

Navegação Local do ATLASCAR2 para Condução Autónoma e Assistência ao Condutor

Ricardo Luís Fernandes Silva ¹

Orientador: Professor Doutor Vítor Santos

Coorientador: Professor Doutor Paulo Dias

¹ricardolfsilva@ua.pt

Departamento de Engenharia Mecânica
Universidade de Aveiro

26 de julho de 2018



Sumário

- 1** Introdução
- 2** Revisão da literatura
- 3** Infraestrutura experimental
- 4** Solução de navegação local
- 5** Ambiente de simulação
- 6** Resultados
- 7** Conclusões e trabalhos futuros

Introdução

1 Introdução

- Assistência à condução /
Condução autónoma
- Enquadramento e motivação
- Objetivos

2 Revisão da literatura

3 Infraestrutura experimental

4 Solução de navegação local

5 Ambiente de simulação

6 Resultados

7 Conclusões e trabalhos futuros

Assistência à condução / Condução autónoma

- Segundo a Organização Mundial de Saúde, em 2013 ocorreram cerca de 1,25 milhões de mortes nas estradas.
- A indústria automóvel introduziu Sistemas Avançados de Assistência à Condução (ADAS) para aumentar a segurança.
- Os sistemas ADAS evoluíram para sistemas mais complexos com a finalidade da Condução Autónoma (AD).



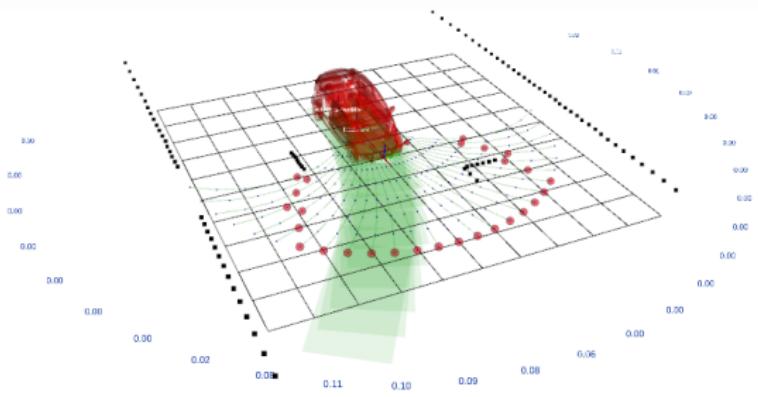
O projeto ATLAS

- O projeto ATLAS foi criado no Lab. de Automação e Robótica (LAR), para o estudo de sensores e sistemas avançados de condução.
- As primeiras experiências ocorreram em modelos à escala.
- Recentemente, foram instrumentados dois automóveis reais, o ATLASCAR1 e o ATLASCAR2 (plataforma atual).



Enquadramento

- A proliferação de tecnologias como o LIDAR e o GPS expandiu a investigação da AD a institutos e universidades.
- A capacidade da recolha de dados reais, por parte do ATLASCAR2, permite a que a UA também explore esta área.
- Esta dissertação surge para desenvolver um módulo de navegação local para assistência ao condutor na tomada de decisão imediata.



Objetivos

- Identificação e seleção das estruturas de dados mais adequadas aos algoritmos de navegação local.
 =
- Identificação e seleção dos algoritmos de navegação local mais adequados aos ambientes esperados para o ATLASCAR2.
 =
 =
- Implementação e teste do algoritmo selecionado para fazer a navegação local.
- Desenvolvimento de uma aplicação de monitorização do algoritmo de navegação local.

Revisão da literatura

1 Introdução

2 Revisão da literatura

- Planeamento em AD
- Algoritmos

3 Infraestrutura experimental

4 Solução de navegação local

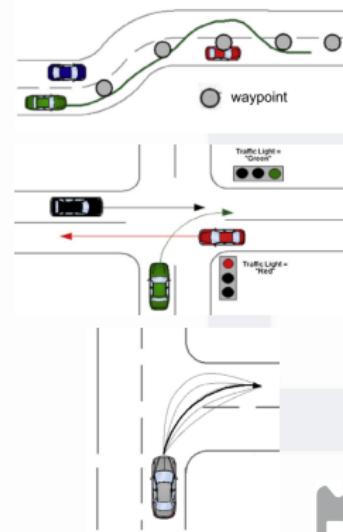
5 Ambiente de simulação

6 Resultados

7 Conclusões e trabalhos futuros

Tipos de planeamento

- O planeamento em AD pode ser dividido em dois superiores: o planeamento local e o planeamento global (rotas).
- Por sua vez o planeamento local é responsável por 3 atividades bem distintas:
 - Planeamento do caminho;
 - Planeamento de manobras;
 - Planeamento da trajetória.



Planeamento da trajetória

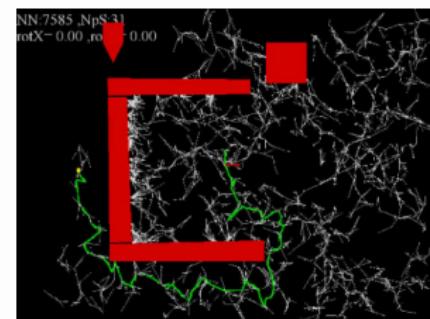
"Trajectory planning is concerned with the real-time planning of the actual vehicle's transition from one feasible state to the next, satisfying the vehicle's kinematic limits based on vehicle dynamics and constrained by the navigation comfort, lane boundaries and traffic rules, while avoiding, at the same time, obstacles including other road users as well as ground roughness and ditches."¹

¹Christos Katrakazas et al. "Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions". Em: *Transportation Research Part C: Emerging Technologies* 60 (2015), pp. 416–442. DOI: 10.1016/j.trc.2015.09.011.

Algoritmos de planeamento de trajetória

- Os algoritmos de planeamento de trajetória calculam a trajetória ótima entre a posição atual e o objetivo.

- Campos de potencial
- A*
- JPS
- VFH
- PRM
- RRT

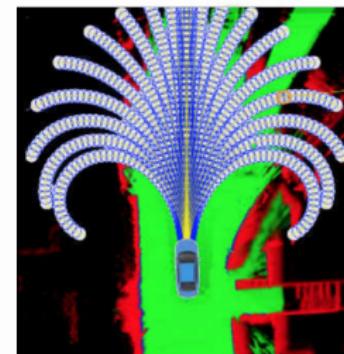


- ✓ Trajetória ótima de acordo com os parâmetros do algoritmo e o espaço de planeamento.
- ✗ Execução mais demorada, principalmente quando aplicados em grelhas.

Algoritmos de planeamento por hipóteses

- Os algoritmos de planeamento por hipóteses são algoritmos de avaliação e seleção de trajetórias hipotéticas, dentro de um conjunto discreto.

- Grelhas evidenciais;
- Abordagem por múltiplas hipóteses.



- ✓ Planeamento mais rápido e reativo.
- ✗ Não determinam a trajetória ótima.

Infraestrutura experimental

1 Introdução

2 Revisão da literatura

3 Infraestrutura experimental

- Hardware
- Software

4 Solução de navegação local

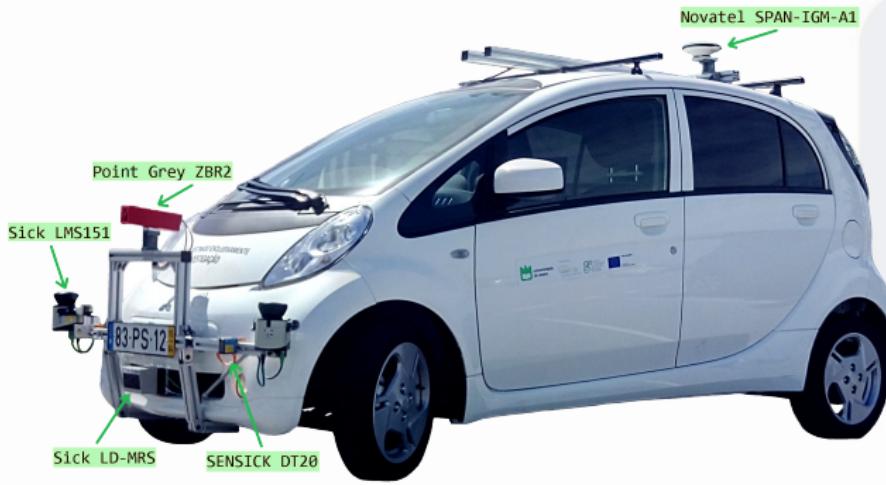
5 Ambiente de simulação

6 Resultados

7 Conclusões e trabalhos futuros

Plataforma ATLASCAR2

- Baseada num automóvel elétrico, um *Mitsubishi i-MiEV*.
- Possui instalados sensores LIDAR, de localização (GNSS), e visão. Existe ainda um servidor para processamento de dados e uma UPS.
- Nesta dissertação foram utilizados, diretamente, os sensores LIDAR.



LIDARs

- Deteção de obstáculos através de laser pulsado.
- *Sick LMS151*
 - 1 feixe planar com abertura de 270° ;
 - Deteção de obstáculos desde 0,5 m até 50 m.
- *Sick LD-MRS400001*
 - 4 feixes planares distados de $0,8^\circ$ e com abertura de 85° ;
 - Deteção de obstáculos desde 0,5 m até 300 m.



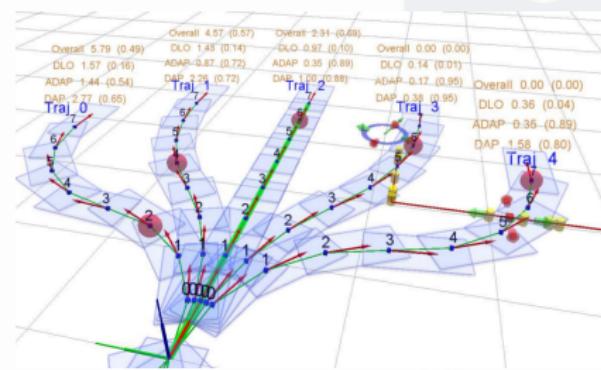
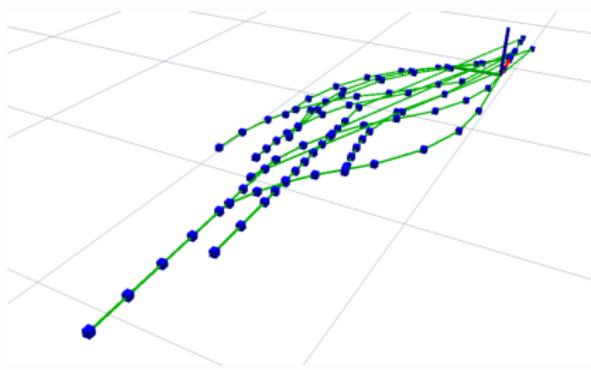
ROS e Gazebo

- A arquitetura de *software* da plataforma ATLASCAR2 é baseada na *framework* ROS.
- ROS
 - Desenvolvimento de software para robôs;
 - Arquitetura modular;
 - *Open Source* com grande comunidade ativa;
 - Possibilidade de programação em várias linguagens.
- Gazebo
 - Simulação de robôs com sensores incorporados em múltiplos ambientes;
 - Integração com ROS.



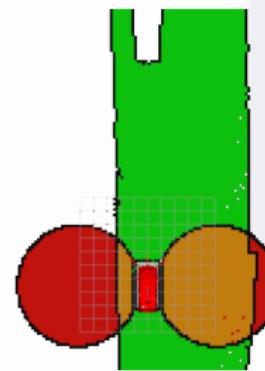
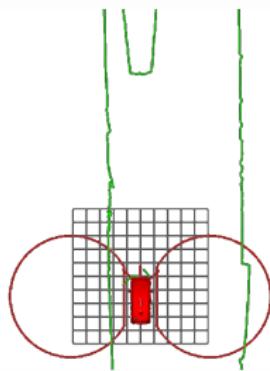
Trajectory planner package

- Aplica o algoritmo de planeamento por múltiplas hipóteses com trajetórias dedicadas à manobra de estacionamento.
- Existe ainda um simulador para obstáculos virtuais que permite a deteção de colisões.



Free space detection package

- Pacote ROS responsável pela fusão dos dados dos 3 LIDARs numa nuvem de pontos única.
- Permite ainda a representação do espaço livre para navegação de duas formas distintas:
 - Polígono de espaço livre;
 - Grelha de ocupação.



Solução de navegação local

1 Introdução

2 Revisão da literatura

3 Infraestrutura experimental

4 Solução de navegação local

- Solução implementada
- Geração de trajetórias

- Deteção de colisões
- Definição do ponto atrator
- Pontuação das trajetórias

5 Ambiente de simulação

6 Resultados

7 Conclusões e trabalhos futuros

Algoritmo selecionado

■ RRT e suas variantes:

- ✓ Consideração das restrições não holonómicas do veículo e rápida exploração do espaço de planeamento.
- ✗ Forte dependência da heurística de marcação aleatória dos pontos candidatos.

■ A* híbrido:

- ✓ Trajetórias executáveis por veículos não holonómicos e natureza ótima da trajetória traçada.
- ✗ Planeamento em grelhas com forte dependência da discretização do espaço de planeamento.

■ Planeamento por múltiplas hipóteses:

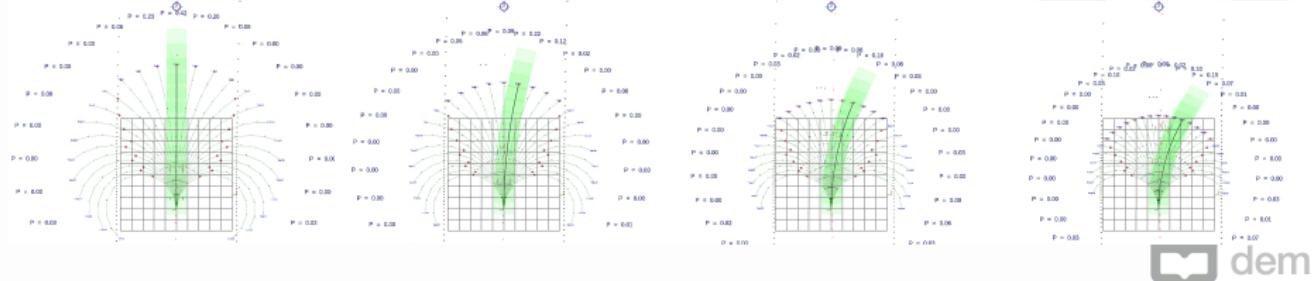
- ✗ Não executa o planeamento completo da trajetória.
- ✓ Capacidade de operação em tempo real.

Trajetórias

- A geração das trajetórias hipotéticas, em arco, é realizada de acordo com a velocidade de veículo, que varia linearmente com a direção.
- O comprimento mínimo das trajetórias é de 4,5 m para garantir a ausência de colisões e permitir inversão de marcha.

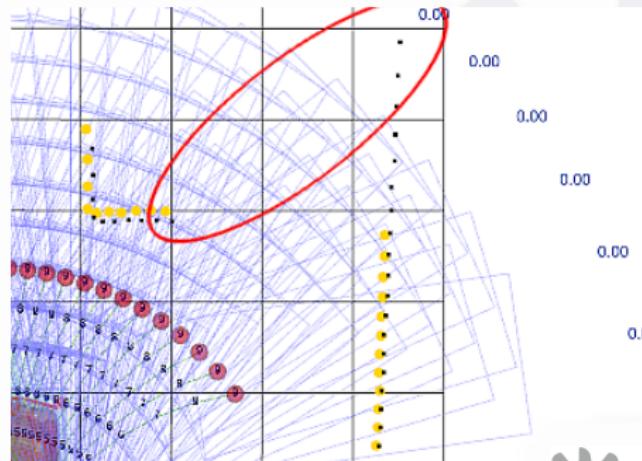
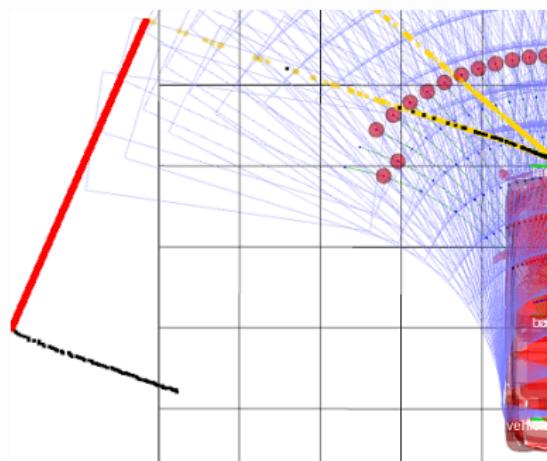
$$\left[\begin{array}{c} \prod_x \\ \prod_y \end{array} \right] = \left[\begin{array}{c} \frac{D}{\tan(\alpha)} \cdot \sin \left(\frac{A \cdot \tan(\alpha)}{D} \right) \\ \frac{D}{\tan(\alpha)} \cdot \left(1 - \cos \left(\frac{A \cdot \tan(\alpha)}{D} \right) \right) \end{array} \right]$$

$$D = \frac{v^2}{2 \cdot \mu \cdot g}$$



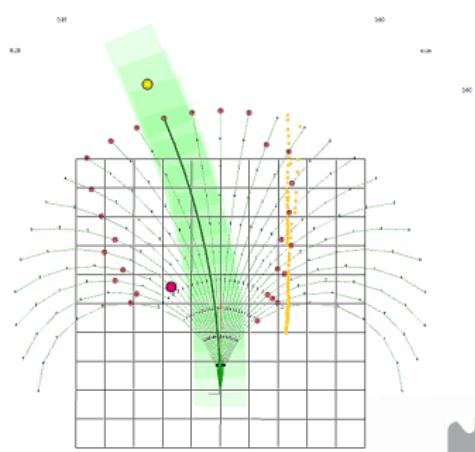
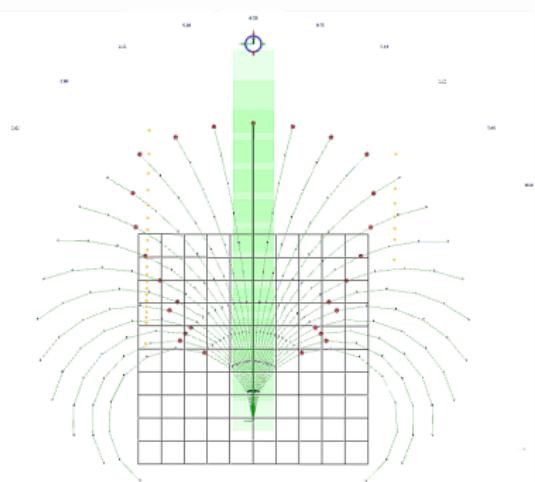
Colisões

- Inicialmente as colisões eram detetadas por interseções de linhas.
- De forma a eliminar o problema da criação de barreiras virtuais passaram a ser determinadas por inclusões poligonais, com recurso ao *winding number*.



Ponto atrator

- O planeamento necessita de um objetivo, neste caso o ponto atrator.
- Em simulação, o ponto atrator é definido pela posição de um marcador em Rviz.
- Em ambiente real, o ponto atrator depende dos *way points* seguinte e anterior.



Parâmetros da trajetória

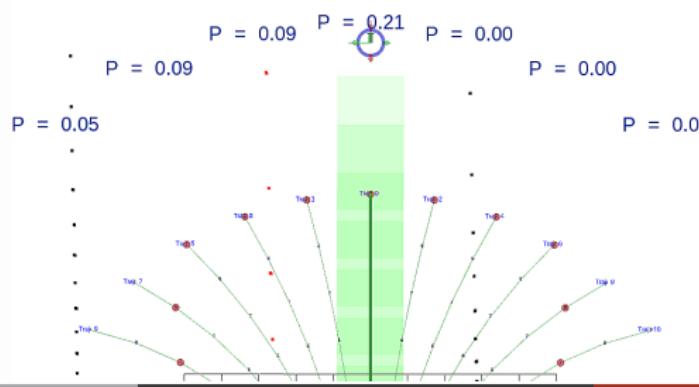
- Depois de detetadas as colisões e identificado o ponto atrator, cada trajetória é avaliada individualmente de acordo com os parâmetros:
 - FS — espaço livre;
 - CL — linha central;
 - DAP_{norm} — distância normalizada ao ponto atrator;
 - $ADAP_{norm}$ — distância normalizada angular ao ponto atrator;
 - DLO_{norm} — distância normalizada mínima aos obstáculos;
 - $m_{1,2,3}$ — pesos dos parâmetros de distância.

$$P_{traj} = (m_1 \cdot DAP_{norm} + m_2 \cdot ADAP_{norm} + m_3 \cdot DLO_{norm}) \cdot CL \cdot FS$$

- É selecionada a trajetória que apresentar maior pontuação.

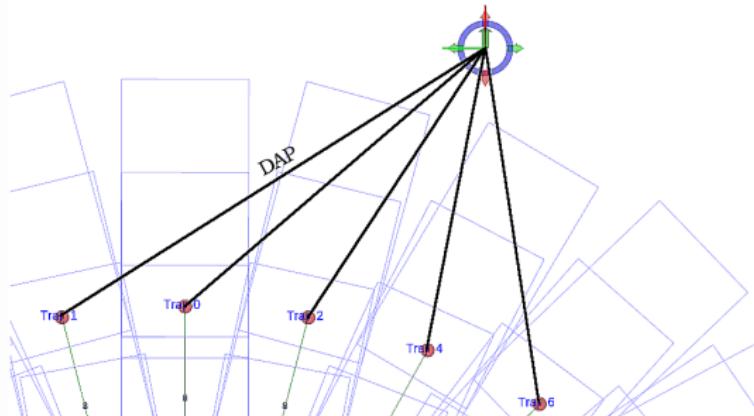
FS e CL

- O parâmetro FS é um parâmetro exclusivo, ou seja, elimina as trajetórias que incluem pontos da nuvem de obstáculos.
 - $FS = 1$ se não existe colisão ou $FS = 0$ se existe colisão.
- O parâmetro CL deve ser pesado consoante o tipo de linha que a trajetória hipotética intersetar.
 - $CL = 1$ quando não existe interseção ou $CL \in [0; 1[$ quando existe interseção.



DAP, ADAP e DLO

- DAP representa a distância linear do ponto mais próximo da trajetória ao ponto atrator.
- ADAP representa a distância angular entre ponto mais próximo da trajetória e o ponto atrator.
- DLO dá a distância linear do obstáculo mais próximo a qualquer ponto da trajetória.



Ambiente de simulação

1 Introdução

2 Revisão da literatura

3 Infraestrutura experimental

4 Solução de navegação local

5 Ambiente de simulação

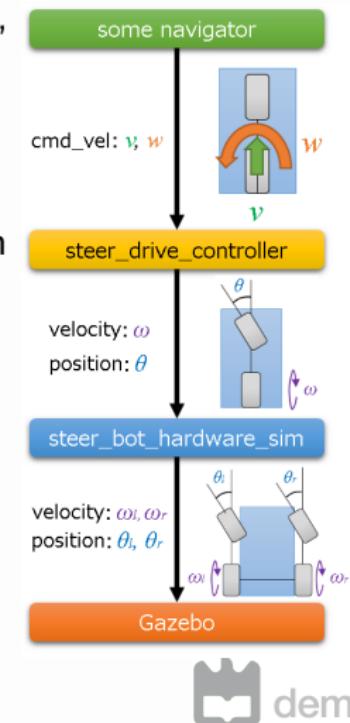
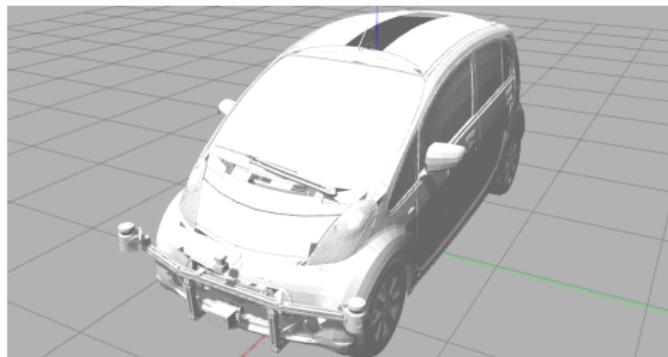
- Definição dos elementos
- Simulações realizadas
- Parâmetro de avaliação

6 Resultados

7 Conclusões e trabalhos futuros

Definição do modelo

- Adaptação do robô com cinemática de *Ackermann*, *CIR-KIT-Unit03*, ao modelo do ATLASCAR2.
- A velocidade e direção dadas pelo algoritmo são publicadas no tópico *cmd_vel*.
- A velocidade linear e direção são transformadas em velocidade de rotação e ângulos de direção.



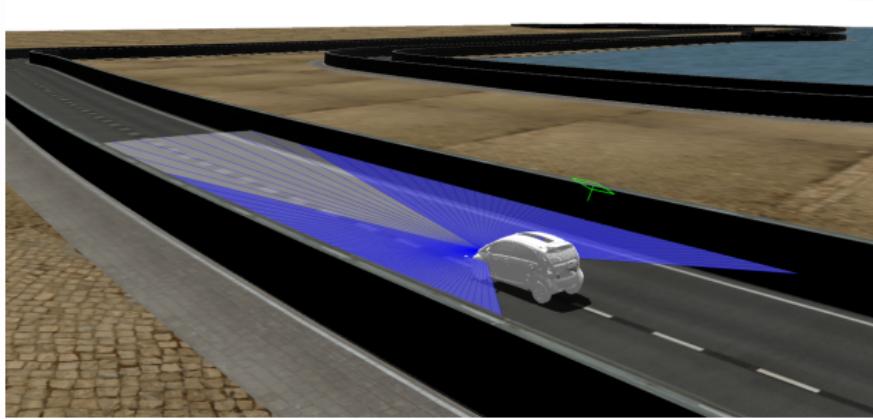
Definição de sensores

- Para a simulação dos sensores LIDAR foi utilizado o *plugin gazebo_ros_head_hokuyo_controller*.
- Foi utilizado um sensor na posição central com grande abertura de feixe e características semelhantes aos LIDARs 2D.

```
<gazebo reference="front_bottom_lrf">
    <sensor type="ray" name="front_bottom_lrf_hokuyo_sensor">
        <update_rate>40</update_rate>
        <ray>
            <scan>
                <horizontal>
                    <resolution>0.25</resolution>
                </horizontal>
            </scan>
            <range>
                <max>50.0</max>
            </range>
        <plugin name="gazebo_ros_head_hokuyo_controller" filename="libgazebo_ros_laser.so">
        </plugin>
    </sensor>
</gazebo>
```

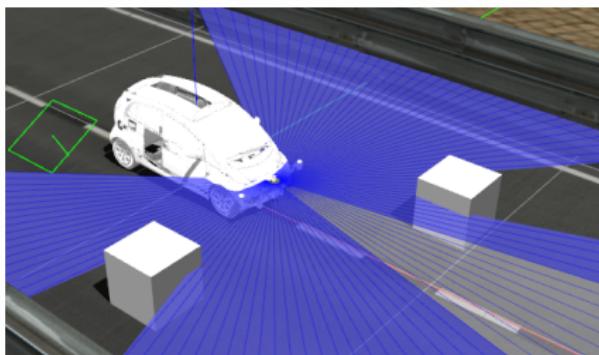
Simulações

- O ambiente de simulação utilizado assenta numa pista de Fórmula 1, cujos limites são definidos por muros verticais.
- As simulações realizadas podem ser enquadradas em dois tipos:
 - Simulações simples;
 - Simulações considerando a linha central.



Avaliação do algoritmo

- A avaliação do planeamento é realizada através da medição da distância mínima a que o modelo se encontra dos obstáculos.
- Esta informação é publicada no tópico *min_dist_to_obstacles* para poder ser analisada e comparada.



```
(...)
// avaliação da simulação
if (n==0 && trajectory->score.EVAL>DL0prev)
{
    trajectory->score.EVAL = DL0prev;
}
(...)
```

Resultados

1 Introdução

2 Revisão da literatura

3 Infraestrutura experimental

4 Solução de navegação local

5 Ambiente de simulação

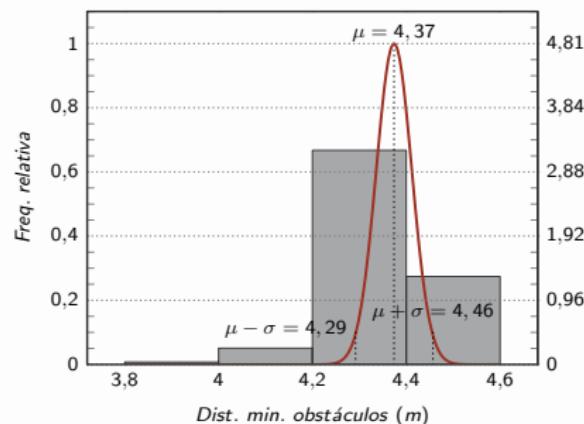
6 Resultados

- Simulações simples
- Simulações com linhas
- Testes reais

7 Conclusões e trabalhos futuros

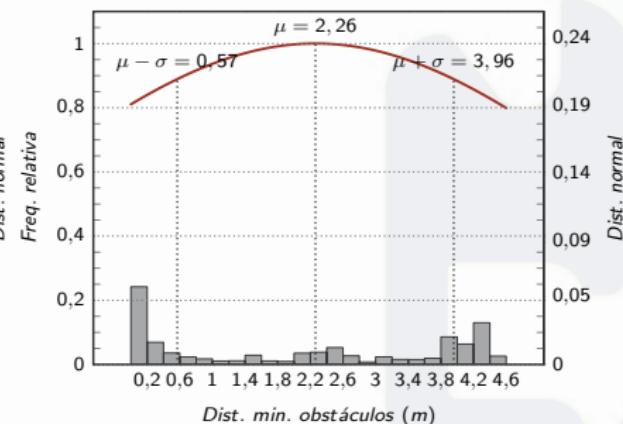
Testes padrão

- Simulações realizadas considerando apenas dados de obstáculos físicos, $v = 5 \text{ m/s}$ e $l = 3,24 \text{ m}$.
- A distância de referência é $4,45 \text{ m}$.



$$\{DAP = 0,1; ADAP = 0; DLO = 0,9\}$$

■ Mínimo = 3,83 m.

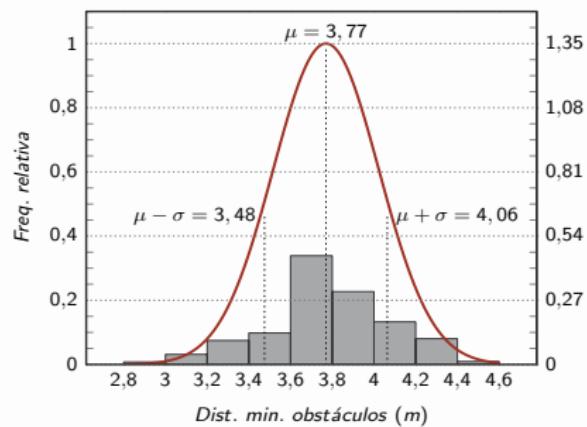


$$\{DAP = 0,9; ADAP = 0; DLO = 0,1\}$$

■ Mínimo = 0,01 m.

Trajetórias de comprimento aumentado

- Simulações realizadas considerando apenas dados de obstáculos físicos, $v = 5 \text{ m/s}$ e $l = 12,96 \text{ m}$, calculado para uma velocidade de 10 m/s .



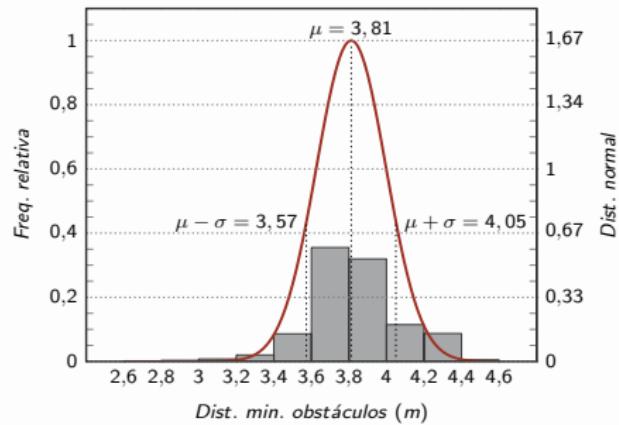
	Teste padrão	Trajetória aumentada
$\mu \text{ (m)}$	4,37	3,77
$\sigma \text{ (m)}$	0,08	0,30
Mínimo (m)	3,83	2,81

$$\{DAP = 0,1; ADAP = 0; DLO = 0,9\}$$

- Mínimo = 2,81 m.

Trajetórias de comprimento variável

- Simulações realizadas considerando apenas dados de obstáculos físicos e trajetórias de comprimento variável, $v = 1$ a 10 m/s e $l = 4,5$ a $12,96$ m.

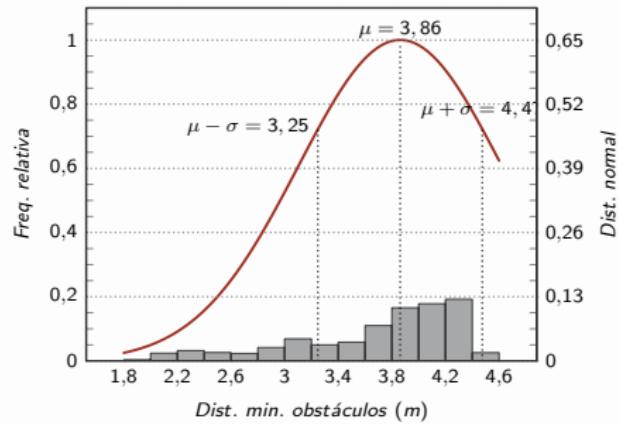


$$\{DAP = 0,1; ADAP = 0; DLO = 0,9\}$$

- Mínimo = 2,74 m.

Simulações considerando a linha central

- Simulações realizadas considerando dados de obstáculos físicos e da linha central ($CL = 0,2$), com trajetórias de comprimento variável.
- A distância máxima de referência é 3,57 m.

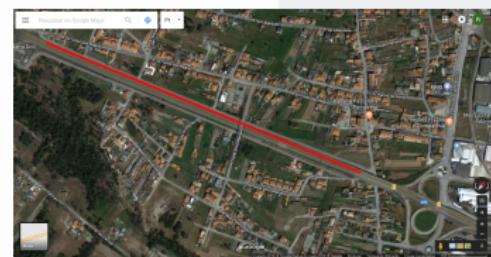
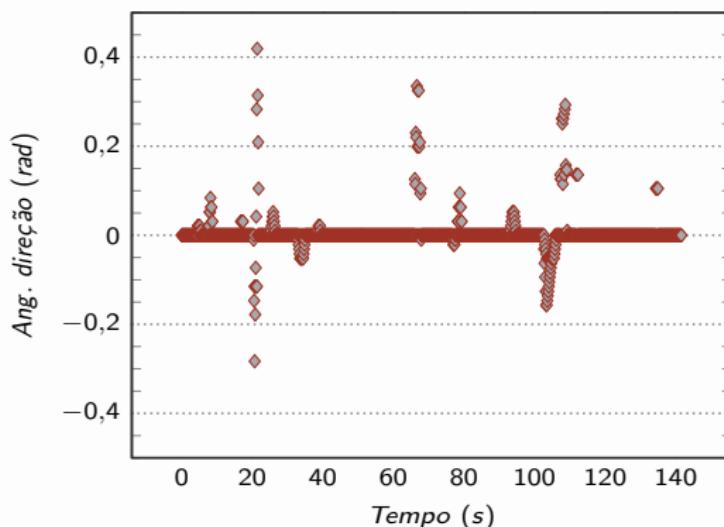


$$\{DAP = 0,1; ADAP = 0; DLO = 0,9\}$$

- $\text{Tempo}(d_{min} < 3,57) = 87,7\%$.

Avaliação em autoestrada

- Depois de realizada a integração com outros *packages*, responsáveis pelo planeamento global, foram realizados testes reais.
- Os resultados destes testes ficaram condicionados pela falta de informação das linhas da estrada e da centralina do carro.



Avaliação em autoestrada



Conclusões e trabalhos futuros

1 Introdução

2 Revisão da literatura

3 Infraestrutura experimental

4 Solução de navegação local

5 Ambiente de simulação

6 Resultados

7 Conclusões e trabalhos futuros

- Conclusões
- Trabalhos futuros

Resumo

- Pesquisa e seleção fundamentada de um algoritmo de navegação local e das estruturas de dados necessárias à sua operação:
 - ✓ Identificação e seleção das estruturas de dados;
 - ✓ Identificação e seleção dos algoritmos de navegação local.
- Integração do algoritmo de planeamento por múltiplas hipóteses na estrutura de *software* existente no ATLASCAR:
 - ✓ Implementação do algoritmo de navegação local;
 - ✓ Monitorização do algoritmo de navegação local;
 - ✓ Resolução das limitações encontradas na geração de trajetórias e deteção de colisões.
- Desenvolvimento de simulador no modelo do ATLASCAR2 em Gazebo e aplicações reais:
 - ✓ Teste do algoritmo de navegação local;
 - ✓ Integração com *software* de navegação global.

Conclusões

- O peso dos parâmetros de pontuação das trajetórias influencia a qualidade do planeamento. Quanto maior o parâmetro DLO mais seguro este se torna.
- Valores extremos do comprimento das hipotéticas trajetórias influenciam negativamente a qualidade do planeamento.
- A utilização de trajetórias de comprimento variável não se revelou tão benéfica quanto o esperado devido dispersão das mesmas com a velocidade.
- A realização de testes reais demonstrou a falta de informação complementar ao algoritmo e a necessidade da alteração da avaliação das trajetórias consoante o ambiente de planeamento.

Trabalhos futuros

- Avaliação do algoritmo em ambiente real:
 - Acesso a dados da centralina via OBD-II;
 - Atuação e controlo no ATLASCAR2.
- Deteção dos limites visuais da estrada e identificação segmentação de objetos.
- Teste de outros algoritmos em manobras específicas como ultrapassagens e cruzamentos com ou sem sinalização luminosa.



Navegação Local do ATLASCAR2 para Condução Autónoma e Assistência ao Condutor

Obrigado

Ricardo Silva

Aveiro, 26 de julho de 2018