

Departamento de Ciências e Tecnologias da Informação

**Modelação e Validação de Processos de *Car-Following* em Netlogo
para Simulação de Fluxo de Tráfego Rodoviário**

Bruno Daniel Cerdeira Rodrigues

Dissertação submetida como requisito parcial para obtenção do grau de:
Mestre em Engenharia Informática
Ramo de Sistemas de Informação e Gestão do Conhecimento

Orientador:
Prof. Doutor Nuno David
Instituto Superior de Ciências do Trabalho e da Empresa
Instituto Universitário de Lisboa

Outubro, 2011

"Bugs will appear in one part of a working program when another 'unrelated' part is modified"

Murphy Law

Resumo

O problema do tráfego rodoviário tem vindo a assumir uma crescente importância devido aos seus efeitos negativos na sociedade moderna, capazes de originar a diminuição da produtividade, da qualidade de vida e a degradação do meio ambiente. No domínio da simulação social, o investimento em simuladores baseados em agentes capazes de analisar os fenómenos de tráfego rodoviário tem aumentado. Uma das áreas fundamentais no estudo de modelos de tráfego é o fluxo gerado pela interacção dos veículos. Este processo designa-se *car-following*.

Com o objectivo de aprofundar a integração de algoritmos de *car-following* em plataformas e ambientes de simulação social, este trabalho analisa, estende e implementa dois algoritmos de *car-following* no ambiente Netlogo. Os dois algoritmos baseiam-se em abordagens distintas: um inspirado por sistemas multiagente, baseado em estados cognitivos e regras de decisão; e outro baseado na dinâmica de posição e da velocidade dos veículos. Os dois algoritmos são analisados comparativamente, destacando-se os seus pontos fortes, pontos fracos e a sua validade. Os resultados obtidos validam de diferentes maneiras o impacto dos comportamentos produzidos no fluxo de tráfego e sugerem que o algoritmo inspirado em sistemas multiagente aparenta produzir comportamentos mais realistas.

Palavras-Chave:

car-following, sistemas multi-agente, Netlogo, IDM.

Abstract

The problem of road traffic has been assuming a growing importance due to its negative effects on modern society, which can lead to lower productivity, declining of quality of life and environmental degradation. In the area of social simulation the investment in agent-based simulators, capable of analysing these phenomena, has been increased. One of the key areas in the study of traffic road models is the flow generated by the interaction of vehicles. This process is called car-following.

In order to deepen the integration of car-following algorithms across platforms and social simulation environments, this work analyses, extends and implements two car-following algorithms in Netlogo. These algorithms are based on different approaches: one inspired by multi-agent systems, based on decision-rules and cognitive states; and another based on the dynamics of position and vehicle speed. Both algorithms are analysed and compared, highlighting their strengths, weaknesses and its validity. The results authenticate the impact of behaviours produced in the flow of traffic in different ways and suggests that the algorithm based on multi-agent systems seems to produce more realistic behaviour.

Keywords:

car-following, multi-agent systems, Netlogo, IDM

Agradecimentos

Concluído o último passo desta desafiante etapa de formação escolar, gostaria de reconhecer a influência de quem me apoiou e ajudou a crescer. Sem esse fantástico apoio, o percurso teria sido muito mais difícil. Neste trabalho está um pouco de todos vós.

Desde já, um agradecimento especial ao meu orientador, Professor Doutor Nuno David pelo seu apoio, paciência, disponibilidade e orientação, mantendo-me sempre concentrado nos objectivos traçados e na construção desta dissertação.

Pela disponibilidade e auxílio prestado, um agradecimento à equipa da AutoMotor, em especial ao director António Pereira e ao jornalista Bruno Castanheira.

Aos meus colegas de licenciatura e mestrado no ISCTE-IUL com quem convivi nos últimos cinco anos, parceiros de festas e muitas horas de estudo, sempre com o espírito de entreatajuda presente.

À minha família, especialmente à minha mãe, e aos meus amigos quero agradecer toda a amizade, o apoio e a paciência que tiveram, sobretudo pelos momentos que tiveram de abdicar e nos dias em que nada me corria de feição.

Índice de Conteúdos

Resumo	i
<i>Abstract</i>	ii
Agradecimentos	iii
Índice de Conteúdos	iv
Índice de Tabelas.....	vii
Índice de Figuras	viii
Lista de Abreviaturas	x
CAPITULO 1 Introdução	1
1.1. Objectivos.....	3
1.2. Hipóteses	3
1.3. Estrutura da Dissertação.....	4
CAPITULO 2 Estado da Arte.....	5
2.1. Modelação e simulação de tráfego rodoviário.....	5
2.2. Abordagens e técnicas	6
2.2.1. Modelos macroscópicos.....	7
2.2.2. Modelos mesoscópicos	8
2.2.3. Modelos microscópicos.....	8
2.3. Modelos de <i>Car-Following</i>	10
2.3.1. Modelo <i>Gazis-Herman-Rothery</i> (GHR).....	11
2.3.2. Modelos de distância de segurança ou prevenção de colisões	11
2.3.3. Modelos lineares (ou Modelos de <i>Helly</i>)	12
2.3.4. Modelos de psicofísica ou de acção	12
2.3.5. Modelos de <i>Fuzzy-Logic</i>	12
2.4. Sinopse	13
CAPITULO 3 Modelação Multiagente de <i>Car-Following</i>	15
3.1. <i>VeniceTable: Interactive Traffic Simulation Table</i>	15
3.2. Modelo de Paulo Duarte (em curso).....	16
3.2.1. Ambiente e interligação SMA/SIG	17
3.2.2. Propriedades do agente veiculo-condutor.....	18

3.2.3.	Ciclo de vida do modelo	20
3.2.4.	Algoritmo de <i>car-following</i>	20
3.3.	Modelação do processo de <i>car-following</i>	22
3.3.1.	<i>Multi-Agent Systems for Simulating Traffic Behaviors</i>	22
3.3.2.	<i>Intelligent Driver Model</i>	26
3.4.	Modelo e ciclo de vida do agente veiculo-condutor	28
3.5.	Sinopse	29
CAPITULO 4	Implementação do Modelo	31
4.1.	Ferramentas de simulação	31
4.1.1.	O ambiente Netlogo	33
4.2.	Estrutura da implementação	35
4.2.1.	Unidades de tempo de ciclo de vida de uma simulação	36
4.2.2.	Construção do ambiente	36
4.2.3.	Modelo de agente	36
4.2.4.	Ciclo de vida dos agentes	38
4.2.5.	<i>Multi-Agent Systems for Simulating Traffic Behaviors</i>	39
4.2.6.	<i>Intelligent Driver Model</i>	39
4.2.7.	Geração e injeção de agentes no ambiente	39
4.3.	Interface gráfico	41
4.4.	Sinopse	43
CAPITULO 5	Testes e Análise de Resultados	45
5.1.	Especificação do percurso, parametrizações e abordagem para validação	45
5.1.1.	Parametrizações do modelo	46
5.1.2.	Validação de resultados	46
5.2.	Análise de resultados	47
5.2.1.	Fase 1: Análise ao comportamento isolado de um agente	47
5.2.2.	Fase 2: Comportamento dos agentes em fluxo de tráfego	59
5.2.3.	Análise macroscópica de tráfego	64
5.3.	Sinopse	72
CAPITULO 6	Conclusão	73
6.1.	Conclusões gerais	73

6.2. Contribuições.....	73
6.3. Limitações e trabalho futuro.....	74
Bibliografia.....	75
Anexos	81
Anexo A – Implementação em Netlogo do modelo de <i>car-following</i>	81
Anexo B – Notação utilizada para os fluxogramas.....	107

Índice de Tabelas

Tabela 1 - Características e evolução dos modelos.....	17
Tabela 2 - Framework resultante do modelo MAS/STB. Os comportamentos adicionados encontram-se sublinhados.	26
Tabela 3 - Comparação entre as quatro ferramentas de modelação genéricas e a ferramenta dedicada, adaptado de Castle, et al. (2006) e Gilbert (2008)	32
Tabela 4 - Parametrização do modelo e algoritmos	46
Tabela 5 – Primeira sequência de testes destinada ao comportamento de um único veículo ...	47
Tabela 6 - Condições iniciais para o 1º teste	48
Tabela 7 - Medições da capacidade de aceleração feitas pela AutoMotor.....	53
Tabela 8 - Condições iniciais para o 2º teste	54
Tabela 9 - Avaliação da distância de paragem segundo as medições efectuadas pela AutoMotor.....	57
Tabela 10 - Condições iniciais para o 3º teste.	58
Tabela 11 – Segunda sequência de testes de comportamento destinada às interações entre veículos	60
Tabela 12 - Condições iniciais do primeiro teste.....	60
Tabela 13 - Condições para o segundo teste da segunda fase.....	63
Tabela 14 – Terceira fase de testes dedicada à análise macroscópica de tráfego.....	65
Tabela 15 - Condições para os testes da terceira fase.	65

Índice de Figuras

Figura 1 - Exemplo de uma artéria congestionada da cidade de Lisboa.....	1
Figura 2 - A maioria dos acidentes rodoviários são causados por decisões incorrectas por parte dos condutores (Trigoso, 2011a; Trigoso, 2011b; Marques, 2011).....	3
Figura 3 - Modelo macroscópico de tráfego rodoviário. As cores representam a densidade de tráfego existente na via.....	7
Figura 4 - Representação de um modelo rodoviário de Autómato Celular.....	8
Figura 5 - Modelo multiagente de tráfego rodoviário, representando um ambiente com alto nível de realismo.....	10
Figura 6 - Imagem do projecto VeniceTable a correr em Netlogo.....	16
Figura 7 - Camadas existentes no modelo multiagente e a interacção com componente SIG ..	17
Figura 8 - Mapa das características do agente veiculo-condutor, sugeridas por Kesting, et al. (2008).....	19
Figura 9 - Propriedades do modelo de agente de Paulo Duarte (em curso).....	20
Figura 10 - Ciclo de Vida do Modelo de Paulo Duarte (em curso).....	20
Figura 11 - Fluxograma referente ao algoritmo de car-following.....	21
Figura 12 - Exemplos dos resultados comportamentais dos agentes.....	21
Figura 13 - Exemplo de estímulos recebidos pelo agente.....	22
Figura 14 - Demonstração dos níveis de busca pelo agente.....	23
Figura 15 - Demonstração dos estímulos recebidos. A distância S_{α} é definida por $x_{\alpha-1} - x_{\alpha} - l_{\alpha-1}$	27
Figura 16 - Novo modelo de Agente.....	28
Figura 17 - Ciclo de vida do agente.....	29
Figura 18 - Exemplo do interface Netlogo, <i>Traffic Grid Model</i> (Wilensky, 2003).....	33
Figura 19 - Imagem de uma <i>turtle</i> (roxo) a circular no <i>link</i> (ligação cinzenta). O <i>patch</i> está presente a branco, em segundo plano.....	35
Figura 20 - Representação gráfica dos <i>nós</i> (bolas coloridas), <i>links</i> (ligações entre <i>nós</i>) e <i>patches</i> (fundo da imagem).....	35
Figura 21 - Exemplo das cinco fases de implementação definidas.....	35
Figura 22 - Ciclo de vida dos processos referentes aos SMA.....	36
Figura 23 - Subprocessos utilizados na construção do ambiente.....	36
Figura 24 - Comparação entre modelo de agente definido na modelação e a estrutura de dados de implementação.....	37
Figura 25 - Fluxograma relativo ao processo de Recolha de Informação.....	38
Figura 26 - Fluxograma referente ao processo de Análise e Decisão.....	38
Figura 27 - Fluxograma relativo ao processo de realização da Acção de movimento.....	38
Figura 28 – Sequência de processos dedicados à geração e injeção de agentes.....	40
Figura 29 - Novo interface gráfico.....	41

Figura 30 - Secção esquerda do interface em detalhe	42
Figura 31 - Topo direito do ecrã em detalhe	42
Figura 32 - Percurso rodoviário simulado. Os veículos circulam à direita do <i>link</i> e no sentido contra horário.....	45
Figura 33 - Tempo que um agente isolado demora a percorrer o percurso	48
Figura 34 - Velocidade instantânea de um veículo isolado.....	49
Figura 35 – Aceleração instantânea de um veiculo isolado	49
Figura 36 - Velocidade instantânea de um veículo isolado em detalhe.....	51
Figura 37 - Velocidade instantânea atingida pelo modelo Rakha, et al. (2001) e validado por Snare (2002) em aceleração livre.....	51
Figura 38 - Aceleração instantânea de um veículo em detalhe	52
Figura 39 - Aceleração instantânea atingida pelo modelo Rakha, et al. (2001) e validado por Snare (2002) em aceleração livre.....	52
Figura 40 - Velocidade dos agentes em travagem para $V_{Max} = 90$ km/h.	55
Figura 41 - Aceleração dos agentes em travagem para $V_{Max} = 90$ km/h.	55
Figura 42 - Desaceleração do agente em função da velocidade e da distância percorrida	56
Figura 43 – Velocidade dos agentes quando $V_{Inicial} > V_{Max}$ ($V_{Inicial} = 90$ km/h, $V_{Max} = 50$ km/h)...	58
Figura 44 – Aceleração dos agentes quando $V_{Inicial} > V_{Max}$ ($V_{Inicial} = 90$ km/h, $V_{Max} = 50$ km/h)...	59
Figura 45 - Velocidade instantânea atingida pelos pares de veículos utilizando o MAS/STB....	61
Figura 46 – Distância mútua entre os pares de veículos para o MAS/STB.....	61
Figura 47 - Velocidade instantânea atingida pelos pares de veículos utilizando o IDM.....	62
Figura 48 - Distância mútua entre veículos pares para o IDM.	62
Figura 49 - Desaceleração dos pares de agentes face a um obstaculo.....	64
Figura 50 - Distância relativa entre algoritmos	64
Figura 51 - Velocidade-Densidade para MAS/STB com taxa de injeccção de 25%.	66
Figura 52 - Velocidade-Densidade para IDM com taxa de injeccção de 25%.....	66
Figura 53 - Velocidade-Densidade para MAS/STB com taxa de injeccção de 100%.	67
Figura 54 - Velocidade-Densidade para IDM com taxa de injeccção de 100%.....	67
Figura 55 - Fluxo-Densidade para MAS/STB com taxa de injeccção de 25%.....	68
Figura 56 - Fluxo-Densidade para IDM com taxa de injeccção de 25%.	68
Figura 57 - Fluxo-Densidade para MAS/STB com taxa de injeccção de 100%.....	69
Figura 58 - Fluxo-Densidade para IDM com taxa de injeccção de 100%.....	69
Figura 59 - Velocidade-Fluxo para MAS/STB com taxa de injeccção de 25%.	70
Figura 60 - Velocidade-Fluxo para IDM com taxa de injeccção de 25%.....	70
Figura 61 - Velocidade-Fluxo para MAS/STB com taxa de injeccção de 100%.	71
Figura 62 - Velocidade-Fluxo para IDM com taxa de injeccção de 100%.....	71

Lista de Abreviaturas

Abreviatura	Significado
SMA	Sistemas Multiagente
SIG	Sistemas de Informação Geográfica
GHR	Modelo <i>Gazis-Herman-Rothery</i>
MAS/STB	<i>Multi-Agent Systems for Simulating Traffic Behaviors</i>
IDM	<i>Intelligent Driver Model</i>
MASON	<i>Multi-Agent Simulator of Neighborhoods</i>
REPAST	<i>Recursive Porous Agent Simulation Toolkit</i>
SUMO	<i>Simulation of Urban Mobility</i>
cv	<i>cavalos (medida referente à potência do motor do automóvel)</i>

CAPITULO 1

Introdução

A questão do tráfego rodoviário tem vindo a assumir uma crescente importância por diversas razões. As sociedades industrializadas exigem uma rede eficiente de transportes para o seu bom funcionamento e desenvolvimento. Para o cidadão, a mobilidade oferecida através da rede de transportes é algo considerado como parte integrante da sua qualidade de vida, do seu sentido de auto-realização e da sua liberdade individual (Kesting, et al., 2008).

Mas com o aumento da densidade populacional, do poder de compra e da qualidade de vida aumentou também o volume de trânsito nas vias, em boa parte originado pela circulação de veículos transportando apenas uma pessoa. Devido às suas limitações estruturais, as vias começaram a não ter capacidade para escoar o excesso de volume de tráfego originando fenómenos como o congestionamento, ilustrado na Figura 1.



Figura 1 - Exemplo de uma artéria congestionada da cidade de Lisboa

Por outro lado, com o aumento de volume de trânsito, aumentaram também o número de acidentes e a necessidade de obras de reparação das vias, reduzindo consideravelmente a sua capacidade de escoamento. Este fenómeno origina diversos problemas, tais como o aumento da frequência de engarrafamentos, atrasos nas deslocações, perda de produtividade, aumento do *stress* e diminuição do bem-estar. Em paralelo, a qualidade de vida nos centros urbanos diminui, devido à degradação do meio ambiente e da saúde pública, provocado pela poluição sonora e pelas emissões dos veículos, podendo atingir valores similares aos de uma fábrica industrial.

Neste contexto, têm sido postas em prática medidas relacionadas com a construção e/ou modificações físicas da infra-estrutura, criando alternativas e aumentando a capacidade de

escoamento. Mas tal tem-se revelado insuficiente já que, em vários casos, a falta de espaço disponível, as exigências sociais e ambientais e os altos investimentos na construção e posterior manutenção têm mostrado que esta não é uma solução duradoura. Actualmente procura-se equilibrar três aspectos através da implementação de políticas: o desejo de mobilidade pessoal, as preocupações da sociedade com o impacto ambiental e o consumo energético.

Dado o conflito existente entre os recursos disponíveis, as preocupações ecológicas e o aumento do tráfego rodoviário, começou-se a despender mais atenção ao estudo do fluxo de tráfego. Nesse contexto, tem-se investido em simuladores capazes de analisar e facilitar a sua compreensão. Com a sua utilização, consegue-se otimizar o controlo de tráfego de forma mais eficiente, estudar os efeitos de implementação de políticas difíceis de prever e avaliar o impacto de cenários hipotéticos no tráfego, tudo isto de forma economicamente mais rentável. Com efeito, a utilização de simuladores de tráfego tem-se mostrado cada vez mais expressiva, atingindo hoje imensos domínios, tais como (Champion, et al., 1999):

- O *design* e a melhoria de equipamentos automobilísticos – o poder de simulação das ferramentas permite poupar tempo e dinheiro, testando os equipamentos e validando-os, antes do investimento necessário para a sua construção e aplicação;
- A formação – as simulações em tempo real permitem produzir variados cenários para a formação de pessoal de centros de controlo de tráfego ou condutores de veículos.
- A avaliação de novas estruturas (estradas, sinalização, portagens, etc) – a construção de uma infra-estrutura é dispendiosa. Com a simulação é possível quantificar a eficiência do funcionamento de uma nova estrutura antes da sua construção.
- A segurança e o meio ambiente – algumas aplicações neste domínio envolvem estudos de auto-estradas e veículos inteligentes, análise de comportamentos do condutor em situação de perigo, reconstrução de acidentes e avaliações ao nível de emissões de gases dos veículos.
- As investigações do fluxo de tráfego – com o objectivo de melhorar o fluxo de tráfego através de estudos matemáticos e estatísticos recolhidos em simulações.

Mas apesar da importância deste assunto, a literatura existente ainda não apresentou um modelo abrangente capaz de capturar, satisfatoriamente, a riqueza e complexidade do tráfego real (Benjaafar, et al., 1997). Entendemos ainda muito pouco do que está envolvido na “tarefa da condução” que é, para além de andar, comunicar e comer, a actividade mais executada em todo o mundo (Rothery, 1999). São os comportamentos individuais dos condutores que originam muitos dos problemas que se reflectem no fluxo de tráfego, exemplificado na Figura 2. Ao estudo dos processos pelos quais os condutores se seguem uns aos outros no fluxo de tráfego chamamos *car-following*.



Figura 2 - A maioria dos acidentes rodoviários são causados por decisões incorrectas por parte dos condutores (Trigoso, 2011a; Trigoso, 2011b; Marques, 2011).

1.1. Objectivos

O âmbito desta dissertação centra-se na modelação e implementação de algoritmos de “*car-following*”. Pretende-se auxiliar a consolidação de um modelo de simulação através da utilização do ambiente Netlogo e de Sistemas de Informação Geográfica (SIG). São os seguintes objectivos a atingir:

- Estudar e consolidar o ambiente de simulação de tráfego rodoviário em Netlogo com ligação a SIG, originalmente proposto por Fabio Carrera e estendido por Paulo Duarte (em curso);
- Especificar e implementar no simulador dois algoritmos de *car-following*, adicionando-lhes ainda especificações adicionais para o seu adequado funcionamento. Os algoritmos são baseados em duas abordagens distintas: uma orientada a regras e outra orientada a equações dinâmicas;
- Especificar e implementar um modelo de agente capaz de integrar as funcionalidades introduzidas pelos dois algoritmos;
- Avaliar o ambiente e efectuar a validação do realismo e desempenho dos dois algoritmos de *car-following*;

1.2. Hipóteses

No âmbito dos objectivos traçados, são as seguintes hipóteses deste trabalho:

- H1 – O Netlogo permite a construção de modelos de simulação relativamente complexos com ligação a SIG. Nesse sentido, possui as necessárias para a construção de uma plataforma ou ambiente de simulação de tráfego rodoviário capaz de integrar

vários modelos de *car-following*. Espera-se desenvolver uma plataforma que beneficie das vantagens da utilização de SIG em ambientes de tráfego rodoviário;

- H2 – A implementação de dois algoritmos de *car-following* com abordagens distintas de modelação resultará em resultados quantitativa e qualitativamente diferentes. Um dos algoritmos está é inspirado em sistemas multiagente, baseado em estados cognitivos e regras. O outro baseia-se numa única equação que modela a dinâmica de posições e da velocidade dos veículos. Espera-se que ambos os algoritmos mostrem realismo satisfatório na replicação dos comportamentos de *car-following*. Espera-se também que o algoritmo baseado em regras apresente resultados que, do ponto de vista qualitativo, poderão ser considerados realistas;

1.3. Estrutura da Dissertação

Face aos objectivos desta dissertação, a estrutura deste trabalho segue a seguinte sequência: o Capítulo 2 apresenta o estado de arte referente às abordagens e técnicas utilizadas em modelos de simulação de fluxo de tráfego e modelos de *car-following*. O Capítulo 3 apresenta em algum detalhe os dois modelos de *car-following* que servem de base a esta dissertação, incluindo os seus algoritmos, as extensões consideradas pertinentes para o seu funcionamento, o modelo de agente utilizado e o ciclo de vida da simulação. No Capítulo 4 descrevem-se os principais aspectos da implementação dos algoritmos e do modelo de agente. Finalmente no Capítulo 5, relata-se os testes efectuados e os resultados obtidos. O Capítulo 6 apresenta as conclusões e recomendações para o futuro.

CAPITULO 2

Estado da Arte

Este capítulo enquadra os objectivos desta dissertação com a literatura existente. São explicados os conceitos de modelação e simulação e a sua relação com o estudo do tráfego rodoviário. São apresentadas as abordagens e metodologias mais comuns e habitualmente aplicadas em simuladores com este contexto. São também ilustradas diversas perspectivas distintas para modelação de *car-following*, com o objectivo de demonstrar que, para além da abordagem multiagente utilizada, existe uma enorme variedade de outras abordagens.

2.1. Modelação e simulação de tráfego rodoviário

A modelação e a simulação computacional de processos têm uma enorme importância nas ciências e engenharias, facilitando a compreensão de sistemas complexos e a sua dinâmica. A mente humana é um bom exemplo de um sistema deste género. Possui a capacidade de modelar o mundo à nossa volta, simulando possíveis acontecimentos ou rotinas do nosso dia-a-dia, experimentando diferentes abordagens e avaliando os possíveis resultados. Mas em termos de processos complexos e dinâmicos tem dificuldades em avaliar de forma eficaz, apanhando-nos muitas vezes desprevenidos o que nos leva a ter, por vezes, respostas inadequadas aos acontecimentos em causa.

Foi esta motivação que levou ao desenvolvimento de modelos computacionais para simulação dos processos dinâmicos, capazes de compreender a dinâmica de sistemas complexos, de lidar com uma grande quantidade de dados e relações complexas e com os seus resultados. Devido à sua capacidade de processamento e a alguma precisão, obtêm resultados de uma maneira mais eficaz que a mente humana. Este desenvolvimento levou ao aumento de novas oportunidades em praticamente todos os domínios da experiência humana (Bossel, 1994).

De forma a entender as ideias dos conceitos de modelação e simulação é necessário perceber os componentes que os formam. Temos então:

- O Sistema – existe e opera no tempo e espaço;
- O Modelo – é uma representação simplificada de um sistema em algum ponto específico do tempo e espaço, destinado a promover a compreensão do sistema real;
- A Simulação – é a manipulação de um modelo que permite que sejam perceptíveis as interacções existentes;

Um sistema é entendido como uma entidade que mantém a sua existência através da interacção das suas partes. Um modelo, por sua vez, é a representação simplificada do sistema real a que se destina promover a compreensão. Se um modelo possuir detalhe reduzido, é provável que os resultados não facilitem o seu entendimento decorrente da sua

falta de realismo. Por outro lado, caso tenha demasiados detalhes, a sua complexidade excessiva pode complicar e impedir a compreensão do modelo. A simulação é a vertente computadorizada, ou seja, é a execução das instruções especificadas pelo modelo ao longo do tempo para estudar as implicações das interacções definidas (Bellinger, 2004).

Neste contexto, o nível de importância da modelação de simulações rodoviárias tem crescido nos últimos tempos, devido à necessidade de entender os vários fenómenos existentes no tráfego. Potencia-se a optimização do controlo de tráfego de uma forma eficaz, através da criação de cenários reais ou hipotéticos. Consegue-se, igualmente, produzir análises de resultados com um custo efectivo mais reduzido, tendo em conta a especificidade e a imprevisibilidade do funcionamento do tráfego rodoviário (Kesting, et al., 2008).

As primeiras pesquisas sobre a física dos engarrafamentos e fluxo de tráfego (áreas na simulação de tráfego rodoviário) começaram há cerca de meio século (Lighthill & Whitham, 1955) e têm continuado, utilizando técnicas e ferramentas cada vez mais avançadas. A maioria dos estudos lida com o aspecto “mecanizado” do tráfego (desde a densidade de tráfego nas vias até à afinação do tempo dos sinais luminosos) e na optimização destes processos. Uma das disciplinas que mais tem conquistado a atenção, recentemente, é o processo de *car-following*. Trata-se do estudo comportamental dos condutores com base na reacção destes a estímulos provenientes do veículo da frente.

2.2. Abordagens e técnicas

Muitos físicos e matemáticos começaram a direccionar a sua atenção para a dinâmica do tráfego. Foi com a convicção de que as suas áreas de conhecimento poderiam contribuir para a revelação e explicação dos efeitos existentes no trânsito, que o desenvolvimento das técnicas e dos modelos começou a progredir rapidamente. Actualmente podem identificar-se mais de 100 modelos diferentes, sugeridos por vários autores. Apesar das diferenças, existem pontos em comum que nos permitem englobá-los em determinadas abordagens metodológicas tais como:

- Os modelos microscópicos;
- Os modelos mesoscópicos;
- Os modelos macroscópicos;

Tendo como referencia a evolução temporal, desde os primeiros modelos até aos mais actuais, podemos afirmar que cada década teve a sua abordagem metodológica dominante. Nos anos 50, os primeiros modelos de tráfego rodoviário que tentavam explicar os saltos de densidade do tráfego eram descritos por um modelo de dinâmica de fluidos para ondas cinemáticas. Na década seguinte (anos 60), iniciou-se em grande escala o estudo do comportamento de *car-following*, utilizando os modelos de micro-simulação. Na década de 70 apareceram os modelos de dinâmica de gases que explicavam a mudança espaço-temporal da velocidade. Nos anos 80 reinou a simulação macroscópica e de dinâmica de fluidos e os anos 90 foram dominados

pela utilização de autómatos celulares no estudo do tráfego de veículos e pela investigação das soluções desenvolvidas nos últimos 50 anos.

Helbing (2001) definiu critérios para a avaliação dos melhores modelos de tráfego e considerou que eles devem conter variáveis e parâmetros intuitivos para uma correcta calibração e solidez, com resultados realistas que facilitem a medição. Para além disso, um bom modelo deve conseguir reproduzir com alguma qualidade todas as características do fluxo de tráfego, desde o congestionamento até a dinâmicas complexas como velocidade de propagação, movimentos de onda no trânsito ou o escoamento dos engarrafamentos.

2.2.1. Modelos macroscópicos

Os modelos macroscópicos foram definidos para olhar para o fluxo de tráfego numa maneira global, concentrando a sua utilidade na descrição do fluxo de veículos em termos de densidade e velocidade média na estrada devido à sua eficiência numérica, como podemos observar pela Figura 3. Outras vantagens na sua utilização centram-se na afinidade com os dados empíricos e com a investigação analítica, e a possibilidade de simular a dinâmica de tráfego em vias com várias faixas. A sua simulação é segmentada, isto é, é feito à zona ao contrário dos modelos microscópicos que fazem o rastreamento individual de veículos. Originalmente foram desenhados para modelar e simular o tráfego em determinadas sub-redes de transporte tais como auto-estradas, ruas e avenidas, e estradas rurais. O trabalho de Helbing (1995) é um exemplo neste campo, ao construir um modelo que identifica as relações existentes entre o diagrama fundamental e as equações macroscópicas de tráfego.

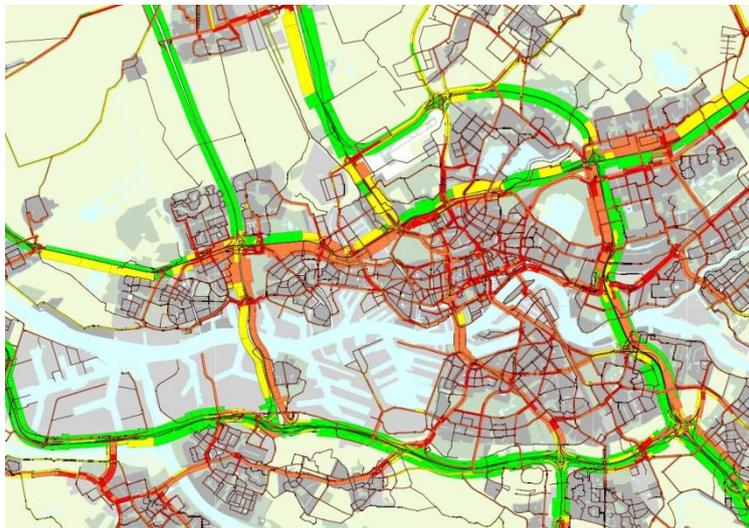


Figura 3 - Modelo macroscópico de tráfego rodoviário. As cores representam a densidade de tráfego existente na via

2.2.2. Modelos mesoscópicos

Estes modelos combinam muitas das características dos modelos microscópicos e macroscópicos, produzindo resultados menos consistentes que os primeiros mas com uma análise superior aos últimos. Conseguem simular comportamentos individuais dos agentes mas descrevem actividades e interacções de conjunto, sendo mais adequados para efeitos de controlo de comportamentos.

2.2.3. Modelos microscópicos

Os modelos microscópicos centram-se nos detalhes do fluxo de tráfego e nas interacções que ocorrem entre veículos e entre a estrada e o veículo. A maioria dos modelos dá primazia às características dos veículos-condutores de modo a personalizar os resultados das interacções. Este tipo de modelo é utilizado maioritariamente na modelação do *car-following* tentando reproduzir realisticamente comportamentos humanos normais, ignorando alguns factores que o tornam incerto. Sendo este tipo de abordagem o mais adequado ao desenvolvimento dos objectivos desta dissertação, irei apresentar dois tipos de modelos potencialmente úteis para os objectivos traçados: Autómatos Celulares e os Sistemas Multiagentes.

Autómatos Celulares

A aplicabilidade dos autómatos celulares tem estimulado a investigação na compreensão e controlo das instabilidades do tráfego, em boa parte devido à sua velocidade de processamento e eficiência, consequência das suas características. São conceptualmente simples mas conseguem produzir comportamentos complexos. O espaço é representado de forma discreta utilizando células (Figura 4), existindo um número finito de estados e com a actualização temporal a ser feita de forma paralela. Mas a rápida velocidade de processamento é ganha à custa da forma menos detalhada na representação dos veículos, exibindo comportamentos pouco realistas. Podemos tomar como exemplo a aceleração. Como a movimentação é feita de forma discreta deparamo-nos exclusivamente com acelerações e desacelerações instantâneas.

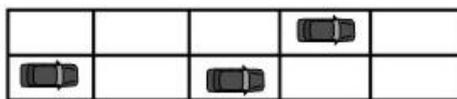


Figura 4 - Representação de um modelo rodoviário de Autómato Celular

Muitos autores defendem a utilidade dos autómatos celulares na captura e relacionamento das dinâmicas microscópicas com os comportamentos macroscópicos. Podemos tomar como exemplo três modelos que estudam aspectos similares: a Regra 184 (Krug, et al., 1988), o

modelo *Nagel-Schreckenberg* (Nagel, et al., 1992) e o modelo *Biham-Middleton-Levine* (Biham, et al., 1992).

O nome Regra 184 provém do código de evolução de estados definido pela *Wolfram*¹. Trata-se de um modelo primitivo, conhecido por resolver e descrever a maioria dos problemas das simulações de partículas e que ganhou alguma notoriedade como modelo de fluxo de tráfego. Através dele é possível prever alguns dos fenómenos usuais de trânsito como a fácil circulação de veículos quando a densidade é reduzida ou o efeito de onda de paragem aquando de uma elevada densidade de veículos. Por sua vez o modelo *Nagel-Schreckenberg*, desenvolvido no início dos anos 90 é um modelo teórico para simulações de tráfego. Foi o primeiro modelo que levou em conta o comportamento imperfeito de condução humana e a conseguir explicar as formações espontâneas de trânsito. Demonstrou que formações como o congestionamento provinham das interacções entre os veículos em determinadas condições, tais como a elevada densidade de veículos na estrada e a sua elevada proximidade. O modelo *Biham-Middleton-Levine*, à semelhança dos anteriores, demonstra a auto-organização do tráfego e os comportamentos quando a densidade aumenta ou diminui.

Sistemas Multiagente (SMA)

A utilização de sistemas multiagentes tem sido cada vez mais frequente. Ao oferecerem grandes potencialidades no estudo de sistemas sociais em domínios multidisciplinares, e permitirem a simulação de fenómenos complexos que não podem ser facilmente descritos analiticamente, tornaram-se um paradigma muito apropriado na modelação de tráfego rodoviário. A sua capacidade de executar simulações discretas no tempo, no espaço e de modelar dinâmica e individualmente os objectos são requisitos importantes para obter simulações que representem de forma muito expressiva o mundo real, tal como ilustrado na Figura 5.

Os sistemas multiagente, como o nome indica, são compostos por entidades autónomas, com um grande nível de interacção e com alguma inteligência. Sendo um paradigma promissor para a construção de sistemas complexos, seria natural pensar que estes modelos fossem rapidamente adoptados e utilizados. Contudo, tal nem sempre se verifica dada as dificuldades provocadas pela inexistência de um modelo universal de desenvolvimento e a propensão a erros (Braubach, et al., 2005). O seu desenvolvimento, normalmente, não se rege por nenhuma metodologia específica e o suporte dado pelas ferramentas de implementação costuma ser limitado, não abrangendo todas as fases do processo.

¹ Vide definição de “Regra 184”, <http://atlas.wolfram.com/01/01/184/>

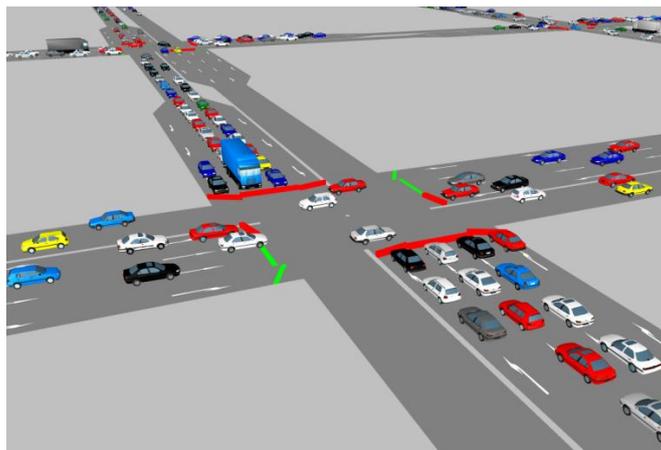


Figura 5 - Modelo multiagente de tráfego rodoviário, representando um ambiente com alto nível de realismo.

Algumas simulações multiagente são utilizadas em estudos de ergonomia e engenharia avançada, na pesquisa de comportamentos no tráfego, e factores humanos (Champion, et al., 1999). O centro de pesquisa da Renault é um exemplo na utilização deste tipo de recursos. Por sua vez, Lansdowne (2006) apresenta um modelo que facilita o estudo do tráfego e faz uma avaliação de infra-estruturas rodoviárias defendendo que este tipo de simulações merece a atenção e o investimento associado por parte dos investigadores.

Com efeito, comparativamente aos autómatos celulares, os sistemas multiagente mostram-se mais adequados aos objectivos desta dissertação. Descrevem os fenómenos emergentes do tráfego com uma boa flexibilidade, principalmente em relação aos modelos geo-espaciais, e produzem resultados mais realistas.

2.3. Modelos de *Car-Following*

Como já foi referido, o *car-following* é um modelo específico de micro-simulação que procura reproduzir o comportamento real dos condutores dos veículos numa via de trânsito com uma faixa de rodagem, reagindo às acções do veículo da frente. A sua origem deve-se, em grande parte, ao estudo da análise de estabilidade de condução em relação às alterações de velocidade. Hoje o *car-following* já possui aplicações práticas no desenvolvimento de sistemas de *cruise-control* e Sistemas Avançados de Controlo do Veículo e Segurança². Estes procuram replicar a condução humana através do controlo da aceleração, mas ignorando os perigos potenciais que poderão ocorrer devido a equívocos ou ao tempo de reacção. A chave deste processo reside em alcançar um algoritmo capaz de reproduzir, realisticamente, o comportamento humano de aceleração e a análise da distância de segurança, baseado na velocidade actual do veículo e nos obstáculos à sua frente.

² Vide "*Advanced Vehicle Control and Safety Systems*", http://kerncog.org/cms/index.php?option=com_content&view=article&id=215&Itemid=305

Nos últimos cinquenta anos foram apresentados vários modelos de *car-following*, todos com bases conceptuais apoiadas em dados empíricos. Apesar de existirem muitas opiniões favoráveis à validade dos modelos para os propósitos a que se destinam, é consensual que existe ainda um longo caminho a percorrer (Brackstone, et al., 1999). Seguidamente apresenta-se alguns modelos sugeridos numa revisão histórica realizada por Brackstone, et al. (1999).

2.3.1. Modelo Gazis-Herman-Rothery (GHR)

O modelo GHR foi um dos primeiros a ser desenvolvido nos finais dos anos 50, baseando-se na ideia de que a aceleração é o componente principal e é obtida na relação entre a distância e a velocidade relativa entre o próprio veículo e o veículo da frente, de acordo com a seguinte fórmula:

$$a_n(t) = c v_n^m(t) \frac{\Delta v(t-T)}{\Delta x^l(t-T)}. \quad (2.1)$$

Na fórmula, a_n é a aceleração do veículo n no tempo t ; v_n representa a velocidade do veículo n ; Δv e Δx representam a velocidade e distância relativa entre veículos; m , c e l são parâmetros constantes a serem determinados pelas condições únicas da estrada e do meio ambiente.

Este modelo possui algumas limitações destacando-se a aceleração nula quando a velocidade relativa entre os veículos é nula, independentemente da distância entre eles. Outra limitação é a incapacidade do veículo perseguidor atingir a sua velocidade ideal devido ao efeito que sofre do veículo líder. A necessidade de constante calibração dos parâmetros relativos às condições de estrada e do meio ambiente é outro aspecto limitativo.

Com efeito, durante as últimas décadas este modelo tem sido alvo de constantes evoluções, através da introdução de novos factores ou novas calibrações dos parâmetros constantes.

2.3.2. Modelos de distância de segurança ou prevenção de colisões

Ao contrário do modelo anterior que especifica uma função de acção-reacção, o modelo de prevenção de colisões procura definir uma distância de segurança, de modo a evitar colisões, através da manipulação das equações de movimento Newtonianas. A fórmula deste modelo é a seguinte:

$$\Delta x(t - T) = \alpha v_{n-1}^2(t - T) + \beta_l v_n^2(t) + \beta v_n(t) + b_0, \quad (2.2)$$

onde Δx representa a distância de segurança para o veículo da frente; t representa o tempo, T é o tempo de reacção; α , β_l e β representam parâmetros de calibração; v_{n-1} é a velocidade do veículo em frente; v_n é a velocidade do veículo em questão.

A facilidade com que se calibra este modelo com pressupostos básicos sobre comportamento do condutor e a inequívoca constatação dos seus resultados por parte de observadores levou ao aumento da sua popularidade.

2.3.3. Modelos lineares (ou Modelos de *Helly*)

Os modelos de *Helly* possuem similaridades com o modelo GHR, acrescentando apenas termos adicionais para uma melhor adaptação da aceleração face ao veículo da frente. O modelo rege-se pelas seguintes fórmulas:

$$a_n(t) = C_1 \Delta v(t - T) + C_2 (\Delta x(t - T) - D_n(t)), \quad (2.3)$$

$$D_n(t) = \alpha + \beta v(t - T) + \gamma a_n(t - T), \quad (2.4)$$

em que a_n representa a aceleração; $D_n(t)$ é a distância desejada entre veículos; C_1 , C_2 , α , β e γ representam parâmetros de calibração obtidos através de experiências e/ou observações; Δv a velocidade relativa; Δx a distância relativa entre veículos; t representa o presente instante temporal; T é o tempo de reacção do condutor.

Comparativamente ao modelo GHR, o modelo de *Helly* apresenta resultados muito semelhantes, concluindo-se que não acrescenta um valor significativo, seja na forma ou no grau de calibração obtido. A sua maior força reside na incorporação de um elemento de “memória”, que em situações já calculadas não volta a computar o valor de aceleração.

2.3.4. Modelos de psicofísica ou de acção

Estes modelos baseiam-se no conceito de que os condutores se apercebem quando um limite é atingido, através da mudança aparente do tamanho do veículo da frente ou à velocidade relativa. Ao ultrapassar um limite de segurança definido, o condutor opta por desacelerar evitando uma colisão. Infelizmente é difícil chegar a uma conclusão sobre a validade destes modelos, na medida em que têm sido feitos poucos trabalhos de investigação desde os anos 60. Apesar do modelo aparentemente simular comportamentos realistas, a calibração de elementos individuais e limites não tem sido bem-sucedida.

2.3.5. Modelos de *Fuzzy-Logic*

Os modelos de *fuzzy-logic* representam o próximo passo lógico na tentativa de descrever com precisão o comportamento do condutor. As primeiras versões deste modelo tentaram *fuzzificar* o modelo GHR utilizando a posição, a distância relativa e a aceleração do veículo como *inputs*. Neste tipo de modelo, as decisões dos condutores são calculadas com base num conjunto de regras de condução desenvolvidas através da experiência, limitando a sua complexidade e

fornecendo meios para uma verificação formal da estabilidade local e global do fluxo de tráfego. Apesar de aparentar ser mais realista do que o modelo GHR existem algumas limitações. A aceleração do veículo é difícil de ser medida e as situações reais de trânsito são demasiado complexas para poderem ser correctamente simuladas.

2.4. Sinopse

Espera-se ter demonstrado a importância deste tipo de modelos de simulação na compreensão da dinâmica de sistemas complexos. Trata-se da melhor maneira de obter resultados próximos da realidade de forma eficaz e económica. As abordagens metodológicas de análise de tráfego mostraram possuir objectivos distintos. Para o objectivo desta dissertação, iremos utilizar uma abordagem microscópica com o intuito de capturar detalhes do fluxo de trânsito, assim como as interações entre veículos. Sobre estes modelos, ilustrou-se a grande variedade de modelos de *car-following* já existente. Estes modelos partilham o objectivo comum de calcular a reacção (ou aceleração) correcta para o estímulo recebido. Cada modelo tem o seu próprio conjunto de condições e regras para o cálculo da aceleração e do tipo de estímulo recebido, que vão desde a utilização de diversas propriedades, como a distância para o veículo da frente e a velocidade relativa entre os veículos, até à utilização de comportamentos e emoções produzidas pelos condutores.

CAPITULO 3

Modelação Multiagente de *Car-Following*

Neste capítulo apresenta-se o modelo de simulação com especial incidência sobre os processos de *car-following* e a ligação do ambiente de simulação a Sistemas de Informação Geográfica (SIG). É enquadrada a arquitectura de dois modelos de simulação de tráfego de forma a reflectir sobre possíveis abordagens de modelação a realizar. É analisada a interligação entre os SMA e os SIG de forma a entender como é construído o ambiente de simulação. A estrutura dos algoritmos de *car-following* que se pretende implementar é analisada reunindo as suas características essenciais e necessidades. Com o mapeamento das especificidades dos algoritmos, é possível conceber o ciclo de vida que o agente deve respeitar, assim como estruturar o modelo de agente.

3.1. *VeniceTable: Interactive Traffic Simulation Table*

Actualmente os modelos disponíveis e desenvolvidos em Netlogo com interligação entre SMA e SIG são em numero reduzido. Um desses exemplos, é o *VeniceTable: Interactive Traffic Simulation Table*³. O modelo criado por Stephen Guerin e Fabio Carrera, consiste numa simulação financiada pelo projecto *CIVITAS - MOBILIS*⁴ da União Europeia com o objectivo de “ajudar cidades a alcançar um sistema de transporte urbano sustentável, ecológico e eficiente, através da implementação e avaliação de um conjunto ambicioso de políticas e tecnologias”.

Uma das cidades escolhidas por este projecto foi Veneza, situada no norte de Itália. É composta por cerca de 117 pequenas ilhas, interligadas por 409 pontes e canais de água sendo este o principal meio de comunicação utilizado dentro da cidade. Devido às limitações óbvias na infra-estrutura disponível e ao aumento do fluxo de turismo, o sistema de vias de transporte da cidade não conseguiu dar resposta às necessidades. Era preciso rever todas as operações no sistema de canais da cidade de modo a reduzir o ruído proveniente do tráfego de barcos e minimizar os graves danos arquitectónicos provocados pelas esteiras nos canais.

A solução passou por criar uma simulação multiagente de tráfego marítimo com base em dados reunidos por vários observadores em pontos importantes nos canais. Estes dados permitiram definir comportamentos para o deslocamento dos barcos, sujeito às restrições do volume total do canal e à criação de regulamentos de sentido único de tráfego em determinados canais. Foi também implementada uma ligação a dados provenientes de SIGs para a formação da infra-estrutura da cidade. Este modelo permitiria às partes interessadas planear, experimentar e otimizar regulamentos capazes de atingir os objectivos necessários.

³ Vide “*VeniceTable: Interactive Traffic Simulation Table*”, <http://redfish.com/SFComplex/projects/veniceTable.html>

⁴ Vide projecto *CIVITAS-MOBILIS*, http://www.civitas-initiative.org/project_sheet?lan=en&id=2

Apesar de ter sido um projecto patrocinado pela União Europeia, pelo município de Veneza e de ter prestado um valioso auxílio na implementação de medidas, não figura em nenhuma publicação na literatura científica. O projecto foi desenvolvido inicialmente em Netlogo, mas a implementação parece ter sido descontinuada, tendo o projecto optado pela construção de uma mesa ou de um quadro branco interactivo⁵ para exibir e facilitar a interacção com simulações. Actualmente, o modelo construído em Netlogo possui várias camadas SIG representativas de diversos tipos de visualização do ambiente geográfico e das vias de transporte fluviais disponíveis, como ilustrado na Figura 6.

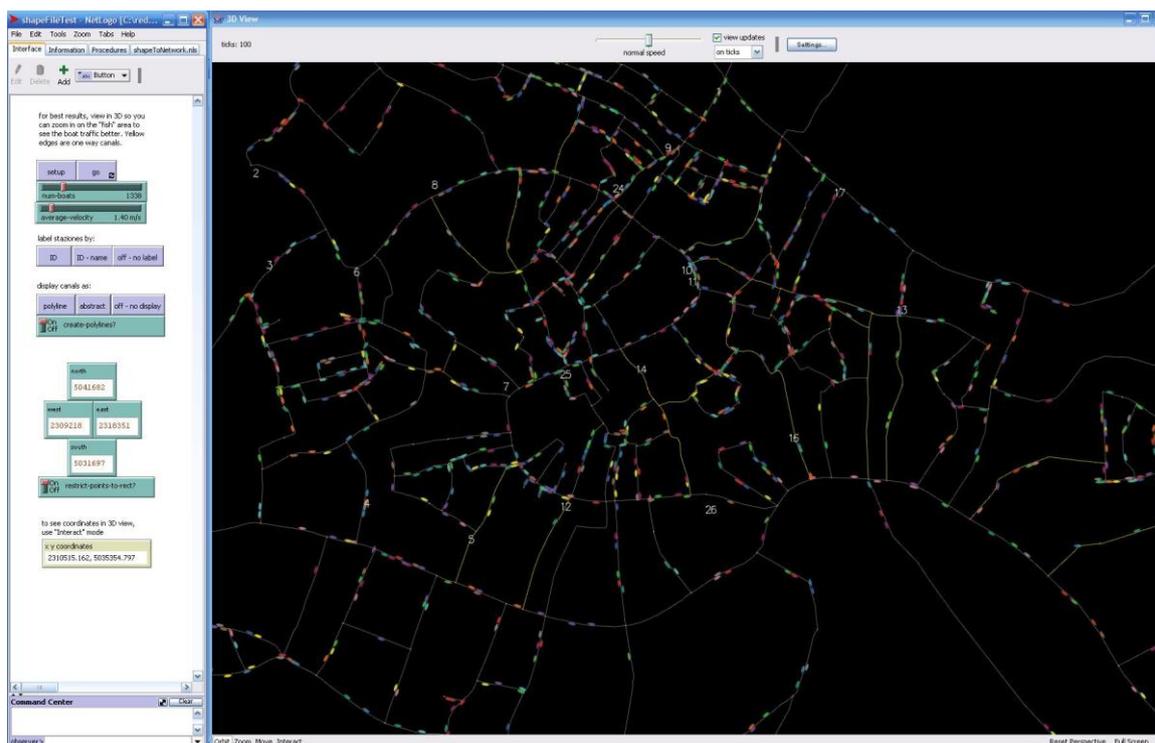


Figura 6 - Imagem do projecto VeniceTable a correr em Netlogo

3.2. Modelo de Paulo Duarte (em curso)

Foi este projecto europeu que levou Paulo Duarte (em curso) a estender o modelo de ligação entre SMA e SIG, de forma a obter um protótipo de escoamento de tráfego rodoviário.

O modelo, o qual irei estender, baseia-se em diversas funcionalidades implementadas no projecto *VeniceTable*. A construção do ambiente de simulação no Netlogo através da leitura de ficheiros SIG provém da *VeniceTable*. As extensões de Paulo Duarte envolvem a simulação de comportamentos simples nos agentes (semelhante ao *car-following*) destacando-se as variações de velocidade, a manutenção de uma distância de segurança e o respeito a algumas regras de trânsito (caso do limite máximo de velocidade e o sentido de trânsito da via em que

⁵ Trata-se de uma mesa ou quadro que maximiza a interactividade com utilizador, vide <http://www.simtable.com/>, como exemplo.

circula). Podemos observar um resumo das características de cada modelo e a sua evolução na Tabela 1.

<i>VeniceTable: Interactive Traffic Simulation Table</i>	Modelo de Paulo Duarte (em curso)	Presente Dissertação (2011)
Leitura de várias camadas de SIG;	Leitura e representação de diversas propriedades nos SIG;	-
-	Comportamentos primitivos de fluxo de trânsito, similares ao <i>car-following</i> ;	Implementação de processos realistas de <i>car-following</i> ;
Navegação aleatória dos agentes nos canais;	Implementação em curso de noções de prioridade e identificação de sinalização no ambiente;	-
Agentes não detectam colisões;	Agentes detectam e evitam colisões;	-
-	-	Criação de um modelo de agente;
-	-	Definição do ciclo de vida do agente;

Tabela 1 - Características e evolução dos modelos

3.2.1. Ambiente e interligação SMA/SIG

Observando a Figura 7 podemos constatar que arquitectura da simulação divide-se em dois componentes: o sistema geográfico de informação e o modelo multiagente.

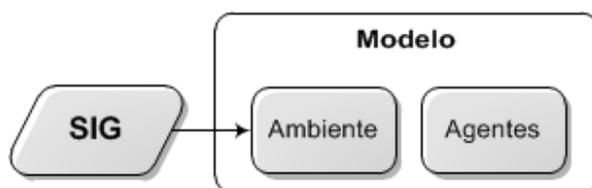


Figura 7 - Camadas existentes no modelo multiagente e a interacção com componente SIG

Muitos autores têm veiculado a importância crucial do ambiente geográfico. A própria definição clássica de agente autónomo faz menção ao ambiente: trata-se de uma entidade capaz de decidir as suas acções, tendo em consideração as mudanças no ambiente de que é parte constituinte, tentando cumprir os seus objectivos (Franklin, et al., 1996). A partir desta

definição, Weyns, et al. (2005) afirma que os agentes necessitam do ambiente para viver e interagir, uma vez que possuem um “corpo” delineado no ambiente em questão. Só após a criação de um mundo em que os agentes poderão interagir é que a simulação se tornará praticável. É neste contexto que entra a utilização de SIG.

Os SIG são um sistema de informação especialmente desenvolvido para armazenar, analisar e apresentar todos os tipos de dados geográficos referenciados, conseguindo modelar o espaço geográfico através de camadas ou níveis de informação geográfica. Isto possibilita a criação ou a combinação de diferentes tipos de ambientes. Analisando a Figura 7, podemos verificar que os dados para a construção do ambiente provêm de ficheiros localizados no componente SIG, exterior ao modelo. Esta situação deve-se às características complementares entre os SIG e os SMA, onde a vantagem dos SIG na modelação do espaço geográfico contrasta com a sua ineficiência em simular fenómenos dinâmicos, sendo esta uma qualidade dos SMA. Devido ao aumento de simulações em ambientes geográficos, com mobilidade e comportamento inteligente, houve um maior interesse em melhorar técnicas de integração entre SMA e os SIG (Gimblet, 2002). De acordo com Gonçalves, et al. (2004), podemos acoplar os SIG e os SMA para atingir um bom nível de interoperabilidade de duas maneiras:

- Acoplamento Dinâmico – onde o acesso aos dados dos SIG pode ser feito durante a execução da simulação, ou seja, o agente pode determinar certas distâncias ou efectuar interacções com o ambiente, através de operações nos SIG. Exemplificando, se um agente-veículo colidir com um sinal de trânsito, o simulador determina uma colisão e elimina a referência a esse sinal nos SIG;
- Acoplamento Estático – onde os dados são previamente carregados para o simulador, isolando o SMA e os SIG durante a simulação de qualquer tipo de interacção.

Analisando a Figura 7 podemos observar que o acoplamento é realizado de forma estática, devido à existência de duas camadas independentes mas interligadas dentro do modelo. Uma onde são encapsulados todos os processos funcionais associados ao comportamento dos agentes e outra associada à infra-estrutura do ambiente simulado, possuindo todos os processos de leitura do ambiente de dados do SIG e posterior construção do ambiente no modelo.

3.2.2. Propriedades do agente veículo-condutor

Em qualquer simulação multiagente é necessário definir as propriedades dos agentes para que estes possam executar o que foram programados, seja interagir com outros agentes ou com o ambiente. Neste caso específico em que pretendemos simular comportamentos de veículos, Kesting, et al. (2008) define as características mais importantes que afectam a condução humana agrupando-as pelo tipo de influência (interna ou externa ao agente) e pela escala temporal tal como observado na Figura 8.

As operações de curto prazo (*Operation*) realizam-se em curtos espaços de tempo, normalmente no intervalo de um segundo, e são constituídas por operações como aceleração e desaceleração. São essencialmente respostas directas ao tráfego circundante, onde o comportamento é influenciado por características como a motorização do veículo, por exemplo. Os atributos de comportamento (*Behavior*) variam de segundos a minutos e afectam a condução de forma única para cada agente, uma vez que cada condutor possui a sua própria personalidade, como por exemplo, a distância escolhida para o veículo da frente é subjectiva. Finalmente, a nível estratégico (*Strategy*) a escala temporal é de horas e contempla a decisão de começar uma viagem ou a procura de uma rota numa rede viária.

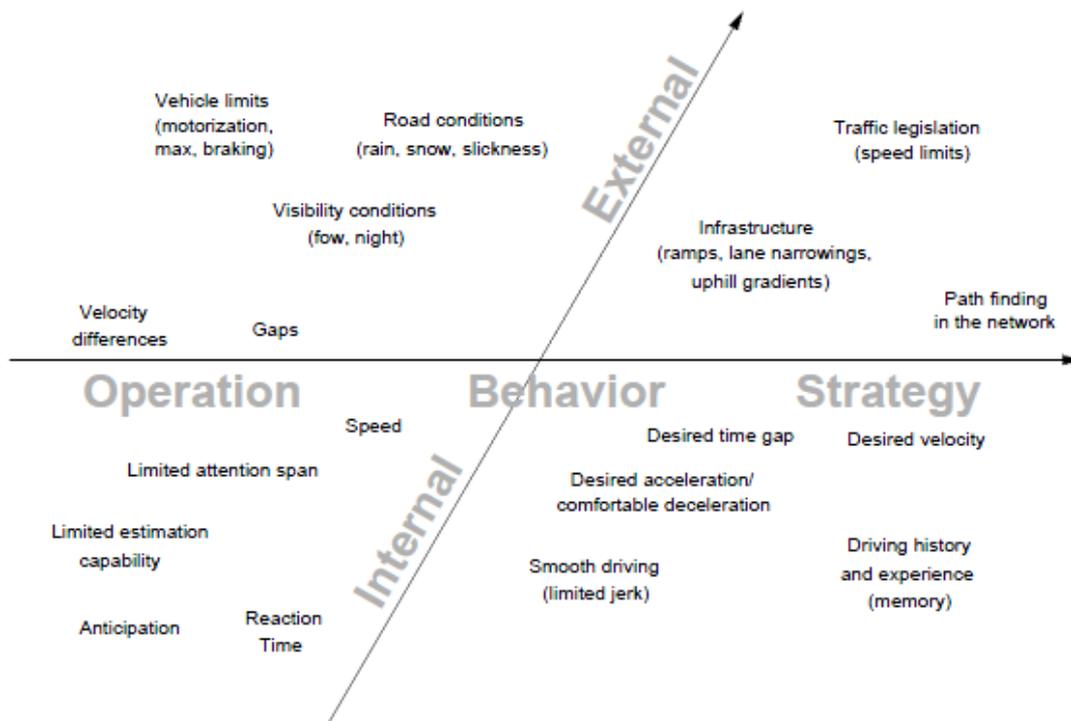


Figura 8 - Mapa das características do agente veículo-condutor, sugeridas por Kesting, et al. (2008)

Os agentes, neste modelo, pretendem representar o conjunto veículo/condutor, evitando a dissociação do comportamento resultante aos dois elementos constituintes. Ou seja, o agente “veículo-condutor” condensa as características humanas do condutor assim como as características mecânicas e físicas do veículo. Avaliando as características do modelo de Paulo Duarte (em curso), as características importantes do agente centram-se nas operações de curto prazo, sendo apresentadas na Figura 9:

Agente
-Tamanho
-Velocidade
-Troço Actual
-Percurso
-Orientação

Figura 9 - Propriedades do modelo de agente de Paulo Duarte (em curso)

Como podemos verificar, o tamanho do veículo é uma necessidade básica representando o componente físico do veículo, sendo a velocidade outro factor básico necessário. De modo a que o agente possa interagir com o ambiente, deverá conseguir identificar o troço onde se encontra assim como possuir o percurso a efectuar e a orientação da via em que se encontra. É relevante notar a ausência da propriedade aceleração, aspecto fundamental nos modelos de *car-following*, opção que será explicada nas seguintes secções.

3.2.3. Ciclo de vida do modelo

O modelo de Paulo Duarte (em curso) é construído de forma iterativa através de duas etapas principais, com funcionalidades relacionadas com as duas camadas existentes. Analisando a Figura 10⁶, podemos verificar que o modelo começa por construir o ambiente que irá sustentar a mobilidade dos agentes, uma característica básica neste tipo de simulações. Criado o ambiente, é inicializada uma segunda etapa, relacionada com o comportamento dos agentes. Nesta etapa os procedimentos funcionam de forma sequencial e circular. Numa primeira instância, os agentes são gerados de acordo com as propriedades do modelo, sendo posteriormente inseridos na simulação. De seguida, os agentes com presença no ambiente começam a interagir mutuamente e a exibir comportamentos definidos pelo algoritmo de *car-following*.



Figura 10 - Ciclo de Vida do Modelo de Paulo Duarte (em curso)

3.2.4. Algoritmo de *car-following*

A ausência da componente de aceleração origina comportamentos irrealistas por parte dos agentes.

⁶ A notação utilizada nos fluxogramas encontra-se explicada no Anexo B.



Figura 11 - Fluxograma referente ao algoritmo de car-following

Observando-se a Figura 11, podemos notar que os agentes começam por actualizar as suas intenções, recolhendo informação do meio geográfico em que se encontram, definindo o percurso possível que irão efectuar. De seguida, buscam pela existência de agentes no seu trajecto, analisando os resultados e decidindo qual a opção a tomar. Finalizam efectuando o deslocamento programado, recomeçando o ciclo. A excepção ocorre se atingirem o final da via, significando que concluíram o seu tempo de vida. A principal limitação deste algoritmo centra-se num aspecto comportamental que induz os agentes a manter uma distância mínima de segurança para com o veículo da frente, ignorando quaisquer outros factores, como a diferença de velocidades entre os agentes ou a constante monitorização da variação da distância entre os agentes. Assim, ao procurarem por veículos à sua frente, os agentes produzem um de dois comportamentos explicados de acordo com a Figura 12:

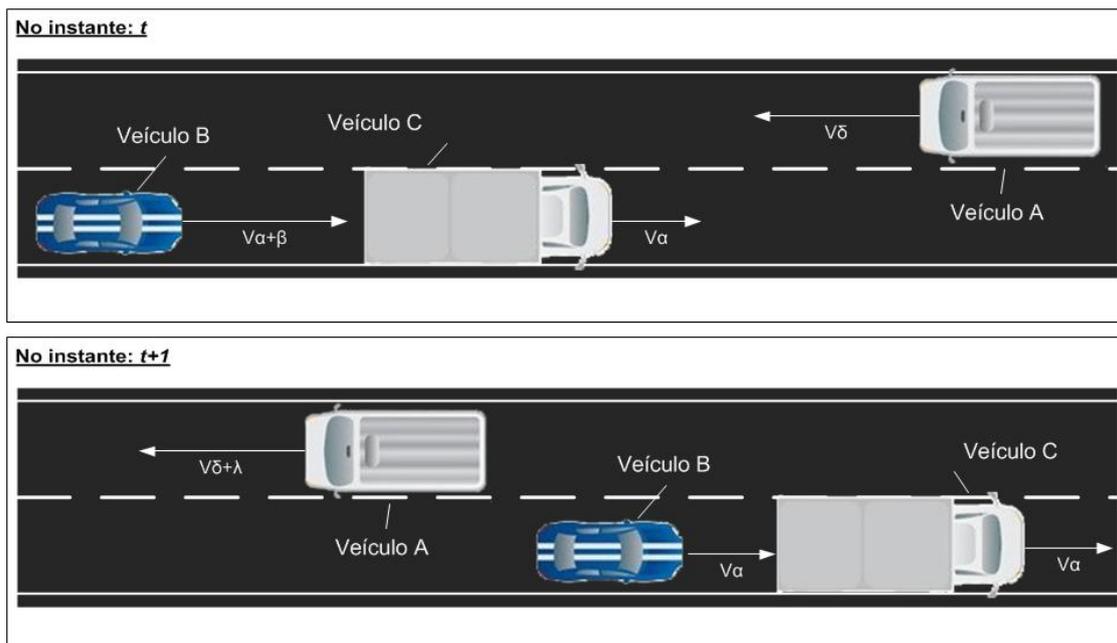


Figura 12 - Exemplos dos resultados comportamentais dos agentes

- Caso 1 – O veículo A (assim como todos os veículos) possui uma aceleração constante para qualquer situação. Isto significa que o veículo A irá aumentar a sua velocidade até encontrar um obstáculo, como é visível no Caso 2;
- Caso 2 – O veículo B, ao possuir uma velocidade instantânea bastante superior à do veículo C e encurtando a distância mútua, só reduz de velocidade quando atinge a distância mínima entre veículos, produzindo uma aceleração negativa irrealista. Em

suma, o valor da velocidade ao tentar acompanhar o do veículo perseguido, origina uma desaceleração abrupta, uma vez que a velocidade depende unicamente de uma distância mínima de segurança para o veículo da frente, ignorando uma série de outros factores relevantes;

3.3. Modelação do processo de *car-following*

Fica clara a necessidade de abordar a problemática do *car-following* para a construção de um ambiente funcional de simulação de tráfego rodoviário em NetLogo. Face à existência de diversos algoritmos de *car-following*, optou-se por implementar dois modelos distintos, de forma a providenciar uma maior flexibilidade de opções no ambiente, mas também para averiguar e confrontar o grau de realismo e de eficiência obtido para cada um dos modelos. O modelo de Dai, et al. (2010), *MultiAgent Systems for Simulating Traffic Behaviors* (MAS/STF) foi escolhido por representar um tipo de algoritmo que se baseia em conceitos cognitivos e em regras, muito apropriados ao paradigma multiagente. Por sua vez, o modelo de Treiber, et al. (2000), *Intelligent Driver Model* (IDM) inspira-se no modelo clássico GHR, baseado em equações dinâmicas.

3.3.1. *Multi-Agent Systems for Simulating Traffic Behaviors*

Este modelo baseia-se nos princípios da 2ª Lei de Movimento de Newton, observado na Eq. 3.1, produzindo um modelo contínuo no espaço e discreto no tempo, o que vai ao encontro das características dos modelos multiagente. Os autores referem que este modelo foi implementado com objectivo de usufruir das características dos sistemas multiagentes através da utilização de regras de comportamento.

$$\begin{cases} v_{t+1} = v_t + a_t \times t, \\ d_t = v_t \times t + \frac{1}{2} \times a_t \times t^2, \end{cases} \quad (3.1)$$

Na equação acima, v_{t+1} e v_t representam a velocidade instantânea no instante de tempo $t+1$ e t , respectivamente; a_t é a aceleração no instante t ; d_t é o deslocamento no instante t .

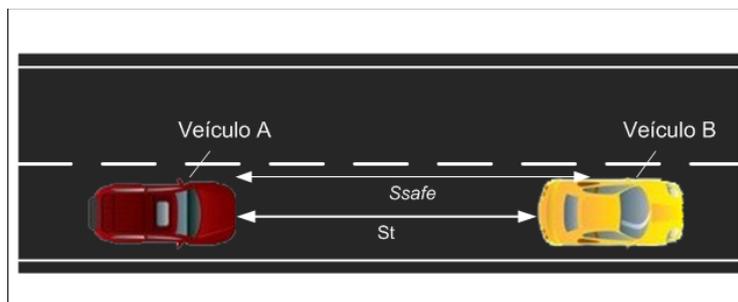


Figura 13 - Exemplo de estímulos recebidos pelo agente

Pela mecânica Newtoniana, a aceleração é considerada como a resposta do veículo a estímulos recebidos na forma de forças, sejam elas forças externas ou interações com os outros veículos no modelo. Considerando a Figura 13, em que o veículo B exerce estímulos sobre o veículo A, podemos identificar esses estímulos como percepções de distância. Essas percepções incluem a distância mútua (S_t e S_{t+1} , distância no instante t e $t+1$ respectivamente), a distância de segurança (S_{safe}) e a distância mínima que terá de existir entre eles (S_{min}). A distância mútua é facilmente entendida por ser determinada através da posição dos veículos. A distância mínima representa o espaço que se deverá manter para evitar um contacto entre veículos. A distância de segurança é dependente da velocidade, tipicamente dada pelo valor máximo obtido entre o triplo da velocidade instantânea no instante $t+1$ do agente ou o triplo da velocidade máxima permitida na via:

$$S_{safe} = \max[3 \times v_{t+1}, 3 \times v_{max}] \quad (3.2)$$

A aceleração calculada é, originalmente, classificada em seis tipos de resposta, divididos em dois níveis de distância de busca e definidos por um *framework* baseado em conceitos cognitivos. Os dois níveis de distância de busca centram-se na busca pela presença de veículos dentro de dois níveis de distância, observado na Figura 14, e como explicado seguidamente.

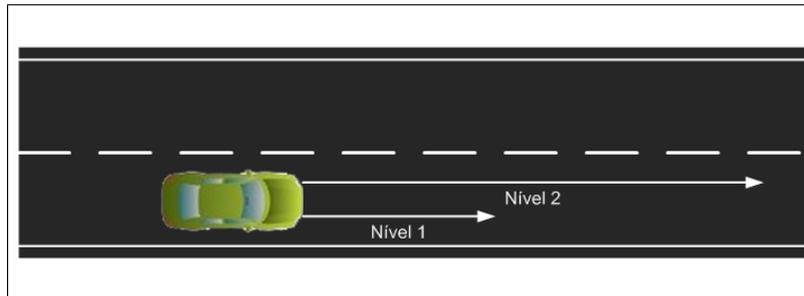


Figura 14 - Demonstração dos níveis de busca pelo agente

Conteúdo original do modelo

O nível 1 equivale à distância de segurança, ou seja, caso exista um veículo dentro da distância de segurança calculada é produzido um dos seguintes comportamentos:

- Travagem – o agente ao determinar que existe um veículo dentro da sua área de segurança ($S_{t+1} < S_{safe}$) e que a distância relativa entre eles está a reduzir ($\Delta S_{t+1} \leq 0$) começará a travar, mantendo uma distancia segura e evitando uma colisão com o veículo à sua frente, através da seguinte fórmula:

$$a_{t+1} = -v_{t+1}^2 / 2 \times (S_{t+1} - S_{min}), \text{ quando } S_{t+1} < S_{safe}, \Delta S_{t+1} \leq 0. \quad (3.3)$$

- Perseguição – se o agente determina que, embora exista um veículo dentro da sua zona de segurança ($S_{t+1} < S_{safe}$), a distância é superior à distancia mínima ($S_{min} < S_{t+1}$) e a distância relativa não está a diminuir ($\Delta S_{t+1} \geq 0$) então o veículo à sua frente está a afastar-se, permitindo que o agente acelere de forma ligeira e segura:

$$a_{t+1} = S_{t+1}/(3 \times v_{max}), \text{ quando } S_{min} < S_{t+1} \leq S_{safe}, \Delta S_{t+1} \geq 0. \quad (3.4)$$

Quando o agente detecta a existência de veículos fora da distância de segurança (nível 2), numa distância equivalente ao triplo da velocidade máxima permitida (adiante designada “distância de busca extensa”) serão gerados mais três comportamentos:

- Aceleração Rápida – se a distância do agente ao veículo for superior à distância de busca extensa ($S_{t+1} > v_{max} * 3$) mas a distancia relativa mútua estiver a aumentar ($\Delta S_{t+1} > 0$) e o veículo circular a uma velocidade abaixo da velocidade média de 12.19 m/s ($v_{t+1} < 12.19$) obtemos uma aceleração mais intensa:

$$a_{t+1} = a_{max} - (a_{max} - 0.37) \times v_{t+1}/12.19, \text{ quando } S_{t+1} > v_{max} \times 3, \\ v_{t+1} < 12.19, \Delta S_{t+1} > 0. \quad (3.5)$$

- Aceleração Lenta – de forma similar ao comportamento acima descrito, quando o agente circula acima da velocidade média ($v_{t+1} \geq 12.19$) sem contudo desrespeitar a velocidade máxima permitida ($v_{t+1} < v_{max}$), acelera de forma ligeira:

$$a_{t+1} = 0.37, \text{ quando } S_{t+1} > v_{max} \times 3, 12.19 \leq v_{t+1} < v_{max}, \Delta S_{t+1} > 0. \quad (3.6)$$

- Limite de velocidade – se o agente identificar um veículo a uma distância segura ($S_{t+1} > v_{max} * 3$) mas tiver ultrapassado o limite de velocidade permitido ($v_{t+1} > v_{max}$), terá de desacelerar de modo a evitar acidentes e a cumprir as regras de trânsito. Esta desaceleração é igual a uma ligeira desaceleração compreendida entre 0 e 1 m/s²:

$$a_{t+1} = -random(0, 1.0), \text{ quando } S_{t+1} > v_{max} \times 3, v_{t+1} > v_{max}. \quad (3.7)$$

Na eventualidade das condições anteriores não satisfazerem o tipo de interacção obtido com um veículo, é aplicado um sexto comportamento:

- Manutenção da velocidade pretendida – o agente mantém a sua velocidade ao detectar um veículo, sem que as condições anteriormente previstas tenham sido cumpridas:

$$a_{t+1} = 0 \text{ para outros casos.} \quad (3.8)$$

Nas fórmulas anteriores (Eq. 3.5 e Eq. 3.6), são utilizados para o cálculo da aceleração valores determinados empiricamente por Treiterer (1975). De acordo com o seu modelo, os veículos possuem, tipicamente, um regime de dupla-aceleração que depende do perfil de velocidade

média, dado por 12.19 m/s. Deste modo um veículo que circule acima da velocidade média deverá ter uma aceleração de 0.37 m/s² e 1.1 m/s² se circular abaixo da velocidade média.

Extensões feitas ao algoritmo

O MAS/STB foi originalmente concebido para funcionar em circuito fechado, ou seja, os veículos possuem todas as informações relativas aos restantes veículos dentro da simulação. Essa não é, contudo, uma situação realista para a simulação de tráfego em larga escala, uma vez que os agentes apenas se devem interessar por outros agentes que exerçam estímulos directos nas suas proximidades. Desta forma, foi necessário adicionar um comportamento de imobilização no nível 1.

- Imobilização – de modo a evitar que o veículo ultrapasse a distância mínima de segurança com o veículo da frente, o veículo deve parar a sua marcha:

$$a_{t+1} = -2 \times v_{t+1}, \text{ quando } S_{t+1} \leq S_{min}, \Delta S_{t+1} < 0. \quad (3.9)$$

Os comportamentos obtidos para o segundo nível de busca foram determinados a partir de adaptações dos comportamentos homólogos:

- Aceleração Rápida em Estrada Livre – o agente ao não detectar a existência de veículos à sua frente acelera de forma intensa por circular abaixo da velocidade média ($v_{t+1} < 12.19$) definida por Treiterer (1975):

$$a_{t+1} = a_{max} - (a_{max} - 0.37) \times v_{t+1}/12.19, \text{ quando } v_{t+1} < 12.19 \quad (3.10)$$

- Aceleração Lenta em Estrada Livre – similar ao modelo anterior mas quando o agente circular acima da velocidade média ($v_{t+1} \geq 12.19$) sem veículos à sua frente:

$$a_{t+1} = 0.37, \text{ quando } v_{t+1} \geq 12.19, \quad (3.11)$$

- Limite de velocidade – se o agente tiver ultrapassado o limite de velocidade permitido ($v_{t+1} > v_{max}$), terá de desacelerar:

$$a_{t+1} = -random(0, 1.0), \text{ quando } v_{t+1} > v_{max}. \quad (3.12)$$

Finalmente, e na ausência do cumprimento das interações anteriores:

- Manutenção da velocidade pretendida – não satisfazendo nenhuma das condições anteriores irá conservar a sua velocidade actual:

$$a_{t+1} = 0 \text{ para outros casos.} \quad (3.13)$$

Na Tabela 2, resumem-se os comportamentos gerados pelo algoritmo final, agrupados de acordo com categorias e níveis.

MAS/STB	Existência de Veículos	Dentro da Distância de Segurança	Travagem
			Perseguição
			<u>Imobilização</u>
		Distância de Busca Extensa	Aceleração Rápida
			Aceleração Lenta
			Limite de Velocidade
	Outras Opções	Manutenção de Velocidade	
	<u>Inexistência de Veículos</u>	<u>Aceleração Rápida em Estrada Livre</u>	
		<u>Aceleração Lenta em Estrada Livre</u>	
		<u>Limite de Velocidade em Estrada Livre</u>	
<u>Manutenção de Velocidade em Estrada Livre</u>			

Tabela 2 - Framework resultante do modelo MAS/STB. Os comportamentos adicionados encontram-se sublinhados.

3.3.2. *Intelligent Driver Model*

O outro algoritmo considerado para implementação é o IDM. Este é um modelo de *car-following* que incorpora diferentes modos de condução para todas as velocidades seja em auto-estrada ou tráfego citadino (Treiber, et al., 2000). A aceleração do veículo é calculada com base na sua posição e velocidade relativamente ao veículo à sua frente num determinado instante, através da seguinte fórmula:

$$\dot{v}_\alpha = \frac{dv_\alpha}{dt} = a \left(1 - \left(\frac{v_\alpha}{v_0} \right)^\delta - \left(\frac{s^*(v_\alpha, \Delta v_\alpha)}{s_\alpha} \right)^2 \right). \quad (3.14)$$

Esta equação divide-se em dois componentes, um relativo à tendência de acelerar e outro relativo à desaceleração.

$$\dot{v}_\alpha = a \left(1 - \left(\frac{v_\alpha}{v_0} \right)^\delta \right), \quad (3.15)$$

$$\dot{v}_\alpha = -a \left(\left(\frac{s^*(v_\alpha, \Delta v_\alpha)}{s_\alpha} \right)^2 \right), \quad (3.16)$$

No primeiro caso (Eq. 3.15), o componente de aceleração estimula o agente a aumentar a velocidade, com o intuito de atingir a velocidade pretendida (v_0), considerando a inexistência de interações directas com outros veículos. A variável a representa a taxa de aceleração

confortável, v_α é a velocidade actual do veículo e o expoente δ representa o expoente de aceleração calculado empiricamente. É perceptível que iremos observar diminuições no valor da aceleração quando a velocidade instantânea do veículo se for aproximando da velocidade pretendida, originando a diminuição gradual da aceleração até atingir o valor nulo.

Por sua vez, o componente de desaceleração (Eq. 3.16) baseia-se no rácio entre uma “distância dinâmica desejada” ($S^*(v_\alpha, \Delta v_\alpha)$) e a distância actual para o veículo da frente (S_α , ver Figura 15). A distância dinâmica desejada é calculada através da seguinte equação:

$$S^*(v_\alpha, \Delta v_\alpha) = S_0 + v_\alpha T + \frac{v_\alpha \Delta v_\alpha}{2\sqrt{ab}}. \quad (3.17)$$

A distância dinâmica desejada é composta por três parâmetros. Uma distância mínima entre veículos (S_0) mantida quando existe inactividade no trânsito, por exemplo; uma distância de segurança ($v_\alpha T$) que garante que numa travagem inesperada o veículo ao circular a uma velocidade v_α tenha tempo suficiente (T – tempo de avanço) para evitar uma colisão; um parâmetro que produz um certo sincronismo com o veículo da frente, calculado através da utilização da velocidade do veículo (v_α), da velocidade relativa entre eles (Δv_α), e das taxas de aceleração e travagem “confortáveis” (a e b , respectivamente).

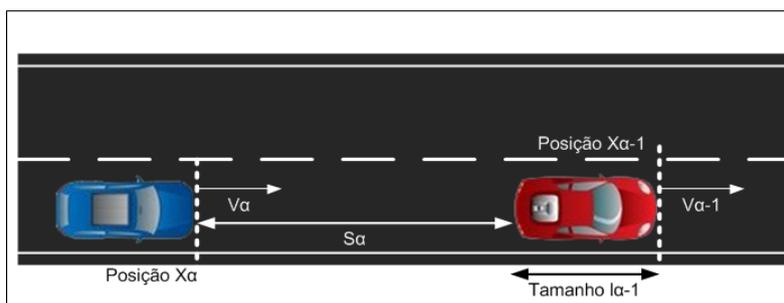


Figura 15 - Demonstração dos estímulos recebidos. A distância S_α é definida por $X_{\alpha-1} - X_\alpha - l_{\alpha-1}$.

Esta distância tenderá a aumentar quando o veículo se aproxima de outros mais lentos, ou a diminuir quando o veículo da frente estiver afastar-se. Podemos afirmar que o valor da desaceleração aumentará sempre que a distância dinâmica desejada for maior que a distância actual ao veículo.

Neste contexto, podemos então definir dois tipos de comportamento realizado pelo agente:

- Comportamento em via livre – o agente ao não encontrar agentes para interagir, utilizará apenas o componente referente à aceleração, observado na Eq. 3.15;
- Comportamento com interacção – o agente ao se encontrar numa via povoada por outros agentes, irá tentar produzir um comportamento equilibrado de interacção utilizando a Eq. 3.14.

3.4. Modelo e ciclo de vida do agente veiculo-condutor

Podemos definir um modelo geral de agente que suporte as características dos dois algoritmos, apresentado na Figura 16.

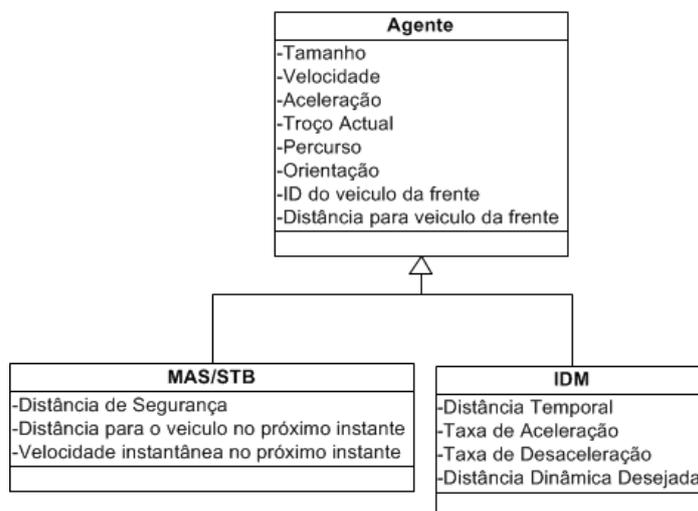


Figura 16 - Novo modelo de Agente

Às características factuais de um veículo (tamanho, velocidade, troço actual) teremos de acrescentar a aceleração, ausente do modelo original. É necessário adicionar características relacionadas com a percepção humana, caso da identificação do veículo da frente, a distância relativa, o sentido de trânsito e o percurso a realizar. Particularizando para o algoritmo MAS/STB teremos de acrescentar a distância de segurança, por se tratar de um componente importante no cálculo do comportamento do condutor. Pelo mesmo motivo, o condutor terá de avaliar a distância para o veículo da frente e a sua própria velocidade no próximo instante temporal. No caso do IDM é necessário definir a distância temporal para o veículo da frente, assim como as taxas de aceleração e travagem confortáveis. A distância dinâmica desejada é a propriedade chave deste algoritmo para os comportamentos interactivos. De salientar o facto de não existir uma propriedade que represente o peso do veiculo. Os algoritmos apresentados não levam em conta que o peso influencia a resistência ao deslocamento do veículo.

O ciclo de vida dinâmico do agente está dividido em dois processos principais. O primeiro destina-se à geração e criação de agentes. O segundo representa o conjunto de processos relativos ao comportamento de *car-following* dentro do ambiente de simulação.

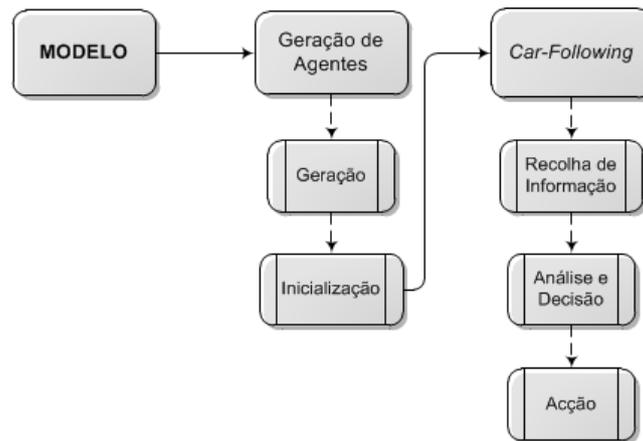


Figura 17 - Ciclo de vida do agente

No primeiro processo na Figura 17 (Geração de Agentes), o agente passa por dois estados. É gerada o formato físico do agente e definido o seu ponto de inserção. De seguida, são preenchidas algumas das propriedades básicas do veículo como o tamanho e a velocidade, aguardando permissão para iniciar a próxima fase. Posteriormente, ao ser inicializado numa lista, são preenchidas as restantes propriedades, relacionadas com a percepção humana, caso do percurso a efectuar e o identificador de veículos, ficando disponível para executar a segunda sequência de processos, destinados ao *car-following*.

Após ser dada a permissão de se transferir da lista para o ambiente, o agente inicia a interacção com os restantes agentes e com o ambiente, tentando definir o comportamento mais adequado. O agente começa por recolher informações ao seu redor, verificando o percurso que pretende fazer e verifica a existência de veículos à sua frente. Caso exista, calcula a distância a que se encontra deste. Após analisar os dados recebidos, o agente toma a decisão de alterar (ou não) a sua velocidade actual calculando uma nova aceleração se necessário. Conclui o seu ciclo de vida com a execução da acção definida.

O tempo de vida do agente termina quando este se apercebe que atingiu o final da via, não encontrando qualquer tipo de percurso que possibilite o seu deslocamento na direcção pretendida.

3.5. Sinopse

Com a descrição do *VeniceTable* e da extensão realizada por Paulo Duarte foi ilustrada a interligação funcional entre um SMA e um SIG através do Netlogo. Conseguiu-se confirmar que a metodologia utilizada no desenvolvimento deste último modelo pode ser reutilizada tendo em atenção as limitações identificadas. Foi nas estruturas e processos que envolvem os comportamentos dos agentes que foram realizadas mudanças e melhoramentos.

A extensão aos modelos *VeniceTable* e de Paulo Duarte envolvem a implementação de dois algoritmos de génese diferente: o MAS/STB e o IDM. O primeiro é um algoritmo recente na área, baseado em conceitos cognitivos e especificado através de regras, onde se mostrou necessário estende-lo com algumas regras adicionais, caso de comportamentos em que o agente circula sozinho. O IDM, por sua vez, é um algoritmo mais citado e conhecido sendo o seu funcionamento especificado através de uma única equação global, em função da dinâmica dos veículos. Tendo em vista a representação dos vários tipos de comportamentos determinados pelos algoritmos, foram adicionadas novas propriedades ao modelo de agente e reformulado o seu ciclo de vida.

CAPITULO 4

Implementação do Modelo

Neste capítulo descreve-se a arquitectura da implementação do modelo, especificado no capítulo anterior. São exibidas, de uma forma genérica, algumas ferramentas que cumprem os critérios definidos para a construção do modelo com especial ênfase no Netlogo, que foi a ferramenta escolhida para a implementação do modelo e dos casos de teste. Descreve-se em algum detalhe as etapas referentes à construção do ambiente do modelo, dos algoritmos de *car-following*, da estrutura de dados dos agentes e demais processos necessários para a execução de simulações.

4.1. Ferramentas de simulação

Na fase de implementação torna-se necessário introduzir as possíveis ferramentas de desenvolvimento. Castle, et al. (2006) apresentam diversas sugestões com vantagens e desvantagens diversas e descritas na Tabela 3.

Idealmente, a ferramenta deve ser conhecida no meio académico, não pode estar descontinuada e deve dar apoio à extensão SIG. Serão cinco as ferramentas a apresentar, quatro de carácter genérico (*Swarm*, *MASON*, *REPAST*, *Netlogo*) e uma dedicada (*SUMO*).

O *Swarm* é uma plataforma de *software* desenvolvida pelo Instituto de Santa Fé e foi especialmente desenhada para o desenvolvimento de simulações multiagente, tendo sido criada para o estudo de sistemas biológicos mas utilizada nas mais variadas disciplinas (Minar, et al., 1996). Trata-se de uma plataforma para modelos baseados em agentes com capacidade para desenhar, descrever e conduzir experiências, possuindo ferramentas bastante úteis, incluindo pacotes para análise gráfica e estatística. A necessidade de possuir uma grande experiência de programação torna difícil o seu rápido aproveitamento, estando a cair em desuso pelos investigadores. A estrutura lógica da interacção dos agentes com os eventos é feita através da linguagem Objective-C.

O *MASON* (*Multi-Agent Simulator Of Neighborhoods*), por sua vez, é uma ferramenta desenhada para simulação de eventos discretos (Luke, et al., 2005). É baseada na linguagem Java e foi bastante influenciada pelo *Swarm*. É rápida computacionalmente, com modelos independentes da visualização e onde os modelos podem ser migrados entre plataformas. Não possui capacidade para análise de resultados, mas possui interligação com os SIG. Apesar de também possuir uma grande curva de aprendizagem, possui imensos tutoriais e modelos para efeitos de auxílio ao utilizador.

O *REPAST* (*Recursive Porous Agent Simulation Toolkit*) é um conjunto de duas bibliotecas mais complexas que as ferramentas anteriores, mas igualmente difícil de dominar. É um dos

ambientes de modelação mais conhecidos e utilizados pela comunidade de simulação social. Possui um interface bastante útil para o desenvolvimento, extensões para a análise de resultados, é completamente orientado a objectos e com suporte integrado a SIG (North, et al., 2007).

O Netlogo (Wilensky, 1999) é um ambiente de simulação muito usado por estudantes e investigadores, graças à sua facilidade de utilização (tanto pelo interface, como pela facilidade de aprendizagem) e pelas suas extensões integradas de análise de resultados e integração com os SIG. É uma ferramenta especialmente desenvolvida para modelar sistemas complexos, capaz de definir instruções a centenas ou milhares de agentes independentes. Este facto torna possível explorar a conexão microscópica entre os comportamentos dos agentes e os padrões que emergem num nível macroscópico.

O SUMO⁷ (*Simulation of Urban Mobility*) é desenvolvido pelo Instituto de Sistemas de Transporte no Centro Aeroespacial Alemão, sendo uma ferramenta dedicada e portátil de simulação microscópica de tráfego rodoviário, ao contrário das anteriores ferramentas generalistas. Foi criada com o intuito de apoiar a comunidade de investigadores com uma ferramenta capaz de implementar e avaliar os algoritmos criados, sem a necessidade de criar novos métodos de construção de redes de estrada ou de dispositivos de controlo de tráfego para obter uma completa simulação de tráfego. A principal aplicabilidade desta ferramenta centra-se na implementação e avaliação de métodos de gestão de tráfego.

	<i>Swam</i>	<i>MASON</i>	<i>REPAST</i>	<i>Netlogo</i>	<i>SUMO</i>
Política de Licenciamento	<i>Open Source</i>	<i>Open Source</i>	<i>Open Source</i>	<i>Freeware</i>	<i>Open Source</i>
Sistema Operativo	Windows, Linux, Unix, Mac OSX	Windows, Linux, Unix			
Curva de Aprendizagem	Lenta	Lenta	Lenta	Rápida	Moderada
Funções SIG integradas	Sim	Sim	Sim	Sim	Não
Funções de análise e estatística Integradas	Sim	Não	Sim	Sim	Não
Modelos Disponíveis	Sim	Sim	Sim	Sim	Sim
Tutoriais/Documentação	Sim	Sim	Sim	Sim	Sim

Tabela 3 - Comparação entre as quatro ferramentas de modelação genéricas e a ferramenta dedicada, adaptado de Castle, et al. (2006) e Gilbert (2008)

⁷ Vide SUMO, <http://sumo.sourceforge.net/>

Com base na relação entre as necessidades da simulação e as características das ferramentas apresentadas e pelo facto de o modelo-base (descrito no capítulo anterior) ter sido desenvolvido utilizando o Netlogo, será esta a ferramenta utilizada. O Netlogo possui um interface gráfico e de desenvolvimento muito intuitivo permitindo que utilizadores iniciados e experientes trabalhem sem dificuldades (Lansdowne, 2006).

4.1.1. O ambiente Netlogo

Como podemos ver pela Figura 18, o interface do Netlogo apresenta ao utilizador três separadores: *Interface*, *Information* e *Procedures*.

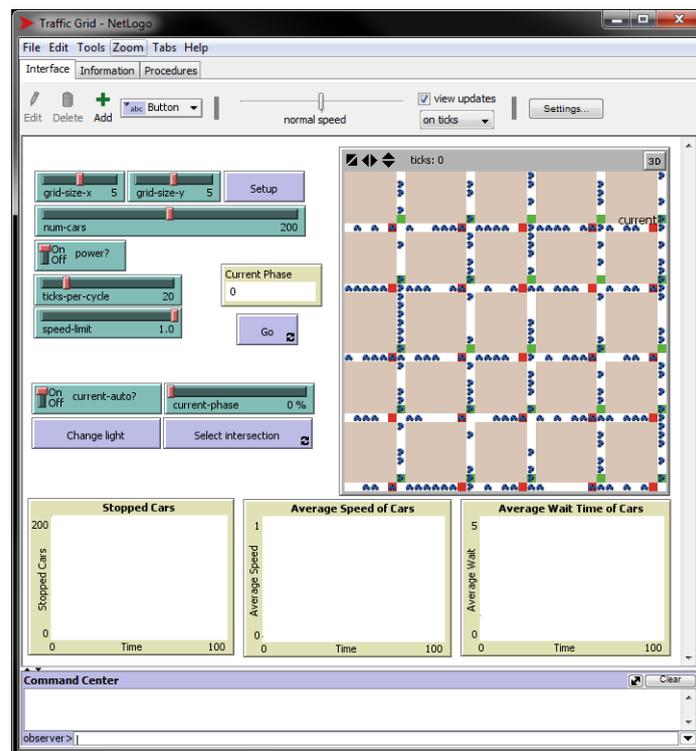


Figura 18 - Exemplo do interface Netlogo, *Traffic Grid Model* (Wilensky, 2003)

O separador *Interface* apresenta os *outputs* provenientes da simulação, o seu resultado visual, assim como o controlo da execução da simulação. Possui vários *widgets* de fácil aplicação para alteração dos parâmetros de simulação e permite a apresentação de resultados na forma de gráficos ou em vários tipos de monitores. O tempo é representado de forma discreta e pode ser ajustado pelo utilizador. Por norma, todos os modelos apresentam botões de *Setup* e *Go*. O primeiro invoca um procedimento de (re)inicialização do modelo e o segundo é responsável por executar a simulação.

O separador *Information*, por sua vez, possui a documentação e descrição relativa à simulação. Fornece uma exposição geral dos objectivos da simulação, das regras de funcionamento e

utilização, de aspectos interessantes a observar alterando as configurações, entre outras informações.

No separador *Procedures* podemos encontrar o código-fonte relativo à simulação, constituído essencialmente por comandos, relatores (*reporters*) e variáveis. Os comandos representam as acções que um agente terá de cumprir enquanto o relator é o responsável pela computação e disponibilização do resultado proveniente de dados. Por sua vez, as variáveis são locais para guardar valores de um agente. De modo a facilitar o trabalho do programador, as variáveis estão divididas em dois tipos: a variável global que pode ser acedida por qualquer agente; e a variável incorporada no agente (local) que guarda diferentes valores.

O Netlogo possui um conjunto de métodos, relatores e variáveis incorporadas e que são designadas de primitivas. Os métodos criados pelo programador são conhecidos como procedimentos. Como já foi referido, o Netlogo é um ambiente de modelação programável de micro-simulação e utiliza quatro tipos de agentes, três deles responsáveis pela construção de modelos: dois programáveis (*turtles* e *patches*) e um interactivo (*links*). O quarto tipo de agente, não utilizado é o *observer*. Este agente pode ser comparado à representação do programador ou utilizador dentro do modelo.

As *turtles* são a representação gráfica dos agentes, podendo possuir diferentes formas, tamanhos e cores, ilustrado na Figura 19. São elas que seguem as instruções que resultam em comportamentos diferentes e simultâneos. Possuem propriedades internas e personalizadas de acordo com o modelo a simular, e podem ser classificadas em tipos de classe (*breeds*) para facilitar a sua distinção.

Como podemos ver pela Figura 20, o mundo Netlogo (também denominado *world*) está dividido em coordenadas, sendo cada coordenada denominada por *patch*, produzindo uma representação discreta do mundo. De forma similar às *turtles*, os *patches* também possuem propriedades internas que lhes permite, por exemplo, mudar a cor. Os *links* têm a função de interligar duas *turtles* normalmente designadas *nós*, sendo representados por uma linha de ligação. Os *links* e as *turtles* são o conjunto responsável pela representação dos traçados lidos para o ambiente de simulação através do SIG.

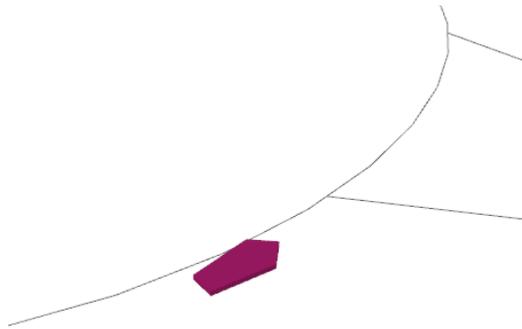


Figura 19 - Imagem de uma *turtle* (roxo) a circular no *link* (ligação cinzenta). O *patch* está presente a branco, em segundo plano.

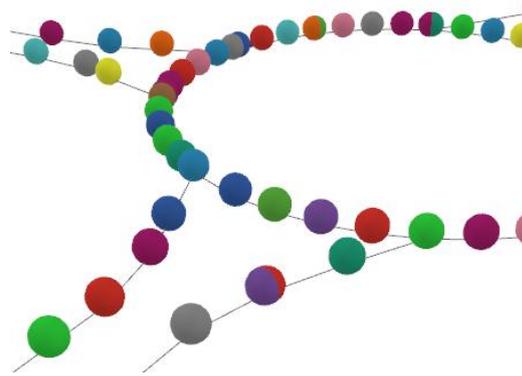


Figura 20 - Representação gráfica dos *nós* (bolas coloridas), *links* (ligações entre nós) e *patches* (fundo da imagem).

4.2. Estrutura da implementação

A implementação do modelo seguiu a estrutura observável na Figura 21, originando seis fases de implementação.



Figura 21 - Exemplo das cinco fases de implementação definidas

Existem cinco fases sequenciais e uma transversal às restantes. Como já foi referido, antes de implementar o SMA é necessário construir o ambiente. Para tal reutilizou-se o conjunto de procedimentos desenvolvido pelos autores do *VeniceTable* e estendido por Paulo Duarte (em curso). Finda esta fase, segue-se a implementação dos processos de *car-following* de forma faseada a par com a implementação do modelo de agente, concluindo com o processo de injeção de veículos compatível com os algoritmos implementados. Na sequência destes processos, o interface passa por constantes alterações de acordo com as necessidades surgidas. O código-fonte pode ser consultado no Anexo A.

4.2.1. Unidades de tempo de ciclo de vida de uma simulação

O Netlogo representa o tempo de forma discreta. No presente modelo, cada instante de tempo (denominado *tick*) equivale a 1 segundo real. Como se pode observar pela Figura 22, são executados dois subprocessos pelo procedimento *GO* em cada *tick*. O primeiro processo refere-se ao cálculo dos valores de velocidade e aceleração dos veículos ao circularem na simulação, ou seja, ao comportamento que o condutor-veículo irá possuir face ao estímulo recebido. O segundo processo é responsável pela geração e inserção de veículos adicionais na simulação.

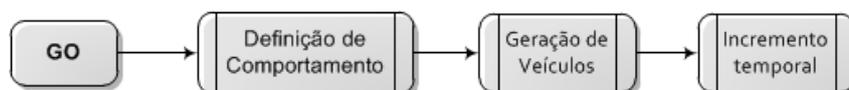


Figura 22 - Ciclo de vida dos processos referentes aos SMA

4.2.2. Construção do ambiente

A construção do ambiente no Netlogo é feito com o auxílio da extensão SIG, necessária para a leitura e transformação dos dados dos ficheiros SIG, gerados em formato *shapefiles*. Estes ficheiros podem prover de cartografias de locais reais, e podem ser criados com várias especificações utilizando ferramentas como o ArcGIS⁸.

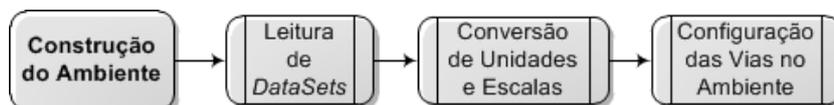


Figura 23 - Subprocessos utilizados na construção do ambiente

De acordo com a Figura 23, inicialmente é carregado o *dataset* correspondente ao ambiente pretendido, criando-se de seguida a área destinada a albergar o ambiente. É feita a conversão de unidades e escalas interligando os limites do mapa com os limites do mundo Netlogo. É formulada uma escala entre a distância métrica do SIG e o tamanho dos *patches* (*meters/patch*), o qual irá auxiliar futuros cálculos. Posteriormente é construída a rede viária, através de ligações (*links*) entre os *nós* ou vértices criados na rede.

4.2.3. Modelo de agente

A implementação do modelo de agente sofreu alguns ajustes devido às características da linguagem de programação do Netlogo. Algumas propriedades que seriam específicas do

⁸ Vide *homepage* do ArcGIS, <http://www.esri.com/software/arcgis/index.html>

agente ou dos algoritmos, foram definidas como variáveis locais de forma a manter eficiência da aplicação. Do mesmo modo, outras propriedades que seriam de âmbito local, interagindo apenas com as propriedades do algoritmo, foram adicionadas ao modelo de agente. O modelo central não sofreu alterações significativas, contudo, como podemos ver pela Figura 24 fizeram-se alterações à estrutura de dados de cada modelo de *car-following*. Observando a ramificação dedicado ao MAS/STB, observa-se que os agentes agora representam também a aceleração no próximo instante de tempo para efeitos de cálculo. A representação da distância para o veículo no próximo instante não está representada na estrutura de dados uma vez que é fácil de calcular localmente. No caso do IDM, a maior parte das propriedades podem ser calculadas localmente. Por essa razão não existe a necessidade de representar propriedades específicas para este algoritmo.

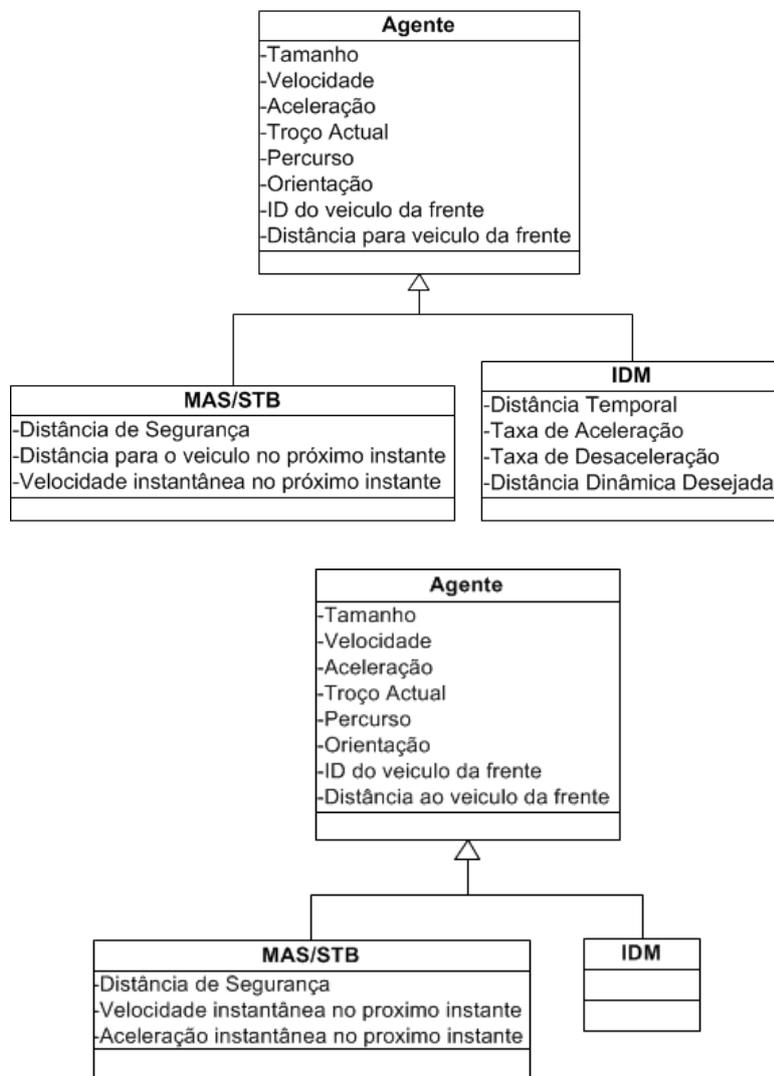


Figura 24 - Comparação entre modelo de agente definido na modelação e a estrutura de dados de implementação

4.2.4. Ciclo de vida dos agentes

Ambos os modelos seguem o ciclo de vida definido no capítulo anterior. Os agentes ao serem inseridos no ambiente executam três processos sequenciais: Recolha de Informação, Análise e Decisão e realização da Acção.

O primeiro procedimento, visível na Figura 25, é utilizado apenas pelo algoritmo MAS/STB, dada a necessidade de calcular a distância de segurança, baseada na velocidade instantânea do veículo. O agente calcula a velocidade instantânea do instante seguinte, baseando-se na velocidade e aceleração actual, calculando de seguida a distância de segurança.

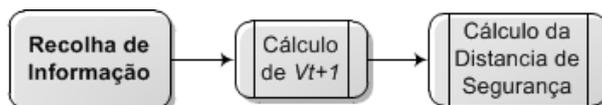


Figura 25 - Fluxograma relativo ao processo de Recolha de Informação

A Figura 26 representa o processo em que o agente define uma rota possível, procurando pelo veículo que lhe é mais próximo nesse percurso. Ao identificar um veículo na sua dianteira, o agente irá reagir de acordo com os estímulos enviados por esse veículo. Caso não encontre qualquer veículo no seu percurso, o agente adopta um comportamento de condução livre.

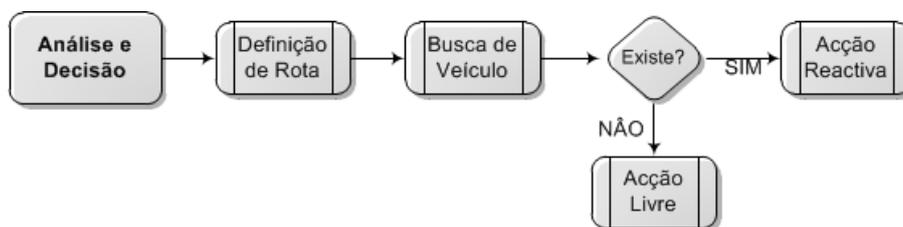


Figura 26 - Fluxograma referente ao processo de Análise e Decisão

A finalizar, na Figura 27 apresenta-se o processo de realizar uma Acção que permite ao agente efectuar o deslocamento programado e actualizar componentes relativas aos veículos.

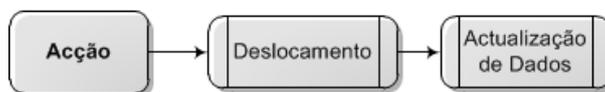


Figura 27 - Fluxograma relativo ao processo de realização da Acção de movimento

A execução dos algoritmos, por sua vez, surge no final da sequência apresentada na Figura 26. Como já foi referido, cada algoritmo possui dois tipos de acção: um em que reage aos estímulos do veículo da frente; e um segundo em que age por sua iniciativa.

4.2.5. *Multi-Agent Systems for Simulating Traffic Behaviors*

Para este modelo de mecânica newtoniana, o ciclo de conduta incide sobre um de dois tipos de acção gerais:

- Na Acção Reactiva, o agente reage às acções do veículo à sua frente, acelerando ou travando de forma a manter sempre uma distância de segurança;
- Na Acção Livre, quando o agente não encontra veículos dentro da sua área de busca, são lhe dadas permissões para acelerar dependendo da sua velocidade actual;

Em ambos os casos, serão aplicadas as condições estudadas no capítulo anterior, de onde o agente obterá um comportamento mais adequado ao ambiente que o rodeia. De modo a facilitar a visualização no modelo, optou-se por corresponder uma cor a cada comportamento:

- Vermelho – Travagem;
- Azul – Perseguição;
- Azul Claro – Imobilização;
- Verde – Aceleração Rápida e Aceleração Rápida em Estrada Livre;
- Castanho – Aceleração Lenta e Aceleração Lenta em Estrada Livre;
- Laranja – Limite de Velocidade e Limite de Velocidade em Estrada Livre;
- Cinzento – Manutenção de Velocidade e Manutenção de Velocidade em Estrada Livre;

4.2.6. *Intelligent Driver Model*

Comparativamente com o modelo anterior, o IDM possui uma implementação mais fácil. Como avalia as situações de uma forma dinâmica não necessita de várias condições para produzir determinados comportamentos.

Também aqui foi implementado uma correlação entre valores da aceleração e cores. Neste caso, as condições são as seguintes:

- Aceleração negativa em acção reactiva - o agente toma uma cor vermelha;
- Aceleração positiva em acção reactiva - observa-se a mudança para cor verde;
- Aceleração nula em acção reactiva - o agente possui uma cor cinzenta;
- Comportamento livre - o agente está representado pela cor azul;

4.2.7. *Geração e injeção de agentes no ambiente*

Ambos os algoritmos utilizam a mesma estrutura para a fase de geração e injeção de agentes, como se verifica pela Figura 28. Em cada instante de tempo, e ao iniciar este procedimento, são gerados agentes de acordo com valores definidos pelo utilizador, após o qual lhe são

preenchidas determinadas propriedades. É-lhes indicado o ponto de inserção (*nó* e o link correspondente) na rede rodoviária. O tamanho do veículo será universal, fixado em 4 metros de comprimento e representará os veículos da classe “Passageiro – Segmento C”^{9,10} de acordo com a *Euro NCAP*¹¹. O veículo possuirá uma velocidade inicial definida pelo utilizador.

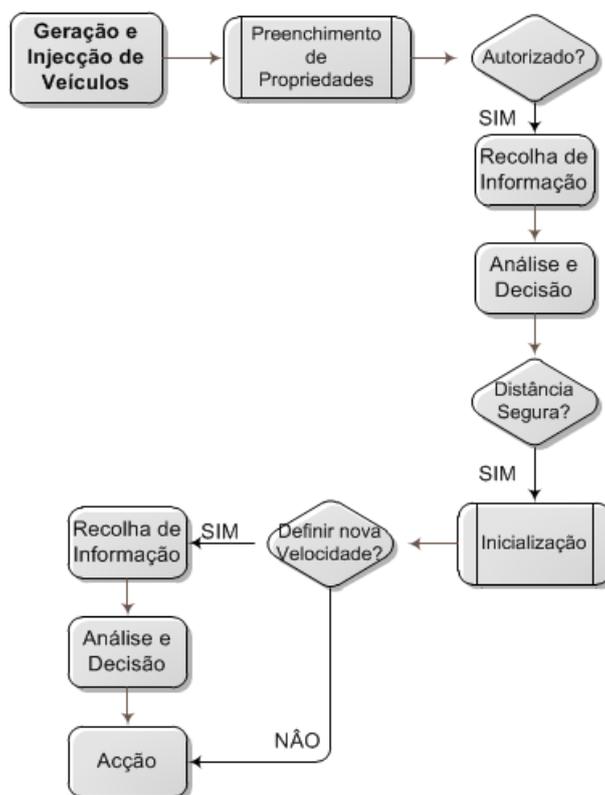


Figura 28 – Sequência de processos dedicados à geração e injecção de agentes

Após preenchidas as propriedades, o agente é inserido numa fila de espera enquanto não lhe é permitida a entrada na simulação, de acordo com uma função binomial de probabilidade. Quando um agente é injectado, executa os procedimentos descritos de Recolha de Informação (apenas para o MAS/STB) e Análise e Decisão com o objectivo de identificar a existência de veículos à sua frente e calcular a distancia mutua. Caso exista uma distância mínima de injecção suficiente, o agente verifica a necessidade de calcular uma nova velocidade ou a manutenção da defendida pelo utilizador. Se a distância for perigosamente próxima o veículo manter-se-á na fila de espera, aguardando nova oportunidade de injecção. Desta forma, a injecção de agentes no ambiente não depende apenas de uma função de probabilidade, mas também do tempo e de uma distância de segurança adequada ao momento.

⁹ Vide classificação automóvel, http://en.wikipedia.org/wiki/Car_classification

¹⁰ Vide segmentação automóvel, http://ec.europa.eu/competition/mergers/cases/decisions/m1406_en.pdf

¹¹ Vide homepage da Euro NCAP, <http://www.euroncap.com/home.aspx>

4.3. Interface gráfico

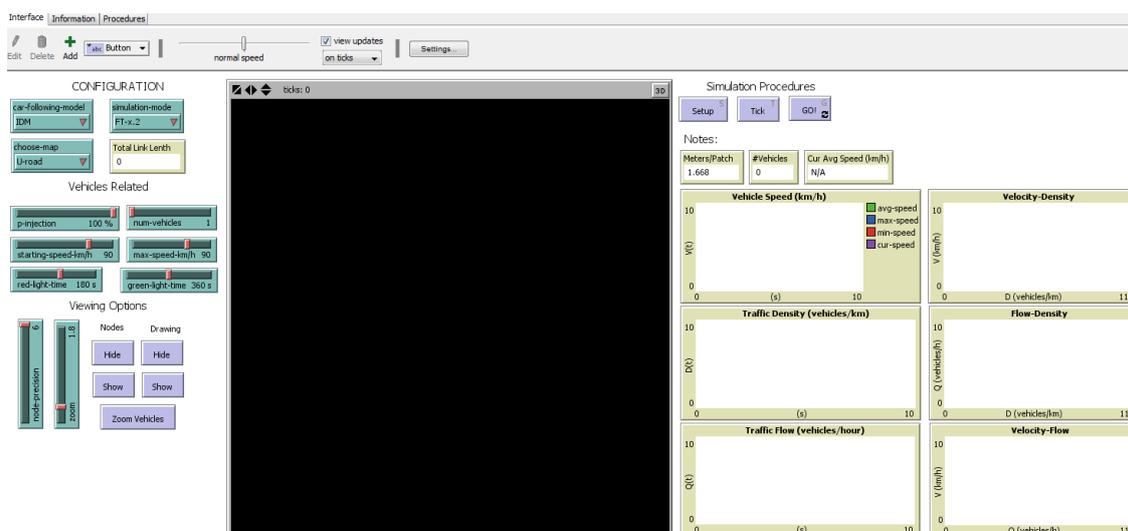


Figura 29 - Novo interface gráfico

Na Figura 29 podemos observar o novo interface gráfico onde ao centro está colocado o monitor de visualização (área cinzenta) que possibilita a observação das movimentações dos agentes no ambiente.

Na Figura 30 encontramos o lado esquerdo do ecrã com três secções de configuração. A primeira diz respeito à escolha do algoritmo de *car-following* e ao tipo de simulação que pretendemos observar. Do mesmo modo é possível escolher o mapa rodoviário que queremos utilizar. Na secção abaixo podemos definir e configurar algumas opções relacionadas com os veículos. É possível definir uma percentagem aceitável para a injeção de veículos por instante de tempo, assim como o número de veículos gerados por instante de tempo. O utilizador também pode definir a velocidade de injeção do agente e a velocidade máxima permitida na via. Os dois últimos *sliders* são específicos para visualizar o comportamento dos agentes quando encontram obstáculos (como veremos no próximo capítulo). Na última secção, de carácter mais gráfico, podemos definir a precisão¹² dos *nós*, o *zoom* do mapa e o tamanho visual dos veículos. É possível mostrar ou esconder os *nós* e realçar a rede viária utilizada para a circulação dos veículos.

¹² Resume-se em definir a precisão da representação das coordenadas dos nós, decorrente da conversão da representação em SIG para variáveis de vírgula flutuante

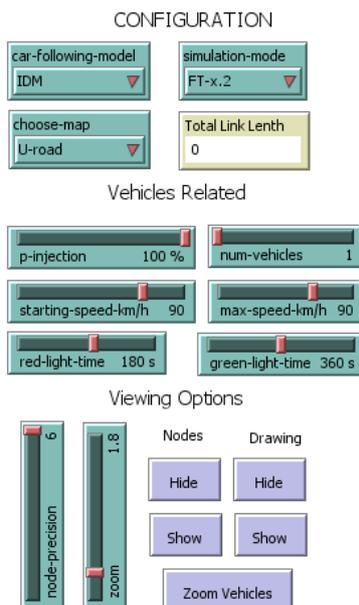


Figura 30 - Secção esquerda do interface em detalhe

Observando a Figura 31, no topo direito do ecrã, encontramos os botões referentes à execução da simulação. O botão *Setup*, à semelhança de outros modelos desta ferramenta, é o responsável pela construção do ambiente de simulação. Os botões de *Tick* e *GO!* executam a simulação mas com comportamentos diferentes. Para o primeiro caso, a simulação é executada segundo a segundo (ou *tick a tick*), enquanto no segundo caso a simulação é executada livremente.

É possível observar algumas características do modelo através de três monitores: o primeiro apresenta o rácio entre metros e *patches* do mundo Netlogo; o segundo contabiliza o total de veículos em estrada naquele instante de tempo; e o terceiro apresenta a velocidade média dos veículos.

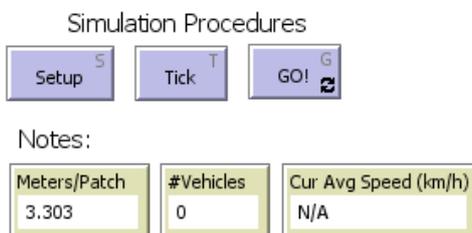


Figura 31 - Topo direito do ecrã em detalhe

4.4. Sinopse

Com este capítulo conclui-se a especificação dos algoritmos e dos modelos de *car-following*. Na fase inicial deste capítulo, realizou-se uma descrição sucinta das diversas ferramentas disponíveis, avaliando as suas características de acordo com as necessidades estabelecidas, tendo-se optado pelo Netlogo.

Através da descrição da estrutura usada para o desenvolvimento do modelo de simulação, foram apresentados os vários subprocessos de desenvolvimento. Começou-se por explicar como são lidos os dados provenientes dos SIG e como são transformados e utilizados para a construção do ambiente de simulação. Ilustrou-se o modelo e o ciclo de vida dos agentes, a implementação dos algoritmos de *car-following* e a lógica subjacente aos mecanismos de geração e injeção de agentes nas simulações. Finalmente, descreveu-se o interface gráfico do ambiente.

CAPITULO 5

Testes e Análise de Resultados

Neste capítulo apresentam-se os testes e os resultados do ambiente construído. São apresentadas as configurações de cada teste realizado e as respectivas métricas a verificar. São testados os componentes de velocidade e aceleração dos agentes de forma individual e em grupo, promovendo a interação com outros agentes. Para além da análise microscópica é realizada ainda uma análise macroscópica ao fluxo de tráfego gerado pela interação dos agentes. Os resultados obtidos são comparados entre si e com outros ensaios de forma a serem validados.

5.1. Especificação do percurso, parametrizações e abordagem para validação

As experiencias efectuadas usam um mapa rodoviário especificado através de um SIG, desenhado para responder às especificações pretendidas. Delimitou-se que o percurso deveria ser de sentido único. Para esse efeito, foram construídos diversos mapas sendo o escolhido um percurso em formato “U” com cerca de 1 quilómetro, ilustrado na Figura 32.

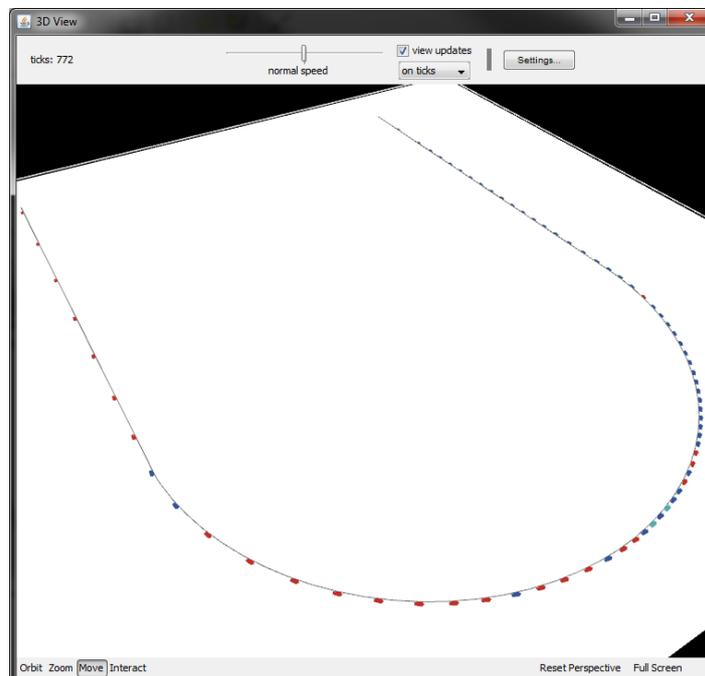


Figura 32 - Percurso rodoviário simulado. Os veículos circulam à direita do *link* e no sentido contra horário.

5.1.1. Parametrizações do modelo

Os testes pretenderam avaliar a capacidade dos algoritmos reproduzirem em Netlogo comportamentos próximos da realidade. Os algoritmos foram testados em condições diversas e os comportamentos obtidos comparados em cenários semelhantes. Para a parametrização dos testes, foram utilizados os valores indicados na Tabela 4.

Modelo Geral					
Velocidade máxima: (V_{Max} , V_0)	Valores diversos.	Distância mínima para veículo da frente: (S_{Min} , S_0)	2 Metros.	Distância observável:	$15 \times V_t$ (m/s).
MAS/STB					
Aceleração lenta: ($Low-Accel$)	0,37 m/s ² .	Máxima aceleração: ($Max-Accel$)	2,78 m/s ² .	Velocidade média: ($Max-Accel$)	12,19 m/s, ≈ 43 km/h.
IDM					
Taxa de aceleração confortável: (A)	1 m/s ⁵ .	Taxa de desaceleração confortável: (B)	2 m/s ² .	Distância temporal: (T)	1,5 Segundos.

Tabela 4 - Parametrização do modelo e algoritmos

As parametrizações escolhidas para os algoritmos MAS/STB e IDM respeitam os valores sugeridos pelos seus autores. No primeiro caso, os valores baseiam-se em trabalhos empíricos de Treiterer (1975) tendo sido explicados anteriormente na secção 3.3.1. Para o caso do IDM, os valores utilizados são considerados realistas pela literatura e são igualmente sugeridos pelos autores nos seus trabalhos originais. Da mesma forma, a distância mínima para o veículo da frente e a distância observável pelo condutor à sua frente respeitam valores considerados realistas na literatura.

5.1.2. Validação de resultados

De modo a validar os resultados obtidos, optou-se, em alguns casos, por utilizar um modelo estudado por (Snare, 2002). O trabalho deste autor aborda o estudo de um modelo dinâmico capaz de prever taxas de aceleração máximas para veículos de passageiros, devidamente validado com dados reais. Procurou-se também reunir dados reais através da publicação

automobilística AutoMotor¹³ que, gentilmente disponibilizou os dados necessários obtidos nos seus testes de estrada.

5.2. Análise de resultados

Os resultados produzidos pelos algoritmos estão agrupados em três conjuntos correspondentes às três fases de teste efectuadas, onde são comparados entre si. São reunidos e analisados os resultados dos componentes de velocidade e aceleração dos agentes de forma individual e em grupo, assim como uma análise macroscópica.

5.2.1. Fase 1: Análise ao comportamento isolado de um agente

Como podemos ver pela Tabela 5, a primeira fase de testes pretende analisar o comportamento isolado de um agente em três cenários distintos, enquanto este permanecer na via de trânsito.

Referência	Descrição
FT-1.1	<u>Teste à capacidade de aceleração de um veículo</u> – o agente é inserido na via imobilizado. Este teste servirá para avaliar o realismo do comportamento de aceleração do agente ao atingir a velocidade desejada;
FT-1.2	<u>Teste à capacidade de travagem de um veículo</u> – o agente terá de ter em atenção a existência de um obstáculo (podemos imaginar a existência de um semáforo) e, caso lhe seja possível, travar. A obstrução permanecerá activa durante um determinado tempo, permitindo ao veículo que retome a sua marcha quando deixar de existir;
FT-1.3	<u>Reacções de um veículo ao entrar na via em excesso de velocidade</u> – serão observadas as reacções do agente ao reduzir a velocidade de forma a manter o limite máximo permitido;

Tabela 5 – Primeira sequência de testes destinada ao comportamento de um único veículo

As condições impostas para os testes acima citados tiveram em conta as métricas que se pretendem avaliar e as condições iniciais pretendidas. Ao longo deste capítulo, as condições são descritas no início de cada teste.

¹³ Vide *homepage* da AutoMotor, <http://www.automotor.xl.pt/>

Teste à capacidade de aceleração de um veículo

Os componentes velocidade e aceleração dos veículos formam o cerne do *car-following*. É imprescindível garantir que os valores calculados demonstrem validade e realismo, por forma a poderem ser utilizados em plataformas de simulação que busquem exactidão nos seus resultados.

Referência	Condições Iniciais	Métricas de Avaliação	Condição de Paragem	Número de Repetições
FT-1.1	Velocidade inicial: 0 km/h; Velocidade máxima: 50, 90, 120 km/h;	Velocidade; Aceleração; Distância Percorrida;	Veículo percorreu a totalidade do percurso;	5

Tabela 6 - Condições iniciais para o 1º teste

Nesta primeira experiência observa-se que o IDM é o algoritmo que demora menos tempo a atingir o final do percurso quando a velocidade máxima é superior a 50 km/h, como podemos ver pela Figura 33.

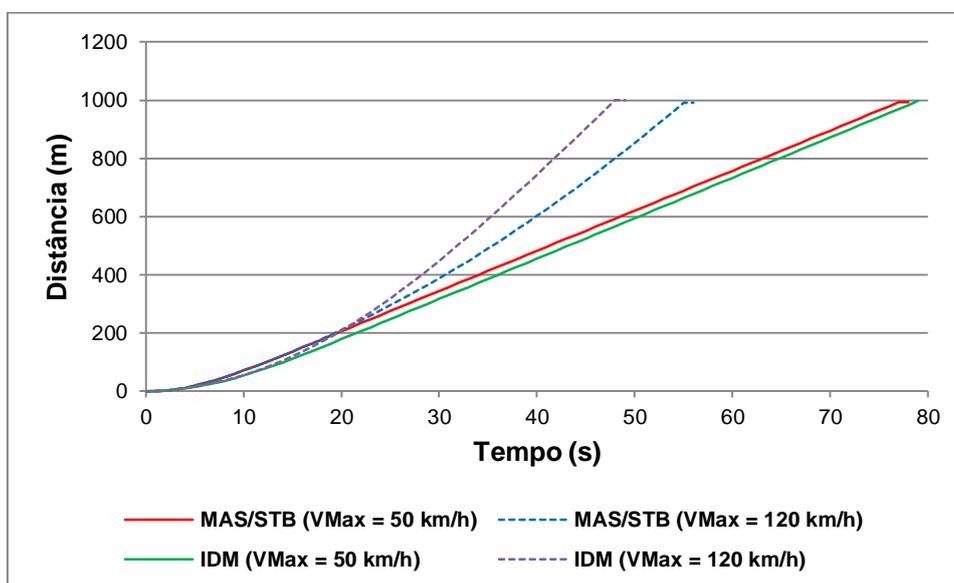


Figura 33 - Tempo que um agente isolado demora a percorrer o percurso

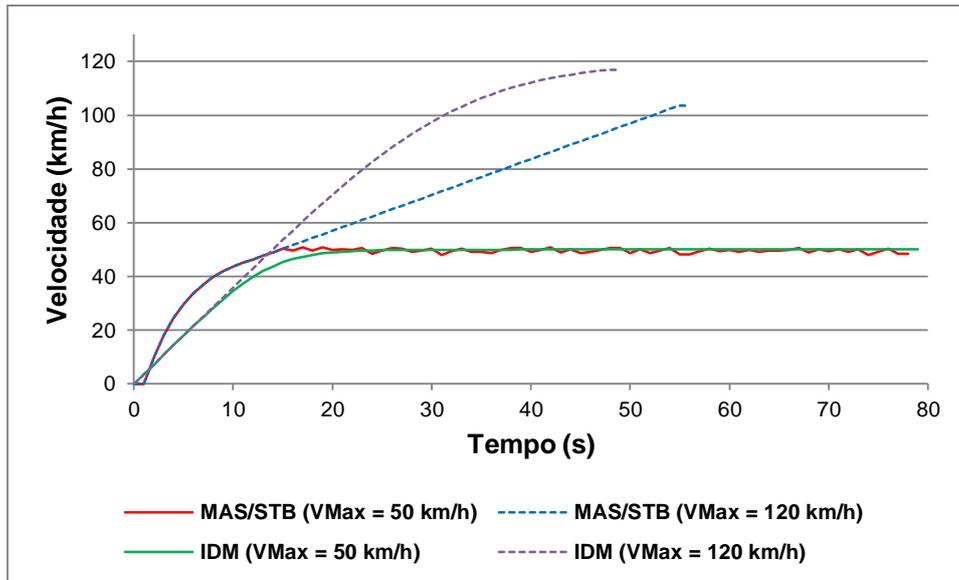


Figura 34 - Velocidade instantânea de um veículo isolado

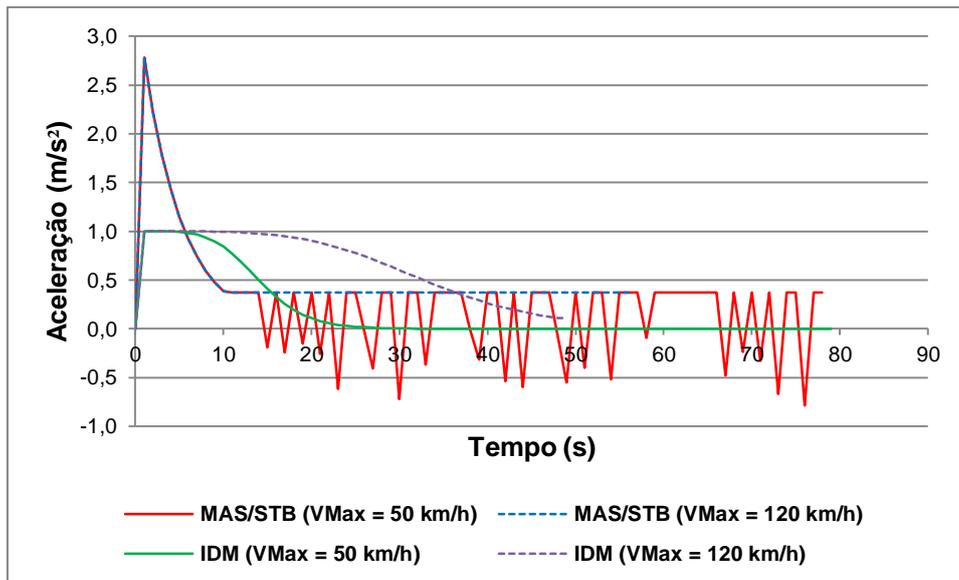


Figura 35 – Aceleração instantânea de um veículo isolado

Este resultado decorre da abordagem adoptada pelos algoritmos em situações de velocidade elevada. Observando a velocidade produzida a partir de uma posição inanimada em ambos os algoritmos (Figura 34) constatamos que o MAS/STB alcança uma velocidade mais elevada nos primeiros dez segundos, fruto do seu método de cálculo. Este comportamento observa-se em todas as experiências realizadas, i.e., independentemente da velocidade máxima permitida o MAS/STB imprime nos veículos um comportamento intenso de arranque, mas realista. O

mesmo se observa com o IDM, embora de uma forma mais gradual e menos intensa. Numa fase posterior, é possível verificar que o MAS/STB, quando atinge a velocidade máxima (50 km/h), demonstra uma certa irregularidade na manutenção da velocidade permitida. Pelo contrário, o IDM demonstra uma tendência para se nivelar de forma estável com a velocidade máxima permitida, resultando numa velocidade constante.

No caso da experiência com $V_{Max} = 120$ km/h, a velocidade do MAS/STB, após um arranque inicial mais intenso, aumenta de forma praticamente constante. No IDM o comportamento é similar, pois o veículo tem um aumento mais intenso de velocidade apesar de constante, independentemente da velocidade instantânea a que o veículo circula, nivelando-se ao aproximar da velocidade pretendida.

Para entender estes resultados, mostra-se útil analisar o gráfico da aceleração da Figura 35. Importa também recordar algumas das características dos algoritmos que influenciam as taxas de aceleração dos veículos. No caso do MAS/STB, os seus autores consideraram um valor máximo de aceleração dos veículos quantificado em $2,78$ m/s², visível na aceleração inicial. Este valor explica o ponto mais elevado do gráfico da Figura 35, comum às duas experiências com o MAS/STB. No caso do IDM, os autores consideraram igualmente um valor que limita a aceleração dado por uma taxa de aceleração confortável limitada a 1 m/s². Em ambos os casos, estes parâmetros limitam a intensidade da aceleração inicial realizada pelos veículos.

Contudo, após algum tempo com a velocidade a aumentar observam-se outros comportamentos distintos. Após o pico inicial, o MAS/STB obtém um rápido decréscimo da aceleração até atingir um valor constante de $0,37$ m/s², correspondente a um outro valor definido pelos autores para os casos em que os veículos circulam acima da velocidade média de 42 km/h. Ultrapassando a velocidade máxima, o MAS/STB é obrigado a travar com uma intensidade máxima de 1 m/s² (correspondente ao comportamento de *Limite de Velocidade*), o que provoca os picos visíveis no final do gráfico.

No caso do IDM a dinâmica produzida por este algoritmo provoca uma (des)aceleração muito mais gradual e suave, observável na Figura 35. Com efeito, o algoritmo beneficia de uma dinâmica extremamente adaptativa, permitindo que o veículo mantenha a velocidade máxima num regime constante.

Claramente, as diferenças observadas nas dinâmicas dos algoritmos decorrem da sua própria natureza. No MAS/STB, a flexibilidade de representação de diversos estados cognitivos dos veículos, através da utilização de regras, provoca alterações mais abruptas nos regimes de velocidade e aceleração dos veículos em circulação. Por outro lado, a menor expressividade de representação do algoritmo IDM é ganha à custa de uma adaptação mais suave e dinâmica nas condições de velocidade e aceleração dos veículos.

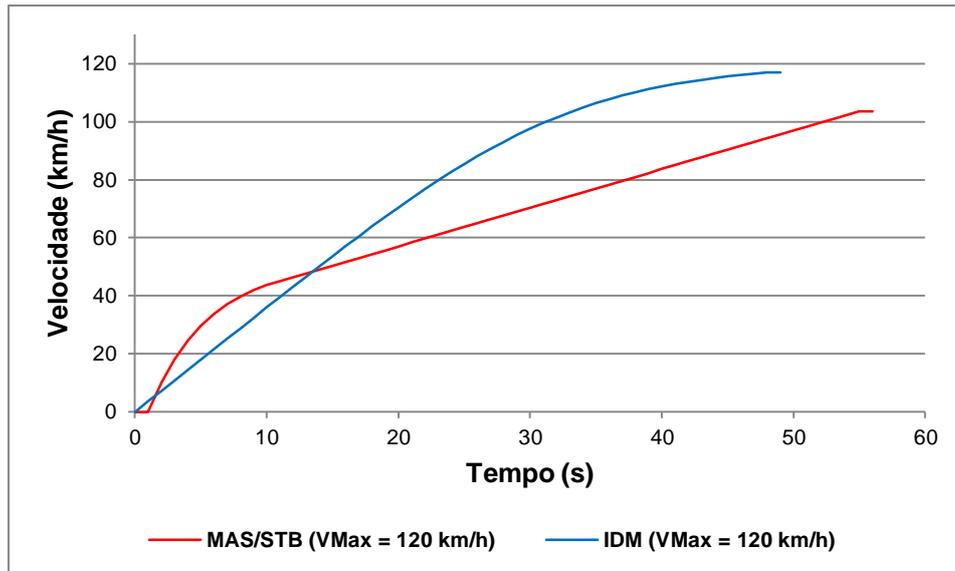


Figura 36 - Velocidade instantânea de um veículo isolado em detalhe

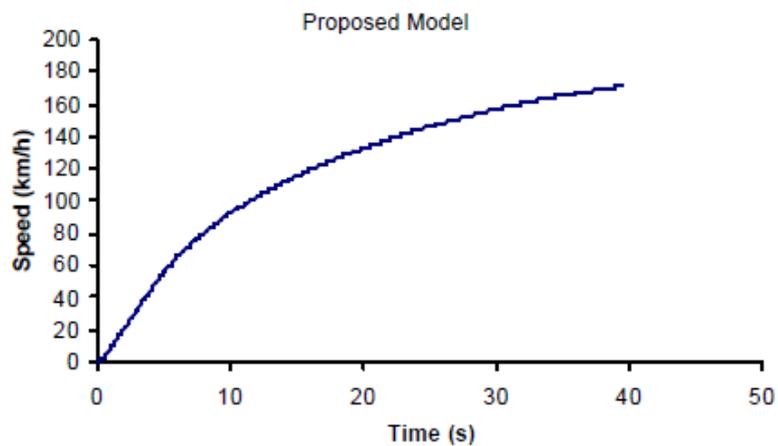


Figura 37 - Velocidade instantânea atingida pelo modelo Rakha, et al. (2001) e validado por Snare (2002) em aceleração livre

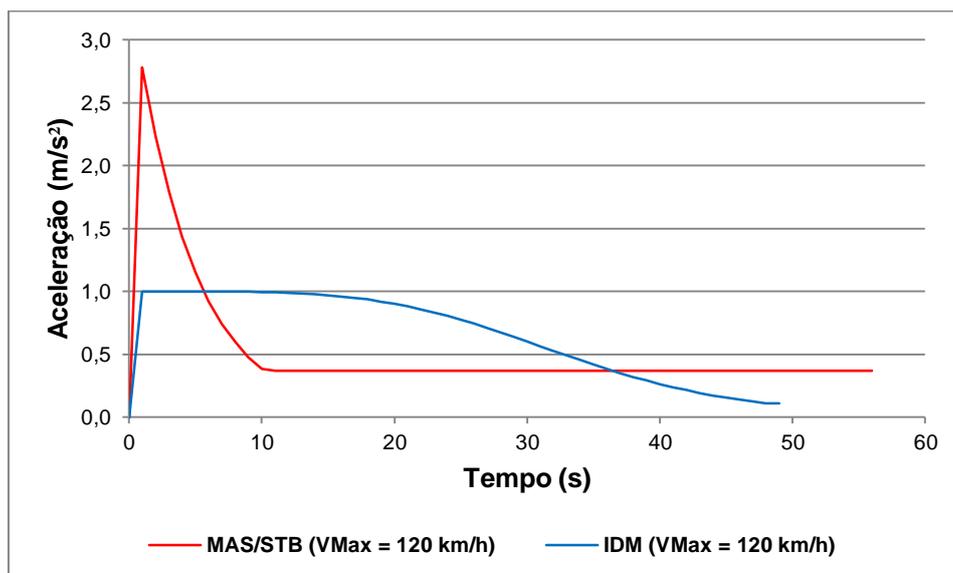


Figura 38 - Aceleração instantânea de um veículo em detalhe

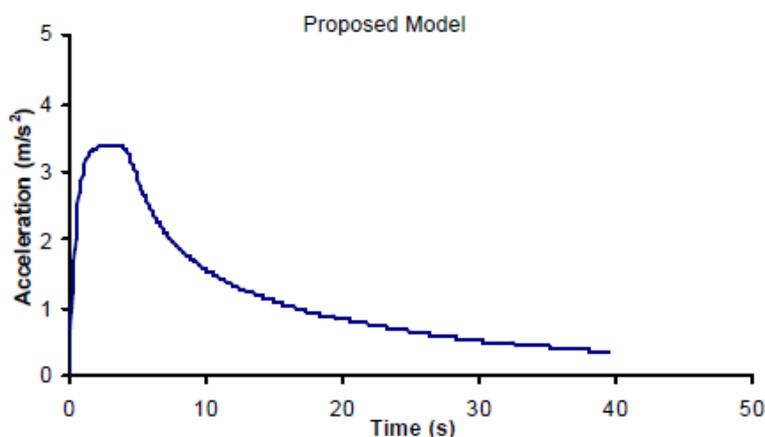


Figura 39 - Aceleração instantânea atingida pelo modelo Rakha, et al. (2001) e validado por Snare (2002) em aceleração livre

Interessa agora comparar estes resultados com outros modelos na literatura. Os valores sugeridos por Snare (2002), apresentados na Figura 37 e Figura 39 representam um modelo de previsão de comportamento de aceleração máxima construído por Rakha, et al. (2001) e validados por Snare através de experiências reais. Neste caso a aceleração máxima de um veículo inicialmente parado durante um percurso de 335 metros. Como se esperava, comparando com os valores produzidos pelos algoritmos em, Figura 36, podemos concluir que os algoritmos demoram bastante mais tempo a atingir valores de velocidade a rondar os 100 km/h. Este facto leva-nos a considerar que ambos os algoritmos estão aptos a exemplificar um tipo de condução menos agressiva, mais cómoda e comum. As mesmas conclusões podem ser deduzidas quando comparamos estes resultados com os dados da Tabela 7 disponibilizados

pela AutoMotor. Nos testes da AutoMotor os veículos necessitaram, em média, de apenas 14,4 segundos para conseguirem atingir os 100 km/h, um valor semelhante ao validado por Snare (2002). Os algoritmos, por sua vez, precisam de cerca de 30 segundos (IDM) a 52 segundos (MAS/STB) em média para atingirem os 100 km/h, confirmando que o tipo de dinâmica definido por ambos os algoritmos não é agressiva.

Modelo	Medições (0 km/h - 100 km/h)
<i>Smart Fortwo 0.8 cdi Pulse</i>	17,5 s
<i>Peugeot 206+ Trendy 1.1</i>	16,9 s
<i>Ford Focus 1.4 Trend</i>	16,2 s
<i>Toyota Yaris 1.0 VVT-i</i>	15,4 s
<i>Opel Corsa 1.2 Enjoy</i>	14,9 s
<i>Seat Ibiza 1.2 Good Stuff</i>	14,6 s
<i>Citroën C3 1.4 HDi Séduction</i>	14,5 s
<i>Ford Fiesta 1.25i Trend</i>	14,4 s
<i>Renault Clio III 1.2 16v Dynamique S</i>	14,4 s
<i>Volkswagen Polo 1.2 Highline</i>	14,3 s
<i>Honda Civic 1.4 Confort</i>	14 s
<i>Fiat Grande Punto 1.3 MTJ Free (90cv)</i>	13,3 s
<i>Renault Mégane III 1.4 TCE Dynamique</i>	12,7 s
<i>Citroën C4 1.6 eHDi (110cv)</i>	12,6 s
<i>Volkswagen Golf VI Variant 1.6 TDI</i>	11,8 s

Tabela 7 - Medições da capacidade de aceleração feitas pela AutoMotor.

Por último, analisando a Figura 38 e a Figura 39 observa-se que os regimes de velocidade e aceleração validados por Snare (2002) se assemelham a um misto dos dois algoritmos. Possui uma aceleração mais intensa que o MAS/STB, e ao atingir o valor máximo mantém-no por algum tempo (à semelhança do IDM). Posteriormente, a desaceleração possui uma curvatura mais próxima da curvatura do MAS/STB, quando comparada com o IDM. Com efeito, nesse período de tempo o MAS/STB realiza uma diminuição da aceleração de forma mais rápida e intensa do que o IDM.

Teste à capacidade de travagem de um veículo

Outra componente testada em ambos os algoritmos refere-se ao comportamento de desaceleração face a obstáculos. De forma a averiguar o realismo dos valores obtidos, iremos utilizar dois métodos: a fórmula da distância de paragem em segurança (Escola de Condução Invicta, 2011) e dados disponibilizados pela AutoMotor.

Referência	Condições Iniciais	Métricas de Avaliação	Condição de Paragem	Número de Repetições
FT-1.2	Intervalos de velocidade: [50-50 km/h], [40-120 km/h], [80-90 km/h]; Tempo de obstrução: 90s; Tempo de passagem: 120s;	Velocidade; Aceleração; Distância Percorrida;	Veículo percorreu a totalidade do percurso;	5

Tabela 8 - Condições iniciais para o 2º teste

A fórmula da distância de paragem permite-nos ter uma ideia da “distância total percorrida entre o momento em que o condutor vê o obstáculo e aquele em que o veículo pára”:

$$d_{Paragem} = d_{Reacção} + d_{Travagem}, \quad (5.1)$$

$$\text{em que } d_{Reacção} = v_{(km/h)}/3.6 \quad (5.2)$$

$$\text{e } d_{Travagem} = v_{(km/h)}^2/100. \quad (5.3)$$

A distância de reacção define a distância que o veículo percorre até o condutor reagir ao obstáculo. A distância de travagem define a distância percorrida entre o momento em que o condutor começa a travar até ao momento em que o veículo pára. Na medida em que os algoritmos testados não consideram o tempo da reacção do condutor iremos apenas comparar os valores produzidos com a fórmula da distância de travagem.

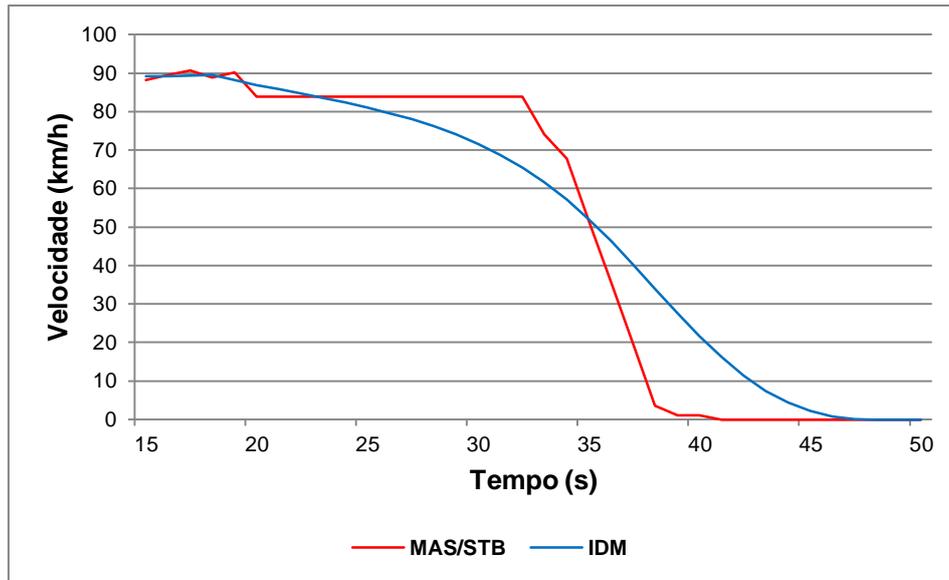


Figura 40 - Velocidade dos agentes em travagem para $V_{Max} = 90$ km/h.

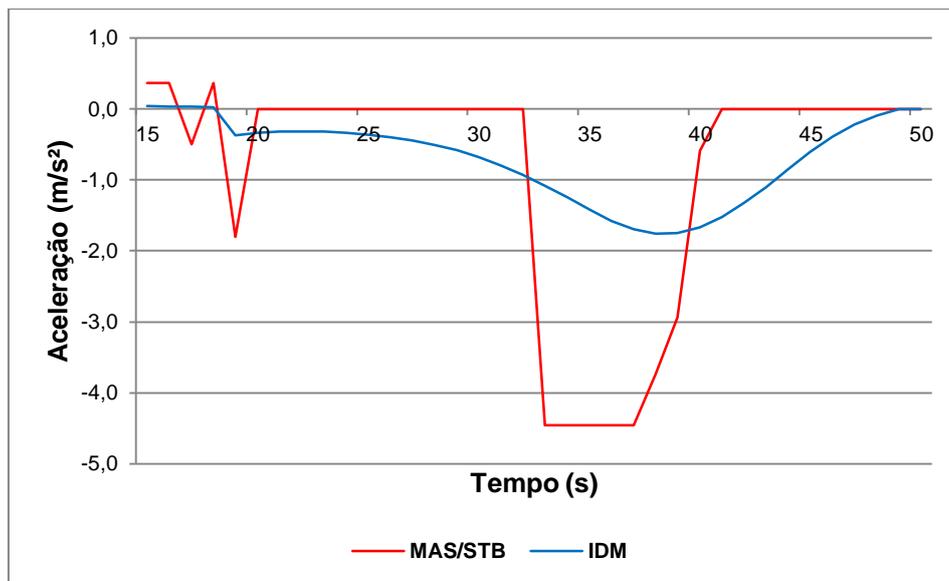


Figura 41 - Aceleração dos agentes em travagem para $V_{Max} = 90$ km/h.

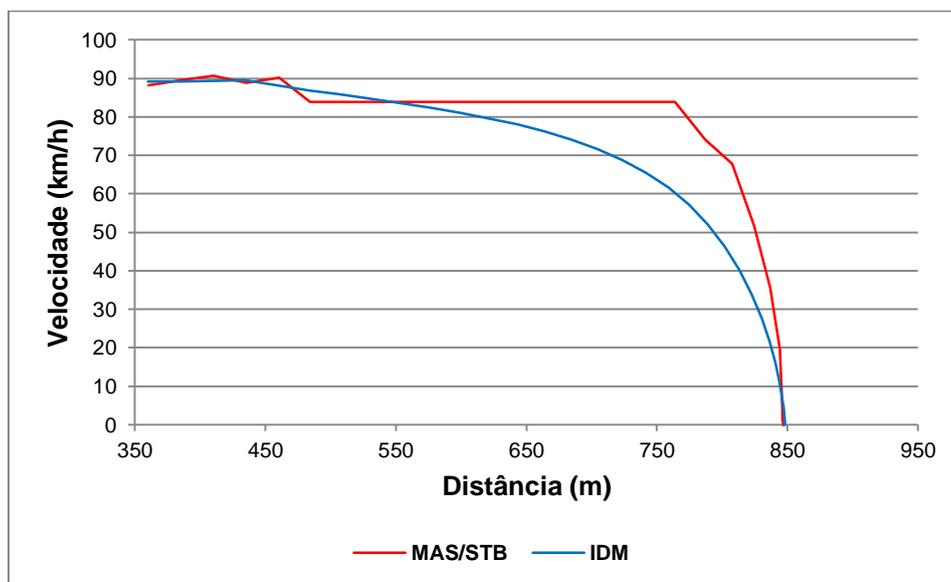


Figura 42 - Desaceleração do agente em função da velocidade e da distância percorrida

A 90 km/h, ambos os algoritmos detectam o obstáculo a cerca de 390 metros. Como podemos observar pela Figura 40 e Figura 41, com a velocidade de um veículo nos 90 km/h, o IDM começa a desacelerar de forma gradual a partir do momento em que detecta o obstáculo. O MAS/STB por sua vez mantém a sua velocidade até que o obstáculo entre na sua área de segurança, o que o leva a executar uma travagem mais intensa. Se analisarmos o componente da velocidade em função da distância percorrida (Figura 42) verificamos que o MAS/STB necessita de 86 metros para parar. O IDM, por sua vez, efectua a sua desaceleração de forma gradual e confortável. Em suma, esta é a grande diferença nos dois algoritmos. As reacções do IDM iniciam-se assim que este algoritmo detecta o obstáculo e adopta um comportamento adaptativo. O MAS/STB, por seu turno, apesar de identificar o obstáculo na mesma altura que o IDM, espera até que este entre na sua área de segurança e só então adopta um regime de desaceleração adequado.

De seguida iremos observar as medições reais de veículos analisados pela AutoMotor. Como podemos ver na Tabela 9 os testes efectuados foram feitos em veículos a circular a 90 km/h, em diferentes condições de teste, isto é, diferentes condições de piso, estado de pneus, meteorologia, etc.

Modelo	Medições (90 km/h - 0 km/h)
<i>Peugeot 206+ Trendy 1.1</i>	34 m
<i>Fiat Grande Punto 1.3 MTJ Free (90cv)</i>	34 m
<i>Smart Fortwo 0.8 cdi Pulse</i>	32 m
<i>Ford Focus 1.4 Trend</i>	32 m
<i>Opel Corsa 1.2 Enjoy</i>	31 m
<i>Seat Ibiza 1.2 Good Stuff</i>	31 m
<i>Renault Clio III 1.2 16v Dynamique S</i>	31 m
<i>Honda Civic 1.4 Confort</i>	31 m
<i>Renault Mégane III 1.4 TCE Dynamique</i>	31 m
<i>Citroën C4 1.6 eHDi (110cv)</i>	31 m
<i>Toyota Yaris 1.0 VVT-i</i>	30 m
<i>Ford Fiesta 1.25i Trend</i>	30 m
<i>Citroën C3 1.4 HDi Séduction</i>	29 m
<i>Volkswagen Polo 1.2 Highline</i>	29 m
<i>Volkswagen Golf VI Variant 1.6 TDI</i>	29 m

Tabela 9 - Avaliação da distância de paragem segundo as medições efectuadas pela AutoMotor.

Voltando a utilizar a fórmula para a distância de paragem para uma velocidade de 90 km/h obtemos:

$$D_{\text{Travagem}} = 90^2/100 = 81 \text{ metros};$$

Analisando os valores reais provenientes da medição da AutoMotor (média de 31 metros para o veículo parar) e os reportados pela fórmula acima, podemos observar que os valores produzidos pelo MAS/STB (por ser o algoritmo que executa uma travagem mais intensa) são superiores aos dados da AutoMotor, mas próximos da fórmula da distância de travagem. Os veículos que utilizaram este algoritmo conseguiram travar num espaço de 86 metros. O IDM, como da última vez, demonstrou uma travagem mais gradual e confortável, mas de acordo com o expectável para este teste.

Comportamento de um veículo ao circular com excesso de velocidade

O terceiro cenário da primeira fase de testes pretende salientar um pormenor detectado durante a fase de implementação dos algoritmos. Regra geral, os ambientes de simulação normalmente estão limitadas a pequenas áreas físicas, em que as simulações apenas representam pequenos ambientes alusivos a quarteirões, rotundas, avenidas, etc. Contudo, importa averiguar o comportamento dos algoritmos em ambientes mais abrangentes. Por exemplo, como se comportam os algoritmos quando se representam vias com características

semelhantes às vias de desaceleração, tais como as saídas comuns das vias rápidas para estradas com limite de velocidade inferior à via originária.

Referência	Condições Iniciais	Métricas de Avaliação	Condição de Paragem	Número de Repetições
FT-1.3	Velocidade inicial: 90, 120 km/h; Velocidade máxima: 50 km/h;	Velocidade; Aceleração; Distância Percorrida;	Veículo percorreu a totalidade do percurso;	5

Tabela 10 - Condições iniciais para o 3º teste.

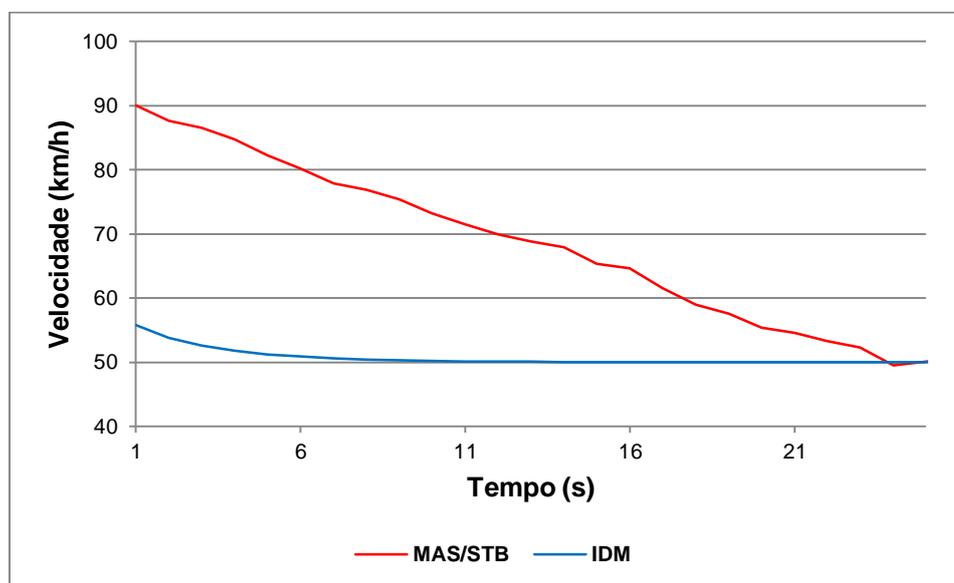


Figura 43 – Velocidade dos agentes quando $V_{Inicial} > V_{Max}$ ($V_{Inicial} = 90$ km/h, $V_{Max} = 50$ km/h).

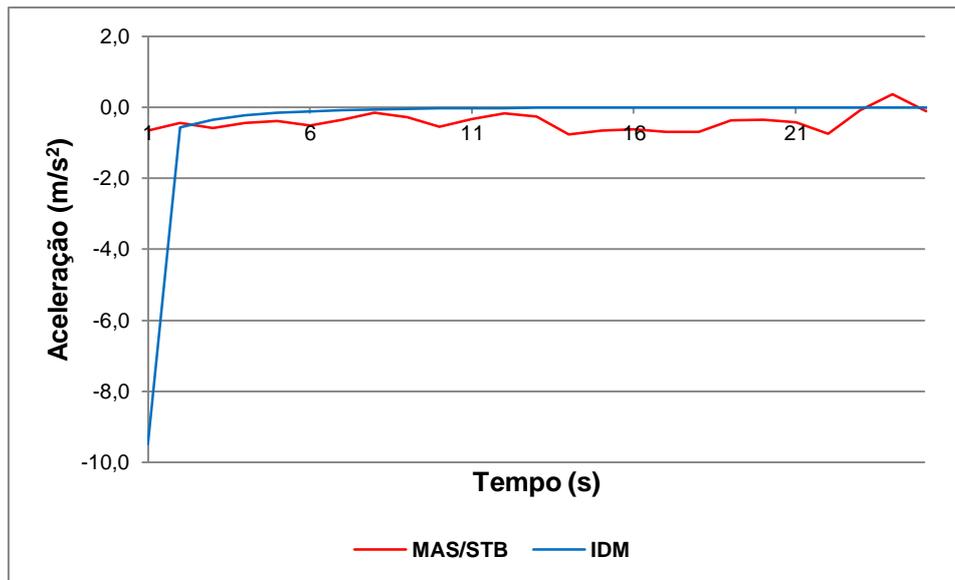


Figura 44 – Aceleração dos agentes quando $V_{\text{Inicial}} > V_{\text{Max}}$ ($V_{\text{Inicial}} = 90 \text{ km/h}$, $V_{\text{Max}} = 50 \text{ km/h}$).

Como podemos observar pelas Figura 43 e Figura 44, apenas o MAS/STB aparenta reproduzir um comportamento real de desaceleração satisfatório ao ser injectado com uma velocidade superior à permitida. O algoritmo executa a regra correspondente (limite de velocidade, analisado na secção 3.3.1) que o leva a reduzir a velocidade de modo a circular à velocidade permitida. O IDM por sua vez, não parece estar preparado para circular acima da velocidade permitida. No momento de injeção do veículo, e devido às características da componente de aceleração da fórmula do IDM (Eq. 3.15) é produzido um regime de desaceleração muito intensa, reduzindo a velocidade do veículo para um valor ligeiramente superior ao máximo permitido em aproximadamente um segundo.

5.2.2. Fase 2: Comportamento dos agentes em fluxo de tráfego

Na segunda fase de testes, serão analisados os resultados gerados por pares de agentes que reagem e produzem estímulos.

Referência	Descrição
FT-2.1	<u>Interacção entre veículos</u> – pretende-se observar o comportamento de circulação obtido pelo par de agentes através da sua reacção aos estímulos do veículo da frente;
FT-2.2	<u>Interacção entre veículos com obstáculos</u> – pretende-se observar os comportamentos reactivos quando existe um semáforo na via;

Tabela 11 – Segunda sequência de testes de comportamento destinada às interacções entre veículos

Interacção entre veículos

De forma a avaliar os comportamentos reactivos dos veículos foram feitas duas análises com um acompanhamento de 60 segundos cada, durante a circulação dos agentes na via. A primeira análise diz respeito ao primeiro par de veículos injectado na via, com o percurso desimpedido. A segunda análise examina um segundo par de veículos injectados posteriormente, quando o fluxo de tráfego atingiu uma estabilidade aceitável.

Referência	Condições Iniciais	Métricas de Avaliação	Condição de Paragem	Número de Repetições
FT-2.1	Velocidade inicial: 40 km/h; Velocidade máxima: 50 km/h;	Velocidade dos pares de veículos; Aceleração dos pares de veículos; Distância percorrida pelos pares de veículos;	5 Minutos	3

Tabela 12 - Condições iniciais do primeiro teste.

As três repetições das experiências resultaram, do ponto de vista qualitativo, em resultados similares. Os gráficos ilustram o comportamento de uma única repetição.

Para esta experiência parametrizou-se o fluxo de injeção para 100%, ou seja, os veículos são injectados sempre que existam condições para tal.

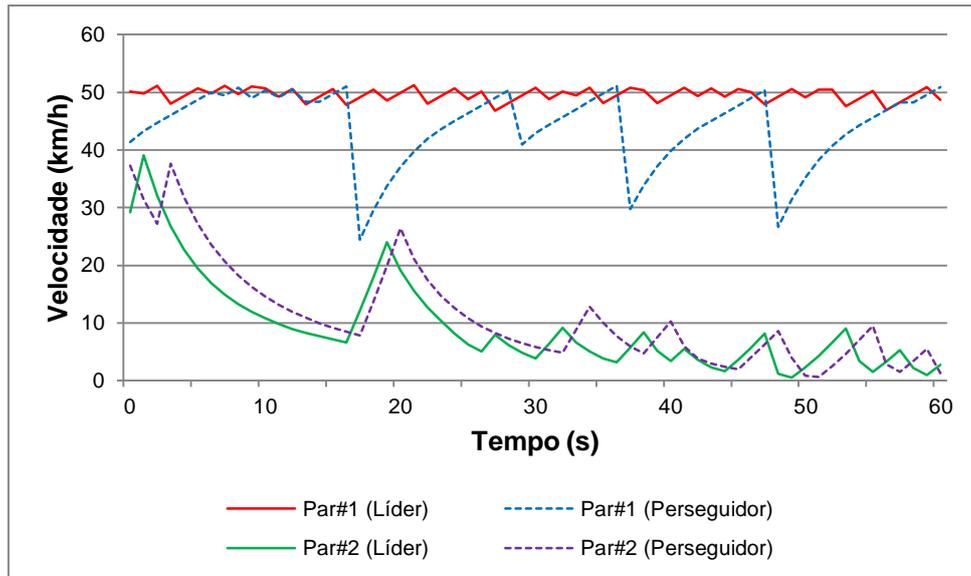


Figura 45 - Velocidade instantânea atingida pelos pares de veículos utilizando o MAS/STB.

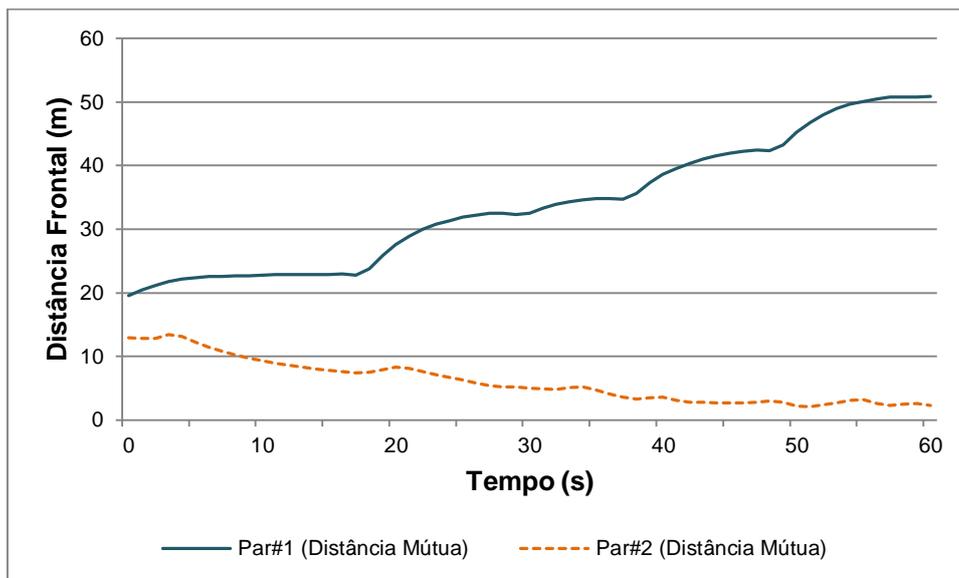


Figura 46 – Distância mútua entre os pares de veículos para o MAS/STB.

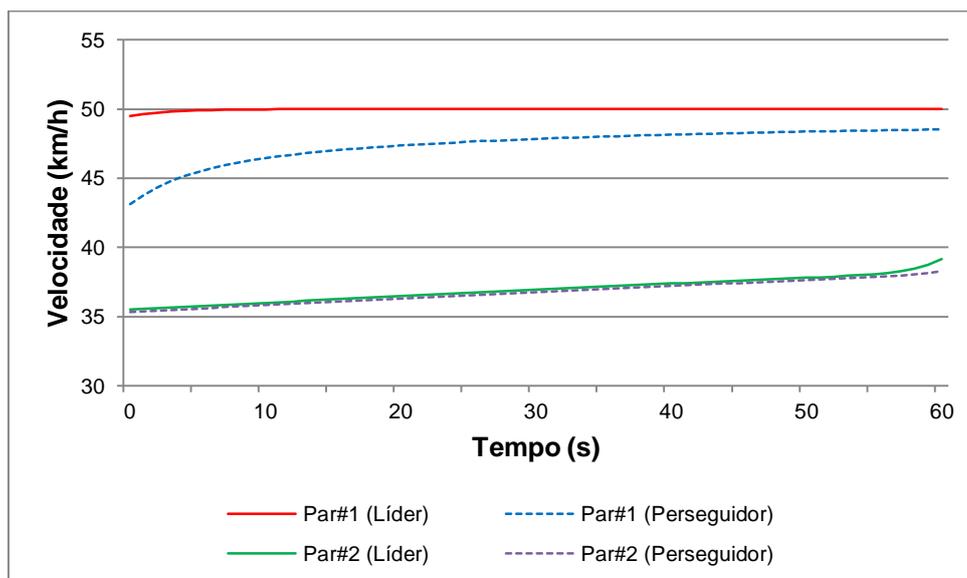


Figura 47 - Velocidade instantânea atingida pelos pares de veículos utilizando o IDM.

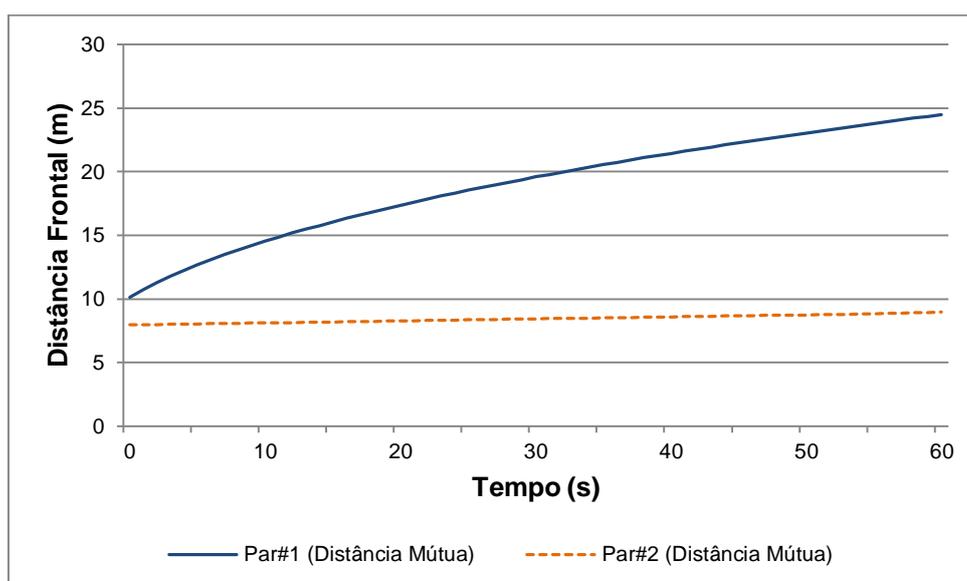


Figura 48 - Distância mútua entre veículos pares para o IDM.

Ao analisar o primeiro par de veículos em ambos os algoritmos observamos uma diferença de comportamento evidente. Focando-nos no MAS/STB (Figura 45) reparamos que o veículo perseguidor efectua travagens fortes devido à acção inconstante do veículo líder que tenta circular à velocidade permitida. Este comportamento leva a que a distância entre veículos aumente com alguma intensidade (Figura 46) aumentando a probabilidade de se formar o chamado “trânsito lento”. O IDM, por seu turno, produz reacções mais suaves na interacção com o veículo líder. Como podemos ver pela Figura 47 o veículo da frente circula a uma velocidade constante, enquanto o veículo perseguidor está a tentar aproximar-se de forma

gradual. Na Figura 48 podemos observar que para o primeiro par de veículos a distância mútua aumenta de forma ligeira apesar da velocidade do perseguidor aparentar um estado de estabilidade. Analisando o segundo par de veículos (Figura 45), observamos que no MAS/STB o fluxo de tráfego encontra-se em circulação lenta. Isto leva a que o par de veículos simule comportamentos de constantes travagens e curtas acelerações. A diminuição da distância mútua é outro efeito que sugere esta conclusão. Em contraste, o IDM impõe um equilíbrio perfeito no segundo par de veículos analisado. As velocidades coincidentes e a distância mútua estabilizada demonstram que a circulação deste fluxo está a ser realizada de forma fluida, mais eficaz do que o MAS/STB.

Comportamento de um par de veículos ao identificar um impedimento na via

Considerou-se interessante observar os comportamentos de um par de veículos quando encontram um obstáculo na via, obrigando-os a travar.

Referência	Condições Iniciais	Métricas de Avaliação	Condição de Paragem	Número de Repetições
FT-2.2	Velocidade inicial: 90 km/h; Velocidade máxima: 90 km/h; Tempo de obstrução: 90s; Tempo de passagem: 120s;	Velocidade dos pares de veículos; Aceleração dos pares de veículos; Distância percorrida pelos pares de veículos;	5 Minutos	3

Tabela 13 - Condições para o segundo teste da segunda fase

Na Figura 49 comparamos os perfis obtidos de velocidade, obtidos 10 segundos após a injeção dos veículos, de modo a diminuir a influência do processo de injeção nos algoritmos.

Observando os valores referentes ao IDM, denota-se um equilíbrio quase perfeito entre os agentes. Ambos efectuem travagens graduais, de forma suave, mantendo uma distância segura mas próxima do veículo da frente. O MAS/STB, como já foi demonstrado anteriormente, trava apenas quando um obstáculo/veículo entra na sua área de segurança. É patente, através da Figura 49 e Figura 50 que o MAS/STB tende a manter uma distância mútua superior ao IDM durante a circulação.

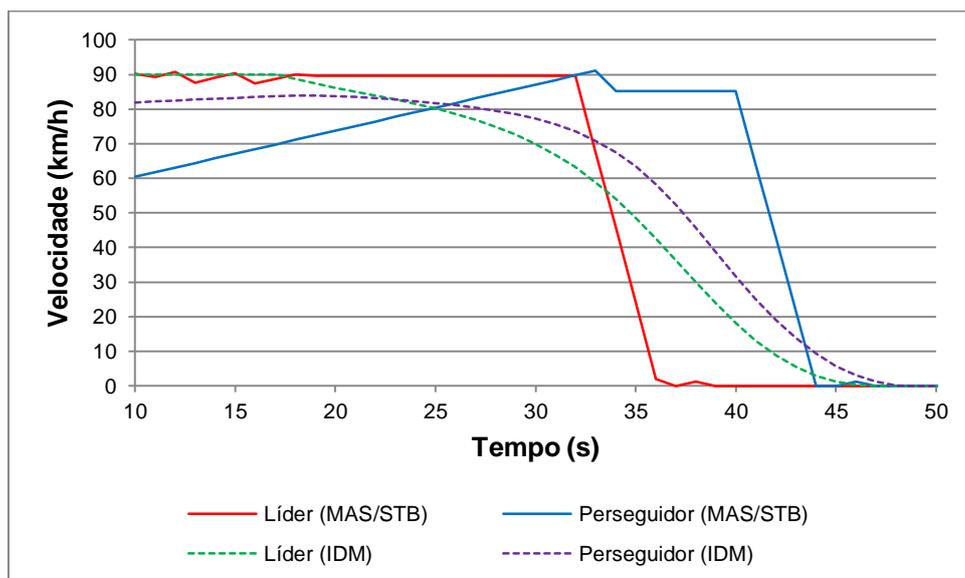


Figura 49 - Desaceleração dos pares de agentes face a um obstaculo.

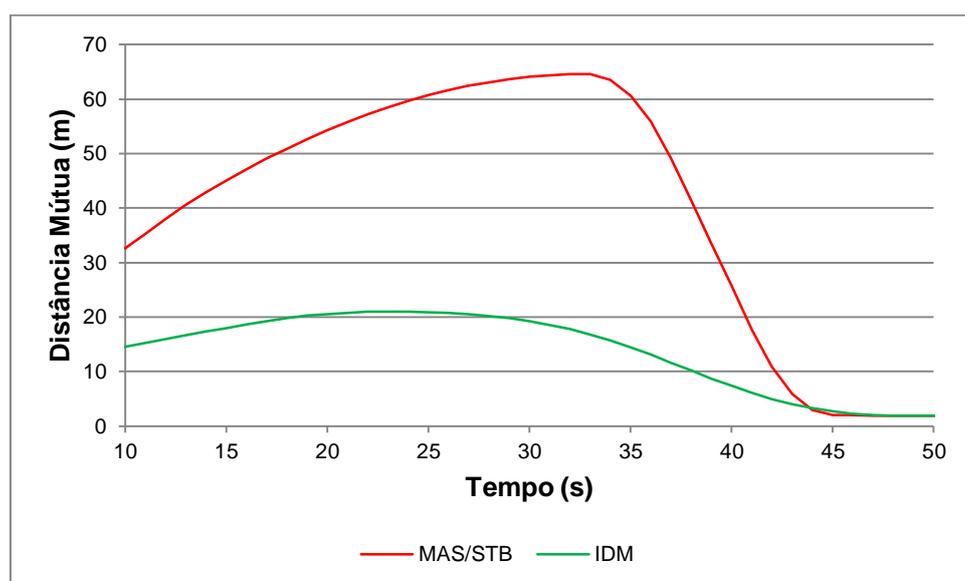


Figura 50 - Distância relativa entre algoritmos

5.2.3. Análise macroscópica de tráfego

A terceira fase de testes pretende realizar avaliações macroscópicas do tráfego automóvel. Serão avaliados os valores da velocidade média, da densidade e do fluxo de tráfego, tal como indicado na Tabela 14.

Referência	Descrição
FT-3.0	Análise macroscópica da velocidade, densidade e fluxo de tráfego num ambiente dinâmico.

Tabela 14 – Terceira fase de testes dedicada à análise macroscópica de tráfego

Apesar das análises macroscópicas serem utilizadas essencialmente para avaliar as capacidades de uma via (Hall, 1999), foi decidido aproveitar as suas características para medir o fluxo, a densidade e a velocidade média de tráfego. Podemos definir o fluxo de veículos como o número de veículos que passam por um determinado local por unidade de tempo, normalmente referido como número de veículos por hora. A densidade, por sua vez, representa o número de veículos por área de estrada normalmente medida através do número de veículos por quilómetro.

Referência	Condições Iniciais	Métricas de Avaliação	Condição de Paragem	Número de Repetições
FT-3.0	Probabilidade de inserção de veículos: 25%, 100%; Velocidade inicial: 90 km/h; Velocidade máxima: 90 km/h;	Velocidade média; Densidade; Fluxo de trânsito; Total de veículos;	Atingir 3 horas de simulação	3

Tabela 15 - Condições para os testes da terceira fase.

Para elaborar a análise é preciso determinar o diagrama fundamental do fluxo de tráfego, que estabelece a relação entre fluxo de tráfego, densidade e velocidade. Este diagrama é calculado através da fórmula:

$$\text{Fluxo } (Q) = \text{Densidade } (D) \times \text{Velocidade } (V). \quad (5.1)$$

Importa definir três gráficos baseados nestes componentes, onde deverá ser possível caracterizar algumas relações:

- Velocidade-Densidade: é de fácil entendimento que quando a densidade de tráfego é baixa, a velocidade tenderá a ser elevada, da mesma maneira que quando a densidade for elevada, a velocidade terá de ser baixa;

- Fluxo-Densidade: relação que pretende demonstrar a capacidade e a condição da estrada em análise, isto é se a estrada consegue produzir fluidez suficiente para a densidade de trânsito que possui;
- Velocidade-Fluxo: relação que demonstra a velocidade ideal para cada nível de fluxo;

De seguida serão analisados três tipos de gráficos obtidos para diferentes parametrizações durante a circulação livre de veículos.

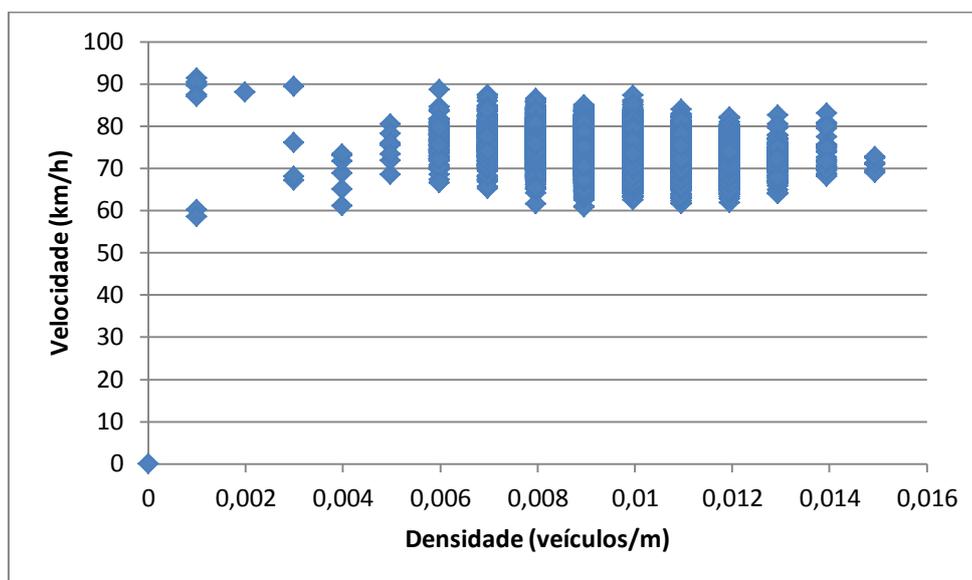


Figura 51 - Velocidade-Densidade para MAS/STB com taxa de injeção de 25%.

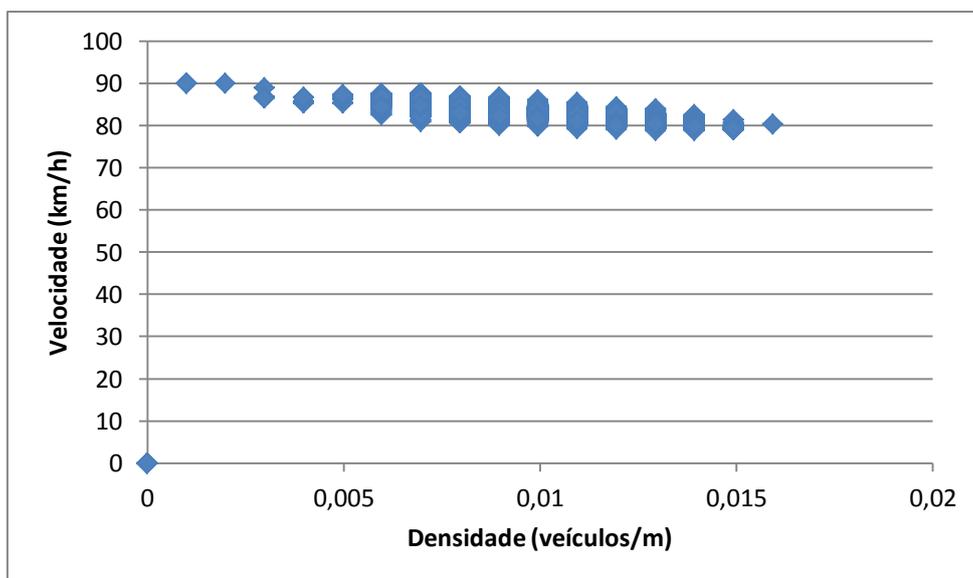


Figura 52 - Velocidade-Densidade para IDM com taxa de injeção de 25%.

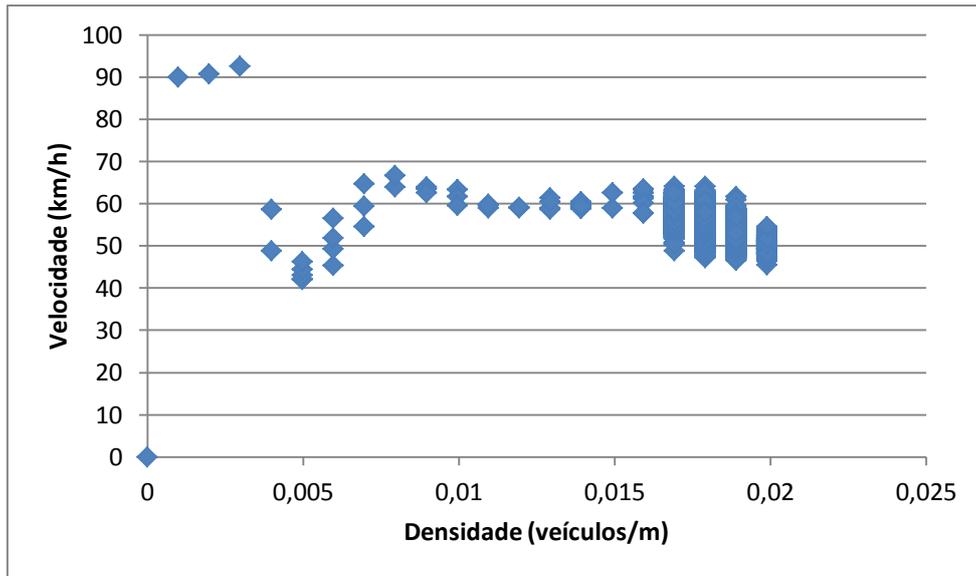


Figura 53 - Velocidade-Densidade para MAS/STB com taxa de injeção de 100%.

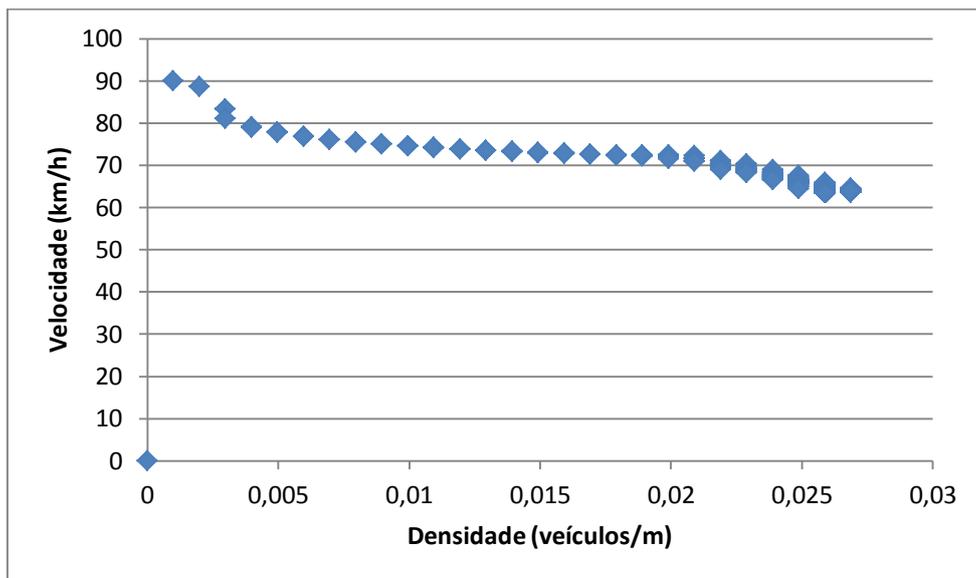


Figura 54 - Velocidade-Densidade para IDM com taxa de injeção de 100%.

Como se pode observar pela Figura 51, Figura 52, Figura 53 e Figura 54 os valores velocidade-densidade produzido pelo IDM são mais lineares e mostram menos dispersão do que os do MAS/STB, em boa parte devido à natureza dos regimes de aceleração do MAS/STB baseado em regras. Por esta razão, os valores de densidade atingidos são também ligeiramente superiores no IDM. No entanto, a maior densidade observada no IDM poder também dever-se à influência indesejada do processo de injeção de veículos no MAS/STB. Com efeito, a natureza mais adaptativa do IDM tende a despachar os veículos da fila de espera de injeção a

uma taxa superior ao MAS/STB, o que pensamos poder estar a influenciar o rigor dos resultados.

Contudo, em ambos os algoritmos é notória a fluidez de tráfego existente apesar de se notar uma ligeira tendência para a velocidade média diminuir com o aumento da densidade.

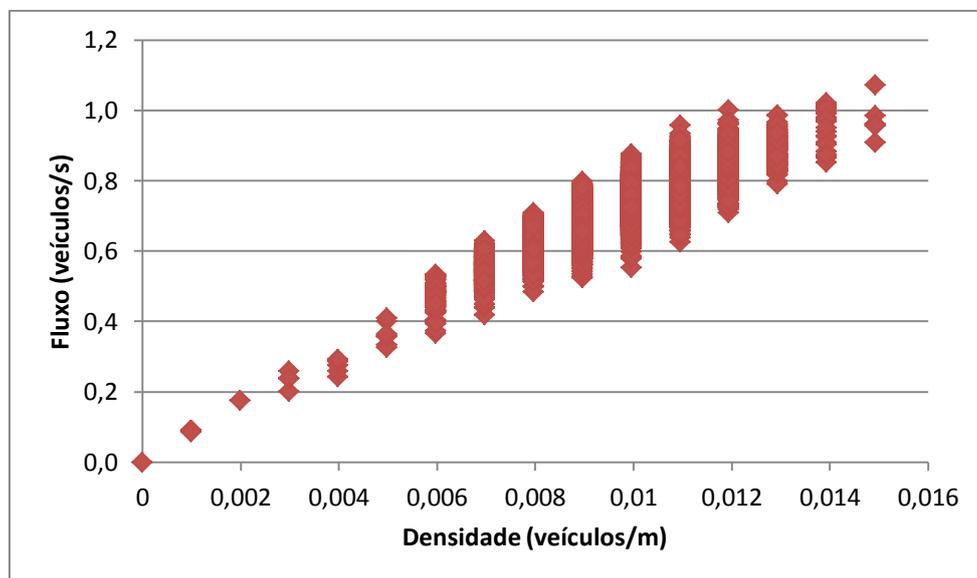


Figura 55 - Fluxo-Densidade para MAS/STB com taxa de injeção de 25%.

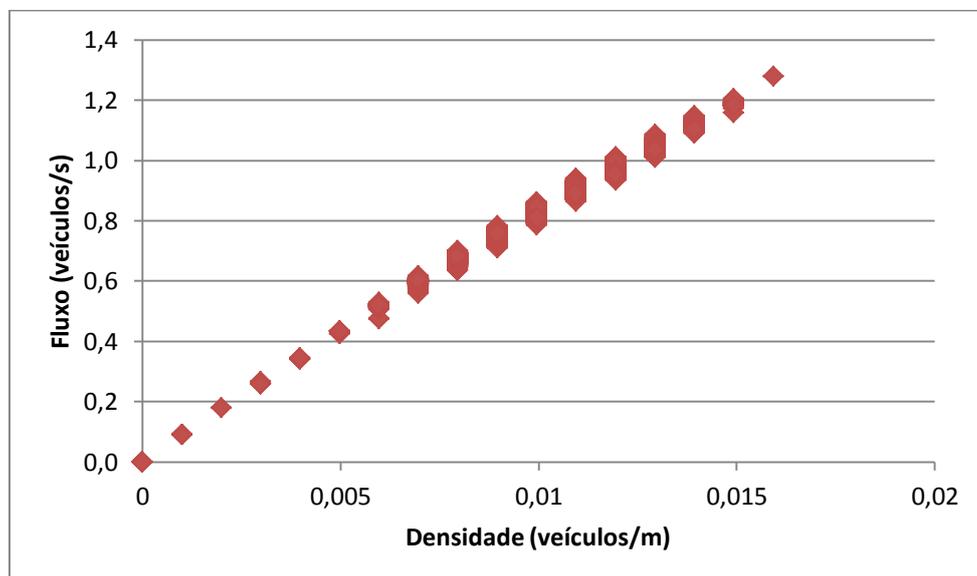


Figura 56 - Fluxo-Densidade para IDM com taxa de injeção de 25%.

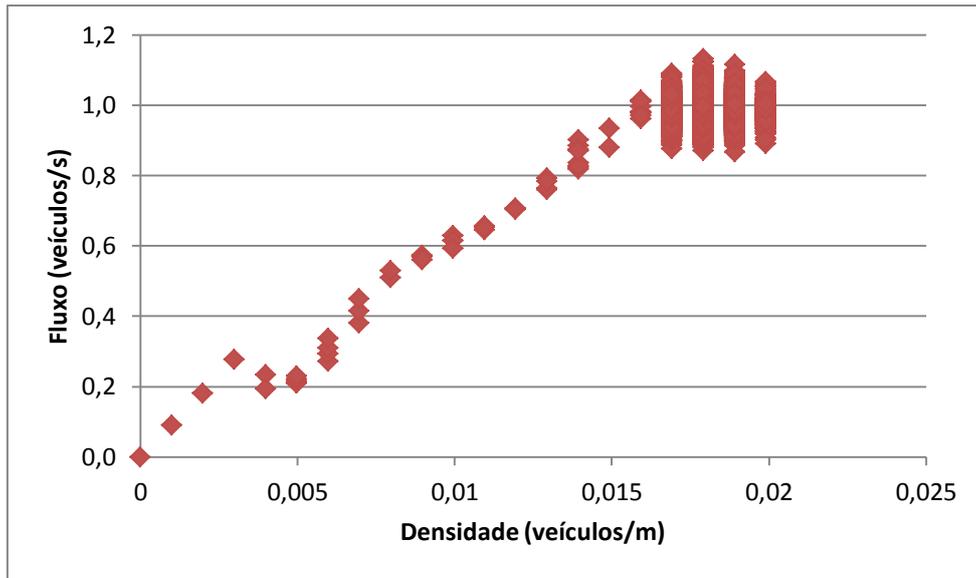


Figura 57 - Fluxo-Densidade para MAS/STB com taxa de injeção de 100%.

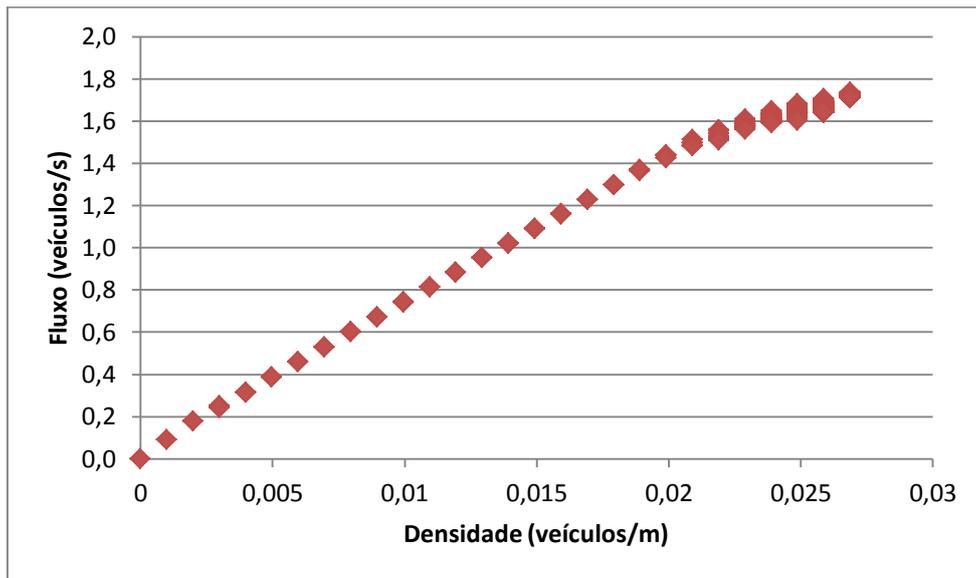


Figura 58 - Fluxo-Densidade para IDM com taxa de injeção de 100%.

Nas figuras Figura 55, Figura 56, Figura 57 e Figura 58 apresentam-se os valores de fluxo de trânsito para cada nível de densidade. Em harmonia com os gráficos que relacionam a velocidade-densidade, o IDM produz valores de fluxo de tráfego superiores e em intervalos menores. O equilíbrio produzido pelo IDM e a maior compatibilidade com o método de injeção fazem com que a dispersão de valores obtidos seja inferior. Geralmente, o gráfico produz uma função com dois ramos. O primeiro ramo inicia-se na origem e segue um sentido crescente até atingir um ponto de fluxo máximo. Atingido este ponto, inicia o outro ramo num sentido decrescente. Nos casos apresentados a simulação gerou quase sempre situações em que a

via não atingiu a sua capacidade máxima. Apenas a Figura 57 demonstra um estado em que a via fica perto de atingir a capacidade máxima, provocando uma tendência ligeiramente decrescente do fluxo.

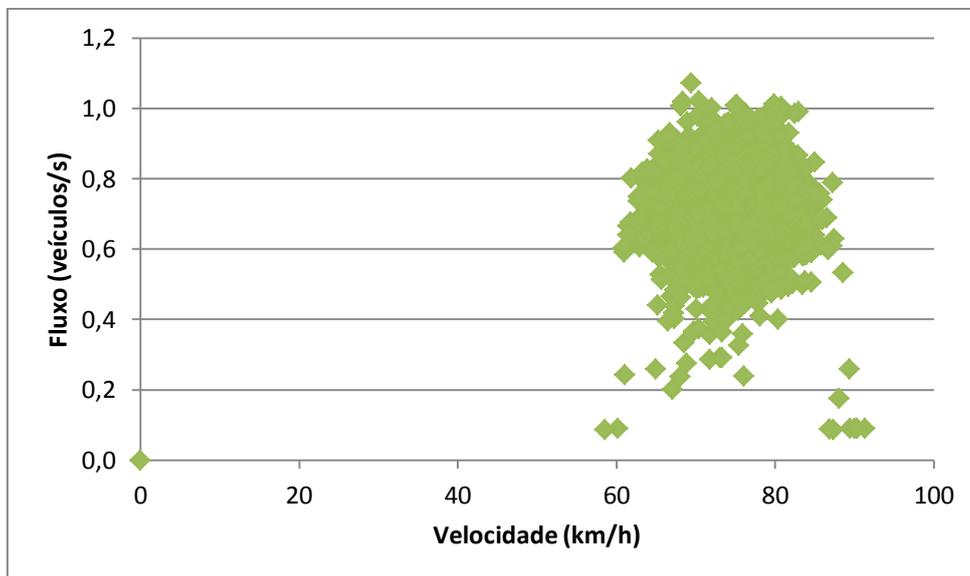


Figura 59 - Velocidade-Fluxo para MAS/STB com taxa de injeção de 25%.

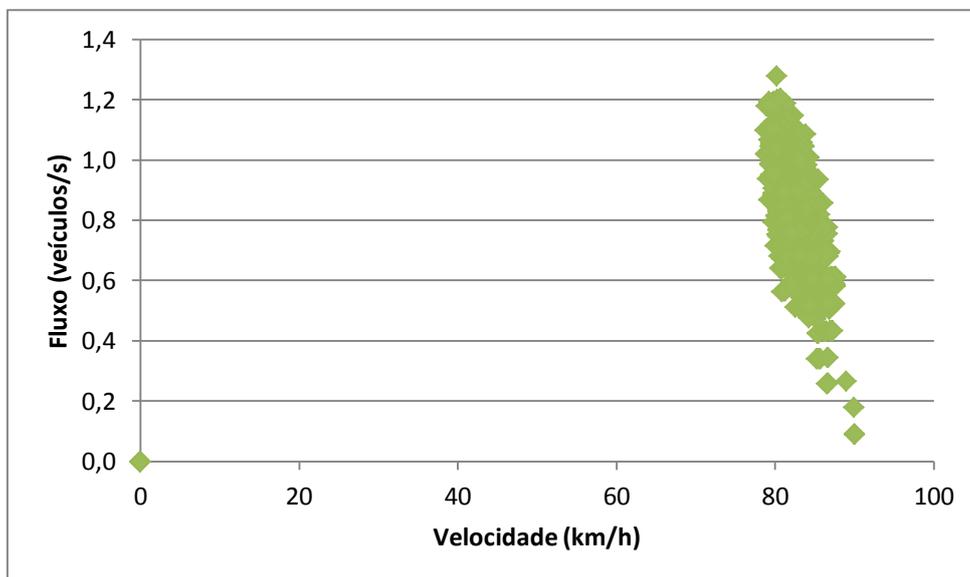


Figura 60 - Velocidade-Fluxo para IDM com taxa de injeção de 25%.

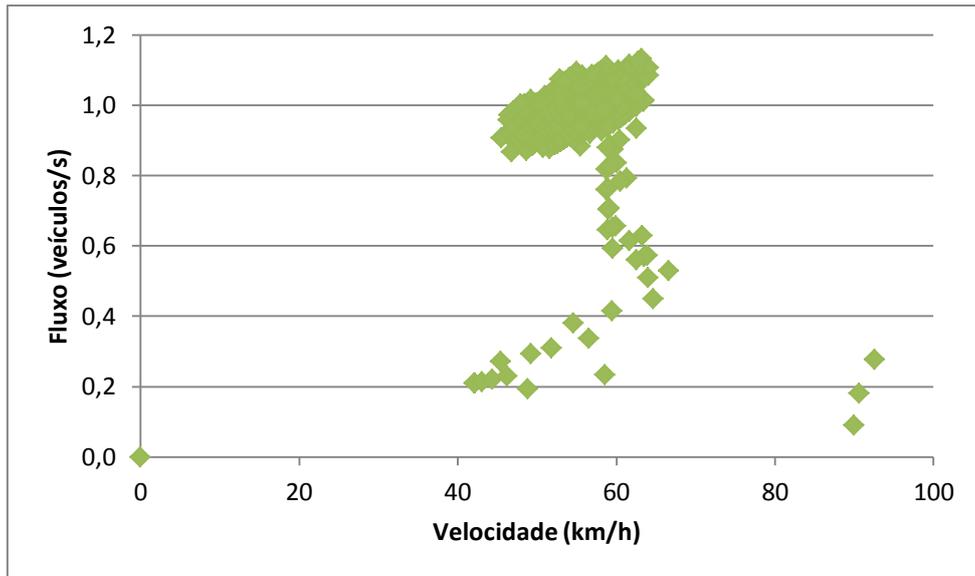


Figura 61 - Velocidade-Fluxo para MAS/STB com taxa de injeção de 100%.

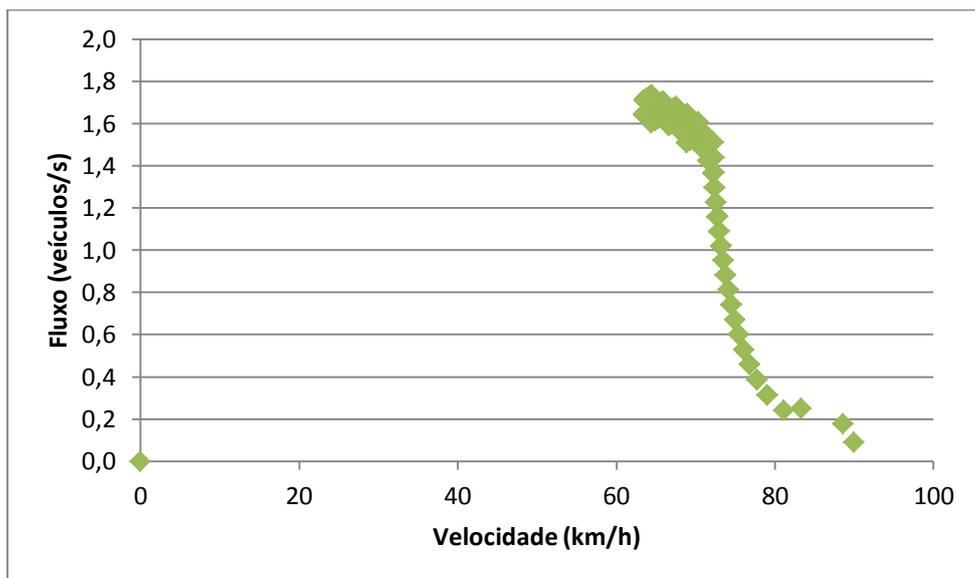


Figura 62 - Velocidade-Fluxo para IDM com taxa de injeção de 100%.

Os gráficos de Velocidade-Fluxo são utilizados para determinar a velocidade a que o fluxo ideal ocorre e são compostos por dois ramos. O primeiro ramo refere-se à livre circulação de veículos e ocorre quando a velocidade é elevada e o fluxo é baixo. Podemos observar este comportamento nas figuras Figura 60, Figura 61 e Figura 62. Quando o fluxo ideal é alcançado, emerge o segundo ramo, demonstrando o início do congestionamento através da diminuição da velocidade e do fluxo. Por exemplo, a Figura 62 podemos observar que o IDM parece atingir um estado ligeiro de início de congestão de tráfego a uma velocidade de 60 km/h.

5.3. Sinopse

Resumindo os resultados podemos afirmar que ambos os algoritmos vão ao encontro de valores que se podem considerar realistas e que, do ponto de vista qualitativo, são relatados na literatura. A elevada capacidade dinâmica adaptativa do IDM leva-o a produzir acções com alterações muito graduais. Isto produz uma uniformização de comportamentos no fluxo de tráfego, onde os comportamentos individuais não diferem significativamente, em média, do conjunto de veículos no fluxo de tráfego. Em contraste, o MAS/STB apresenta-se como um algoritmo vocacionado para SMA, baseado em regras, porventura mais realista. Podemos ainda avançar os seguintes resultados:

- A intensidade da aceleração a baixa velocidade do MAS/STB superou os valores do IDM. No entanto, a estabilidade do cálculo da aceleração do MAS/STB a velocidades elevadas parece estar longe de ser satisfatória. Se por um lado o IDM mantém um regime de velocidade quase totalmente constante no limite de velocidade, o comportamento do MAS/STB produz valores que aparentam demasiadas oscilações;
- Em termos de comportamentos de travagem o IDM mostrou um comportamento menos “agressivo” do que o MAS/STB. Se é verdade que uma travagem gradual face a um obstáculo distante pode ser descrita como o comportamento ideal de um condutor, o comportamento de travagem mais intenso face ao veículo da frente no MAS/STB parece ser mais próximo da realidade;
- O IDM não parece ter um comportamento realista em desaceleração quando circula acima da velocidade ideal;
- Em ambientes de grande densidade de tráfego o MAS/STB produziu comportamentos mais naturais do que o comportamento fluido do IDM. É possível, no entanto, que a parametrização do IDM com valores de desaceleração mais elevados possa reproduzir de forma mais satisfatória fenómenos como o trânsito lento;
- Em relação à análise macroscópica ainda que os resultados obtidos vão ao encontro dos esperados, as condições de injeção não nos permitiram analisar com rigor satisfatório os fenómenos aí observados. Com efeito, devido à sua dinâmica adaptativa, o processo de injeção é bastante mais fácil de realizar com o IDM e, no caso do STD/MAS, perturba os resultados obtidos.

CAPITULO 6

Conclusão

6.1. Conclusões gerais

A ideia por detrás deste trabalho sustentou-se em estudar um dos problemas do século XXI que prejudica a vida de milhões de pessoas diariamente – o tráfego rodoviário. Perspectivou-se desenvolver um módulo de um simulador capaz de reproduzir processos de *car-following* e que possa ser utilizado num ambiente de simulação presentemente a ser construído em Netlogo.

Através da revisão da literatura demonstrou-se a importância deste tipo de ambientes de simulação para um correcto entendimento dos sistemas sociais. Examinaram-se diversas abordagens e algoritmos de *car-following* com o propósito de encontrar os mais indicados para a os objectivos propostos.

Foi precioso o estudo da arquitectura do modelo *VeniceTable* e da extensão desenvolvida por Paulo Duarte, determinando as vantagens e limitações de modelos de simulação multiagente com ligação a sistema de informação geográfica. Através das informações recolhidas foi possível construir o modelo apresentado.

Com a conclusão deste trabalho, foi possível demonstrar a validade, os pontos fortes e fracos de dois algoritmos de *car-following* distintos na sua abordagem. Optou-se por comparar os resultados produzidos por um algoritmo recente baseado em estados cognitivos e regras (MAS/STB) com um algoritmo baseado na dinâmica entre veículos (IDM). Apesar de os resultados que atestam este trabalho terem sido feitos em condições óptimas, ignorando factores externos e comportamentos dos condutores, podemos afirmar que ambos os algoritmos estabelecem uma base considerável para futuros desenvolvimentos do presente ambiente de simulação.

6.2. Contribuições

Com a conclusão deste trabalho pensa-se ter contribuído para o estudo da viabilidade de implementação de dois modelos de *car-following* numa ferramenta de simulação social, bem como para o estudo e validação das suas principais características numa perspectiva comparativa. No alinhamento dos objectivos traçados para este trabalho, são as seguintes as contribuições:

- Estudo e consolidação do modelo de ligação SMA/SIG utilizando Netlogo, tendo-se estendido o ambiente de simulação já existente com diversas funcionalidades, inclusive a disponibilização de dois algoritmos de *car-following*;

- Extensão e contribuição para a validação do algoritmo MAS/STB através da simulação e apresentação de resultados desse algoritmo sob diversas configurações, e através de uma análise comparativa com os resultados obtidos nas mesmas condições por um algoritmo de referência;
- Especificação e implementação de um modelo de agente no ambiente de simulação capaz de incorporar e articular os dois algoritmos de *car-following* referidos;

6.3. Limitações e trabalho futuro

No âmbito deste trabalho, sugerem-se as seguintes indicações para trabalho futuro:

- O modelo apresentado exhibe apenas uma faixa de rodagem para os veículos circularem. Será necessário aumentar a capacidade de gestão do ambiente para várias faixas e introduzir algoritmos de mudança de faixa, obtendo comportamentos de fluxo mais complexos e realistas;
- A adição de factores rodoviários acrescentaria realismo à simulação. Semáforos, passadeiras, sinais de trânsito, regras de cruzamentos e rotundas são aspectos do dia-a-dia que influenciam o comportamento dos condutores. A incorporação no modelo de estilos de condução heterogéneos irá certamente originar resultados diferentes dos aqui apresentados;
- Do ponto de vista da eficiência computacional na implementação, a elaboração do percurso por parte do agente deverá ser melhorada. Neste modelo o veículo calcula um novo percurso a cada segundo a partir da sua posição actual. Por se tratar de um desperdício de recursos de processamento sugere-se a optimização através de duas ideias: o veículo define e constrói a totalidade do seu percurso quando é inserido no ambiente, ou cria-se a possibilidade do veículo guardar em memória o percurso até um *checkpoint*;
- Tendo em consideração as duas abordagens distintas dos dois algoritmos, e tendo em vista o desvanecimento dos consequentes efeitos distintos do processo de injeção na dinâmica dos dois algoritmos (nomeadamente, na análise macroscópica), estudar formas alternativas de desenho das experiências, a fim de obter resultados que sejam independentes do mecanismo de injeção;

Bibliografia

Al-Jameel, H.A. 2009. Examining and Improving the Limitations of the Gazis-Herman-Rothery Car-following Model. 2009.

Batista, Milan e Twrdy, Elen. 2010. Optimal velocity functions for car-following models. *Journal of Zhejiang University-SCIENCE A (Applied Physics & Engineering)*. 2010, Vol. 11 (7), p. 520–529. doi:10.1631/jzus.A0900370.

Bellinger, Gene. 2004. Modeling & Simulation. *Systems Thinking*. [Online] 2004. <http://www.systems-thinking.org/modsim/modsim.htm>.

Benjaafar, Saifallah, Dooley, Kevin e Setyawan, Wibowo. 1997. Cellular Automata for Traffic Flow Modeling. 1997.

Biham, Ofer, Middleton, A. Alan e Levine , Dov . 1992. Self-organization and a dynamical transition in traffic-flow models. *Physical Review A*. 1992, Vol. 46 (10).

Bossel, Harmut. 1994. *Modeling and Simulation: Volume 1*. s.l. : A.K. Peters, 1994.

Brackstone, Mark e McDonald, Mike. 1999. Car-following: a historical review. *Transportation Research*. 1999, Vol. F 2, p. 181-196.

Braubach, Lars, et al. 2005. Deployment of Distributed Multi-agent Systems. 2005.

Cassidy, Michael J. Traffic Flow and Capacity.

Castle, Christian J.E. e Crooks, Andrew T. 2006. Principles and Concepts of Agent-Based Modelling for Developing Geospatial Simulations. *UCL Working Papers Series*. 2006. http://gmu.academia.edu/AndrewCrooks/Papers/117082/Principles_and_Concepts_of_Agent-Based_Modelling_for_Developing_Geospatial_Simulations.

Champion, Alexis, et al. 1999. Traffic generation with the SCANeR II simulator: towards a multi-agent architecture. 1999.

Chowdhury, Debashish, Santen, Ludger e Schadschneider, Andreas. 2000. Statistical physics of vehicular traffic and some related systems. *Physics Reports*. 2000, Vol. 329, p. 199-329.

Dai, Ji Cai e Li, Xia. 2010. Multi-agent systems for simulating traffic behaviors. *Chinese Science Bulletin*. 2010, Vol. 55 n°3, p. 293-300.

Dresner, Kurt e Stone, Peter. 2004. Multiagent Traffic Management: A Reservation-Based Intersection Control Mechanism. *The Third International Joint Conference on Autonomous Agents and Multiagent Systems*. 2004.

D'Souza, Raissa M. The Bigham-Middleton-Levine traffic model. [Online] <http://mae.ucdavis.edu/dsouza/bml.html>.

Escola de Condução Invicta. 2011. O Condutor e o Veículo: Tempo de Reação. *Escola de Condução Invicta*. [Online] 2011. http://www.invicta.com.pt/codigo/tempo_reacao.asp.

Ferreira, Paulo A. F., et al. 2008. Extending Microscopic Traffic Modelling With the Concept of Situated Agents. *International Joint Conference on Agents and Multiagent Systems*. 2008.

Franklin, Stan e Graesser, Art. 1996. *Is it an Agent or just a Program?: A Taxonomy for Autonomous Agents*. 1996. http://www.agent.ai/doc/upload/200302/fran96_1.pdf.

Gilbert, Nigel. 2008. *Agent-Based Models*. 2008.

Gimblet, H.R. 2002. Integrating Geographic Information Systems and Agent-based Modeling Techniques. *Santa Fe Institute Studies in the Sciences of Complexity*. 2002.

Goldsmith, T. C. 1998. Automated Vehicle Guidance (AVCS) - The Real Automobile. [Online] 12 1998. <http://www.azinet.com/articles/real98.htm>.

Gonçalves, António S., Rodrigues, Armanda e Correia, Luís. 2004. Multi-agent simulation within geographic information systems. *Paper presented in the 5th Workshop on Agent-Based Simulation*. 2004.

Gonçalves, António S., Rodrigues, Armanda e Correia, Luís. 2004. Multi-Agent Simulation within Geographic Information Systems. *Proceedings of the 5th International Workshop on Agent-Based Simulation*. 2004, p. 107-112.

Hall, Fred L. 1999. Traffic Stream Characteristics. *Public Roads (US Dept of Transportation)*. 1999, Vol. 62 nº2.

Helbing, Dirk. 1995. Theoretical foundation of macroscopic traffic models. *Physics A: Statistical Mechanics and its Applications*. 1995, Vol. 219.

—. **2001.** Traffic and Related Self-Driven Many-Particle Systems. *Reviews of Modern Physics*. 2001, Vol. 73(4), p. 1067-1141.

Jiang, Bin e Gimblett, H. Randy. 2002. An Agent-Based Approach to Environmental and Urban Systems within Geographic Information Systems. *Integrating Geographic Information Systems and Agent-Based Modeling Techniques*. 2002.

- Karadimas, Nikolaos V., Rigopoulos, George e Bardis, Nikolaos. 2006.** Coupling Multiagent Simulation and GIS – an Application in Waste Management. *WSEAS Transactions on Systems*. 2006, Vol. 5, p. 2367-2371.
- Kesting, Arne, Treiber, Martin e Helbing, Dirk. 2008.** Agents for Traffic Simulation. *Multi-Agent Systems: Simulation and Applications*. 2008. arXiv:0805.0300v1 [physics.soc-ph].
- . **2010.** Enhanced Intelligent Driver Model to Assess the Impact of Driving Strategies on Traffic Capacity. *Philosophical Transactions of the Royal Society A*. 2010, Vol. 368, p. 4585-4605. arXiv:0912.3613v1 [physics.soc-ph].
- Khalesian, M. e Delavar, M. R. 2008.** A multi-agent based traffic network micro-simulation using spatio-temporal GIS. *Center of Excellence in Geomatics Eng. and Disaster Management*. 2008.
- Krug, J. e Spohn, H. 1988.** Universality classes for deterministic surface growth. *Physical Review A*. 1988, Vol. 38 (8).
- Kumar, Sujai e Mitra, Sugata . 2006.** Self-Organizing Traffic at a Malfunctioning Intersection. *Journal of Artificial Societies and Social Simulation*. 2006, Vol. 9 (4).
- Lansdowne, Andrew. 2006.** *Traffic Simulation using Agent-Based Modelling*. University of the West of England. 2006. Tese de Bacharelato.
- Li, Chaoyi, et al. 2008.** Evaluation and Improvement of Required Deceleration Algorithm in Frontal Collision Warning Systems. *Proceedings of the 11th International IEEE: Conference on Intelligent Transportation Systems*. 2008.
- Li, S., et al. 2010.** Modeling and verification of heavy-duty truck drivers' car-following characteristics. *International Journal of Automotive Technology*. 2010, Vol. 11 nº1.
- Luke, Sean , et al. 2005.** MASON: A Multi-Agent Simulation Environment. *Simulation: Transactions of the society for Modeling and Simulation International*. 2005, Vol. 82 (7).
- Marques, Paulo. 2011.** A evolução da Segurança Rodoviária. *AutoMotor*. [Online] 2 Agosto 2011.
<http://www.automotor.xl.pt/Not%C3%ADcias/DetailhedeNot%C3%ADcias/tabid/178/ItemId/9056/Default.aspx>.
- Mathew, Tom V. e Krishna Rao, K V. 2007.** Microscopic traffic flow. *Introduction to Transportation Engineering*. 2007, 34.
- Minar, Nelson, et al. 1996.** The Swarm Simulation System: A Toolkit for Building Multi-agent Simulations. 1996.

Nagel, Kai e Schreckenberg, Michael. 1992. A celular automaton model for freeway traffic. *J. Phys. I France*. 1992, Vol. 2 nº12, p. 2221-2229.

North, M.J., et al. 2007. A Declarative Model Assembly Infrastructure for Verification and Validation. 2007.

Paulo Duarte. Em curso. *Modelação de tráfego rodoviário em Netlogo com integração de Sistemas de Informação Geográfica*. Departamento de Ciências e Tecnologias da Informação, ISCTE-IUL. Em curso. Dissertação de Mestrado.

Pereira, José Luis Ferrás. 2011. *Multi-agent simulation of autonomous vehicles*. Faculdade de Engenharia da Universidade do Porto. 2011. Tese de Mestrado.

Rakha, Hesham, et al. 2001. Vehicle Dynamics Model for Predicting Maximum Truck Acceleration Levels. *Journal of Transportation Engineering*. 2001, Vol. 127 (5).

Rothery, Richard W. 1999. Car Following Models. *Public Roads (US Dept of Transportation)*. 1999, Vol. 62 nº2.

Sirito, Gabriele. 2006. *Bifurcation analysis of microscopic follow-the-leader traffic models*. Department Mathematik, Universität Hamburg. Hamburg: s.n., 2006. Dissertação de Doutoramento. Dissertação.

Snare, Matthew C. 2002. *Dynamics model for predicting maximum and typical acceleration rates of passenger vehicles*. Faculty of the Virginia Polytechnic Institute and State University. 2002. Tese de Mestrado.

Treiber, Martin, Hennecke, Ansgar e Helbing, Dirk. 2000. Congested Traffic States in Empirical Observations and Microscopic Simulations. *Physical Review E*. 2000, Vol. 62, p. 1805-1824. arXiv:cond-mat/0002177v2 [cond-mat.stat-mech].

Treiterer, J. 1975. *Investigation of Traffic Dynamics by Aerial Photogrammetry Techniques*. 1975. EES278.

Trigoso, José Manuel. 2011a. O ambiente rodoviário. *AutoMotor*. [Online] 3 Setembro 2011a. <http://www.automotor.xl.pt/Not%C3%ADcias/DetailhedeNot%C3%ADcias/tabid/178/ItemId/9336/Default.aspx>.

—. **2011b.** O utente é um dos factores chave na ocorrência de acidentes rodoviários. *AutoMotor*. [Online] 8 Outubro 2011b. <http://www.automotor.xl.pt/Not%C3%ADcias/DetailhedeNot%C3%ADcias/tabid/178/ItemId/9572/Default.aspx>.

van den Berg, M., et al. 2003. A macroscopic traffic flow model for integrated control of freeway and urban traffic networks. *Proceedings of the 42nd IEEE Conference on Decision and Control*. 2003.

Vasconcelos, Wamberto, et al. 2002. A Lifecycle for Models of Large Multi-agent Systems. 2002.

Wang, Ziyuan. 2007. *Decentralized Traffic Management Strategies for Sensor-Enabled Cars*. Department of Computer Science and Software Engineering, The University of Melbourne. 2007.

Weng, Yan-lin e Wu, Tie-jun. 2002. Car-following models of vehicular traffic. *Journal of Zhejiang University Science*. 2002, Vol. 3 n°4, p. 412-417.

Weyns, Danny, et al. 2005. Environments in multiagent systems. *The Knowledge Engineering Review*. 2005, Vol. 20 n°2, p. 127-141.
http://liawww.epfl.ch/~schumach/publications/environmentForMAS_ker_2005.pdf.

Wilensky, U. 2003. NetLogo Traffic Grid model.
<http://ccl.northwestern.edu/netlogo/models/TrafficGrid>. 2003.

—. 1999. NetLogo. <http://ccl.northwestern.edu/netlogo/>. 1999.

Zhao, X. e Gao, Z. 2005. A new car-following model: full velocity and acceleration difference model. *Eur. Phys. J. B*. 2005, Vol. 47, p. 145-150.

Anexos

Anexo A – Implementação em Netlogo do modelo de *car-following*

extensions [gis]

globals

[

;; global variables from model

REDE-VIARIA

INJECTION-LINK

INJECTING-CARS

;; global variables from car-following

MAX-SPEED ; speed limit

MIN-DIST ; minimum distance between cars

;; -----global variables for MAS/STB-----

LOW-ACCEL ; maximum acceleration in high speed

MAX-ACCEL ; maximum limit for acceleration

AV-SPEED ; average speed

;; -----global variables for IDM-----

DESIRED-SAFETY-TIME ; time-distance desired to following vehicle

ACCEL-EXP

A ; normal value for acceleration of the vehicle

B ; comfortable breaking value of the vehicle

;; -----PLOTS-----

COUNT-CARS-M

COUNT-CARS-10M

SUM-AVG-SPEED-10M

SPEED-UPPER-LIMIT

DENSITY-UPPER-LIMIT

FLOW-UPPER-LIMIT

HIGHER-AVG-SPEED

HIGHER-DENSITY

HIGHER-FLOW

AVG-SPEED

DENSITY

FLOW

VELOCITY

ACCELERATION

VELOCITY-2

ACCELERATION-2

DISTANCE-DRIVED

DISTANCE-DRIVED-2

DIST-AHEAD

```

DETECTED
TOTAL-COUNT-CARS
;; -----TESTS-----
TOTAL-LENGTH
SINGLE-INJ
TIME-WAITING
TIME-PASSING
TRAFFIC-LIGHT-PLACED?
CAR-SELECTED?
CAR-FOLLOWING?
]

breed [ nodes node ]
breed [ vehicles vehicle ]
breed [ injection-vehicles injection-vehicle ]

vehicles-own
[
  cur-link
  to-node
  planned-route ; calculated path
  speed ; instant speed of the vehicle
  accel ; instant acceleration of the vehicle
  car-ahead ; ID from the vehicle ahead
  dist-to-car-ahead ; distance to vehicle ahead in time [t]
  behav ; driver's current behavior
  ;; -----characteristics for MAS/STB-----
  safe-dist ; safety distance needed to vehicle ahead
  speed-t1 ; instant speed of the vehicle in time [t+1]
  accel-t1 ; instant acceleration of the vehicle in time [t+1]
  ;; -----characteristics for IDM-----
  ;desired-dynamic-distance
  ;; -----characteristics for TESTS-----
  dist-driven
  is-traffic-light?
]

injection-vehicles-own
[
  cur-link
  to-node
  planned-route
  speed
  accel
  car-ahead
  dist-to-car-ahead
  behav
  ;; -----characteristics for MAS/STB-----
  safe-dist
  speed-t1
  accel-t1
  ;; -----characteristics for IDM-----

```

```

;desired-dynamic-distance
;; -----characteristics for TESTS-----
dist-driven
is-traffic-light?
]

links-own [ oneway ]

..          *****
;;          *****
*****
STARTUP/SETUP/GO

to startup ;; called when first loaded
  read-gis-datasets
end

to setup
  print""
  show "Building Ambient Complete"

  clear-all
  read-gis-datasets

  ask patches [set pcolor white]
  setup-world-envelope
  setup-paths-graph

  set TOTAL-LENGTH calc-total-link-length([link-length] of links)

  setup-vehicles
  set SINGLE-INJ false
  set TIME-PASSING ifelse-value(simulation-mode = "FT-x.2" or simulation-mode = "FT-1.2")[ 0
] [ green-light-time ]
  set TIME-WAITING red-light-time
  set TRAFFIC-LIGHT-PLACED? false
  set CAR-SELECTED? nobody
  set CAR-FOLLOWING? nobody

  .*****
  ;
end

to go

  if(SINGLE-INJ = true and count vehicles = 0 and count injection-vehicles = 0)[ stop ] ;;stop
  condition for phase 1

  new-move-behavior
  inject-vehicles INJECTION-LINK

  do-plotting

```

```

;;traffic light conditions
ifelse(TRAFFIC-LIGHT-PLACED? = false)
[
  set TIME-WAITING red-light-time
  set TIME-PASSING TIME-PASSING - 1
]
[
  set TIME-PASSING green-light-time
  set TIME-WAITING TIME-WAITING - 1
]

tick
end

```

```

.. ***** GIS RELATED PROCEDURES
;
*****

```

to read-gis-datasets

```

let road-used 0
if(choose-map = "U-road")[ set road-used "redeviaria/U2/U2.shp" ]
if(choose-map = "L-road")[ set road-used "redeviaria/L/L.shp" ]
if(choose-map = "straight-road")[ set road-used "redeviaria/recta4/Recta4.shp" ]

set REDE-VIARIA gis:load-dataset road-used
end

```

to setup-world-envelope

```

let world (gis:envelope-of REDE-VIARIA) ;; [ minimum-x maximum-x minimum-y maximum-y ]
if zoom != 1 [
  let x0 (item 0 world + item 1 world) / 2      let y0 (item 2 world + item 3 world) / 2
  let W0 zoom * (item 0 world - item 1 world) / 2  let H0 zoom * (item 2 world - item 3 world) / 2
  set world (list (x0 - W0) (x0 + W0) (y0 - H0) (y0 + H0))
]
gis:set-world-envelope (world)
end

```

to setup-paths-graph

```

set-default-shape nodes "circle"
let polylines gis:feature-list-of REDE-VIARIA
foreach polylines [
  let polyline ?
  let vertex vertex-coordenates-of-polyline polyline node-precision
  if length vertex != 0 [
    (foreach butlast vertex butfirst vertex [ if ?1 != ?2 [ ;; skip nodes on top of each other due to
rounding
      let n1 new-node-at first ?1 last ?1
      let n2 new-node-at first ?2 last ?2
      ask n1 [
        create-link-with n2 [
          ;; assign here property-values to own-type link variables
          set oneway gis:property-value polyline "ONEWAY"

```

```

    if (end1 != n1) [ ;; if node already existed oneway may have to be corrected
      ifelse oneway = "FT"
        [ set oneway "TF" ]
        [ if oneway = "TF" [set oneway "FT"]]
    ]
  ]
]
])
]
]
ask nodes [hide-turtle]
end

```

```

to-report new-node-at [x y] ; returns a node at x,y creating one if there isn't one there.
  let n nodes with [xcor = x and ycor = y]
  ifelse any? n [set n one-of n] [create-nodes 1 [setxy x y set size 2 set n self hide-turtle] ]
  report n
end

```

```

to draw-world
  gis:set-drawing-color [255 0 0] gis:draw REDE-VIARIA 1
end

```

```

to-report vertex-coordenates-of-polyline [polyline decimalplaces] ;; returns vertex netlogo coords
of a polyline
  let vertex first (gis:vertex-lists-of polyline)
  set vertex map [gis:location-of ?] vertex
  set vertex remove [] vertex
  if ((length vertex) != 0) [
    set vertex map [map [precision ? decimalplaces] ? ] vertex
  ]
  report vertex
end

```

```

to-report meters-per-patch ;; maybe should be in gis: extension?
  let world gis:world-envelope ; [ minimum-x maximum-x minimum-y maximum-y ]
  let x-meters-per-patch (item 1 world - item 0 world) / (max-pxcor - min-pxcor)
  let y-meters-per-patch (item 3 world - item 2 world) / (max-pycor - min-pycor)
  report mean list x-meters-per-patch y-meters-per-patch
end

```

```

to-report calc-total-link-length [link-list]
  let total-road-length 0

  while[length link-list != 0]
  [
    set total-road-length total-road-length + item 0 link-list
    set link-list remove-item 0 link-list
  ]
  report total-road-length
end

```

```
.. ***** VEHICLES RELATED PROCEDURES
;;
*****
```

to setup-vehicles

```
set MAX-SPEED (max-speed-km/h / 3.6 / meters-per-patch)
set MIN-DIST (2 / meters-per-patch) + (4 / meters-per-patch) ;; 2 meters between vehicles + 4
meters because center mass
```

```
ifelse (car-following-model = "MAS/STB") ;;MAS/STB parameters
```

```
[
  set LOW-ACCEL (0.37 / meters-per-patch)
  set MAX-ACCEL (2.78 / meters-per-patch)
  set AV-SPEED (12.19 / meters-per-patch)
]
```

```
[ ;;IDM parameters
  set ACCEL-EXP 4
  set DESIRED-SAFETY-TIME 1.5
  set A 1 / meters-per-patch
  set B 2 / meters-per-patch
]
```

```
set-default-shape vehicles "boat"
set INJECTION-LINK one-of links
set INJECTING-CARS (list)
end
```

to inject-vehicles [!]

```
if(SINGLE-INJ = false)
[
  set SINGLE-INJ ifelse-value(simulation-mode = "FT-1.x" or simulation-mode = "FT-1.2")[ true
] [ false ]
```

```
if((simulation-mode = "FT-1.2" or simulation-mode = "FT-x.2") and TRAFFIC-LIGHT-
PLACED? = false and TIME-PASSING <= 0)
```

```
[
  insert-traffic-light
]
generate-inj-vehicles ;;generating vehicles
]
```

let inject true

```
while [inject = true and binomial-function > 0 and length INJECTING-CARS != 0] [
  let injection-boat first INJECTING-CARS
```

```
ask injection-boat
[
  nmb-information-gathering
  nmb-analysis-and-decision
```

```

    set inject initialize-inj-vehicles
  ]
]
end

```

to generate-inj-vehicles

```

create-injection-vehicles num-vehicles
[
  hide-turtle
  set size 4 / meters-per-patch
  set speed (starting-speed-km/h / 3.6) / meters-per-patch

  ;set-cur-boat-link l [end1] of l
  ;set-cur-boat-link (link 0 1) (node 0)

  let inj-link 0
  let inj-node 0

  if(choose-map = "straight-road")[ set inj-link (link 0 1) set inj-node [end1] of inj-link ]
  if(choose-map = "L-road")[ set inj-link (link 0 1) set inj-node [end1] of inj-link ]
  if(choose-map = "U-road")[ set inj-link (link 2 46) set inj-node [end2] of inj-link ]

  set-cur-boat-link inj-link inj-node

  set color yellow
  set is-traffic-light? false

  set INJECTING-CARS lput self INJECTING-CARS
]

end

```

to insert-traffic-light

```

create-vehicles 1
[
  hide-turtle

  set size 5 / meters-per-patch

  set shape "x"

  set speed 0

```

```
set color red

let inj-link 0
let inj-node 0

if(choose-map = "straight-road")[ set inj-link (link 0 1) set inj-node [end1] of inj-link ]
if(choose-map = "L-road")[ set inj-link (link 0 1) set inj-node [end1] of inj-link ]
if(choose-map = "U-road")[ set inj-link (link 2 46) set inj-node [end2] of inj-link ]

set-cur-boat-link inj-link inj-node

set planned-route calculate-path (0.85 * TOTAL-LENGTH)
forward-vehicle planned-route (0.85 * TOTAL-LENGTH)

set is-traffic-light? true
set TRAFFIC-LIGHT-PLACED? true

show-turtle

]
end

to-report initialize-inj-vehicles

ifelse(car-ahead = nobody)
[
;show(word""car-ahead" AHEAD - speed: "speed)
set breed vehicles

set INJECTING-CARS remove-item 0 INJECTING-CARS

show-turtle
```

```

set COUNT-CARS-M COUNT-CARS-M + 1
set COUNT-CARS-10M COUNT-CARS-10M + 1

nmb-taking-action
]
[
  let safe-inj-dist ifelse-value(car-following-model = "MAS/STB")[ 0.5 * calc-safe-dist ][ 0.5 *
calc-dynamic-dist ]

  ifelse(dist-to-car-ahead > safe-inj-dist)
  [
    set breed vehicles

    set speed ifelse-value(speed > [speed] of car-ahead)[ [speed] of car-ahead ][ speed ]

    ;show(word""car-ahead" AHEAD @ dist: "dist-to-car-ahead" with "([speed] of car-ahead)" //
safe-dist: "safe-inj-dist" & speed: "speed)

    set INJECTING-CARS remove-item 0 INJECTING-CARS

    show-turtle

    set COUNT-CARS-M COUNT-CARS-M + 1
    set COUNT-CARS-10M COUNT-CARS-10M + 1

    nmb-information-gathering
    nmb-analysis-and-decision
    nmb-taking-action
  ]
  [
    ;show(word"DO NOTHING because "car-ahead" @ "dist-to-car-ahead" & safe: "safe-inj-dist)
  ]
]

```

```
ifelse(simulation-mode = "FT-1.x" or simulation-mode = "FT-1.2")
[
  if(CAR-SELECTED? = nobody)[ set CAR-SELECTED? self ]
]
[
  if(CAR-SELECTED? = nobody and (remainder ticks 60) = 0)[ set CAR-SELECTED? self ]
]
set TOTAL-COUNT-CARS TOTAL-COUNT-CARS + 1
report false
end
```

to new-move-behavior

```
ask vehicles
[
  if (car-following-model = "MAS/STB" and is-traffic-light? = false)
  [
    nmb-information-gathering
  ]
]
```

```
ask vehicles
[
  if(is-traffic-light? = false)[ nmb-analysis-and-decision ]
]
```

```
ask vehicles
[
  if(is-traffic-light? = false)[ nmb-taking-action ]
]
```

```
]
```

```
ask vehicles
```

```
[
```

```
  if(TRAFFIC-LIGHT-PLACED? = true and is-traffic-light? = true and TIME-WAITING <= 0)
```

```
  [
```

```
    set TRAFFIC-LIGHT-PLACED? false
```

```
    die
```

```
  ]
```

```
]
```

```
end
```

```
to NMB-information-gathering
```

```
  set speed-t1 speed + accel
```

```
  if (speed-t1 < 0) [set speed-t1 0]
```

```
  set safe-dist calc-safe-dist
```

```
end
```

```
to NMB-analysis-and-decision
```

```
  ;let checking-distance (3 * (MAX-SPEED + 0.5 * accel) + size)
```

```
  let checking-distance (15 * (speed + size))
```

```
  set planned-route calculate-path (checking-distance)
```

```
  let ahead-data check-vehicle-ahead (checking-distance) planned-route
```

```
  set car-ahead item 2 ahead-data
```

```

ifelse(car-ahead != nobody)
[
  set dist-to-car-ahead item 1 ahead-data ; S[t]

  ;show(word"[is-traffic-light?] of car-ahead: "[is-traffic-light?] of car-ahead = true)" & dist-to-
  car-ahead < 2 * speed: "(dist-to-car-ahead < 2 * speed))

  ifelse ([is-traffic-light?] of car-ahead = true and dist-to-car-ahead < speed)
  [
    set dist-to-car-ahead checking-distance

    set car-ahead nobody

    set accel-t1 ifelse-value (car-following-model = "MAS/STB")[ STB-accel-free-road ][ IDM-
    free-road-term ]
  ]
  [
    set accel-t1 ifelse-value (car-following-model = "MAS/STB")[ STB-accel-car-ahead ][ IDM-
    interaction-term ]
  ]
  [
    set accel-t1 ifelse-value (car-following-model = "MAS/STB")[ STB-accel-free-road ][ IDM-free-
    road-term ]
  ]
end

to NMB-taking-action

ifelse (car-following-model = "MAS/STB")
[
  forward-vehicle planned-route (speed + (0.5 * accel))

  set dist-driven dist-driven + speed + (0.5 * accel)
]

```

```
set accel accel-t1
set speed speed-t1
]
[
set accel accel-t1

if(speed <= 0 and accel < 0)[ set accel 0]

forward-vehicle planned-route (speed + accel)
set dist-driven dist-driven + speed + accel

set speed speed + accel
if(speed < 0) [ set speed 0 ]
]
end

to-report calc-safe-dist
report max (list (speed-t1 * 3)(MAX-SPEED * 3))
end

to-report calc-dynamic-dist
let delta-speed (speed - [speed] of car-ahead)

report MIN-DIST + (speed * DESIRED-SAFETY-TIME) + ((speed * delta-speed) / (2 * sqrt (A *
B)))
end

to-report STB-accel-car-ahead

let front-speed ( [speed] of car-ahead + (0.5 * [accel] of car-ahead)) ;;Speed of vehicle ahead
```

```
let dist-to-car-ahead-t1 dist-to-car-ahead + front-speed - (speed + (0.5 * accel)) ;; S[t+1] -  
Distance to vehicle ahead at t+1
```

```
let delta-dist dist-to-car-ahead-t1 - dist-to-car-ahead ;;Delta Distance between vehices
```

```
if (dist-to-car-ahead-t1 <= MIN-DIST and delta-dist < 0) ;;Vehicle Immobilization
```

```
[
```

```
set color cyan
```

```
set behav "Stopping"
```

```
report (- 2 * speed-t1)
```

```
]
```

```
if (dist-to-car-ahead-t1 < safe-dist and delta-dist < 0) ;;Vehicle Breaking
```

```
[
```

```
set color red
```

```
set behav "Breaking"
```

```
report (0 - (speed-t1 ^ 2) / (2 * (dist-to-car-ahead-t1 - MIN-DIST)))
```

```
]
```

```
if (dist-to-car-ahead-t1 > MIN-DIST and dist-to-car-ahead <= safe-dist and delta-dist >= 0)  
;;Vehicle in Pursuit
```

```
[
```

```
set color blue
```

```
set behav "Pursuit"
```

```
report dist-to-car-ahead-t1 / (3 * MAX-SPEED)
```

```
]
```

```
if (dist-to-car-ahead-t1 > 3 * MAX-SPEED and speed-t1 < AV-SPEED and DELTA-DIST > 0)  
;;Vehicle in Quick Acceleration
```

```
[
    set color green
    set behav "Quick Acceleration"

    report MAX-ACCEL - (MAX-ACCEL - LOW-ACCEL) * speed-t1 / AV-SPEED
]

if (dist-to-car-ahead-t1 > 3 * MAX-SPEED and speed-t1 >= AV-SPEED and speed-t1 < MAX-
SPEED and delta-dist > 0) ;;Vehicle in Acceleration

[
    set color brown
    set behav "Low Acceleration"

    report LOW-ACCEL
]

if (dist-to-car-ahead-t1 > 3 * MAX-SPEED and speed-t1 > MAX-SPEED) ;;Vehicle Breaking for
respectin Speed Limit

[
    set color orange
    set behav "Speed Limit"

    report (0 - random-float 1) * meters-per-patch
]

;;Vehicle Default Behavior

set color grey
set behav "Default"

report 0
end
```

to-report STB-accel-free-road

if (speed-t1 > MAX-SPEED) ;;Vehicle Breaking for respectin Speed Limit

[

set color orange

set behav "Speed Limit in Free Road"

report (0 - random-float 1) / meters-per-patch

]

if (speed-t1 > AV-SPEED) ;;Vehicle with Low Acceleration

[

set color brown

set behav "Low Accel in Free Road"

report LOW-ACCEL

]

if (speed-t1 < AV-SPEED) ;;Vehicle with High Acceleration

[

set color green

set behav "Quick Accel in Free Road"

report MAX-ACCEL - (MAX-ACCEL - LOW-ACCEL) * speed-t1 / AV-SPEED

]

if (speed-t1 = AV-SPEED) ;;Vehicle with Default behavior

[

set color grey

set behav "Default in Free Road"

```
    report 0
  ]
end

to-report IDM-interaction-term

  let desired-dynamic-distance calc-dynamic-dist

  let at1 a * (1 - (speed / MAX-SPEED) ^ ACCEL-EXP - (desired-dynamic-distance / dist-to-car-ahead) ^ 2)

  if (at1 < 0) [set color red set behav "Interaction (Negative)"]
  if (at1 > 0) [set color green set behav "Interaction (Positive)"]
  if (at1 = 0) [set color grey set behav "Interaction (Null)"]

  report at1
end

to-report IDM-free-road-term

  set color blue

  set behav "Free Road"

  let at1 a * (1 - (speed / MAX-SPEED) ^ ACCEL-EXP)

  report at1
end

to-report calculate-path [dist]

  set dist dist - distance to-node

  let l cur-link

  let t-n to-node
```

```

let path (list l)
while [dist >= 0] [
  let next-pair next-pair-link-node l t-n
  set l item 0 next-pair
  if l = nobody
    [report lput nobody path] ;; nobody is the link where the boat dies
  set t-n item 1 next-pair
  set path lput l path
  set dist dist - [link-length] of l
]
report path
end

to-report next-pair-link-node [l n]
  let nextlinks ([my-links] of n) with [self != l and (oneway = "S" or (oneway = "FT" and end1 = n)
or (oneway = "TF" and end2 = n))]
  ifelse count nextlinks = 0 [
    report (list nobody nobody) ;; boat dies here (if boats should not die at the end of a link
    quotate this)
    ;;ifelse n = [end1] of l
    ;;[let new-n [end2] of l report (list l new-n)]
    ;;[let new-n [end1] of l report (list l new-n)]
  ][
    let new-l one-of nextlinks
    ifelse n = [end1] of new-l
      [let new-n [end2] of new-l report (list new-l new-n)]
      [let new-n [end1] of new-l report (list new-l new-n)]
  ]
end

to forward-vehicle [path dist]
  if dist >= 0 [ ;; distance may be zero if boat is changing link

```

```

ifelse dist < distance to-node [
  forward dist
]
let d distance to-node
forward d
set path remove-item 0 path
set-cur-boat-link first path to-node
forward-vehicle path dist - d
]
]
end

to set-cur-boat-link [l n] ;; boat proc
if l = nobody [die] ;; setting cur-link to nobody causes boat to die
set cur-link l
move-to n
ifelse n = [end1] of l [set to-node [end2] of l] [set to-node [end1] of l]
face to-node
end

;; check boats ahead on current and subsequent links
;; reports [possible-path / possible-dist / boat-ahead or nobody if there is none]
to-report check-vehicle-ahead [dist path]
let dist-to-node distance to-node
;; check on current link
let d ifelse-value (dist > dist-to-node) [dist-to-node] [dist]
let boats-ahead vehicles with [to-node = [to-node] of myself and cur-link = [cur-link] of myself]
set boats-ahead boats-ahead with [distance to-node < dist-to-node and dist-to-node - d <=
distance to-node and self != myself]
let boat-ahead min-one-of boats-ahead [distance myself]
if boat-ahead != nobody

```

```

[report (list (list cur-link) (distance boat-ahead) boat-ahead)]

;; a single link to process?
ifelse (dist < dist-to-node)
  [report (list (list cur-link) dist nobody)]

;; otherwise check boats ahead on subsequent links

[report check-vehicle-ahead2 (dist - dist-to-node) dist-to-node (remove-item 0 path) (other-
end2 item 1 path to-node) (list cur-link)]

;; arguments = remaining distance, processed distance, path to process, to-node of next link,
processed path

end

;; checks boats ahead on subsequent links

;; parameters: remaining distance, processed distance, path to process, to-node of next link,
processed path

;; reports [possible-path / possible-dist / boat-ahead or nobody if there is none]
to-report check-vehicle-ahead2 [dist processed-dist path t-n processed-path]
  let l first path ;; current link

  if l = nobody
    [report (list path processed-dist nobody)] ;; boat will die here, no boats ahead beyond this
point

  let l-link-length [link-length] of l

  ifelse dist < l-link-length [ ;; then this is the last link to process

    let boats-ahead vehicles with [to-node = t-n and cur-link = l and distance t-n >= l-link-length -
dist]

    let boat-ahead max-one-of boats-ahead [distance t-n]

    ifelse boat-ahead = nobody [

      ;; add current link to processed-path, there is no boat ahead

      report (list (lput l processed-path) (processed-dist + dist) nobody)

```

```

]]
;; there is boat ahead

  report (list (lput l processed-path) (processed-dist + l-link-length - [distance t-n] of boat-
ahead) boat-ahead)

]

]] ;; then there may be other links to process

let boats-ahead vehicles with [(to-node = t-n) and (cur-link = l)]

let boat-ahead max-one-of boats-ahead [distance t-n]

ifelse boat-ahead = nobody [

  report check-vehicle-ahead2 (dist - l-link-length) (processed-dist + l-link-length) (remove-
item 0 path) (other-end2 item 1 path t-n) (lput l processed-path)

]]

report (list (lput l processed-path) (processed-dist + l-link-length - [distance t-n] of boat-
ahead) boat-ahead)

]

]

end

```

to toggle-vehicles

```

ifelse mean [size] of vehicles = 4

[ ask vehicles [set size size / meters-per-patch set speed speed / meters-per-patch ] ]

[ ask vehicles [set size 4 set speed speed * meters-per-patch ] ]

end

```

```

;; ***** LINK MANAGEMENT ADD-ONS *****
;; *****

```

```

to-report other-end2[l n]

if l = nobody [report nobody]

ifelse n = [end1] of l [report [end2] of l] [report [end1] of l]

end

```

```
.. *****
;;
***** ONE-LINERS
```

to clear-all-but-globals reset-ticks ct cp cd clear-links clear-all-plots clear-output end

to-report mid-nodes report nodes with [count link-neighbors = 2] end

to-report end-nodes report nodes with [count link-neighbors = 1] end

to-report hub-nodes report nodes with [count link-neighbors > 2] end

```
.. ***** PLOTTERS RELATED
;;
*****
```

;; PROCEDIMENTO REFERENTE AOS GRÁFICOS

to do-plotting

if(count vehicles > 0)

[

;;SPEED-Components

set-current-plot "Vehicle Speed (km/h)"

let vt ((sum [speed] of vehicles) / count vehicles)

set SPEED-UPPER-LIMIT ifelse-value(SPEED-UPPER-LIMIT > max [speed] of vehicles)[
SPEED-UPPER-LIMIT][max [speed] of vehicles]

set SUM-AVG-SPEED-10M SUM-AVG-SPEED-10M + vt

set-plot-y-range 0 (ceiling (0.1 + SPEED-UPPER-LIMIT * meters-per-patch * 3.6))

set-plot-x-range (ifelse-value (ticks < 60)[0][ticks - 60]) (ifelse-value (ticks < 60)[60][ticks
)

set-current-plot-pen "avg-speed"

```

plot vt * meters-per-patch * 3.6

set-current-plot-pen "max-speed"
plot (max [speed] of vehicles) * meters-per-patch * 3.6

set-current-plot-pen "min-speed"
plot (min [speed] of vehicles) * meters-per-patch * 3.6

;;DENSITY-Components (vehicles/km)
set-current-plot "Traffic Density (vehicles/km)"

if (remainder ticks 60 = 0 and ticks >= 60)
[
  let dt (COUNT-CARS-M * 4) / (TOTAL-LENGTH * meters-per-patch)

  set DENSITY-UPPER-LIMIT ifelse-value(DENSITY-UPPER-LIMIT > dt)[ DENSITY-UPPER-
LIMIT ][ dt ]

  set-plot-y-range 0 ceiling DENSITY-UPPER-LIMIT

  set-plot-x-range (ifelse-value (ticks < 600)[ 0 ][ ticks - 600 ]) (ceiling (ifelse-value (ticks <
600)[ 600 ][ ticks ]))

  set-current-plot-pen "show-dens"

  plot dt

  set COUNT-CARS-M 0
]

;;FLUX-Components (vehicles/hour)
set-current-plot "Traffic Flow (vehicles/hour)"

```

```

if (remainder ticks 600 = 0 and ticks >= 600)
[
    let qt ((COUNT-CARS-10M * 4 ) / (TOTAL-LENGTH * meters-per-patch)) * (SUM-AVG-
SPEED-10M / 600)

    set FLOW-UPPER-LIMIT ifelse-value(FLOW-UPPER-LIMIT > qt)[ FLOW-UPPER-LIMIT ][ qt
]

    set-plot-y-range 0 ceiling FLOW-UPPER-LIMIT

    set-plot-x-range (ifelse-value (ticks < 3600)[ 0 ] [ ticks - 3600]) (ceiling (ifelse-value (ticks <
3600)[ 3600 ] [ ticks ]))

    set-current-plot-pen "count-flow"

    plot qt

    set COUNT-CARS-10M 0

    set SUM-AVG-SPEED-10M 0

]

```

```

..*****
''
*****

```

:: FUNDAMENTAL DIAGRAM OF TRAFFIC FLOW

```

    set HIGHER-AVG-SPEED ifelse-value (HIGHER-AVG-SPEED > vt)[ HIGHER-AVG-SPEED ][
vt ]

    let dt (count vehicles) / (TOTAL-LENGTH * meters-per-patch)

    set HIGHER-DENSITY ifelse-value (HIGHER-DENSITY > dt)[ HIGHER-DENSITY ][ dt ]

    let qt (vt * meters-per-patch * 3.6) * dt

    set HIGHER-FLOW ifelse-value (HIGHER-FLOW > qt)[ HIGHER-FLOW ][ qt ]

    set-current-plot "Velocity-Density"

    set-plot-y-range 0 ceiling (0.1 + HIGHER-AVG-SPEED * meters-per-patch * 3.6)

    set-plot-x-range 0 ceiling HIGHER-DENSITY

```

```
set-current-plot-pen "vt-dt"
```

```
plotxy dt (vt * meters-per-patch * 3.6)
```

```
set-current-plot "Flow-Density"
```

```
set-plot-y-range 0 ceiling (0.1 + HIGHER-FLOW)
```

```
set-plot-x-range 0 ceiling HIGHER-DENSITY
```

```
set-current-plot-pen "qt-dt"
```

```
plotxy dt qt
```

```
set-current-plot "Velocity-Flow"
```

```
set-plot-y-range 0 ceiling (0.1 + HIGHER-AVG-SPEED * meters-per-patch * 3.6)
```

```
set-plot-x-range 0 ceiling (0.1 + HIGHER-FLOW)
```

```
set-current-plot-pen "vt-qt"
```

```
plotxy qt (vt * meters-per-patch * 3.6)
```

```
set AVG-SPEED (vt * meters-per-patch * 3.6)
```

```
set DENSITY dt
```

```
set FLOW qt
```

```
:: SELECTED VEHICLES
```

```
if(CAR-SELECTED? != nobody)
```

```
[
```

```
  ask CAR-SELECTED?
```

```
  [
```

```
    set VELOCITY (speed * meters-per-patch * 3.6) ;;km/h
```

```
    set ACCELERATION accel * meters-per-patch ;;m/s2
```

```
    set DISTANCE-DRIVED (dist-driven * meters-per-patch) ;; m
```

```
    set DETECTED car-ahead
```

```
  ]
```

```
ask vehicles
```

```
[
```

```

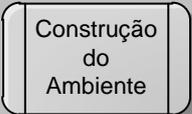
if car-ahead = CAR-SELECTED?
[
  set CAR-FOLLOWING? self
]
]
]
if(CAR-FOLLOWING? != nobody)
[
  ask CAR-FOLLOWING?
  [
    set VELOCITY-2 (speed * meters-per-patch * 3.6) ;;km/h
    set ACCELERATION-2 accel * meters-per-patch ;;m/s2
    set DISTANCE-DRIVED-2 (dist-driven * meters-per-patch) ;; m
    set DIST-AHEAD dist-to-car-ahead
  ]
]
]
end

.....
;;
*****

to-report binomial-function
  let n 1
  let p p-injection / 100
  let r sum n-values n [ifelse-value (p > random-float 1) [1] [0]]
  ;show(word"BIN("r")")
  report r
end

```

Anexo B – Notação utilizada para os fluxogramas

Exemplo	Descrição	
	Rectângulo com descrição a negrito	Esta forma representa o início do processo descrito
	Rectângulo com descrição	Esta forma representa um processo principal dentro de uma cadeia de processos
	Rectângulo com duas barras verticais nas laterais	Esta forma representa um conjunto de subprocessos na cadeia de sistema
	Losango	Esta forma representa a verificação de uma determinada condição descrita