

Controle de Locomoção de uma Cadeira de Rodas com Detecção de Colisão por Inferência Nebulosa

Marcelo A. Lovatte¹, Luis E. M. Lima².

¹ ²Coordenadoria de Engenharia Elétrica, Instituto Federal do Espírito Santo
Avenida Vitória, 1729 - Bairro Jucutuquara
29040-780 - Vitória - ES
¹marcelovatte@hotmail.com
²luisedu@ifes.edu.br

Resumo – Este artigo apresenta um estudo sobre geração de trajetória para uma cadeira de rodas com uso de tração diferencial em seu sistema de locomoção, e comportamento reativo para evitar colisões. A geração de trajetória é baseada no uso de polinômios de terceira ordem, e o comportamento reativo obtido com inferência nebulosa. Esta mescla permite a geração de uma trajetória entre dois pontos quaisquer de um ambiente (inicial e final), caracterizados por suas coordenadas cartesianas e a orientação da cadeira, sendo evitadas colisões ao longo da execução desta trajetória a partir de uma inferência nebulosa que utiliza como variáveis de entrada a velocidade de locomoção atual e a distância ao obstáculo, e como variável de saída, a velocidade futura da cadeira. Uma cadeira de rodas real foi dotada de sensores de distância (ultrassom) e de velocidade (*encoders*) para que a detecção de obstáculos e a geração de trajetória fossem implementados em um sistema embarcado.

Palavras chave – Cadeira de rodas, Geração de trajetória, Lógica Nebulosa, Comportamento Reativo, Microprocessamento.

1 Introdução

Segundo o IBGE (Instituto Brasileiro de Geografia e Estatística), os resultados do Censo 2000 mostram que, aproximadamente, 24,6 milhões de pessoas, ou seja, 14,5% da população brasileira apresentam algum tipo de necessidade especial. Em busca de contribuições para a solução dos problemas dessa grande demanda populacional, que abrangem desde a falta de planejamento urbano adequado até o preço das cadeiras de rodas automáticas (acima de R\$ 5 mil), este trabalho visa a obtenção de melhorias para a potencialidade das cadeiras de rodas, quanto à mobilidade e transporte em ambientes fechados. O objetivo é dotar uma cadeira de rodas com autonomia suficiente para locomoção em ambiente estruturado, com trajetória pré-definida e a capacidade de evitar colisões.

A navegação autônoma de plataformas móveis é um foco de pesquisa bastante explorado na atualidade, e têm sido obtidas importantes contribuições a este tema com o uso de técnicas de inteligência artificial [1]. Um subproblema deste enfoque é a geração de trajetória, cuja solução, é a definição de um caminho que permita o deslocamento da plataforma de uma posição e orientação inicial até a posição e orientação final pré-estabelecida, respeitando condições cinemáticas de interesse, e evitando colisões.

Neste trabalho, é apresentada uma proposta de solução para a navegação autônoma de uma cadeira de rodas, com sistema de locomoção em configuração tração-diferencial, utilizada em ambiente fechado. Esta proposta se baseia em uma mescla de geração de trajetória por uso de polinômios paramétricos de terceiro grau proposto por Pedrosa [2], e a detecção de obstáculos com ajuste da velocidade de locomoção, a partir de uma inferência nebulosa sobre a distância da cadeira ao obstáculo e a velocidade atual da cadeira.

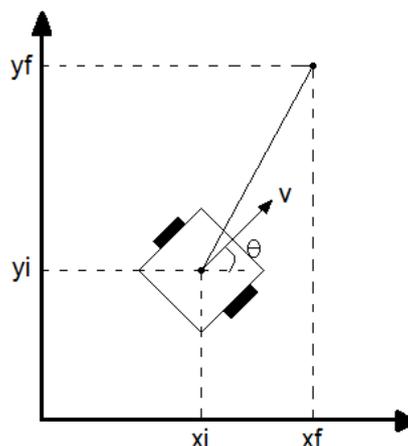


Figura 1 – Representação da cadeira no espaço.

A inferência nebulosa utilizada, para definição do comportamento reativo “evitar colisões”, permite uma eficiente aproximação entre a decisão computacional e a decisão humana, o que viabiliza o ajuste da dirigibilidade da cadeira ao conforto do usuário, com suaves transições entre distintas situações durante a locomoção [3].

Com o uso da lógica nebulosa foi possível tratar relações entre velocidade (medida por *encoder* óptico) e distância (medida por ultrassom) atribuindo a cada uma destas variáveis três distintas classificações (valores linguísticos), a citar: curta, média e longa para distância e baixa, média e alta para velocidade. A partir destas relações, a inferência nebulosa define a velocidade compatível à locomoção, aliando conforto para o usuário e segurança ao evitar colisões.

Os resultados desta proposta já foram comprovados através de simulação nos modos comando manual, geração de trajetória e comando vagar aleatoriamente pelo ambiente, assim como testes reais em ambiente fechado.

2 Geração de Trajetória

O controle de sistemas embarcados é caracterizado por baixo custo computacional, tanto em hardware como em software, o que viabiliza portabilidade, velocidade de processamento, controle em tempo real e autonomia por mais tempo de uso. Em função destas características, foi escolhido o método de geração de trajetória proposto por Pedrosa [2], pois a definição do caminho é obtida através de uma equação fechada, e de processamento rápido o suficiente para permitir o cálculo de posição e orientação da cadeira a cada passo de amostragem.

Para uma trajetória cujos parâmetros iniciais e finais são:

$$x_i = 30; \quad x_f = 50; \quad y_i = 20; \quad y_f = 70; \quad \theta_i = 0; \quad \theta_f = 30;$$

Os polinômios de terceiro grau apresentados abaixo representam geometricamente a trajetória, parametrizada por λ (sendo $\lambda \in (0,1)$ e para $\lambda=0$ início da trajetória e $\lambda=1$ final da trajetória) e definem a posição cartesiana do robô ponto-a-ponto da trajetória gerada.

$$x(\lambda) = a_0 + a_1\lambda + a_2\lambda^2 + a_3\lambda^3; \tag{1}$$

$$y(\lambda) = b_0 + b_1\lambda + b_2\lambda^2 + b_3\lambda^3; \tag{2}$$

Sendo:

$$\begin{cases} a_0 = x_i \\ a_1 = \text{arbitrário} \\ a_2 = \text{arbitrário} \\ a_3 = \Delta_x - a_2 - a_1 \\ b_0 = y_i \\ b_1 = d_i a_1 \\ b_2 = 3(\Delta_y - d_f \Delta_x) + d_f a_2 - 2(d_i - d_f) a_1 \\ b_3 = 3d_f \Delta_x - 2\Delta_y - d_f a_2 - 2(d_i - d_f) a_1 \end{cases} \tag{3}$$

Onde $d_i = \tan \theta_i$, $d_f = \tan \theta_f$, $\Delta_x = x_f - x_i$ e $\Delta_y = y_f - y_i$

Logo:

$$x(\lambda) = 30 + 40\lambda - 40\lambda^2 + 20\lambda^3; \tag{4}$$

$$y(\lambda) = 20 + 142,6\lambda^2 - 92,6\lambda^3; \tag{5}$$

$$\theta(\lambda) = \tan^{-1} = [y(\lambda) - y(\lambda - 1)]/[x(\lambda) - x(\lambda - 1)] \tag{6}$$

O ângulo de orientação θ é imposto de forma a respeitar o deslocamento da cadeira ao longo da trajetória, portanto, a cada passo de amostragem o ângulo dependerá da posição anterior ($\lambda-1$) e da posição atual (λ). Para a condição inicial ($\lambda=0$), a equação 6 não é satisfeita, então:

$$\theta(0) = \theta_i \tag{7}$$

Através da ferramenta de simulação, a trajetória é traçada conforme a figura abaixo:

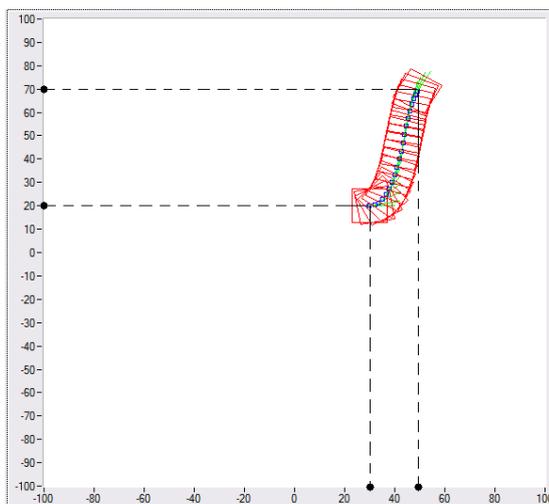


Figura 2 – Trajetória gerada a partir de polinômios de terceiro grau.

3 Inferência Nebulosa

Com as informações recebidas dos sensores ultrassônicos para medição da distância da cadeira aos obstáculos, e o sensoriamento com *encoders* para a velocidade de deslocamento, é possível identificar a tendência de colisão durante a realização da trajetória desejada.

Neste trabalho a inferência nebulosa foi implementada com duas funções de pertinência de entrada, definidas para as variáveis linguísticas velocidade de locomoção e distância à frente entre a cadeira e o obstáculo, sendo estas compostas por três conjuntos nebulosos com valores linguísticos baixa, média e alta para velocidade, e curta, média e longa para distância. Esta inferência inclui ainda a variável de saída que é definida como velocidade futura da cadeira, cuja função de pertinência também é subdividida em três conjuntos nebulosos para os valores baixa, média e alta [4].

Os conjuntos nebulosos mais utilizados são aqueles que possuem uma função de pertinência regular, ou seja, sem mudanças bruscas, que mostram claramente a passagem progressiva da não-satisfação para a satisfação da propriedade à qual eles estão associados [5]. O triângulo e o trapézio foram utilizados, pois representam bem estes tipos de conjuntos, indicados na figura abaixo:

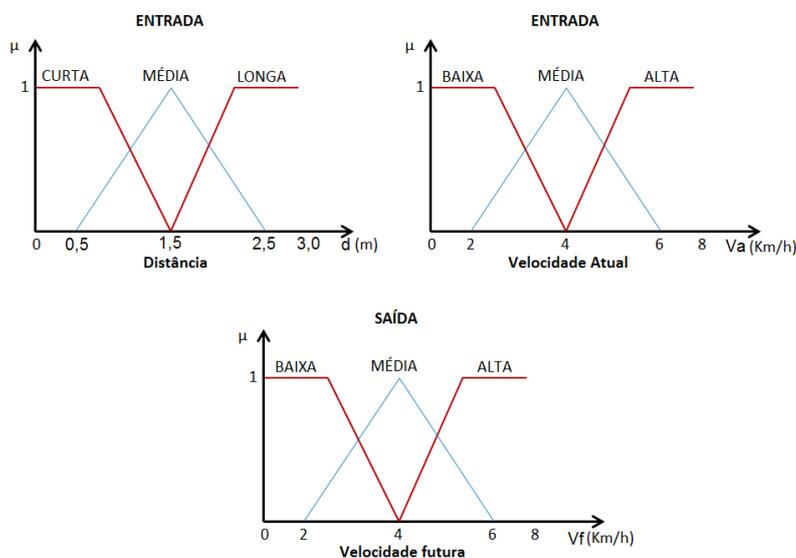


Figura 3 – Funções de Pertinência.

A partir dos conjuntos nebulosos de entrada e saída, foram definidas nove regras (tabela 1) que mapeiam as condições atuais de locomoção na definição da velocidade futura da cadeira. A quantidade total de regras geradas é resultado da combinação entre número de conjuntos nebulosos de entrada ($n = 6$) e do número de entradas ($p = 2$), calculados a partir da relação a seguir:

$$N^{\circ} \text{ de regras} = C(n, p) - 2 \times C\left(\frac{n}{2}, p\right); \quad (8)$$

Sendo:

$$C(n, p) = n! / p! (n - p)!; \quad (9)$$

Logo:

$$N^{\circ} \text{ de regras} = C(6, 2) - 2 \times C(3, 2) = 9 \text{ regras};$$

Na equação 7, uma combinação foi subtraída, pois, as regras deve ser definidas apenas pela combinação de valores linguísticos de funções de pertinência distintas.

O método de inferência utilizado foi o de máx-min de Mamdani [6], e as regras compostas de acordo com o seguinte padrão:

SE a distância à frente é curta **E** a velocidade atual é alta **ENTÃO** a velocidade futura é baixa.

Tomando como referência o “modus ponens generalizado”, podemos representar a regra acima da seguinte forma:

$$\mu_{baixa}(v_{futura}) = \min\{\mu_{curta}(d), \mu_{alta}(v_{atual})\}; \quad (10)$$

Onde:

$\mu_{baixa}(v_{futura})$: Grau de pertinência da variável velocidade futura para o conjunto nebuloso “baixa”;

$\mu_{curta}(d)$: Grau de pertinência da variável distância para o conjunto nebuloso “curta”;

$\mu_{alta}(v_{atual})$: Grau de pertinência da variável velocidade atual para o conjunto nebuloso “atual”.

Tabela 1 – Regras Fuzzy utilizadas no sistema.

Entrada		Saída	Representação
Distância à frente	Velocidade atual	Velocidade futura	
Curta	Baixa	Baixa	$\mu_{baixa}(v_{futura}) = \min\{\mu_{curta}(d), \mu_{baixa}(v_{atual})\}$
Curta	Média	Baixa	$\mu_{baixa}(v_{futura}) = \min\{\mu_{curta}(d), \mu_{média}(v_{atual})\}$
Curta	Alta	Média	$\mu_{média}(v_{futura}) = \min\{\mu_{curta}(d), \mu_{alta}(v_{atual})\}$
Média	Baixa	Média	$\mu_{média}(v_{futura}) = \min\{\mu_{média}(d), \mu_{baixa}(v_{atual})\}$
Média	Média	Média	$\mu_{média}(v_{futura}) = \min\{\mu_{média}(d), \mu_{média}(v_{atual})\}$
Média	Alta	Média	$\mu_{média}(v_{futura}) = \min\{\mu_{média}(d), \mu_{alta}(v_{atual})\}$
Longa	Baixa	Média	$\mu_{média}(v_{futura}) = \min\{\mu_{longa}(d), \mu_{baixa}(v_{atual})\}$
Longa	Média	Alta	$\mu_{alta}(v_{futura}) = \min\{\mu_{longa}(d), \mu_{média}(v_{atual})\}$
Longa	Alta	Alta	$\mu_{alta}(v_{futura}) = \min\{\mu_{longa}(d), \mu_{alta}(v_{atual})\}$

A definição das regras deve também considerar algumas restrições dos dispositivos usados para realizar o sensoriamento do ambiente. Por exemplo, o sensor ultrassônico utilizado para medir distância a um obstáculo, tem alcance máximo de três metros, dessa maneira para uma distância considerada “longa” a cadeira de rodas não poderá ter uma velocidade de locomoção máxima.

A partir da definição das regras, e dos valores de entrada (distância e velocidade atual) fuzzificados, a inferência nebulosa processa o operador mín sobre as pertinências de cada antecedente (conforme já foi descrito acima) para definição da pertinência do consequente de cada regra disparada, e o operador máx para a definição da pertinência resultante dos consequentes inferidos (disparados), em cada conjunto nebuloso da função de pertinência de saída. Os coeficientes das regras de saída podem ser representados da seguinte forma:

$$\mu_{baixa}(v_{futura}) = \text{máx}\{\mu_{baixa-disparada}(v_{futura})\}; \quad (11)$$

$$\mu_{média}(v_{futura}) = \text{máx}\{\mu_{média-disparada}(v_{futura})\}; \quad (12)$$

$$\mu_{alta}(v_{futura}) = \text{máx}\{\mu_{alta-disparada}(v_{futura})\}; \quad (13)$$

No processo de defuzzificação foi escolhido o método do centro de área [7], em função de seu baixo custo computacional o que favorece o uso de uma arquitetura de controle embarcada de baixa complexidade.

4 Simulação

A geração de trajetória e a detecção de obstáculos foram programadas no software LabWindows/CVI 9.0[8], com uso da linguagem de programação C. No ambiente de simulação é possível movimentar a cadeira através de três opções: a primeira opção permite que o usuário movimente a cadeira manualmente em todas as direções, como em uma cadeira real, e mesmo que o usuário insista em realizar um movimento que irá resultar em uma colisão, a cadeira permanecerá parada.

A figura 4 apresenta a interface da ferramenta de simulação na opção vagar, sendo que a cadeira se move de forma totalmente aleatória, sem nenhuma interferência do usuário em seu controle. Quando há uma possível colisão com qualquer obstáculo, um novo caminho aleatório é executado. Como mostra a figura 4, duas opções estão selecionadas: “Exibir trajetória” e “Obstáculo 1”. Através da opção “exibir trajetória”, todo o caminho percorrido pela cadeira no ambiente é ilustrado, dessa maneira fica evidente o processamento da inferência nebulosa para a detecção de obstáculos, sendo que, quando há uma possível colisão os quadrados do rastro se aproximam (indicando frenagem) e quando não há obstáculos próximos, os quadrados do rastro se afastam (indicando aceleração). Já a opção “obstáculo 1”, insere no ambiente de navegação um novo obstáculo além das paredes.

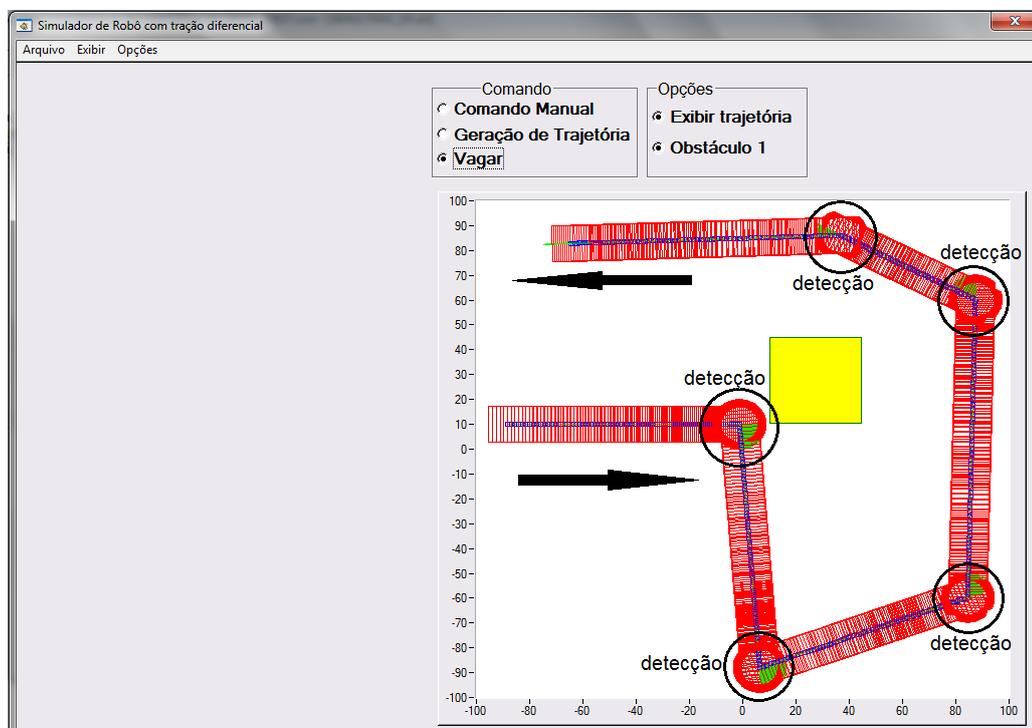


Figura 4 – Ambiente de simulação executando o comando “vagar” com pontos de detecção indicados.

O último comando para navegação é a geração de trajetórias. Neste modo o usuário informa ao simulador as posições e orientações inicial e final de cada trajetória a ser executada. A figura 5 ilustra a execução de uma trajetória com coordenadas iniciais (-80,-75), coordenadas finais (70,60), ângulo inicial igual a 90 graus e ângulo final igual a zero.

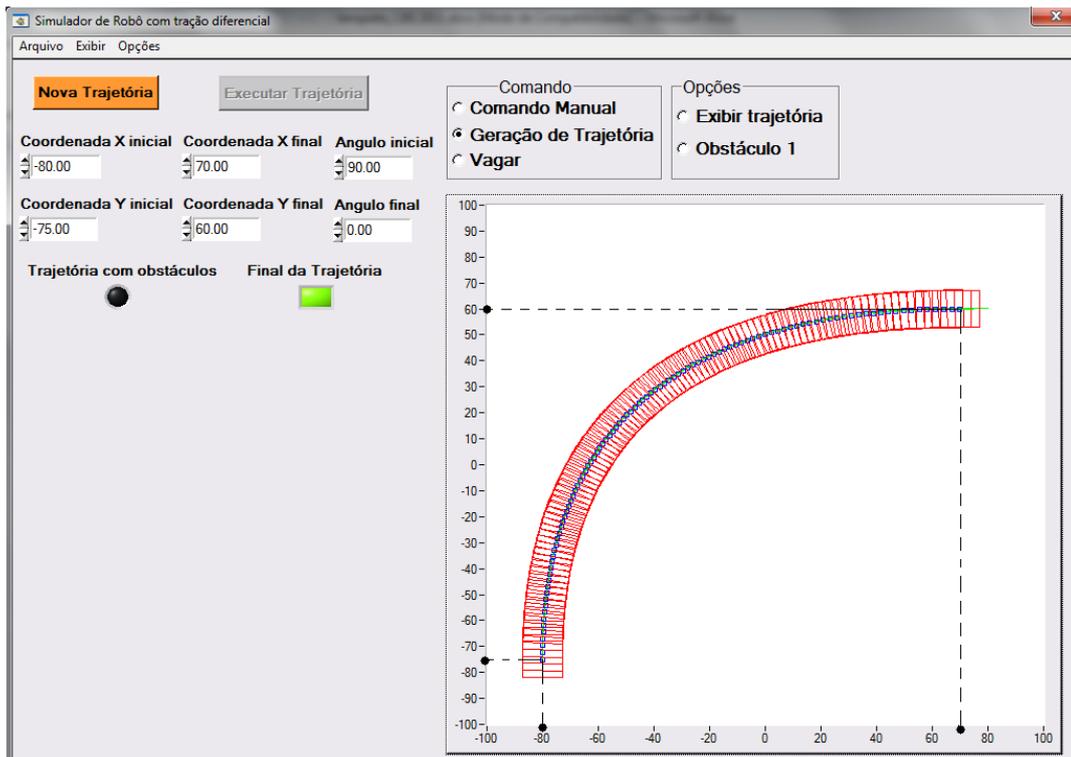


Figura 5 – Execução de uma trajetória sem obstáculos.

No caminho da trajetória da figura 5 não há obstáculos, dessa maneira o caminho é traçado sem que haja nenhuma interrupção. Mas se durante a execução de uma trajetória um obstáculo for detectado, a cadeira irá parar e efetuar um recuo para que um desvio possa ser realizado.

Para realizar o desvio, o usuário deve inserir as coordenadas para onde a cadeira de rodas deve ser locomover, e após o desvio, a trajetória será concluída automaticamente se não ocorrer outra possível colisão. O fluxograma da figura 6 apresenta o encadeamento de processos para a geração de trajetória com o desvio de obstáculo.

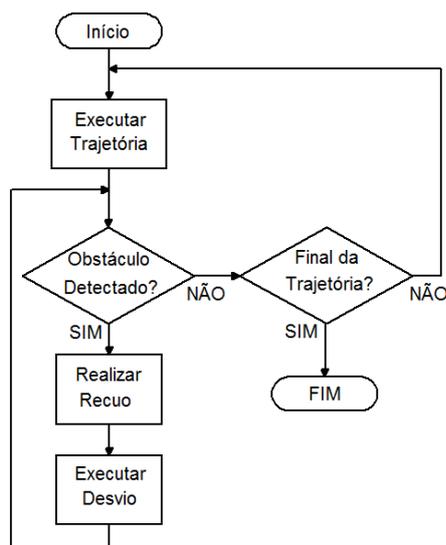


Figura 6 – Gerando uma trajetória com desvio de obstáculos.

Na figura 7, o processo descrito acima é executado para as coordenadas iniciais (80,-70), coordenadas finais (-20,80) e orientações inicial e final iguais a zero. Quando o obstáculo é detectado, o usuário insere as coordenadas de desvio (-10,-10).

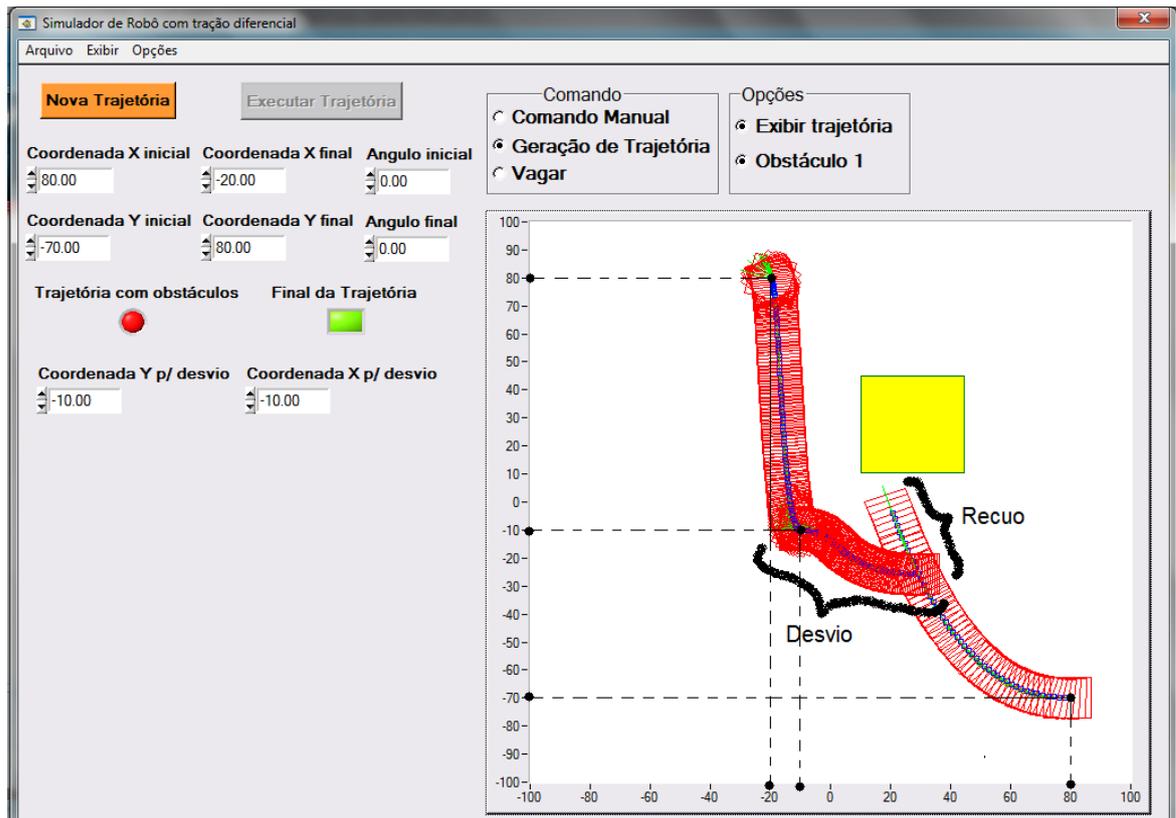


Figura 7 – Execução de uma trajetória com obstáculos.

5 Cadeira de Rodas Real

A cadeira de rodas está sensoriada por dois *encoders* (um para cada roda) para medição de velocidade e um sensor ultrassônico para medição de distância à frente em relação a obstáculos. A medição de distância através de um sensor ultrassônico impõe algumas restrições como: curta distância de alcance, reflexão em quinas gerando um falso obstáculo e pouca precisão para detecção de pequenos obstáculos. Estas restrições foram constatadas após vários testes realizados, sendo a inferência nebulosa sintonizada para o melhor desempenho de monitoramento em ambiente fechado. O tratamento das informações recebidas dos sensores é realizado a partir de um microcontrolador PIC 16F877A[9].



Figura 8 – Cadeira de rodas adaptada.

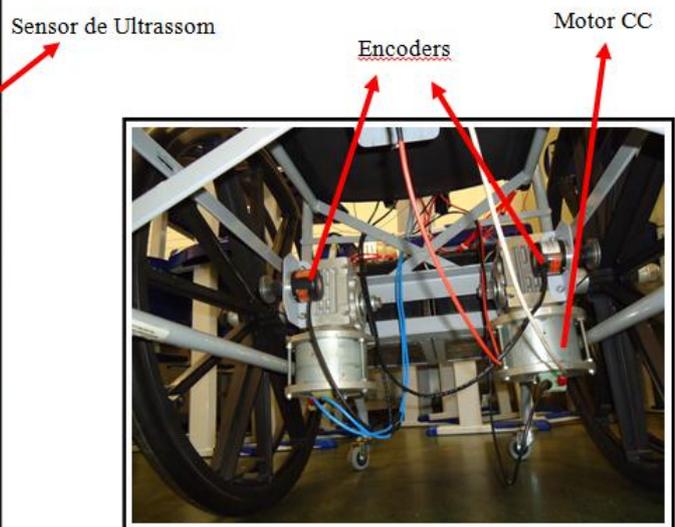


Figura 9 – Motores CC acoplados as rodas caracterizando a tração diferencial.



Figura 10 – Reação da cadeira de rodas ao encontro de um obstáculo.

6 Conclusões

O estágio atual de desenvolvimento deste trabalho permitiu a comprovação da validade dos resultados de simulação através de um sistema de controle embarcado em uma cadeira real. A detecção de obstáculos a frente da cadeira define a reação com redução de velocidade e parada evitando a colisão, o que já permite a locomoção da cadeira em ambiente com organização estática preservando a integridade física do usuário. O uso da inferência nebulosa proporciona suavidade no controle da reação à tendência de colisão, pois, o ajuste das funções de pertinência de entrada e saída viabiliza o controle fino de transição entre os níveis de velocidade futura a partir da transição de estados da velocidade atual e distância aos obstáculos. A partir da implementação do comportamento reativo, inicia-se o desenvolvimento do controle para a execução de trajetórias pré-definidas pelo método de polinômios de terceira ordem. A integração entre a realização de trajetórias e o comportamento reativo dota a cadeira com a capacidade de executar tarefas, em ambiente estruturado, com metas que incluam locomoção entre origem e destino livres de colisões, viabilizando também, para trabalhos futuros, a otimização de rotas.

7 Referências

- [1] ANDRADE, J. F. A. Geração de Trajetórias para Robôs Móveis Autônomos Via Redes Neurais Artificiais. 2001. 150 f. Dissertação (Mestrado em Engenharia Mecânica) – Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas. 2001.
- [2] PEDROSA, D. P. F., MEDEIROS, A. A. D. e ALSINA, P. J. Geração de caminhos ponto a ponto para robôs móveis com rodas, Congresso Brasileiro de Automática, Natal. 2002..
- [3] KULITZ, H. R. Modelagem e Controle Fuzzy de Robôs e Veículos Multiarticulados. 2004. Tese (Doutorado em Engenharia Elétrica) – Universidade Federal do Espírito Santo, Vitória. 2004.
- [4] JUNIOR, M. B. V.; KRONBAUER, D.; MIRANDA, P. Sistema de Controle Difuso de Mamdani Aplicações: Pêndulo Invertido e outras. 2003.61 f. Projeto de graduação em Análise de Sistemas – Centro de Ciências Exatas e Tecnologia, Universidade Federal de Mato Grosso do Sul, Mato Grosso do Sul. 2003.
- [5] CAMPOS, M. M.; SAITO, K. Sistemas Inteligentes em Controle e Automação de Processos. Rio de Janeiro: Ciência Moderna, 2004. 235 p.
- [6] MAMDANI, E.; ASSILIAN, S. “An experiment in linguistic synthesis with a fuzzy logic controller”, International Journal of Man Machine Studies, Vol. 7, No. 1, pp. 1-13. 1975.
- [7] PATEL, A. V. Simplest Fuzzy PI Controllers under Various Defuzzification Methods. International Journal of Computational Cognition, Vol. 3, No. 1, Março 2005.
- [8] NATIONAL INSTRUMENTS. LabWindows/CVI: User Manual. 1998. 239 p.
- [9] MICROCHIP TECHNOLOGY. PIC16F87X Datasheet: 28/40-Pin 8-Bit CMOS FLASH Microcontrollers. 2001. 216 p.