Instituto de Ciências Matemáticas de São Carlos

ISSN - 0103-2569

COMPORTAMENTO DE EVITAR OBSTÁCULO PARA SISTEMA DE NAVEGAÇÃO

FÁBIO ROBERTO MELFI
RENATO CORREIA DE BARROS
ANDRÉ C. P. CARVALHO

Nº 61

RELATÓRIOS TÉCNICOS DO ICMSC

São Carlos
Ago./1997

Comportamento de Evitar Obstáculo para Sistema de Navegação

Fábio Roberto Melfi **Renato Correia de Barros** **André C. P. Carvalho**

Departamento de Ciências da Computação e Estatística - ICMSC - USP e ILTC

Laboratório de Inteligência Computacional - LABIC

Av. Dr. Carlos Botelho, 1465 - Caixa Postal 668

CEP 13560-970 - São Carlos - SP - Brasil

e-mail: { fmelfi, rbarros, andre }@icmsc.sc.usp.br

Abstract

This work presents a neural network based solution to the problem of obstacle avoidance for a mobile robot's navigation system.

In order to safely move a robot in its environment, we use a navigation system that is able to avoid obstacles. It is too dificult to derivate an algorithm that can do this task in a step by step fashion, because there are too many variables involved. However, it is possible to provide a set of input / output examples that can be fed to a neural network, so this is the approach used.

Key-words: Robotics, Navigation, Neural Network, PCRobots, SNNS.

Resumo

Neste trabalho, é apresentada uma solução baseada em redes neurais para o comportamento de evitar obstáculo, em um sistema de navegação para um robô móvel.

Para mover um robô em seu ambiente, utiliza-se um sistema de navegação, que possa desviar dos eventuais obstáculos. Derivar um algoritmo que realize esta tarefa passo a passo é difícil, devido a grande quantidade de fatores variáveis. Entretanto, é possível obter exemplos de entradas e saídas, que podem ser alimentadas em uma rede neural. Esta é a abordagem utilizada.

Palavras-Chaves: Robótica, Navegação, Rede Neural, PCRobots, SNNS.

1. Introdução

Robôs vem sendo utilizados na prática, principalmente em indústrias, desde os anos 60. Em sua maioria, ainda hoje, os robôs são na verdade braços mecânicos. Estes braços têm seu campo de trabalho previamente definidos, e são imóveis, o que lhes dá flexibilidade limitada. Entretanto, há uma tendência crescente para o desenvolvimento de robôs móveis, também conhecidos como AGV - Autonomous Guided Vehicles. Estes robôs são capazes de se movimentar em seu ambiente, e por isso abrem espaço para novas aplicações, que não seriam conseguidas sem esta tecnologia.

Para que os robôs móveis sejam efetivos, e encontrem larga utilização, devem ser adaptativos, comunicativos e seguros. Mas uma coisa todos os robôs móveis tem em

comum: a necessidade de um sistema de navegação. Este sistema de navegação é responsável por guiar o robô a seu destino rapidamente e sem colidir com obstáculos.

Existem muitas abordagens para a resolução do problema de navegação. Neste artigo, exploramos a alternativa de aplicar a tecnologia de Redes Neurais ao problema da navegação. Redes Neurais são sistemas inspirados no funcionamento do cérebro humano, e que possuem a capacidade de resolver problemas através de aprendizado, ou seja, não é necessário que expliquemos passo a passo como resolver o problema. Basta fornecermos suficientes exemplos de entradas e saídas desejadas, que o sistema de rede neural irá aprender como mapear as entradas nas saídas.

O artigo está estruturado da seguinte forma: os requerimentos de um sistema de navegação são definidos, então são apresentadas as soluções mais comuns para o problema. A seguir, expomos a tecnologia de redes neurais, e como ela pode ser aplicada ao sistema de navegação. São mostrados os resultados obtidos, e finalmente as conclusões do trabalho são mostradas.

2. Sistema de Navegação

Em um determinado ambiente, o robô deve chagar a seu destino o mais rapidamente possível, evitando os obstáculos que encontrar pelo caminho. Neste trabalho, o ambiente é dados por um programa de simulação de robôs, o PCRobots, versão 1.3. Neste software, o usuário pode definir um labirinto de 100x100 posições, onde cada posição pode estar ocupada ou desocupada. Assim, é possível formar muros, corredores,

obstáculos, etc. O robô deve então ser guiado automaticamente dentro deste labirinto, evitando os obstáculos. A Figura 1 exemplifica um labirinto.

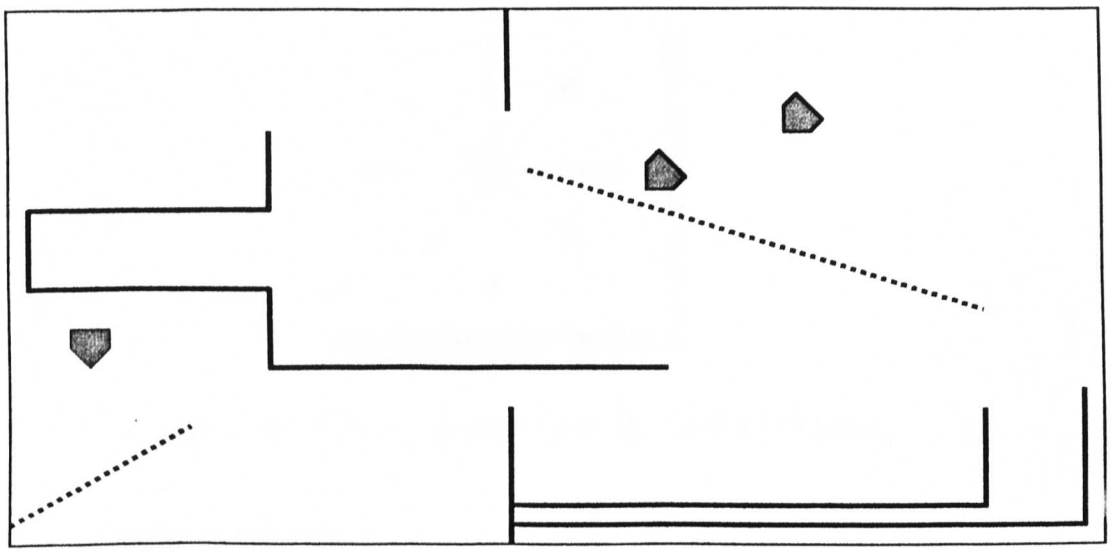

Figura 1. Um labirinto usado pelo programa PCRobots.

O robô dispõe de uma matriz de sensores de tamanho 9x9, cujo ponto central (Linha 4, Coluna 4) é representado pelo próprio robô, ou seja, ele "enxerga" 4 posições adiante em cada lado. A Figura 2 mostra o robô inserido em sua matriz de sensores.

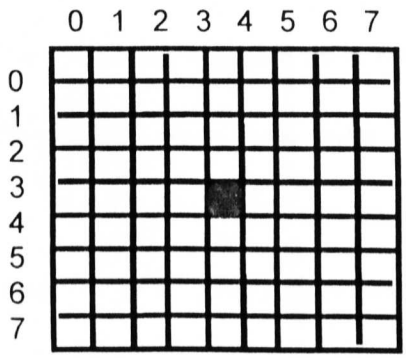

Figura 2: Mapa dos sensores do robô. O robô está na posição (4,4), representado por um círculo.

As ações disponíveis ao robô incluem girar de 0^{o} a 359^{o}, e andar a uma determinada velocidade .

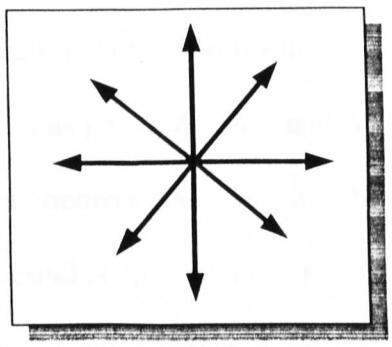

Figura 3: O robô pode andar em qualquer direção, de 0 a 359 graus.

3. Abordagens ao Problema

Diversas propostas para um sistema de navegação têm sido feitas, cada uma com diferente grau de sucesso [2,3,4]. A mais antiga provavelmente é a extensão dos métodos de controle de braços mecânicos, conhecido como robótica clássica. Nesta abordagem, o ambiente e o robô são modelados matematicamente, e uma solução é obtida através de cálculo.

A meta é obter a trajetória ótima pela qual o robô possa chegar no menor tempo possível. Infelizmente esta solução exige que tanto o robô quanto o ambiente sejam conhecidos em detalhes *a priori*, ao ponto em que modelos apurados possam ser construídos. Isto nem sempre é possível, principalmente porque o ambiente pode ser dinâmico, ou seja, variar de navegação para navegação, além dos erros de leitura e acionamento do robô não permitirem a correta geração e aplicação do modelo matemático. Por exemplo, se o modelo exige que em determinado ponto o robô acione sua junta 3 em 15^{o}, é possível que devido a problemas mecânicos, esta junta se desloque

na realidade apenas 13°. Este tipo de erro se acumula no tempo, e ao final os resultados podem ser completamente errados.

Pesquisadores de Inteligência Artificial se ocuparam do problema de planejar a trajetória de um robô, e seu mais famoso resultado é o planejador STRIPS. Nele, as ações possíveis de serem realizadas pelo robô são modeladas em termos de suas pré-condições, as condições que devem ocorrer para que o robô possa realizar esta ação, e as pós condições, que são os estados que acontecerão depois do robô ter realizado a ação. Através de busca neste espaço de operadores, uma seqüência de ações que levem a satisfazer um conjunto de metas é obtido. Existe uma dificuldade com este tipo de planejador primitivo, que é conhecido como a "Anomalia de Sussman". Isto ocorre quando, na busca da satisfação de um objetivo, um outro, previamente alcançado é destruído. Quando esta interação destrutiva entre objetivos ocorre, dizemos que os objetivos não são independentes.[10]

Planejadores que superam estas dificuldades constituíram o próximo passo das pesquisas. Hoje são conhecidos planejadores que tratam com este tipo de interação entre metas. Porém, como se poderia esperar, com este aumento de capacidade, a complexidade dos algoritmos também aumenta. Para tratar de um ambiente dinâmico, os navegadores baseados em sistemas de planejamento, planejam uma rota inicial, e ao encontrarem desvios neste plano, como a presença de obstáculos não antecipados, um novo plano deve ser gerado para tratar esta nova situação. Este plano deve ser gerado enquanto o robô já está andando, e este processo de remendar planos no meio do caminho é invocado muitas vezes no decorrer de uma missão. Isto requer muita computação, o que degrada a performance do robô.

Rodney Brooks, do MIT, propôs então um conceito revolucionário, onde seus robôs agiriam reativamente, sem qualquer planejamento. Este conceito de robôs reativos foram implementados em seus robôs insetos, dos quais *Genghis* e *Attila* são os mais famosos exemplos. Entretanto, esta proposta não foi bem sucedida mesmo nestes robôs mais simples, e muito menos pôde ser escalada para robôs mais complexos, como robôs móveis. Prova disso é a mudança radical de Brooks, agora pesquisando o andróide 'Cog'[9]. Mas a idéia de reatividade ainda foi aproveitada em parte.

Juntando as abordagens de planejamento e reatividade, temos o que se chama de controle baseado em comportamento (Behavior Based Control). Nesta abordagem, as ações são consideradas comportamentos, e os comportamentos são seguidos até que algum fato externo exija que sejam trocados. Por exemplo, o robô pode ter um comportamento de seguir uma fonte de luz. Se o robô se depara com um obstáculo, o comportamento de contornar obstáculo é invocado, e quando o obstáculo é contornado, o comportamento de seguir a fonte de luz é retomado. Esta é a técnica mais utilizada, e apresenta certo grau de sucesso na prática.

Neste trabalho, foi implementado o comportamento de desvio de obstáculo, que pode ser utilizado como um módulo de um sistema baseado em comportamentos, podendo ser invocado quando o robô percebe um obstáculo que não permite ir direto a seu destino.

4. Redes Neurais

Uma rede neural é um sistema inspirado no funcionamento do cérebro humano. Um grupo de unidades de processamento, representando os neurônios, chamados de neurônios de entrada, recebem informação de seu ambiente. Estes neurônios são conectados a outro grupo de neurônios, que constituem a camada oculta. Esta camada oculta consegue identificar padrões que ocorrem na camada de entrada, e por sua vez são conectados a uma outra camada, que pode ser outra camada oculta, ou a camada de saída, que é formada pelos neurônios de saída. Cada interconexão tem um valor associado a ela, conhecido por peso. Dependendo do valor dos pesos, diferentes padrões de entrada podem resultar no disparo de um ou mais neurônios de saída. Através de treinamento, os neurônios de saída são ensinados a dar a resposta correta dependendo do estímulo apresentado aos neurônios de entrada.

O cérebro humano, de onde os cientistas tiraram inspiração para a construção das redes neurais, apresenta características interessantes, como: Paralelismo massivo, representação e computação distribuídas, habilidade para aprender, habilidade de generalizar, adaptabilidade, processamento de informação contextual inerente, tolerância a falhas, e baixo consumo de energia.

Os modernos computadores, baseados na arquitetura de Von Neumman, são melhores que as pessoas em cálculos matemáticos e manipulação de símbolos relacionados. Porém, em determinadas tarefas, como por exemplo reconhecer um rosto familiar, as pessoas são muito melhores que os computadores. Espera-se, com a utilização das redes neurais, dotar o computador de algumas das características encontradas nestas redes neurais naturais - as pessoas.

Atualmente, redes neurais têm sido usadas para resolver as seguintes classes de problemas: classificação de padrões, categorização/agrupamento, aproximação de funções, previsão, otimização, memórias endereçadas pelo conteúdo, e controle.

As vantagens da utilização de uma rede neural para resolver um determinado problema são, entre outras, sua habilidade para aprender por si mesmas, ou seja, damos exemplos do que queremos que elas aprendam, e ela mesma se ajusta para aprender.

5. Rede Neural como Solução para o problema de evitar obstáculo

Para evitar obstáculos, enquanto o robô está andando num ambiente, muitos fatores devem ser levados em consideração, por exemplo: a distância do robô ao obstáculo, a forma do obstáculo, a orientação do obstáculo, o contexto do movimento, e para onde o robô estava se dirigindo antes de encontrar o obstáculo. Com tantos fatores a levar em consideração, não chaga a ser surpresa o fato de ser difícil obter um algoritmo que diga passo a passo quais são os movimentos corretos, para cada possível obstáculo, que esperamos que o robô realize.

Entretanto, podemos dar exemplos do que gostaríamos que fosse feito pelo robô, ao vermos uma situação. Assim, se pudermos ver o que seus sensores estão lendo em um dado instante, podemos dizer qual a melhor ação que o robô poderia fazer naquela situação.

É precisamente este tipo de informação, os chamados pares de entrada / saída que as redes neurais necessitam para processar. Ao final do treinamento da rede, ela deverá ser capaz de tomar as decisões que foram exemplificadas, e melhor ainda, poderá tomar

ações mesmo para situações que ela não tenha visto antes, que é a aplicação de sua capacidade de generalização.

Para o treinamento da rede, os pares de entrada foram formados pela leitura da matriz de sensores, de 9 x 9 elementos, mais a informação sobre para onde o robô estava se dirigindo ao encontrar o obstáculo (para cima, para baixo, esquerda ou direita). As direções foram definidas como quatro elementos, totalizando assim, 85 elementos de entrada. A saída foi definida como sendo a direção que e robô deve tomar ao encontrar o problema exemplificado. As posições são identificadas como: 1 0 0 0, para cima. 0 1 0 0 para baixo. 0 0 1 0 para a esquerda, e 0 0 0 1 para a direita.

6. Resultados

Para treinar a rede neural, foi utilizado o algoritmo Backpropagation, que é baseado na técnica de correção de erros. O simulador de redes neurais SNNS foi utilizado para implementar a rede.

A rede foi inicialmente definida como tendo 85 elementos de entrada, e 4 elementos de saída. Os elementos da camada intermediária não podem ser definidos *a priori*, porque não é conhecido um método para sua determinação. Por isso, testes com vários números de neurônios na camada oculta foram realizados. Veja na Tabela 1.

Outro parâmetro testado, foi a taxa de aprendizado, que determina quão rapidamente a rede neural consegue aprender os padrões a ela apresentados. Como no caso da camada oculta, nenhum método para sua determinação é conhecido, o que requer

a realização de testes para determinar a melhor taxa. Temos um resumo dos resultados obtidos na Tabela 2, para uma camada oculta.

Neurônios na Camada Oculta (N)	Menor Erro alcançado
5	15,244
10	11,279
15	11,217
20	15,629
30	14,538

Tabela 1: Exploração preliminar das possíveis configurações da camada intermediária

Taxa de aprendizado	N=10	N=13	N=15	N=17
0,02	12,659	13,246	11,099	13,222
0,04	11,279	10,856	13,325	13,055
0,06	11,677	14,108	8,8971	10,370
0,09	11,350	11,887	12,521	11,823
0,14	16,302	11,422	13,435	12,730
0,2	18,285	11,232	9,992	14,481

Tabela 2: Ajuste das melhores configurações encontradas

Obtivemos ao final, utilizando 15 neurônios em uma única camada oculta, e taxa de aprendizado de 0,06, um erro final de 8,9%.

7. Conclusão

A rede neural se mostrou uma alternativa viável para a criação de módulos de comportamento para a navegação de robô, principalmente devido a sua capacidade de generalização para situações não vistas anteriormente.

Outros métodos, como aprendizado simbólico, levam a uma grande especialização, degradando a performance do sistema para situações não encontradas anteriormente. Já a rede neural não sofre deste problema.

A derivação de algoritmos não consegue tratar de situações inesperadas, entrando em colapso. A rede neural, entretanto apresenta degradação suave, ou seja, ainda sob circunstâncias inesperadas, fornece respostas razoáveis, baseadas em sua experiência prévia. Finalmente, o tempo de treinamento da rede neural é mais rápido do que o tempo para se derivar um algoritmo.

A continuação desta pesquisa é implementar novos comportamentos a serem utilizados em um sistema de navegação baseado em comportamentos, tais como comportamento exploratório e comportamento de seguir, para verificar se a abordagem de redes neurais se mostra eficiente nestas tarefas também. Outra extensão que será feita é colocar o sistema para ser utilizado em um robô real.

8. Bibliografia

[1] Groover, M., Weiss, M., Nagel, R., Odrey, N., Robótica - Tecnologia e Programação - São Paulo, McGraw-Hill, 1989.

[2] Biewald, R., A neural network controller for the navigation and obstacle avoidance of a mobile robot, in Neural Networks for Robotic Control - Theory and Applications (1996) Zalzala, A.M.S., Morris, A.S. (eds), Ellis Horwood.

[3] Moreno, L., Salichs, M.A ., Gachet, D., Pimentel, J., Arroyo, F., Gonzalo, A ., Neural Networks for mobile robot piloting control, in Neural Networks for Robotic Control - Theory and Applications (1996) Zalzala, A.M.S., Morris, A.S. (eds), Ellis Horwood.

[4] Liscano, R., Manz, A ., Stuck, E., Fayek, R. E., Tigli, J., Using a blackboard to integrate multiple activities and achieve strategic reasoning for mobile robot navigation, IEEE Expert, April, 1995, pp. 24-36.

[5] Kaiser, M., Klingspor, V., Millán, J., Accame, M., Walner, F., Dillman, R., Using machine learning techiques in real world mobile robots, IEEE Expert, April, 1995, pp. 37-45.

[6] Jain, A. K., Mao, J., Artificial neural networks: a tutorial, IEEE Computer, 29:3, pp. 31-44.

[7] Pal, S. K., Srimani, P. K., Neurocomputing: motivations, models anda hybridization, IEEE Computer, 1995, 29:3, pp. 24-28.

[8] Zeichick, A. L., Neural networks: a brief history in time, Neural Networks Special Report, AI Expert, 1992, pp. 5.

[9] Brooks, R. A., From Eawings to Humans, to appear in Robotics and Autonomous Systems.

[10] Nilson, N. J., Principles of Artificial Intelligence, 1980, Tioga: Palo Alto, Ca.