respeito da sequência de movimentos do braço dados os dois ângulos a serem aplicados, desta forma, inicialmente seria realizado o movimento relativo a  $\theta_1$  e depois o relativo a  $\theta_2$ .

Embora este problema seja de fácil solução em duas dimensões, sua aplicação tridimensional é complicada, normalmente sendo feita no âmbito de dois planos, realizando o movimento inicialmente em um plano e depois no outro, é nosso intuito, numa fase posterior deste estudo, verificar se em uma aplicação tridimensional a rede responderia satisfatoriamente, não sendo necessário usar deste recurso.

Necessitamos então de uma ferramenta que forneça os pontos intermediários de uma trajetória estando a distância entre estes limitada ao passo permitido para o braço.

Se reformarmos a estratégia de treinamento e avaliação da rede de posicionamento podemos verificar algumas peculiaridades:

- o treinamento foi feito considerando um número limitado de pontos ocupando uma porção central do plano de atuação;
- a origem do braço (ponto de fixação) era na posição  $(0,0)$ .

Essa segunda observação nos é muito importante agora, pois implica em uma questão difícil, ou seja, qual teria sido o procedimento mais adequado no caso da origem do braço ser variável; devido ao aumento da complexidade do problema causado por este último fator, uma solução seria a translação das coordenadas a atingir de modo a podermos considerar o braço sempre fixo à origem, desta forma, o problema reduzir-se-ia ao anterior.

Problema semelhante ocorre também agora quando os pontos iniciais da trajetória podem ser variados criando infinitas combinações de ângulos e distâncias passíveis de serem usadas como amostra de treinamento. Podemos então tentar reduzir a complexidade do problema da seguinte forma:

- por uma translação simples, passa-se a considerar o ponto central do plano de atuação como a origem de todas as trajetórias;
- tendo em vista o fato de haver um passo limite para cada movimento do braço, necessitaríamos apenas da direção do movimento, que pode ser determinada através do ângulo formado pelos pontos inicial e final da trajetória em relação ao eixo das abscissas.

Assim a nossa rede neural determinadora de trajetórias a partir do ponto  $(0.5;0.5)$  receberia como entrada o seno e o cosseno de um ângulo devolvendo um ponto intermediário da trajetória na direção dada distante um passo pré-determinado.

### 4.3 Problema de Trajetória com Obstáculos

Em muitos ambientes de atuação de um braço mecânico, temos a presença de corpos que podem impedir o movimento do braço. Uma característica muito desejável seria a existência de recursos que alterassem a trajetória inicialmente traçada executando o contorno dos obstáculos.

Para podermos modelar este problema definiremos algumas novas premissas:

-Devido a extrema complexidade do assunto, faremos uma primeira simplificação que é a de considerarmos as hastes do braço estando em plano superior ao dos obstáculos, assim, a parte do braço que é suscetível a colisões consiste apenas do ponto de atuação.

-No ponto de atuação, temos oito sensores equiangularmente distribuídos que detectam obstáculos distantes um valor pré-determinado e que fornece como saída a distância normalizada entre o sensor e o obstáculo, no nosso caso, 0 significa ausência de obstáculo enquanto 1 iminência de colisão.

O nosso modelo basear-se-á na abstração de que uma trajetória em um ambiente com obstáculos está sujeita a duas forças: uma atrativa exercida pelo ponto de destino e outra repulsiva exercida pelos obstáculos. Assim o movimento pode ser visto como a resultante da composição destas forças, quando o ponto de atuação se aproxima de um obstáculo, a força repulsiva obriga uma mudança de direção de modo que não haja colisão.

Verificando Borenstein[10], fica patente ser a complexidade do problema bem maior do que o simples desvio do obstáculo por composição de forças, podem ocorrer diversas situações

por ele tratadas que devemos também modelar.

Um primeiro problema surge quando temos “paredes” que devem ser contornadas, se considerarmos apenas o esquema acima proposto teremos as seguintes questões:

-Tendo em vista a força repulsiva ser variável, a trajetória descrita pelo PA (ponto de atuação) será sinuosa, pois este tenderá a se aproximar do obstáculo quando a repulsiva for menor, causando o seu aumento e novo afastamento.

-Outra ocorrência dependente da extensão da parede é a seguinte situação: a parede estando perpendicular à força atrativa, gerando uma força repulsiva de mesma direção e sentido contrário à primeira, assim, o nosso ponto mover-se-ia apenas nesta direção, aproximando e afastando do obstáculo até que atingisse um ponto de equilíbrio quando estagnaria.

-Se houver mais de uma parede teremos um problema ainda mais sério. Um exemplo ocorre quando temos duas paredes formando um ângulo (vide Figura 3c). Se já estivermos em condições de realizar o caminhamento paralelo proposto acima, e em determinado instante a direção da trajetória formar um ângulo com a força atrativa maior que noventa graus, quando a força repulsiva do obstáculo cessar, o PA retornará novamente à concavidade formada pelas paredes.

Ainda segundo Borenstein[10], estes seriam os problemas mais complexos de serem solucionados quando desejamos evitar colisões. Devemos inicialmente definir quais seriam as condições de ocorrência de cada uma das situações.

O primeiro e segundo casos são caracterizados pela necessidade de um caminhamento paralelo ao obstáculo onde a direção de deslocamento não diferenciasse por mais de noventa graus da direção do objetivo. Uma maneira de determinarmos a necessidade de um caminhamento paralelo é verificando os resultados do sensoriamento, assim, se para um dado sensor que apresente um resultado acima de um limiar e os sensores a ele adjacentes com resultados semelhantes e menores que o do primeiro. Esta situação cessaria quando os sensores em questão apresentassem seus resultados mais baixos que um certo valor pré-determinado.

No terceiro item, a maneira mais adequada de realizar a detecção é a mesma adotada em Borenstein[10], ou seja, quando a direção de deslocamento formar um ângulo maior que noventa graus com a força atrativa, configura-se a necessidade de contorno que só cessa quando o ângulo supra citado for menor que noventa graus.

Tendo em vista o fato de todos os casos terem como solução o caminhamento paralelo ao obstáculo, podemos unificar as variações do problema de contorno, cuja trajetória solução é aquela que seja perpendicular à força repulsiva na direção que vinha sendo realizado o deslocamento.

Ou seja, a solução do problema de trajetória com obstáculos seria realizada com duas redes, uma primeira que alteraria o valor do ponto determinado na rede de trajetória linear de acordo com a configuração momentânea de obstáculos e uma outra rede encarregada do contorno que determina os incrementos a serem aplicados na posição atual de PA de modo que o obstáculo seja evitado.

O algoritmo de trajetória é apresentado na Figura 2.

## 5 Redes Neurais e Robótica

Tomamos como ponto partida o artigo de Josin[8], que trata da utilização de uma rede neural para comandar um braço mecânico com dois graus de liberdade rotacionais.

Para tal é proposta uma topologia de rede neural de três níveis, com dois nodos de entrada (coordenadas x e y da posição desejada), dois de saída (os ângulos a serem aplicados) e trinta e dois nodos intermediários. O algoritmo de treinamento da rede é uma versão modificada do “back-propagation” já apresentada.

Durante este artigo também é relatada a capacidade de interpolação do modelo proposto,

```

PI:=posição atual
PF:=ponto destino
while Distância(PI,PF) < Eps do
begin
  Determina PI com rede de trajetória
  if Existe Obstáculo
  then begin
    if Vai Contornar
    then Determina PI com rede de contorno
    else Altera PI com rede de obstáculo
  end
  Posiciona no novo PI com rede de posicionamento
end
end

```

Figura 2: Algoritmo de trajetória

ou seja, é necessário que se treine apenas um certo número de pontos, pois a rede seria capaz de interpolar os demais, este número é definido como em torno de dez.

Partimos então para investigações práticas das propostas feitas, verificando inicialmente a capacidade de aprendizagem da rede em questão. Esta é influenciada por vários fatores:

-Parâmetros de execução: já citados acima, normalmente são responsáveis por uma aprendizagem lenta, rápida e até por uma não aprendizagem, além de afetarem o grau da mesma.

-Topologia: alguns estudiosos defendem a idéia que uma rede armazena uma quantidade de padrões em torno de 15% do número de nodos da rede. Sabemos que algumas topologias são mais adequadas que outras, entretanto não temos ciência de uma maneira formal de estabelecer arquiteturas que possibilitem melhor desempenho, embora saibamos ser a influência desta maior que a própria estratégia de treinamento adotada.

-Inicialização de pesos: outra questão em aberto é no que tange a determinação de pesos mais adequados para o estado inicial de uma rede, o que poderia ser um fator de melhor e mais rápida aprendizagem, mas também não temos nada muito definido sobre o assunto na literatura.

## 6 Investigações sobre o problema de posicionamento

Inicialmente nossas investigações centraram-se na verificação da capacidade de aprendizagem do modelo proposto, isto foi feito de maneira empírica, para melhor entendermos os resultados obtidos, vamos especificar o ambiente de execução:

**Padrões** entende-se por padrões àquelas informações que desejamos que a rede aprenda, em Josin[8] é relatado que a rede aprende bem até dez pontos. Durante os testes a serem descritos utilizamos conjuntos de pontos variados a fim de verificar o mais adequado; cada um dos diferentes padrões foi denominado de amostra;

**Algoritmo** todo o nosso trabalho foi feito através de repetições de determinadas condições de execução por um certo número de iterações. Cada iteração compreendeu:

Para cada ponto da amostra:

- a) forneça os valores de  $X$  e  $Y$  e avalie os de  $\theta_1$  e  $\theta_2$ ;
- b) compare  $\theta_1$  e  $\theta_2$  calculados com os corretos;
- c) extraia os dados para as medidas de verificação (vide a seguir);
- d) treine a rede com base nos valores corretos.

**Convergência** um conceito bastante relativo é o de convergência, à primeira vista podemos entender como sendo a tendência à proximidade dos resultados obtidos pela rede aos valores corretos, ou seja, é a variação do erro apresentado pela rede no decorrer de um certo espaço de tempo ou número de iterações durante o processo de aprendizado.

**Medidas de verificação** com base nos resultados obtidos nas sucessivas avaliações, podemos avaliar a qualidade do treinamento por alguns critérios:

- a) Desvios individuais - para podermos avaliar a convergência de cada ponto da amostra treinado calculamos a diferença percentual entre os ângulos corretos e os fornecidos pela rede;
- b) Energia - considera-se energia do sistema o somatório absoluto dos erros (diferença entre os ângulos encontrados e corretos). Desta forma, quanto menor a energia, melhor a convergência obtida.
- c) Inversões - em um sistema convergente, tanto os valores da energia quanto os desvios individuais geralmente são sequencialmente decrescentes, se isso não acontece, podemos chegar a uma situação de divergência. Assim uma inversão ocorre quando uma medida de verificação das citadas é maior que a obtida na iteração anterior.
- d) Média dos desvios individuais - é a média das variações entre os desvios individuais ou energia, indicando quanto estes se alteram a cada iteração;
- e) Erro interpolação - como já foi dito anteriormente, o modelo deveria ter uma capacidade interpolatória, assim para cada amostra avaliamos uma certa quantidade de pontos internos equitativamente distribuídos e calculamos a distância média entre os pontos encontrados e os corretos.

Para os primeiros testes realizados seguimos exatamente o proposto em Josin[8], desta forma foi utilizada uma rede com três níveis e trinta e dois nodos na camada intermediária e o algoritmo de treinamento foi a versão modificada do "back-propagation" já apresentada. Uma observação é que nestes primeiros testes sempre que nos referirmos a valor alto e baixo dos parâmetros de execução, estes serão 0.9 e 0.1 respectivamente. Quanto à amostra, foi utilizado um conjunto de nove pontos, sendo um deles o centro do plano de atuação (0.5; 0.5) e os demais distribuídos pela borda do quadrado determinado por 0.2 e 0.8 em ambos os eixos. Todas as investigações descritas a seguir foram realizadas utilizando um simulador de redes neurais por nós implementado.

## 6.1 Processo de treinamento

Começamos realizando um treinamento exaustivo onde aplicamos parâmetros já utilizados anteriormente em experimentos com funções combinatórias (XOr, And, conversão binário-BCD e teste de paridade), onde obtivemos resultados satisfatórios com um valor baixo para aprendizagem (um décimo) e um valor alto para amortecimento (nove décimos). Foi realizado um grande número de iterações com esta configuração e notou-se que após uma certa quantidade de iterações, ocorria uma estagnação em um patamar.

Partimos então para uma verificação em torno dos parâmetros inicialmente utilizados com vistas a descobrir valores que nos permitissem uma melhor convergência, isto foi feito variando a aprendizagem e amortecimento isoladamente.

Analisando os resultados, vemos que o aumento da aprendizagem implica em uma convergência mais rápida. Já o aumento do amortecimento implica em uma convergência mais lenta, embora a influência da sua variação no processo seja bem menor.

Considerando a tendência observada de maior convergência com valores de aprendizagem mais altos, resolvemos verificar todo o intervalo de parâmetros para nos certificarmos que esta tendência se confirmaria. Podemos extrair três conclusões a partir deste teste:

- Quanto maior a aprendizagem, maior a convergência (pelo menos em 500 iterações);
- Quanto maior a convergência, maior o número de inversões ocorrido no treinamento, o que pode ser entendido como uma medida de instabilidade do sistema, o que não é necessariamente ruim como veremos posteriormente;

-Podemos notar que nos testes onde o amortecimento era maior, o número de inversões foi menor, embora isso não tenha alterado significativamente a convergência nem sido fator determinante do número de inversões, confirmando nossas observações anteriores de sua menor importância ante a da aprendizagem.

O algoritmo de treinamento tenta minimizar o erro associado à configuração atual da rede, se visualizarmos este processo graficamente, veremos que o valor do erro associado às possíveis configurações da rede podem ser vistas como uma superfície irregular com vários pontos altos e baixos, a minimização é feita através de um caminhar pela superfície na direção oposta ao gradiente da mesma, a amplitude deste caminhar é dada pelos parâmetros de execução, principalmente a aprendizagem, assim, quando esta é alta é dado um grande “passo”, caso contrário, um pequeno “passo”; o que nós pretendemos é atingir o ponto mais baixo da superfície (menor erro), durante o processo, entretanto, podemos ter alguns problemas:

- Atingirmos um buraco que não é o mais fundo globalmente, mas a menor cota de uma área limitada, o que chamamos de um mínimo local;

- Darmos passos tão grandes, de modo que conseguimos apenas circular o ponto desejado, sem realmente atingí-lo.

Verificando Kickpatrick[9], tomamos contato com a técnica denominada “simulated annealing”, a qual consiste basicamente do seguinte: para evitar a convergência a mínimos locais, sugere-se inicialmente verificar caminhos aparentemente piores tendo em vista a hipótese de estarmos em uma região de mínimo local, na qual não devemos permanecer, posteriormente o tamanho do passo deve ser reduzido. No nosso caso, isso é feito quando usamos uma alta aprendizagem, o que possibilita um caminhar rápido e muitas vezes acidentado (alto número de inversões). Para resolver o segundo problema citado acima, necessitaríamos de parâmetros de execução que nos possibilitassem atingir a menor cota da região que estivéssemos, o que seria feito com caminhamentos menores e mais apurados. Transferindo estas formulações para o nosso modelo, poderíamos começar o treinamento com uma alta aprendizagem e um baixo amortecimento fazendo uma ampla verificação do universo de soluções e completando o trabalho com um treinamento mais direcionado que proporcionasse uma convergência mais efetiva através de uma baixa aprendizagem e alto amortecimento (menor número de inversões).

Realizamos então uma série de sequências com duas mil iterações, divididas em segmentos de quinhentas iterações que utilizaram cada um dos pares A (baixa aprendizagem) ou B (alta aprendizagem).

Analisando os resultados percebemos que os testes que iniciaram com alta aprendizagem tiveram uma diminuição brusca de energia, enquanto que na curva referente ao parâmetro A isso não acontece.

Podemos observar ainda as seguintes peculiaridades verificando os resultados posteriores às quinhentas primeiras iterações com B:

- Ocorre uma estagnação se continuamos a aplicar B;
- Ao aplicarmos A, ocorre uma brusca diminuição de energia, seguida de uma estagnação;
- Se após a aplicação de A, aplicarmos novamente B, ocorre um salto para valores próximos aos obtidos anteriormente com o último parâmetro, seguindo-se uma estagnação;
- Se novamente aplicarmos A, ocorre uma nova diminuição brusca, seguida de estagnação;

Pelas reações observadas, podemos afirmar ter atingido os pontos de estabilização do parâmetros A e B, a partir dos quais a diminuição de energia se faz lentamente.

Um fato interessante é que obtivemos o patamar referente a  $\Lambda$  em pouco mais de quinhentas iterações, enquanto que a repetição exaustiva exigiu mais de duas mil iterações para alcançar este valor.

Dando continuidade a esta investigação no que diz respeito a minimizar o número de iterações necessárias para uma convergência razoável, observamos dois fatos interessantes:

-Para que ocorra o salto descrito quando da aplicação de  $A$  após  $B$ , são necessárias por volta de quatrocentas iterações com o último parâmetro, para quantidades menores, este salto não ocorre;

-Quando da aplicação de  $A$ , ocorre um salto brusco nas três primeiras iterações, seguido de um salto divergente de menor amplitude e posterior estagnação.

Percebemos que o aumento da aprendizagem melhora significativamente o resultado do treinamento. Feitos os testes, a convergência foi maior tanto quanto o valor da aprendizagem utilizado, da mesma forma o salto ocorre atingindo níveis melhores. Um dado interessante é que conforme se aumenta a aprendizagem também são necessárias menos iterações para que seja atingido o patamar de saturação (ou estagnação), entretanto, o valor da energia destes patamares é tão maior quanto a aprendizagem utilizada.

Se procurarmos entender estas observações, concluímos que quanto maior o passo que se dê, mais rápido se chega ao objetivo, entretanto, mais imprecisa é a chegada; assim, quando executamos um treinamento mais direcionado, atingimos os pontos desejados identicamente, com a vantagem de um menor número de iterações. Podemos sintetizar da seguinte forma: quando fazemos uma busca ampla é necessária uma certa proximidade dos melhores valores para que o treinamento direcionado surta o efeito esperado. Não sabemos como determinar este valor, para o nosso caso ele foi encontrado empiricamente como sendo em torno de quatrocentas iterações.

Uma outra importante observação que fizemos foi no tocante a validade da energia como medida de avaliação, o que foi feito comparando os valores de energia com os respectivos erros de interpolação.

No que tange a capacidade extrapolatória, se tomarmos o parâmetro de melhor resultado e compararmos seus erros médios de interpolação e extrapolação, veremos que ocorre uma degradação da precisão.

## 6.2 Variáveis do processo de aprendizado

Uma primeira investigação versou sobre topologias e teve por principal finalidade determinar topologias alternativas para o modelo. Assim, para sequências de treinamento idênticas comparamos topologias diferentes, variando número de nodos na camada intermediária e número de camadas.

As topologias de quatro níveis apresentaram resultados piores em relação às de três níveis, enquanto que estas últimas mostraram resultados semelhantes, principalmente nas sequências de maior convergência. Um fato realmente interessante é que a topologia de três níveis com dezesseis nodos na camada intermediária apresentou resultados pouco piores que os da com trinta e dois nodos ou quarenta e oito nodos na mesma camada, o que pode implicar em economia de tempo e processamento. Podemos concluir que o aumento do número de camadas não melhora o desempenho da rede, importando realmente apenas o número de nodos em paralelo do nível intermediário.

Verificamos se o aumento ou a diminuição da amplitude desta amostra, ou seja, a área coberta por ela, teria alguma influência. Para uma avaliação independente da amplitude utilizada usamos como medida o erro total de interpolação e extrapolação.

Os melhores resultados foram obtidos quando utilizamos uma amplitude média ocupando 36% da área de atuação. Amplitudes maiores ou menores implicaram em resultados piores. Todas as amplitudes testadas ocuparam um espaço central do plano de atuação.

Esta última conclusão tem sua importância quando reflete a influência da natureza da amostra no sucesso do processo de treinamento, indicando ter a rede uma limitação na amplitude de dados que podem ser “aprendidos”.

Dissemos anteriormente que na proposta do modelo em Josin[8], foi afirmado que a rede aprende bem em torno de dez pontos, tanto que, baseados nesta afirmação, usamos nove pontos nas investigações feitas. Verificamos então, o que aconteceria se utilizássemos cinco e treze pontos na amostra, mantendo constantes os outros parâmetros de treinamento.

Verificamos que o número de pontos mais adequados realmente é nove, pois embora apresente um resultado inicial pior, o seu resultado final é superior. Podemos fazer uma interessante analogia neste caso: em muitos campos da ciência, quando vamos estudar um assunto, normalmente não estamos aptos a estudá-lo em profundidade, pois isso poderia ser um tanto confuso (amostra de 13 pontos), enquanto que informações superficiais não dariam uma correta noção do que está sendo estudado (amostra de 5 pontos), assim, o ideal é que obtenhamos uma relativa massa de informações sobre o assunto estudado de modo a podermos definir as linhas gerais do mesmo.

### 6.3 Função aprendizagem

Nas investigações relatadas até o presente momento, realizamos os treinamentos usando sempre os mesmos parâmetros de execução por um certo período, sendo estes alterados independente da configuração momentânea da rede. Seria bastante desejável a obtenção de uma função que determinasse, por exemplo, a taxa de aprendizagem a ser utilizada de acordo com o erro instantâneo. Poderíamos, desta forma, evitar os períodos divergentes obtendo uma convergência mais rápida. Na literatura por nós verificada não encontramos nada sobre o assunto.

Na tentativa de determinarmos a função desejada utilizamos três tipos de funções: linear, bi-linear (duas retas interligadas) e sigmóide. Os melhores resultados foram obtidos para aquelas funções que apresentam maiores ordenadas para valores de erro abaixo de 30%.

Estes resultados nos indicam que manter uma alta taxa de aprendizagem para erros pequenos é a melhor estratégia, o que justifica o sucesso dos testes anteriores (onde se usou função degrau). Resta determinar quando devemos mudar os parâmetros.

Uma proposta seria o desenvolvimento de algoritmos que automaticamente fizessem esta redução a partir da detecção de uma sequência divergente de certa duração; outra seria a aplicação do degrau para valores de erro abaixo de um certo limiar.

No primeiro caso, fizemos testes usando como durações para sequências divergentes 10, 30, 50, 70 e 90 iterações; os resultados observados foram bastante semelhantes, sendo o melhor o referente a 50 iterações. Entretanto, a melhoria obtida não é suficiente para podermos afirmar ser este o melhor valor. Outra conclusão que podemos tirar é que a rede não dá sinais de estar se aproximando do seu limite de convergência, simplesmente diverge.

Já na aplicação da função degrau, o melhor desempenho é quando não há aplicação do degrau em momento algum, ou seja, o degrau só é aplicado quando o erro é zero, o que não acontece.

Depois de todas estas tentativas, podemos concluir que todos os resultados indicam que não há, a priori, uma função que determine a taxa de aprendizagem a ser aplicada em função das variáveis do processo, embora muitas vezes mais acidentada, a utilização de uma aprendizagem alta até que haja uma situação de divergência é a melhor maneira encontrada.

### 6.4 Aplicação prática: construção de círculos

Um exemplo interessante onde podemos verificar a exatidão dos dados fornecidos pela rede neural de posicionamento é quando realizamos a construção de círculos. Pela própria forma

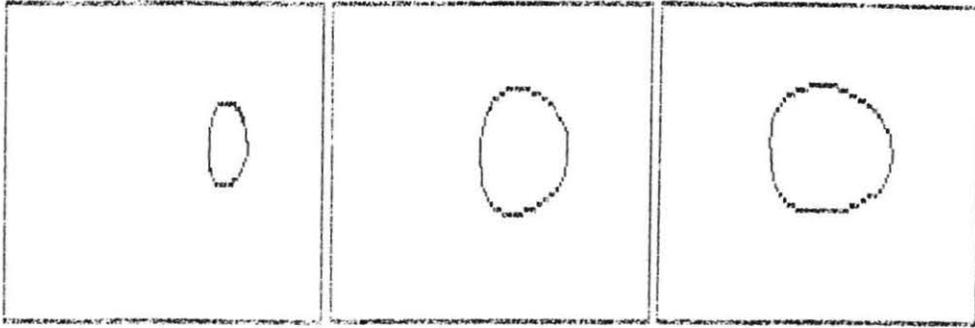


Figura 3: Aplicação prática: construção de círculos

geométrica que estes apresentam podemos visualizar o grau de aprendizado da rede.

Desta forma, usamos a sequência de treinamento de melhor resultado (com os parâmetros 0.99/0.1), para desenharmos os círculos. Eles foram construídos com os pontos resultantes da avaliação pelas redes especificadas dos pontos de um círculo de raio pré-determinado.

Para acompanharmos o processo de aprendizagem, fizemos a construção com as redes obtidas nas primeiras 100 e 200 iterações onde vemos claramente a melhora do resultado de uma para outra, tendência confirmada em 500 iterações. Para ilustrar a melhora conseguida com o salto quando da aplicação da baixa aprendizagem após 500 iterações, como mostrado na Figura 3.

## 7 Investigações sobre o problema de trajetória

Uma primeira modelagem para o problema de trajetória seria aquela que considera o ambiente de atuação do braço desprovido de obstáculos, ou seja, a trajetória entre dois pontos seria retilínea.

Esclarecemos desde já que todo o estudo de trajetória aqui apresentado é no sentido de determinarmos os pontos intermediários que compõem uma determinada trajetória. Não nos preocuparemos com problemas mais ligados à área de robótica como a maneira que o braço muda de uma configuração para outra e outros aspectos mais inerentes à robótica (descritos em Craig[12]).

Conforme definido anteriormente, dados os pontos inicial e final da trajetória, determinamos a direção (ângulo com a horizontal) do vetor que vai do primeiro ao segundo, do qual são apresentados à rede o seno e cosseno.

Na presente investigação tivemos basicamente duas medidas de avaliação:

**Energia** identicamente ao problema de posicionamento, definiremos energia como sendo o somatório das diferenças absolutas entre a amostra e os resultados da rede para ela.

**Distância** realizamos a avaliação para 40 pontos ao redor do centro do quadrante, somando as distâncias entre os pontos encontrados e corretos.

Nosso trabalho começou com uma varredura do espaço de parâmetros de aprendizagem e amortecimento, onde, à semelhança dos experimentos de posicionamento, os de maior taxa de aprendizagem tiveram melhor desempenho, estas taxas variaram entre 0.1 e 0.9.

Para estes testes iniciais fizemos uma analogia ao modelo de posicionamento. Assim usamos oito pontos igualmente distribuídos ao redor do centro do plano de atuação distantes 0.1 daquele.

Também variamos o número de pontos da amostra comparando-os com a quantidade usada inicialmente. Foram feitos testes com 4, 8 e 16 pontos. Percebemos ser realmente um número em torno de dez o mais adequado para o tamanho da amostra de treinamento.

Os pontos são gerados numa direção dada distantes um raio pré-fixado do ponto original, que é no centro do quadrante. Avaliamos as reações da rede para diferentes amostras, ou seja, quais os valores mais adequados para treinamento. O valor mais adequado foi o usado inicialmente (0.1).

## 8 Investigações sobre trajetórias com obstáculos

Vimos anteriormente que seriam necessárias duas redes neurais para que solucionássemos o problema corrente, uma para realizar o desvio de obstáculos e outra que possibilitasse o contorno de estruturas mais complexas.

Desta forma, tínhamos uma primeira rede que receberia como entrada os valores dos sensores (já normalizados) e respondendo com um ponto que fosse na direção do vetor repulsivo, esta distância também deveria ser constante em relação ao ponto atual, numa estratégia semelhante à utilizada na trajetória sem obstáculos.

A segunda rede (contorno) receberia os mesmos sensores como entrada e responderia com um ponto que estivesse perpendicularmente direcionado em relação à força repulsiva, sendo o seu sentido determinado pela variação entre os dois últimos pontos.

Se observarmos bem, veremos que estas redes diferem em apenas um aspecto: a orientação do ponto de saída. A saída de ambas seria idêntica se rotacionássemos o conjunto de sensores de um ângulo reto, sendo o sentido definido como no último parágrafo.

Podemos concluir que precisamos de apenas uma e não duas redes, com a ressalva que para obtermos os resultados desejados da segunda devemos realizar a rotação adequada dos sensores de entrada.

Outra consideração que devemos fazer é no sentido de que as saídas também são utilizadas de forma diferenciada, assim enquanto os resultados da rede de obstáculos são adicionados ponderadamente aos resultados da rede de trajetória para que se obtenha o incremento do novo ponto, os da rede de contorno já constituem o valor a ser adicionado.

Uma questão bastante difícil foi no tocante a determinar as constantes físicas do sistema como raio de atuação dos sensores, limiares de ativação das redes e valores a ponderarem os resultados de cada rede.

Como medida de avaliação usamos o erro total da amostra que consiste do somatório das distâncias entre os pontos obtidos e os da amostra. Consideramos não serem necessárias outras medidas pela própria forma de utilização conjunta das várias redes.

Novamente se confirmou como mais adequada a utilização de taxas de aprendizagem altas, já no tocante à topologia utilizamos uma com 32 nodos no nível intermediário por ter a melhor relação custo-benefício.

Podemos visualizar o funcionamento do sistema na Figura 4 onde percebemos a formação de trajetórias retilínea, caminho paralelo e contorno respectivamente.

## 9 Conclusões

A utilização de “parâmetros conjugados” mostrou-se bastante mais eficiente que a utilização isolada de apenas um par de parâmetros, mesmo que exaustiva. Entretanto ainda ocorre uma

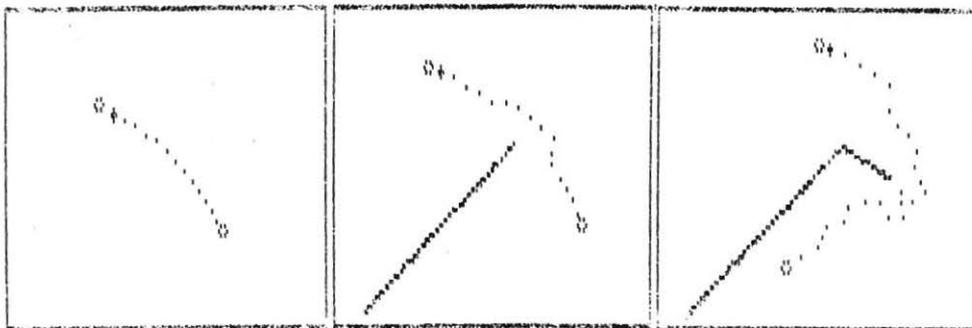


Figura 4: Formação de trajetórias com redes neurais

estabilização após certo número de iterações, embora esta seja com um grau de convergência melhor e superior aos alcançados anteriormente.

Todo este trabalho foi feito no sentido de investigarmos a capacidade interpolatória da rede, o que foi conseguido. Outro objetivo foi a determinação dos melhores parâmetros que influenciam o treinamento para que este seja rápido e efetivo, cuja importância seria quando de uma utilização prática do modelo em outras aplicações ou ambientes.

## 10 Perspectivas Futuras

Concluídos este trabalhos, podemos verificar a adaptação dos modelos a um maior número de graus de liberdade, e até dimensões (uma aplicação tridimensional), o que significaria voltarmos ao início da investigação, buscando talvez um modelo geral para este tipo de aplicação.

## Referências

- [1] Coiffet, Philippe. *Robot Technology-Modelling and Control*, Volume 1, Prentice-Hall, London, 1983.
- [2] Rumelhart, D.E. & McClelland. *Parallel Distributed Processing : Explorations the Microstructure of Cognition*, Volume 1: Foundations, MIT Press, 1986.
- [3] Lippmann, Richard P. "An Introduction to Computing With Neural Nets", *IEEE ASSP Magazine*, April 1987, 4-22.
- [4] Rangwala, Sabbir S. & Dornfeld, David A., "Learning and Optimization of Machining Operations Using Computing Abilities of Neural Networks", *IEEE Transactions on Systems, Man and Cybernetics*, Vol 19, No 2, March/April 1989.
- [5] Jones, William P. & Hoskins, Josiah. "Back Propagation", *Byte*, October 1987, 155-162.
- [6] Josin, Gary. "Neural Networks Heuristics", *Byte*, October 1987, 183-192.
- [7] Kung, Sun-Yuan & Hwang, Jeng-Neng, "Neural Networks Architectures for Robotic Applications", *IEEE Transactions on Robotics and Automation*, Vol 5, No 5, October 1989, 641-657.

- [8] Josin, Gary. "Integrating Neural Networks with Robots", *AI Expert*, April, 1987, 51-58.
- [9] Kickpatrick, S., Gelatt, C.A. Jr & Vecchi, M.R. "Optimization by Simulated Annealing", *Science*, No 220, 1983, 671-680.
- [10] Borenstein, J. & Korem, Y. "Real-Time Obstacle Avoidance for Fast Mobile Robots", *IEEE Transactions on Systems, Man and Cybernetics*, No 5, Vol. 19, September/October 1989, 1179-1186.
- [11] Nagata, S., Sekiguchi, M. & Asakawa, K. "Mobile Robot Control by a Structured Hierarchical Neural Network", *IEEE Control Systems Magazine*, April, 1990, 69-76.
- [12] Craig, J. *Introduction to Robotics*, Addison Wesley, 1986.