

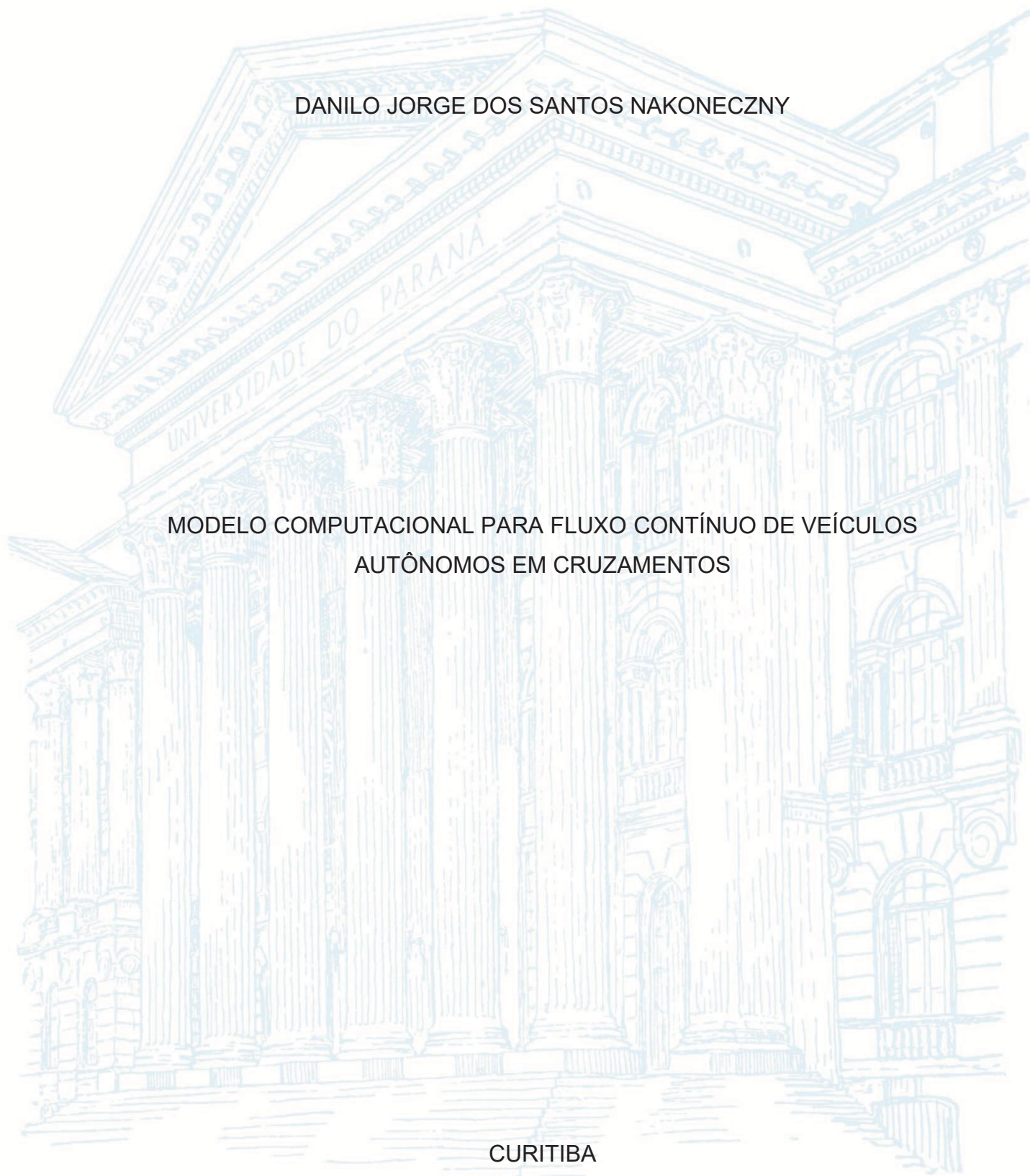
UNIVERSIDADE FEDERAL DO PARANÁ

DANILO JORGE DOS SANTOS NAKONECZNY

MODELO COMPUTACIONAL PARA FLUXO CONTÍNUO DE VEÍCULOS
AUTÔNOMOS EM CRUZAMENTOS

CURITIBA

2023



DANILO JORGE DOS SANTOS NAKONECZNY

MODELO COMPUTACIONAL PARA FLUXO CONTÍNUO DE VEÍCULOS
AUTÔNOMOS EM CRUZAMENTOS

Tese apresentada ao curso de Pós-Graduação em Métodos Numéricos em Engenharia, Área de Concentração em Programação Matemática, Setor de Ciências Exatas e Tecnologia, Universidade Federal do Paraná, como requisito parcial à obtenção do título de Doutor em Métodos Numéricos em Engenharia.

Orientador: Prof. Dr. Eloy Kaviski

CURITIBA

2023

DADOS INTERNACIONAIS DE CATALOGAÇÃO NA PUBLICAÇÃO (CIP)
UNIVERSIDADE FEDERAL DO PARANÁ
SISTEMA DE BIBLIOTECAS – BIBLIOTECA DE CIÊNCIA E TECNOLOGIA

Nakoneczny, Danilo Jorge dos Santos

Modelo computacional para fluxo contínuo de veículos autônomos em cruzamentos / Danilo Jorge dos Santos Nakoneczny. – Curitiba, 2023.

1 recurso on-line : PDF.

Tese (Doutorado) - Universidade Federal do Paraná, Setor de Ciências Exatas, Programa de Pós-Graduação em Métodos Numéricos em Engenharia.

Orientador: Eloy Kaviski

1. Veículos autônomos. 2. Rodovias - Interferências e interseções. 3. Trânsito - Controle eletrônico. 4. Modelo Computacional. I. Universidade Federal do Paraná. II. Programa de Pós-Graduação em Métodos Numéricos em Engenharia. III. Kaviski, Eloy. IV. Título.

Bibliotecário: Elias Barbosa da Silva CRB-9/1894



TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação MÉTODOS NUMÉRICOS EM ENGENHARIA da Universidade Federal do Paraná foram convocados para realizar a arguição da tese de Doutorado de **DANILO JORGE DOS SANTOS NAKONECZNY** intitulada: **MODELO COMPUTACIONAL PARA FLUXO CONTÍNUO DE VEÍCULOS AUTÔNOMOS EM CRUZAMENTOS**, sob orientação do Prof. Dr. ELOY KAVISKI, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de doutor está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 16 de Agosto de 2023.

Assinatura Eletrônica

21/08/2023 11:00:05.0

ELOY KAVISKI

Presidente da Banca Examinadora

Assinatura Eletrônica

21/08/2023 12:44:30.0

JULIO CÉSAR NIEVOLA

Avaliador Externo (PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ)

Assinatura Eletrônica

26/08/2023 10:16:47.0

ANSELMO CHAVES NETO

Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Assinatura Eletrônica

18/08/2023 08:44:00.0

MARCELO RODRIGUES BESSA

Avaliador Externo (UNIVERSIDADE FEDERAL DO PARANÁ)

AGRADECIMENTOS

Agradeço a Deus, por ter me fortalecido e dado saúde e paciência para que eu pudesse realizar este trabalho.

À minha esposa Sahra, pela compreensão da minha ausência, por me motivar e ajudar a prosseguir em meus sonhos.

Aos meus pais Joacir e Rosangela, pelo cuidado, carinho e investimento que realizaram em minha vida, e o apoio as minhas decisões.

Ao Eloy Kaviski, meu Orientador, por acreditar, confiar e me ajudar em muitos encontros que tivemos para obtenção deste trabalho, mesmo que por muitas vezes precisasse ser de forma remota.

À Liliana M. Gramani, minha professora, por sempre ter se dispor em ajudar com sugestões e conselhos no início deste trabalho e por seus incentivos.

À CAPES, por ter concedido a bolsa de estudo de maneira que fosse possível me dedicar com melhor qualidade de tempo aos estudos.

Aos meus amigos e colegas de pós-graduação.

Agradeço também a todos a minha volta que de alguma maneira contribuíram para a concretização deste estudo.

*"O conselho da sabedoria é: procure obter sabedoria;
use tudo que você possui para adquirir entendimento."*

Provérbios 4:7

RESUMO

Nesta tese apresenta-se o desenvolvimento de um modelo computacional não linear, capaz de realizar o gerenciamento da travessia de veículos autônomos nos cruzamentos viários, de maneira que estes possam fluir continuamente pelas estradas, sem que haja a necessidade desses veículos pararem nas faixas de retenção nas interseções viárias. E baseando-se neste modelo, foi implementado um algoritmo para simular o fluxo de tais veículos em um cruzamento com vias de 3 faixas e sentido único. Também foi desenvolvido um método específico para simular o comportamento dos veículos autônomos durante a travessia, considerando veículos de comprimentos variados. E para resolver o modelo foram construídos 2 métodos que apresentam soluções em tempo real de execução da simulação. Buscando avaliar o desempenho dos métodos foram realizadas 690 mil simulações ao todo, onde a distância de atuação do controlador de interseção variou em até 24 distâncias distintas, sendo 10 mil simulações para cada distância considerada e para 3 demandas diferentes de veículos das vias. Dessa forma, notou-se uma relação na eficiência dos métodos com as distâncias de atuação e que a média das velocidades dos veículos ao passarem pela interseção fica próxima à velocidade dos mesmos no início da simulação.

Palavras-chave: Gerenciamento de Veículos Autônomos; Controlador de Interseção viária; Controle Inteligente de Tráfego; Simulação de Veículos Autônomos; Modelo Computacional.

ABSTRACT

This thesis presents the development of a nonlinear computational model capable of managing the crossing of autonomous vehicles at road intersections, allowing them to flow continuously on the roads without the need to stop at the holding areas in the intersections. Based on this model, an algorithm was implemented to simulate the flow of such vehicles at a three-lane, one-way intersection. A specific method was also developed to simulate the behavior of autonomous vehicles during the crossing, considering vehicles of varying lengths. And to solve the model, 2 methods were built those present real-time solutions for running the simulation. In order to evaluate the performance of these methods, a total of 690 thousand simulations were conducted. Where the actuation distance of the intersection controller varied in up to 24 different distances, with 10 thousand simulations performed for each distance, and considering three different vehicle demand scenarios. As a result, a relationship was observed between the efficiency of the methods and the operating distances, with the average vehicle speeds as they pass through the intersection remaining close to their speeds at the beginning of the simulation.

Keywords: Autonomous Vehicle Management; Road Intersection Controller; Intelligent Traffic Control; Autonomous Vehicle Simulation; Computational Model.

LISTA DE FIGURAS

FIGURA 1 – DISTÂNCIA ENTRE VEÍCULOS CONSECUTIVOS.....	27
FIGURA 2 – CRUZAMENTO DE DUAS VIAS COM 3 FAIXAS	31
FIGURA 3 – ILUSTRAÇÃO DA FUNÇÃO OBJETIVO.....	34
FIGURA 4 – FLUXOGRAMA DA SIMULAÇÃO DE TRÁFEGO CONTÍNUO	39
FIGURA 5 – FLUXOGRAMA DO MÉTODO 1	41
FIGURA 6 – MÉTODO 2.....	43
FIGURA 7 – HISTOGRAMAS DO NÚMERO DE COLISÕES SEM A INTERFERÊNCIA DO CONTROLADOR PARA UMA DEMANDA DE 20 VEÍCULOS.....	46
FIGURA 8 – HISTOGRAMAS DO NÚMERO DE COLISÕES SEM A INTERFERÊNCIA DO CONTROLADOR PARA UMA DEMANDA DE 30 VEÍCULOS.....	46
FIGURA 9 – HISTOGRAMAS DO NÚMERO DE COLISÕES SEM A INTERFERÊNCIA DO CONTROLADOR PARA UMA DEMANDA DE 40 VEÍCULOS.....	47
FIGURA 10 – GRÁFICO DO NÚMERO DE COLISÕES COM A ATUAÇÃO DO CONTROLADOR PARA 20 VEÍCULOS	50
FIGURA 11 – GRÁFICO DO NÚMERO DE COLISÕES COM A ATUAÇÃO DO CONTROLADOR PARA 30 VEÍCULOS	53
FIGURA 12 – GRÁFICO DO NÚMERO DE COLISÕES COM A ATUAÇÃO DO CONTROLADOR PARA 40 VEÍCULOS	56
FIGURA 13 – GRÁFICO DO NÚMERO DE COLISÕES COM A ATUAÇÃO DO CONTROLADOR PARA AS DEMANDAS DE 20, 30 E 40 VEÍCULOS	61
FIGURA 14 – HISTOGRAMAS DAS MÉDIAS DAS VELOCIDADES PARA D = 100, 300 E 500 M.....	90
FIGURA 15 – HISTOGRAMAS DAS MÉDIAS DAS VELOCIDADES PARA D = 700, 900 E 1100 M.....	91
FIGURA 16 – HISTOGRAMAS DAS MÉDIAS DAS VELOCIDADES PARA D = 1300, 1500 E 1700 M.....	92
FIGURA 17 – HISTOGRAMAS DAS MÉDIAS DAS VELOCIDADES PARA D = 1900, 2100 E 2300 M.....	93

LISTA DE TABELAS

TABELA 1 – VALORES MÍNIMOS, MÁXIMOS, AMPLITUDE, MÉDIA E DESVIO PADRÃO DA MÉDIA DAS VELOCIDADES PARA CADA $D = \{100, 300, 500, 700, \dots, 2100, 2300\}$ COM 20 VEÍCULOS.....	48
TABELA 2 – NÚMERO DE COLISÕES COM E SEM O USO DO CONTROLADOR E NÚMERO DE SIMULAÇÕES COM COLISÕES PARA CADA DISTÂNCIA D COM 20 VEÍCULOS	49
TABELA 3 – VALORES MÍNIMOS, MÁXIMOS, AMPLITUDE, MÉDIA E DESVIO PADRÃO DA MÉDIA DAS VELOCIDADES PARA CADA $D = \{100, 300, 500, 700, \dots, 3900, 4100\}$ COM 30 VEÍCULOS.....	51
TABELA 4 – NÚMERO DE COLISÕES COM E SEM O USO DO CONTROLADOR E NÚMERO DE SIMULAÇÕES COM COLISÕES PARA CADA DISTÂNCIA D COM 30 VEÍCULOS	52
TABELA 5 – VALORES MÍNIMOS, MÁXIMOS, AMPLITUDE, MÉDIA E DESVIO PADRÃO DA MÉDIA DAS VELOCIDADES PARA CADA $D = \{100, 300, 500, 700, \dots, 4500, 4700\}$ COM 40 VEÍCULOS.....	54
TABELA 6 – NÚMERO DE COLISÕES COM E SEM O USO DO CONTROLADOR E NÚMERO DE SIMULAÇÕES COM COLISÕES PARA CADA DISTÂNCIA D COM 40 VEÍCULOS	55
TABELA 7 – VALORES MÍNIMOS, MÁXIMOS, AMPLITUDE, MÉDIA E DESVIO PADRÃO DA MÉDIA DAS VELOCIDADES PARA CADA $D = \{100, 200, 300\}$ COM 20 VEÍCULOS.....	57
TABELA 8 – NÚMERO DE COLISÕES COM E SEM O USO DO CONTROLADOR E NÚMERO DE SIMULAÇÕES COM COLISÕES PARA CADA DISTÂNCIA D COM 20 VEÍCULOS USANDO O MÉTODO 2.....	58
TABELA 9 – VALORES MÍNIMOS, MÁXIMOS, AMPLITUDE, MÉDIA E DESVIO PADRÃO DA MÉDIA DAS VELOCIDADES PARA CADA $D = \{100, 200, 300, 400\}$ COM 30 VEÍCULOS.....	58
TABELA 10 – NÚMERO DE COLISÕES COM E SEM O USO DO CONTROLADOR E NÚMERO DE SIMULAÇÕES COM COLISÕES PARA CADA DISTÂNCIA D COM 30 VEÍCULOS USANDO O MÉTODO 2	59

TABELA 11 – VALORES MÍNIMOS, MÁXIMOS, AMPLITUDE, MÉDIA E DESVIO PADRÃO DA MÉDIA DAS VELOCIDADES PARA CADA $D = \{100,$ $200, 300, 400, 500\}$ COM 40 VEÍCULOS	59
TABELA 12 – NÚMERO DE COLISÕES COM E SEM O USO DO CONTROLADOR E NÚMERO DE SIMULAÇÕES COM COLISÕES PARA CADA DISTÂNCIA D COM 40 VEÍCULOS USANDO O MÉTODO 2	60

SUMÁRIO

1	INTRODUÇÃO	12
1.1	HIPÓTESE	13
1.2	OBJETIVOS	13
1.3	JUSTIFICATIVAS	14
1.4	ORGANIZAÇÃO DO DOCUMENTO.....	14
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	REVISÃO BIBLIOGRÁFICA	16
2.2	COMENTÁRIOS.....	25
2.3	DEFINIÇÕES.....	26
3	MODELO E MÉTODO	27
3.1	CÁLCULO DA DISTÂNCIA ENTRE VEÍCULOS CONSECUTIVOS.....	27
3.2	CÁLCULO DA DISTÂNCIA MÍNIMA ENTRE VEÍCULOS CONSECUTIVOS ..	29
3.3	ANÁLISE DE COLISÃO NO CRUZAMENTO	30
3.4	MODELO COMPUTACIONAL	33
3.5	FORMULAÇÃO DO ALGORITMO.....	36
3.5.1	Algoritmo de simulação	37
3.5.2	Método 1 para tomada de decisão.....	40
3.5.3	Método 2 para tomada de decisão.....	42
4	RESULTADOS	45
4.1	RESULTADOS COM O MÉTODO 1	48
4.2	RESULTADOS COM O MÉTODO 2	57
5	CONCLUSÃO E RECOMENDAÇÕES	62
	REFERÊNCIAS	65
	APÊNDICE 1 – CÓDIGO DO PROGRAMA ELABORADO EM PASCAL COM O MÉTODO 1	68
	APÊNDICE 2 – CÓDIGO DO MÉTODO 2	85
	APÊNDICE 3 – HISTOGRAMAS DAS VELOCIDADES MÉDIAS	89

1 INTRODUÇÃO

Os veículos vêm sendo aperfeiçoados desde sua criação, buscando trazer cada vez mais conforto e segurança para seus ocupantes. Por este motivo, a ideia de se desenvolver um veículo que seja autônomo é um sonho que não nasceu neste século XXI, mas sim, desde o início do século XX, mais precisamente, em 1921 foi exibido o primeiro veículo não tripulado, sendo este controlado por sinais de rádio (ESTADÃO, 2020). Desde então, a indústria automobilística e algumas empresas de tecnologia têm investido em pesquisas para o desenvolvimento de veículos autônomos. Hoje, além dos estudos de melhoria dos veículos autônomos, abriu-se novas frentes de estudos no contexto da inserção desses veículos no trânsito, como discutido por Heymann e Degani (2019). Segundo Tachet, Santi, *et al.* (2016), com a direção autônoma o tempo de resposta para iniciar um cruzamento pode ser a metade do tempo que um condutor humano levaria, resultando em um aumento da capacidade do tráfego veicular em 1,5 vezes e uma redução de 2 vezes no atraso, pois a maior capacidade do sistema viário está mais relacionada com um fluxo consistente e com uma velocidade constante, do que os veículos precisarem trafegar mais rapidamente. Portanto, o desenvolvimento de controladores semafóricos para veículos autônomos trará significativas melhorias na qualidade do trânsito, onde a grande redução do número de paradas em cruzamentos será o maior diferencial.

Buscar por melhorias no trânsito vai além dos benefícios com a diminuição do tempo de deslocamento, mas sim, reflete em uma melhor qualidade de vida tanto das pessoas que compõem o trânsito, quanto a aquelas que não estão diretamente envolvidas nele. Um exemplo é a diminuição da emissão de gases por conta da queima de combustível, melhorando a qualidade do ar e com isso a saúde das pessoas que vivem em cidades de médio e grande porte. Os motoristas também se beneficiariam com um trânsito repleto de veículos autônomos, pois quem era motorista passará a ser um ocupante, fato esse que poderá trazer uma melhor saúde mental ao se remover o fator estresse que o trânsito gera aos motoristas (AHMED e RONY, 2021).

1.1 HIPÓTESE

Dado um cenário onde todos os veículos que trafegam em uma região sejam totalmente autônomos, pode-se implementar controladores de semáforos mais eficientes, os quais permitirão um controle de tráfego com fluxo contínuo nos cruzamentos, de tal maneira que o número de veículos autônomos retidos pelo semáforo se aproxime de zero, sem que haja colisão dos veículos em movimentos conflitantes na área de cruzamento. O novo algoritmo para o controlador de interseção aqui proposto, o qual poderá viabilizar tal eficiência dos controladores, se responsabiliza em fornecer uma janela de tempo-espço para cada veículo realizar o cruzamento, sem que haja a necessidade que os veículos solicitem uma reserva de tempo-espço.

1.2 OBJETIVOS

- Objetivo Geral

Desenvolvimento de um modelo computacional que maximize o fluxo contínuo de veículos autônomos em cruzamentos, sem que haja colisão de tais veículos que trafegam em movimentos conflitantes.

- Objetivos Específicos

- a) Simular o tráfego de veículos totalmente autônomos em vias conflitantes;
- b) Implementar dois métodos para gerenciar a travessia dos veículos autônomos em cruzamentos;
- c) Utilizar as simulações para avaliar a eficiência dos métodos adotados;
- d) Analisar a existência de alguma correlação da eficiência dos métodos com a distância de atuação do controlador e com as diferentes demandas de veículos.

1.3 JUSTIFICATIVAS

A criação de veículos totalmente autônomos, ou seja, que dispensam a necessidade de um motorista para conduzi-los já passou a ser realidade. Portanto, a cada ano que se passa os veículos autônomos tendem a ser cada vez mais comuns nas ruas, pois estima-se que 75% dos veículos do mundo serão autônomos até o ano de 2040 (RELEASES, 2012), fato este que permitirá o desenvolvimento de controladores de semáforos próprios para tais veículos. No entanto, o fato dos veículos autônomos levarem, ainda, alguns anos para se tornarem comuns, faz com que se tenham poucos estudos sobre como obter proveito da tecnologia de tais automóveis, para um melhor controle de tráfego nos cruzamentos.

Os novos controladores podem tornar os semáforos inteligentes no quesito de reduzirem o tempo de deslocamento dos veículos autônomos, os quais poderão trafegar com velocidade quase constante, garantido um melhor conforto para os ocupantes dos mesmos, pois o algoritmo de controle semaforico dessa pesquisa evita que os veículos parem nos cruzamentos. Outra vantagem é a contribuição com o meio ambiente, pois com a redução de paradas implica em um menor gasto de energia motora, sendo que este fato também auxilia na melhora de autonomia de deslocamento. Segundo Sousa (2011), o motorista demora uma média de 1,3 segundos para perceber e reagir mediante a mudança do sinal luminoso do semáforo de verde para amarelo, da mesma forma ocorre para a mudança do sinal luminoso de vermelho para verde, ou seja, além do tempo de espera para o semáforo abrir, o motorista perde alguns segundos para perceber que o semáforo abriu e mais um tempo de reação para iniciar o movimento acelerado do veículo. Portanto, se subtrair tais tempos do sistema viário, melhora-se a fluidez do trânsito.

1.4 ORGANIZAÇÃO DO DOCUMENTO

A tese está organizada da seguinte forma:

- No Capítulo 2 é apresentada uma revisão bibliográfica.

- No Capítulo 3 é feita a descrição do modelo computacional proposto e um estudo dos parâmetros a serem usados no modelo. Depois, apresentam-se os algoritmos de controle simultâneo para o cruzamento em interseções viárias para veículos autônomos.
- No Capítulo 4 encontra-se a descrição do local de estudo e os resultados obtidos.
- No Capítulo 5 são apresentadas as conclusões e algumas sugestões para a realização de futuras pesquisas.
- No Apêndice 1 encontra-se o algoritmo de simulação com o método 1.
- No Apêndice 2 encontra-se o algoritmo apenas do método 2.
- E, finalmente, no Apêndice 3 encontra-se alguns histogramas das velocidades médias dos veículos quando aplicado o método 1.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo apresenta-se uma revisão bibliográfica dos principais trabalhos estudados sobre controle de veículos autônomos e comentários sobre os diferenciais da proposta deste trabalho com os outros similares. E por fim, algumas definições pertinentes ao trabalho são exibidas.

2.1 REVISÃO BIBLIOGRÁFICA

No artigo desenvolvido por Vanmiddlesworth, Dresner e Stone (2008) os autores propuseram uma série de protocolos para realizar coordenação de veículos autônomos em interseções viárias, onde o uso de infraestruturas para tal controle não se faz necessária, pois o mecanismo é baseado em comunicação ponto-a-ponto, de maneira que os agentes (veículos autônomos) tenham todas as informações relevantes do trecho da malha viária por onde estão trafegando, a posição e a velocidade que o veículo se encontra. Através de uma rede sem fio os agentes se comunicam entre si. Tal comunicação se dá por duas mensagens, sendo uma de *reivindicação* para que se reserve um espaço e tempo na interseção para o veículo e um valor booleano é passado para indicar se o veículo remetente está parado ou não no cruzamento, e a outra mensagem é o *cancelar* uma reserva de espaço e tempo, esta mensagem pode vir por parte do próprio agente que fez uma *reivindicação* ou de outro agente que já tinha a reserva do espaço e tempo reivindicado. Quando um agente se comunica com outros, a mensagem enviada inclui o Número de Identificação do Veículo (*VIN*) e a Identificação da Mensagem (*message_id*), onde um contador gera um *message_id* novo para cada nova mensagem. Além do protocolo de comunicação foram adotadas algumas regras de prioridade e domínio para evitar conflitos entre os agentes e outras regras para impedir a ação de agentes maliciosos. Como resultado o trabalho apresentou, através de simulações computacionais de um cruzamento com quatro paradas, que o sistema não gerenciado desenvolvido pode lidar com o dobro da frequência de veículos do que o sistema de sinal de pare, mantendo aproximadamente o mesmo tempo de atraso, onde para o sistema de sinal de pare o tráfego começa a ficar inviável acima de 0,35 veículos por segundo. Portanto, o modelo descrito no artigo possui vantagens significativas sobre o sistema

com sinalização de placas de pare, onde este é melhor empregado do que semáforos em cruzamentos com baixo fluxo de veículos.

A transição dos veículos não autônomos para veículos autônomos nas rodovias levará um longo tempo segundo Au, Zhang e Stone (2016), e pensando nesse processo de mudança, os autores apresentaram em seu artigo um novo protocolo chamado *Semi-Autonomous Intersection Management (SemiAIM)*, que permite que veículos semiautônomos e totalmente autônomos atravessem um cruzamento com várias direções simultaneamente. O protocolo que foi proposto é uma adaptação do protocolo proposto pelos autores Dresner e Stone (2008), e o controle da interseção é baseado em reserva, onde os veículos solicitam um intervalo de espaço-tempo na interseção e o gerente de interseção decide se deve conceder ou rejeitar as reservas solicitadas de acordo com a política de controle da interseção. Com isso, foram realizadas várias simulações considerando um cruzamento com 3 faixas para cada direção, onde os veículos foram gerados seguindo a Distribuição de Poisson com um nível de tráfego de 360 veículos por hora em cada faixa. Sendo o desempenho da interseção medido pelo *atraso médio*, onde este foi calculado pelo aumento de tempo de percurso em relação a uma travessia livre, ou seja, sem que o veículo precisasse reduzir a sua velocidade. Os veículos implementados de forma incremental nas simulações foram de 5 tipos: veículos não autônomos, 3 tipos de veículos semiautônomos e os totalmente autônomos. E por meio de resultados empíricos os autores mostraram que esse novo protocolo pode diminuir 46% o atraso do tráfego quando a maioria dos veículos deixa de ser não autônomos para ser semiautônomos e quando a taxa de veículos totalmente autônomos aumenta o atraso médio diminui ainda mais. Fatos esses que mostram que o *SemiAIM* é uma boa solução para o processo de transição de veículos comuns para veículos autônomos.

Para resolver o problema de navegação autônoma, o autor Pissardini (2014) propôs uma arquitetura de tomada de decisão baseando-se na sistematização das fontes de dados, a qual o veículo deve seguir uma hierarquia de prioridade para o tratamento das informações obtidas pelo mesmo, e também em uma Grade de Prioridades desenvolvida pelo autor. A arquitetura proposta foi denominada de VAT-EVA, e busca atender qualquer tipo de veículo autônomo para qualquer cenário de operação, sendo assim, sua formulação apoia-se em um modelo cinemático simples.

Na arquitetura proposta, a tomada de decisão é o principal processo, o qual busca organizar e gerar opções de ações para a locomoção do veículo, sendo este um processo cíclico integrado ao sistema de mobilidade e que através de um conjunto de hierarquias, busca simular o comportamento de um condutor humano para a tomada de decisões. A Grade de Prioridades composta na arquitetura VAT-EVA tem como objetivo modelar a situação presente do trânsito a qual o veículo está inserido, e estabelecer qual região ao redor do veículo possui o maior espaço livre para sua navegação combinado ao planejamento de sua rota. A arquitetura foi implementada e testada em uma plataforma robótica que simulou três tipos de condução, sendo elas: navegação realizada por um condutor humano; navegação autônoma reativa, utilizando-se apenas de dados fornecidos pelos sensores do veículo, e por fim, a navegação híbrida, que combina o planejamento da rota com informações providas dos sensores para realizar a navegação. Segundo os testes realizados, o veículo robótico denominado Robodeck, executou sua missão mesmo que os dados de posicionamento tenham sido obtidos de maneira simulada, e também o veículo respeitou as regras de trânsito impostas nos cenários de navegação. No entanto, o veículo de teste (Robodeck) colidiu algumas vezes, mas segundo o autor, isso ocorreu pela limitação de aquisição de dados e a baixa cobertura de sensores.

No trabalho descrito por Wei (2015) foram apresentados 4 algoritmos de desvio de obstáculos aplicável a veículos autônomos, sendo eles: algoritmo do Inseto 1 e 2; algoritmo do Inseto Tangencial e por último o algoritmo de Campo de Força Virtual. Como já sugere o nome, os três primeiros algoritmos foram inspirados na forma como os insetos atuam para realizar o desvio de obstáculos fixos, como, as formigas. Os obstáculos são identificados pelo veículo através dos sensores que o compõem e para realizar o desvio, o movimento do veículo se divide tanto em direção ao destino final, quanto ao contorno do obstáculo. Onde cada algoritmo possui o seguinte método:

- Inseto 1: O veículo move-se em direção ao destino e se encontrar um obstáculo no caminho, então este o contorna, obtendo em todo o momento a distância entre o ponto de contorno até seu destino. Após realizar o contorno completo do obstáculo o veículo tem as informações das distâncias dos pontos de contorno ao destino. Se existir um ou mais pontos que possibilite o veículo seguir em direção

ao destino, então o melhor é escolhido com base na menor distância deste ao destino e o veículo move-se até esse ponto e segue em direção ao alvo. Caso não haja nenhum ponto desobstruído, o destino é inalcançável e o algoritmo se encerra sem êxito.

- Inseto 2: É uma melhoria do Inseto 1, onde o veículo traça seu percurso até o destino e caso se depare com algum obstáculo, este o contorna até cruzar com a rota previamente traçada, sendo assim, o veículo não precisa contornar o obstáculo por completo se o caminho estiver desobstruído, otimizando assim, a distância percorrida em relação ao método anterior. Caso o veículo contorne completamente o obstáculo, então significa que não existe um caminho desobstruído.
- Inseto Tangencial: É um aperfeiçoamento do algoritmo do Inseto e para este método o veículo necessita de um sensor que realize detecções em 360° com um alcance de raio R . Quando um obstáculo é detectado, este passa a ser mapeado, e parte do contorno do obstáculo passa a ser conhecido, tendo assim, informações dos pontos limites desta parte do contorno, com isso o algoritmo calcula qual dos dois pontos é mais próximo do destino, buscando assim contornar o obstáculo de forma a tangenciar este ponto. No entanto, à medida que o veículo se aproxima do obstáculo, novos pontos limites são encontrados, alterando-se o ponto de referência a ser tangenciado, até que o veículo tenha desviado do obstáculo mantendo-se em direção ao seu destino. Caso o veículo tangencie todo o obstáculo, então o algoritmo é encerrado, pois o caminho encontra-se obstruído. Nota-se que, quanto maior é o R (alcance do sensor), melhor será a eficácia do algoritmo ao realizar o desvio do obstáculo.
- Campo de Força Virtual: Este método é composto pela combinação de dois métodos, sendo eles: Função Potencial e o de Grade de Certeza. O primeiro método é responsável por atribuir ao veículo e ao obstáculo um sinal positivo e ao destino um sinal negativo, onde se aplica a lógica da atração para sinais opostos e repulsão para sinais iguais, portanto, o veículo se move na direção da resultante das forças de tração e repulsão, sendo assim atraído pelo destino e repellido do

obstáculo. O método de Grade de Certeza fica responsável por mapear o ambiente através dos dados fornecidos pelos sensores, onde a grade próxima a borda do obstáculo recebe um valor que é a intensidade da força de repulsão, sendo que, quanto mais próximo é a distância dos veículos a borda do obstáculo, então maior é a intensidade de repulsão.

O autor Wei (2015), escolheu em seu trabalho realizar a implementação do algoritmo do Inseto 2 na linguagem Java e testá-lo em uma plataforma robótica, usando 4 sensores ultrassônicos, distribuídos um para cada lado do *Robodeck* (carrinho robótico). Após fazer experimentos com 3 cenários distintos, conclui-se que o algoritmo adotado se mostrou eficaz para realizar o desvio e contorno dos obstáculos que não foram mapeados inicialmente. E quando o obstáculo tem a forma convexa, o algoritmo do Inseto 2 se mostra mais eficiente que o algoritmo de Campo de Força Vetorial, por não ficar preso em um ponto de mínimo intermediário entre o obstáculo e o destino. Durante os experimentos ocorreram algumas falhas oriundas da capacidade do *Robodeck* na precisão ao fazer curvas, e também da eficiência e o número de sensores utilizados no *Robodeck*.

Em outro trabalho envolvendo o estudo do deslocamento de veículos autônomos, apresentado por Boas (2011), foi implementado um algoritmo para resolver o problema de mapeamento e a localização simultânea dos veículos autônomos nos ambientes. O algoritmo busca reduzir os erros de posicionamento do veículo e dos obstáculos que este identifica dentro de um mapa construído pelo veículo para conhecer o espaço ao seu redor. O mapa é representado por uma matriz dinâmica que representa um plano cartesiano, sendo cada elemento, um quadrado do *grid*. Inicialmente a matriz possui em seus elementos o valor zero e quando os sensores do veículo identificam algum obstáculo é somado o valor um para aquela posição. Para auxiliar na construção do mapa, foi adotado um método probabilístico através do Teorema de Bayes, baseando-se na sobreposição das informações da localização dos obstáculos. E a posição do veículo é dada pelo DGPS (*Differential Global Position System*), onde é refinada por um filtro ($a - b$) para estimar a posição do veículo dentro do seu próprio mapa. Após realizar alguns testes com veículo autônomo em 4 circuitos, o método apresentado pelo autor Boas (2011), se mostrou

eficiente por fazer uma representação mais fiel da posição dos obstáculos e do veículo no mapa.

Buscando realizar uma navegação segura de um veículo autônomo ao planejar o movimento e o desvio de obstáculos, Lima (2010) implementou métodos que realizam o controle das velocidades para o veículo. E para o planejamento do movimento utilizou-se o método de Campos Vetoriais de Velocidade, o qual foi complementado pelo método da Janela Dinâmica (DWA), responsável pelo desvio de obstáculos não previamente identificados. Assim sendo, o DWA calcula uma alternativa melhor de movimento, quando a opção dada pelo Campo Vetorial de Velocidade é inválida devido a possibilidade de colidir com algum obstáculo. A identificação dos obstáculos se dá por leitura de sensores integradas ao veículo, onde o ambiente é descrito por uma grade de ocupação, semelhante ao trabalho de Boas (2011), porém os valores da grade de ocupação são definidos por uma simplificação do Filtro de Bayes, pelo fato do ambiente considerado ser estático, ou seja, as leituras dos sensores não dependem das medições anteriores. Como o campo vetorial de velocidade é um método desenvolvido para fornecer vetores de velocidade para conduzir um robô holonômico pontual, fez-se necessário o autor Lima (2010) resolver este problema para um veículo autônomo. E a solução encontrada foi o método de Linearização por Realimentação Estática (FBL).

Para testar os métodos, foram realizados dois tipos de testes, uma simulação computacional, a qual forneceu bons valores dos pesos para se calibrar a função objetivo do DWA, e outro em um veículo autônomo desenvolvido pela Universidade Federal de Minas Gerais, denominado como CADU. Nos testes realizados com o veículo em um ambiente estático, se mostrou segura sua navegação com a utilização dos métodos propostos pelo autor, e segundo este, conclui-se também, que a utilização de mais sensores pode melhorar o desempenho da navegação do veículo, podendo este realizar testes mais complexos de navegação.

Diferente dos três últimos trabalhos descritos anteriormente, os quais buscaram desenvolver e testar soluções para o controle de movimentação lateral de veículos autônomos, o trabalho de Dias (2013) buscou trazer um modelo de controle longitudinal para veículos autônomos, ou seja, um controle sobre as intensidades a serem aplicadas na aceleração e na frenagem. Entretanto, o modelo descrito por Dias

(2013) se concentrou especificamente no controle da velocidade longitudinal, através de um modelo matemático da dinâmica do veículo, ao invés, da outra forma comumente usada para abordar o problema em questão, que se baseia na lógica *fuzzy* (FREITAS, 2010) a qual não exige um elevado conhecimento da dinâmica do veículo. A modelagem desenvolvida, envolve a força de inércia sobre o veículo, sendo está igual ao somatório das seguintes forças: arrasto aerodinâmico, frenagem, peso relativo ao grau de inclinação, resistência viscosa dos componentes do sistema de transmissão do veículo e a força exercida pelo motor às rodas ligadas ao sistema de transmissão. Portanto, os principais elementos envolvidos na dinâmica do veículo foram o motor-transmissão e a carroceria, pretendendo assim, gerar comandos ao veículo para compensar a não linearidade da dinâmica do mesmo e como consequência reduzir as trocas bruscas de marcha. O modelo descrito no trabalho de Dias (2013) foi implementado e testado em um veículo autônomo desenvolvido pela Universidade Federal de Minas Gerais, o já mencionado CADU, no trabalho anterior, o teste consistiu em duas voltas completas de um percurso, totalizando um deslocamento de 3 km em um tempo de 10 minutos, onde, durante o teste, variou-se os graus de velocidade e até mesmo uma parada total adicionada ao teste. Os resultados, segundo o autor foram razoáveis, pois não necessitou da intervenção humana e o veículo apresentou um melhor comportamento na aceleração do que na desaceleração, mas a modelagem apresentou dificuldades por exigir informações detalhadas dos elementos não lineares envolvidos na dinâmica longitudinal de um veículo.

Para os veículos autônomos obterem uma trajetória com a intenção de evitar obstáculos, um modelo não linear de controle preditivo (NMPC) foi desenvolvido por ABBAS, MILMAN e EKLUND (2017), os quais consideraram também os veículos autônomos viajando em velocidades realistas. Sendo o NMPC testado por meio de simulações em cenários onde os obstáculos são estáticos nas estradas. E para simplificar a dinâmica do veículo, um modelo de bicicleta foi considerado para prever os estados futuros do veículo na estrutura. Para validar a eficiência do modelo, foi comparado com um modelo de veículo denominado *CarSim*, sendo este não linear e de alta fidelidade. Os resultados mostram que o controlador NMPC fornece desempenho de rastreamento online satisfatório em um cenário realista em velocidades normais de estrada enquanto ainda satisfaz as restrições de tempo real.

No teste final, o NMPC mostrou-se capaz de navegar com sucesso em vias desconhecidas e complexas, evitando obstáculos. Foi observado que horizontes mais longos proporcionaram um melhor desempenho na evasão de obstáculos, permitindo manobras antecipadas e reduzindo a invasão da faixa contrária. Foi estabelecido um limite de 25 iterações por etapa de tempo do controlador para manter a implementação em tempo real dentro do tempo limite. No entanto, percebeu-se que ao limitar o número de iterações a estabilidade pode ser comprometida.

Buscando resolver outro problema, que os veículos autônomos vão precisar lidar, que é a decisão de mudança de faixa, principalmente em avenidas onde o fluxo é intenso, os autores JI, ORSAG e HAN (2021) propuseram um modelo de decisão baseado no “Jogo de *Stackelberg*”, onde este jogo é hierárquico e cada jogador recebe a função de líder ou seguidor. Portanto, o veículo que pretende mudar de faixa precisa interagir com o veículo da faixa que deseja ir, e para saber a intenção do veículo, se é de recusar ou aceitar que o outro siga a sua frente, então sua velocidade é monitorada. Com isso, se o veículo da outra faixa desacelera, então o veículo autônomo que deseja mudar de faixa interpreta que o outro aceitou a sua entrada, mas antes de efetivamente mudar de faixa calcula a viabilidade, como por exemplo, se haverá espaço suficiente. Portanto, o método consiste basicamente em comparar a possibilidade do veículo autônomo que deseja mudar de faixa, com a possibilidade de aceitação do veículo que se encontra na faixa desejada. Sendo esta possibilidade calculada por meio de uma função denominada de Polidez (P_i), a qual é estimada por meio de uma modificação do “Modelo de Motorista Inteligente”, o qual representa o processo de tomada de decisão de um motorista. Então, o modelo proposto pelos autores, abrange a interação tanto entre veículos autônomos, como também, de um veículo autônomo e um veículo controlado por um humano. Os autores testaram o modelo para 3 cenários, onde ambos consistiam em um fluxo denso e um deslocamento lento, sendo os parâmetros para as simulações como: aceleração constante de $0,97 \text{ m/s}^2$, velocidade máxima de $2,5 \text{ m/s}$, velocidade lateral (para mudança de faixa) de 2 m/s , comprimento de todos os veículos iguais a 5 m e distância de segurança entre os veículos de 2 m . Após os testes, pode ser concluído que o método desenvolvido para a tomada de decisão de mudança de faixa, pode ser aplicado para situações reais, tanto para veículos conectados e não conectados. E o fato do veículo que busca mudar de faixa não precisar interagir simultaneamente com

os outros veículos da via, possibilita um custo computacional baixo, viabilizando a implementação nos *hardwares* de veículos autônomos.

Uma preocupação que surge com a integração dos veículos autônomos no trânsito, tem sido no sentido de como será a interação destes veículos com os pedestres e como os veículos autônomos poderão identificar a intenção dos pedestres em atravessar uma via ou não. Buscando explorar e propor soluções para resolver tais conflitos, alguns pesquisadores HEYMANN e DEGANI (2019), RASOULI e TSOTSOS (2020) e G. MARKKULA, PORTOULI, *et al.* (2020) têm sugerido o uso de protocolos, inseridos na programação de tais veículos para lidar com estas situações, onde também os pedestres precisarão seguir algumas regras e para algumas interseções será indicado fazer mudanças na infraestrutura, com o possível uso de sensores antes da faixa de pedestre.

No trabalho de ZHAO, KNOOP e WANG (2020) foi estabelecido um modelo de movimento veicular bidimensional com base no controle ótimo do caminho mais rápido e mais suave da trajetória para veículo de condução humana, considerando movimentos em cruzamentos, onde não existem pistas estruturadas. O modelo foi pensado para ajudar os projetistas de infraestrutura a avaliar o desempenho de segurança e eficiência de diferentes esquemas de projeto e a refinar e otimizar os esquemas. Porém, o modelo pode ser aplicado como parte do algoritmo de controle em veículos autônomos no momento de realizar alguma curva, pois duas variáveis de controle são consideradas para representar como os motoristas programam seu comportamento de direção, incluindo girar o volante e acionar os pedais de freio ou acelerador. Portanto, o método utilizou tanto a direção quanto a aceleração como entradas do modelo, onde a distância percorrida é principal variável independente, em vez de adotar o tempo como a principal variável. O modelo se assemelha a entradas em curvas realizadas por condutores humanos, considerando o emprego da direção (volante) e aceleração longitudinal (pedais), sendo tais controles realizados pela decisão de um algoritmo de solução numérica iterativa, baseado no Princípio Mínimo de Pontryagin. Posteriormente o modelo foi aperfeiçoado ao considerar a influência da interação dos outros veículos ao apresentarem a intenção de realizar uma conversão para a esquerda ou à direita (ZHAO, KNOOP e WANG, 2022).

O modelo desenvolvido por LIU, ZHAO, *et al.* (2022), apresenta uma abordagem de controle conjunto, o qual otimiza simultaneamente os intervalos de tempo dos semáforos e as trajetórias dos veículos em pelotões, onde são considerados Veículos Autônomos e Conectados (CAV) nas proximidades das interseções semaforizadas, com vias de sentido duplo e duas faixas para cada sentido. As acelerações do veículo e os comprimentos de fase do sinal foram otimizados em conjunto com o objetivo de maximizar o conforto de condução e minimizar o atraso de viagem, sujeito a restrições de movimento. Os semáforos foram otimizados para liberar o maior número de veículos dentro do ciclo, sob as taxas de chegada de veículos desequilibradas, assim o tempo verde raramente é desperdiçado sem descarregar nenhum veículo. A abordagem além de considerar as taxas de chegada de veículos, também levou em conta as velocidades e posições dos veículos para buscar liberar mais veículos durante as fases verdes. Portanto, o modelo obtido apresentou características não lineares, mas posteriormente passou por um processo de linearização da função objetivo e de algumas restrições. Com isso, o problema pode ser resolvido usando o modelo de Programação Linear Inteira Mista (MILP). Com tudo, pode ser concluído que apesar do modelo poder ser resolvido com êxito, a dimensionalidade do problema de controle aumenta drasticamente com o aumento do número de veículos, resultando em um longo tempo de processamento e não sendo possível sua solução em tempo real atualmente.

2.2 COMENTÁRIOS

Nesta tese, apesar de possuir uma ideia semelhante a outros trabalhos em que são realizadas reservas de espaço-tempo, apresenta um novo modelo que não trabalha com pedido de reserva e confirmação de aceitação ou não, pois o controlador de interseção passa algumas instruções aos controles do veículo, de maneira a garantir um cruzamento bem sucedido. Outro diferencial no presente trabalho com os demais descritos na literatura, é o fato deste trazer um modelo computacional para um controlador de cruzamento (interseção) de veículos autônomos de maneira que estes podem fluir continuamente nas vias, ou seja, não sendo necessário parar nos cruzamentos. Além do modelo, foram realizadas simulações computacionais

baseadas no mesmo, onde, apesar da estrutura viária considerada ser mais simples que algumas simulações da literatura, os tamanhos dos veículos simulados foram variados, fato este levado em consideração por alguns autores AU, ZHANG e STONE (2016), mas não empregado em suas simulações. E a velocidade inicial considerada nesta tese, é superior à usada por Tian, Li, *et al.* (2020) em suas simulações, apesar deste trazer um modelo de gerenciamento de cruzamento não sinalizado.

2.3 DEFINIÇÕES

Segundo o Código de Trânsito Brasileiro em seu artigo 1º, parágrafo 1, o Trânsito é definido como: Considera-se trânsito a utilização das vias por pessoas, veículos e animais, isolados ou em grupos, conduzidos ou não, para fins de circulação, parada, estacionamento e operação de carga ou descarga (BRASIL. CÓDIGO DE TRÂNSITO BRASILEIRO, 1997).

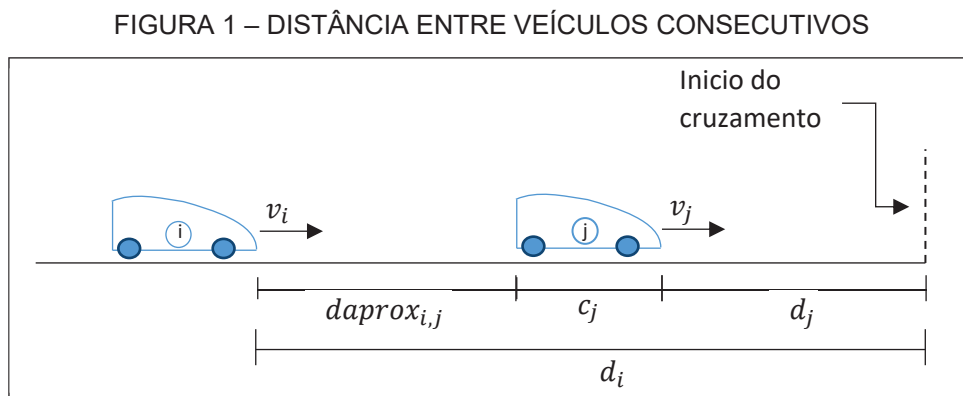
Os veículos autônomos podem ser classificados por 6 níveis de automação, onde, no nível mais elevado o desempenho do sistema é equivalente ao de um ser humano dirigindo para todos os cenários, sem nenhuma restrição (ALOUFI, 2018), (SHADRIN e IVANOVA, 2019), ou seja, não sendo necessário nenhuma intervenção humana para que o veículo se desloque de seu ponto de origem ao seu destino. Portanto, para o desenvolvimento deste trabalho define-se que todo veículo autônomo mencionado se refere ao veículo totalmente autônomo (nível 6), onde estima-se que este nível de automação será alcançado no ano de 2035 (BIMBRAW, 2015).

3 MODELO E MÉTODO

O modelo computacional desenvolvido leva em conta algumas situações para que os veículos realizem a travessia pelo cruzamento de forma intermitente e segura, onde algumas situações não podem ocorrer, sendo elas: a colisão de um veículo na traseira do veículo que está a sua frente, a colisão de veículos no cruzamento, o veículo ultrapassar a velocidade máxima permitida da via e o veículo parar na pista sem necessidade. Portanto, para resolver essas situações, algumas análises e cálculos foram realizados, os quais são desenvolvidos a seguir.

3.1 CÁLCULO DA DISTÂNCIA ENTRE VEÍCULOS CONSECUTIVOS

Este cálculo consiste em determinar qual será a distância entre um veículo e outro imediatamente a sua frente, independentemente do tipo de movimento que os veículos estejam realizando, seja ele um movimento uniformemente variado (MUV) ou apenas um movimento uniforme (MU). Para melhor entendimento do cálculo a ser exibido, observa-se a Figura 1 a seguir.



FONTE: O autor (2023)

Na Figura 1 acima, v_i e v_j são as velocidades dos respectivos veículos, onde estes encontram-se a uma distância d_i e d_j do início do cruzamento, sendo c_j o comprimento do veículo mais à frente. Portanto, ao analisar a Figura 1 tem-se que a distância entre os veículos i e j pode ser representada pela Equação (1) adiante:

$$d_{approx_{i,j}} = d_i - d_j - c_j. \quad (1)$$

Como as distâncias d_i e d_j variam em função do tempo, então estas podem ser calculadas através da equação horária do espaço, conforme a equação a seguir:

$$s = s_0 + v_0 t + \frac{a t^2}{2}, \quad (2)$$

Onde,

t : tempo [s];

s : distância percorrida em t segundos [m];

s_0 : distância inicial ($t = 0$) [m];

a : taxa de aceleração [m/s^2];

v_0 : velocidade inicial [m/s];

Fazendo uso da Equação (2) e substituindo s por d_i e d_j , s_0 por d_{0i} e d_{0j} , v_0 por v_{0i} e v_{0j} e a por a_i e a_j . Pelo fato das distâncias d_i e d_j tenderem a diminuir com o passar do tempo, então as duas últimas parcelas da Equação (2) recebem sinais negativos. Com tudo, descreve-se as seguintes equações:

$$d_i = d_{0i} - v_{0i} t - \frac{a_i t^2}{2}, \quad (3)$$

e

$$d_j = d_{0j} - v_{0j} t - \frac{a_j t^2}{2}. \quad (4)$$

Substituindo as Equações (3) e (4) na Equação (1) temos,

$$d_{approx_{i,j}} = d_{0i} - d_{0j} - c_j + (v_{0j} - v_{0i})t + (a_j - a_i) \frac{t^2}{2}. \quad (5)$$

A Equação (5) fornece a cada instante t , qual será a distância de aproximação dos veículos i e j que são consecutivos um ao outro na mesma faixa de uma via. Na situação ilustrada pela Figura 1 podem ocorrer 3 casos em relação a distância dos veículos, sendo estes:

1º caso – Se os veículos i e j se moverem com a mesma velocidade ($v_i = v_j$), então a distância entre eles permanecerá constante com o passar do tempo.

2º caso – Se a velocidade do veículo j for maior que a velocidade do veículo i ($v_i < v_j$), então a distância entre eles aumentará com o passar do tempo.

3º caso – Se a velocidade do veículo i for maior que a velocidade do veículo j ($v_i > v_j$), então a distância entre eles diminuirá com o passar do tempo.

Dos casos apresentados acima, o 3º caso é o que merece maior atenção, pois indica uma futura colisão do veículo i na traseira do veículo j . Portanto, a distância de aproximação entre esses veículos precisará ser monitorada a cada instante, onde a distância entre eles não poderá ser menor que uma distância mínima necessária para se realizar a desaceleração do veículo i . Esta distância mínima será discutida no tópico a seguir.

3.2 CÁLCULO DA DISTÂNCIA MÍNIMA ENTRE VEÍCULOS CONSECUTIVOS

O cálculo da distância mínima entre dois veículos consecutivos em movimento uniformemente variado (MUV) ou movimento uniforme (MU), é de extrema importância para se determinar qual a distância mínima necessária para que um certo veículo desacelere até alcançar a mesma velocidade que o veículo a sua frente, antes que ocorra uma colisão entre eles. Para tal cálculo considera-se a Equação (7) de Torricelli a seguir:

$$v^2 = v_0^2 + 2a\Delta s, \quad (7)$$

onde,

v : velocidade final [m/s];

v_0 : velocidade inicial [m/s];

Δs : distância percorrida [m];

a : taxa de aceleração [m/s^2].

Considerando os veículos i e j , conforme representado na Figura 1, e a Equação (7), substituindo a velocidade final v pela velocidade final que o veículo i deverá assumir, sendo esta igual à velocidade do veículo j ($v = v_j$), e a velocidade inicial v_0 sendo a velocidade atual do veículo i ($v_0 = v_i$). Como o presente cálculo

refere-se ao 3º caso apresentado anteriormente, onde a velocidade do veículo i é maior que a velocidade do veículo j ($v_i > v_j$), então a taxa de aceleração a do veículo i deve ser substituída por uma taxa de desaceleração de . Com isso, a distância percorrida Δs fornecerá a distância mínima entre os veículos i e j ($\Delta s = dmin_{i,j}$), sendo assim,

$$v_j^2 = v_i^2 - 2 \cdot de \cdot dmin_{i,j}, \quad (8)$$

e isolando a variável $dmin_{i,j}$ da Equação (8), temos:

$$dmin_{i,j} = \frac{v_j^2 - v_i^2}{-2de}. \quad (9)$$

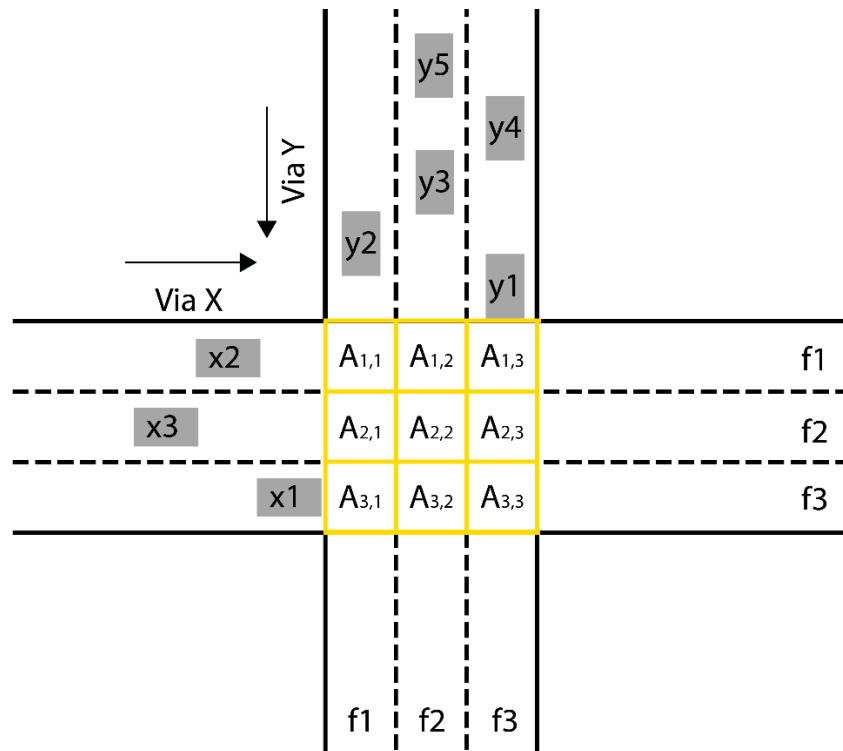
Logo, a Equação (9) fornece qual deverá ser a distância mínima entre os veículos i e j .

3.3 ANÁLISE DE COLISÃO NO CRUZAMENTO

Analisando o comportamento dos veículos em um cruzamento, onde o fluxo deverá fluir de forma intermitente. Pode-se esperar desta situação dois casos, um caso onde haverá colisão e outro que não haverá.

A seguir analisa-se qual será as condições para que não haja colisões de veículos no cruzamento. Para isso, será considerado um cruzamento perpendicular, onde cada via possui três faixas de sentido único, conforme ilustrado na Figura 2 a seguir:

FIGURA 2 – CRUZAMENTO DE DUAS VIAS COM 3 FAIXAS



FONTE: O autor (2023).

Na Figura 2 acima, x1, x2 e x3 nos retângulos em cinza representam os veículos que estão na via x e os y1 ao y5 são os veículos da via y, e f1, f2 e f3 indicam as faixas das vias. Os retângulos com arestas amarelas representados por $A_{1,1}$, $A_{1,2}$, ..., $A_{3,3}$ são as áreas formadas pelas interseções das faixas das vias x e y no cruzamento. Assim, considera-se as áreas como elementos de uma matriz $A_{p,w}$, onde p representa o número da faixa na via x e w representa o número da faixa na via y. Portanto, para que não haja colisões no cruzamento, as áreas $A_{p,w}$ não poderão ser ocupadas por mais de um veículo ao mesmo tempo. Para analisar se haverá colisão ou não, calcula-se quais serão os instantes de tempo que os veículos entrarão e sairão das áreas que compõem a sua respectiva faixa, onde tais instantes de tempo serão representados por $tex_{A_{p,w},ix}$, $tey_{A_{p,w},iy}$, $tsx_{A_{p,w},ix}$ e $tsy_{A_{p,w},iy}$. Por exemplo, se tomarmos o veículo x1, então calcula-se os instantes que este entra nas áreas $A_{3,1}$, $A_{3,2}$ e $A_{3,3}$, sendo $tex_{A_{3,1},1}$, $tex_{A_{3,2},1}$ e $tex_{A_{3,3},1}$ respectivamente, e os instantes que este veículo sai destas áreas, sendo $tsx_{A_{3,1},1}$, $tsx_{A_{3,2},1}$ e $tsx_{A_{3,3},1}$. Logo, o veículo x1 ocupará a área $A_{3,1}$ no intervalo de $tex_{A_{3,1},1}$ à $tsx_{A_{3,1},1}$ segundos, a área $A_{3,2}$ no intervalo de $tex_{A_{3,2},1}$ à $tsx_{A_{3,2},1}$ segundos e por fim a área $A_{3,3}$ no intervalo de $tex_{A_{3,3},1}$ à $tsx_{A_{3,3},1}$ segundos.

Os instantes de tempo, entretanto, são obtidos pelas seguintes equações:

$$tex_{A_{p,w},ix} = \frac{d_{ix} + (we - 1)ly}{v_{ix}}, \quad (10)$$

$$tsx_{A_{p,w},ix} = \frac{d_{ix} + ws.ly + c_{ix}}{v_{ix}}, \quad (11)$$

$$tey_{A_{p,w},iy} = \frac{d_{iy} + (pe - 1)lx}{v_{iy}} \quad (12)$$

e

$$tsy_{A_{p,w},iy} = \frac{d_{iy} + ps.lx + c_{iy}}{v_{iy}}, \quad (13)$$

onde,

$tex_{A_{p,w},ix}$: instante que o veículo ix na via x entra na área $A_{p,w}$ [s];

$tsx_{A_{p,w},ix}$: instante que o veículo ix na via x sai da área $A_{p,w}$ [s];

$tey_{A_{p,w},iy}$: instante que o veículo iy na via y entra na área $A_{p,w}$ [s];

$tsy_{A_{p,w},iy}$: instante que o veículo iy na via y sai da área $A_{p,w}$ [s];

v_{ix} : velocidade atual do veículo ix na via x [m/s];

v_{iy} : velocidade atual do veículo iy na via y [m/s];

d_{ix} : distância atual do veículo ix na via x ao início do cruzamento [m];

d_{iy} : distância atual do veículo iy na via y ao início do cruzamento [m];

c_{ix} : comprimento do veículo ix [m];

c_{iy} : comprimento do veículo iy [m];

pe : número da faixa da via x a qual o veículo entrará;

ps : número da faixa da via x a qual o veículo sairá;

we : número da faixa da via y a qual o veículo entrará;

ws : número da faixa da via y a qual o veículo sairá;

l_x : largura da faixa da via x [m];

l_y : largura da faixa da via y [m].

Portanto, para que um veículo realize a travessia em segurança, uma das inequações abaixo deverá ser satisfeita:

$$tsx_{A_p,w,ix} - tey_{A_p,w,iy} + \xi \leq 0 \quad (14)$$

ou

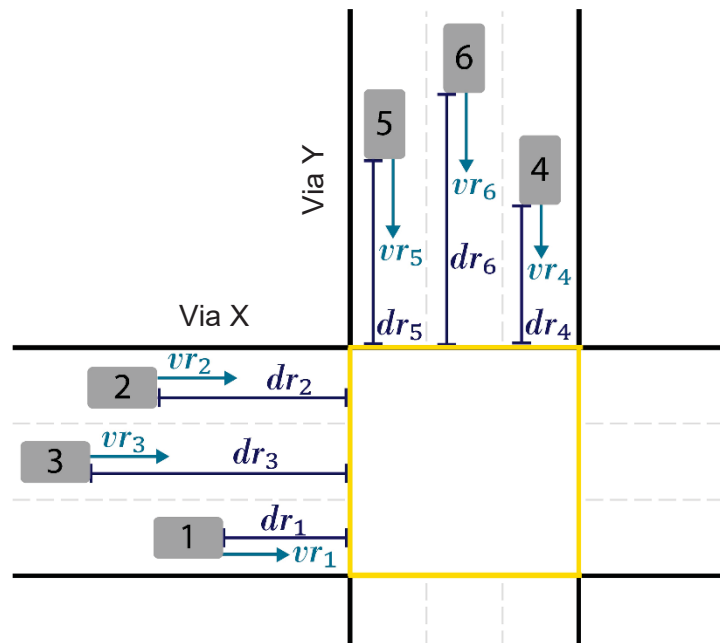
$$tsy_{A_p,w,iy} - tex_{A_p,w,ix} + \xi \leq 0. \quad (15)$$

Quando a Inequação (14) é satisfeita, indica que o veículo i da via x realizará a travessia com sucesso antes que o veículo i da via y , pois os tempos de entrada e saída do veículo da via x nas áreas de cruzamento, são menores que os tempos de entrada e saída do(s) veículo(s) da via y nestas áreas, considerando uma folga de tempo de pelo menos ξ milésimos de segundo. E no caso, em que a Inequação (15) é satisfeita, indica que o veículo iy realizará a travessia com sucesso antes do veículo ix .

3.4 MODELO COMPUTACIONAL

O modelo computacional a seguir é dinâmico e se baseia em uma escala microscópica, pois a posição e a velocidade dos veículos precisarão ser monitoradas individualmente e ajustadas se necessário a cada instante. O presente modelo tem como objetivo, além de fornecer uma travessia segura, minimizar o tempo de viagem dos veículos no raio de atuação do controlador. Portanto, a função objetivo tem como alvo, maximizar a soma dos produtos da velocidade pela distância. Sendo que esta distância é considerada a partir do ponto em que a velocidade do veículo não é mais ajustada pelo controlador, até o fim da travessia. Para melhor compreender a função objetivo, observe a Figura 3 a seguir:

FIGURA 3 – ILUSTRAÇÃO DA FUNÇÃO OBJETIVO



FONTE: O autor (2023).

A função objetivo atingirá seu valor máximo quando o algoritmo do controlador for capaz de determinar a maior velocidade possível para cada veículo, no momento em que eles estiverem com a maior distância possível do início da interseção, de forma que as velocidades dos veículos permaneçam constantes até cruzar completamente a interseção.

$$\text{Max} \sum_{i=1}^n vr_i \cdot dr_i$$

S. a

$$daprox_{ix,jx} - dmin_{ix,jx} - ds \geq 0, ix, jx \in X; \quad (16)$$

$$daprox_{iy,jy} - dmin_{iy,jy} - ds \geq 0, iy, jy \in Y; \quad (17)$$

$$(tsx_{A_p,w,ix} - tey_{A_p,w,iy} + \xi \leq 0) b_i \leq 0, p = 1, \dots, P, w = 1, \dots, W, ix \in X \text{ e } iy \in Y \quad (18)$$

$$(tsy_{A_p,w,iy} - tex_{A_p,w,ix} + \xi)(b_i - 1) \leq 0, p = 1, \dots, P, w = 1, \dots, W, \quad (19)$$

$$ix \in X \text{ e } iy \in Y$$

$$\epsilon \leq v_{ix} \leq v_{max,x}, 0 < \epsilon \leq 10^{-3} \quad (20)$$

$$\epsilon \leq v_{iy} \leq v_{max,y}, 0 < \epsilon \leq 10^{-3} \quad (21)$$

$$\forall v_{ix}, v_{iy} \in \mathbb{R}_+, \forall tex_{A_{p,w},ix}, tey_{A_{p,w},iy}, tsx_{A_{p,w},ix} \text{ e } tsy_{A_{p,w},iy} \in \mathbb{R}_+, \forall b_i \in B$$

Onde, em termos gerais $i \in X \cup Y$, sendo o conjunto X formado pelos veículos que se encontram na via x e o conjunto Y formado pelos veículos que se encontram na via y .

n : número de veículos dentro do raio de controle do controlador, ou seja, $n = |X \cup Y|$;

$tex_{A_{p,w},ix}$: instante que o veículo ix na via x entra na área $A_{p,w}$ [s];

$tsx_{A_{p,w},ix}$: instante que o veículo ix na via x sai da área $A_{p,w}$ [s];

$tey_{A_{p,w},iy}$: instante que o veículo iy na via y entra na área $A_{p,w}$ [s];

$tsy_{A_{p,w},iy}$: instante que o veículo iy na via y sai da área $A_{p,w}$ [s];

$dapprox_{ix,jx}$: distância do veículo ix em relação ao veículo jx à sua frente na via x [m];

$dmin_{ix,jx}$: distância mínima entre o veículo ix e o veículo jx à sua frente na via x [m];

$dapprox_{iy,jy}$: distância do veículo iy em relação ao veículo jy à sua frente na via y [m];

$dmin_{iy,jy}$: distância mínima entre o veículo iy e o veículo jy à sua frente na via y [m];

dr_i : distância que o veículo i estava do cruzamento quando sua velocidade deixa de ser reajustada [m];

ds : distância de segurança entre um veículo e o outro à sua frente [m];

v_{ix} : velocidade do veículo ix [m/s];

v_{iy} : velocidade do veículo iy [m/s];

vr_i : velocidade do veículo i quando sua velocidade deixa de ser reajustada [m/s];

$v_{max,x}$: velocidade máxima na via x [m/s];

$v_{max,y}$: velocidade máxima na via y [m/s].

Como restrições, utilizam-se as Inequações (16) e (17) que garantem que os veículos se mantenham a uma distância segura dos veículos a sua frente, onde a distância entre eles não poderá ser inferior ao parâmetro ds , e as Inequações (18) e (19) são para garantir que não haja colisões no cruzamento, entre veículos com movimentos conflitantes. Porém, as Inequações (18) e (19) não podem ser satisfeitas simultaneamente para o mesmo par de veículos autônomos conflitantes ix e iy , portanto, como a variável b_i pertence a um conjunto binário B , esta é responsável por realizar tal controle. E por fim, as Inequações (20) e (21) que limitam as velocidades mínima e máxima, onde a mínima é tão próxima de zero quanto se deseja e a velocidade máxima não pode ser superior à permitida na via.

A variável de decisão do modelo apresenta-se de forma oculta, sendo esta, o tipo de movimento que o automóvel deve realizar para o próximo segundo, podendo ser acelerado, constante ou desacelerado a_{de} . Portanto, a taxa de aceleração é a variável de decisão, pois esta influencia na velocidade do veículo no próximo instante e por sua vez impacta nos instantes de tempo que o veículo entra e sai de cada subárea do cruzamento.

3.5 FORMULAÇÃO DO ALGORITMO

A resolução do modelo computacional aqui apresentado pode variar de uma baixa a uma alta complexibilidade, onde tudo depende do número de veículos em cada via, bem como a velocidade e o comprimento de cada um, e a distância que cada veículo se encontra do outro a sua frente, pois o modelo precisa ser resolvido em tempo real levando em consideração cada um desses aspectos. Portanto, uma abordagem para este problema considera como parâmetro uma área de controle, onde os veículos que se encontram a partir de uma determinada distância D para chegar ao cruzamento, passam a ter sua taxa de aceleração gerenciadas pelo controlador do cruzamento o qual o veículo está se aproximando.

Os veículos que se encontram na região de controle passam então, a fazer parte do modelo computacional. A cada segundo o modelo é atualizado com o número de veículos, a velocidade, a distância que cada um se localiza do início da interseção e os instantes de tempo que cada veículo entra e sai das subáreas do cruzamento. Por se tratar de um problema dinâmico, então como solução do modelo, é aceito qualquer método que atenda todas as restrições do modelo, sem se prender ao fato de ser uma solução ótima, pois o espaço de busca para cada iteração da simulação é da ordem de I^n , onde I é a quantidade de taxas de aceleração e desaceleração; e n o número de veículos na área de controle.

O código de todo o programa desenvolvido foi implementado na linguagem Pascal usando a IDE Lazarus (LAZARUS AND FREE PASCAL TEAM, 1993). E as simulações foram realizadas em um computador com processador Intel® Core i7-7500U CPU 2.7 GHz, memória RAM de 8 GB e 1T de HDD, sistema operacional de 64 bits e Windows 10 Pro.

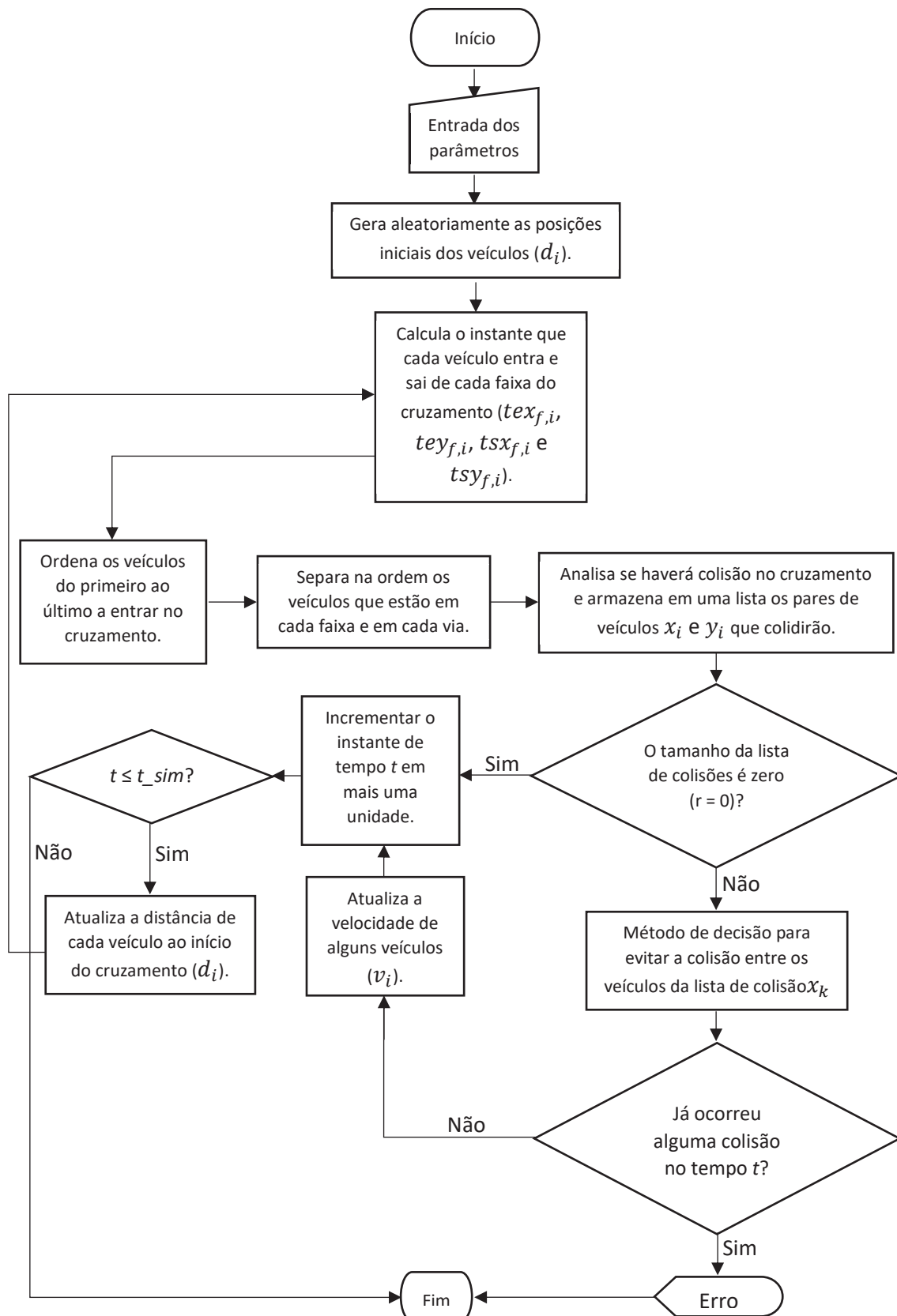
3.5.1 Algoritmo de simulação

Como dados iniciais para a simulação, são fornecidos a distância de controle D ; o número de veículos n ; o número de faixa em cada via F ; a largura das faixas ly e lx ; a velocidade inicial v_i de cada automóvel; a velocidade máxima permitida nas vias $v_{max,x}$, $v_{max,y}$; a taxa de aceleração a e taxa de desaceleração de dos veículos; distância de segurança entre um veículo e o outro à sua frente ds e o tempo de simulação t_{sim} em segundos. A partir de tais dados o programa gera aleatoriamente o comprimento c_i de cada veículo, a via e a faixa f a qual será inserido e a sua distância inicial d_i , sendo esta, a distância que o veículo i se encontra do início do cruzamento, onde estas últimas três informações caracterizam a posição inicial do automóvel, caso esta posição já esteja sendo ocupada por outro veículo, então sorteia-se uma nova posição. Em seguida o algoritmo calcula em quais instantes de tempo cada veículo entra e sai de cada faixa perpendicular a qual este se encontra $tex_{Ap,w,ix}$, $tEy_{Ap,w,iy}$, $tsx_{Ap,w,ix}$ e $tsy_{Ap,w,iy}$. Tais informações são armazenadas em uma matriz, onde cada linha da matriz contém os dados de um veículo, portanto os veículos são identificados pelo próprio índice da linha da matriz. E nas colunas da matriz os dados são armazenados na seguinte ordem: c_i , via x ou y, f , v_i , d_i , $tex_{Ap,w,ix}$, $tEy_{Ap,w,iy}$, $tsx_{Ap,w,ix}$, $tsy_{Ap,w,iy}$, a_{de} e j , onde a_{de} indica o tipo de movimento que o veículo deverá realizar, se for -1 terá que desacelerar, se 0 o movimento é uniforme e 1 para acelerar, e o j representa a identificação do veículo que se encontra imediatamente a frente.

A simulação é iniciada com $t = 0$, e uma lista é gerada com a identificação dos veículos na ordem do primeiro ao último a chegar no cruzamento. Em seguida eles são separados pela via e faixa a qual se encontram, então calcula se haverá colisão entre os veículos que passarão pelo cruzamento. Caso haja veículos em risco de conflito, eles são identificados e armazenados em uma lista de possíveis colisões. Se o tamanho desta lista for zero, então as velocidades dos veículos são mantidas e o tempo é incrementado em mais uma unidade. Se o tempo de simulação não terminou, as distâncias dos veículos em relação ao início do cruzamento são atualizadas; calcula se haverá colisão entre os veículos no cruzamento e repetem-se os demais processos descritos anteriormente. Caso a lista de possíveis colisões seja

maior que zero ($r > 0$), então algumas decisões são efetuadas para impedir a ocorrência de tais colisões, com base em um processo. Este processo consiste basicamente em fazer com que alguns veículos acelerem e/ou desacelerem ($a_{de} = 1$ ou -1). Portanto, um método que realize escolhas corretas de quais automóveis deverá acelerar, desacelerar ou manter sua velocidade constante, é a chave do sucesso para o controlador de tráfego contínuo. Os métodos aqui desenvolvidos para tais tomadas de decisão serão exibidos mais adiante, denominados como método 1 e método 2. Após a execução do método, analisa-se se alguma colisão já ocorreu no presente tempo t , se sim, o método precisa ser aperfeiçoado e a execução pode ser finalizada, caso contrário, as velocidades dos veículos são atualizadas e o tempo é incrementado em mais uma unidade, caso o tempo de simulação não tenha terminado, as distâncias dos veículos em relação ao início do cruzamento são atualizadas e repete-se os demais processos já descritos anteriormente. Na Figura 4 a seguir, apresenta-se em mais detalhes um fluxograma do algoritmo da simulação descrita.

FIGURA 4 – FLUXOGRAMA DA SIMULAÇÃO DE TRÁFEGO CONTÍNUO



FONTE: O AUTOR (2023).

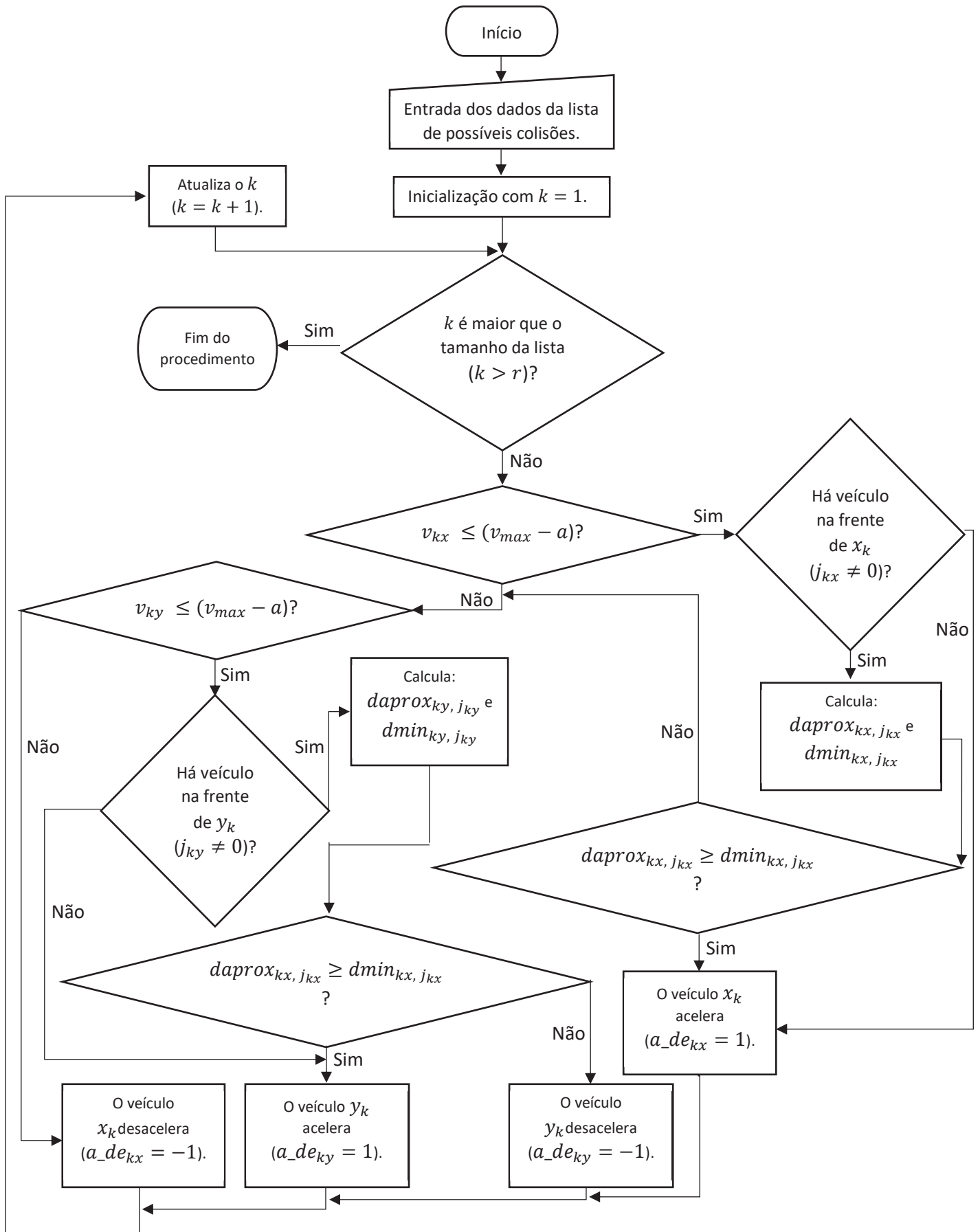
No fluxograma da Figura 4, a simulação é interrompida caso se registre uma colisão efetiva, mas para analisar o desempenho do método de tomada de decisão adotado, pode-se realizar o registro desta ocorrência e dar continuidade a simulação até que o critério de parada seja atendido. Na simulação é gerada uma lista de pares de colisões entre os veículos da via x e y, e nesta podem aparecer mais de uma vez o mesmo automóvel se colidindo, mas com outro veículo. Portanto, os pares de veículos não se repetem na mesma lista no instante t .

3.5.2 Método 1 para tomada de decisão

O método 1 de tomada de decisão para evitar colisões empregado na simulação do problema, consiste em enviar uma ordem de aceleração ou desaceleração ($a_{de} = 1$ ou -1) para cada par de veículos que irão se colidir (x_k e y_k , sendo $k = 1, \dots, r$), onde apenas um dos veículos do par terá sua aceleração modificada no instante t . Tem-se que r corresponde ao número de pares de colisões entre os veículos da via x e y, onde cada par de automóveis que irão se colidir é denotado pelo índice k . Portanto, o método 1 analisa inicialmente se o veículo x_k pode acelerar, avaliando se caso este acelere, se sua nova velocidade não excederá a máxima permitida pela via ou se poderá colidir-se com o veículo logo a sua frente. Caso uma destas situações possa ocorrer, então com base nos mesmos critérios é verificado se o veículo y_k pode acelerar, mas se este também não puder acelerar, então analisa-se qual dos veículos pode realizar a desaceleração, considerando que a nova velocidade não pode ser menor que a mínima permitida e que o veículo logo atrás tenha condições de acompanhá-lo sem colidir com o mesmo.

O fluxograma da Figura 5 apresentada a seguir, demonstra com mais detalhes como funciona o processo para evitar as colisões do método 1.

FIGURA 5 – FLUXOGRAMA DO MÉTODO 1



FONTE: O AUTOR (2023).

Na Figura 5 observa-se que o método 1 busca alterar a aceleração de apenas um dos veículos do par (x_k, y_k) no instante t .

3.5.3 Método 2 para tomada de decisão

O método 2 propõe uma tomada de decisão mais eficaz que a descrita anteriormente, pois busca explorar várias combinações possíveis de aceleração ou desaceleração para cada iteração da simulação, e avalia qual solução reduz o número de possíveis colisões (r). O método busca inicialmente encontrar um conjunto de soluções que satisfaça as Inequações (16), (17), (20) e (21), onde destas é escolhida a solução que melhor atenda as Inequações (18) e (19). Para melhor entendimento do algoritmo proposto, é exibido na Figura 6 o seu pseudocódigo:

FIGURA 6 – MÉTODO 2

Algoritmo: Minimiza colisões por iteração.

Entrada: Lista dos pares de veículos que vão se colidir (x_k, y_k) para $k = 1, \dots, r$;
 Lista de informações de tempo e espaço de cada x_k e y_k ($Inf x_k$ e $Inf y_k$);
 Taxa de aceleração (a) e desaceleração (de);
 Tamanho do passo em segundos (p);
 conte = 1;

Enquanto conte \leq ns **faça:**
Para $k = 1$ até r **faça:**
 Atribui um tipo de movimento no conjunto de possíveis soluções para x_k ($Px_k \leftarrow \{0\}$);
 Atribui um tipo de movimento no conjunto de possíveis soluções para y_k ($Py_k \leftarrow \{0\}$);
Se $v_{kx} \leq (v_{max,x} - a)$ e $(d_{prox_{kx}, j_{kx}} \geq d_{min_{kx}, j_{kx}})$ **então**
 $Px_k \leftarrow \{1\}$;
Se $v_{kx} > de \cdot p$ **então**
 $Px_k \leftarrow \{-1\}$;
Se $v_{ky} \leq (v_{max,y} - a)$ e $(d_{prox_{ky}, j_{ky}} \geq d_{min_{ky}, j_{ky}})$ **então**
 $Py_k \leftarrow \{1\}$;
Se $v_{ky} > de \cdot p$ **então**
 $Py_k \leftarrow \{-1\}$;
Se $\{1\} \in Px_k$ e $\{-1\} \in Py_k$ e $\{1\} \notin Py_k$ **então**
 $Sx_k \leftarrow$ sorteia um valor do conjunto Px_k ;
 $Sy_k \leftarrow$ sorteia um valor do conjunto $(Py_k - Sx_k)$;
Senão, se $\{1\} \in Px_k$ e $\{-1\} \in Py_k$ e $\{1\} \notin Py_k$ **então**
 $Sy_k \leftarrow$ sorteia uma solução do conjunto Py_k ;
 $Sx_k \leftarrow$ sorteia uma solução do conjunto $(Px_k - Sy_k)$;
Se $Px_k = Py_k$ **então**
 $b =$ sorteia um valor binário $\{0$ ou $1\}$;
Se $b = 1$ **então**
 $Sx_k \leftarrow$ sorteia uma solução do conjunto Px_k ;
 $Sy_k \leftarrow$ sorteia uma solução do conjunto $(Py_k - Sx_k)$;
Senão,
 $Sy_k \leftarrow$ sorteia uma solução do conjunto Py_k ;
 $Sx_k \leftarrow$ sorteia uma solução do conjunto $(Px_k - Sy_k)$;
Fim-para
 Para o conjunto solução gerado Sx e Sy calcula o número de possíveis colisões (r');
Se $r' < r$ **então**
 Armazena o conjunto solução gerado Sx e Sy ;

cont = cont + 1;

Fim-enquanto
Saida: melhor solução encontrada Sx e Sy .

FONTE: O autor (2023)

O método proposto na Figura 6 explora um número limitado ns de possíveis soluções a cada iteração e escolhe aquela que minimiza o número de possíveis colisões r .

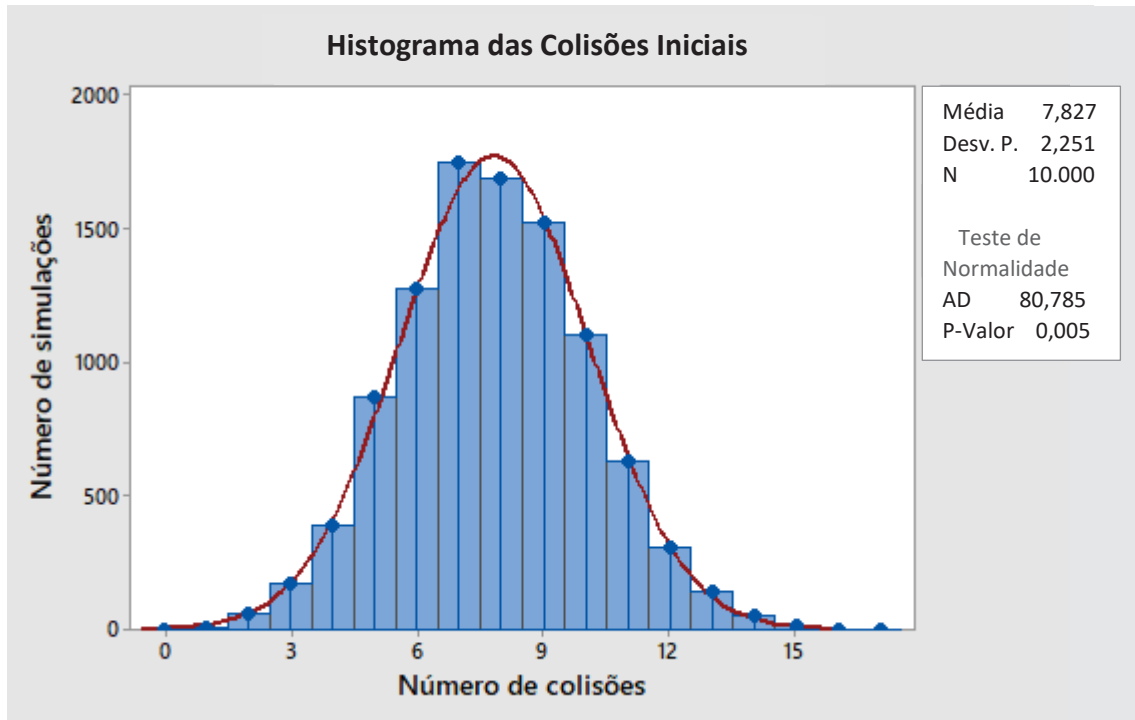
Porém, no primeiro método descrito isso não ocorre, pois para cada iteração ele busca apenas uma solução factível ou quase factível. Para ambos os métodos o objetivo principal é encontrar uma solução que atenda a todas as restrições do modelo apresentado, embora as Inequações (18) e (19) sejam as mais complexas.

4 RESULTADOS

Os dados considerados para as simulações realizadas foram: a distância de controle D , a qual foi alterada entre os testes; o número de faixas $F = 3$ para cada via; a largura das faixas l_y e l_x ; a velocidade inicial de cada automóvel como constante, sendo $v_i = 11 \text{ m/s}$; o número de veículos em cada simulação $n = 20, 30 \text{ e } 40$, onde inicialmente todos os veículos são gerados aleatoriamente e distribuídos, dentro de um intervalo de espaço igual a 200 m para cada via; a velocidade máxima permitida em ambas as vias são iguais, onde $v_{max,x} = v_{max,y} = 25 \text{ m/s}$; a taxa de aceleração a e taxa de desaceleração de dos veículos fixadas em 3 m/s^2 e 2 m/s^2 respectivamente, sendo estes valores dentro das taxas empregadas por um motorista humano, em uma condução considerada confortável aos ocupantes do veículo (BAE, MOON e SEO, 2019); a distância de segurança entre dois veículos consecutivos $ds = 1 \text{ m}$ e o tempo de simulação $t_{sim} = (D + 200)/2 \text{ s}$. A partir de tais dados o programa gera aleatoriamente: o comprimento c_i de cada veículo entre 3 e 7 metros, a via e a faixa f a qual será inserido e a sua distância inicial d_i , sendo esta, a distância que o veículo i se encontra do início do cruzamento, onde estas últimas três informações caracterizam a posição inicial do automóvel, caso esta posição já esteja sendo ocupada por outro veículo e não respeite uma distância mínima de 3 vezes a distância de segurança ds , então sorteia-se uma nova posição. O código da simulação e o método 1 apresenta-se no apêndice 1 e o método 2 localiza-se no apêndice 2.

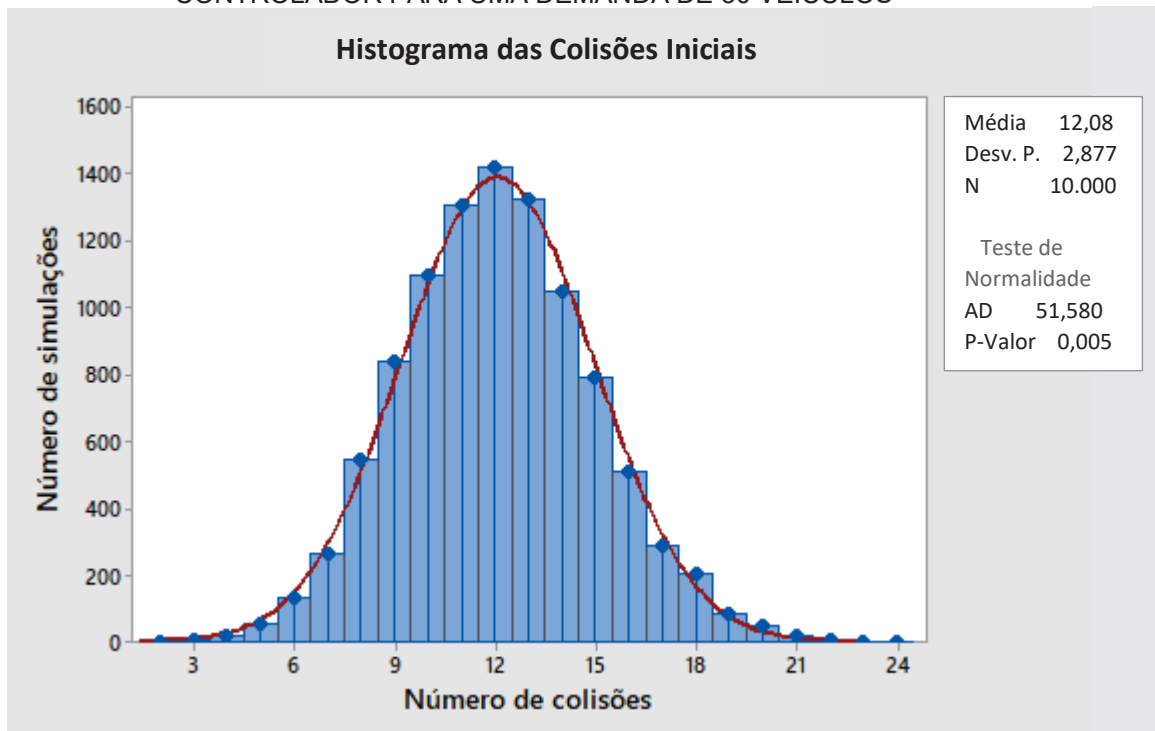
Para medir a eficiência do algoritmo em gerenciar a travessia dos veículos em um cruzamento de maneira que as colisões sejam evitadas, foi calculado no início de cada simulação se haveria colisões e quantas seriam caso os automóveis seguissem sem nenhuma interferência do controlador de interseção. Com isso, os histogramas presentes nas Figuras 7, 8 e 9 a seguir mostram tal situação para uma demanda de $n = 20, 30 \text{ e } 40$ veículos respectivamente, sendo realizado 10 mil simulações para cada caso, onde as distâncias de controle D e o método de tomada de decisão não influenciam o comportamento dos histogramas.

FIGURA 7 – HISTOGRAMAS DO NÚMERO DE COLISÕES SEM A INTERFERÊNCIA DO CONTROLADOR PARA UMA DEMANDA DE 20 VEÍCULOS



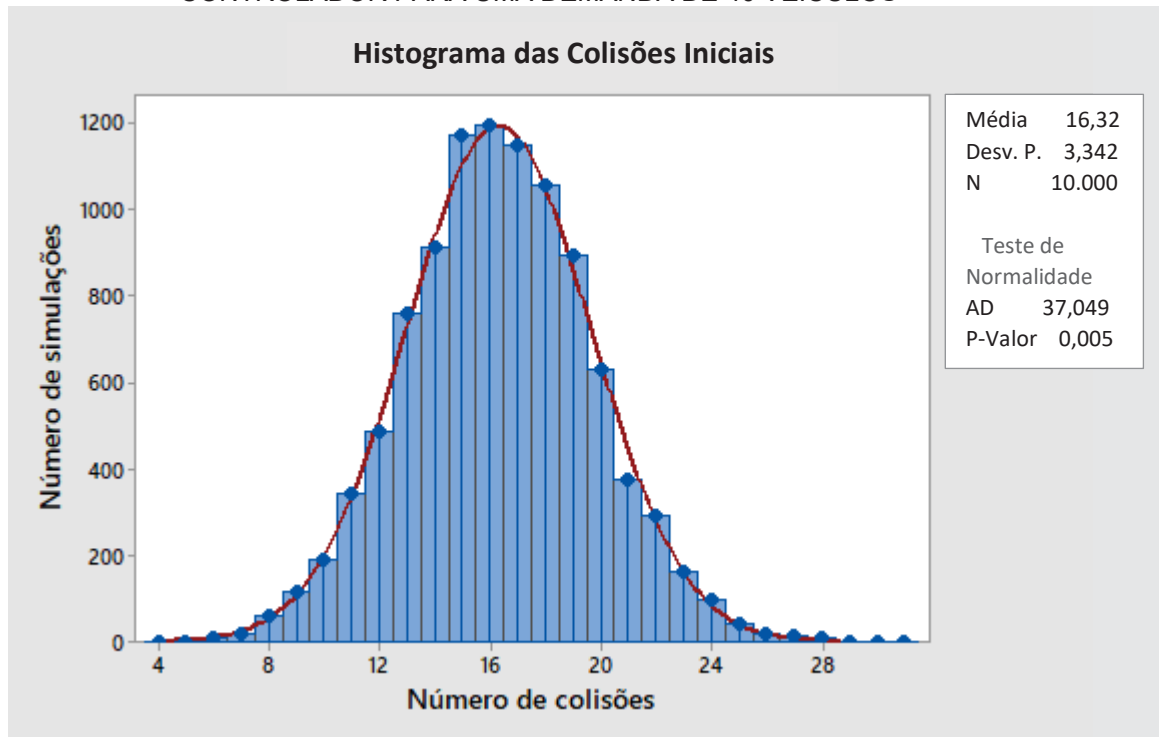
FONTE: O autor (2023).

FIGURA 8 – HISTOGRAMAS DO NÚMERO DE COLISÕES SEM A INTERFERÊNCIA DO CONTROLADOR PARA UMA DEMANDA DE 30 VEÍCULOS



FONTE: O autor (2023).

FIGURA 9 – HISTOGRAMAS DO NÚMERO DE COLISÕES SEM A INTERFERÊNCIA DO CONTROLADOR PARA UMA DEMANDA DE 40 VEÍCULOS



FONTE: O autor (2023).

Observando os histogramas das Figuras 7, 8 e 9, note-se que a média de colisões por simulação tem sido aproximadamente 8, 12 e 16 respectivamente, sendo um aumento proporcional ao número de veículos considerado em cada caso. E o mesmo acontece para o desvio padrão que tem aumentado de 2,251 para 2,877 e 3,342 conforme o aumento de veículos. Ambos os histogramas, apesar de possuírem um formato similar a uma distribuição Normal, não seguem este tipo de distribuição, pois o p-valor para todos os casos é inferior a 0,05. E o teste de Anderson-Darling (AD) acima possui valores muito elevados para considerar as distribuições apresentadas como uma Normal, pois devido a amostra ser muito grande, a hipótese zero para ser aceita, o desvio padrão precisaria ser muito baixo. Tal fato mostra a dificuldade de possuir uma previsibilidade das colisões, uma vez que as variáveis de tempo e espaço influenciam de forma direta, podendo todos os veículos colidirem ou nenhum.

4.1 RESULTADOS COM O MÉTODO 1

Os resultados exibidos a seguir referem-se a 10 mil simulações para cada distância de controle $D = \{100, 300, 500, 700, \dots, 2100, 2300\}$ e considerando uma demanda de 20 veículos ($n = 20$), buscando analisar a eficiência do algoritmo implementado, para evitar colisões quando a distância de controle do controlador de interseção é ampliada, e avaliar a média das velocidades dos veículos em cada simulação ao passarem pelo cruzamento. Portanto, na Tabela 1 a seguir, apresentam-se as médias das velocidades mínimas, máximas, amplitude, desvio padrão e a média das velocidades médias.

TABELA 1– VALORES MÍNIMOS, MÁXIMOS, AMPLITUDE, MÉDIA E DESVIO PADRÃO DA MÉDIA DAS VELOCIDADES PARA CADA $D = \{100, 300, 500, 700, \dots, 2100, 2300\}$ COM 20 VEÍCULOS

D [m]	v_{min} [m/s]	v_{max} [m/s]	Amplitude [m/s]	Média Veloc. [m/s]	Desv. P [m/s]
100	9.62	16.1	6.48	12.645	0.76353
300	6.85	15.95	9.10	11.867	0.73288
500	7.15	15.48	8.33	11.520	0.78417
700	6.66	14.05	7.39	11.202	0.85510
900	6.29	13.45	7.16	11.025	0.91381
1100	6.71	12.95	6.24	10.938	0.93476
1300	5.19	12.75	7.56	10.891	0.94390
1500	6.47	12.8	6.33	10.888	0.93606
1700	6.30	12.95	6.65	10.884	0.92080
1900	5.70	12.65	6.95	10.889	0.92072
2100	5.74	12.8	7.06	10.893	0.89714
2300	6.03	12.95	6.92	10.888	0.90899

FONTE: O autor (2023).

Analisando as médias mínimas e máximas das velocidades, notam-se maiores valores para $D = 100$ m e tende a diminuir na maioria dos casos quando D aumenta, mas o desvio padrão aumenta conforme D varia até 1300 m e a amplitude fica próximo de 7 m/s para a maioria dos casos. E quanto maior a distância de controle, a média das velocidades durante o cruzamento se aproxima de 10,88 m/s, ficando próximo da velocidade inicial de 11 m/s.

Na Tabela 2 a seguir, é apresentada a soma do número de colisões sem a interferência do controlador, o número de colisões efetivas com a atuação do

controlador, sendo este valor calculado pela soma de colisões ocorridas em cada uma das 10 mil simulações, e quantas destas simulações houve colisões para cada distância de controle D considerada.

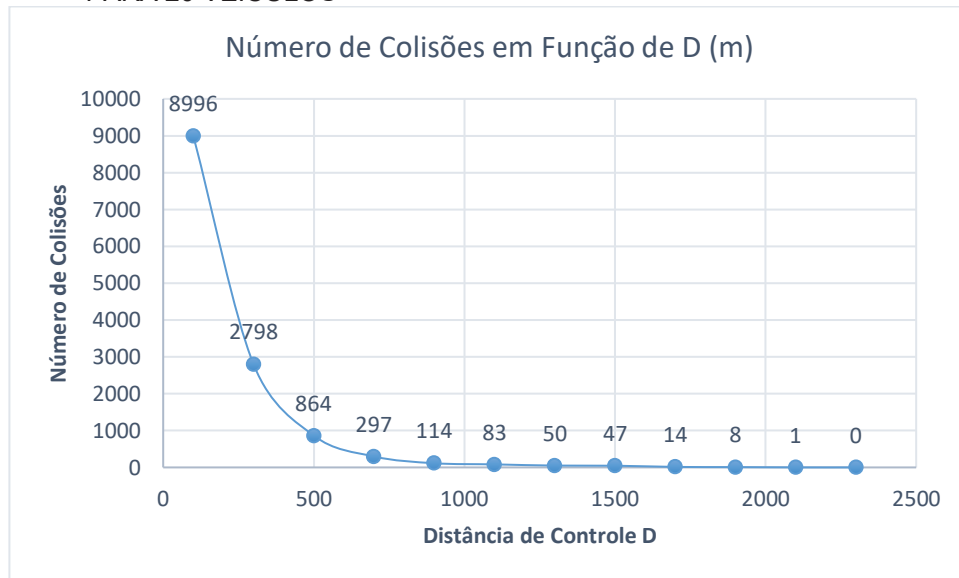
TABELA 2 – NÚMERO DE COLISÕES COM E SEM O USO DO CONTROLADOR E NÚMERO DE SIMULAÇÕES COM COLISÕES PARA CADA DISTÂNCIA D COM 20 VEÍCULOS

D [m]	Colisões iniciais	Colisões efetivas	simulações com ocorrência de colisões
100	78269	8996	4838
300	78517	2798	1457
500	78098	864	433
700	78595	297	151
900	78570	114	69
1100	78379	83	48
1300	78261	50	25
1500	77945	47	25
1700	77867	14	12
1900	78485	8	5
2100	77939	1	1
2300	78192	0	0

FONTE: O autor (2023).

Analisando a Tabela 2, tem-se que o número de colisões inicialmente sem a interferência do controlador é muito elevado e este valor não sofre grandes variações com relação as diferentes distâncias, fato este já previsto por não receber influência das diferentes distâncias de controle D . Na última coluna da tabela nota-se que o número de simulações em que ocorreram colisões diminuíram conforme D aumentou, e apenas para D igual a 1300 m e 1500 m não houve redução, no entanto, o número de veículos que colidiram reduziu de 50 para 47 ao observar a terceira coluna para os mesmos valores de D . A seguir, na Figura 10, tem-se um gráfico do número de colisões efetivas quando ocorre a atuação do controlador, conforme a terceira coluna da Tabela 2, em função de cada valor de D considerado.

FIGURA 10 – GRÁFICO DO NÚMERO DE COLISÕES COM A ATUAÇÃO DO CONTROLADOR PARA 20 VEÍCULOS



FONTE: O autor (2023).

Na Figura 10 acima, fica evidente a influência da distância de controle no número de colisões efetivas, onde esta diminui exponencialmente com o aumento da distância D .

O intervalo de tempo desde o primeiro veículo a iniciar o cruzamento e o último a terminar a travessia foi em média inferior a 19 segundos para $D = 100 \text{ m}$. Portanto, passam pelo cruzamento 1,1 veículos por segundo, mas esta frequência é reduzida com a elevação da distância de controle D .

A seguir apresentam-se os resultados de 10 mil simulações para cada distância de controle $D = \{100, 300, 500, 700, \dots, 3900, 4100\}$ e considerando uma demanda de 30 veículos ($n = 30$). Portanto, na Tabela 3 a seguir, apresentam-se as médias das velocidades mínimas, máximas, amplitude, desvio padrão e a média das velocidades médias.

TABELA 3 – VALORES MÍNIMOS, MÁXIMOS, AMPLITUDE, MÉDIA E DESVIO PADRÃO DA MÉDIA DAS VELOCIDADES PARA CADA $D = \{100, 300, 500, 700, \dots, 3900, 4100\}$ COM 30 VEÍCULOS

D [m]	V_min [m/s]	V_max [m/s]	Amplitude [m/s]	Média Veloc. [m/s]	Desv. P [m/s]
100	10.49	15.37	4.88	12.731	0.65171
300	8.20	15.73	7.53	11.923	0.75456
500	6.98	15.53	8.55	11.464	0.91905
700	6.60	15.3	8.70	10.880	0.99952
900	6.01	14.63	8.62	10.456	1.03777
1100	6.68	14.08	7.40	10.291	1.04183
1300	5.59	14.32	8.73	10.194	1.05576
1500	6.32	13	6.68	10.154	1.05615
1700	5.40	12.6	7.20	10.142	1.04589
1900	6.48	12.7	6.22	10.146	1.04661
2100	5.64	12.5	6.86	10.191	1.01859
2300	6.43	12.5	6.07	10.218	1.00771
2500	5.83	12.73	6.90	10.203	1.04614
2700	6.01	12.5	6.49	10.248	1.04680
2900	5.82	12.67	6.85	10.270	1.03494
3100	5.69	12.4	6.71	10.246	1.03054
3300	5.59	12.6	7.01	10.233	1.03900
3500	5.86	12.6	6.74	10.257	1.02811
3700	5.70	12.67	6.97	10.244	1.03506
3900	6.13	12.5	6.37	10.264	1.02811
4100	5.78	12.57	6.79	10.251	1.02647

FONTE: O autor (2023).

Observando a Tabela 3, nota-se que as médias mínimas e máximas das velocidades tende a diminuir na maioria dos casos quando D aumenta, mas o desvio padrão possui uma variação muito pequena quando D é maior que 900 m e a amplitude em média continua próximo de 7 m/s. E a média das velocidades durante o cruzamento se estabiliza com um valor próximo a 10,25 m/s para as distâncias de controle mais elevadas.

Na Tabela 4 a seguir, ainda considerando uma demanda de 30 veículos, é apresentada a soma do número de colisões sem a interferência do controlador, o número de colisões efetivas com a atuação do controlador e quantas das 10 mil simulações houve colisões para cada distância de controle D aplicada.

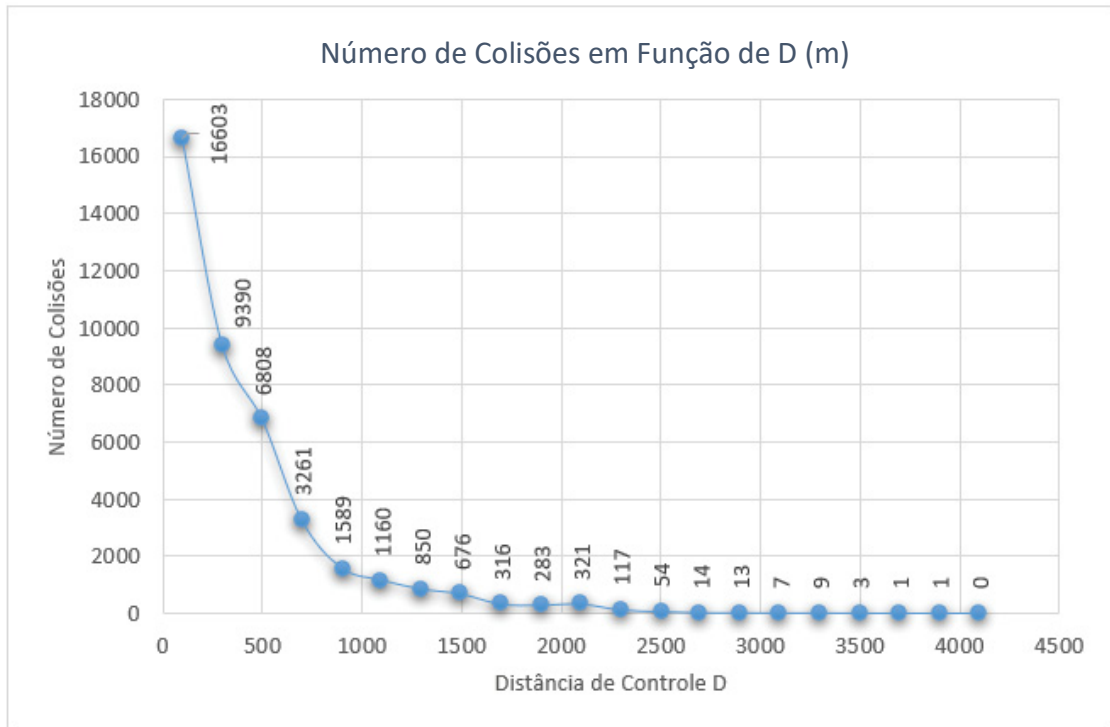
TABELA 4 – NÚMERO DE COLISÕES COM E SEM O USO DO CONTROLADOR E NÚMERO DE SIMULAÇÕES COM COLISÕES PARA CADA DISTÂNCIA D COM 30 VEÍCULOS

D [m]	Colisões iniciais	Colisões efetivas	simulações com ocorrência de colisões
100	120782	16603	6947
300	120688	9390	3463
500	120338	6808	2261
700	120853	3261	1077
900	120941	1589	542
1100	120356	1160	334
1300	120522	850	196
1500	120576	676	158
1700	120612	316	100
1900	120357	283	75
2100	120654	321	68
2300	121387	117	30
2500	120493	54	13
2700	120440	14	6
2900	120831	13	5
3100	120727	7	3
3300	120707	9	3
3500	120664	3	2
3700	120591	1	1
3900	120257	1	1
4100	120928	0	0

FONTE: O autor (2023).

Na Tabela 4, tem-se que o número de colisões inicialmente sem a interferência do controlador fica próximo de 120.500 para todas as distâncias de controle D . Nas últimas colunas da tabela nota-se que o número de simulações em que ocorreram colisões e o número de colisões totais reduziram até zero conforme D aumentou, mas a partir de D igual a 2700 m a diminuição tem sido baixa, de maneira que para as distâncias 3700 m e 3900 m não houve redução. A seguir, na Figura 11, tem-se um gráfico do número de colisões efetivas quando ocorre a atuação do controlador, conforme a terceira coluna da Tabela 4, em função de cada distância considerada, onde a demanda de veículos é igual a 30.

FIGURA 11 – GRÁFICO DO NÚMERO DE COLISÕES COM A ATUAÇÃO DO CONTROLADOR PARA 30 VEÍCULOS



FONTE: O autor (2023).

O comportamento do gráfico da Figura 11 se assemelha ao gráfico da Figura 10, onde ocorre uma redução exponencial do número de colisões efetivas com o aumento da distância D .

O intervalo de tempo desde o primeiro veículo a iniciar o cruzamento e o último a terminar a travessia tem sido em média de 24,4 segundos para $D = 100 \text{ m}$. Portanto, aumentou em 5,4s quando comparado com a demanda de 20 veículos, mas a frequência de 1,1 veículos por segundo no cruzamento se manteve.

A seguir, apresentam-se os resultados de 10 mil simulações para cada distância de controle $D = \{100, 300, 500, 700, \dots, 4500, 4700\}$ e considerando uma demanda de 40 veículos ($n = 40$). Portanto, na Tabela 5 a seguir, apresentam-se as médias das velocidades mínimas, máximas, amplitude, desvio padrão e a média das velocidades médias, onde para exibição de tais resultados foram realizados um total de 240 mil simulações.

TABELA 5 – VALORES MÍNIMOS, MÁXIMOS, AMPLITUDE, MÉDIA E DESVIO PADRÃO DA MÉDIA DAS VELOCIDADES PARA CADA $D = \{100, 300, 500, 700, \dots, 4500, 4700\}$ COM 40 VEÍCULOS

D [m]	V_min [m/s]	V_max [m/s]	Amplitude [m/s]	Média Veloc. [m/s]	Desv. P [m/s]
100	10.75	15.5	4.75	12.783	0.58136
300	8.28	15.59	7.31	11.983	0.74323
500	7.56	16.13	8.57	11.464	0.96137
700	6.34	16	9.66	10.652	1.12303
900	5.16	14.08	8.92	9.979	1.13176
1100	5.31	13.15	7.84	9.623	1.14117
1300	5.41	13.48	8.07	9.439	1.14410
1500	4.75	12.83	8.08	9.331	1.14802
1700	5.30	12.78	7.48	9.297	1.14283
1900	5.36	12.35	6.99	9.271	1.16744
2100	5.35	12.58	7.23	9.277	1.16267
2300	5.26	12.44	7.18	9.290	1.15488
2500	5.11	12.43	7.32	9.332	1.12791
2700	4.90	12.35	7.45	9.392	1.13424
2900	5.09	12.43	7.34	9.452	1.12140
3100	5.12	12.35	7.23	9.496	1.11454
3300	5.22	12.35	7.13	9.546	1.10989
3500	5.37	12.5	7.13	9.612	1.10493
3700	5.18	12.38	7.20	9.647	1.08818
3900	5.11	12.43	7.32	9.659	1.08538
4100	5.21	12.5	7.29	9.669	1.09615
4300	4.39	12.43	8.04	9.686	1.08580
4500	5.34	12.43	7.09	9.641	1.09179
4700	5.15	12.28	7.13	9.675	1.08907

FONTE: O autor (2023).

Analisando a Tabela 5 observa-se que tanto as médias mínimas, máximas, amplitude e as médias das velocidades médias diminuem com o aumento da distância D e tendem a atingir uma estabilidade para as distâncias maiores que 1700 metros. Porém, o desvio padrão se elevou até a distância de 2700 m e a partir desta, voltou a reduzir ficando em torno 1,1. E a média das velocidades médias durante o cruzamento se aproxima de 9,7 m/s, ficando abaixo da velocidade inicial de 11 m/s e também quando comparado com as outras análises para as demandas menores de veículos.

A seguir na Tabela 6, apresenta-se a soma do número de colisões sem a interferência do controlador, o número de colisões efetivas com a atuação do

controlador e quantas das 10 mil simulações houve colisões para cada distância de controle D considerada.

TABELA 6 – NÚMERO DE COLISÕES COM E SEM O USO DO CONTROLADOR E NÚMERO DE SIMULAÇÕES COM COLISÕES PARA CADA DISTÂNCIA D COM 40 VEÍCULOS

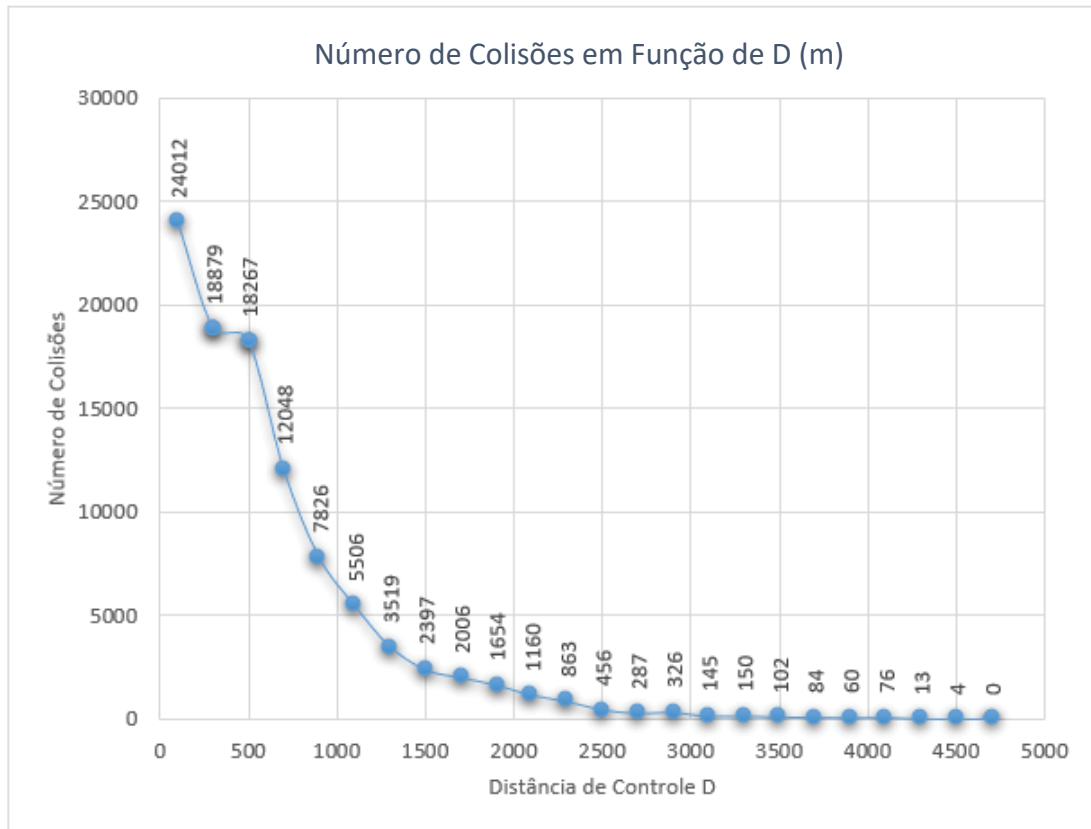
D [m]	Colisões iniciais	Colisões efetivas	simulações com ocorrência de colisões
100	163180	24012	8159
300	163060	18879	5219
500	163196	18267	4356
700	163797	12048	2840
900	163263	7826	1687
1100	163133	5506	1047
1300	163029	3519	645
1500	163558	2397	458
1700	162531	2006	356
1900	163026	1654	246
2100	163821	1160	187
2300	163099	863	130
2500	162656	456	78
2700	163131	287	58
2900	162723	326	52
3100	162834	145	26
3300	163300	150	23
3500	162997	102	22
3700	162572	84	13
3900	163896	60	7
4100	162815	76	12
4300	162736	13	3
4500	162729	4	1
4700	162834	0	0

FONTE: O autor (2023).

Analisando a Tabela 6, tem-se que o número de colisões inicialmente sem a interferência do controlador é muito elevado, como o esperado, independente das distâncias. O número colisões efetivas e o número de simulações com ocorrência de colisões diminuíram conforme D aumentou, mas para a distância D igual a 4100 m houve um aumento das colisões em comparação a distância de 3900 m redução, sendo este um comportamento isolado em relação aos demais casos. A Figura 12 a seguir exibe um gráfico do número de colisões efetivas quando ocorre a atuação do

controlador, conforme a terceira coluna da Tabela 6, em função de cada distância de controle para a demanda de 40 veículos.

FIGURA 12 – GRÁFICO DO NÚMERO DE COLISÕES COM A ATUAÇÃO DO CONTROLADOR PARA 40 VEÍCULOS



FONTE: O autor (2023).

Na Figura 12 apresentada, percebe-se que para as distâncias de 300 e 500 *m* o número de colisões efetivas reduziu a uma taxa menor que o comum para as demais distâncias consideradas. E nota-se que o comportamento do gráfico continua sendo próximo de uma função exponencial.

O intervalo de tempo desde o primeiro veículo a iniciar o cruzamento e o último a terminar a travessia foi em média de 36,5 segundos para $D = 100 \text{ m}$. E a frequência de veículos a passar pelo cruzamento tem permanecido em torno de 1,1 veículos por segundo, sendo a mesma para as outras demandas vistas anteriormente. No apêndice 3 encontram-se alguns histogramas complementares referentes ao método 1.

4.2 RESULTADOS COM O MÉTODO 2

Os resultados exibidos a seguir referem-se a 10 mil simulações para cada distância de controle $D = \{100, 200, 300\}$ e considerando uma demanda de 20 veículos ($n = 20$). As mesmas análises realizadas para o método 1 são aplicadas para o método 2, onde seus resultados são exibidos a seguir. Portanto, na Tabela 7 a seguir, apresenta-se as médias das velocidades mínimas, máximas, amplitude, desvio padrão e a média das velocidades médias.

TABELA 7 – VALORES MÍNIMOS, MÁXIMOS, AMPLITUDE, MÉDIA E DESVIO PADRÃO DA MÉDIA DAS VELOCIDADES PARA CADA $D = \{100, 200, 300\}$ COM 20 VEÍCULOS

D [m]	V_min [m/s]	V_max [m/s]	Amplitude [m/s]	Média Veloc. [m/s]	Desv. P [m/s]
100	5.78	12.25	6.47	10.468	0.81343
200	5.79	12.2	6.41	10.527	0.74984
300	4.95	12.05	7.10	10.535	0.70080

FONTE: O autor (2023).

Observando a Tabela 7, nota-se uma redução das velocidades mínimas, máximas e desvio padrão, conforme a distância aumentou, mas a velocidade média sofreu uma pequena elevação. Fato este diferente do apresentado na Tabela 1 referente ao método 1 com a mesma demanda de veículos, onde houve uma diminuição da velocidade média com o aumento da distância. Porém, aqui a velocidade média em geral tem sido inferior ao apresentado na Tabela 1, ficando próximo a 10,5 m/s.

Na Tabela 8 a seguir, apresenta-se a soma do número de colisões sem a interferência do controlador, o número de colisões efetivas com a atuação do controlador e quantas das 10 mil simulações houve colisões para cada uma das 3 distâncias de controle considerada.

TABELA 8 – NÚMERO DE COLISÕES COM E SEM O USO DO CONTROLADOR E NÚMERO DE SIMULAÇÕES COM COLISÕES PARA CADA DISTÂNCIA D COM 20 VEÍCULOS USANDO O MÉTODO 2

D [m]	Colisões iniciais	Colisões efetivas	simulações com ocorrência de colisões
100	77910	6694	1790
200	78166	78	32
300	78012	0	0

FONTE: O autor (2023).

Analisando a Tabela 8, fica evidente uma rápida redução do número de colisões efetivas e entre as simulações, com o incremento de apenas 100 metros na distância de controle. Portanto, para $D = 100, 200$ e 300 , verifica-se que a média de colisões por simulação tem reduzido de 3,74 para 2,43 e zero respectivamente.

As próximas duas tabelas a seguir, representam os resultados para 10 mil simulações para cada distância de controle $D = \{100, 200, 300, 400\}$, quando a demanda é de 30 veículos ($n = 30$). Portanto, na Tabela 9 a seguir, apresentam-se as médias das velocidades mínimas, máximas, amplitude, desvio padrão e a média das velocidades médias.

TABELA 9 – VALORES MÍNIMOS, MÁXIMOS, AMPLITUDE, MÉDIA E DESVIO PADRÃO DA MÉDIA DAS VELOCIDADES PARA CADA $D = \{100, 200, 300, 400\}$ COM 30 VEÍCULOS

D [m]	V_min [m/s]	V_max [m/s]	Amplitude [m/s]	Média Veloc. [m/s]	Desv. P [m/s]
100	6.43	12.13	5.70	10.352	0.80263
200	5.03	11.87	6.84	10.153	0.82535
300	6.11	11.8	5.69	10.168	0.76498
400	4.58	11.83	7.25	10.197	0.72231

FONTE: O autor (2023).

As velocidades e a amplitude da Tabela 9 não seguem um padrão crescente ou decrescente dos valores conforme a distância aumenta, mas a média das velocidades tende a ficar próximo a 10,2 m/s e o desvio padrão diminuiu para a distância de 400 m.

A soma do número de colisões sem a interferência do controlador, o número de colisões efetivas com a atuação do controlador e quantas das 10 mil simulações houve colisões para cada uma das 4 distâncias de controle considerada, são apresentadas na Tabela 10 a seguir:

TABELA 10 – NÚMERO DE COLISÕES COM E SEM O USO DO CONTROLADOR E NÚMERO DE SIMULAÇÕES COM COLISÕES PARA CADA DISTÂNCIA D COM 30 VEÍCULOS USANDO O MÉTODO 2

D [m]	Colisões iniciais	Colisões efetivas	simulações com ocorrência de colisões
100	120482	20346	3018
200	120784	1550	412
300	120697	63	20
400	120212	0	0

FONTE: O autor (2023).

Na Tabela 10, nota-se um comportamento similar ao apresentado na Tabela 8, porém o número colisões efetivas neste caso são 3 vezes maiores para a distância de controle de 100 m. E as colisões foram zeradas quando a distância passou a ser de 400 m.

As tabelas a seguir representam os resultados para 10 mil simulações para cada distância de controle $D = \{100, 200, 300, 400, 500\}$, quando a demanda de veículos é igual a 40 ($n = 40$). Portanto, na Tabela 11 a seguir, apresentam-se as médias das velocidades mínimas, máximas, amplitude, desvio padrão e a média das velocidades médias.

TABELA 11 – VALORES MÍNIMOS, MÁXIMOS, AMPLITUDE, MÉDIA E DESVIO PADRÃO DA MÉDIA DAS VELOCIDADES PARA CADA $D = \{100, 200, 300, 400, 500\}$ COM 40 VEÍCULOS

D [m]	V_min [m/s]	V_max [m/s]	Amplitude [m/s]	Média Veloc. [m/s]	Desv. P [m/s]
100	5.98	11.78	5.80	10.221	0.78366
200	5.74	11.7	5.96	9.830	0.91435
300	5.22	11.58	6.36	9.769	0.84915
400	5.68	11.5	5.82	9.842	0.78086
500	5.88	11.5	5.62	9.915	0.71581

FONTE: O autor (2023).

Os valores das velocidades máximas da Tabela 11 exibiram uma pequena redução quando a distância aumentou até 400 m e o desvio padrão baixou para D maior que 200 m. E a média as velocidades ficaram em torno de 9,9 m/s.

Na Tabela 12 a seguir, tem-se a soma do número de colisões sem a interferência do controlador, o número de colisões efetivas com a atuação do

controlador e quantas das 10 mil simulações houve colisões para cada uma das 5 distâncias de controle quando a demanda de veículos é igual a 40.

TABELA 12 – NÚMERO DE COLISÕES COM E SEM O USO DO CONTROLADOR E NÚMERO DE SIMULAÇÕES COM COLISÕES PARA CADA DISTÂNCIA D COM 40 VEÍCULOS USANDO O MÉTODO 2

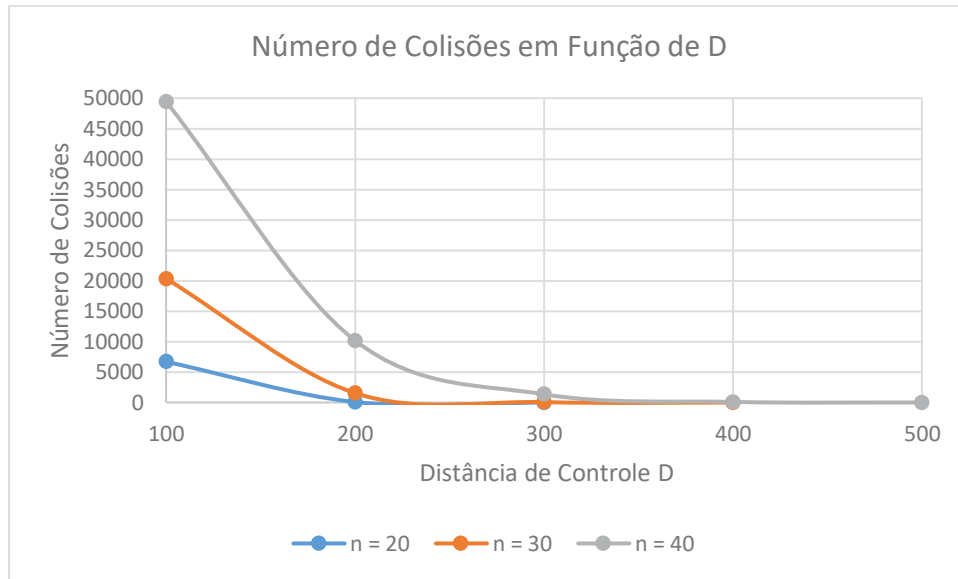
D [m]	Colisões iniciais	Colisões efetivas	simulações com ocorrência de colisões
100	162686	49446	4717
200	162896	10187	1397
300	162976	1349	258
400	163277	97	13
500	163122	0	0

FONTE: O autor (2023).

O número de colisões iniciais apresentado na Tabela 12 tem ficado em torno de 163 mil, seguindo a mesma ordem de grandeza apresentada para esta coluna na Tabela 6, onde a demanda de veículos também é igual a 40. Para as informações do número de colisões efetivas e simulações com ocorrência de colisões, nota-se um decréscimo quando a distância de controle aumenta. Portanto, as colisões deixam de aparecer quando a distância de controle é igual a 500 m.

A Figura 13 a seguir exibe um gráfico do número de colisões efetivas para cada demanda de veículos considerada no método 2, conforme os valores apresentados na terceira coluna das Tabelas 8, 10 e 12.

FIGURA 13 – GRÁFICO DO NÚMERO DE COLISÕES COM A ATUAÇÃO DO CONTROLADOR PARA AS DEMANDAS DE 20, 30 E 40 VEÍCULOS USANDO O MÉTODO 2



FONTE: O autor (2023).

Analisando o gráfico da Figura 13, nota-se que, quanto maior o número de veículos mais colisões ocorrem. E a distância de controle para zerar as colisões são de 300, 400 e 500 m para as demandas de 20, 30 e 40 veículos respectivamente. Portanto, para cada 10 veículos a mais no sistema, foram necessários 100 metros a mais no raio de atuação do controlador para resolver completamente os problemas de colisão.

5 CONCLUSÃO E RECOMENDAÇÕES

Nesta tese foi apresentado um novo modelo computacional, um simulador de tráfego contínuo e 2 métodos para buscar uma solução para o modelo desenvolvido, o qual possibilita gerenciar a travessia de veículos autônomos em um cruzamento simplificado de maneira que os veículos não necessitam ficar retidos antes da faixa de intersecção.

Considerando os resultados gerados, conclui-se que o método 1 proposto apresentou bons resultados, sem apresentar colisões na maioria das simulações, ou seja, foi capaz de gerenciar os veículos autônomos no cruzamento. Porém, o algoritmo empregado não foi eficiente para evitar 100% das colisões independentemente da distância de controle D considerada. A completa extinção das colisões só foi possível para as distâncias mais elevadas, sendo estas de 2300, 4100 e 4700 m quando as demandas foram respectivamente de 20, 30 e 40 veículos. Observando o método apresentado na Figura 5, nota-se que este não garante o fim de todas as colisões em uma única iteração, sendo assim, na próxima iteração pode aparecer um mesmo par de veículos na nova lista de possíveis colisões, sendo este o motivo do número de colisões chegar a zero apenas para elevados valores de D , pois quanto maior é a distância de controle, mais tempo o algoritmo possui para ajustar as velocidades dos veículos.

Quando analisados os resultados no método 2, foi notado que as distâncias de controle de até 500 m foram suficientes para zerar todas as colisões, mesmo para a maior demanda testada de 40 veículos. Portanto, o método 2 desempenhou resultados muito mais satisfatórios que o método 1, pois, ao invés de buscar apenas uma única possível solução a cada iteração, este analisa uma gama de possibilidades a cada passo da simulação. Isso ocorre pois, quanto maior a distância, mais tempo o método possui para ajustar as velocidades dos veículos. Em ambos os métodos o número de colisões seguiu um padrão semelhante a uma função exponencial inversa, mostrando empiricamente que os mesmos produzem uma convergência do número de colisões para zero, conforme D aumenta. Entretanto, o segundo método mostrou uma taxa de convergência média de 9,1 vezes maior que primeiro método, mas o tempo de execução se elevou por explorar um espaço de busca maior, porém o

aumento do tempo não foi elevado o suficiente para impedir a solução de ser obtida em tempo real, ou seja, em menos de 1s por iteração.

Os métodos desenvolvidos, embora não possam lidar com maiores demandas de veículos para curtas distâncias de controle, foram capazes de produzir respostas de espaço-tempo para cada veículo realizar a travessia com segurança, onde esta resposta foi gerada em tempo real, enquanto os veículos se deslocavam em direção ao cruzamento. Portanto, os métodos se mostram muito ágeis em exibir uma solução ao passo que a cada segundo o cenário se alterava. Ambos os métodos foram capazes de produzir uma possível solução em menos de 1 segundo de execução do código, onde este feito se mostrou conciso para todas as 690 mil simulações realizadas. Tal feito pode ser desempenhado por um computador convencional, mostrando que é não necessário um supercomputador para o controlador semafórico realizar o gerenciamento dos veículos autônomos. Porém, a latência da comunicação do controlador com o veículo deverá ser baixa.

O modelo proposto poderá lidar com interseções, onde o fluxo seja no máximo de 2340 veículos por hora, para uma intersecção com as mesmas características da Figura 2, em que os comprimentos das vias sejam de pelo menos 300 m cada. Lembrando que tal aplicação só poderá ser usada quando todos os veículos da malha viária passarem a ser autônomos.

Para futuras pesquisas que visem eliminar totalmente as colisões, mesmo para a menor distância de controle considerada, propõe-se o melhoramento do método de tomadas de decisão, buscando como sugestão, realizar uma redução do incremento do tempo de 1 segundo para uma fração deste tempo nas iterações, assim o modelo discreto poderá se assemelhar com um modelo contínuo, conforme o incremento do tempo tender a zero. Com isso, a velocidade dos veículos se estabilizará mais rapidamente, reduzindo o desconforto dos ocupantes dos automóveis (JIN, HAO e YANG, 2021). Porém, o incremento do tempo deverá ser condizente com o tempo de comunicação do controlador com o veículo e também o tempo de resposta do automóvel, assim sendo, conforme a tecnologia avançar nesse sentido, menor poderá ser o incremento do tempo. Outra proposta é o desenvolvimento de um método de tomada de decisão baseado em redes neurais, onde o método aqui proposto possa ser usado para treinar a rede. Portanto, abre-se espaço para que novas pesquisas proponham métodos de otimização mais eficientes

para resolver o modelo apresentado, pois o método desenvolvido tem como ponto forte o fato de resolver o modelo em tempo real, mas apresenta como ponto negativo a necessidade de uma longa distância de controle para garantir sua confiabilidade. E para futuros trabalhos podem-se incluir novas restrições ao modelo, para considerar curvas a direita e a esquerda, de maneira que a conversão se assemelhe com o comportamento de um condutor humano ao realizá-la. Portanto, tornando o modelo mais próximos a realidade.

REFERÊNCIAS

- ABBAS, M. A.; MILMAN, R.; EKLUND, J. M. Obstacle Avoidance in Real Time With Nonlinear Model Predictive Control of Autonomous Vehicles. **Canadian Journal of Electrical and Computer Engineering**, v. 40, n. 1, p. 12-22, 2017. ISSN 0840-8688.
- AHMED, N.; RONY, R. J. Understanding self-reported stress among drivers and designing stress monitor using heart rate variability. **Quality and User Experience - Springer**, v. 6, n. 1, p. 1-21, 2021.
- ALOULFI, N. **Autonomous vehicle scheduling at intersections based on production line technique**. Dissertação, California State University Dominguez Hills. Carson, p. 140. 2018.
- AU, T.-C.; ZHANG, S.; STONE, P. Autonomous Intersection Management for Semi-Autonomous Vehicles. In: TEODOROVIC, D. **Handbook of Transportation**. [S.l.]: Routledge, 2016. p. 88 - 104.
- BAE, I.; MOON, J.; SEO, J. Toward a Comfortable Driving Experience for a Self-Driving Shuttle Bus. **Electronics**, v. 8, n. 9, p. 943, Agosto 2019.
- BIMBRAW, K. **Autonomous cars**: Past, present and future a review of the developments in the last century, the present scenario and the expected future of autonomous vehicle technology. Informatics in Control, Automation and Robotics (ICINCO). Colmar, França: IEEE. 2015. p. 191-198.
- BOAS, E. R. V. **Mapeamento e Localização Simultânea de Ambientes Dinâmicos Aplicados na Navegação de Veículo Autônomo Inteligente**. Dissertação, Universidade Federal de Itajubá. Itajubá, p. 80. 2011.
- BRASIL. CÓDIGO DE TRÂNSITO BRASILEIRO. **jusbrasil**, 1997. Disponível em: <<https://www.jusbrasil.com.br/topicos/10632298/paragrafo-1-artigo-1-da-lei-n-9503-de-23-de-setembro-de-1997>>. Acesso em: 08 jul. 2020.
- DIAS, J. E. A. **Modelagem longitudinal e controle de velocidade de um carro autônomo**. Dissertação, Universidade Federal de Minas Gerais. Belo Horizonte, p. 121. 2013. (779).
- DRESNER, K.; STONE, P. A Multiagent Approach to Autonomous Intersection Management. **Journal of Artificial Intelligence Research (JAIR)**, Março 2008. 591 - 656.
- ESTADÃO. Summit Mobilidade Urbana 2020. **Estadão**, 2020. Disponível em: <<https://summitmobilidade.estadao.com.br/carros-autonomos/primeiro-projeto-de-carro-autonomo-data-de-1920/>>. Acesso em: 23 set. 2020.
- FREITAS, E. J. D. R. **Controle longitudinal de um veículo autônomo**. Trabalho de Conclusão de Curso, Universidade Federal de Minas Gerais. Belo Horizonte, p. 1-9. 2010.

HEYMANN, M.; DEGANI, A. **Autonomous vehicle interactions with other road users: CONFLICTS AND RESOLUTIONS**. Proceedings of the Tenth International Driving Symposium on Human Factors in Driver Assessment. Iowa, IA: Public Policy Center. 2019. p. 419-425.

JI, K.; ORSAG, M.; HAN, K. Lane-Merging Strategy for a Self-Driving Car in Dense Traffic Using the Stackelberg Game Approach. **Electronics**, v. 10, n. 8, Abril 2021. ISSN 10.3390.

JIN, H.; HAO, H.; YANG, X. Continuous and Discrete Analysis of Local Stability for Car-Following Model With Speed-Based Desired-Headway. **IEEE Transactions on Intelligent Transportation Systems**, v. 23, n. 6, p. 5374-5386, 2021.

LAZARUS AND FREE PASCAL TEAM. Lazarus IDE. **Lazarus**, 1993. Disponível em: <<https://www.lazarus-ide.org/>>. Acesso em: Agosto 2019.

LIMA, D. A. D. **Navegação Segura de um Carro Autônomo Utilizando Campos Vetoriais e o Método da Janela Dinâmica**. Dissertação, Universidade Federal de Minas Gerais. Belo Horizonte, p. 124. 2010.

LIU, M. et al. A single-layer approach for joint optimization of traffic signals and cooperative vehicle trajectories at isolated intersections. **Transportation research part C: emerging technologies - Elsevier**, v. 134, p. 103459, 2022.

MARKKULA, G.; MADIGAN, R.; NATHANAEL, D.; PORTOULI, E.; LEE, Y. M.; DIETRICH, A.; MERAT, N. Defining interactions: a conceptual framework for understanding interactive behaviour in human and automated road traffic. **Theoretical Issues in Ergonomics Science**, v. 21, n. 6, p. 728-752, Março 2020. ISSN 1464-536X.

PISSARDINI, R. D. S. **Veículos Autônomos de Transporte Terrestre: Proposta de Arquitetura de Tomada de Decisão para Navegação Autônoma**. Dissertação, Escola Politécnica da Universidade de São Paulo. São Paulo, p. 182. 2014.

RASOULI, A.; TSOTSOS, J. K. Autonomous Vehicles That Interact With Pedestrians: A Survey of Theory and Practice. **IEEE Transactions on intelligent Transportation Systems**, v. 21, n. 3, p. 900-918, Março 2020.

RELEASES, I. N. IEEE news releases. **IEEE**, 2012. Disponível em: <http://www.ieee.org/about/news/2012/5september_2_2012.html>. Acesso em: 14 Julho 2021.

SHADRIN, S. S.; IVANOVA, A. A. Analytical review of standard Sae J3016 taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles with latest updates. **Avtomobil'. Doroga. Infrastruktura.**, v. 21, n. 3, p. 10, 2019. ISSN 2409-7217.

SOUSA, L. A. P. D. **Taxas de desaceleração e tempos de percepção e reação dos motoristas em interseções semaforizadas**. Dissertação - UFRJ. Rio de Janeiro, p. 90. 2011.

TACHET, R.; SANTI, P.; SOBOLEVSKY, S.; REYES-CASTRO, L. I.; FRAZZOLI, E.; HELBING, D.; RATTI, C.. Revisiting Street Intersections Using Slot-Based Systems. **PLOS ONE**, p. 1-13, Março 2016. Disponível em: <<https://doi.org/10.1371/journal.pone.0149607>>.

TIAN, R. et al. Game-theoretic Modeling of Traffic in Unsignalized Intersection Network for Autonomous Vehicle Control Verification and Validation. **Transações de sistemas de transporte inteligentes**, n. IEEE, novembro 2020. ISSN 1558-0016.

VANMIDDLESWORTH, M.; DRESNER, K.; STONE, P. **Replacing the Stop Sign: Unmanaged Intersection Control for Autonomous Vehicles**. Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS). Estoril, Portugal: Padgham, Parkes, Müller and Parsons (eds.). 2008. p. 1413-1416.

WEI, D. C. M. **Método de desvio de obstáculos aplicado em veículo autônomo**. Dissertação, Escola Politécnica da Universidade de São Paulo. São Paulo, p. 100. 2015.

ZHAO, J.; KNOOP, V. L.; WANG, M. Two-dimensional vehicular movement modelling at intersections based on optimal control. **Transportation Research Part B: Methodological**, v. 138, p. 1-22, Abril 2020.

ZHAO, J.; KNOOP, V. L.; WANG, M. Microscopic Traffic Modeling Inside Intersections: Interactions Between Drivers. **Transportation Science**, v. 57, p. 135-155, Julho 2022.

APÊNDICE 1 – CÓDIGO DO PROGRAMA ELABORADO EM PASCAL COM O MÉTODO 1

```

program simulacao_cruzamento_autonomo;
  {Tese de Doutorado
  Orientador: Eloy Kaviski
  Coorientadora: Liliana M. Gramani
  Data:08/08/2019 início do programa
  Simula o controle inteligente para veículos autonomos em um cruzamento de
  sentido único com 3 faixas(pistas) para cada direção se sentido único.}

uses crt;

{----- Declaração das variaveis -----}
var
  i,j,k,w,r, {controle para iterações}
  comp,
  fim, {tempo total de simulação}
  cont, {contador}
  cont_coli, {Contador de colisões efetivas}
  ite, {controle de Intersção entre as faixas}
  posi,
  id,idx,idy, {auxiliar para o id}
  frente, {id do veiculo que esta a frente.}
  a, {ID do veiculo}
  f, {número de faixas da via x e y}
  n, {número de veiculos inicialmente}
  m, {numero de veiculos que cruzou}
  t, {tempo em segundo da simulação}
  ncruzou, {número de veiculos que passou o cruzamento}
  isim, {Conta o número de simulações}
  n_sim {Número de simulações}
  :integer;
  passou:string; {Usada para analisar se o veiculo "passou" pelo cruzamento
  ou não.}
  cx,{comprimento da via x}
  cy:integer;{comprimento da via y}
  lx,nx:array[1..3] of integer;{largura e numero de cada faixa da via x}
  ly,ny:array[1..3] of integer;{largura e numero de cada faixa da via y}
  n_cruzou: array[1..300] of integer;{informa o numero de veiculos que já
  atravessou o cruzamento em cada instante}
  analise: array[1..10000,1..8] of real; { Matriz com resultados
  estatisticos de cada simulação }
  matriz_posi: array[1..1000,1..1000] of integer; { Matriz com as posições
  iniciais dos veiculos, que é gerada aleatoriamente}
  ordenado:array[1..200] of integer; {ordena os veiculos começando do mais
  proximo do cruzamento}
  colisao:array[1..200,1..2] of integer; {mostra quais veiculos irão se
  colidir}
  orx,ory,px, py:array[1..3,1..200] of integer;{ordem dos veiculos em cada
  faixa da via x e y}
  distancia:array[1..200] of real; {valor da soma da distancia do veiculo ao
  cruzamento com o comprimento do veiculo}
  inf, inicial:array[1..200,1..15] of real; {cada linha contem as
  informações de um veiculo da malha.}
  dc, {distancia do veiculo ao cruzamento + comprimento do veiculo}
  dist, {distancia do veiculo em relação ao veiculo da frente}

```

```

dapro,{distancia de aproximação}
dapro_const, {distancia de aproximação com a velocidade constante}
dmin, {distancia minima de segurança}
dmin_ini, {distancia minima inicial de segurança}
vmax, {velocidade maxima permitida}
vf, {velocidade do veiculo da frente}
vt, {velocidade do veiculo de trás}
menor, {usado para a ordenação da menor distancia}
ac,{taxa de aceleração}
de,{taxa de desaceleração}
desc,{taxa de desaceleração axiliar}
c, {comprimento do veiculo.}
xy,{0 veiculo estará na via x = 1 ou y = 2}
v, {velocidade}
dmc, {distancia minima inicial até o cruzamento}
d, {distância do choque frontal até o inicio do cruzamento}
daux,{Variavel auxiliar para a distancia}
te1,{tempo de entrada na 1ª faixa}
te2,{tempo de entrada na 2ª faixa}
te3,{tempo de entrada na 3ª faixa}
ts1,{tempo de saida na 1ª faixa}
ts2,{tempo de saida na 2ª faixa}
ts3,{tempo de saida na 3ª faixa}
ac_de,{controla se o MUV: Se acelerar = 1, se desacelerar = -1 ou MU: Se
velocidade constante = 0}
ta, {tempo de aceleração}
s:real; {controla o sentido veiculo 1 ou -1}
densidade, {razao da soma abaixo pela soma do comprimento das pistas}
soma, {soma dos comprimentos de todos os veiculos nas ruas}
min_t, {tempo que o primeiro cruzou}
min_v, {veiculo mais lento ao cruzar}
max_t, {tempo que o ultimo cruzou}
max_v: real; {veiculo mais rápido ao cruzar}
ira_coli:integer;
ordena, controle:boolean;
arq_s, arq_coli, arq_analise:text; {imprime informações em arquivos txt}

procedure ordena_separa();
{Inicia a ordenação dos veiculos pela menor distancia do final do mesmo
até
o inicio do cruzamento, ou seja, o menor valor de d+c vem primeiro!}
begin
for a:=1 to n do
    distancia[a]:= inf[a,1] + inf[a,5];

i:=1;
menor:=50000;
id:=1;
while (i<= n) do
    begin
    for a:=1 to n do
        if (distancia[a] < menor) then
            begin
                id:=a;
                menor:=distancia[a];
            end;
        menor:=50000;
        distancia[id]:=50000;
        ordenado[i]:=id;
        i:=i+1;
    end;
end;

```

```

{ Realiza-se a indentificação de cada veiculo em cada via e faixa que se
encontra na ordem de menor dist. encontrada na ordenação.}
for i:= 1 to f do
begin
  nx[i]:=0;
  ny[i]:=0;
end;

for id:=1 to n do
begin
  a:=ordenado[id];
  if (inf[a,2] = 1) then
begin
  i:=round(inf[a,3]);
  nx[i]:=nx[i]+1;
  orx[i,nx[i]]:=a;
  if (nx[i] = 1) then
    inf[a,15]:=0
  else
    inf[a,15]:= orx[i,nx[i]-1];
end
else
begin
  i:=round(inf[a,3]);
  ny[i]:=ny[i]+1;
  ory[i,ny[i]]:=a;
  if (ny[i] = 1) then
    inf[a,15]:=0
  else
    inf[a,15]:= ory[i,ny[i]-1];
end;
end;
end;

end;
{----- Fim -----}

{ Calcula a dist. que o veiculo esta de quem esta a frente & dist. min.
segura}
procedure distancia_do_proximo_frente();
begin
  {S7 - S1 = V7.ta7 + a.t^2/2 + x - V1.ta7 - a.ta1^2/2 - C1}
  {dapro:= inf[a,4] + ac/2 + (inf[a,5]- inf[frente,5]) - inf[frente,1] -
inf[frente,4] - ac*inf[frente,12]/2;}

  {sinais trocados: S7 - S1 = - V7.ta7 - a.t^2/2 + x + V1.ta7 + a.ta1^2/2 -
C1 (alterado em 31/03/2020)}
  dapro:= - inf[a,4] - ac/2 + (inf[a,5]- inf[frente,5]) - inf[frente,1] +
inf[frente,4] + ac*inf[frente,12]/2;
  vt:= inf[a,4] + ac;
  vf:= inf[frente,4] + ac*inf[frente,12];
  dmin:= dmin_ini + (vf*vf - vt*vt)/(-2*de);
end;

{----- Fim do calculo -----}

{----- Procedimento de atualização de tempos para o cruzamendo -----}
procedure instantes_de_cruzamento();
begin
  c:=inf[a,1];
  v:=inf[a,4];

```

```

    daux:=inf[a,5];
    if (v > 0.0) then
      Begin

        if (inf[a,2]=1) then
          begin
            for i:=1 to 3 do
              begin
                inf[a,i+5]:= real(t) + daux/v; {tempo de entrada 6,7,8}{somei
"t" em 21/04/2021}
                daux:= daux + ly[i];
                dc:=daux+c;
                inf[a,i+8]:= real(t) + dc/v; {tempo de saida 9,10,11}{somei "t"
em 21/04/2021}
              end;
            end

          else
            begin
              for i:=1 to 3 do
                begin
                  inf[a,i+5]:= real(t) + daux/v; {tempo de entrada 6,7,8} {somei
"t" em 21/04/2021}
                  daux:= daux + lx[i];
                  dc:=daux+c;
                  inf[a,i+8]:= real(t) + dc/v; {tempo de saida 9,10,11} {somei
"t" em 21/04/2021}
                end;
              end;
            end;
          end;
        end;

        {----- Fim -----}

        {----- Procedimento imprime ruas -----}
        procedure ruas();
          begin

            for i:=1 to 3 do
              begin
                gotoxy(3,i);write('|');
                gotoxy(7,i);write('|');
                gotoxy(11,i);write('|');
                gotoxy(15,i);write('|');
              end;

            for i:=1 to 20 do
              begin
                j:= i div 2 + 1;
                gotoxy(3,9+i);write('|');
                gotoxy(7,9 + 2*j);write('|');
                gotoxy(11,9 + 2*j);write('|');
                gotoxy(15,9+i);write('|');
              end;

            gotoxy(1, 3); write('___');
            gotoxy(16, 3);
            write('_____');
            gotoxy(1, 5); write('--');
            gotoxy(16, 5);

```

```

write('-----');
-----');
gotoxy(1, 7); write('--');
gotoxy(16, 7);
write('-----');
-----');
gotoxy(1, 9); write('___');
gotoxy(16, 9);
write('_____');
_____');
end;

{----- Analiza se houvera colisao -----}
}
procedure analisa_se_tem_colisao();
begin
  {Analiza se haverá colisao e mostra quem se colidirá}
  r:=1;
  for i:=1 to f do
    begin
      ite := f - i + 1;
      for j:=1 to nx[i] do
        begin
          for k:=1 to f do
            begin
              w:=1;
              while (ny[k]>= w) do
                begin
                  {if (inf[orx[i,j],6] < inf[ory[k,w],6]) and
(inf[orx[i,j],8+k] < inf[ory[k,w],8+k]) then}
                  {if (inf[orx[i,j],5+k] < inf[ory[k,w],5+ite]) and
(inf[orx[i,j],8+k] < inf[ory[k,w],8+ite]) then}
                  if (inf[orx[i,j],5+k] < inf[ory[k,w],5+ite]) and
(inf[orx[i,j],8+k] < inf[ory[k,w],5 +ite]) then
                    begin
                      w:=ny[k]+1;
                    end
                  else
                    {if (inf[orx[i,j],6] > inf[ory[k,w],6]) and
(inf[orx[i,j],8+k] > inf[ory[k,w],8+k]) then}
                    {if (inf[orx[i,j],5+k] > inf[ory[k,w],5+ite]) and
(inf[orx[i,j],8+k] > inf[ory[k,w],8+ite]) then}
                    if (inf[orx[i,j],5+k] > inf[ory[k,w],5+ite]) and
(inf[orx[i,j],5+k] > inf[ory[k,w],8+ite]) then
                      w:=w+1
                    else
                      begin
                        if (inf[orx[i,j],5] > -9.0)and(inf[ory[k,w],5]> -
9.0) then
                          begin
                            colisao[r,1]:=orx[i,j];
                            colisao[r,2]:=ory[k,w];
                            r:=r+1;

                            {Conta o número de colisões que não foram
evitadas!}

                            if (inf[orx[i,j],5]<= 0.0)or(inf[ory[k,w],5]<=0.0)
then
                              cont_coli:= cont_coli + 1;
                            end;

```

```

                w:=w+1;
            end;
        end;
    end;

    end;
    r:=r-1; {r-1 é o número de colisões}
end;
{----- Fim da Analise -----}

{===== Fim dos procedimentos
=====}

begin
{----- inicialização -----}
    randomize;
    m:=0;
    n:=20;
    f:=3;
    ac:=3.0;
    de:=2.0;
    desc:= de;
    dmin_ini:= 1.0;
    dmc:=2300;

    vmax:=25.0;{velocidade maxima igual para todos as vias}
    fim:= 300; {Tempo total de simulação em segundos.}
    for i:=1 to f do
        begin
            lx[i]:=3;
            ly[i]:=3;
        end;

        {ruas(); } {Linha comentada para não imprimir os carros na tela
07/06/2021}

{----- Inicializando a Matriz de Analise e variaveis auxiliares -----
----}

    n_sim:=10000; {Se alterar aqui vai precisar mudar a dimensão da "matriz
analise"}

    for i:= 1 to n_sim do
        begin
            for j:= 1 to 8 do
                begin
                    analise[i,j]:=0.0;
                end;
            end;
        end;

{----- Inicio Varias simulações -----
----}
    for isim:=1 to n_sim do
        begin
            t:= 0;
            cont_coli:= 0;
            ncrizou:= 0;
            soma:=0;

```

```

cx:=201;
cy:=201;
v:=11.0; {velocidade inicial igual para todos os veiculos}
min_t:=1000.0;
min_v:=1000.0;
max_t:=0.0;
max_v:=0.0;

for a:=1 to n do
begin
  colisao[a,1]:=0;
  colisao[a,2]:=0;

  c:= 3 + random(5);
  soma:= soma + c;
  densidade:= soma/(f*(cx + cy - 2));{Foi substituido + 60 por -2, pois os
veiculos são gerados no intervalo de 60 a 140m }
  inf[a,1]:= c; {comprimento do veiculo}
  inf[a,2]:= 1 + random(2); {via x ou y}
  inf[a,3]:= 1 + random(f); {faixa 1, 2 ou 3}
  inf[a,4]:= v; {velocidade}

  if (inf[a,2]=1) then
    d:= dmc + random(cx) {sorteia uma posição inicial com relação ao
comprimento da via x}{alterei de 60 p/ 80 em 30/04/2021}
  else
    d:= dmc + random(cy);{sorteia uma posição inicial com relação ao
comprimento da via y}{alterei de 60 p/ 80 em 30/04/2021}

  cont:=1;

  {--- verifica se a posição inicial não esta ocupando o mesmo lugar no
espaço que
outro veiculo, se estiver é encontrado uma nova posição não ocupada ainda -
---}
  while (cont < a ) do
  begin
    if (inf[a,2]=inf[a-cont,2]) and (inf[a,3]=inf[a-cont,3])and
((d + c +3*dmin_ini) >= inf[a-cont,5])and
(d-3*dmin_ini <= inf[a-cont,5]+ inf[a-cont,1]) then {Inclui soma e
subtração de "dmin_ini" em 30/04/2021}
    begin
      cont:=1;
      if (inf[a,2]=1) then
        d:= dmc + random(cx) {alterei de 60 p/ 80 em 30/04/2021}
      else
        d:= dmc + random(cy); {alterei de 60 p/ 80 em 30/04/2021}
    end
  else
    cont:=cont + 1;

  end;
  { ----- fim da verificação -----}

  inf[a,5]:= d; {distancia do veiculo ao cruzamento}

  if (inf[a,2]=1) then
  begin
    for i:=1 to 3 do
    begin
      inf[a,i+5]:= d/v; {tempo de entrada 6,7,8}
    end
  end
end;

```

```

        d:= d + ly[i];
        dc:=d+c;
        inf[a,i+8]:= dc/v; {tempo de saida 9,10,11}
    end;
end

else
begin
    for i:=1 to 3 do
        begin
            inf[a,i+5]:= d/v; {tempo de entrada 6,7,8}
            d:= d + lx[i];
            dc:=d+c;
            inf[a,i+8]:= dc/v; {tempo de saida 9,10,11}
        end;
    end;

    inf[a,12]:=0.0;
    inf[a,13]:=1.0;
    inf[a,14]:=1.0;
end;
{----- Fim da inicialização -----}

{Inicia a ordenação dos veiculos pela menor distancia do final do mesmo até
o inicio do cruzamento, ou seja, o menor valor de d+c vem primeiro!}

{inicial:= inf; } {copia as informações iniciais dos veiculos (antes de
iniciar a simulação)}{Comentada 05/05/2021}

ordena_separa();
{----- Fim da ordenação -----}

{----- Inicio da Simulação -----}
----}
{comp | via |faixa |velc. | dist. | tel | te2 | te3 | ts1 | ts2 |
ts3 |ac_de | ta | S | frente}
{ruas(); }
{assign(arq_coli,'C:\Users\Danilo\Documents\Danilo\facul
Danilo\sim_coli.txt');
rewrite(arq_coli);
writeln(arq_coli,'Numero de colisoes após t segundos:'); } {Comentado em
04/05/2021}

{textcolor(15);
gotoxy(25,20); write('HAVERA COLISAO ENTRE OS VEICULOS'); {Quando a
simulação é para muitos carros melhor deixar comentado este trecho}
gotoxy(25,21); write('via x | via y'); }

while ((t <= fim) and (ncruzou < n)) do
begin
    analisa_se_tem_colisao(); {verificação ocorrencia de colisão}

    if (t = 0) then analise[isim,7]:= r;

    {----- Imprime quais as colisão -----}

    {writeln(arq_coli,'t =',t,' n_colisoes =', r);
for i:=1 to r do
begin
    {gotoxy(28,21+i);
textcolor(15);

```

```

        write(colisao[i,1]:2,' ',colisao[i,2]:2); }
        {writeln(arq_coli,colisao[i,1]:2,' ',colisao[i,2]:2);} {Comentado
em 04/05/2021}
        end;
        {gotoxy(40,21); write(' d_aprox | d_minima'); } }

{----- Decisão para evitar colisão (MÉTODO 1) -----}
if (r>0) then
begin
for i:=1 to r do
begin
{Analisar se o veiculo pode acelerar:}
{- se acelerar a velocidade não será superior a maxima?}
a:= colisao[i,1];
if (((inf[a,5]<= dmc) and(inf[a,5]> -9.0)) and
((inf[colisao[i,2],5]<= dmc) and (inf[colisao[i,2],5]> -9.0)))
then {If incluso em 03/05/2021}
begin
if (inf[a,4] <= vmax - ac) then
begin
{calculo da distancia com o veiculo da frente se acelerar}
frente:=round(inf[a,15]);
if (frente <> 0) then {se não for o primeiro veiculo então
calcula}
begin
{- se acelerar houvera distancia segura para freiar e não
colidir com o veiculo da frente?}
distancia_do_proximo_frente();

{writeln(arq_coli,'a= ',a,'acelera? dmin =',dmin:4:1,'
dapro =',dapro:4:1); } {Comentado em 04/05/2021}
if (dapro >= dmin) then
inf[a,12]:= 1 {se sim, então acelera.}
else
begin {se não, analisa-se o veiculo da outra via}
a:= colisao[i,2];
if (inf[a,4] <= vmax - ac) then
begin
{calculo da distancia com o veiculo da frente se
acelerar}
frente:=round(inf[a,15]);
if (frente <> 0) then {se não for o primeiro veiculo
então calcula}
begin
{- se acelerar houvera distancia segura para
freiar e não colidir com o veiculo da frente?}
distancia_do_proximo_frente();
{writeln(arq_coli,'a= ',a,'acelera? dmin
=',dmin:4:1,' dapro =',dapro:4:1);} {Comentado em 04/05/2021}

if (dapro >= dmin) then
inf[a,12]:= 1 {se sim, então acelera.}
else
begin {se não, desacelera}
inf[a,12]:= -1;
end;
end
else
inf[a,12]:= 1;

end {-----}

```

```

        else
            begin
                a:= colisao[i,1];
                inf[a,12]:= -1;{se não, desacelera}
            end;
        end;
    end {if (frente <> 0) then}
else
    inf[a,12]:= 1;
end
else {se o veiculo da via x não puder acelerar}
begin
    {Analisar se o veiculo da via y pode acelerar;}
    {- se acelerar a velocidade não será superior a maxima?}
    a:= colisao[i,2];
    if (inf[a,4] <= vmax - ac) then
        begin
            {calculo da distancia com o veiculo da frente se acelerar}
            frente:= round(inf[a,15]);
            if (frente <> 0) then {se não for o primeiro veiculo então
calcula}
                begin
                    {- se acelerar houvera distancia segura para freiar e
não colidir com o veiculo da frente?}
                    distancia_do_proximo_frente();
                    {writeln(arq_coli,'a= ',a,'acelera? dmin =',dmin:4:1,'
dapro =',dapro:4:1);}{Comentado em 04/05/2021}
                    if (dapro >= dmin) then
                        inf[a,12]:= 1 {se sim, então acelera.}
                    else
                        begin {se não, desacelera}
                            inf[a,12]:= -1;
                        end;
                    end
                end
            else
                inf[a,12]:= 1;
            end
        end
    else
        begin
            a:= colisao[i,1];
            inf[a,12]:= -1;
        end;
    end;
end;
end;
end;
end;
}
{----- Fim da decisão -----}

{ Analisa se há algum veiculo lento desnecessariamente }
for a:= 1 to n do
begin
    if (inf[a,4]< 2) then
        begin
            frente:= round(inf[a,15]);
            distancia_do_proximo_frente();
            if (dapro >= dmin) then {Analisa se pode acelerar}
                begin
                    ira_coli:=0;
                    i:=1;
                end;
            end;
        end;
    end;
end;

```

```

j:=round(inf[a,2]); {???? j deve receber o numero da via e
não da faixa!!! trocar 3 por 2 (Arrumei 05/04/21) }
while (colisao[i,j] <> a)and(i<=n) do
  i:=i+1;

if (colisao[i,j] = a) then
  ira_coli:=1;

if (ira_coli=1) then
begin
  inf[a,12]:= 1;
end
else
begin
v:=inf[a,4];
d:=v*ta + ac*ta/2.0; {obs: posso eliminar o 'ta' se
não for mudar}
v:= v + ac*ta;          {V = Vo + a.t ...Nova velocidade}
inf[a,4]:= v;
inf[a,5]:=inf[a,5] - d; {distancia do veiculo ao
cruzamento cruzamento}
instantes_de_cruzamento();
analisa_se_tem_colisao(); {verificação ocorrencia de
colisão}

i:=1;
while (colisao[i,j] <> a)and(i<=n) do
  i:=i+1;

if (colisao[i,j] = a) then {Se haverá colisão então
desisto de acelerar}
begin
v:= v - ac*ta;
inf[a,4]:= v; {Esta linha não tinha, inclui esta
em 01/05/2021}

inf[a,5]:=inf[a,5] + d;
d:= inf[a,5];
instantes_de_cruzamento();
end;
end;
end;
end;

{ Fim na Analise }

{ Atualiza posição, velocidade e tempos de entrada e saida no cruzamento
}
ordena:= false;
for a:=1 to n do
begin
ta:=inf[a,13];
if (inf[a,12] = 0)or(inf[a,12] = 2)or(inf[a,12] = 3) then
begin
d:=inf[a,4]*ta;
inf[a,5]:=inf[a,5] - d; {distancia do veiculo ao cruzamento}
end
else
if (inf[a,12] = 1)or(inf[a,12]= -2) then
begin
{writeln(arq_coli,'auto =', a,' acelerou');}{Comentado em
04/05/2021}

```

```

v:=inf[a,4];
d:=v*ta + ac*ta/2.0; {obs: posso eliminar o 'ta' se não for
mudar}
v:= v + ac*ta;      {V = Vo + a.t ...Nova velocidade}
inf[a,4]:= v;
inf[a,5]:=inf[a,5] - d; {distancia do veiculo ao cruzamento
cruzamento}
{if (inf[a,12] = 1) then
  inf[a,12] := 2
else}
  inf[a,12] := 0; {para de acelerar}

  instantes_de_cruzamento();
ordena:= true;
s:=a;
end
else if (inf[a,12] = -1)or(inf[a,12] = 4) then
begin
  {writeln(arq_coli,'auto =', a,' desascelerou');}{Comentado em
04/05/2021}
  v:=inf[a,4];
  if (v < de) then {Removi o sinal de igual e deixei de dividir por
2 a velocidade. Alterado 09/04/21}
  Begin
    desc:= v;
  end
  else if (v <= 0.0) then      {Inclui este else em 03/05/2021}
  begin
    desc:= 0.0;
    v:= 0.0;
  end;

  d:=v*ta - desc*ta/2.0; {obs: posso eliminar o 'ta' se não for
mudar}
v:= v - desc*ta;      {V = Vo + a.t ...Nova velocidade}
inf[a,4]:= v;
inf[a,5]:=inf[a,5] - d; {distancia do veiculo ao cruzamento
cruzamento}
desc:=de;

  {if (inf[a,12] = -1) then
    inf[a,12] := -2
  else }
    inf[a,12] := 0;{para de desacelerar}

  instantes_de_cruzamento();
ordena:= true;
s:=a;      {Quem será ordenado!}
end;
{if(inf[a,12] >= 2) then inf[a,12]:= inf[a,12]+ 1;}
colisao[a,1]:=0;
colisao[a,2]:=0;
end;
{----- Fim das atualizações -----}

{----- Analise se houvera colisão na traseira e corrige veloc. -----
---}
for a:=1 to n do
begin
  frente:= round(inf[a,15]);

```

```

    if (inf[a,5] > -9.0) and (inf[frente,5] > -9.0) then {este "if" foi
incluido em 03/05/2021}
    begin

        if (frente <> 0) then {se não for o primeiro veiculo então calcula}
        begin
            {a distancia que estarei do veiculo a minha frente no proximo
segundo.}
            {dapro_const = d7-d1-cl+v1-v7}
            vt:= inf[a,4];
            vf:= inf[frente,4];
            desc:=de;
            if ((vt <= de) and (vt>0.0)) then {Inclui + 1 condição
vt>0.0, em 03/05/2021 }
            begin
                de:= vt/2.0;
            end;

            dapro_const:= inf[a,5]-inf[frente,5]-inf[frente,1] + vf - vt;
            dmin:= dmin_ini + (vf*vf - vt*vt)/(-2.0*de);

            if (dapro_const < dmin)then {se sim: reduz a velocidade}
            begin
                v:=vt;
                d:=v*ta - de*ta/2; {obs: posso eliminar o 'ta' se não for
mudar}

                v:= v - de*ta; {V = Vo + a.t ...Nova velocidade}
                inf[a,4]:= v;
                {writeln(arq_coli,'auto =', a,'reduziu v =',v);} {Linha
para teste 01/05/2021}{Comentado em 04/05/2021}
                inf[a,5]:=inf[a,5] - d; {distancia do veiculo ao
cruzamento}

                instantes_de_cruzamento();
                ordena:= true;
            end;
        end;
        de:=desc;
    end;
end;

{-----Fim da analise e correção -----}

if (ordena = true) then
begin
    ordena_separa();
    {gotoxy(65,22);
    write(s:3:1); }
end;
{----- imprime simulação -----}

{for i:=1 to f do {Linha comentada para não imprimir os
carros na tela 07/06/2021}
begin
{----- Imprime os veiculos na via X ----- }
textbackground(5);
for j:=1 to nx[i] do
begin
    px[i,j]:= 15 + round(inf[orx[i,j],5]);
    if (px[i,j] > 0) and (px[i,j]< 60) then
    begin
        gotoxy(px[i,j] + 1, 2 + 2*i);
    end;
end;
end;
end;

```

```

        textcolor(15);
        write(orb[i,j]:2);
        for cx:=3 to round(inf[orb[i,j],1]) do
            begin
                gotoxy(px[i,j] + cx, 2 + 2*i);
                textcolor(15); write(' ');
            end;
        end
    else
        {px[i,j]:=1; }
    end;
{----- Imprime os veiculos na via y ----- }
textbackground(3);
for j:=1 to ny[i] do
    begin
        py[i,j]:= 15 + round(inf[ory[i,j],5]);
        if (py[i,j] > 0) and (py[i,j] < 20 ) then
            begin
                gotoxy(4*i, py[i,j]+1);
                textcolor(15);
                write(ory[i,j]:2);
                for cy:= 2 to round(inf[ory[i,j],1]) do
                    begin
                        gotoxy(4*i, py[i,j]+ cy);
                        textcolor(15);
                        write(' ');
                    end;
                end
            else
                {py[i,j]:=1; }
            end;
    end;
delay(300);
{----- Apaga os veiculos impresos na via x ----- }
for i:=1 to f do
    begin
        for j:= 1 to nx[i] do
            begin
                if (px[i,j] > 0) and (px[i,j]< 60) then
                    for cx:=1 to round(inf[orb[i,j],1]) do
                        begin
                            textbackground(0);
                            gotoxy(px[i,j] + cx, 2 + 2*i);
                            textcolor(0);
                            write(orb[i,j]:2);
                        end;
                    end;
            end;
{----- Apaga os veiculos impresos na via y ----- }
        for j:= 1 to ny[i] do
            if (py[i,j] > 0) and (py[i,j] < 20 ) then
                for cy:= 1 to round(inf[ory[i,j],1]) do
                    begin
                        textbackground(0);
                        gotoxy(4*i, py[i,j]+ cy);
                        textcolor(0); write(' ');
                    end;
                end;
        end;}

{-----Apaga distancia de aproximação e dist. minima -----}
{if t < 80 then
    begin

```

```

textcolor(0);
for i:= 1 to (n div 3) do
begin
gotoxy(40, 22 + i);
write(' ');
end;

{----- Apaga lista de colisões -----}
textcolor(0);
for i:= 1 to (n div 3) do
begin
gotoxy(25, 21 + i);
write(' ');
end;

end;} {Linha comentada para não imprimir os carros na tela
07/06/2021}
{----- Fim da impresssão -----}
{----- Calcula quantos veiculos passaram o cruzamento no tempo t --- }
n_cruzou[t]:= 0;
m := 0;
for i:= 1 to n do
begin
if (inf[i,5]<= -9) then
m:= m+1;
end;
n_cruzou[t]:=m;
ncruzou := m;
m:=0;
{----- Fim do Calculo -----}
t:=t+1;
end;

{----- Analises da simulação -----}
analise[isim,1]:= densidade;

for i:=1 to n do
begin
if (inf[i,4]< min_v ) then {busca velocidade minima}
min_v:= inf[i,4];

if (inf[i,4]> max_v ) then {busca velocidade maxima}
max_v:= inf[i,4];

analise[isim,4]:= analise[isim,4] + inf[i,4]; {Soma as velocidades
finais}

if (inf[i,11]< min_t ) then {busca tempo minima}
min_t:= inf[i,11];

if (inf[i,11]> max_t ) then {busca tempo maxima}
max_t:= inf[i,11];
end;
analise[isim,2]:= min_v;
analise[isim,3]:= max_v;
analise[isim,4]:= analise[isim,4]/n; {Média das velocidades finais}
analise[isim,5]:= min_t;
analise[isim,6]:= max_t;

analise[isim,8]:= cont_coli; {Número de colisões efetivas}

```

```

    {----- fim das Analises -----}
end;
{close(arq_coli); } {Comentado em 04/05/2021}
{----- Fim da Simulação -----}

{----- Imprimi na tela -----}

assign(arq_s, 'C:\Users\Danilo\Documents\simul_auto.txt');
rewrite(arq_s);
writeln;
writeln(arq_s, 'A matriz com informacao de cada veiculo:');
writeln(arq_s, 'ID comp | via |faixa |velc. | dist. | tel | te2 |
te3 | ts1 | ts2 | ts3 |ac_de | ta | S| frente');
for i:=1 to n do
begin
write(arq_s, i);
for j:=1 to 15 do
begin
{write(arq_s, ' ', inf[i,j]:5:2, '|'); }
write(arq_s, ' ', inf[i,j]:5:2);
end;
writeln(arq_s);
{writeln(arq_s, '-----
-----')};
writeln(arq_s); }
end;
writeln(arq_s, 'densidade = ', densidade:5:3);
writeln(arq_s, 'Colisões efetivas = ', cont_coli);

writeln(arq_s, 'A matriz com informacao Iniciais de cada veiculo:');
writeln(arq_s, 'ID comp via faixa velc. dist. tel te2 te3 ts1 ts2 ts3
ac_de ta S frente');
for i:=1 to n do
begin
write(arq_s, i);
for j:=1 to 15 do
begin
{write(arq_s, ' ', inf[i,j]:5:2, '|'); }
write(arq_s, ' ', inicial[i,j]:5:2);
end;
writeln(arq_s);
end;

for i:=1 to fim do
begin
writeln(arq_s, 'Até o tempo t =', i, 's, passou ', n_cruzou[i], '
veiculo(s)');
end;
close(arq_s);
{readln;} {Comentei em 05/05/2021}

{----- Imprime a matriz de analise -----}
assign(arq_analise, 'C:\Users\Danilo\Documents\analise.txt');
rewrite(arq_analise);

writeln(arq_analise, 'sim densidade v_min v_max v_med t_ini t_fim
coli_ini coli_efetiva');
for i:=1 to n_sim do
begin
write(arq_analise, i);

```

```
for j:=1 to 8 do
  begin
    write(arq_analise, ' ', analyse[i,j]:6:2);
    end;
  writeln(arq_analise);
end;

close(arq_analise);
end.
```

APÊNDICE 2 – CÓDIGO DO MÉTODO 2

```

{----- Decisão para evitar colisão (MÉTODO 2)-----}
vetor_r[round(t)]:=r;
if (r>0) then
begin
aux_r:= r;
rr:=r;
cont_sol:= 0;
aux_inf:= inf;
aux_colisao:= colisao;
n_sol:= 71*rr; {não melhorou, mesmo resultado quando fixo em 1000}

while ((aux_r <> 0) and (cont_sol <= n_sol)) do
begin

for i:=1 to rr do
begin
{----- Via X -----}

a:= colisao[i,1];
P_solu[i,1] :=0;
P_solu[i,2] :=0;
sol[i,1] :=0;
sol[i,2] :=0;

{Analisar se o veiculo pode acelerar:}
{- se acelerar a velocidade não será superior a maxima?}
if (((inf[a,5]<= dmc) and(inf[a,5]> -9.0)) and
((inf[colisao[i,2],5]<= dmc) and (inf[colisao[i,2],5]> -9.0)))
then {If incluso em 03/05/2021}
begin

if (inf[a,4] <= vmax - ta*ac) then
begin
begin
{calculo da distancia com o veiculo da frente se acelerar}
frente:=round(inf[a,15]);
if (frente <> 0) then {se não for o primeiro veiculo então
calcula}
begin
{- se acelerar haveria distancia segura para freiar e não
colidir com o veiculo da frente?}
distancia_do_proximo_frente();

{writeln(arq_coli,'a= ',a,'acelera? dmin =',dmin:4:1,'
dapro =',dapro:4:1); } {Comentado em 04/05/2021}
if (dapro >= dmin) then
P_solu[i,1]:= 1 {se sim, então acelera.}
end
else P_solu[i,1]:= 1; {se sim, então acelera.}
end;

if (inf[a,4] >= ta*de) then
begin
P_solu[i,1]:= P_solu[i,1] + 2; {se sim, então desacelera.}

```

```

end;

{----- Via y -----}
a:= colisao[i,2];

{Analisar se o veiculo pode acelerar;}
{- se acelerar a velocidade não será superior a maxima?}
if (inf[a,4] <= vmax - ta*ac) then
begin
  {calculo da distancia com o veiculo da frente se acelerar}
  frente:=round(inf[a,15]);
  if (frente <> 0) then {se não for o primeiro veiculo então
calcula}
begin
  {- se acelerar houvera distancia segura para freiar e não
colidir com o veiculo da frente?}
  distancia_do_proximo_frente();

  {writeln(arq_coli,'a= ',a,'acelera? dmin =',dmin:4:1,'
dapro =',dapro:4:1); } {Comentado em 04/05/2021}
  if (dapro >= dmin) then
    P_solu[i,2]:= 1 {se sim, então acelera.}
  end
  else P_solu[i,2]:= 1; {se sim, então acelera.}
end;

if (inf[a,4] >= ta*de) then
begin
  P_solu[i,2]:= P_solu[i,2] + 2; {se sim, então desacelera.}
end;

{Analise qual veiculo tem menos opções de alteração de movimento
e passa a ser o primeiro}
if (P_solu[i,1] < P_solu[i,2]) then
begin
  pri:=1;
  seg:=2;
end
else
begin
  pri:=2;
  seg:=1;
end;
{Sorteia uma solução aleatoria para o primeiro veiculo
escolhido}
if (P_solu[i,pri] = 3) then
  sol[i,pri]:=random(3)
  {sol[i,pri]:=1 }
else if (P_solu[i,pri] = 2) then
  sol[i,pri]:=2*random(2)
  else if (P_solu[i,pri] = 1) then
    sol[i,pri]:= random(2);

{Sorteia uma solução aleatoria para o segundo veiculo, escolhido
excluindo a decisão dada para o 1º veiculo}
P_solu[i,seg] := P_solu[i,seg] - sol[i,pri];

```

```

if (P_solu[i,seg] = 3) then
  sol[i,seg]:=random(3)
  {sol[i,seg]:=1 }
else if (P_solu[i,seg] = 2) then
  sol[i,seg]:=2*random(2)
  else if (P_solu[i,seg] = 1) then
    sol[i,seg]:=random(2);

if (sol[i,pri]=2) then
  sol[i,pri]:= -1;

if (sol[i,seg]=2) then
  sol[i,seg]:= -1;

if (sol[i,pri] = sol[i,seg]) then {Só irá acontecer no caso
que ambos sejam zeros.}
  if (P_solu[i,seg] = 3) then
    sol[i,seg] := 1 + random(2)
  else if (P_solu[i,seg] = 2) then
    sol[i,seg]:= -1
  else if (P_solu[i,seg] = 1) then
    sol[i,seg]:= 1;

if (sol[i,seg]=2) then
  sol[i,seg]:= -1;

{Transcreve as soluções para a matriz auxiliar para executar o
movimento}
  inf[colisao[i,1],12]:= sol[i,1];
  inf[colisao[i,2],12]:= sol[i,2];
end;
end;

simula_uma_iteracao();
analisa_se_tem_colisao(); {verificação número de colisões}
cont_coli:= 0;

if (r <= aux_r ) then
  begin
    aux_r := r;
    for j:= 1 to rr do
      begin
        aux_sol[j,1]:= sol[j,1];
        aux_sol[j,2]:= sol[j,2];
      end;
    end;

  inf:= aux_inf;
  colisao:= aux_colisao;
  cont_sol:= cont_sol + 1;

end;
{Transcreve as soluções para a matriz auxiliar para executar o
movimento}
{if (rr = aux_r) then
  begin
    for i:=1 to rr do

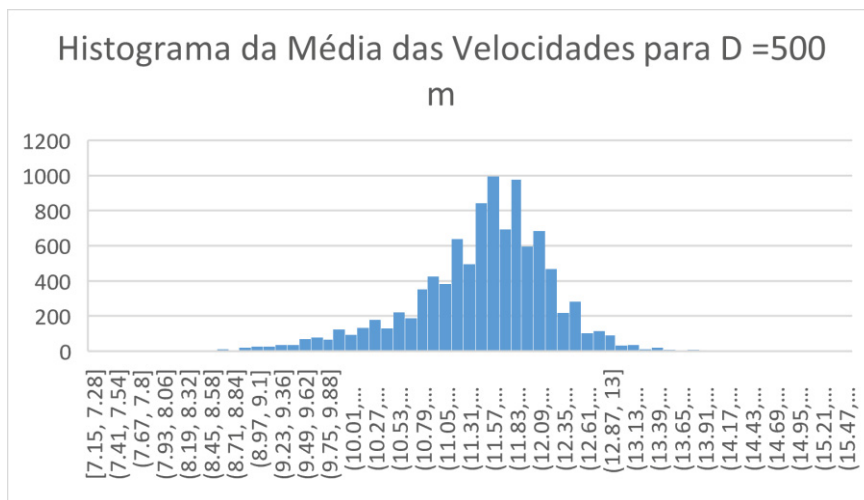
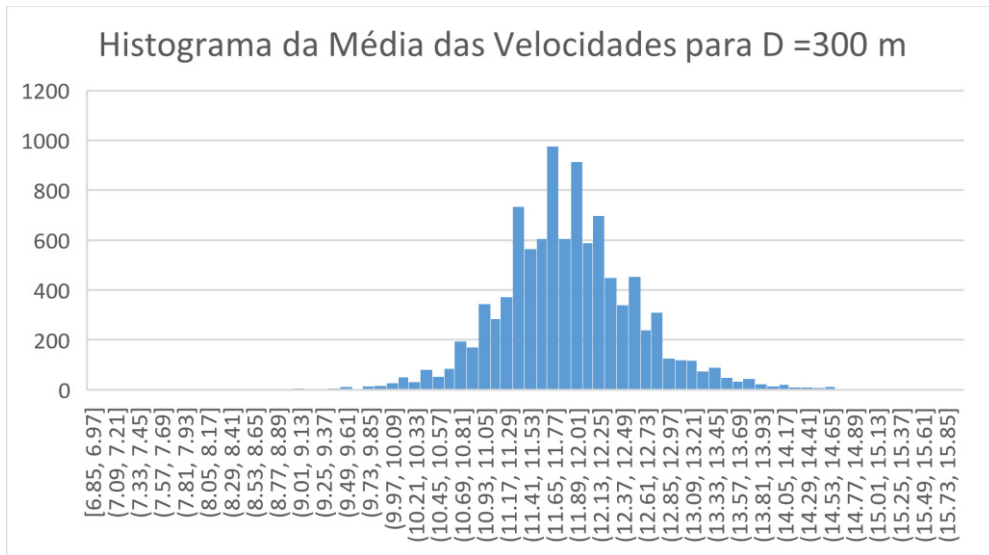
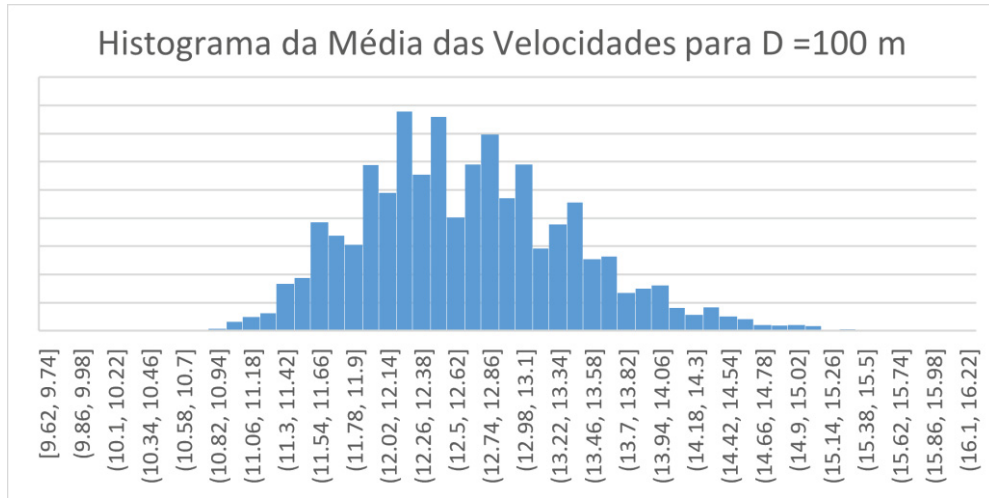
```

```
begin
  inf[colisao[i,1],12]:= sol[i,1];
  inf[colisao[i,2],12]:= sol[i,2];
end;
end
else }
for i:=1 to rr do
begin
  inf[colisao[i,1],12]:= aux_sol[i,1];
  inf[colisao[i,2],12]:= aux_sol[i,2];
end;
end;
{----- Fim da decisão -----}
```

APÊNDICE 3 – HISTOGRAMAS DAS VELOCIDADES MÉDIAS

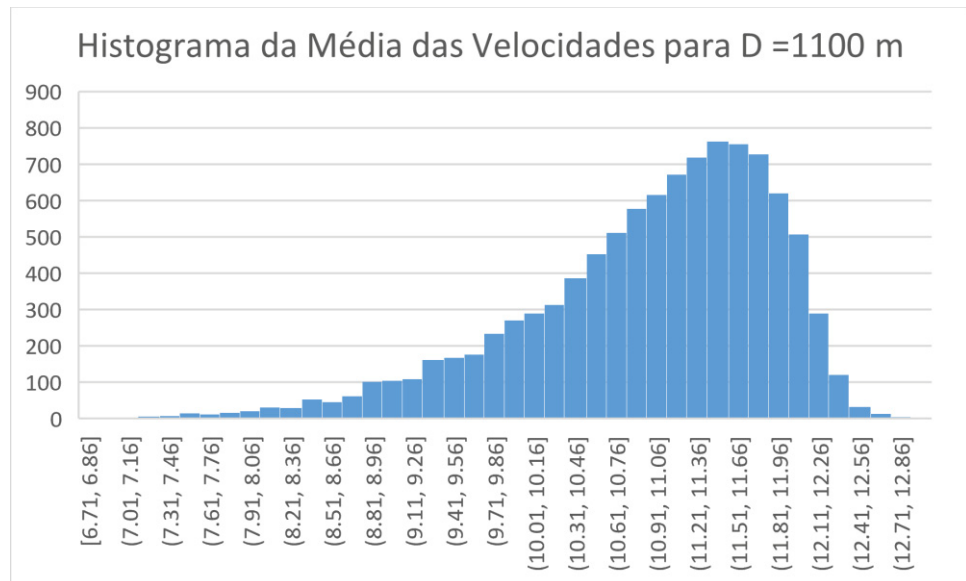
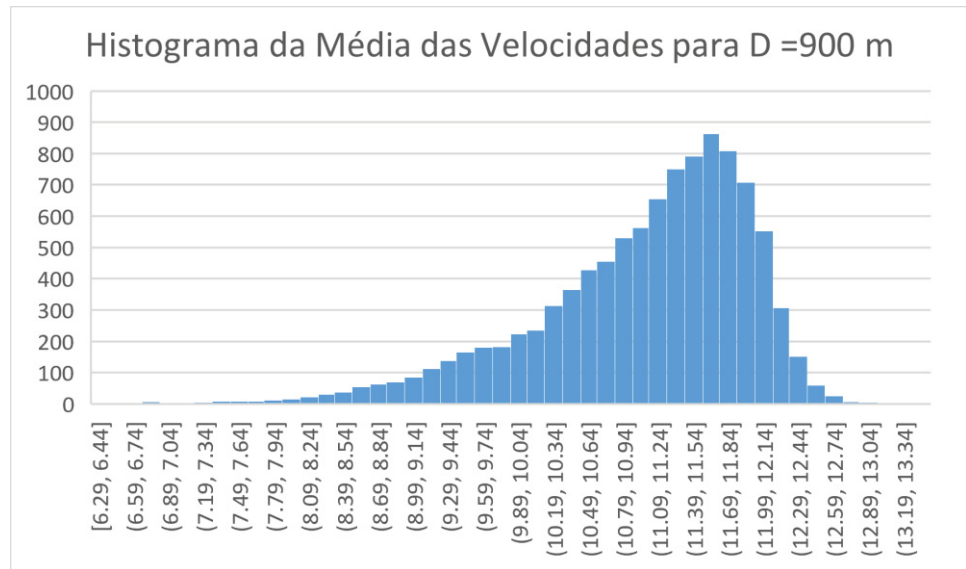
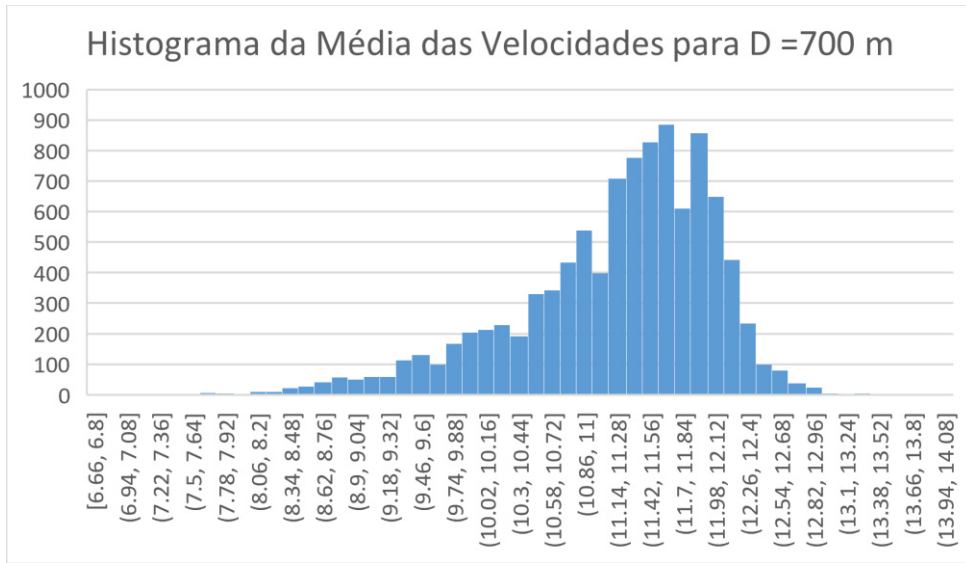
As Figuras 14 a 17 a seguir apresentam os histogramas das médias das velocidades em [m/s] para cada distância de controle D considerada para o método 1.

FIGURA 14 – HISTOGRAMAS DAS MÉDIAS DAS VELOCIDADES PARA D = 100, 300 E 500 M



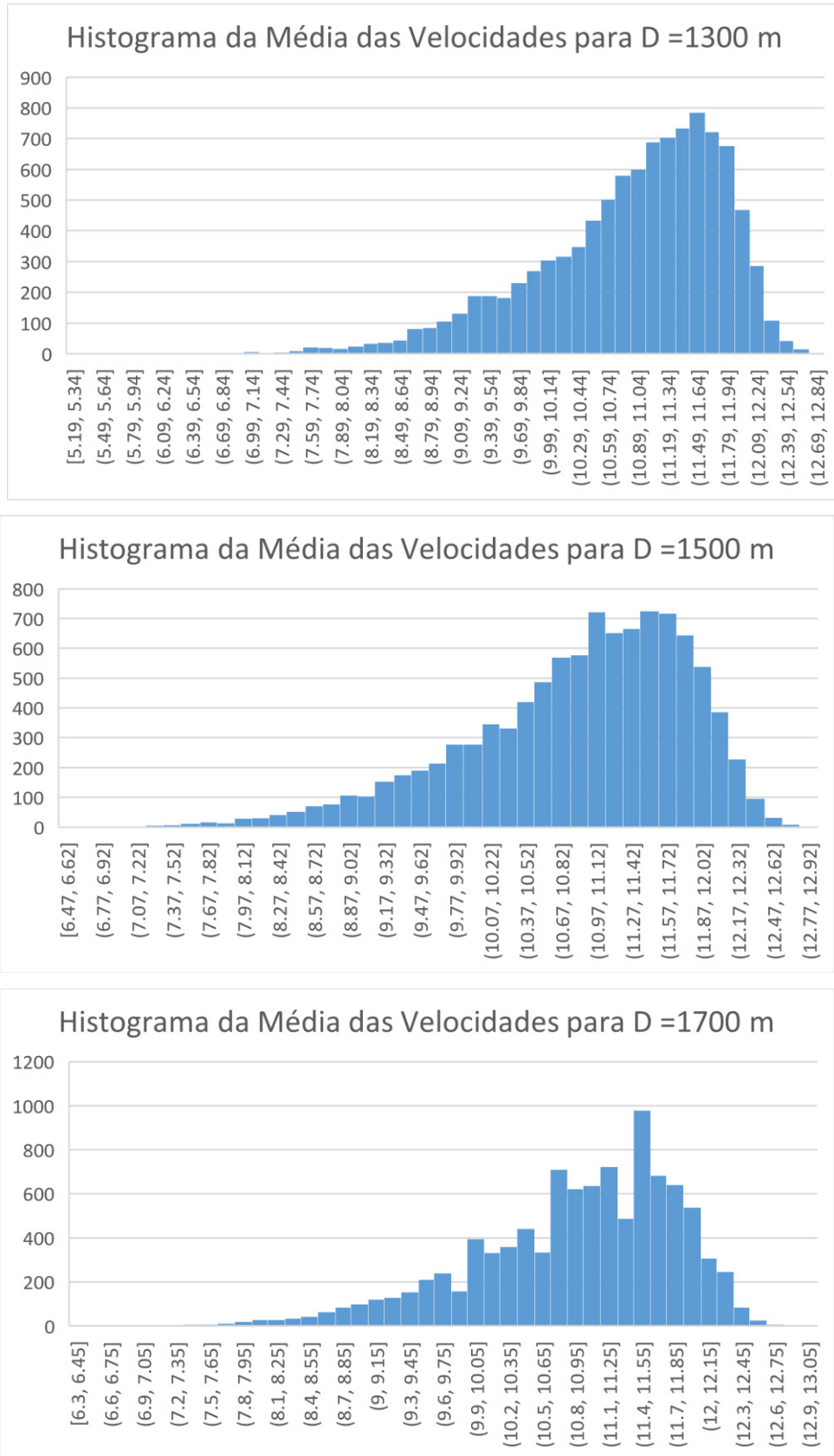
FONTE: O autor (2021).

FIGURA 15 – HISTOGRAMAS DAS MÉDIAS DAS VELOCIDADES PARA D = 700, 900 E 1100 M



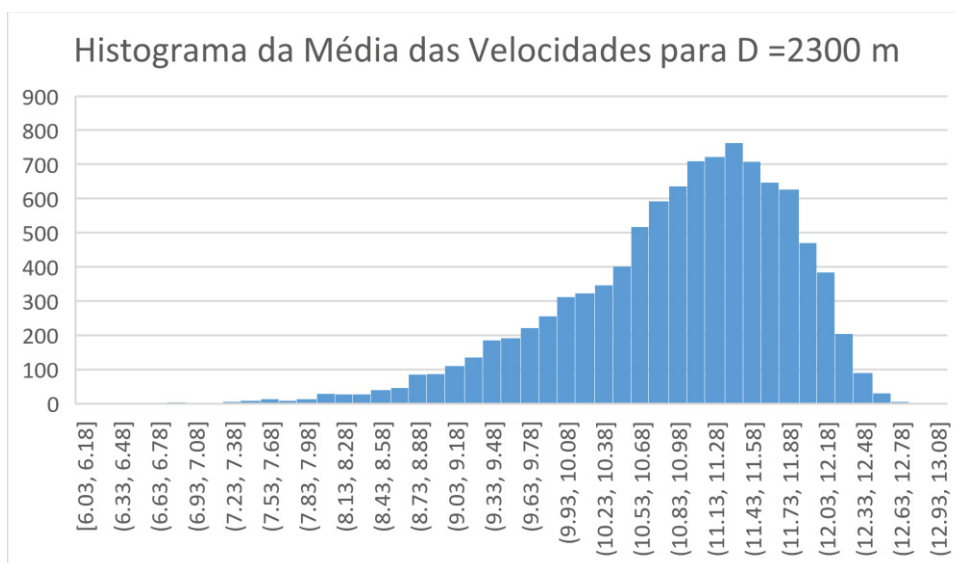
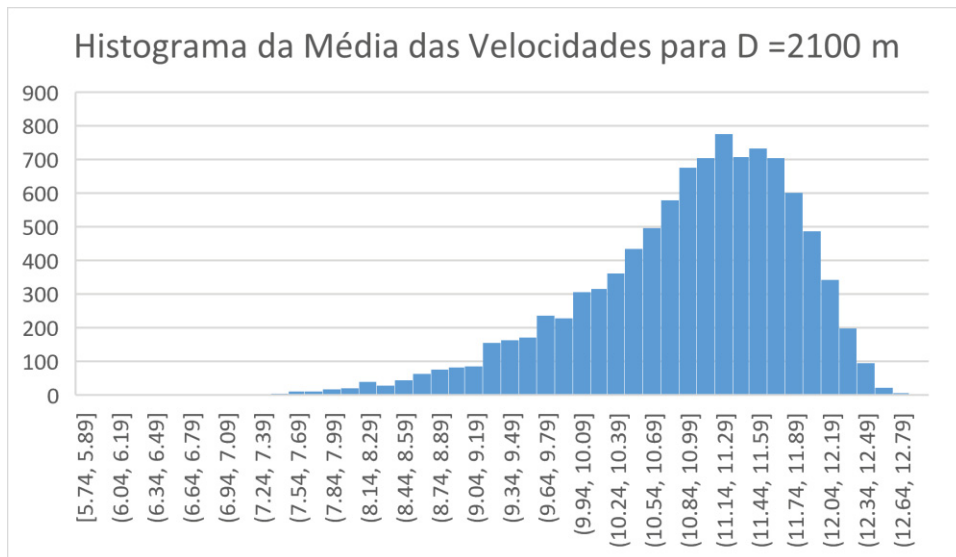
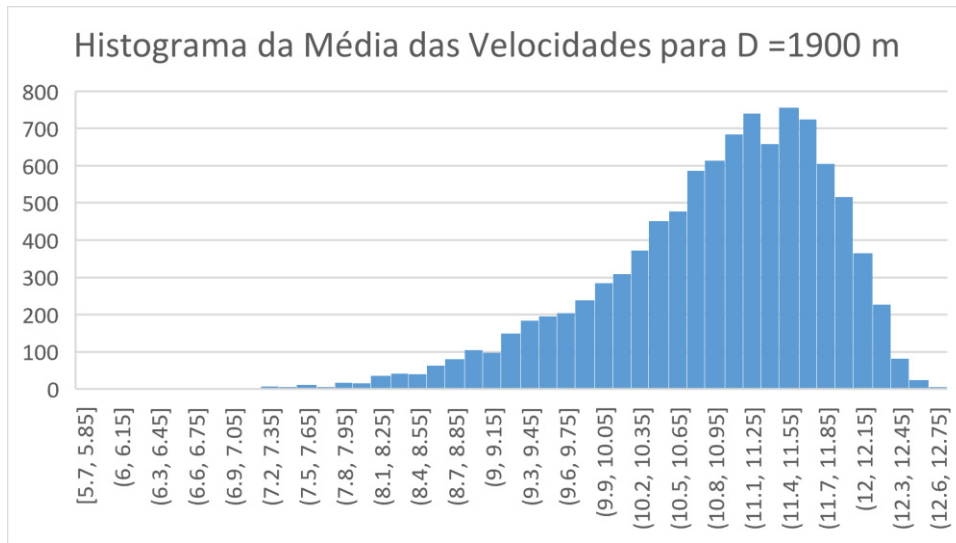
FONTE: O autor (2021).

FIGURA 16 – HISTOGRAMAS DAS MÉDIAS DAS VELOCIDADES PARA D = 1300, 1500 E 1700 M



FONTE: O autor (2021).

FIGURA 17 – HISTOGRAMAS DAS MÉDIAS DAS VELOCIDADES PARA D = 1900, 2100 E 2300 M



FONTE: O autor (2021).

Nas Figuras 14 a 17 observa-se que os histogramas se aproximam de uma distribuição Normal, mas não efetivamente, pois todos os testes de Kolmogorov-Smirnov realizados para cada conjunto das médias das velocidades, apresentaram p-valores inferiores a 0,01. Na Figura 14 quando $D = 100, 300$ e 500 m a média das médias das velocidades são maiores que na Figura 17, onde os valores de D são maiores.