Navegação e mapeamento simultâneo em um carro autônomo baseado na lógica nebulosa

Álvaro Luiz Sordi Filho▪, Leonardo Presoto de Oliveira▫, André Schneider de Oliveira,
João Alberto Fabro e Marco Aurélio Wehrmeister

DAINF - Departamento de Informática
Universidade Tecnológica Federal Paraná
Curitiba, Brasil
▪alalvaro_7@hotmail.com
▫leonardopoliveira@hotmail.com

Resumo—Este trabalho apresenta o controle de navegação e mapeamento de um carro autônomo por sistema nebuloso (Fuzzy) que permite o desvio automático de obstáculos em ambientes desconhecidos. A estratégia baseia-se em um mapa do ambiente, que é criado de acordo com a navegação, para planejar as trajetórias através do algoritmo de busca A*. A abordagem proposta é avaliada em um ambiente virtual, onde o carro autônomo deve movimentar-se entre diversos obstáculos.

Palavras-chave — Carro autônomo, lógica nebulosa, mapeamento automático de obstáculos, algoritmo A*.

I. INTRODUÇÃO

A autonomia é a capacidade de um veículo (ou robô) movimentar-se por um ambiente conhecido, parcialmente conhecido ou desconhecido, a partir das percepções do ambiente (sensoriamento), de forma a construir o mapa (mapeamento), possibilitando o planejar e replanejar rotas de forma a atingir um ponto de destino, desviando dos obstáculos, sem nenhuma interferência de algum operador externo.

A lógica nebulosa é uma ferramenta para o desenvolvimento de estratégias de controle que permite uma maior flexibilidade nas regras do controlador, e consequentemente, permite a implementação da capacidade de autonomia. O modo de operação *Fuzzy* permite diferentes abordagens, como por exemplo em [1], onde é discutida uma forma de implementação da lógica nebulosa em sistemas reconfiguráveis aplicada ao controle de velocidade de cruzeiro de um carro autônomo. Outra abordagem semelhante é apresentada em [2]. A lógica nebulosa também pode ser aplicada para o controle de motores em um carro elétrico como discutido em [3].

Estratégias para Sistemas Avançados de Assistência ao Motorista (ADAS-*Advanced Driving Assistant Systems*) são também uma aplicação adequada para os sistemas nebulosos. Em [4] é discutida uma abordagem *Fuzzy* para o ajuste de um controle PID para o seguimento de pistas em uma autovia. Em [5] é apresentada uma abordagem para o evitamento de colisão (ou atropelamento) de veículos autônomos, pela detecção de pedestres. Em [6] é proposta uma abordagem nebulosa para o controle de cruzeiro de carros autônomos com aprendizagem online.

A navegação de veículos em ambientes dinâmicos, como estradas e autovias, é uma tarefa complexa devido a grande variedade de obstáculos e alterações no ambiente que podem ocorrer (defeitos na pista, falta de sinalização, interação com outros veículos, passagem de animais ou pedestres, entre outros). A aplicação de sistemas nebulosos é bastante promissora para essas situações, como visto em [7], onde é apresentada uma abordagem para navegação em ambientes desconhecidos baseada em diferentes tipos de sensores. Em [8] é apresentada a aplicação de sistemas *Fuzzy* para o controle distribuído de um veículo autônomo.

O foco principal deste trabalho é a capacidade de autonomia de veículos, que nada mais é que um agente capaz de extrair as informações do meio e se utilizar destas informações para mover-se de maneira eficiente no mundo. O ramo de aplicações destes robôs é enorme, podendo ser usados tanto para fins industriais (AGVs - Veículos auto guiados), militares (UAVs - veículos aéreos não tripulados), exploração, entre outros [9].

Este trabalho propõe a aplicação de um sistema nebuloso para o controle da navegação de um carro autônomo inserido em um ambiente estático, parcialmente observável (o sistema de percepção não tem acesso a todas as informações do ambiente), determinístico, discreto e de agente único. Os experimentos de validação são realizados utilizando um ambiente de simulação, e um modelo realístico de um veículo de passeio dotado com sensores e atuadores disponíveis comercialmente.

Nas seções seguintes serão discutidos os conceitos e técnicas envolvidos na simulação (Seção 2), a metodologia utilizada para o desenvolvimento (Seção 3), os resultados obtidos e as conclusões finais (Seção 4).

II. VEÍCULO AUTÔNOMO

Veículos autônomos podem ter diferentes topologias, dependendo principalmente da mobilidade e manobrabilidade

necessária para a aplicação. Um carro autônomo pode ser considerado como um robô móvel com diferentes sistemas para percepção do ambiente, com a capacidade de movimentar-se independente de comandos de um operador externo. Este veículo possui um sistema de movimentação chamado de geometria de Ackerman, a qual é composta por quatro rodas, duas fixas e duas direcionais. Uma discussão mais profunda sobre esse tipo de topologia pode ser visto em [10].

A percepção do ambiente é realizada por meio de sensores que realizam a medição de grandezas do ambiente. Carros autônomos, tradicionalmente, utilizam-se sensores para identificação de obstáculos (câmera, radar, LIDAR-*Light Detection And Ranging*, entre outros). Nesse contexto, o presente trabalho aborda um automóvel sedan tradicional, modelo Renault Fluence, que foi modelado em um sistema de experimentação virtual V-REP (*Virtual Robot Experimentation Platform*), da Coppelia Robotics [11]. No veículo foram dispostos quatro sensores radar (ultrassom) de médio alcance para identificar obstáculos nas laterais (tradicionalmente utilizado para a mudança de faixa e estacionamento automático). Na parte frontal, foi posicionado um radar (ultrassom) de longo alcance, para detectar obstáculos mesmo sob altas velocidades. Na parte traseira, foram posicionados sensores de estacionamento (ultrassom) e, em cima do capô, um sensor LIDAR (Hukuyo, que é um sensor a laser) para identificação e mapeamento do ambiente, como visto na Figura 1.

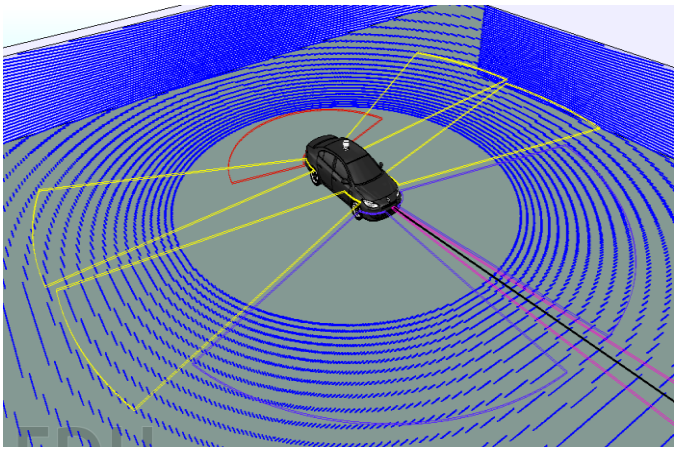

Figura 1 - Veículo autônomo modelo Renault Fluence, e alcance dos instrumentos de medição adicionados.

III. NAVEGAÇÃO E MAPEAMENTO

A autonomia do automóvel Renault Fluence é atingida com a implementação do conjunto de técnicas denominado como SLAM (*Simultaneous Localization and Mapping*) que permite que a construção e atualização do mapa sejam feitas durante a navegação do robô (uma discussão mais profunda sobre esta abordagem é apresentada em [12]).

O SLAM é um dos problemas mais discutidos na área da robótica. Esta técnica se refere à como o robô irá se portar em um ambiente desconhecido e como irá constituir o conhecimento que adquire na medida em que caminha pelo ambiente. A percepção do ambiente é feita por meio de seus sensores que podem ser classificados quanto a medição como: proprioceptivos (medem as grandezas internas ao veículo) ou exteroceptivos (medem informações do ambiente). Ainda podem ser categorizados de acordo com sua forma de medição como: passivos (medem a energia do ambiente) ou ativos (emitem sua própria energia) [12]. Esses sensores detectam as informações que o robô precisa para gerar seu mapa interno de navegação. A figura 2 representa a estrutura interna de um sistema de SLAM.

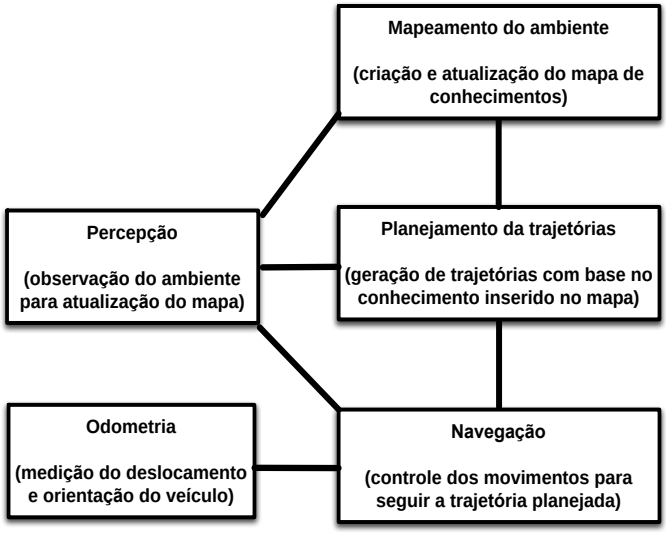

Figura 2 - Estrutura do Mapeamento e Navegação Simultâneos (SLAM).

Os sensores captam as percepções do ambiente e enviam para o mecanismo de Mapeamento, responsável por gerar e atualizar o mapa de conhecimentos. As informações do ambiente permitem que o robô possa se localizar (internamente no ambiente) e, assim, deliberar sobre de que maneira irá navegar. Esta estrutura também prevê que um objeto que não estava presente no ambiente possa aparecer em um segundo momento, onde o mapa é atualizado de acordo com a percepção atual.

A constante atualização do mapa é requisito da técnica de SLAM, sendo uma característica primordial, pois, se o robô navegar em um lugar e adicionar obstáculos ao mapa, em uma segunda navegação esse obstáculo pode ter sido deslocado e não mais existir conforme especificado no mapa. Neste caso, se o mapa for estático, o robô terá o mesmo comportamento de quando o obstáculo existia, ocasionando em erros.

A construção do mapa de conhecimentos é realizada por meio de uma estrutura de dados do tipo mapa de custos (*costmap*), no qual, o robô define os custos de todos os pontos presentes no mapa. O mapa de custos é utilizado para determinar o trajeto de navegação (planejamento da trajetória), a qual será o sinal de referência para o controlador de navegação. A trajetória é especificada de acordo com os

custos do mapa, buscando sempre o trajeto de menor custo [13].

As informações inseridas no mapa possuem erros de posicionamento e tamanho, em virtude dos erros de medição e odometria, os quais interferem diretamente na navegação e podem levar a colisões. Assim, todo objeto incluso no mapa é contornado por uma região de sombreamento, com custos inferiores ao do obstáculo, que diminuem prioridade da passagem do veículo nas regiões de imprecisão. Esta característica é fundamental em situações nas quais o veículo precisa passar por áreas estreitas e deve avaliar se ele realmente consegue passar ou não naquele espaço.

A. Planejamento da trajetória

Segundo Delling [14], o algoritmo A* busca e encontra, se possível, um caminho de menor custo a partir de um determinado nó inicial a um nó objetivo (de um ou mais objetivos possíveis). Para isso o algoritmo A* percorre um grafo (mapa) e segue o caminho com menor custo total esperado, mantendo uma fila de prioridade ordenada de segmentos de caminho alternativo ao longo do percurso.

O algoritmo A* usa uma função de custo do nó n (denotado geralmente f (n)) definida pelo custo já conhecido ,g(n), somado ao custo da estimativa heurística, h(n), para determinar a ordem em que a busca visita nós na árvore. Esta função de custo é a soma de duas funções: f(n) = g(n) + h(n).

Delling [14] também ressalta que h(n) deve ser uma heurística admissível, ou seja, ela não deve superestimar o custo para o destino. Assim, para uma aplicação para obtenção do menor caminho entre dois pontos, h(n) pode representar a distância em linha reta para o objetivo, uma vez que é fisicamente a menor distância possível entre dois pontos ou nós. Essa também é chamada de distância euclidiana.

Se a heurística h satisfaz a condição adicional de h(x) <= d(x, y) + h(y) para cada aresta (x, y) do grafo (em que d indica o comprimento dessa borda), h é então chamado consistente. Neste caso, o algoritmo A* pode ser implementado de forma mais eficiente, a grosso modo, nenhum nó precisa ser processado mais do que uma vez e A* é equivalente a executar o algoritmo de Dijkstra com o custo reduzido d '(x, y) = d (x, y) - h (x) + h (y) [10].

O algoritmo A* é responsável por permitir ao veículo autônomo planejar o caminho ótimo entre o ponto que ele se encontra até seu ponto objetivo. Foi feita uma adaptação no algoritmo A* que ao invés de retornar todos os pontos do caminho ótimo, o algoritmo A* retorna apenas os pontos de conversão presentes no caminho ótimo. Foi determinado que ponto de conversão é um ponto no qual o carro deve fazer uma curva, o caminho entre dois ponto de conversão deve ser feito em linha reta. Por exemplo, se o carro está no espaço (0,0) (x,y); e deve chegar ao ponto (10,10). O algoritmo A* pode retornar o seguinte caminho: (0,1), (0,2), (0,3), (0,4), (0,5), (0,6), (0,7), (0,8), (0,9), (0,10), (1,10), (2,10), (2,10), (4,10), (5,10), (6,10), (7,10), (8,10),(9,10),(10,10); ao todo são vinte pontos. Com a adaptação feita o algoritmo retornará apenas (0,0), (0,10), (10,10). Desta forma, é possível determinar que um linha reta entre os pontos (0,0) e (0,10), e outra linha reta entre os pontos (0,10), (10,10), e este será o caminho que o carro seguirá.

B. Navegação

Sistemas nebulosos são baseados na lógica Fuzzy que possui características diferentes da lógica tradicional. Uma proposição lógica tradicional tem duas situações concretas: é completamente verdadeiro ou é inteiramente falso. Porém, na lógica *fuzzy*, uma premissa pode variar entre falso ou verdadeiro, ou seja, em vez de ser apenas valores 0 ou 1, pode ocorrer de uma premissa assumir um valor entre 0 e 1. Sendo possível acontecer de uma premissa ser parcialmente verdadeira ou falsas [15].

A correta aplicação de um algoritmo baseado na lógica Fuzzy depende também do conhecimento que o especialista (projetista) tem sobre o sistema, já que as funções podem ser mapeadas por funções que são subjetivas e não probabilísticas.

A partição de um conjunto Fuzzy atribui graus de pertinência para os elementos deste conjunto. Os dados utilizados para a criação das regras foram gerados com base em testes feitos na plataforma. Foram testados aspectos como tempo de frenagem e aceleração do veículo simulado. A partir destes testes foi criada uma tabela de dados, que serviu de entrada para a criação das regras de inferência. Para a criação das regras de inferência Fuzzy foram utilizados os conceitos do algoritmo de Wang Mendel [16].

Este algoritmo de Wang Mendel é utilizado para a extração automática das regras pertinentes a um Sistema Fuzzy e pode ser resumido nos seguintes passos:Definir o número de termos linguísticos e particionar o universo de todas as variáveis de entrada; Criar uma regra Fuzzy para cada elemento do conjunto de pontos de treinamento;

- Para cada variável de entrada, selecionar a função de pertinência de maior grau;Calcular o grau de ativação de todas as regras, utilizando um operador apropriado.

No veículo autônomo o sistema Fuzzy tem por objetivo controlar não só a velocidade do carro, mas também controlar a rotação das rodas, quando esta se faz necessária. No total são aplicados três controladores Fuzzy.

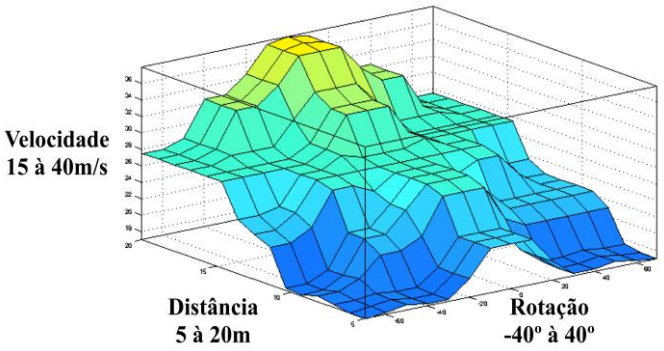

Figura 3 - Sistema de controle que auxilia a evitar colisão do carro com os obstáculos do ambiente.

O primeiro sistema Fuzzy é composto por duas entradas (distância e rotação das rodas dianteiras) e uma saída (velocidade). Objetivo deste sistema Fuzzy é garantir que o carro não vá colidir com o obstáculo. A Figura 3 ilustra este sistema. A variável de distância varia entre 5 e 20 metros, a variável de rotação entre -40º (orientação à esquerda) e 40º (orientação à direita) e a saída varia de 15 a 40 m/s. Outro sistema de controle simples complementa a operação e garante que se o carro estiver a 0.5 metros do obstáculo, é efetuada uma ré, pois, daquela posição já não seria mais possível manobrar e evitar o obstáculo.

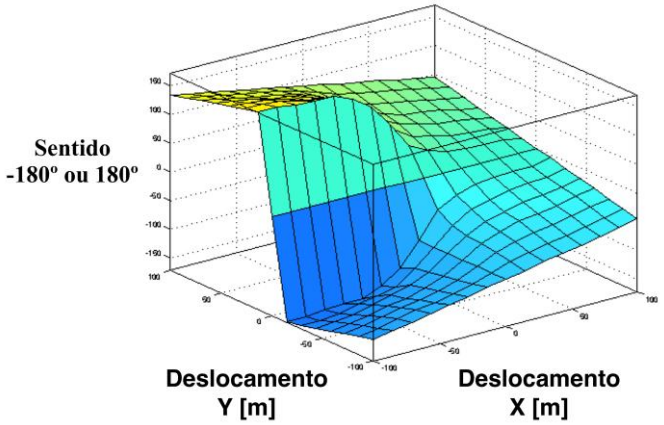

Figura 4 - Sistema de controle que determina qual a diferença entre a orientação do carro e a orientação do ponto objetivo.

Quando o carro está a uma distância menor que 5m, o controle é feito com base nos sensores dianteiros e faz com que o carro vire para o lado mais próximo ao ponto desejado. Este controle determina a diferença entre a orientação que o carro está e a orientação que ele deveria estar para seguir o caminho ótimo. A saída é o ângulo que o carro deve virar para seguir o caminho desejado e varia de 180º(esquerda) à 180º(direita), como ilustrado na Figura 4.

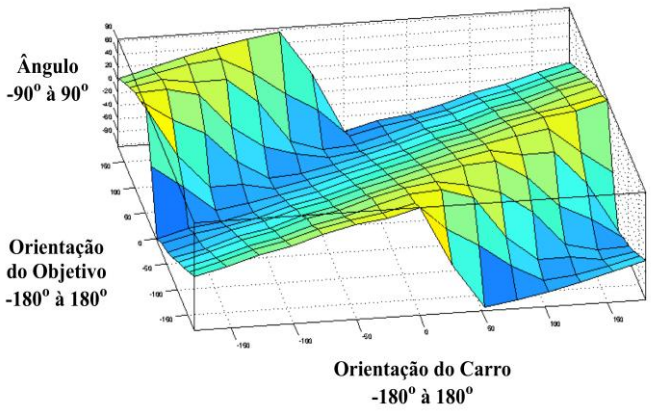

Figura 5 - Sistema Fuzzy que supervisionar o giro do carro, para que o mesmo alcance a orientação correta para chegar até o objetivo.

O terceiro sistema de controle Fuzzy (Figura 5) supervisiona o giro do carro e determina tanto rotação das rodas, quanto velocidade do movimento. Este controle é importante por que caso o carro se encontre em um lugar estreito, a manobra não poderá ser executada em apenas um movimento.

C. Árvore de Decisão

Árvore de decisão é a forma mais simples dentre os modelos de decisão utilizados, porém é bastante eficaz. Dentre os seus principais pontos fortes podem ser destacados a sua transparência e a facilidade de desenvolver as alternativas à serem escolhidas.

A estrutura de árvores de decisão são muito parecidas com a estrutura if-then, uma estrutura muito utilizada em sistemas especialistas e sistemas de classificação [17].

Na entrada de uma árvore de decisão são recebidos atributos que podem ser contínuos ou discretos, então, a árvore chegará à uma decisão final com base em seus testes. Na estrutura da árvore, cada nó representa um teste diferente e cada nó folha desta árvore representa um valor que será retornado caso este nó folha seja alcançado.

Na árvore de decisão cada nó representa o conhecimento do especialista que conduz a busca para um de seus nós filhos. Sendo assim na medida que se desce na árvore a configuração desejada do sistema vai sendo selecionado e desta forma se escolhe o comportamento desejado[18].

No veículo autônomo a árvore de decisão é utilizada para selecionar qual grupo de ações o agente deve tomar, ou seja, decidir se ele deve apenas seguir o caminho determinado pelo planejador de trajetória, desviar de algum obstáculo, ou até parar e efetuar uma ré caso o espaço disponível não seja suficiente para a manobra.

D. Estudo de caso

O estudo de caso é um automóvel Renault Fluence que é colocado em um ambiente virtual com diversos obstáculos.

Como definido pela geometria de Ackerman, apenas as rodas dianteiras determinam a direção, ou seja, as rodas traseiras são livres. Quando o carro precisa fazer uma curva, é necessário agir somente nas duas rodas da frente. O tamanho do carro é de aproximadamente 4 metros de comprimento por 2 metros de largura.

Figura 6 – Automóvel autônomo - Renault Fluence virtual.

O ambiente é um espaço fechado de 100m x 100m, conforme ilustrado na figura 7. Neste espaço foram definidos caminhos pelos quais o carro pode seguir e também obstáculos que ele deve desviar. Ao todo são quatro obstáculos nos formatos de cubos com tamanhos variados. Além dos obstáculos, existem paredes que limitam a passagem do carro.

As pistas laterais (delimitadas pelas paredes) são estreitas para dificultar uma possível curva que o carro precise fazer. Essa dificuldade faz com que o controle tenha que ser mais eficiente, pois, dependendo da velocidade não será possível que o carro consiga virar sem movimenta-se no sentido contrário (marcha ré) ou manobrar o carro para se adaptar a curva.

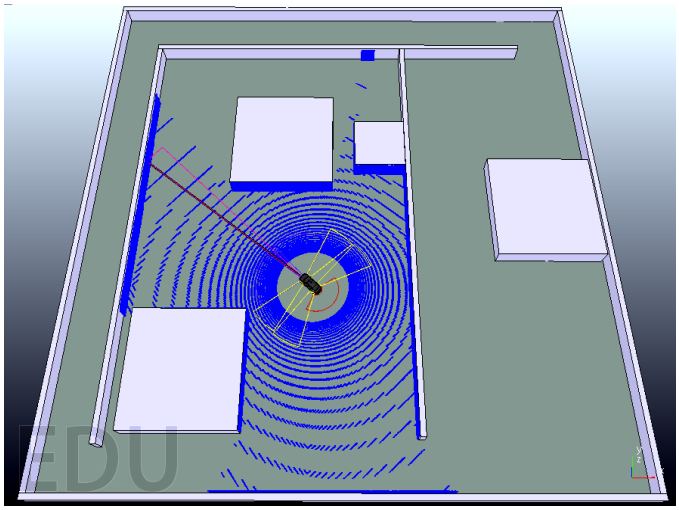

Figura 7 - Um carro que interage neste ambiente, que a principio não é conhecido (mas vai sendo mapeado), que tem como objeto chegar em um determinado ponto, percorrendo o caminho ótimo (utiliza-se do algoritmo A* para definir o caminho).

O experimento inicia-se com o automóvel parado em um determinado ponto e o mapa de conhecimentos vazio. Nesse instante, é designado o ponto objetivo para o carro autônomo. A primeira tarefa é realizar o planejamento da trajetória com base no mapa de conhecimento, então o trajeto é calculado pelo A* e então quando o algoritmo determina os pontos "sub-objetivos", o carro começa a se movimentar pelo caminho determinado. Os controladores Fuzzy são responsáveis por fazer o carro seguir a trajetória planejada.

O carro segue o caminho ótimo, ao mesmo tempo atualiza o mapa do ambiente. Quando eventualmente encontra um obstáculo muito próximo ou sob a trajetória planejada, a ação de controle Fuzzy opera com base nos sensores dianteiros para que o veículo desvie dos obstáculos.

A Figura 8 representa o mapa do ambiente sendo construído na medida em que o automóvel caminha. A área mais clara representa a posição onde o obstáculo foi encontrado; enquanto a área em cinza representa a inflação que o carro tem de calcular em volta dos obstáculos.

Figura 8 - Ambiente parcialmente mapeado pelo automóvel com auxilio do sistema de percepção.

Quando o obstáculo é superado, o controle indica que o carro deve voltar a seguir o caminho determinado pelo planejador de trajetória, porém o carro pode não estar mais orientado de forma que só necessite andar em frente. Nesse instante outro controlador Fuzzy é ativado para corrigir a orientação do veículo de acordo com a trajetória desejada (determinada pelo planejador de trajetória).

Após adequar a orientação do veículo ao caminho ótimo, o carro volta a seguir o caminho determinado pelo planejador de trajetória até alcançar seu objetivo. Devido ao tamanho do carro, foi determinado que o objetivo é qualquer ponto à um raio de 1 metros do objetivo original (margem de tolerância). Esta medida foi tomada, para evitar que o carro fique fazendo pequenas manobras para estar exatamente no ponto desejado, esta manobras levavam muito tempo e são insignificantes uma vez que o carro já estava acima do ponto e só estava tentando alinhar o centro do carro com o ponto objetivo.

O controlador Fuzzy para a correção de direção, utilizado de modo iterativo, possibilita o ajuste fino da direção (orientação) do veículo. Já o Fuzzy que controla a velocidade do carro em função da distância dos obstáculos foi importante para garantir que o carro não colida com os objetos.

A correção na direção do veículo, é uma dificuldade intrínseca ao problema, pois, como trata-se de um veículo de passeio, qualquer ajuste de direção requer movimentações contínuas para frente e para trás, até que se atinja o ângulo desejado.

IV. CONSIDERAÇÕES FINAIS

As simulações mostraram que o sistema responde de maneira eficaz. Inicialmente em um ambiente totalmente desconhecido, dado um ponto objetivo, o robô calcula a rota ótima, sendo esta uma linha reta. À medida que o robô descobre objetos e atualiza o seu mapa de conhecimentos, quando houver um obstáculo em sua trajetória a mesma é recalculada.

O método A* utilizado, apesar de produzir o resultado esperado, exige relativamente muito tempo computacional, fazendo com que o carro, na simulação, fique parado enquanto o planejador de trajetória encontra o caminho ótimo. Uma opção seria utilizar o algoritmo de Dijkstra, entretanto, para este caso ele demandaria mais tempo que o algoritmo A*, pois mesmo que o algoritmo de Dijkstra encontre um caminho, ainda sim precisaria testar todas as outras possibilidades para garantir que o caminho encontrado realmente é o melhor possível.

Os sistemas de controle nebulosos (Fuzzy) comprovaram a sua eficiência em relação ao que era esperado. O controle de orientação para o próximo sub-objetivo retornou respostas válidas para qualquer situação em que o carro e o objetivo se encontrassem.

O *SLAM* utilizado no trabalho também mostrou-se satisfatório na função de mapear o ambiente. O detalhe a ser destacado neste ponto é que a área de inflação no mapeamento pode interferir de maneira decisiva no desenvolvimento da trajetória. No caso deste trabalho foi necessário aumentar a área de inflação para garantir que o automóvel não colidisse com as quinas presentes no ambiente.

O veículo autônomo é capaz de navegar em ambientes diferentes do proposto, com curvas mais suaves, e pistas menos estreitas. Além de ambientes diferentes, sugere-se implementar um controle para tornar a transição da direção do veículo entre sub-objetivos mais suaves. O algoritmo desenvolvido neste projeto necessita primeiro chegar ao ponto destino (ou sub-objetivo) para calcular o próximo trajeto. Uma melhoria interessante poderia ser que uma vez que o veículo detecte que não há mais obstáculos entre a sua posição e o sub-ojetivo, já poderia começar a efetuar uma curva na direção do próximo sub-objetivo. Isto melhoraria o percurso principalmente em casos que é necessário virar 90° de um sub-objetivo a outro.

AGRADECIMENTOS

À Fundação Araucária e a Renault do Brasil pelo financiamento parcial desde projeto de pesquisa.

REFERÊNCIAS

[1] Nadia Nedjah, Paulo Renato S.S. Sandres, Luiza de Macedo Mourelle, Customizable hardware design of fuzzy controllers applied to autonomous car driving, Expert Systems with Applications, Vol. 41, Issue 16, November 2014.

[2] Dalibor Petković, Mirna Issa, Nenad D. Pavlović, Lena Zentner, Intelligent rotational direction control of passive robotic joint with embedded sensors, Expert Systems with Applications, Vol. 40, Issue 4, March 2013.

[3] Xu peng, Hou zhe, Guo Guifang, Xu Gang, Cao Binggang, Liu Zengliang, "Driving and control of torque for direct-wheel-driven electric vehicle with motors in serial", Expert Systems with Applications, Vol. 38, Issue 1, January 2011

[4] Qing Wang; Shou-Zhi Xu; Hong-Lei Xu, "A Fuzzy Control Based Self-Optimizing PID Model for Autonomous Car Following on Highway," *Wireless Communication and Sensor Network (WCSN), 2014 International Conference on* , vol., no., pp. 395-399, 13-14 Dec. 2014

[5] Llorca, D.F.; Milanes, V.; Alonso, I.P.; Gavilan, M.; Daza, I.G.; Perez, J.; Sotelo, M.A., "Autonomous Pedestrian Collision Avoidance Using a Fuzzy Steering Controller," *Intelligent Transportation Systems, IEEE Transactions on* , Vol.12, no.2, pp. 390-401, June 2011

[6] E. Onieva, J. Godoy, J. Villagrá, V. Milanés, J. Pérez, On-line learning of a fuzzy controller for a precise vehicle cruise control system, Expert Systems with Applications, Vol. 40, Issue 4, March 2013, pp. 1046-1053

[7] Farooq, U.; Hasan, K.M.; Amar, M.; Asad, M.U., "Design and implementation of fuzzy logic based autonomous car for navigation in unknown environments," *Informatics, Electronics & Vision (ICIEV), 2013 International Conference on* , pp.1-7, 17-18 May 2013

[8] Humberto Martínez-Barberá, David Herrero-Pérez, "Multilayer distributed intelligent control of an autonomous car", Transportation Research Part C: Emerging Technologies, Vol. 39, February 2014, pp. 94-112

[9] Terano, Toshiro, Kiyoji Asai, and Michio Sugeno, eds. *Applied fuzzy systems*. Academic Press, 2014.

[10] Ackermann, J., T. Bünte, and D. Odenthal. "Advantages of active steering for vehicle dynamics control." (1999).

[11] Rohmer, E.; Singh, S.P.N.; Freese, M., "V-REP: A versatile and scalable robot simulation framework," *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on* , pp.1321-1326, 3-7 Nov. 2013

[12] Siegwart, Roland, Illah Reza Nourbakhsh, and Davide Scaramuzza. *Introduction to autonomous mobile robots*. MIT press, 2011.

[13] King, Jennifer, and Maxim Likhachev. "Efficient cost computation in cost map planning for non-circular robots." *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*. IEEE, 2009.

[14] Delling, D. and Sanders, P. and Schultes, D. and Wagner, D. (2009). "Engineering route planning algorithms". *Algorithmics of large and complex networks*. Springer. pp. 117–139.

[15] Melo, L.G. "Sistema Fuzzy Probabilístico Geração Automática de regras e Defuzzificação Bayesiana". Dissertação de Mestrado em Informática Industrial. Universidade Tecnológica Federal do Paraná, Curitiba. 2011.

[16] Wang, Li-Xin, and Jerry M. Mendel. "Fuzzy basis functions, universal approximation, and orthogonal least-squares learning.", *IEEE Transactions on Neural Networks* 3.5 (1992): 807-814.

[17] de Soárez, Patrícia Coelho, Marta Oliveira Soares, and Hillegonda Maria Dutilh Novaes. "Modelos de decisão para avaliações econômicas de tecnologias em saúde." *Revista Ciência & Saúde Coletiva* 19.10 (2014).

[18] Pozzer, Cesar Tadeu. "Aprendizado por árvores de decisão." Disponível: http://www-usr.inf.ufsm.br/~pozzer/disciplinas/pj3d_decisionTrees.pdf. Acesso em: 14 dez. 2010, *Universidade Federal de Santa Maria, Rio Grande do Sul* (2006).