

# Controle inteligente de um robô móvel utilizando imagens

Leandro T. Hoffmann<sup>1</sup>, Ana Paula A. Castro<sup>1</sup>, José Demísio S. da Silva<sup>1</sup>

<sup>1</sup>Instituto Nacional de Pesquisas Espaciais – INPE  
Laboratório Associado de Computação e Matemática Aplicada  
Av. dos Astronautas, 1758 – 12227-010 – São José dos Campos, SP.

{hoffmann, apaula, demisio}@lac.inpe.br

**Resumo.** *Este trabalho descreve três sistemas de navegação autônoma diferentes, baseados em imagens, que foram desenvolvidos utilizando-se lógica nebulosa e redes neurais artificiais. Afim de validar os sistemas de navegação, foram realizados experimentos com um robô móvel que navega, equipado com uma câmera, sobre uma pista. A câmera provê imagens com informações da estrada, utilizadas para guiar o veículo em direção e velocidade a ser tomada em cada ponto da pista. O desempenho dos sistemas de navegação neurais são comparados a um controlador nebuloso desenvolvido neste trabalho.*

**Abstract.** *This paper describes three different image based autonomous navigation systems that uses fuzzy logic and artificial neural networks based control. To validate the navigation systems, experiments were conducted using a mobile robot, which navigates on a road with an onboard camera. The camera provides images with road information used to guide the vehicle as to the direction and speed to follow from each track position. The performances of different neural network based navigation systems are compared to a fuzzy logic based controller.*

## 1. Introdução

Navegação autônoma é um tema muito estudado e pesquisado na área de Inteligência Artificial [Kosaka and Pan, 1995] [Li et al., 1997] [Li et al., 2003] [Meng and Kak, 1993], da qual diferentes paradigmas vêm sendo utilizados para abordar o problema usando diferentes sensores para recuperar a informação do ambiente, para uma navegação segura e eficiente. A navegação autônoma está relacionada à habilidade de um veículo mover-se sem intervenção humana até alcançar o seu objetivo, em um ambiente no qual nenhuma informação, em princípio, está disponível. Várias arquiteturas diferentes para navegação autônoma têm sido propostas e podem ser categorizadas como: hierárquico, com um alto nível (considera modelos e planos) e baixo nível (considera detecções e execuções de ação); arquiteturas baseadas em comportamento, no qual as seqüências complexas das ações são encontradas por várias combinações de unidades simples; e arquiteturas híbridas que combinam duas ou mais técnicas [Heinen, 2001].

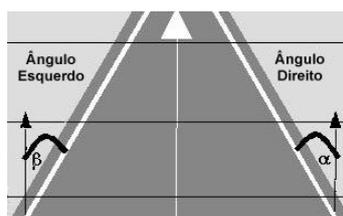
Em geral, os sistemas de navegação utilizam um conjunto de sensores para extrair informações do seu ambiente e então tomar decisões. Neste trabalho, utiliza-se operadores de visão computacional para efetuar a coleta de informações do ambiente. Vários trabalhos na literatura mostram o uso de imagens para navegação autônoma [Castro et al., 2001] [Li et al., 1997] [Sargent et al., 1997] [Shah and Aggarwal, 1996] [Talluri and Aggarwal, 1993]. Diferentes técnicas de visão computacional podem ainda ser usadas para recuperar informações da cena 3D de imagens 2D, como por exemplo, a abordagem em [Silva and Simoni, 2002].

Esse artigo apresenta o desenvolvimento de um modelo computacional de um sistema de navegação autônoma adaptável baseado em imagens usando redes neurais artificiais, como continuidade do trabalho iniciado em Castro et al (2001). Um robô move-se pelo ambiente no qual as imagens fornecem as informações necessárias para navegação. Um sistema de redes neurais determina a direção a ser seguida e a velocidade a ser empregada com base nas informações de pista, automaticamente se adaptando ao ambiente e corrigindo sua trajetória. O sistema proposto usa uma abordagem baseada em sensor para navegação autônoma, e apresenta um comportamento análogo a um humano ao navegar em um ambiente. Os experimentos relatados nesse artigo foram realizados com um veículo controlado remotamente a partir de um módulo de controle adicional adaptado ao controle remoto. As características da pista são extraídas por um operador de visão computacional e transformadas em informação de direção da pista que são apresentadas ao sistema de navegação. Neste trabalho, três sistemas de navegação foram desenvolvidos, um baseado em lógica nebulosa e os outros dois em redes neurais artificiais.

Este trabalho está organizado da seguinte maneira: Seção 2 descreve um operador de detecção de bordas baseado em autômato; Seção 3 apresenta um sistema de navegação por inferência em lógica nebulosa; Seção 4 relata dois sistemas de navegação que utilizam redes neurais artificiais; Seção 5 descreve o processo de envio de comandos dos sistemas de navegação ao robô autônomo, onde utilizou-se uma abordagem heurística e outra com redes neurais; Seção 6 traz alguns resultados de experimentos realizados; Seção 7 conclui o trabalho e discute trabalhos futuros.

## 2. Operador visual baseado em autômato

As faixas da pista fornecem informações a serem utilizadas pelo sistema de navegação do robô para controlar sua trajetória e velocidade. As direções das faixas são computadas considerando um sistema de coordenada Cartesiana, onde assume-se que o centro de cada imagem capturada é perpendicular ao plano de visão da câmera (campo de visada do motorista). Essa modelagem da pista é ilustrada pela Figura 1, onde a linha branca no centro da imagem representa um eixo de referência virtual.



**Figura 1: Ângulos esquerdo e direito das faixas da pista. A seta central representa um eixo de referência virtual.**

Em [Castro et al., 2001] a detecção da direção angular das faixas foi realizada por um operador de gradiente, a partir de algumas linhas selecionadas na imagem. Neste trabalho, o operador de gradiente foi substituído por um Algoritmo Detector de Faixas (ADF) baseado em autômato finito (AF), que analisa os níveis de cinza (NC) dos elementos da imagens (*pixels*).

A Figura 2 mostra um diagrama de blocos com a seqüência de procedimento para se usar o operador baseado em AF para computar os ângulos das faixas da pista. Um autômato finito consiste num conjunto finito de estados  $Q$ , uma função  $\delta$  de transição de estados (a qual depende do estado atual do autômato) e de símbolos selecionados a partir de um alfabeto finito  $\Sigma$ . O estado inicial do autômato é representado por  $q_0$  e  $F$  é o conjunto de estados finais [Hopcroft and Ullman, 1979]. No ADF descrito neste

artigo, o conjunto de estados do AF é  $Q = 0, 1, 2, 3, 4, 5, 6$ ; o alfabeto é o conjunto  $\Sigma = claro, naoClaro, escuro, naoEscuro, fim$ ; o estado inicial é  $q_0 = 0$ ; o conjunto de estados finais é  $F = 4, 5, 6$ ; e a função de transição de estados é definida pela Tabela 1.

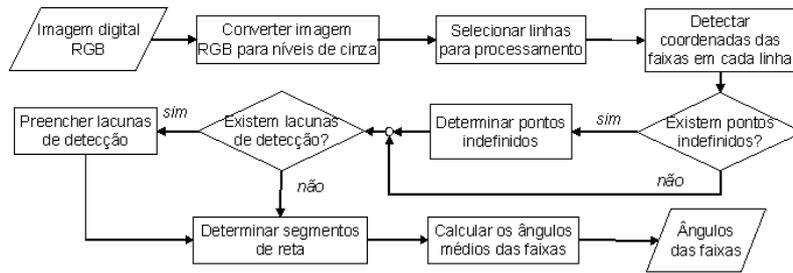


Figura 2: Diagrama de blocos do sistema de processamento de imagens.

Tabela 1: Função de Transição de Estados.

símbolo atual	símbolo lido				
	claro	nãoClaro	escuro	nãoEscuro	fim
0	1	0	-	-	4
1	-	-	2	1	5
2	3	2	-	-	5
3	-	-	6	3	6

O ADF seleciona símbolos  $sb$  de  $\Sigma$  para alimentar o AF. É feita uma conversão das características extraídas de cada *pixel* para símbolos que são imediatamente interpretados pelo AF. Quando o ADF processa o último *pixel* numa dada linha, ele insere o símbolo *fim* em  $sb$ . Caso contrário, o mapeamento de símbolos é feito baseado no NC do *pixel* atual e no estado corrente do AF, como descrito pela Tabela 2. Dois limiares relativos às condições de iluminação (*limiarSuperior* e *limiarInferior*) devem ser ainda definidos como parâmetro de simulação.

Tabela 2: Relação do mapeamento dos símbolos do autômato finito a partir da extração de características da imagem.

FA currentstate	Pixel Gray Level Feature	$\Sigma$ symbol chosen
0	$NC \leq \text{limiarSuperior}$	nãoClaro
0	$NC > \text{limiarSuperior}$	claro
1	$NC \leq \text{limiarInferior}$	escuro
1	$NC > \text{limiarInferior}$	nãoEscuro
2	$NC \leq \text{limiarSuperior}$	nãoClaro
2	$NC > \text{limiarSuperior}$	claro
3	$NC \leq \text{limiarInferior}$	escuro
3	$NC > \text{limiarInferior}$	nãoEscuro

Quando ambas as faixas da pista foram detectadas, numa linha da imagem, fica trivial identificar qual coordenada pertence a faixa da esquerda e qual pertence a faixa da direita.

Todavia, quando apenas uma coordenada é encontrada, o módulo de *identificação de pontos indefinidos*, na Figura 2, determina a qual faixa a pista o ponto pertence. Caso nenhum ponto foi detectado numa dada linha da imagem, então o módulo de *preenchimento de lacunas* é acionado (Figura 2). Ao final do processo, dois vetores contendo

as coordenadas de pontos pertencentes à faixa da esquerda e da direita são obtidos. Esses vetores determinam segmentos lineares que aproximam as faixas da pista. A última etapa do ADF na Figura 2 computa a inclinação média dos ângulos das faixas, usando os coeficientes angulares dos segmentos lineares encontrados.

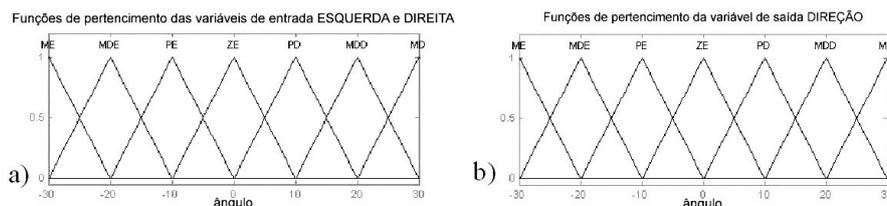
### 3. Sistema de navegação por inferência Fuzzy

Normalmente, as pessoas não usam informação numérica precisa em suas tomadas de decisão, apesar disto, elas são capazes de realizar tarefas de controle altamente precisas. Lógica Nebulosa (do inglês *Fuzzy Logic*) é baseada na teoria dos conjuntos nebulosos e é conhecida por ser um método organizado para lidar com dados imprecisos. Além disto, é um método apropriado para tratar incerteza, quando existe uma lacuna de precisão na informação, como por exemplo "muito quente", "um pouco frio" e "um pouco a esquerda". Estes tipos de informações devem ser formuladas matematicamente para que os computadores possam processá-la. Assim, a Lógica Fuzzy provê médias para incorporar a maneira de pensar do ser humano, em áreas aplicadas.

Na teoria clássica dos conjuntos, um conjunto inclui ou não um dado elemento, permitindo assim apenas dois valores de pertencimento: 0 ou 1 (falso ou verdadeiro). Na teoria dos conjuntos nebulosos, por outro lado, um elemento pode pertencer a um conjunto nebuloso com um certo grau de pertencimento. Logo, a Lógica Nebulosa é uma lógica de multi-valor que permite termos lingüísticos para definir valores intermediários, além de valores convencionais do tipo "sim/não", "verdadeiro/falso" ou "preto/branco".

Neste trabalho, um sistema de controle baseado em Lógica Nebulosa é utilizado no processo de tomada de decisão do re-planejamento da trajetória de navegação de um robô. Um Sistema de Inferência Nebuloso (SIN) é desenvolvido para decidir a direção e a velocidade que o robô autônomo deve tomar, a partir da sua posição atual na pista. Para tanto, os valores médios dos ângulos das faixas da pista (descritos na seção 2) são inicialmente codificados (*fuzzified*). Em seguida, as variáveis lingüísticas resultantes alimentam um conjunto de regras nebulosas que suportam o sistema de inferência. A decisão final é finalmente decodificada (*defuzzified*) e o sistema de controle do robô altera sua trajetória de navegação apropriadamente.

Para controlar a direção e a velocidade, o SIN representa situações em que os ângulos medidos são transformados em variáveis lingüísticas do tipo "muito muito à esquerda (MME)", "muito à direita (MD)" e "pouco à esquerda (PE)". Os limites dos ângulos foram definidos entre  $[-30, 30]$  para ambas variáveis de entrada e para a variável de saída de direção (Figura 3). Para a variável de saída velocidade, seus limites foram definidos entre  $[0, 8]$  (Figura 4). Foi utilizado o sistema de inferência *Mandani-Min* para as regras [Tsoukalas and Uhrig, 1997].



**Figura 3: Conjuntos de pertencimento das (a) variáveis de entrada (esquerda e direita); (b) e da variável de saída direção.**

O conjunto de regras para controle de direção é representado pela Figura 5(a), enquanto que as regras para controle de velocidade pela Figura 5(b). Uma regra é ob-

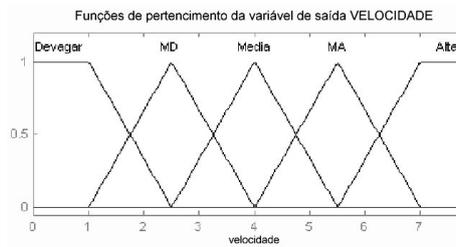


Figura 4: Conjuntos de pertinência da variável de saída *velocidade*.

tida na Figura 5 cruzando os valores de entrada do sistema, como por exemplo no conjunto de direção (Figura 5(a)) cruzando ângulo da direita *PE* (colunas) com o ângulo da esquerda *PD* (linhas) obtém-se *ZE*. Em outras palavras: if esquerda=*PE* and direita=*PD* then direcao=*ZE*.

DIREITA ESQUERDA	ME	MDE	PE	ZE	PD	MDD	MD
ME	ME	MDE	MDE	MDE	MDE	PE	ZE
MDE	MDE	MDE	PE	PE	PE	ZE	PD
PE	MDE	PE	PE	PE	ZE	PD	MDD
ZE	MDE	PE	PE	ZE	PD	PD	MDD
PD	MDE	PE	ZE	PD	PD	PD	MDD
MDD	PE	ZE	PD	PD	PD	MDD	MDD
MD	ZE	PD	MDD	MDD	MDD	MDD	MD

(a)

DIREITA ESQUERDA	ME	MDE	PE	ZE	PD	MDD	MD
ME	A	MA	M	M	M	D	A
MDE	MA	MA	M	MD	D	A	D
PE	M	M	MD	D	A	D	M
ZE	M	MD	D	A	D	MD	M
PD	M	D	A	D	MD	M	M
MDD	D	A	D	MD	M	MA	MA
MD	A	D	M	M	M	MA	A

(b)

Figura 5: Conjunto de regras do Sistema de Inferência Nebuloso para controle de (a) *direção* e (b) *velocidade*.

Procurou-se com essas regras cobrir toda superfície de controle, com transições suaves no controle de direção (Figura 6(a)). No caso do controle de velocidade, visa-se respostas rápidas em curvas acentuadas e retas, mas movimentos suaves em curvas amenas (Figura 6(b)).

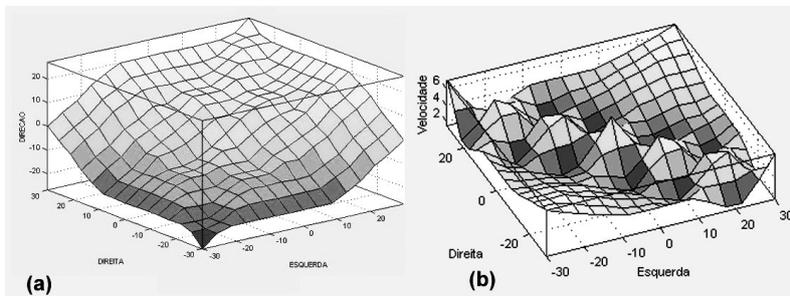


Figura 6: Superfícies de controle de (a) *direção* e (b) *velocidade*.

#### 4. Sistema de navegação baseado em Redes Neurais

Diferentemente de computadores digitais, as pessoas não necessitam de informações numéricas precisas para tomar decisões, podendo executar um controle altamente adaptativo. Nesta seção é proposto um mecanismo automatizado equivalente a um observador humano, que manualmente determina que direção seguir conforme as informações visuais da pista. Tal mecanismo necessita de adaptabilidade, sendo as Redes Neurais artificiais (RNA) propícias devido a sua capacidade de adaptação [Haykin, 2001]. Redes Neurais Artificiais são sistemas computacionais compostos por vários elementos que se inter-relacionam de forma conexa, inspirados no funcionamento de neurônios biológicos.

No trabalho anterior [Castro et al., 2001] foi desenvolvido um sistema de navegação autônoma adaptável, baseado em imagens utilizando controle por lógica nebulosa. O sistema nebuloso determina a direção a ser seguida e a velocidade a ser empregada

para o movimento do robô. Neste artigo, é apresentado um sistema de redes neurais para a tarefa de navegação. Tal qual a lógica nebulosa, a rede neural determina a direção a ser seguida para cada ponto da pista, sendo capaz de tomar decisões de replanejamento da trajetória. A inferência é decidir que direção e velocidade o veículo deve tomar a partir da posição atual.

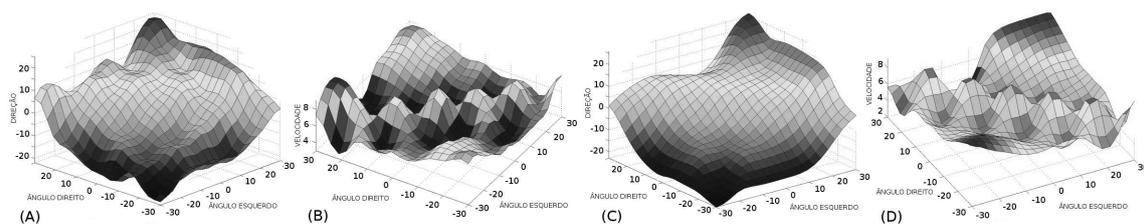
Foram utilizadas duas arquiteturas de redes neurais: rede com Perceptrons de Múltiplas Camadas (PMC) e rede de Funções de Base Radial (FBR). O método de aprendizado utilizado no treinamento das redes foi o algoritmo de retropropagação do erro (*error backpropagation*) [Haykin, 2001]. O treinamento do algoritmo das redes neurais consistiu na geração das informações obtidas pelo sistema nebuloso apresentado na seção 3, tendo como objetivo aprender as decisões tomadas pelo sistema de lógica nebulosa, para diversas combinações diferentes dos exemplos médios comentados anteriormente. A seguir, as duas arquiteturas de RNA utilizadas neste trabalho são brevemente descritas.

#### 4.1. Perceptrons de Múltiplas Camadas

A arquitetura dessa rede é composta por: duas entradas (ângulo da esquerda e ângulo da direita), duas camadas camadas escondidas (a primeira com 40 neurônios e a segunda com 20 neurônios) e dois neurônios na camada de saída (direção e velocidade). A função de ativação das unidades neuronais consiste na Logística Sigmóide. O Erro Quadrático Médio obtido ao final do treinamento foi  $3,58 \times 10^{-4}$  e as superfícies de controle estão ilustradas na Figura 7(a)(b).

#### 4.2. Rede de Funções de Base Radial

Na segunda rede utilizada, a arquitetura é composta por duas entradas (ângulo da esquerda e ângulo da direita), 100 neurônios na camada escondida, e dois neurônios na camada de saída (direção e velocidade). Os neurônios da camada escondida são compostos por funções de base radial do tipo *Gaussianas* bidimensionais, com espalhamento de 0,086. O vetor de centros das unidades foi determinado por um Mapa de Kohonen [Haykin, 2001]. O Erro Quadrático Médio obtido ao final do treinamento foi  $2,01 \times 10^{-3}$  e as superfícies de controle estão ilustradas na Figura 7(c)(d).



**Figura 7: Superfícies de controle do sistema neural baseado em PMC para (a) direção, (b) velocidade, e FBR para (c) direção e (d) velocidade.**

### 5. Envio de comandos ao robô

Todos sistemas de navegação desenvolvidos neste trabalho produzem valores de controle de velocidade que devem ser enviados ao robô. Essas saídas são representados por valores em ponto flutuante que pertencem ao universo de discurso da variável lingüística do controlador nebuloso. A fim de efetuar o controle físico do robô, é necessário que o mapeamento desse valor para o universo físico do veículo. Dois sub-sistemas diferentes foram implementados para efetuar o mapeamento da saída da velocidade para um valor de tempo de acionamento do robô. O primeiro sub-sistema utiliza um mapeamento linear para produzir um Comando de Controle de Tempo ( $cct_1$ ), como mostra a Equação 1, onde,

$v$  é a saída de velocidade do sistema de navegação,  $\alpha_1$  e  $\alpha_2$  são coeficientes lineares e  $\lambda$  é um limiar de velocidade.

$$cct_1 = \begin{cases} \alpha_1 v, v \leq \lambda \\ \alpha_2 v, v > \lambda \end{cases} \quad (1)$$

Os coeficientes foram testados experimentalmente para se atingir um controle suave do movimento do veículo.

Um sub-sistema mais complexo realiza um mapeamento não-linear para enviar ao robô um comando de Controle de Tempo ( $cct_2$ ), baseado em RNA's do tipo PML. A saída da rede neural é ainda ponderada por um coeficiente de amplitude  $\alpha$ .

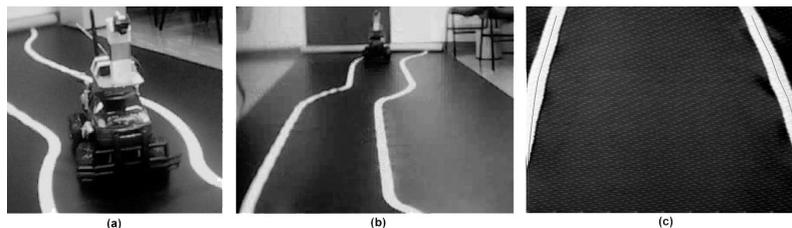
Esse sub-sistema é composto de duas redes neurais com duas camadas ocultas, com dez e cinco unidades, respectivamente. A primeira RNA é ativada quando o robô decide mover-se para frente, enquanto que a segunda RNA é utilizada para situações de movimentos em curva. Os dados de treinamento foram obtidos em experimentos, onde mediu-se a distância percorrida pelo robô na pista, num dado instante de tempo.

## 6. Resultados

Seis experimentos foram conduzidos afim de se validar os sistemas de navegação apresentados. Os experimentos estão identificados combinando-se o código do sistema de navegação com o código do sub-sistema de mapeamento utilizado, como mostra a Tabela 3. Além disto, na Tabela 3, estão descritos os valores utilizados nos parâmetros dos sub-sistemas de mapeamento. Os experimentos foram realizados com um robô comandado por rádio frequência (RF), equipado com uma câmera que transmite imagens da pista, alimentando assim o sistema de visão computacional para extração de características. A Figura 8(a) apresenta uma imagem do robô sobre a pista, com a câmera montada sobre ele. O robô move-se sobre uma pista de 7 metros, guiando-se entre duas faixas brancas delimitadoras. A Figura 8(b) mostra o robô no ponto inicial da pista e na Figura 8(c) é possível observar a respectiva imagem capturada da pista pela câmera do robô. Nota-se ainda o esqueleto das faixas da pista extraída pelo ADF, utilizado para inferir as direções da estrada.

**Tabela 3: Identificação dos experimentos realizados**

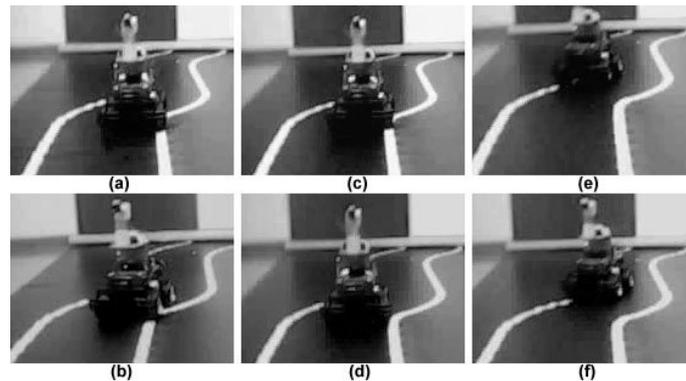
Sist. Navegação	Sub-sistema mapeamento	$\alpha_1$	$\alpha_2$	$\lambda$	$\alpha$
SIF (0)	Linear (A), N-linear (B)	0,10	0,06	2,50	3,00
RNA-PMC (1)	Linear (A), N-linear (B)	0,10	0,06	2,50	3,00
RNA-FBR (2)	Linear (A), N-linear (B)	0,10	0,06	2,50	3,00



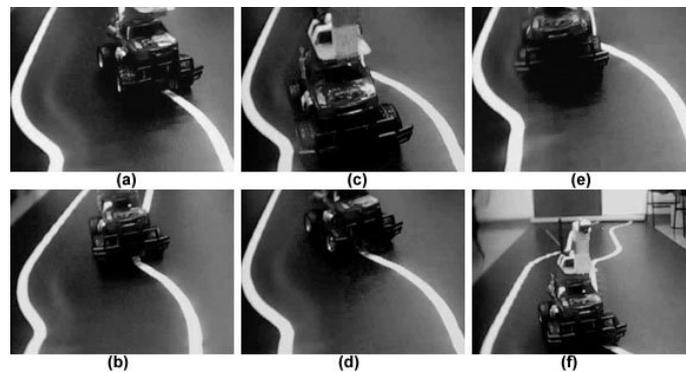
**Figura 8: (a) Robô e sua câmera montada; (b) imagem do veículo posicionado no ponto inicial da pista; e (c) a respectiva imagem capturada da pista.**

Cada sistema de navegação cumpriu o objetivo de guiar o veículo até o final da pista. Todavia, em algumas situações o robô cruzou os limites das faixas, voltando para o

centro da pista em seguida. Durante a navegação do robô foi observado que as 2 curvas da pista são pontos críticos para a tomada de decisão. A Figura 9 mostra o robô na primeira curva, durante os seis experimentos realizados. Nesta curva, o único sistema de navegação que não conduziu o veículo para cima das faixas foi do experimento 1B (Figura 9(d)). Nos experimentos 2A e 2B o veículo tocou a faixa da direita, enquanto nos demais experimentos o robô tocou a faixa da esquerda. Já na segunda curva, como pode ser visto na Figura 10, no experimento 1A (Figura 10(c))o veículo não tocou as faixas, enquanto que nos demais experimentos sim. Apesar de visualmente os controladores utilizados nos experimentos 1A e 1B terem demonstrado melhor desempenho em uma das curvas, vale ressaltar que nos demais experimentos o comportamento do robô foi mais estável, uma vez que o mesmo tocou pouco as faixas em ambas as curvas.



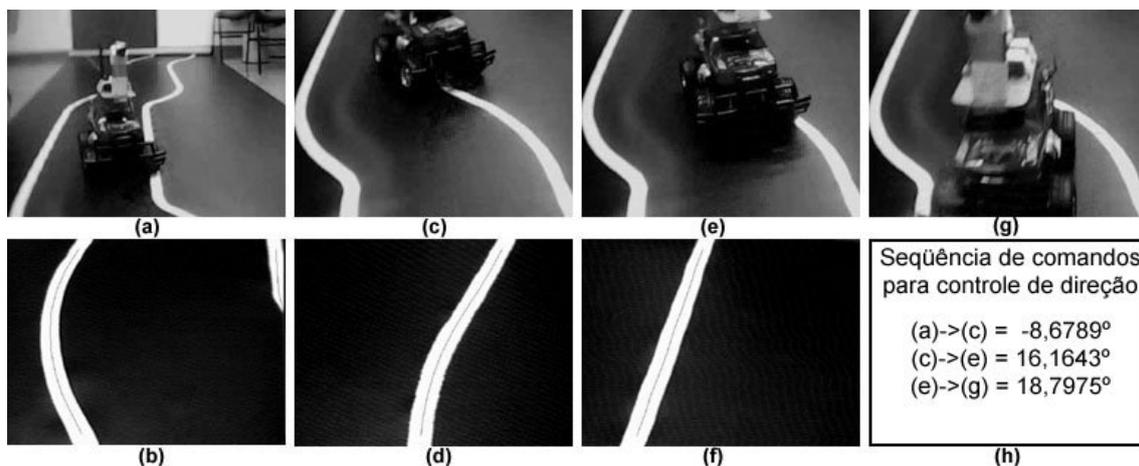
**Figura 9: Posição do robô na primeira curva da pista, durante os experimentos (a) 0A, (b) 0B, (c) 1A, (d) 1B, (e) 2A e (f) 2B.**



**Figura 10: Posição do robô na segunda curva da pista, durante os experimentos (a) 0A, (b) 0B, (c) 1A, (d) 1B, (e) 2A e (f) 2B.**

Apesar dos sistemas de navegação não estarem ainda perfeitamente afinados, o que levou o veículo eventualmente a cruzar uma das faixas da pista, os controladores se mostraram robustos, uma vez que o robô conseguiu replanejar sua trajetória com sucesso, e retornar à pista. Esta situação é ilustrada pela seqüência de imagens da Figura 11, do experimento 2B. Na posição do veículo da Figura 11(a), a imagem da pista (Figura 11(b)) levou o sistema de navegação a produzir uma saída contrária da ideal (Figura 11(h)). Todavia, em seguida, os próximos comandos trouxeram o robô para seu curso correto na pista, como é visto na seqüência de imagens da Figura 11(c)(e)(f).

Outra comparação foi realizada durante os experimentos afim de encontrar o sistema de navegação mais rápido. A Tabela 4 mostra que os sistemas de navegação baseado em redes neurais desempenharam um controle em menor tempo de processamento, quando comparados aos sistemas nebulosos. Vale ressaltar que esses tempos podem ainda ser melhorados, uma vez que não se utilizou nenhuma técnica de processamento paralelo



**Figura 11: Sequência de movimentos do carrinho, no experimento 2B. (a,c,e,g) mostram o posicionamento do carrinho na pista em função do tempo e (b,d,f) as respectivas imagens capturas da pista. (h) descreve a seqüência de comandos gerados a partir das imagens (b,d,f).**

nos cálculos. O número de comandos necessários para o robô percorrer toda pista foi menor no sistema de navegação baseado em redes neurais PMC, o que leva a um período de tempo menor para desempenhar a tarefa de navegação. Este número de comandos menor deve-se ao fato da RNA-PMC ter produzido comandos de velocidade mais elevados do que os demais controladores.

**Tabela 4: Tempo de processamento médio e número de comandos obtidos nos experimentos realizados**

Experimento	Tempo médio de processamento (s)	Nro. Comandos
0A	0,75	34
0B	0,75	32
1A	0,31	22
1B	0,34	15
2A	0,15	40
2B	0,14	32

## 7. Conclusões

Este trabalho buscou o desenvolvimento de um sistema de navegação autônomo baseado em imagens, que permitiu a um robô móvel a tomar decisões no re-planejamento e correção de sua trajetória. Para tanto, foram utilizadas informações visuais da estrada, imitando o comportamento humano de dirigir, através da extração de características da pista, realizada por um operador de detecção de bordas baseado de autômato. Dois sistemas de navegação foram desenvolvidos utilizando redes neurais, treinadas com dados provenientes do sistema baseado em lógica nebulosa. Os resultados de implementações mostram que os sistemas de navegação baseado em inferência por lógica fuzzy e redes neurais artificiais foram capazes de controlar indiretamente a navegação do robô, mesmo utilizando-se dados de sensores e atuadores imprecisos.

Os sistemas baseados em redes neurais artificiais alcançaram um desempenho computacional superior quando comparado ao sistema por lógica nebulosa. O sistema baseado em RNA-PMC obteve o melhor traçado nas curvas, enquanto que o sistema baseado em RNA-FBR levou a um comportamento mais estável nas curvas. O uso de sistemas de navegação rápidos computacionalmente encoraja a construção de sistemas para

tempo real, porém, deve-se aprimorar a precisão dos comandos de controle. As saídas dos sistemas de navegação foram mapeados para comandos de controle por dois sub-sistemas diferentes. O uso do sub-sistema  $tcc_2$  necessita de melhor reinamento de suas redes neurais, através de uma nova base de dados que cubra todo limite de possibilidades.

Nos experimentos realizados não foram consideradas situações imprevistas, como por exemplo obstáculos na pista. Todavia, o controle de colisão com obstáculos na pista pode ser realizado através da verificação de modificação das condições da pista, utilizando informações de contraste de imagens. Ressalta-se ainda que a escolha de um operador de visão computacional é um fator sensível, quando se trabalha com esses tipos de sensores. Em trabalhos futuros, busca-se aumentar a eficiência do processo de tomada de decisão através do uso de controladores que evitem comandos de recuperação de trajetória.

## Referências

- Castro, A. P. A., Silva, J. D. S., and Simoni, P. O. (2001). Image based autonomous navigation fuzzy logic control. In *IJCNN International Joint Conference on Neural Networks*, volume 1, pages 2200–2205, Washington.
- Haykin, S. (2001). *Redes neurais: princípios e prática*. Bookman, Porto Alegre.
- Heinen, F. J. (2001). Sistema de controle híbrido para robôs móveis autônomos. Master's thesis, Universidade do Vale do Rio dos Sinos, São Leopoldo.
- Hopcroft, J. E. and Ullman, J. D. (1979). *Introduction to automata theory, languages, and computation*. Addison-Wesley, Massachusetts.
- Kosaka, A. and Pan, J. (1995). Purdue experiments in model-based vision for hallway navigation. In *Proceedings of Workshop on Vision for Robots in IROS'95*, pages 87–96.
- Li, I., Xu, C., Xiao, Q., and Xu, X. (2003). Visual navigation of an autonomous robot. In *International Conference on Robotics and Automation*, pages 14–19, Taiwan.
- Li, W., Lu, G., and Wang, Y. (1997). Recognizing white line markings for vision guided vehicle navigation. *Fuzzy Reasoning - PRL*, 18(8):771–780.
- Meng, M. and Kak, A. C. (1993). Mobile robot navigation using neural networks and nonmetrical environment models. *IEEE Control systems*, pages 30–39.
- Sargent, R., Bailey, B., Witty, C., and Wright, A. (1997). Dynamic object capture using fast vision tracking. *Artificial Intelligence Magazine*, 18(1):65–72.
- Shah, S. and Aggarwal, J. K. (1996). Mobile robot navigation and scene modeling using stereo fish-eye lens system.
- Silva, J. D. S. and Simoni, P. O. (2002). The correspondence problem: an uncertainty reasoning approach. In *IASTED International Conference on Artificial Intelligence and Applications*, pages 210–215, Málaga.
- Talluri, R. and Aggarwal, J. K. (1993). Autonomous navigation in cluttered outdoor environments using geometric visibility constraints. In *Proceedings of Intl. Conf. on Intelligent Autonomous Systems, IAS-3*.
- Tsoukalas, L. H. and Uhrig, R. E. (1997). *Fuzzy and neural approaches in engineering*. Wiley & Sons, New York.