

Estacionamento de um Veículo de Forma Autônoma Utilizando Redes Neurais Artificiais

Milton Heinen², Fernando Osório², Farlei Heinen³ e Christian Kelber³

¹Universidade do Vale do Rio dos Sinos (UNISINOS)

²Mestrado em Computação Aplicada, ³Engenharia da Computação
CEP 93022-000 - São Leopoldo - RS - Brasil

mheinen@turing.unisinos.br, {fosorio, farlei, kelber}@unisinos.br

Resumo. Este artigo tem por objetivo apresentar o sistema SEVA3D, que é capaz de simular de modo realista o estacionamento de um veículo em uma vaga paralela. O sistema proposto permite controlar um carro através da leitura de um conjunto de sensores do tipo sonar, gerando de modo autônomo os comandos de aceleração e de giro de direção, de modo a estacionar o carro na vaga. O sistema conta com um controlador implementado através de uma rede neural do tipo Jordan, e os resultados obtidos demonstram que o controlador é capaz de estacionar corretamente o carro baseado apenas nas informações provenientes dos sensores.

1. Introdução

Os veículos autônomos (RMA - Robôs Móveis Autônomos) tem atraído a atenção de um grande número de pesquisadores, devido ao desafio que este novo domínio de pesquisas propõe: dotar sistemas de uma capacidade de raciocínio inteligente e de interação com o meio em que estão inseridos [Bekey 2005]. Os RMAs podem perceber o ambiente em que estão inseridos através da leitura de seus sensores (infravermelho, Sonar, lasers, câmeras de vídeo, etc), e através desta percepção sensorial eles podem planejar melhor as suas ações [Medeiros 1998, Heinen 1999].

Atualmente os robôs móveis atuam em diferentes áreas, como desarmamento de bombas, exploração de ambientes hostis, e a condução de veículos robotizados. Alguns exemplos de RMA's são: o sistema desenvolvido pelo NavLab da CMU [Pomerleau 1990, Batavia et al. 1996] que é capaz de conduzir uma caminhonete pelas estradas americanas; os robôs do tipo *rover* enviados para Marte pela NASA [Stone 1996]; o robô Dante, que explora o interior de vulcões [Lemonick 1994]; e o sistema de controle de um veículo Li-gier elétrico desenvolvido pelos pesquisadores do INRIA na França [Garnier et al. 1999].

A partir dos estudos e dos trabalhos de pesquisa desenvolvidos pelo Grupo de Inteligência Artificial (GIA-PIPCA¹) e pelo Grupo de Pesquisas em Veículos Autônomos (GPVA²) da Unisinos, foram criadas as bases para o desenvolvimento de aplicações na área de robótica autônoma móvel. Destaca-se particularmente o desenvolvimento do sistema SEVA3D (Simulador de Estacionamento de Veículos Autônomos em um ambiente tridimensional), que realiza o estacionamento de um veículo não-holonômico simulado em uma vaga paralela de forma autônoma.

¹GIA-PIPCA – <http://inf.unisinos.br/~osorio/gia.html>

²GPVA – <http://www.exatec.unisinos.br/~autonom/>

O sistema SEVA3D³ [Heinen et al. 2006b] original (SEVA3D-A) utiliza um modelo tridimensional do ambiente, sensores do tipo sonar, e realiza o controle do veículo através de um autômato finito baseado em regras. A tarefa de estacionamento com o SEVA3D-A funcionou de modo satisfatório, mas as regras do autômato tiveram que ser codificadas manualmente, o que é uma tarefa bastante árdua, e o autômato resultante se mostrou pouco robusto a situações não esperadas [Heinen and Osório 2005].

Por estes motivos, foi decidido o desenvolvimento uma nova versão do sistema, chamada de SEVA3D-N [Heinen et al. 2006a] onde ao invés de um autômato finito, o controle do veículo é realizado por uma rede neural do tipo Jordan [Braga et al. 2000]. A vantagem desta abordagem é que as regras não precisam ser codificadas manualmente, pois a rede neural pode aprender a partir do exemplo de uma manobra de estacionamento.

2. Trabalhos relacionados

Estudos relativos ao estacionamento de veículos de forma autônoma vem sendo realizados por diversos grupos de pesquisa, dentre os quais é possível destacar os estudos realizados pelo INRIA [Paromtchik and Laugier 1996, Laugier et al. 1998, Garnier et al. 1999] para o controle e estacionamento de um veículo Ligier elétrico de forma autônoma. O veículo em questão foi equipado com quatorze sensores do tipo sonar, como mostra a Figura 1. Para que o cálculo da profundidade da vaga fosse possível, foi necessária a instalação

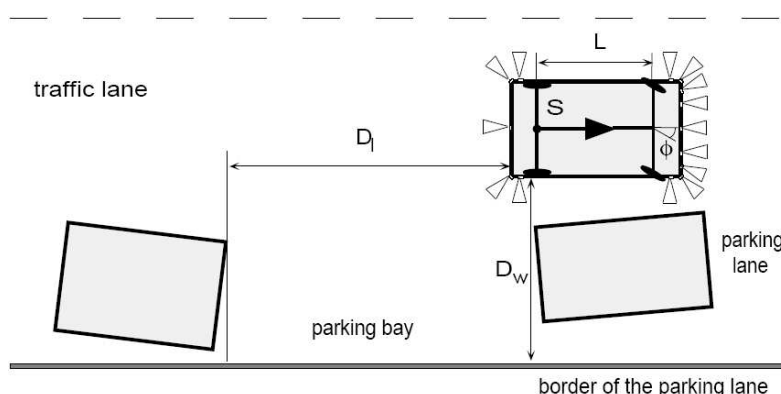


Figura 1. Modelo do INRIA [Laugier et al. 1998]

de uma barreira de altura moderada junto ao meio-fio, para que os sensores pudessem detectá-lo com precisão [Paromtchik and Laugier 1996].

As principais desvantagens da abordagem descrita acima são: (i) a necessidade da instalação de uma barreira, o que dificulta o estacionamento em vias convencionais; e (ii) a necessidade da utilização de muitos sensores (quatorze) do tipo sonar. No SEVA3D, foram utilizados apenas seis sensores. Outra desvantagem do modelo do INRIA é que a tarefa de estacionamento teve de ser codificada manualmente. Já no SEVA3D-N, a tarefa de estacionamento pode ser aprendida automaticamente pela rede neural a partir de exemplos.

2.1. Redes neurais artificiais

Através de um modelo abstrato e simplificado dos neurônios humanos é possível desenvolver um simulador que seja capaz de classificar, generalizar e aprender funções des-

³SEVA3D – <http://inf.unisinos.br/~osorio/seva3d/>

conhecidas. Um dos modelos de aprendizado neural mais utilizados na atualidade é o modelo denominado Backpropagation [Rumelhart et al. 1986]. Para que ocorra o aprendizado, é utilizado um conjunto de dados de exemplos de padrões com as respostas esperadas (padrões e classes correspondentes), que é dividido em uma base de aprendizado e uma base de validação (avaliação da generalização). Este tipo de aprendizado é conhecido como aprendizado supervisionado com validação cruzada [Haykin 1999].

Para a implementação do SEVA3D-N, foi utilizado o simulador de redes neurais *Stuttgart Neural Network Simulator - SNNS*⁴, que é um dos simuladores de redes neurais mais completos e eficientes, e possui diversas ferramentas adicionais que permitem a geração de *scripts*, a geração de gráficos e a análise dos resultados obtidos. O algoritmo de aprendizado utilizado no SEVA3D-N foi o RPROP [Riedmiller and Braun 1993], que é um aperfeiçoamento do algoritmo de aprendizado do Backpropagation original.

3. Simulador SEVA3D-N

Os principais componentes do simulador SEVA3D-N [Heinen et al. 2006a] são:

- Simulador do ambiente virtual;
- Modelo sensorial;
- Modelo cinemático (deslocamento do veículo);
- Comandos relacionados ao deslocamento (avançar, recuar e velocidade) e ao giro do veículo (rotação da direção);
- Sistema de controle do veículo durante o estacionamento (rede neural).

3.1. Ambiente virtual

Para a implementação do ambiente virtual do SEVA3D, foi utilizado o simulador SimRob3D [Heinen 2002, Heinen and Osório 2002], desenvolvido pelo Grupo de Inteligên-



Figura 2. Ambiente modelado

cia Artificial (GIA-PIPCA) e pelo Grupo de Pesquisas em Veículos Autônomos (GPVA) da Unisinos. Este simulador tem como principal característica o fato de utilizar um ambiente tridimensional para a navegação dos robôs móveis simulados. O ambiente pode ser modelado em diversos softwares de modelagem tridimensional existentes no mercado, e

⁴SNNS – <http://www-ra.informatik.uni-tuebingen.de/SNNS/>

permite que sejam especificados os diversos elementos presentes no ambiente (objetos, luzes, texturas), o que resulta em um nível de realismo muito superior aos obtidos em simuladores bidimensionais.

O simulador SimRob3D possui diferentes modelos sensoriais e cinemáticos, permitindo a configuração de diversos tipos de robôs. A Figura 2 mostra uma imagem do ambiente virtual utilizado durante a manobra de estacionamento. O simulador permite que a posição da câmera seja alterada, o que permite que a manobra de estacionamento possa ser assistida de qualquer ângulo.

3.2. Modelo sensorial

Os sensores de distância simulam sensores do tipo sonar [Bekey 2005], que são capazes de determinar a distância entre o veículo e os obstáculos presentes no ambiente: outros carros e a calçada. Os seis sensores utilizados estão distribuídos em pontos estratégicos do carro, como mostra a Figura 3. Foram implementados apenas os sensores da lateral



Figura 3. Localização dos sensores

direita do veículo, pois o modelo se restringiu ao estacionamento em vagas paralelas localizadas no lado direito do carro, caso típico em pistas de duas vias. Os sensores **V[2]** e **V[3]** precisaram ser instalados com uma certa inclinação em relação ao solo, para que fosse possível detectar o meio-fio. O SEVA3D permite que seja utilizada qualquer configuração (localização e quantidade) de sensores, mas nos experimentos descritos neste artigo apenas a configuração da Figura 3 foi utilizada.

Os sonares são simulados através da definição de uma seção cônica no espaço virtual (simulando o cone no qual as ondas sonoras se propagam em um sonar real), onde os objetos que estiverem dentro do volume formado pelo cone são detectados. A intersecção entre os objetos e o cone do sonar (espaço de percepção) é detectada através de uma abordagem estocástica. Diversas linhas de detecção de objetos (raios) são gerados a partir da posição do sensor e direcionados de acordo com a orientação espacial do sonar, permanecendo dentro do volume formado pelo cone. Uma técnica de RayCast⁵ foi utilizada para traçar os raios, que são distribuídos aleatoriamente no volume formado pelo cone do sonar. Se algum dos raios colidir com algum dos objetos presentes no ambiente, a distância do sensor até o ponto de colisão é informada.

Além dos sensores do tipo sonar, foi utilizado um sensor do tipo odômetro, que serve somente para verificar se o tamanho da vaga do estacionamento é grande o suficiente para permitir a manobra. O uso do odômetro torna o estacionamento possível de ser

⁵RayCast é uma técnica da computação gráfica que simula os efeitos físicos associados com a propagação de raios de luz [Foley 1994]

realizado independente da posição relativa dos outros carros. De fato, mesmo se não houverem outros carros estacionados para servir como pontos de referência, o estacionamento é possível graças ao uso do odômetro.

3.3. Modelo cinemático

A movimentação do veículo respeita o modelo de cinemática Ackerman, no qual um veículo é representado por um volume retangular suportado por quatro rodas, onde as rodas

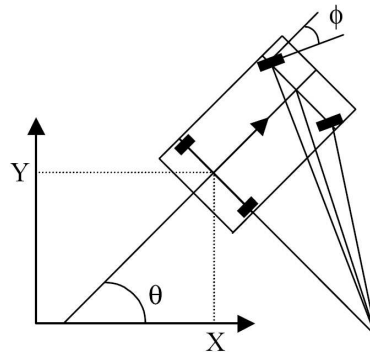


Figura 4. Modelo cinemático

traseiras possuem um eixo fixo e as rodas dianteiras podem ser direcionadas, através do giro da barra da direção (Figura 4) [Dudek and Jenkin 2000]. As coordenadas do veículo são definidas por (x, y, θ) , onde x e y definem o ponto médio do eixo traseiro do veículo (ponto 0) e θ indica a sua orientação (ângulo em relação à direção de referência). O deslocamento do veículo é descrito pelas equações [Paromtchik and Laugier 1996]:

$$\begin{cases} \dot{x} = v \cos \phi \cos \theta, \\ \dot{y} = v \cos \phi \sin \theta, \\ \dot{\theta} = \frac{v}{L} \sin \phi, \end{cases} \quad (1)$$

onde v representa a velocidade longitudinal do veículo, ϕ representa o giro da direção, e L indica o comprimento do eixo das rodas.

3.4. Autômato finito

No SEVA3D-N, a tarefa de estacionamento é realizada por uma rede neural que implementa um autômato finito. A Figura 5 mostra o diagrama de estados do autômato finito aprendido pela rede neural. Foram utilizados os seguintes estados:

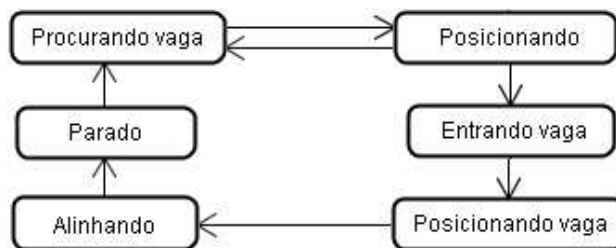


Figura 5. Estados do autômato

Parado: Estado inicial do autômato, antes do veículo começar a se deslocar;

Procurando vaga: Início da manobra de estacionamento, quando o veículo se desloca para frente com a direção reta, buscando encontrar uma vaga. Se encontrar, o estado muda para *Posicionando*;

Posicionando: O veículo se desloca para a frente, a fim de atingir uma posição que torne possível o estacionamento, e também para verificar se a vaga é suficientemente grande. Se não for, o estado volta para *Procurando vaga*. Se a vaga for de tamanho suficiente, o estado muda para *Entrando vaga*;

Entrando vaga: O carro vira a direção para a direita e começa a se movimentar para trás, de forma a entrar na vaga. Quando o sensor **V[2]** (Figura 3) detectar o meio-fio da calçada, o estado muda para *Posicionando vaga*.

Posicionando vaga: neste estado, o veículo prossegue para trás, mas com a direção voltada para à esquerda. Quando o sensor **V[3]** detectar o meio-fio da calçada, o estado muda para *Alinhando*;

Alinhando: neste estado o veículo é movimentado de forma a ficar a uma distância razoável em relação aos carros estacionados à frente ou atrás. Após o alinhamento estar concluído, o estado muda para *Parado* e a manobra é encerrada com sucesso.

3.5. Controle neural

Os resultados obtidos com o controle baseado no autômato do SEVA3D-A levaram ao estudo de uma alternativa para a aquisição de conhecimentos, visto que o processo de codificação de regras é muito trabalhoso, além de não garantir uma grande robustez do sistema na presença de ruídos (e.g. valores inexatos dos sensores e situações não previstas). Buscou-se então a implementação de uma solução baseada no “aprendizado prático” a partir de uma base de exemplos de condução do veículo autônomo. A solução adotada foi a utilização de redes neurais artificiais (RNA) [Braga et al. 2000, Rumelhart et al. 1986, Haykin 1999], com aprendizado supervisionado, capazes de aprender bases de exemplos que demonstram como se deve estacionar o carro.

O primeiro passo foi a construção das bases de exemplos empregadas no aprendizado da rede neural. Adaptou-se o simulador de modo a gerar um “arquivo de log”, contendo o registro do estado dos sensores, estado do autômato e comandos enviados aos atuadores (velocidade e rotação), gerados pelo SEVA3D-A. O arquivo gerado permite que seja treinada uma rede neural, que inicialmente deve ser capaz de reproduzir o comportamento do autômato. Posteriormente foi utilizado o SEVA3D para gerar novos exemplos da tarefa de estacionamento, com o veículo sendo controlado por um ser humano.

O modelo de RNA adotado foi o modelo de Jordan [Ludik et al. 1997], baseado em redes do tipo MLP (Multi-Layer Perceptron) [Braga et al. 2000], e o algoritmo de aprendizado foi o Resilient Propagation [Riedmiller and Braun 1993]. O princípio de funcionamento da rede neural assim definida é o uso de um conjunto de entradas que servem para indicar o estado atual da rede (compatível com o autômato) e um outro conjunto de saídas da rede que servem para indicar o estado que a rede deve assumir (novo estado do autômato) em função de suas entradas. Deste modo a rede neural é capaz de receber como entrada o seu estado atual e o estado dos sensores (Figura 6) e determinar em sua saída se deve manter o estado atual ou passar para um novo estado, baseada nas informações que recebeu.

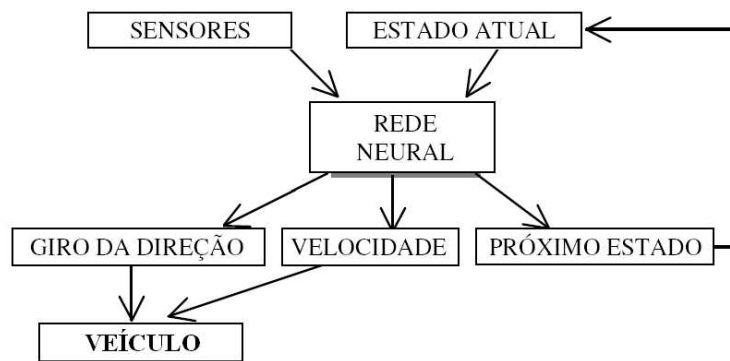


Figura 6. Esquema da rede neural utilizada

O próximo estado (atual ou novo), obtido na saída da rede é então re-injetado na entrada da rede, obtendo assim um modelo similar ao proposto por Jordan, onde a saída da rede é re-injetada na entrada formando uma recorrência, e obtendo assim as chamadas unidades de contexto [Ludik et al. 1997]. É através das unidades de contexto que temos uma informação sobre o estado atual do autômato/rede.

As variáveis (atributos) de entrada da rede empregados foram: o estado dos seis sensores de proximidade, o odômetro e uma indicação do estado atual do processo de estacionamento. Na saída da rede é obtido o estado dos atuadores (velocidade e rotação), assim como uma indicação do próximo estado do processo de estacionamento. Deste modo, a rede irá controlar os atuadores, mas também irá decidir quando se deve passar de um estado a outro no autômato de controle. A Figura 6 apresenta o esquema das entradas e saídas da RNA adotada.

A base de aprendizado utilizada para treinar a rede neural usou uma codificação binária do tipo “1 entre N” para representar a maioria das entradas e saídas da rede:

- Entrada: o estado atual (6 estados = 6 entradas binárias, com 1 bit representando cada estado); o estado dos sensores (6 sensores = 6 entradas numéricas normalizadas entre 0 e 1); odômetro (numérico);
- Saída: a velocidade (5 velocidades = 5 saídas binárias: frente rápida, frente lenta, parado, ré lenta e ré rápida), o giro da direção (5 rotações possíveis = 5 saídas binárias: giro máximo à esquerda, giro médio à esquerda, direção reta, giro médio à direita e giro máximo à direita) e o próximo estado (que também codifica 5 estados possíveis em 5 saídas binárias).

4. Implementação

Para a implementação do modelo, foi utilizado o simulador SimRob3D [Heinen 2002], e foi construído um modelo de tridimensional de uma via com carros estacionados, para que fosse possível realizar o estacionamento de forma adequada (Figura 2). O veículo modelado para realizar o estacionamento é uma reprodução de um veículo real do tipo Mini-Baja Buggy, disponível para pesquisas em nossos laboratórios. Este veículo foi desenvolvido pelo Grupo de Pesquisas em Veículos Autônomos (GPVA) da Unisinos [Kelber et al. 2005].

O veículo real foi automatizado e pode ser controlado por dispositivos remotos, como um telefone celular. A Figura 7(a) mostra o veículo Mini-Baja Buggy real, e a Fi-

gura 7(b) mostra o modelo virtual utilizado pelo SEVA3D. Atualmente o veículo real está sendo equipado com os dispositivos necessários para realizar a tarefa de estacionamento (sensores do tipo sonar), para que em breve venha a ser utilizado para validar o modelo de controle do SEVA3D.



(a) Veículo real

(b) Veículo simulado

Figura 7. Veículo Mini-Baja automatizado

O simulador SEVA3D realiza uma integração discreta das fórmulas da Equação 1, de forma que a simulação realizada seja bastante próxima à realidade. Para acompanhar a manobra de estacionamento, o usuário pode visualizar o ambiente com a câmera virtual posicionada em qualquer ponto. A implementação do SEVA3D foi validada através de diversos testes preliminares, e nestes testes foi verificado que o modelo simulado se comporta de forma similar à realidade.

5. Resultados

Os resultados das simulações do aprendizado da RNA, para um conjunto de 10 simulações realizadas, foram os seguintes: foi obtida uma taxa de aprendizado com um acerto médio de 96.83% nas respostas da rede; foram utilizados três neurônios na camada oculta e o número médio de épocas de aprendizado foi 296 (melhor época de generalização). Os exemplos não aprendidos corretamente não comprometem a capacidade da rede em trocar de estado (momento crítico no controle do veículo), de forma que em todas as dez simulações o controlador conseguiu estacionar corretamente o veículo.

A distância final do carro em relação ao meio-fio (após o estacionamento) foi em média 21.03cm, e o desvio padrão de 5.92cm, o que demonstra que o SEVA3D se comportou de forma robusta no estacionamento do veículo. A Figura 8 mostra o exemplo de uma manobra de estacionamento⁶.

6. Conclusões e perspectivas

Este artigo teve como objetivo descrever o sistema SEVA3D, que é um simulador utilizado para o controle de veículos autônomos durante o estacionamento em vagas paralelas, utilizando um ambiente tridimensional realista e sensores do tipo Sonar. Os resultados obtidos nas simulações demonstraram que o sistema de controle possui a capacidade de controlar

⁶Alguns vídeos do estacionamento estão presentes em <http://inf.unisinos.br/~osorio/seva3d/>

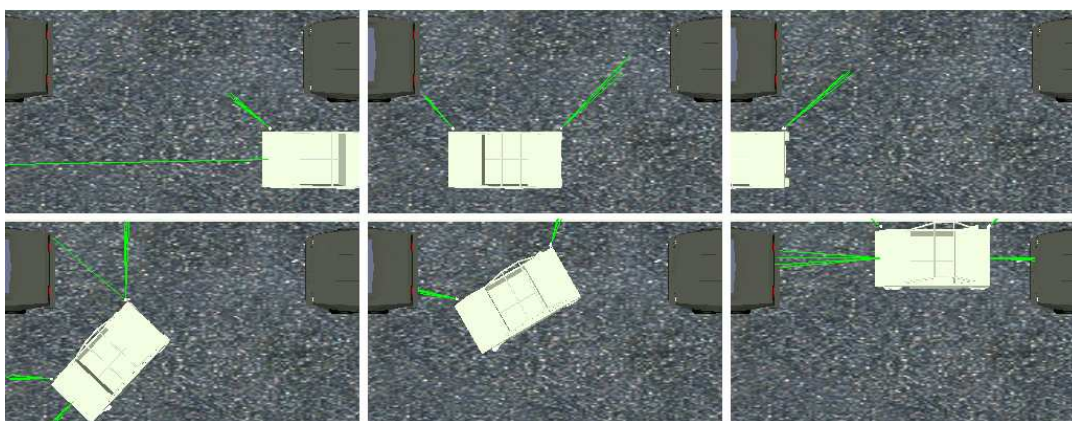


Figura 8. Exemplo de uma manobra de estacionamento

corretamente o veículo, cumprindo seu objetivo principal: estacionar o veículo de modo autônomo corretamente na vaga, sem bater nos demais elementos presentes no ambiente. As verificações, tanto numéricas quanto visuais, permitiram constatar que a tarefa pôde ser corretamente executada na maioria das simulações realizadas, demonstrando que o sistema de controle do SEVA3D é bastante estável.

Atualmente o veículo real que serviu de modelo para o veículo virtual está sendo equipado com os sensores do tipo sonar, e em breve ele será para validar o sistema de controle do SEVA3D.

Referências

- Batavia, P., Pomerleau, D., and Thorpe, C. (1996). Applying advanced learning algorithms to alvinn. Technical Report CMU-RI-TR-96-31, Carnegie Mellon Univ. (CMU), Pittsburgh, PA.
- Bekey, G. A. (2005). *Autonomous Robots: From Biological Inspiration to Implementation and Control*. MIT Press, Cambridge, MA.
- Braga, A. d. P., Ludermir, T. B., and Carvalho, A. C. P. d. L. F. (2000). *Redes Neurais Artificiais - Teoria e Aplicações*. LTC Editora, Rio de Janeiro, Brazil.
- Dudek, G. and Jenkin, M. (2000). *Computational Principles of Mobile Robotics*. Cambridge Univ. Press, Cambridge, UK.
- Foley, J. D. (1994). *Introduction to Computer Graphics*. Addison-Wesley, xxviii.
- Garnier, P., Fraichard, T., Laugier, C., Paromtchik, I., and Scheuer, A. (1999). Motion autonomy through sensor-guided manoeuvres. In *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, Kyongju, Korea.
- Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation*. Prentice-Hall, Upper Saddle River, NJ, 2 edition.
- Heinen, F. J. (1999). *Robótica Autônoma: Integração entre Planificação e Comportamento Reativo*. UNISINOS Editora, São Leopoldo, RS, Brazil.
- Heinen, F. J. (2002). Sistema de controle híbrido para robôs móveis autônomos. Master's thesis - applied computing, Universidade do Vale do Rio dos Sinos (UNISINOS), São Leopoldo, RS, Brazil.

- Heinen, F. J. and Osório, F. S. (2002). HyCAR - a robust hybrid control architecture for autonomous robots. In *Proc. Hybrid Intelligent Systems (HIS)*, volume 87, pages 830–840, Santiago, Chile. IOS Press.
- Heinen, M. R. and Osório, F. S. (2005). Estacionamento de um veículo de forma autônoma simulado em um ambiente tridimensional realístico. In *Anais do Semimco*, pages 56–65, Blumenau, SC, Brazil.
- Heinen, M. R., Osório, F. S., Heinen, F. J., and Kelber, C. (2006a). SEVA3D: Using artificial neural networks to autonomous vehicle parking control. In *Proc. IEEE Int. Joint Conf. Neural Networks (IJCNN)*, to appear, Vancouver, Canada.
- Heinen, M. R., Osório, F. S., Heinen, F. J., and Kelber, C. (2006b). Uso de realidade virtual no desenvolvimento de um sistema de controle do estacionamento de veículos autônomos. In *Proc. of VIII Symposium on Virtual Reality*, pages 245–256, Belém, PA, Brazil. Editora CESUPA.
- Kelber, C., Jung, C. R., Osório, F. S., and Heinen, F. J. (2005). Electrical drives in intelligent vehicles: Basis for active driver assistance systems. In *Proc. IEEE Int. Symposium on Industrial Electronics (ISIE)*, volume 4, pages 1623–1628, Dubrovnik, Croatia.
- Laugier, C., Fraichard, T., Paromtchik, I. E., and Garnier, P. (1998). Sensor based control architecture for a car-like vehicle. In *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, pages 165–185, Victoria, Canada.
- Lemonick, M. (1994). Dante tours the inferno. *Time Magazine - Time Domestic/Science*, 144(7).
- Ludik, J., Prins, W., Meert, K., and Catfolis, T. (1997). A comparative study of fully and partially recurrent networks. In *Proc. IEEE Int. Conf. Neural Networks (ICNN)*, volume 1, pages 292–297, Houston, TX.
- Medeiros, A. (1998). Introdução à robótica. In *Anais do XVII Encontro Nacional de Automática*, volume 1, pages 56–65, Natal, RN, Brazil.
- Paromtchik, I. E. and Laugier, C. (1996). Autonomous parallel parking of a nonholonomic vehicle. In *Proc. IEEE Int. Symposium on Intelligent Vehicles (IV)*, pages 13–18, Tokyo, Japan.
- Pomerleau, D. (1990). Neural network based autonomous navigation. *Vision and Navigation - The CMU Navlab*.
- Riedmiller, M. and Braun, H. (1993). A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In *Proc. IEEE Int. Conf. Neural Networks (ICNN)*, pages 586–591, San Francisco, CA.
- Rumelhart, D., Hinton, G., and Williams, R. (1986). *Learning Internal Representations by Error Propagation*. MIT Press, Cambridge, MA.
- Stone, H. W. (1996). Mars pathfinder microrover - a small, low-cost, low-power spacecraft. In *Proc. AIAA Forum on Advanced Developments in Space Robotics*, Madison, WI.