FEDERAL UNIVERSITY OF SANTA CATARINA
TECHNOLOGICAL CENTER
GRADUATE PROGRAM IN AUTOMATION AND SYSTEM ENGINEERING

Elham Ahmadi

Signal-Free Path-Free Intersection Control:
A Model Predictive Contouring Control Approach

Florianópolis
2023

Elham Ahmadi

Signal-Free Path-Free Intersection Control:
A Model Predictive Contouring Control Approach

Ph.D. Dissertation submitted to the Graduate Program in automation and system engineering of the Federal University of Santa Catarina for the Ph.D. Degree in Automation and Systems Engineering.
Supervisor:: Prof. Rodrigo Castelan Carlson, Dr.
Co-supervisor:: Prof. Werner Kraus Junior, Dr.

Florianópolis
2023

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Ahmadi, Elham
Signal-Free Path-Free Intersection Control : A Model Predictive Contouring Control Approach / Elham Ahmadi ; orientador, Rodrigo Castelan Carlson, coorientador, Werner Kraus Junior, 2023.
122 p.

Tese (doutorado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia de Automação e Sistemas, Florianópolis, 2023.

Inclui referências.

1. Engenharia de Automação e Sistemas. 2. Veículos conectados sob condução automatizada. 3. Interseções urbanas. 4. Controle de contorno preditivo baseado em modelo. 5. Prevenção de colisões. I. Castelan Carlson, Rodrigo. II. Kraus Junior, Werner. III. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia de Automação e Sistemas. IV. Título.

Elham Ahmadi

Signal-Free Path-Free Intersection Control:
A Model Predictive Contouring Control Approach

O presente trabalho em nível de doutorado foi avaliado e aprovado por banca examinadora composta pelos seguintes membros:

Prof. Amit Bhaya, Dr.
Universidade Federal do Rio de Janeiro

Prof. Konstantinos Ampountolas, Dr.
University of Thessaly

Prof. Ubirajara Franco Moreno, Dr.
Universidade Federal De Santa Catarina

Certificamos que esta é a **versão original e final** do trabalho de conclusão que foi julgado adequado para obtenção do título de Ph.D. in Automation and Systems Control Engineering.

Documento assinado digitalmente
Julio Elias Normey Rico
Data: 18/12/2023 16:19:38-0300
CPF: ***.840.859-**
Verifique as assinaturas em https://v.ufsc.br

Prof. Julio Elias Normey Rico, Dr.
Coordenador do Programa

Documento assinado digitalmente
Rodrigo Castelan Carlson
Data: 18/12/2023 15:14:44-0300
CPF: ***.743.969-**
Verifique as assinaturas em https://v.ufsc.br

Prof. Rodrigo Castelan Carlson, Dr.
Orientador

Florianópolis, 2023.

ACKNOWLEDGEMENTS

I would like to express my heartfelt gratitude to my advisor Professor Rodrigo Castelan Carlson and my co-advisor Professor Werner Kraus Junior, whose unwavering support, invaluable guidance, and profound insights have been instrumental in shaping this thesis. Your dedication to fostering intellectual growth and your commitment to excellence have been truly inspiring, and I am deeply thankful for the privilege of learning under your mentorship. I extend my sincere appreciation to Professor Giuseppe Notarstefano, Professor Eduardo Camponogara, and Professor Ehsan Taheri for their enlightening guidance. Your expertise and encouragement have played a pivotal role in broadening my horizons and refining my academic pursuits. I must also extend my deepest appreciation to my beloved husband, whose unwavering encouragement and patient support have been my pillars of strength throughout this academic endeavor. Your belief in me and your steadfast presence has been my driving force, and I am grateful beyond words for your love and understanding. Last but not least, I want to express my heartfelt thanks to my family for their constant love, support, and encouragement. This thesis would not have been possible without the contributions, guidance, and unwavering support of these remarkable individuals, and for that, I am truly humbled and grateful.

RESUMO

Durante décadas, o projeto das redes de tráfego seguiu um padrão convencional, com faixas de tráfego fixas, caminhos predefinidos dentro dos cruzamentos e sinalização/semáforos de tráfego controlando movimentos conflitantes. No entanto, o panorama dos transportes está passando por uma mudança profunda com o advento dos veículos conectados sob condução automatizada (CVAD). Esta tecnologia transformadora preparou o terreno para reimaginar a própria essência da gestão de tráfego. Os pesquisadores buscam um futuro mais inteligente, seguro e eficiente, investigando cruzamentos sem semáforos para remover a sinalização de tráfego tradicional. Isto aumenta a eficiência do tráfego através de intervalos mais curtos entre veículos e da eliminação do tempo perdido em interseções antes semaforizadas. No entanto, apesar de algumas melhorias na flexibilidade, os veículos ainda estão vinculados a faixas e caminhos predeterminados, restringindo a capacidade e o potencial viário. Para romper o paradigma predominante, esta tese dá um passo à frente ao propor o controle de interseção sem semáforos e com caminhos livres (*Signal-free Path Free Intersection Control* - SPIC). Trata-se de uma estratégia de coordenação de veículos em interseções urbanas que oferece melhor aproveitamento do espaço da interseção visando melhorar a eficiência e a segurança do tráfego. O SPIC libera os veículos da necessidade de seguir caminhos predefinidos nas interseções, permitindo que percorram trajetórias ótimas. Introduzimos duas formulações de controle ótimo para resolver o problema SPIC em uma interseção urbana. O SPIC é transformado em um problema de programação não linear usando séries finitas de Fourier ou curvas de Bézier, juntamente com noções de discretização. Os resultados numéricos revelam trajetórias quasi-ótimas e sem colisões, com o método Bézier mostrando um desempenho ligeiramente melhor. No entanto, ambas as abordagens estão limitadas a um conjunto existente de veículos com estados iniciais e finais pré-definidos. Para resolver a limitação destas primeiras abordagens, esta tese também apresenta uma abordagem de horizonte deslizante baseada em otimização que é formulada como um problema de controle de contorno preditivo não linear baseado em modelo (*Nonlinear Model Predictive Contouring Control* - NMPCC). Em particular, propomos o algoritmo SPIC. Este algoritmo estende o NMPCC padrão (*Extended Nonlinear Model Predictive Contouring Control* - ENMPCC) para ser adaptado para resolver o problema SPIC. Ele maximiza o progresso do veículo no cruzamento, lidando com chegadas contínuas, combinando planejamento e seguimento do caminho e garantindo prevenção explícita de colisões para segurança. No entanto, soluções em tempo real podem ser computacionalmente caras devido à não linearidade e à não convexidade do problema. Para melhorar o desempenho computacional, incorporamos um modelo linear variável no tempo (*Linear Time Varying* - LTV), usado no NMPCC padrão, no problema ENMPCC, produzindo o LTV-ENMPCC. O método proposto está encapsulado no simulador SPIC (SPIC-Sim), uma ferramenta de simulação que permite a modelagem da plena utilização de uma intersecção urbana. Os resultados da simulação apresentam trajetórias ótimas e livres de colisão para CVAD com melhor utilização do espaço de interseção, resultando em menor tempo de viagem necessário para os veículos atravessarem a interseção, e em segurança.

Palavras-chave: Veículos conectados sob condução automatizada. Interseções urbanas. Controle de contorno preditivo baseado em modelo. Prevenção de colisões.

RESUMO EXPANDIDO

INTRODUÇÃO

Os Veículos Conectados sob Condução Automatizada (*Connected Vehicles under Automated Driving* - CVAD) apresentam uma tecnologia transformadora com potencial para melhorar o fluxo de tráfego, a segurança, a redução de poluentes, a sustentabilidade e a eficiência de combustível. Aproveitando recursos avançados de automação, os CVAD oferecem benefícios como regulação de tráfego, redução de fatalidades por erros humanos e maior acessibilidade. As interseções urbanas, críticas para os sistemas de transporte, sofrem com perdas de eficiência, colisões e congestionamentos, representando ameaças à vida econômica e social. Os métodos tradicionais de controle de tráfego têm limitações e, apesar dos avanços, o paradigma organizacional estabelecido para a infra-estrutura rodoviária permanece praticamente inalterado. Para enfrentar estes desafios, é necessária uma nova estratégia organizacional adaptada ao CVAD nas interseções urbanas.

OBJETIVOS

Esta tese visa priorizar o desenvolvimento de estratégias de controle eficientes para coordenar os CVAD de forma eficaz e segura em interseções urbanas complexas. Para permitir que os veículos circulem com mais flexibilidade e também utilizem plenamente o espaço de interseção, nesta tese focamo-nos na situação em que os veículos não estão vinculados a caminhos predefinidos e faixas viárias fixas, o que é referido como interseção sem caminhos. Permite-se conduzir em qualquer lugar da superfície bidimensional da interseção. Em geral, os benefícios das interseções sem caminhos são um comportamento de condução mais suave, maior capacidade da interseção e redução do congestionamento do tráfego. Soma-se a isso o fato de semáforos com configurações de sinalização inadequadas ou que não estão dinamicamente adaptados ao tráfego em tempo real, não só diminuem a eficiência e a segurança do tráfego, mas podem até causar atrasos e congestionamentos nas interseções. Nesta tese consideramos interseções livres de semáforos e com caminhos livres.

Em particular, propomos um problema de controle de interseção livre de semáforos e com caminhos livres (*Signal-fee Path-free Intersection Control* - SPIC) para coordenar CVAD em interseções urbanas. O SPIC permite movimentos suaves e seguros dos veículos sem a necessidade dos semáforos tradicionais ou caminhos predefinidos. Primeiro, um problema de controle ótimo baseado na teoria das séries finitas de Fourier e curvas de Bézier é proposto para resolver o problema que leva à resolução de um problema de programação não linear (*NonLinear Programming* - NLP). Além disso, um método de controle de contorno preditivo de modelo não linear estendido e adaptado (*Nonlinear Model Predictive Contouring Control* - NMPCC) é introduzido para resolver a solução do problema SPIC, ao qual nos referimos como NMPCC estendido (*Extended Nonlinear Model Predictive Contouring Control* - ENMPCC). O método ENMPCC gera trajetórias CVAD ideais e livres de colisões com base em caminhos de referência independentes do tempo e continuamente diferenciáveis, com foco na maximização do progresso ao longo dos caminhos. Para garantir a prevenção de colisões, as áreas ocupadas pelos CVAD são modeladas como conjuntos politópicos em que cada conjunto deve manter uma distância mínima de segurança dos demais conjuntos. Para aumentar a eficiência numérica, incorporamos um modelo linear variável no tempo (*Linear Time Varing* - LTV), usado no NMPCC padrão, no problema ENMPCC, produzindo LTV-ENMPCC.

METODOLOGIA

A metodologia desta tese gira em torno de abordar o problema SPIC através de métodos de controle ótimo e também modelar métodos de controle de contorno preditivo:

O Capítulo 3 apresenta o Controle de Interseção Sem Caminho e Sem Sinal, que é um problema novo para o gerenciamento de interseções urbanas. O SPIC promove uma melhor utilização do espaço de interseção para melhorar a eficiência geral do tráfego, aproveitando o potencial dos CVAD para permitir a travessia sem caminhos numa interseção sem sinal, que é referida como *plaza*. No *plaza*, a interseção transforma-se num espaço delimitado, onde caminhos livres são permitidos desde que as trajetórias resultantes não se cruzem. Na verdade, além de evitar colisões entre veículos, as restrições para os limites do *plaza* devem ser projetadas para impedir que os CVAD violem seus limites. Para este fim, restrições adicionais são impostas ao problema SPIC para definir a geometria do plaza.

O Capítulo 4 introduz o problema de controle ótimo de trajetória de interseção (*Intersection Trajectory Optimal control Problem* - ITOP) como um tipo específico de problema SPIC. O ITOP pode ser definido como a obtenção de trajetórias ótimas, sem caminhos pré definidos, nas quais os veículos partem de seus estados iniciais e, após cruzarem uma interseção viária, chegam aos estados finais para minimizar um ou mais critérios e satisfazer os limites físicos da praça e as restrições para evitar colisões. O ITOP permite o uso de uma estrutura juntamente com dois métodos: um baseado nas séries finitas de Fourier (*Finite Fourier Series* - FFS) e outro utilizando curvas Bézier. Esses métodos são empregados para resolver o ITOP de forma eficaz. De acordo com as informações recebidas sobre os estados do veículo, e também a geometria da interseção, os métodos FFS e curvas de Bézier podem gerar trajetórias quase ótimas e livres de colisão dos CVAD, considerando a ausência de semáforos e caminhos predeterminados. Em particular, o uso dos métodos FFS e Bézier e noções de discretização convertem o problema ITOP em um problema de programação não linear, com coeficientes de Fourier ou Bézier como parâmetros desconhecidos.

O Capítulo 5 tem como objetivo apresentar uma nova formulação para o problema SPIC e introduzir um novo método e algoritmo que ofereça uma solução eficaz para resolver o problema SPIC. Para fazer isso, propomos uma abordagem de horizonte deslizante baseada em otimização que é formulada como um controle de contorno preditivo de modelo não linear (NMPCC). Em particular, estendemos e adaptamos o método NMPCC padrão para resolver o problema SPIC, ao qual nos referimos como NMPCC estendido (ENMPCC). O método ENMPCC gera trajetórias CVAD ideais e livres de colisões com base em caminhos de referência independentes do tempo e continuamente diferenciáveis, concentrando-se em maximizar o progresso ao longo dos caminhos e, ao mesmo tempo, garantir a segurança. Para garantir a prevenção de colisões entre CVAD, os veículos são modelados como conjuntos politópicos, onde cada conjunto é obrigado a manter uma distância mínima de segurança dos demais conjuntos. A natureza não linear e não convexa do problema SPIC, decorrente do modelo do veículo e das restrições de segurança, apresenta desafios computacionais. Empregamos modelos lineares variantes no tempo (LTV) obtidos através da linearização de funções não lineares. Além disso, aproveitamos a teoria da dualidade para suavizar as restrições para evitar colisões e permitir a utilização de abordagens de solução eficientes.

CONSIDERAÇÕES FINAIS

A tese conclui demonstrando a superioridade dos métodos propostos, nomeadamente ENMPCC e LTV-ENMPCC, em termos de tempo total gasto, consumo de combustível e tempo de cálculo em comparação com a condução convencional baseada em caminhos.

Palavras-chave: Veículos conectados sob condução automatizada. Interseções urbanas. Problema de controle ótimo. Problema de controle de contorno preditivo baseado em modelo. Prevenção de colisões. Otimização convexa.

ABSTRACT

For decades the design of traffic networks has followed a conventional pattern, with fixed traffic lanes, predefined paths within intersections, and traffic signs/signals controlling conflicting movements. However, the landscape of transportation is experiencing a profound shift with the advent of connected vehicles under automated driving (CVAD). This transformative technology has set the stage for reimagining the very essence of traffic management. Researchers aim for a smarter, safer, and more efficient future by investigating signal-free intersections to remove traditional traffic signs/signals. This boosts traffic efficiency through shorter headways and elimination of lost time. However, despite some flexibility improvements, vehicles are still bound to predetermined lanes and paths, restricting the capacity and potential of roads and intersections. To disrupt the prevailing paradigm, this thesis takes a step forward by proposing signal-free path-free intersection control (SPIC). It is a strategy for vehicle coordination at urban intersections that offers better use of the intersection space targeting to improve traffic efficiency and safety. SPIC liberates vehicles from predefined paths within intersections, empowering them to traverse along optimal trajectories. We introduce two optimal control formulations to address the SPIC problem at an urban intersection. The SPIC is transformed into a non-linear programming (NLP) problem using either the finite Fourier series (FFS) or Bézier curves, along with discretization notions. Numerical results reveal near-optimal collision-free trajectories, with the Bézier method showing slightly better performance. However, both approaches are limited to an existing set of vehicles with pre-defined initial and final states. To tackle the limitation of these first approaches, this thesis also presents an optimization-based receding horizon approach that is formulated as a nonlinear model predictive contouring control (NMPCC) problem. In particular, we propose the SPIC algorithm. This algorithm extends the standard NMPCC (ENMPCC) to be tailored for solving the SPIC problem. It maximizes vehicle progress at the intersection, handling continuous arrivals, combining path planning and following, and ensuring explicit collision avoidance for safety. However, real-time solutions can be computationally expensive due to non-linearity and non-convexity. To enhance capabilities, we incorporate a linear time-varying (LTV) model, used in standard NMPCC, into the ENMPCC problem, yielding LTV-ENMPCC. The proposed method is encapsulated in the SPIC simulator (SPIC-Sim), a simulation tool that enables the modeling of full utilization of an urban intersection. The simulation results present optimal and collision-free trajectories for CVAD with improved intersection space utilization , resulting in lower travel time required for vehicles to traverse the intersection, with guaranteed safety.

Keywords: Connected vehicles under automated driving. Urban intersections. Model predictive contouring control. Collision avoidance.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

BCs	Boundary Conditions
CCP	Contouring Control Problem
CCS	Cartesian Coordinate System
CG	Center of Gravity
C-V2X	Cellular Vehicle-to-Everything
CVAD	Connected Vehicles under Automated Driving
DPs	Discretization Points
DSRC	Dedicated Short-Range Communication
ENMPCC	Extended Nonlinear Model Predictive Contouring Control
EoM	Equations of Motion
FCFS	First-Come First-Served
FFS	Finite Fourier Series
IB	Intersection Boundary
ISO	International Organization for Standardization
ITOP	Intersection Trajectory Optimal control Problem
LQR	Linear Quadratic Regulators
LTV	Linear Time-Varying
LTV-EMPCC	Linear Time-Varying Extended Model Predictive Contouring Control
MILP	Mixed-Integer Linear Programming
MINLP	Mixed-Integer Non-Linear Programming
MPC	Model Predictive Control
NLP	Non-Linear Programming
NMPCC	Nonlinear Model Predictive Contouring Control
OCP	Optimal Control Problem
ODEs	Ordinary Differential Equations
PB-MPC	Path-Based Model Predictive Control
SAE	Society of Automotive Engineers
SB	Shape-Based
SPIC	Signal-free Path-free Intersection Control
SPIC-Sim	SPIC Simulator
TTS	Total Travel Spent
UI	User Interface
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-Everything

LIST OF SYMBOLS

n	Dimension of decision variables
o	Number of the inequality constraints
p	Number of the equality constraints
f_0	Objective function of a convex optimization problem
L	Lagrangian function
q	Lagrangian dual function
p^*	Optimal value of a convex optimization problem
x^*	Optimal point of a convex optimization problem
d^*	Optimal point of the dual problem
f	Nonlinear function that defines the dynamic of the system
$\mathbf{x}$	Vector of state variables
$\mathbf{u}$	Vector of control inputs
t	Time
k	Discrete time index
Δt	Sampling time
i	Vehicle index
h	Number of intersection plaza boundaries
T_{f}	Completion time
x	Longitudinal position
y	Lateral position
ψ	Orientation angle
β	Slip angle
θ	Vehicle progress
$\hat{\theta}$	Approximated Vehicle progress
ν	Virtual input
$\nu_{\max}$	Maximum value for the virtual input
δ	Steering angle
$\dot{\delta}$	Steering angle rate
a	Acceleration
a_x	Acceleration in coordinate x
a_y	Acceleration in coordinate y
v	Speed
v_x	Speed in coordinate x
v_y	Speed in coordinate y
Δv	Total speed increment
$v_{\max}$	Maximum speed
$a_{\max}$	Maximum acceleration

$\delta_{\max}$	Maximum staring angle
$\dot{\delta}_{\max}$	Maximum staring angle rate
$v_{\min}$	Minimum speed
$a_{\min}$	Minimum acceleration
$\delta_{\min}$	Minimum staring angle
$\dot{\delta}_{\min}$	Minimum staring angle rate
d_{s}	Minimum safe distance between vehicles
$\mathcal{J}$	Performance index
w_1	Weighting parameter associated to Δv
w_2	Weighting parameter associated to T_{f}
$N_{\mathbf{z}}$	Number of Fourier terms
τ	Scaled time
$\mathbf{z}$	Vector of position state variables
$\mathbf{z}_{\mathrm{a}}$	Vector of the approximated position state variables
$\mathbf{z}_{\mathrm{I}}$	Initial position state variables
$\mathbf{z}_{\mathrm{F}}$	Final position state variables
$\dot{\mathbf{z}}_{\mathrm{I}}$	Initial speed state variables
$\dot{\mathbf{z}}_{\mathrm{F}}$	Final speed state variables
m	Number of discretization points
$A_{\mathbf{z}}$	Matrix of Fourier coefficients
$X_{\mathbf{z}}$	Vector of unknown Fourier/Bézier coefficients
$F_{\mathbf{z}}$	Constant vector of Fourier/Bézier terms obtained from the boundary conditions
n_{a}	Number of discretization points used for initialization of the Fourier/Bézier decision variables
$n_{\mathbf{z}}$	Number of Bézier terms
$P_{\mathbf{z}}$	Unknown Bézier coefficients
$B_{\mathbf{z}}$	Bernstein basis polynomials
T_{a}	Initialization of the completion time
S	Distance between the origin and destination of the vehicle.
T_{B}	Time of the Bézier method
T_{F}	Time of the finite Fourier series method
T_{c}	Computation time
N_{v}	Number of vehicles
w_{r}	Road width
l_{r}	Road length
L	Vehicle's wheelbase
w_{v}	Vehicle width
l_{v}	Vehicle length
μ	Friction coefficient

g	Gravitational acceleration
E_{c}	Lateral (contouring) error
$\hat{E}_{\mathrm{c}}$	Approximated lateral error
E_{l}	Longitudinal (lag) error
$\hat{E}_{\mathrm{l}}$	Approximated longitudinal error
x^{ref}	Reference path with respect to the x axis
y^{ref}	Reference path with respect to the y axis
ϕ	Tangent angle to the reference path
ρ	Polytopic set
λ	Lagrangian multiplier vector associated with inequality constraint
s	Lagrangian multiplier vector associated with equality constraint
Q	ENMPCC state cost weight
R	ENMPCC control input cost weight
q_{c}	Lateral error cost weight
q_{l}	Longitudinal error cost weight
q_{θ}	Progress cost weight
q_{x}	path-based MPC longitudinal weight
q_{y}	path-based MPC lateral weight
r_{ν}	Virtual input cost weight
r_{v}	Acceleration cost weight
r_{δ}	Steering angle cost weight
$\mathcal{X}$	State variable feasible set
$\mathcal{U}$	Controll input feasible set
$\bar{\mathbf{x}}$	Predicted vector of state variables
$\bar{\mathbf{u}}$	Predicted vector of control inputs
$\hat{E}_{\mathrm{c,app}}$	Linearized lateral error
$\hat{E}_{\mathrm{l,app}}$	Linearized longitudinal error
N_{h}	Prediction horizon
N_{I}	Number of road lanes
l_{f}	Distance from front axle to the vehicle center of gravity
l_{r}	Distance from rear axle to the vehicle center of gravity

CONTENTS

1 INTRODUCTION

Connected Vehicles under Automated Driving (CVAD) have emerged as a revolutionary technology with the potential to improve the flow of traffic, safety, pollutant emissions reduction, sustainability, and fuel efficiency. By leveraging advanced automation capabilities, these vehicles offer numerous benefits, such as traffic regulation in congested cities, a reduction in fatalities caused by human errors, and increased accessibility for the elderly and individuals with disabilities (ANDERSON et al., 2014). Thanks to extensive efforts in both academia and industry over the past few decades, a high level of automated driving has become a feasible option for future transportation (BUEHLER; IAGNEMMA; SINGH, 2009).

Furthermore, CVAD have the capability to establish communication with each other and the infrastructure, thereby enhancing the decision-making process. In addition, the integration of automation and communication technologies empowers vehicles to collaborate and make decisions that surpass the capabilities of human drivers. By fostering a connected, automated, and cooperative vehicle environment, advanced strategies for intersection management can be implemented, surpassing the effectiveness of existing technologies such as traffic lights (CHEN, L.; ENGLUND, 2016). Hence, the deployment of CVAD has opened avenues for improving vehicular traffic efficiency, especially in complex urban intersections.

Urban intersections are a crucial part of the transportation systems. Intersecting roadways are necessary to connect people driving, walking, and bicycling from one route to another. However, where roads intersect and paths cross, the resulting conflict points create circumstances with efficiency loss and where crashes can occur. Traffic congestion at intersections is one of the serious threats to the economic and social life of modern societies as well as to the environment, which calls for serious and fundamental solutions. The congestion may cause extreme travel delays and consequently increase pollution and fuel consumption, and decrease traffic safety. In fact, each year roughly one–quarter of traffic fatalities and about one–half of all traffic injuries in the United States are attributed to intersections (FHWA, 2021). Moreover, the collision between vehicles at intersections accounts for anywhere between 25% and 45% of all collisions (DRESNER; STONE, 2008).

The shared road space in an intersection is subject to vehicles with different origins and destinations, leading to conflicts. The conventional way of managing these conflicts is by defining a limited set of allowed paths and associated traffic movements (ROESS, 2004). Right of way for each movement or set of non-conflicting movements is granted sequentially by traffic signals. The research on traffic control at intersections achieved expressive improvements in traffic efficiency (PAPAGEORGIOU et al., 2003). However, the organization of traffic within intersections into allowed paths/movements, the lost times due to traffic cycles, and the high headways kept by human drivers are factors in conventional driving that limit the increase of traffic efficiency. With the advent of CVAD, many methods for coordinating traffic at signal-free intersections have been developed, aiming to improve traffic efficiency by enabling shorter headways and eliminating

lost time (CHEN, L.; ENGLUND, 2016; RIOS-TORRES; MALIKOPOULOS, 2017).

The majority of these methods deal with how to allocate time windows of a scarce resource, the intersection space, to different vehicles. However, since they preserve the notion of vehicular movement and an associated number of allowed paths in the intersection, they do not offer the greatest possible efficiency (AHMADI et al., 2021). Indeed, despite the higher capacity obtained by the elimination of the traffic light cycle and by smaller headways between vehicles enabled by CVAD, the capacity of the intersection ends up being limited by the relationship between paths constrained to pre-established vehicular movements.

Some works sought to increase the use of intersection space by enabling a different organization of traffic at intersections. Stevanovic and Mitrovic (2020) allowed the real-time reversal of the direction of approaching and leaving lanes and corresponding paths. Although the left- and right-turning movements traverse the intersection without any conflicts, the through vehicles are still required to reserve time-space slots to avoid a collision. Another work proposed to expand the number of allowed paths for some traffic movements (HE et al., 2018). Vehicles on any lane were allowed to turn in any direction, at the cost of increasing the number of conflicts inside the intersection. It was shown that the method outperformed conventional driving. However, even in these latter approaches, the number of allowed paths remains restricted, constraining the achievable capacity at the intersection.

As a matter of fact, despite these advances, the paradigm established decades ago for the organization of road infrastructure remains roughly the same. To bridge this gap and advance the current state of the art, it is crucial to develop a novel organizational strategy specifically tailored for CVAD at urban intersections. Additionally, future research endeavors should focus on the implementation of an efficient control strategy to effectively and safely coordinate the movement of CVAD at intersections. This novel organization poses several challenges in terms of coordinating vehicles, ensuring safety, and optimizing the performance of CVAD at complex and unpredictable urban intersections.

1.1 MOTIVATION

Most of the aforementioned methods for intersection management rely on the concept of vehicle movements at an intersection, restricting the possible or allowed paths, thus limiting the use of intersection space, and (i) assume a predefined limited set of possible paths within the intersection; (ii) assume that the CVAD speeds are set to be constant; (iii) consider the intersection with the conflict regions or define vehicle prioritization to avoid collision between vehicles; (iv) none of them took the collision between vehicles and intersection boundaries into consideration; and (v) the majority of these methods are formulated as mixed-integer linear programming (MILP) or mixed-integer non-linear programming (MINLP) problems, which have high complexity due to the many integer variables included. This results in, particularly at peak hours, a waste of scarce intersection space and a loss of efficiency. Consequently, CVAD mobility at the urban intersections is far from being fully exploited. To ensure the high traffic efficiency

and safety of CVADs, it is imperative to devise a novel approach for their coordination.

This thesis is devoted to addressing the intricate problem of coordinating connected vehicles under automated driving within signal-free and path-free intersections. Its overarching objective is to overcome existing limitations and challenge the longstanding paradigm of road utilization, ultimately aiming to optimize traffic efficiency in urban intersections. By prioritizing the principles of traffic safety, this study endeavors to pave the way for a paradigm shift in the traditional approaches to road usage at intersections. The potential for enhancing the usage of available intersection infrastructure arises when vehicles are granted the freedom to fully exploit the intersection space through the optimization of their trajectories along all feasible and free paths.

To enable vehicles to drive with more flexibility, and also fully utilize the intersection space, in this thesis we focus on the situation in which vehicles are not bound to predefined paths and fixed road lanes, which is referred to as *path-free intersection*. It enables driving anywhere on the two-dimensional surface of the intersection. In general, the benefits of the path-free intersection are smoother driving behavior, more intersection capacity, and reduced traffic congestion. In addition, the fact that traffic lights with inappropriate signal settings or the ones that are not dynamically adapted to real-time traffic, not only decrease traffic efficiency and safety but may even cause delay and congestion at the intersections (CHEN, L.; ENGLUND, 2016), in this thesis we consider *signal-free intersection*. In summary, this research is founded upon the following principles:

- The traffic signals are no longer used for the CVAD coordination at the intersection, i.e., we have a signal-free intersection.
- The CVAD paths are no longer fixed and are not confined to the predefined limited set of possible paths within the intersection, i.e., we have a path-free intersection.
- The speeds of CVAD are not restricted to specified patterns or they are no longer set to be constant.
- Applicability to any intersection layout.
- Assurance of collision-free interaction between every pair of vehicles.
- Imposition of constraints on vehicles to remain within the intersection boundaries.
- All CVAD have exact knowledge of their surroundings.
- All CVAD have the capability to establish communication with one another.

1.2 GOALS AND CONTRIBUTIONS

This thesis puts forward a novel strategy for vehicular traffic that leads to free use of the intersection space in the era of connected vehicles under automated driving (CVAD), preserving the safety restrictions, so as to obtain its optimal utilization. To this end, we consider an intersection as an empty space free of movement-related horizontal markings or structural

restrictions, except for the intersection boundaries. We call this space a *plaza*, a signal-free and path-free intersection. The objective is to propose a novel control strategy for the coordination of CVAD at an intersection plaza, which is called *Signal-free Path-free Intersection Control* (SPIC). SPIC coordinates CVAD by calculating a sequence of control inputs for each vehicle that allows them to cross the intersection efficiently without colliding with each other and with the plaza's boundaries.

The difficulties of solving the SPIC problem potentially arise in the following main aspects; (i) the non-linearity of the vehicle dynamics; (ii) handling both the static and dynamic obstacles in a dynamic environment, the plaza; (iii) the non-convexity and high dimensionality of the traffic safety and path constraints; and (iv) the lack of necessary flexibility of the traditional simulation modeling tools to accurately model the full utilization of intersection space, posing a significant barrier to the modeling of different traffic organizations in urban networks.

The main contributions of this thesis are summarized as follows:

- **Modeling**:
  - Enable path-free traversing of connected vehicles under automated driving at signal-free intersections, which is referred to as *plaza*, enhances the efficient use of intersection space.
  - Present and consider the real-world situations including the vehicle's dynamic, different completion times for each connected vehicle under automated driving, and so on, in the SPIC problem formulation.
  - Introduce effective strategies for collision avoidance constraints based on safe distances between CVAD in any direction at all times.
  - Design efficient strategies to account for the geometry of the intersection plaza in the SPIC problem and impose constraints to avoid collision between vehicles and the plaza's boundaries.
  - Develop concrete and a well-posed nonlinear optimal control problem for the SPIC problem.
- **Solution method**:
  - Develop an efficient near-optimal method for the generation of feasible and collision-free trajectories that utilizes the Finite Fourier Series (FFS) or Bézier curves for solving SPIC problem.
  - Develop an algorithm based on an extension of the standard nonlinear model predictive contouring control, called ENMPCC, for solving SPIC problem, which results in optimal and collision-free trajectory generation of CVAD in an intersection plaza.
  - Present linear time-varying (LTV) modeling framework, called LTV-ENMPCC, to reduce computational effort needed to solve SPIC problem, which leads to a more scalable solution strategy.

- **Simulation**:
  - Develop the SPIC simulator (SPIC-Sim), a simulation modeling tool that allows for the modeling of flexible utilization of urban intersections. SPIC-Sim enables the investigation of both conventional path-based (lane-based) and path-free (lane-free) traffic control scenarios.
  - Realistic simulation of traffic scenarios involving both straight and turning movements and with the continuous arrival of vehicles from all possible directions in the intersection plaza.
  - Comparison of the SPIC problem with the conventional path-based (lane-based) driving approach at the intersections.

1.3 TERMINOLOGY AND DELIMITATION

In the literature, there are a variety of names for the type of vehicle considered in this work; autonomous vehicles, driverless vehicles, connected vehicles, cooperative vehicles, unmanned vehicles, automated vehicles, connected and automated vehicles, and robotic vehicles, among others. For a comprehensive discussion on this subject, we recommend referring to the work by Shladover (2018).

With the aim of establishing consistent terminology and comprehension of automation systems for on-road motor vehicles, SAE International, more recently with the participation of ISO, has maintained since 2014 a document that establishes a taxonomy and definitions of terms in this domain (INTERNATIONAL, 2021). All terms listed in the previous paragraph do not meet the SAE International taxonomy and definitions. Any reference to automation must be made to driving and not to the vehicle. For more detail, it is recommended to review Section 7.1 of the document prepared by International (2021). Nonetheless, the SAE International publication still presents a gap that doesn't align with the type of vehicle in this work. The aspect of communication and cooperation between vehicles or with the infrastructure remains unaddressed.

In this thesis, the vehicles are referred to as *Connected under Automated Driving* (CVAD). The term "connected" explicitly indicates the potential or requirement for communication. The term "automated driving" serves SAE International by linking automation to driving and not to the vehicle. In this sense, for this work, according to the SAE International classification of six levels (0 to 5) of driving automation, a driving automation system of level 4 or higher is considered.

Moreover, in the literature, there are two names for the type of intersection considered in this work; lane-free intersection (LI, B. et al., 2018; AMOUZADI; ORISATOKI; DIZQAH, 2022) and continuous free space (LI, B. et al., 2020). We suggest reading (SEKERAN et al., 2022) for a discussion on the subject. In this thesis, as we focus on urban intersections, we adopt the terminology *path-free intersection*, in which the intersection transforms into a boundary-

constrained free space, where free paths are permitted on the condition that the resulting trajectories do not collide.

While the proposed formulation of the SPIC problem is quite versatile and capable of modeling any arbitrary road layouts, this thesis primarily focuses on an urban intersection as an illustrative example in simulations.

The scope of the method can be broadened to encompass a network of intersections, enabling its application beyond single intersections. The conceptualization of the connection between two intersections as a lane-free traffic roadway, as proposed by Papageorgiou et al. (2021) and Malekzadeh, Papamichail, and Papageorgiou (2021), offers a promising avenue for further exploration.

Connected vehicles are considered capable of communicating with each other and also the infrastructure. Communication is considered to be ideal. The time for collecting data from all vehicles, processing, and calculating the control is considered instantaneous.

1.4 THESIS OUTLINE

The remainder of the thesis is laid out as follows. Chapter 2 briefly covers the main background and the basic principles used throughout this thesis. In this chapter, concepts of automated driving and connected vehicles are introduced, followed by a brief literature review on CVAD coordination at intersections. The trajectory optimization problem is then presented, along with explanations of optimal control and model predictive control techniques. Important definitions and theoretical results from convex optimization are summarized. Finally, the text discusses the collision avoidance problem. Chapter 3 introduces the Signal-free Path-free Intersection Control (SPIC) problem which promotes better use of the intersection space to improve overall traffic efficiency by leveraging the potential of the CVAD. Chapter 4 introduces the intersection trajectory optimal control problem (ITOP), viewed as a specific instance of the SPIC problem. The primary goal of the ITOP is to determine vehicle trajectories while satisfying plaza boundaries and collision avoidance constraints. To achieve this, we present two shape-based methods, relying on finite Fourier series (FFS) and Bézier curves. These techniques convert ITOP into a nonlinear programming (NLP) problem that can be solved by an NLP solver. Chapter 5 introduces a new formulation for the SPIC problem. We build on a model predictive control (MPC) suitable for trajectory optimization/planning, known as nonlinear model predictive contouring control (NMPCC). In fact, based on an extension of the standard NMPCC, which refers to the ENMPCC method, a solution for solving the SPIC problem is proposed. Finally, Chapter 6 concludes this thesis, with some final remarks and comments on future research.

We note that we benefited from the assistance of ChatGPT in revising the text of this thesis. Around 5 percent of the text was revised using ChatGPT.

2 BACKGROUND

This chapter provides a brief introduction to all the basic principles that will be used throughout this thesis. Sections 2.1 and 2.2 present the concepts of automated driving and connected vehicles, respectively. Section 2.3 is a brief literature review on CVAD coordination at intersections. Section 2.4 presents the trajectory optimization problem. Section 2.5 and Section 2.8 explain the optimal control and model predictive control techniques, respectively. Section 2.9, summarizes several important definitions and theoretical results from convex optimization, which plays a crucial role in optimization theory and methods. Finally, Section 2.10 discusses the collision avoidance problem.

2.1 AUTOMATED DRIVING

Automated driving, refers to the operation of vehicles in which the driving tasks are performed by an automated system or technology, rather than relying solely on human drivers. In automated driving systems, a combination of sensors, cameras, radar, lidar, and advanced algorithms are employed to perceive the vehicle's surroundings, make decisions, and control the vehicle's movements. These systems can handle various aspects of driving, including acceleration, braking, steering, and navigation, with limited or no human intervention. The goal of automated driving is to enhance safety, efficiency, and convenience in transportation by reducing human errors and enabling more efficient traffic flow.

The SAE International has defined six levels of automation for driving systems, commonly known as the SAE levels of automation (INTERNATIONAL, 2021). These levels provide a framework for categorizing the capabilities and responsibilities of automated driving systems. Here is a brief overview of each level:

- Level 0 (No Automation): the driver is in complete control of all driving tasks. There is no automation present.
- Level 1 (Driver Assistance): certain driving tasks, such as steering or acceleration/deceleration, can be automated by the vehicle. However, the driver remains responsible for overall vehicle control and must actively monitor the driving environment.
- Level 2 (Partial Automation): the vehicle can simultaneously control multiple driving tasks, such as steering and acceleration, under certain conditions. The driver is still required to remain engaged and monitor the driving environment, ready to intervene if needed.
- Level 3 (Conditional Automation): the vehicle is capable of managing most aspects of the driving tasks under specific conditions. The driver is allowed to disengage from actively monitoring the driving environment but must be ready to take control when prompted by the system.

- Level 4 (High Automation): the vehicle can perform all driving tasks under predefined conditions and does not require driver intervention. However, the system's operational domain may be limited to specific geographic areas or road types.
- Level 5 (Full Automation): the vehicle is capable of performing all driving tasks under any condition that a human driver could handle. No human intervention is required or expected.

2.2 CONNECTED VEHICLES

Connected vehicles refer to vehicles that are equipped with advanced communication technologies, allowing them to connect and exchange data with other vehicles, infrastructure, and various external sources. These vehicles use wireless communication systems, such as Dedicated Short-Range Communication (DSRC) or Cellular Vehicle-to-Everything (C-V2X), to establish connectivity (SHLADOVER, 2018).

Connected vehicles encompass various communication paradigms, including Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), and Vehicle-to-Everything (V2X) interactions (SHLADOVER, 2018). These interactions leverage advanced wireless communication technologies to enable seamless data exchange between vehicles, roadside infrastructure, and other entities (SIEGEL; ERB; SARMA, 2018). They can exchange information about their location, speed, acceleration, and other relevant data. This communication enables the vehicles to share real-time information, such as traffic conditions, road hazards, and other relevant updates, enhancing situational awareness and safety. In the context of this thesis, vehicle connectivity refers to the capability of automated vehicles to exchange with each other real-time information about their current states (position and speed), and also the intersection geometry, in an asynchronous mode.

2.3 CVAD COORDINATION AT INTERSECTIONS

CVAD coordination at signal-free intersections is a cutting-edge approach that leverages advanced technologies and artificial intelligence to enhance the efficiency, sustainability, and safety of road intersections. Traditional traffic control systems, such as traffic lights and stop signs, often lead to congestion, delays, and accidents due to limited coordination and adaptation capabilities. In contrast, CVAD coordination at intersections employs an appropriate framework where CVAD communicate with each other and the infrastructure to make real-time decisions. By dynamically adjusting their speed, trajectory, and timing, these vehicles can seamlessly navigate intersections without the need for traditional traffic control mechanisms. The implementation of CVAD coordination at signal-free intersections has shown promising results in reducing traffic congestion, enhancing traffic flow, minimizing fuel consumption, sustainability, and improving overall road safety (ZHANG, Y.; CASSANDRAS, 2019). Moreover, the integration of CVAD coordination with smart city infrastructure and emerging technologies, such as V2V and V2I

communication, has the potential to further optimize intersection management, leading to more sustainable and efficient transportation systems (AHMED et al., 2022).

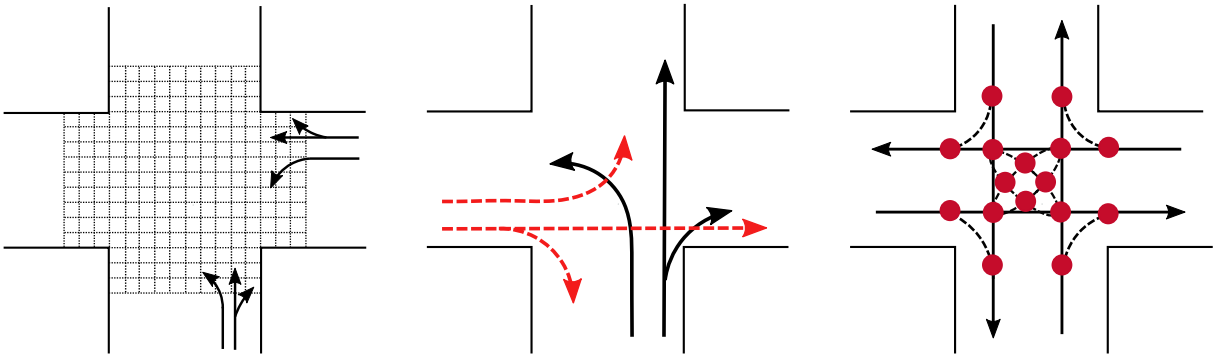
The proposed approaches for modeling intersections thus far can be summarized in four categories (CHEN, L.; ENGLUND, 2016; MÜLLER, 2018):

- Cells: the intersection is discretized in space and time, being divided into cells while time windows in each cell are allocated to vehicles so that there are no collisions. Figure 1a shows an intersection divided into cells.

  Dresner and Stone (2004) proposed a model based on space and time discretization. The authors also extended this control scheme in Dresner and Stone (2008). They divided the intersection in a cell grid and proposed a reservation-based mechanism based on a first-come first-served (FCFS) policy. The FCFS enables the vehicles to reserve in advance the required space-time so that once the reservation was fulfilled, the vehicles kept the same speed to cross the intersection. In fact, the approaching vehicles send requests to the intersection controller to reserve spaces and time slots within the intersection area. Reservation requests can be accepted or rejected based on their conflicts with previous reservations. Moreover, the booking agent is capable of suggesting new, free routes for vehicles that have their requests denied, to avoid a large number of requests. A similar approach was followed by Schepperle and Böhm (2009) who proposed a valuation-aware traffic control system in which vehicles negotiate their time slots with each other and then adapt their speed according to the time slot obtained. In any case, the vehicles followed predefined paths.

- Path-based: a limited set of paths are allocated to vehicles in such a way that vehicles on conflicting paths do not collide. In general, the paths are defined based on the traffic engineering concept of vehicle movements. Figure 1b illustrates some possible movements performed by vehicles in an intersection.

  Lee and Byungkyu Park (2012) introduced an optimization problem that minimizes the overlap of vehicles with potential transversal collisions between connected vehicles at an intersection. The problem is solved in parallel using genetic algorithms and solvers. Simulation results showed that the performance of the proposed method is better than that of traffic lights. However, their solution approach offers no guarantee of performance. Yu et al. (2019) proposed a mixed-integer linear program (MILP) model to cooperatively optimize the trajectories of CVAD along a signal-free corridor. They modeled the interactions of vehicle trajectories at the microscopic level, by considering the coordination between vehicle trajectories. Nonetheless, the vehicle paths are assumed fixed and they are not optimized. Moreover, at higher demand levels the computational burden of the proposed method is heavy. As a conclusion, considering the predefined paths inside an intersection simplifies the problem in such a way that recognizing potential collisions is not a challenging task.



(a) (b) (c)

Figure 1 – Intersection modeling: (a) cells; (b) paths; and (c) collision regions/points (red circles) (adapted from Lei Chen and Englund (2016))

- Conflicting regions/points: roughly a combination of the previous two, in which only the points or regions where conflicts between paths occur are discretized, and time windows are allocated to vehicles to pass through these points. Figure 1c illustrates an intersection with highlighted conflict regions.

  One example of this modeling approach is proposed by Feng Zhu and Ukkusuri (2015), who formulated a linear programming problem with an integer solution so that traffic flows pass safely through the conflict points of an isolated intersection. Mirheli et al. (2019) presented a modeling strategy that prioritizes conflicting movements at a signal-free intersection in a distributed manner. The method converges to a near-optimal solution using an iterative framework, through solving a mixed-integer nonlinear programming problem. The iterative process continues so that CVAD solve their trajectory planning problem and update their solutions until they reach a consensus. Another work for this strategy proposed by Levin and Rey (2017), in which the intersection manager assigns optimal reservations to CVAD through solving a MILP. The passing order and intersection entrance time are simultaneously derived for all the CVAD. However, neither vehicle trajectory coordination nor lane-changing behaviors were taken into consideration. Müller, Carlson, and Kraus (2016) considered an intersection with the conflict regions and proposed an optimal arrival time scheduling (OATS) for the CVAD at a single intersection. OATS is modeled as a centralized MILP problem that guaranteed minimum aggregate arrival time and the absence of vehicle collisions. First, the location and speed of the CVAD in each time step are gathered to provide the shortest and longest travel times to reach an intersection. Then, the aggregated arrival time for all vehicles is minimized and the speed profiles are assigned to vehicles to approach the intersection.

- Path-free: the vehicles are allowed to make full use of the intersection space via the definition of their trajectories for any possible path as long as the resulting trajectories do not conflict with each other. The research on this subject is rather limited. Bai Li

et al. (2018) proposed a centralized optimal control problem (OCP) for a number of CVAD crossing an intersection without using predefined paths, which is solved using a two-stage strategy. The first stage is performed online and guides the CVAD to a standard formation before they enter the intersection. In the second stage, the CVAD in formation cross the intersection according to optimized trajectories that were computed offline. To handle collision-free constraints, They assumed that each CVAD and each street block of the intersection is rectangular. When they expected two rectangles not to collide, all four edge points on one rectangle should remain outside the other rectangular region. As the authors mentioned, the computation complexity of this method is too high, so that for 24 CVAD, if 10000 computers work simultaneously, 358.2 years would be needed to build a complete standard case.

Bai Li et al. (2020) simplified the collision-avoidance constraints, a couple of circles are used to approximate the rectangular street block, and two equal-sized discs are utilized to cover the rectangular CVAD. To solve the problem, a centralized OCP is formulated and then solved numerically. They proposed an algorithm for providing suitable trajectories as initial guesses for the nonlinear programming formulation, speeding up the solution. These works evolved into a batch-processing framework for autonomous intersection management that is an integration of planning and reservation methods (LI, B. et al., 2021). The former handles the batches macroscopically, and the latter optimizes the cooperative trajectories in one batch microscopically. A minimum-time nonlinear OCP formulation was also attempted (AMOUZADI; ORISATOKI; DIZQAH, 2022). The non-convex collision avoidance constraints were reformulated by the use of the dual problem theory. This approach reduced the intersection crossing time, but the scale of collision avoidance constraints made the method computationally expensive. These nonlinear OCP-based methods do not take into account the continuous arrival of vehicles, are conducted offline, require prior knowledge of the initial and final states of the CVAD, and impose the same travel times on all CVAD.

A strikingly different approach was proposed by Roca et al. (2016, 2020), who relied on emergent behaviors by modeling vehicle behavior based on a hierarchical set of rules, similar to the ones used for modeling flocks. The simplicity of the approach results in low computation complexity but also in high sensitivity to minor changes in the rules. Finally, Neto and Joannis (2021) used formal methods to model the behavior at intersections to obtain safe trajectories within the intersection space. The nature of the formulation and employed methods leads, however, to a combinatorial explosion of the states even for small instances.

2.4 TRAJECTORY OPTIMIZATION

The term trajectory optimization for motion planning refers to a set of methods that are used to find the best choice of trajectory, that respects the system's dynamics between an initial and a final state, taking into account a set of constraints (BETTS, 1998; KELLY, 2017). It is a special type of optimization problem where the decision variables are functions, rather than real numbers. In fact, this problem generates a set of collision-free and feasible trajectories and selects an optimal trajectory from the set based on optimization of the objective function.

Trajectory optimization is sometimes referred to as motion planning and incorrectly as path planning; trajectory optimization differs from path planning since it is parametrized by time (KATRAKAZAS et al., 2015). Generally, path planning generates a sequence of configuration states (way-points) in space taking into account safety, comfort, and accuracy. The path planner does not answer how to move along the path based on the time, speed, and kinematics of vehicles (GALCERAN; CARRERAS, 2013). On the other hand, trajectory optimization is a real-time transition of a vehicle from one feasible state to the next feasible state taking into account the vehicle's kinematic constraints (speed constraint, acceleration constraint, etc.) and it must produce safe, human-like, and human-aware trajectories in a wide range of driving scenarios (SHARMA; SAHOO; PUHAN, 2021).

The trajectory optimization problem has been well studied in the field of robotics (TAZAKI; MUROOKA, 2020), astrodynamics (SHIRAZI; CEBERIO; LOZANO, 2018), unmanned aerial vehicles (LIU; BUCKNALL, 2018), and underwater vehicles (YOUAKIM; RIDAO, 2018). Inspired by earlier studies on motion planning of robot vehicles in other contexts (KATRAKAZAS et al., 2015; SHARMA; SAHOO; PUHAN, 2021) and driven by rapid implementations of communication technologies (MELSON; MA, 2021), studies on CVAD trajectory optimization in the road traffic context have received attention. They mostly focus on the non-intersecting roads and freeways for connected and non-connected vehicles (MAKANTASIS; PAPAGEORGIOU, 2018; TYPALDOS; PAPAGEORGIOU; PAPAMICHAIL, 2021). We omit the details of these works to maintain the focus of this chapter since we do not treat the same problem as stated herein. On the other hand, intersection management strategies often involve trajectory optimization to some degree. Trajectory optimization remains complex for vehicular traffic at intersections, for instance, the risk of collision is increased at the intersections compared to the motion of the CVAD along a freeway.

2.5 OPTIMAL CONTROL PROBLEMS

The trajectory optimization in road intersections calls for sophisticated approaches, such as optimal control methods (LEWIS; VRABIE; SYRMOS, 2012; PADEN et al., 2016). Optimal control theory formulations have the benefit of obtaining a control input that minimizes a given cost function while satisfying the dynamic constraints of the vehicles explicitly. There are two general techniques for solving trajectory optimization problems: direct and indirect methods

(RAO, 2009). The indirect methods consider the dualized form of the equations including states and co-states through the application of Pontryagin's maximum principle (SAERENS; DIEHL; VAN DEN BULCK, 2010). Indirect methods depend strongly on the accuracy of the initial guess and also double the size of the problem by introducing co-states that are not physically intuitive. The direct methods, on the other hand, convert the continuous optimal control problem into an NLP problem, considering different schemes of control and states parameterization (BETTS, 2010). Although direct methods are less accurate than indirect methods, they are usually preferred due to the following reasons (CONWAY, 2012): (i) reduction of the problem sensitivity to initial guesses, (ii) ease of the numerical implementation, (iii) larger domain of convergence, and (iv) availability of efficient numerical packages and solvers.

One type of direct method is the one in which only the state variables are interpolated and control variables are considered in the objective function. This method is sometimes referred to as the shape-based (SB) method since it depends on the shape of the state variables (PETROPOULOS, Anastassios E; LONGUSKI, 2004; GONDELACH; NOOMEN, 2015). The SB methods assume the trajectory respects a certain shape (which can be fully described analytically), and the parameters defining this shape are then computed to ensure the trajectory satisfies some boundary conditions (states at departure and arrival, travel time, and so on). The analytical formulation reduces the computational load significantly, which is a major advantage of the SB methods over the traditional direct and indirect methods. Many kinds of SB methods have been proposed by researchers (PETROPOULOS, Anastassios Evangelos, 2001; ZENG; GENG; WU, 2017). Although there exist multiple SB methods proposed in the literature, the Fourier series and Bézier curve are widely utilized within this kind of method (TAHERI; ABDELKHALIK, 2012; CHOE, 2017; MINGYING et al., 2020), due to (i) fast computation speeds, and (ii) they do not assume a specific shape for the trajectory; rather, it assumes an approximation for the trajectory shape in terms of the FFS or Bézier expansion of the states. For every different selection of the Fourier or Bézier coefficients, a different shape is obtained.

In the context of spacecraft trajectory generation, FFS and Bézier methods are among the methods used for the rapid generation of feasible trajectories, introduced by Taheri and Abdelkhalik (2012) and Fan et al. (2020). These methods provide approximated solutions to the trajectory optimization problem of dynamical systems. In contrast to the spacecraft case, in urban traffic systems, road vehicles require different considerations: (i) in the case of road vehicles, the trajectories are limited to the geometry of the intersection while in the spacecraft the trajectories are not necessarily limited to any predefined geometry; (ii) the coordination system in an intersection is two-dimensional (Cartesian) while spacecraft use a three-dimensional system (Cylindrical or Spherical); (iii) in the case of intersection management, each vehicle has its own completion time, while the flight time of the spacecraft trajectories are a fixed value; and (iv) it is not reasonable to assume the initial and final speeds of vehicles in an intersection to be zero which is valid for spacecraft.

2.6 CONTOURING CONTROL

In contouring control, it is not necessary to track a time-parameterized reference trajectory accurately, as long as the system follows the reference path (KOREN; LO, 1992). In such problems, the contouring error is used to denote the error component orthogonal to the reference trajectory, Contouring error is the deviation of the cutter position of a controlled object/tool from a reference path while the tracking error is the difference between the predefined reference path and measured system output. Figure 2 shows the difference between these errors. The point R shows the value of the reference at the current time, while P is the actual object/tool position. Point A is the point on the reference path closest to P. The contouring errors in the x and y axes, ϵ_x and ϵ_y, are measured from point A, while the x and y components of the tracking error, e_x and e_y, are measured from the point R.

Cross-coupled control, introduced by Yoram Koren (1997), seeks to minimize contouring errors instead of tracking errors. This is accomplished by estimating the contouring error of the x and y axes, ϵ_x and ϵ_y, based on the tracking error, e_x and e_y, using a contouring error model. A feedback control law is designed to minimize these contouring errors, and the output of the contouring error controller is added to the control inputs of the individual axes. Several other control approaches aim to minimize the contouring error rather than only tracking error. McNab and Tsao (2000) implements receding horizon linear quadratic control using a cost function penalizing contouring error as well as tracking error and control effort. A similar approach using model predictive control is proposed by El Khalick M. and Uchiyama (2011), where separate weighting factors are applied to the errors orthogonal and tangential to the desired path. Contouring error controllers attempt to address the path-following control problem indirectly by placing more emphasis on deviation from the path rather than tracking error. However, these approaches have no ability to affect the path evolution.

2.7 PATH FOLLOWING CONTROL

In trajectory planning/tracking approaches, the reference path is converted to a time-dependent reference trajectory using an appropriate trajectory planning technique. Trajectory tracking controllers are then employed to track the reference trajectory. A general architecture for these approaches is provided in Figure 3. In contrast, path-following controllers determine the evolution of the reference path and the plant inputs simultaneously using available feedback. Such controllers address the path-following control problem directly. A general architecture for path-following approaches is shown in Figure 4. The distinction between tracking and path-following is important to consider. Tracking requires each object or tool to follow a time-parameterized reference trajectory governed by a trajectory planner or path governor (GILBERT; KOLMANOVSKY, 2002; SUSANU; DUMUR, 2006) while minimizing tracking error. Path-following, on the other hand, only requires an object or a tool to travel along a reference path, which is not time-parameterized, without the need to track it precisely.

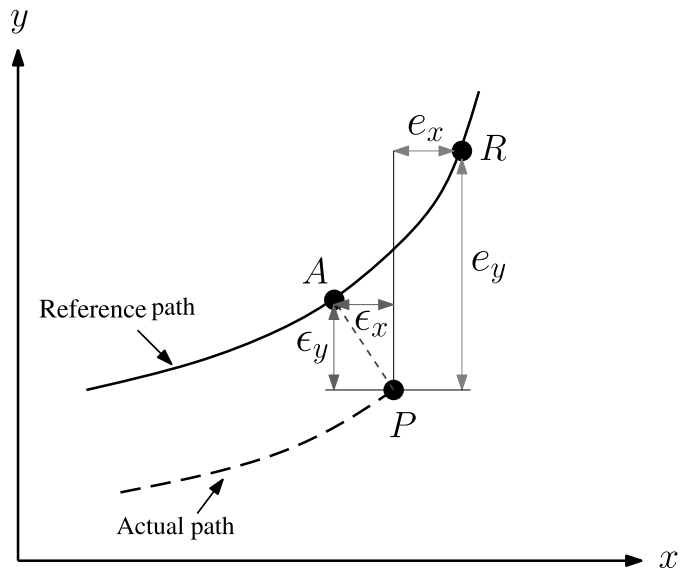


Figure 2 – Difference between contour error and tracking error (KOREN, Yoram, 1997); ϵ_x and ϵ_y are the x and y components of the contouring error while e_x and e_y are the x and y components of the tracking error.

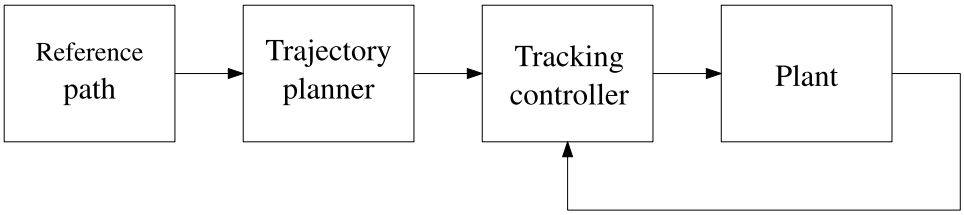


Figure 3 – Conventional architecture of the trajectory planning and tracking controller (LAM, 2012).

Dačić and Kokotović (2006) introduced a path-following controller for linear non-minimum phase systems where a feedback law is designed for the path parameter θ to stabilize the zero dynamics. Dacic, Nesic, and Kokotovic (2007) extended the approach to nonlinear systems with unstable zero dynamics, where only practical convergence to the path is guaranteed. Hauser and Hindman (1995) proposed a method to convert trajectory tracking controllers to path-following controllers for linearizable feedback systems using an appropriate mapping from the current state to the path parameter θ. Path-following control schemes also arise in wheeled mobile robots (SAMSON, 1992), which rely on computing a projection of the robot onto the reference path. These path-following control approaches discussed thus far do not consider any constraints on the system. Kanjanawanishkul, Hofmeister, and Zell (2009) combined path-following control with trajectory tracking control for mobile robots using non-linear model predictive control. Moreover, Faulwasser, Kern, and Findeisen (2009) proposed model predictive path-following control (MPFC) for general nonlinear continuous-time systems. Following a model predictive control framework, a cost function is minimized, subject to constraints, at each sample. In contrast to tracking MPC, The MPFC approach determines the plant inputs and path parameter evolution simultaneously. As a receding horizon structure is used in MPFC, feedback may be taken into account at each time step, while the cost function may be selected to reflect

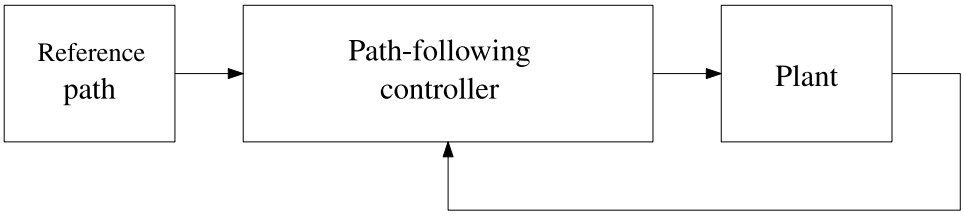


Figure 4 – Conventional architecture of the path-following controller (LAM, 2012).

desired control objectives such as minimizing traversal time or contouring error.

2.8 MODEL PREDICTIVE CONTROL

Conventional methods for trajectory optimization and motion planning, such as feedback controllers (AGHA-MOHAMMADI et al., 2011) or iterative linear quadratic regulators (LQR) (NAGARIYA; SARIPALLI, 2020), often involve decoupling lateral and longitudinal dynamics and may not fully consider state and actuator constraints. In this regard, model predictive control (MPC) is a viable alternative to these approaches (ROSSITER, 2018; SCHWENZER et al., 2021). In MPC, the goal is to use a model of the system to predict the system's output for a number of time steps in the future while minimizing an objective function that defines the distance between the predicted outputs and some given desired outputs. This problem is a reference tracking problem since at each sampling time the outputs of the system are forced to track a reference signal at a certain time. A key advantage of the MPC is that the constraints can be explicitly considered in the optimization problem. By solving the corresponding optimization problem, a sequence of control inputs is obtained over part or the whole time horizon. However, only the first control input is applied to the system. In the next time step, the whole process is repeated for the computation of the next control input with the horizon shifted one step ahead in time. Thus, the system operates in a receding horizon fashion. The number of time steps ahead utilized for prediction is called the prediction horizon.

Nonlinear model predictive contouring control (NMPCC) (LAM; MANZIE; GOOD, 2010) is a control strategy that combines the principles of path-following control (FAULWASSER; KERN; FINDEISEN, 2009) with the concept of contouring control (KOREN; LO, 1992) and model predictive control (ROSSITER, 2018). NMPCC contains an analytical description of the reference paths, parameterized with a tailored path parameter instead of time. The objective of NMPCC is to minimize the projected distance between the current position of a tool or object and a reference path while maximizing its progress along the path. NMPCC offers the advantage of combining both path planning and path tracking in a single nonlinear optimization problem through the application of contouring control. In contrast to the state-of-the-art motion planning techniques that involve a two-level optimization process, where an upper-level optimizer generates time-dependent trajectories and a lower-level optimizer tracks the generated trajectory accurately (GILBERT; KOLMANOVSKY, 2002; LI, N. et al., 2017), the NMPCC combines these levels into a single optimal control problem. While NMPCC has been applied to drone

racing (ROMERO et al., 2022), car racing (LINIGER; DOMAHIDI; MORARI, 2015), parallel autonomy for self-driving cars (SCHWARTING et al., 2017), and winding roads (LEVY; HADDAD, 2021, 2022), this work applies NMPCC to a complex driving scenario, signal-free path-free urban intersection, which is expected to be more challenging.

2.9 CONVEX OPTIMIZATION AND DUALITY

In this section, the basic terminology of the convex optimization problem is discussed (BOYD, S. P.; VANDENBERGHE, 2004; BERTSEKAS, 2009). The optimality conditions of convex problems are also discussed. The general form of a convex optimization problem is:

$$\begin{aligned} &\text{minimize } f_0(x) \\ &\text{subject to } g_i(x) \leq 0, \forall i = 1, \ldots, o \\ &\qquad\qquad a_i^T x = b_i, \forall i = 1, \ldots, p \end{aligned} \tag{1}$$

where $f_0 : \mathcal{R}^n \to \mathcal{R}$ is a convex function called the objective function, n is the dimension of decision variables, $g_i : \mathcal{R}^n \to \mathcal{R}, i = 1, \ldots, o$ define the inequality constraints which are all convex functions, and finally $a_i^T x = b_i, \forall i = 1, \ldots, p$ define the equality constraints which are affine functions (and therefore also convex). In summary, a convex optimization problem consists of minimizing a convex function over a convex set. If the objective function is zero, then the problem is called a feasibility problem. Regarding problem (1), some important definitions and terminologies apply and are discussed in the following.

Definition 1 *A point $x \in \mathcal{R}^n$ is called feasible if it satisfies all equality and inequality constraints. A set of feasible points is called the feasible set. Accordingly, problem* (1) *is called feasible if the feasible set is not empty.*

Definition 2 *A point $x^* \in \mathcal{R}^n$ is called the optimal point if it is feasible and minimizes the objective function f_0. An optimal point is called global if it is optimal in the entire feasible set and it is called local if it is optimal in a feasible neighborhood of x^*.*

Definition 3 *The optimal value, p^*, of problem* (1) *is defined as*

$$p^* = \min\{f(x) \,|\, g_i(x) \leq 0, i = 1, \ldots, o, a_i^T x = b_i, i = 1, \ldots, p\}.$$

Proposition 1 *For any convex problem in the form of* (1) *every local optimal point is also global.*

After discussing the general form of convex optimization problems, the next subsection will introduce the duality theory pertaining to these problems.

The basic idea in Lagrangian duality is to take the constraints into account by augmenting the objective function with a weighted sum of the constraint functions. Considering problem (1),

the Lagrangian Function $L : \mathcal{R}^n \times \mathcal{R}^o \times \mathcal{R}^p \rightarrow \mathcal{R}$ is defined as:

$$L(x,\lambda, s) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) + \sum_{i=1}^{p} s_i(a_i^T x - b_i) \tag{2}$$

where, $\lambda_i(s_i)$ is the Lagrange multiplier associated with the i-th inequality (equality) constraint. The vectors, λ and s are called dual variables associated with problem (1).

Proposition 2 *For any feasible point $x \in \mathcal{R}^n$, the Lagrangian function induces a lower bound on the objective function if λ_i is a positive number.*

In order to find the best lower bound, the Lagrangian dual function $q : \mathcal{R}^n \times \mathcal{R}^p \rightarrow \mathcal{R}$ is constructed as follows:

$$q(\lambda, s) = \min_x L(x, \lambda, s) \tag{3}$$

Proposition 3 *The dual function $q(\lambda, s)$ is a concave function and hence, a global maximum point exists.*

Proposition 4 *For any $\lambda \geq 0$ the dual function yields a lower bound on the optimal value p^* of problem* (1)*, i.e.,*

$$q(\lambda, s) \leq p^*. \tag{4}$$

The lower bound provided by the dual function is not necessarily the best dual bound. According to the concavity property of the dual function, the best lower bound is obtained by maximizing the dual function. This is another optimization problem, which is called the dual problem is defined as:

$$\underset{\lambda,s}{\text{maximize}} \;\; q(\lambda,s) \tag{5}$$

$$\text{subject to} : \lambda \geq 0 \tag{6}$$

The optimal value of the dual problem, d^*, is the best lower bound on p^*. In particular, we have

$$d^* \leq p^*.$$

The difference $p^* - d^*$ is called the duality gap. In general and for general nonlinear problems (not necessarily convex) the duality gap is usually nonzero. However, for convex problems and under certain conditions, called Slater's condition, the duality gap is zero, i.e.,

$$d^* = p^*.$$

The former and the latter case are called weak and strong duality, respectively.

By using the concept of dual problem and strong duality, the necessary and sufficient conditions (KKT conditions) of optimality for the convex problem can be stated in the following proposition.

Proposition 5 *Consider the convex problem* (1) *and assume the strong duality holds. Then* x^* *is the optimal point if the following conditions are satisfied:*

$$
\begin{aligned}
& g_i(x^*) \leq 0, \;\; i = 1, \ldots, o, \\
& a_i^T x^* = b_i, \;\; i = 1, \ldots, p, \\
& \lambda_i^* \geq 0, \;\; i = 1, \ldots, o, \\
& \lambda_i^* g_i(x^*) = 0, \;\; i = 1, \ldots, o, \\
& \nabla_x L(x^*, \lambda^*, s^*) = 0.
\end{aligned}
$$

In summary, for any optimization problem with differentiable objective and constraint functions for which strong duality holds, any pair of primal and dual optimal points must satisfy the KKT conditions.

2.10 COLLISION AVOIDANCE

Collision avoidance is an integral part of motion planning and trajectory optimization and plays a crucial role in ensuring safety and mitigating potential risks in various domains. Whether in transportation, robotics, or any autonomous system involving moving objects, collision avoidance is of utmost importance (BERNTORP, 2017; HUANG; TEO; TAN, 2019; VAGALE et al., 2021). By actively detecting and predicting potential collisions, and taking preventive measures, collision avoidance systems can help prevent accidents, minimize damages, and save lives. The ability to anticipate and avoid collisions not only enhances the safety of vehicles and machinery but also improves overall operational efficiency. Collision avoidance in robotic motion planning refers to the process of ensuring that a planned trajectory is safe and collision free (MINGUEZ; LAMIRAUX; LAUMOND, 2016). This can include being free of collisions between the controlled object and static obstacles in its environment, collisions between the controlled object and other objects or dynamic obstacles in motion, as well as self collisions between different parts of the controlled object itself. From CVAD navigating busy roads to industrial robots working alongside humans, collision avoidance technology is a fundamental component that enables safe and reliable operations. Emphasizing the development and implementation of effective collision avoidance strategies is paramount to creating a safer and more sustainable future (MINGUEZ; LAMIRAUX; LAUMOND, 2016).

Many effective methods exist to solve the problem of collision avoidance in path/trajectory planning in the fields of unmanned aerial vehicles, robots, and CVAD vehicles. Earlier works considered the obstacle avoidance problem in a continuous time framework using dynamic optimization (SUNDAR; SHILLER, 1995; HAGENAARS; IMURA; NIJMEIJER, 2004). The set of states that can be steered to a target set, while satisfying bound constraints and avoiding obstacles, is a level set of the value function of the dynamic optimization problem, obtained by solving a Hamilton-Jacobi–Bellman equation. Another approach is to include an additional potential function for avoiding obstacles in the cost function, thus converting the collision

avoidance problem into an unconstrained optimization problem allowing gradient-based solvers (KIM; SHIM; SASTRY, 2002). However, the construction of potential functions in a general framework is difficult. Other techniques include the A^* method, the V-graph method, and the Rapidly-exploring Random Trees method (AGGARWAL; KUMAR, 2020; MAC et al., 2016). These methods usually do not consider dynamic models, or convert the model and constraints into simple geometric constraints, and then obtain geometric paths connecting the start and terminal points, meeting obstacle avoidance constraints and other constraints through different ideas.

Machine learning techniques can be used to learn collision avoidance policies from data. This approach involves training a model on a dataset of collision-free trajectories and then using the model to predict a safe trajectory for the robot (VIRDI, 2018). Deep learning-based pure planning (HONG; SAPP; PHILBIN, 2019) or joint planning (EVERETT; CHEN, Y. F.; HOW, 2018) and prediction approaches learn typical distributions of interaction and motion patterns and, hence, are supposed to yield collision-free and human-like trajectories. Like for many problems, these learned approaches are arguably best in average solution quality even at solution times required by automated driving. However, solutions that can guarantee feasible or safe trajectories are still in their infancy (BROSOWSKY et al., 2021).

To address the issue of collision avoidance, an approach that can be adopted involves the formulation of a trajectory optimization problem that incorporates a dynamic model, performance index, and diverse constraints. This methodology takes into account the perspectives of optimal control and collision avoidance concerns, and subsequently solves the optimization problem using either analytical or numerical methods (SHIRAZI; CEBERIO; LOZANO, 2018). Considering optimization-based collision avoidance, the design of a collision avoidance algorithm is influenced by various factors. Ericson (2004) categorizes these factors into different aspects, including the representation of the application domain, the types of queries, parameters of the simulation environment, performance considerations, robustness, as well as ease of implementation, and use. To maintain conciseness and prioritize the ideas employed in this thesis, the remainder of this section will focus on the first three factors.

The first factor, application domain representation, refers to the geometrical representation used for the bodies of the controlled objects. It can be thought of as the process of choosing which shape we want to use to model the area occupied by controlled objects. The chosen geometric representations have a direct influence on the algorithms that have to be utilized for solving the optimization problem. The second factor refers to the different types of queries and the nature of the queries we aim to pose to the collision detection system. To illustrate, we might seek to address whether multiple objects intersect or collide at a specific moment in time. Alternatively, we may inquire about potential approaches to resolve a collision between two objects. The objective of these inquiries is to determine whether there is any overlap between two or more objects, taking into account their positions, velocities, and orientations relative to a coordinate frame at a specific moment in time. The ability to address these questions forms the

foundation of any collision avoidance algorithm. In the context of automated system trajectory optimization/planning, our primary focus lies in ascertaining if two objects intersect and, if so, devising a strategy to prevent them from colliding. Finally, the third factor refers to the environment simulation parameters, i.e., to the several parameters in the simulation that have a direct impact on the collision avoidance algorithm like the number of controlled objects, their relative sizes, and positions.

Collision avoidance can be viewed in terms of a distance formulation since it implies that the distance between every two controlled objects should be greater than the minimum safe distance. In various studies, static/dynamic obstacles are modeled using circles, cylinders, spheres, or cones (ZHAO; ZHOU, 2013; DAM, 2019; BROSSETTE; WIEBER, 2017). In this scenario, a collision is deemed to occur if the Euclidean distance between the objects is less than a specified minimum safe distance. Representing the problem in this manner transforms the optimization problem into a non-convex NLP problem. Although NLP solvers offer flexibility, they can be computationally demanding and challenging for solving collision avoidance problems. To address these challenges, linearization, and convexification methods are often employed to reduce the computational complexity of the problem (MORGAN, 2015; CHU, 2015; NAIR; TSENG; BORRELLI, 2022). These techniques aim to approximate the non-convex problem as a convex or linear one (D'ASPREMONT; BOYD, S., 2003). Alternatively, some studies have adopted a polygonal representation for two-dimensional objects or a polyhedral representation for three-dimensional objects to formulate collision avoidance constraints. In such cases, a mixed-integer programming formulation is commonly used (KEVICZKY et al., 2008). Furthermore, some research endeavors have explored leveraging strong duality (BOYD, S. P.; VANDENBERGHE, 2004; ZHANG, X.; LINIGER; BORRELLI, 2020), Farkas' Lemma (GERDTS et al., 2011), or polar set (PATEL; GOULART, 2011) representations to exactly reformulate the collision avoidance constraints into expressions compatible with NLP solvers. However, these approaches introduce additional variables and constraints to the problem.

2.11 CONCLUSION

This chapter briefly reviewed some of the important topics that are needed throughout this thesis, such as the concept of automated driving, connected vehicles, and the notion of connected vehicles under automated driving at signal-free intersections. In addition, basic fundamentals of trajectory optimization, optimal control, model predictive control, and convex optimization were presented. Lastly, we reviewed the theoretical concepts of collision avoidance constraints and their applications in transportation and robotics. With the information presented in this chapter, we are now ready to begin with the main contributions of this thesis.

3 SIGNAL-FREE PATH-FREE INTERSECTION CONTROL PROBLEM

This chapter introduces the Signal-free Path-free Intersection Control (SPIC) which is a novel problem for urban intersection management. SPIC promotes better use of the intersection space to improve overall traffic efficiency by leveraging the potential of the CVAD to enable path-free traversing at a signal-free intersection, which is referred to as *plaza*. In the plaza, the intersection transforms into a boundary-constrained space, where free paths are permitted on the condition that the resulting trajectories do not intersect. In fact, in addition to vehicle-to-vehicle (V2V) collision avoidance, the constraints for the plaza's boundaries must be designed to disallow the CVAD from violating the limits of the plaza. To this end, additional constraints are imposed on the SPIC problem to define the geometry of the plaza.

In this chapter, we provide a detailed explanation of the SPIC problem. The notion of the plaza is introduced in Section 3.1, followed by the presentation of the vehicle model in Section 3.2. Furthermore, Sections 3.3 and 3.4 introduce the safety constraints, while Section 3.5 presents the concept of the objective function. Lastly, the overall formulation of the SPIC problem is outlined in Section 3.6, with a concluding summary provided in Section 3.7.

3.1 PLAZA MODELING

We consider a schematic example of the plaza in Figure 5 to clarify this idea in more detail. Figure 5a depicts a plaza P within which, as an example, two vehicles with trajectories T_1 and T_2 can travel between any two points, e.g., approaching and departing roads. These vehicles are free to adjust their trajectories without being bound to pre-specified paths or movements. The plaza P can have varied layouts, concerning the number of intersecting roads and shapes, to enable the modeling of various intersections. Figures 5b and 5c show the vehicles' coordinates in time. Figure 5d shows that as long as the trajectories of the two vehicles do not cross at the same time and at the same location, there is no collision. Analogously, this means that at all times a minimum safe distance, d_{s}, is maintained between every two vehicles as shown in Figure 5d.

According to the concept of the plaza, there is no need to continue with path-based roads and intersections and mimic path-based driving tasks. This enables increasing the intersection space utilization, thus allowing for higher flow and capacity. Therefore, in contrast to the conventional path-based (lane-based) urban roads and intersections, this flexibility offers more benefits under the scenario with heavy traffic flow, especially, with heavy left-turn movements (PARK, S.; RAKHA, 2010). The objective is to find the optimal trajectories of CVAD, without predefined paths, that should be followed by the vehicles so as to minimize one or more criteria, such as the intersection delay, fuel consumption, emissions, and passenger comfort, while strictly avoiding vehicle collision.

It should be noted that SPIC must include both static and dynamic obstacles which makes the problem more difficult because of the time-varying nature of vehicle positions in the plaza. In fact, in addition to preserving collision avoidance between the trajectories of every two vehicles,

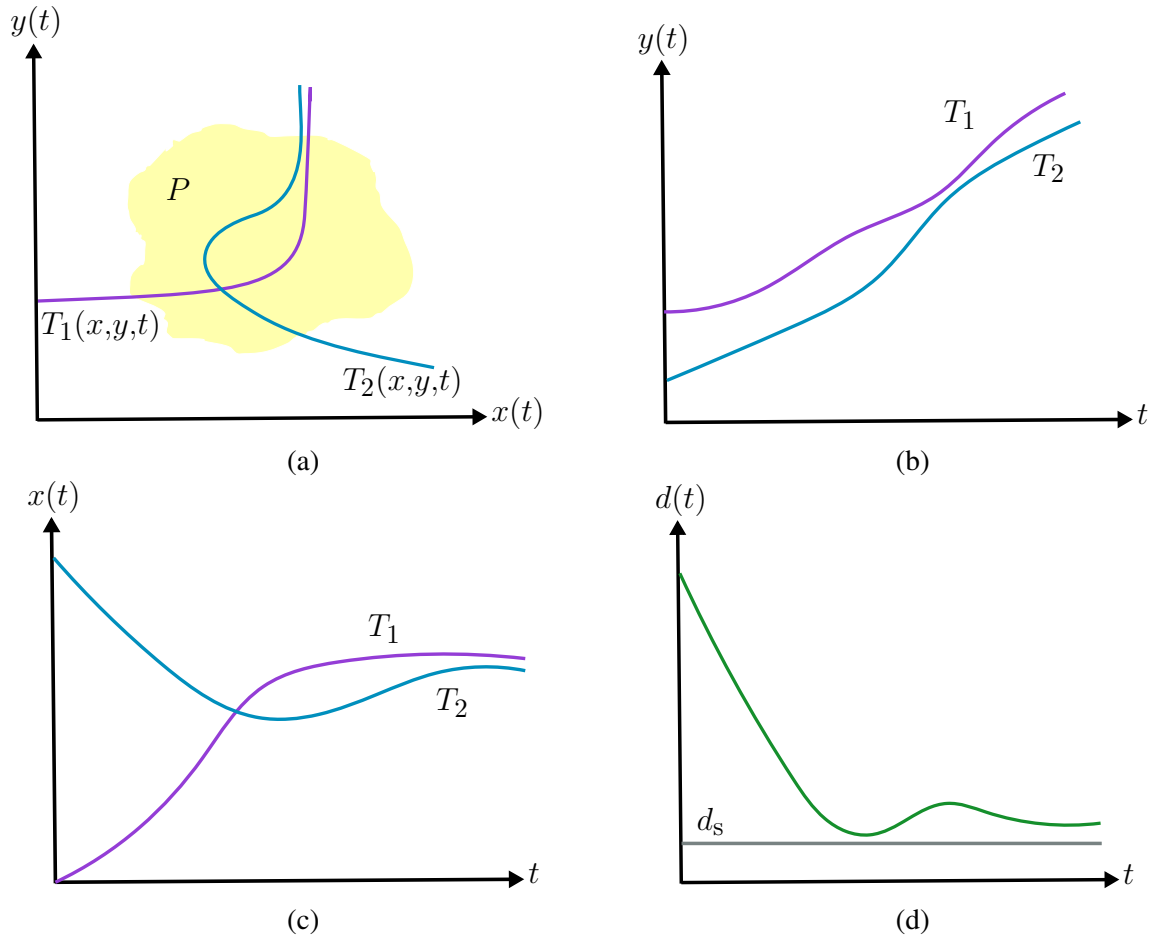


Figure 5 – (a) Plaza P and trajectories T_1 and T_2 of two CVAD in the Cartesian coordinate system; (b) vertical coordinate of the trajectories; (c) horizontal coordinate of the trajectories; and (d) distance between two vehicles and minimum safe distance d_s.

SPIC must produce trajectories that do not collide with the boundaries of the plaza, as long as CVAD are inside the plaza. To do so, at all times a minimum safe distance is maintained in any direction between every two vehicles as well as with intersection boundaries. To clarify this, we consider the following example for a typical four-leg intersection as a plaza which is shown in Figure 6.

We consider the case in which two CVAD streams are approaching a plaza from north and west, respectively. Figure 6a and Figure 6b show, respectively, the trajectories of these two CVAD and the relative distance between them in a case where no constraints were defined to avoid collision. Subsequently, as can be seen from the figure, these two traffic streams potentially collide with each other, i.e., they cannot keep a safe distance. Now, suppose the case we impose traffic safety constraints into the SPIC problem. In this case, the trajectories of the two CVAD and the relative distance between them are shown in Figure 6c and Figure 6d, respectively. As depicted in Figure 6c, the trajectories of both CVAD deviate to the right-hand side of the intersection in order to avoid a possible collision. Although the continuous trajectories of the two CVAD are in fact intersecting, as shown in Figure 6d, the trajectories do not cross at the same time and the same location. Thus the relative distance between them remains above a minimum

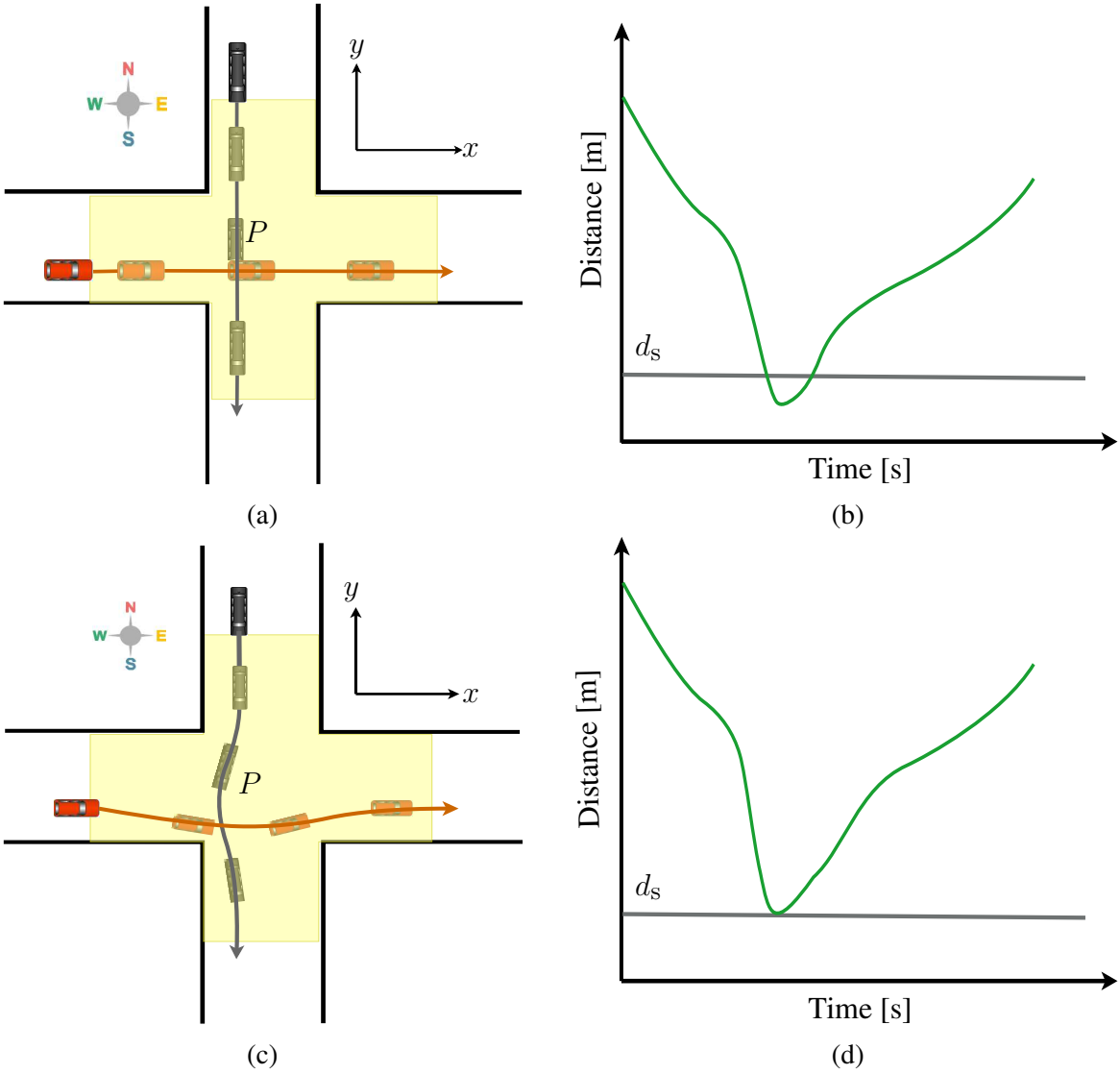


Figure 6 – Evaluation of the traffic safety constraint; (a) trajectories of the two CVAD and (b) the relative distance between them before applying collision avoidance constraints; (c) trajectories of the two CVAD and (d) the relative distance between them after applying collision avoidance constraints. When the constraint is active, the distance between two CVAD remains above the minimum safe distance d_s.

safe distance, and therefore there will be no collision.

3.2 DYNAMIC MODEL

A dynamical model is a mathematical representation used to describe the time-dependent behavior of a system (BRIN; STUCK, 2002). One of the most common forms of dynamical models is expressed through ordinary differential equations (ODEs). The general form of an ODE-based dynamical model to be controlled is described as:

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)), \tag{7}$$

In this equation, $\mathbf{x}(t)$ represents the state variables of the system, $\mathbf{u}(t)$ is the vector of control inputs, and t denotes time. The function $f(\cdot)$ describes a nonlinear function that defines the dynamic of the system. The continuous-time dynamical model (7) can be discretized to simulate the system's behavior over discrete time steps. This discretization is often achieved using numerical integration methods like the Euler method. The discretized equations with sampling time Δt can be expressed as:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t f(\mathbf{x}_k, \mathbf{u}_k), \tag{8}$$

with k the discrete time index and $t = k\Delta t$, $\mathbf{x}_k$ is the state variables vector at time k, and $\mathbf{u}_k$ is the control inputs vector at time k. Discretized dynamical models are essential in numerical simulations and numerical solutions of differential equations, enabling us to understand and predict complex system dynamics.

3.3 VEHICLE-TO-VEHICLE COLLISION AVOIDANCE CONSTRAINTS

Vehicle-to-vehicle (V2V) collision avoidance constraints present unique challenges for coordinating vehicles at the plaza. Two vehicles may end up occupying the same space at the same time, leading to a potential collision if they both maintain their current speed. In such cases, at least one of the vehicles needs to adjust its speed and/or change course to avert the collision, allowing the other vehicle to pass through safely. While collision-free trajectory optimization is well-established for single vehicles to avoid known obstacles, the complexity increases when multiple vehicles interact in scenarios like urban intersections or highways. Navigating through urban environments demands heightened safety measures due to the increased need for obstacle avoidance, contrasting with the relatively straightforward maneuvers often encountered on highways with minimal obstacle avoidance.

In essence, the interactions between vehicles at the plaza necessitate the introduction of constraints to generate collision-free trajectories for the vehicles. Unlike existing strategies for vehicular traffic safety, we are concerned with the position of vehicles and minimum safe distances between them in any direction and not with a following or lateral headway. This is because the paths of vehicles are not predefined. To ensure strict collision avoidance, a minimum safety distance, denoted as d_{s}, must be maintained in all directions between every pair of vehicles.

3.4 PLAZA BOUNDARIES CONSTRAINTS

In addition to V2V collision avoidance, for a signal-free path-free intersection plaza, it is crucial to ensure the prevention of collisions between vehicles and intersection plaza boundaries. Without traditional traffic signals and with path-free movements, the reliance on the geometric layout becomes even more critical to facilitate the safe movement of vehicles. By incorporating constraints that define the geometry and boundaries of the plaza, boundary-aware trajectories

can be generated which ensure the vehicles remain within the plaza boundaries while navigating through it.

3.5 OBJECTIVE FUNCTION

The objective function is essential in an optimization problem as it quantifies the performance of solutions and provides a well-defined target for achieving optimal results. By converting real-world problems into mathematical representations, the objective function enables systematic search algorithms to identify the most favorable solutions while considering trade-offs between competing objectives. In vehicle coordination at intersections, common objective functions include minimizing total travel time or delay, maximizing intersection throughput, and ensuring safety by reducing collision risk and promoting compliance with traffic regulations. These objectives aim to optimize traffic flow and enhance overall efficiency and safety at intersections.

3.6 OVERALL FORMULATION OF THE SPIC PROBLEM

The SPIC problem formulation can be written as follows:

$$
\begin{aligned}
&\min \quad \text{Objective function,} \\
&\text{s.t.} \\
&\qquad \text{Vehicle's model,} \\
&\qquad \text{Vehicle's kinematic constraints,} \\
&\qquad \text{Vehicle's initial and final constraints,} \\
&\qquad \text{Vehicle-to-vehicle collision avoidance constraints,} \\
&\qquad \text{Plaza boundaries constraints.}
\end{aligned}
\tag{9}
$$

3.7 CONCLUSION

In conclusion, designing a controller or motion planner plays a critical role in the navigation of the CVAD through the plaza in the SPIC problem. Its ability to bridge the gap between environmental perception and vehicle response is essential to guide the vehicle's actions and ensure vehicles remain on the boundaries of the plaza while avoiding collisions. The successful implementation of an effective controller significantly contributes to the overall safety and performance of vehicles' coordination.

In the next chapters, this thesis presents novel methods to address the SPIC problem using optimal control and model predictive control techniques. The proposed approaches leverage the theory of finite Fourier series (FFS) and Bézier curves, as well as nonlinear model predictive contouring control (NMPCC) to effectively address the collision-free trajectory optimal control problem outlined in (9). These innovative methods hold significant promise in enhancing path-following capabilities while ensuring collision avoidance in complex scenarios.

4 AN OPTIMAL CONTROL APPROACH FOR THE SPIC PROBLEM

The problem of signal-free path-free intersection control (SPIC) for the coordination of connected vehicles under automated driving (CVAD) at the intersection plaza was presented in Chapter 3. In this chapter, we introduce the intersection trajectory optimal control problem (ITOP) as a specific type of the SPIC problem. ITOP can be stated as finding optimal trajectories, without predefined paths, in which the vehicles depart from their initial states and, after crossing a road intersection, arrive at final states so as to minimize one or more criteria and satisfy the plaza physical limits and collision avoidance constraints. The optimality is measured concerning minimum travel time and/or minimum total acceleration.

ITOP enables using a framework along with two shape-based methods; one based on the finite Fourier series (FFS) and another using Bézier curves. These methods are employed to solve the ITOP effectively. It should be noted that other direct methods for solving optimal control problems can be employed to provide a solution for ITOP. According to the receiving information about vehicle states, and also intersection geometry, the FFS and Bézier curves methods can generate near-optimal and collision-free trajectories of the CVAD, considering the absence of signals and predetermined paths. In particular, the use of the FFS and Bézier methods and discretization notions convert the ITOP problem to a nonlinear programming (NLP) problem, with Fourier or Bézier coefficients as the unknown parameters.

In Section 4.1 the ITOP is further detailed, encompassing intersection plaza modeling, vehicle state equations, performance criteria, and constraints. Section 4.2 introduces intersection trajectory optimization methods, leveraging FFS and Bézier approaches for state representation and NLP transcriptions. Section 4.3 showcases numerical outcomes, drawing comparisons between FFS and Bézier methods. Lastly, Section 4.4 offers conclusion and preliminary ideas for the next chapter.

4.1 ITOP FORMULATION

In this section, we model the environment of the intersection plaza and the vehicles. Each vehicle is described by two simple state equations. Subsequently, we give mathematical formulations of the set of constraints that have to be satisfied for the desired trajectories to be feasible. The set of constraints includes vehicle dynamic constraints and traffic safety constraints, which consist of V2V collision avoidance constraints and plaza boundary constraints. Finally, an objective function is presented.

4.1.1 Plaza modeling

A four-leg intersection is shown in Figure 7 as a plaza. In the figure, two vehicles are shown with example trajectories T_1 and T_2. The X and Y axes represent the central lines of the intersection on the Cartesian coordinate system (CCS). This intersection is modeled simply by its

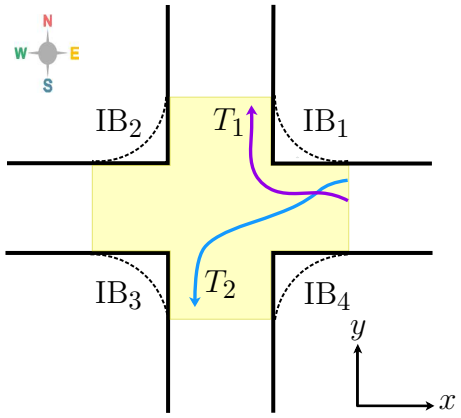


Figure 7 – A typical four-leg intersection as a plaza (yellow area) with approximated boundaries (dashed lines), and example trajectories T_1 and T_2 of two CVAD.

four intersection boundaries (IB) shown by the dashed lines in Figure 7. Each IB$_h$, $h = 1, \ldots, 4$, is modeled by an exponential function given by:

$$y_h = f_h(x(t)), \tag{10}$$

where,

$$f_h(x(t)) = r_{0,h} + r_{1,h} \cdot \mathrm{e}^{r_{2,h} \cdot (x(t) + r_{3,h})}, \tag{11}$$

with parameters $r_{0,h}$, $r_{1,h}$, $r_{2,h}$, and $r_{3,h}$ shape the function according to the intersection geometry and h being the number of IB utilized for defining the plaza boundaries constraints.

4.1.2 Vehicle's equations of motion

A simplified equation of motion (EoM) is used that models vehicles as particles as follows (although other models can be used):

$$\begin{cases} \ddot{x}_j(t) = a_{x_j}(t), \\ \ddot{y}_j(t) = a_{y_j}(t), \end{cases} \tag{12}$$

with a_{x_j} and a_{y_j} the acceleration of vehicle j in coordinates x and y in the CCS, respectively, and t the continuous time. The total (absolute) acceleration of vehicle j is given by:

$$a_j(t) = \sqrt{a_{x_j}^2(t) + a_{y_j}^2(t)}. \tag{13}$$

Given N_v vehicles at the plaza, the total speed increment is defined as:

$$\Delta v(t) = \Delta v_1(t) + \ldots + \Delta v_{N_\mathrm{v}}(t), \tag{14}$$

with $\Delta v_j(t)$ the speed increment of the j-th vehicle:

$$\Delta v_j(t) = \int_0^{T_\mathrm{f}} a_j(t)\mathrm{d}t, \; j = 1, \ldots, N_\mathrm{v}, \tag{15}$$

in which T_f stands for the completion time, i.e., the time taken by the vehicles to cross the plaza.

4.1.3 Vehicle's kinematic constraints

To guarantee that the speeds and accelerations of the vehicles are within admissible values, the following constraints are imposed for CVAD j:

$$\begin{aligned} &a_j(t) \leq a_{\max}, \\ &0 \leq v_j(t) \leq v_{\max}, \end{aligned} \tag{16}$$

with a_j the total acceleration of vehicle j, v_j the speed of vehicle j, and $a_{\max}$ and $v_{\max}$ the maximum total acceleration and maximum speed, respectively.

4.1.4 Initial and final constraints

In addition to vehicles' kinematic constraints, initial and final constraints are defined for the trajectory of each CVAD as follows,

$$\begin{aligned} &\mathbf{z}_j(t=0) = \mathbf{z}_{j,0}, \;\; \mathbf{z}_j(t=T_\mathrm{f}) = \mathbf{z}_{j,\mathrm{f}} \\ &\dot{\mathbf{z}}_j(t=0) = \dot{\mathbf{z}}_{j,0}, \;\; \dot{\mathbf{z}}_j(t=T_\mathrm{f}) = \dot{\mathbf{z}}_{j,\mathrm{f}}, \end{aligned} \tag{17}$$

where $\mathbf{z}_j(t) = [x_j(t) \; y_j(t)]^\top$ with $x_j(t)$ and $y_j(t)$ the position of vehicle j in each coordinate x and y, respectively, $\mathbf{z}_{j,0}$ and $\mathbf{z}_{j,f}$ the given initial and final position state variables, respectively, and $\dot{\mathbf{z}}_{j,0}$ and $\dot{\mathbf{z}}_{j,\mathrm{f}}$ are the given initial and final speed state variables of each vehicle j, respectively.

4.1.5 V2V collision avoidance constraints

In order to strictly avoid collisions, a minimum safe distance, d_s, must be kept in any direction between every two vehicles:

$$d_{ij}(t) \geq d_\mathrm{s}, \tag{18}$$

in which $d_{ij}(t)$ is the distance between vehicles i and j, $i = 1, \ldots, N_\mathrm{v}$, $j = 1, \ldots, N_\mathrm{v}$, with N_v the number of vehicles in the plaza, and $i < j$.

4.1.6 Plaza boundaries constraints

In addition to V2V collision avoidance, constraints must be enforced to disallow the CVAD from violating the boundaries of the plaza. Then, the intersection's geometric constraints, based on (10), that ensure there are no collisions of CVAD with the boundaries, are:

$$\begin{cases} y_j(t) \leq f_h(x_j(t)), \text{ if } h = 1, 2 \\ y_j(t) \geq f_h(x_j(t)), \text{ if } h = 3, 4 \end{cases} \quad \forall j, \; \forall t. \tag{19}$$

with f_h is defined in (11).

4.1.7 Performance index

The performance index is chosen as a weighted sum of Δv and T_{f} that should be minimized:

$$\mathcal{J} = w_1 \cdot \Delta v + w_2 \cdot T_{\mathrm{f}}, \tag{20}$$

with $w_1 \geq 0$ and $w_2 \geq 0$ weighting parameters. Minimizing Δv and T_{f} are conflicting objectives that have a direct effect on fuel consumption and comfort versus speed and completion time.

4.1.8 Overall ITOP Formulation

Given the EoM, constraints, and performance index, the ITOP can be written as follows:

$$\begin{aligned}
&\text{minimize } \mathcal{J} \; in \; (20), \\
&\text{s.t.} \\
&\text{Vehicle's kinematic model } (12), \\
&\text{Vehicle's kinematic constraints } (16), \\
&\text{Initial and final constraints } (17), \\
&\text{V2V collision avoidance constraints } (18) \\
&\text{Plaza boundaries constraints } (10) \text{ and } (19).
\end{aligned} \tag{21}$$

The states are the position and speed of each vehicle j in each coordinate x and y ($x_j(t)$, $y_j(t)$, $v_{x_j}(t)$, and $v_{y_j}(t)$), and the control variables are $a_{x_j}(t)$, $a_{y_j}(t)$, and T_{f}. Moreover, the resulting collision-free trajectory optimization problem is non-linear and non-convex duo to the objective function and also V2V collision avoidance constraints.

4.2 INTERSECTION TRAJECTORY OPTIMIZATION METHOD

In this section, the FFS and Bézier curves methods are proposed to solve the ITOP for coordinating the CVAD at the intersection plaza. In these methods, the state variables, i.e., positions and speeds, are interpolated, and control variables, i.e., accelerations and completion time, are considered in the objective function. Then, the FFS or Bézier representations of state variables are imposed on the dynamics, and the required acceleration to realize the resulting trajectories are evaluated. Finally, the ITOP is reduced to a system of algebraic equations in the Fourier series or Bézier coefficients, and a collision-free trajectory optimization problem is formulated.

4.2.1 FINITE FOURIER SERIES

Inspired by the flexibility of the Fourier series to approximate any continuous function, in this work, we propose to customize the FFS method to solve the ITOP and generate feasible trajectories of the CVAD, based on the works by Taheri and Abdelkhalik (2016) and Mingying

et al. (2020) applied to spacecraft. The trajectories of the CVAD can be defined, by a given parameterized FFS whose coefficients must be optimized.

4.2.1.1 Fourier approximations

Each position state variable of each vehicle j in each coordinate x and y (x_j and y_j) is approximated by a FFS as:

$$\mathbf{z}(\tau) = \frac{b_0^{\mathbf{z}}}{2} + \sum_{n=1}^{N_{\mathbf{z}}} \left(b_n^{\mathbf{z}} \cos\left(\pi n \tau\right) + c_n^{\mathbf{z}} \sin\left(\pi n \tau\right) \right), \tag{22}$$

with $N_{\mathbf{z}}$ the number of Fourier terms used for approximating position state variable $\mathbf{z}$, and $b_0^{\mathbf{z}}$; $b_n^{\mathbf{z}}$ and $c_n^{\mathbf{z}}$ the corresponding coefficients to be determined, and τ the scaled time such that:

$$0 \leq \tau \leq 1, \quad \tau = \frac{t}{T_{\mathrm{f}}}. \tag{23}$$

The speed state variables of each vehicle j in each coordinate x and y (v_{x_j} and v_{y_j}) are the first derivatives of (22) with respect to the scaled time, τ. The corresponding first and second derivatives with respect to the scaled time can be readily obtained, For further details, please refer to the Appendix A. Afterward, this representation of state variables is imposed to the dynamics and the required acceleration to realize the resulting trajectories are evaluated.

4.2.1.2 Boundary conditions

For each vehicle, we know the boundary conditions (BCs), i.e., the initial and final positions and speeds in the coordinate system. The BCs with respect to scaled time for each vehicle are:

$$\begin{aligned} \mathbf{z}(0) &= \mathbf{z}_{\mathrm{I}}, \quad \mathbf{z}'(0) = T_{\mathrm{f}} \dot{\mathbf{z}}_{\mathrm{I}}, \\ \mathbf{z}(1) &= \mathbf{z}_{\mathrm{F}}, \quad \mathbf{z}'(1) = T_{\mathrm{f}} \dot{\mathbf{z}}_{\mathrm{F}}, \end{aligned} \tag{24}$$

and the labels I and F refer to 'initial' and 'final', respectively, while the prime denotes the derivative with respect to the scaled time and the dot the derivative with respect to the conventional time. These relations are obtained through the chain rule resulting in:

$$\frac{\mathrm{d}}{\mathrm{d}t} = \frac{1}{T_{\mathrm{f}}} \frac{\mathrm{d}}{\mathrm{d}\tau}.$$

The previous relation is also needed to obtain the time derivative of the states and the following relation is needed to obtain the time second derivative of the states:

$$\frac{\mathrm{d}^2}{\mathrm{d}t^2} = \frac{1}{T_{\mathrm{f}}^2} \frac{\mathrm{d}^2}{\mathrm{d}\tau^2}.$$

The advantage of using the BCs is that they are physically meaningful quantities in contrast with the Fourier coefficients. The direct estimation of Fourier coefficients is not straightforward because the sum of a series of unique sin and cos terms is not known prior. Therefore, it is easier

and more physically meaningful to extract some of the unknown Fourier coefficients in terms of the physical values to reduce the total number of design variables. Another advantage of the BCs for solving some of the Fourier parameters is that we avoid the introduction of extra equality constraints.

4.2.1.3 Using the BCs for expressing some coefficients

Some of the coefficients can be expressed in terms of the BCs and the other coefficients. This reduces the number of unknown Fourier coefficients. Next, the expressions of the first four coefficients are derived.

By manipulating algebraically (22), its first derivative, and (24), and solving a linear system of equations, it is straightforward to derive the first four coefficients of (22) as:

$$\begin{aligned} b_1^{\mathbf{z}} &= \frac{\mathbf{z}_{\mathrm{I}} - \mathbf{z}_{\mathrm{F}}}{2} - \sum_{n=3}^{N_{\mathbf{z}}} b_n^{\mathbf{z}}; && \text{for } n \text{ odd}, \\ b_2^{\mathbf{z}} &= \frac{\mathbf{z}_{\mathrm{I}} + \mathbf{z}_{\mathrm{F}}}{2} - \frac{b_0}{2} - \sum_{n=4}^{N_{\mathbf{z}}} b_n^{\mathbf{z}}; && \text{for } n \text{ even}, \\ c_1^{\mathbf{z}} &= \frac{1}{2\pi}(\dot{\mathbf{z}}_{\mathrm{I}} - \dot{\mathbf{z}}_{\mathrm{F}}) - \sum_{n=3}^{N_{\mathbf{z}}} n c_n^{\mathbf{z}}; && \text{for } n \text{ odd}, \\ c_2^{\mathbf{z}} &= \frac{1}{4\pi}(\dot{\mathbf{z}}_{\mathrm{I}} + \dot{\mathbf{z}}_{\mathrm{F}}) - \frac{1}{2}\sum_{n=4}^{N_{\mathbf{z}}} n c_n^{\mathbf{z}}; && \text{for } n \text{ even}, \end{aligned} \tag{25}$$

The derivation of (25) from (22) and (24) is provided in the Appendix A.1.

4.2.1.4 Evaluation Points

In order to solve for the unknown Fourier coefficients, the EoM are evaluated at m points, called discretization points (DPs). We consider m DPs with equal time intervals within the scaled time:

$$\tau_1 = 0 < \tau_2 < \cdots < \tau_{m-1} < \tau_m = 1.$$

All the proposed constraints in Section 4.1 are satisfied only at each DP. Thus, to avoid violations between DPs, we must choose a large enough safety distance and/or sufficiently dense DPs. This special set of DPs can be used to evaluate the integral efficiently by using a reduced set of points. Moreover, selecting a suitable number of discretization points is often problem-dependent and is found after a few trials, much like selecting the number of Fourier terms; too few will cause inaccurate results and too many will slow down the solution algorithm.

4.2.1.5 Compact matrix form representation

Since the EoM are evaluated at the DPs, a compact matrix form representation for the position state variables and its derivatives (speed state variables and accelerations) already

incorporating the coefficients from the BCs can be used. Considering bracket $[\cdot]$ as vector representation, we can write the positions state variables at the m DPs as vectors of its values:

$$[\mathbf{z}]_{m\times 1} = [A_{\mathbf{z}}]_{m\times(2N_{\mathbf{z}}-3)}[X_{\mathbf{z}}]_{(2N_{\mathbf{z}}-3)\times 1} + [F_{\mathbf{z}}]_{m\times 1}, \tag{26}$$

with $[A_{\mathbf{z}}]$ a matrix of coefficients depending on $N_{\mathbf{z}}$ and on the DPs, $[X_{\mathbf{z}}]$ the vector of unknown Fourier coefficients to be determined, and $[F_{\mathbf{z}}]$ a vector of terms obtained from the BCs. The values of $[A_{\mathbf{z}}]$ and $[F_{\mathbf{z}}]$ are computed offline and the only unknowns are the FFS coefficients. The definition of these matrices and the representation of the compact matrix form of the first and second derivatives of (26) are both derived in Appendix A.2.

The total acceleration of vehicle j along the trajectory can be represented in the matrix compact form as well by replacing (26) and its derivatives in (12) and (13):

$$\big[a_j\big]_{m\times 1} = \sqrt{\sum_{\forall \mathbf{z}} [a_{\mathbf{z}}]^2_{m\times 1}} \leq [a_{\max}]_{m\times 1}. \tag{27}$$

Note that $a_{\mathbf{z}} = \ddot{\mathbf{z}}$, therefore we need the relations between $\ddot{\mathbf{z}}$ and $\mathbf{z}''$ (see Section 4.2.1.2).

4.2.1.6 Nonlinear programming formulation

Given the compact matrix form (26) and corresponding derivatives, the ITOP described in (21) is transcribed as the following nonlinear programming problem:

$$\begin{aligned}
\min_{[X_{\mathbf{z}}]\forall\mathbf{z}\forall j, T_{\mathrm{f}}} \quad & w_1 \cdot \Delta v(t) + w_2 \cdot T_{\mathrm{f}} \\
\text{s.t.} \quad & \\
& [a_j(t)] \leq [a_{\max}], \\
& 0 \leq [v_j(t)] \leq [v_{\max}], \\
& [d_{ij}(t)] \geq [d_{\mathrm{s}}], \\
& [y_j(t)] \leq [f_h(x_j(t))], \text{ if } h = 1, 2 \\
& [y_j(t)] \geq [f_h(x_j(t))], \text{ if } h = 3, 4
\end{aligned} \tag{28}$$

where $i = 1, \ldots k,\ j = 1, \ldots, k,\ i < j$; w_1 and w_2 are weighting positive parameters, which assign priority to each term of the objective function.

We note that (26) and its derivatives are embedded in (28) through the substitution in (12)–(15). The vector of decision variables is the completion time, T_{f}, and the remaining unknown Fourier coefficients after enforcing the BCs, $[X_{\mathbf{z}}]$. Therefore, the number of decision variables is $N_{\mathrm{v}}(2N_{\mathbf{z}} - 9) + 1$.

4.2.1.7 Initialization of Decision Variables

Due to the NLP formulation, the efficiency of the solvers is sensitive to the initial guess of decision variables. Hence, obtaining a good initial guess is expected to decrease the computation

time considerably. The initialization of the unknown coefficients can be expressed in compact form as:

$$[X_{\mathbf{z}}]_{(2N_{\mathbf{z}}-3)\times 1} = \Big([A_{\mathbf{z}}]_{n_{\mathrm{a}}\times(2N_{\mathbf{z}}-3)}\Big)^{-1}\Big([\mathbf{z}_{\mathrm{a}}]_{n_{\mathrm{a}}\times 1} - [F_{\mathbf{z}}]_{n_{\mathrm{a}}\times 1}\Big), \tag{29}$$

with n_{a} the number of discretization points for the approximation and $[\mathbf{z}_{\mathrm{a}}]$ the approximated position state variables. Then, cubic polynomials can be used to approximate the position state variables at the DPs, and more detail can be found in Appendix A.3.

The initialization of the completion time can be approximated by arbitrarily selecting the time taken by a vehicle to cross in a straight direction from its origin to its destination with maximum total acceleration:

$$T_{\mathrm{a}} = \sqrt{\frac{2S}{a_{\max}}}, \tag{30}$$

where S is the distance between the origin and destination of the selected vehicle.

4.2.2 Bézier Curves

In the previous section, we explored the FFS method for the generation of trajectories. In this section, we incorporate the Bézier curves method (LATTARULO et al., 2018; SCHWUNG; LUNZE, 2021; FAN et al., 2020) and discretization concepts to transform the ITOP into an NLP, where the unknown parameters are the Bézier coefficients. Additionally, we present a novel compact representation of the Bézier curve method, which further reduces the number of decision variables compared to the work by Fan et al. (2020).

4.2.2.1 Bézier Approximations

The Bézier curves (FAROUKI, 2008) have several properties for trajectory optimization that are appropriate for the purpose of this work; (i) the starting and ending points of the curve correspond to the first and final Bézier coefficients, respectively; (ii) the curve completely lies within the convex hull formed by all Bézier coefficients; and (iii) the curves have the advantage of simplicity and curvature continuity. In this part, a Bézier curve is employed to approximate each position state variable of each vehicle j in each coordinate of the CCS (x_j and y_j) as follows:

$$\mathbf{z}(\tau) = \sum_{l=0}^{n_{\mathbf{z}}} B_{\mathbf{z},l}(\tau) P_{\mathbf{z},l}, \tag{31}$$

with $\mathbf{z} = [x_j(\tau)\ y_j(\tau)]^\top$, $0 \leq \tau = t/T_{\mathrm{f}} \leq 1$ the scaled time, $n_{\mathbf{z}}$ the number of Bézier terms (order of the Bézier curve), $P_{\mathbf{z},j}$ the unknown Bézier coefficients to be determined, and $B_{\mathbf{z},l}(\tau)$ the Bernstein basis polynomials given by:

$$B_{\mathbf{z},l}(\tau) = \binom{n_{\mathbf{z}}}{l}\tau^l(1-\tau)^{n_{\mathbf{z}}-l},\ \ l \in \{0,1,\ldots,n_{\mathbf{z}}\}. \tag{32}$$

Accordingly, the first and second derivatives of (31) with respect to the scaled time, τ, can be easily obtained. For more detailed information, please refer to the Appendix B.

4.2.2.2 Boundary conditions

Similar to the FFS method, the BCs, i.e., the initial and final positions and speeds with respect to scaled time for each vehicle are:

$$\mathbf{z}(0) = \mathbf{z}_{\mathrm{I}}, \;\; \mathbf{z}(1) = \mathbf{z}_{\mathrm{F}}, \;\; \mathbf{z}'(0) = T_{\mathrm{f}}\dot{\mathbf{z}}_{\mathrm{I}}, \;\; \mathbf{z}'(1) = T_{\mathrm{f}}\dot{\mathbf{z}}_{\mathrm{F}}. \tag{33}$$

The labels 'I' and 'F' refer to 'initial' and 'final', respectively, the prime denotes the derivative with respect to the scaled time, and the dot the derivative with respect to the continuous time.

4.2.2.3 Using the BCs for expressing some coefficients

By manipulating algebraically (31) and (32) and using the BCs (33), it is straightforward to derive four Bézier coefficients of (31) as a function of given BCs as follows:

$$\begin{aligned} P_{\mathbf{z},0} = \mathbf{z}_{\mathrm{I}}, \quad P_{\mathbf{z},1} = \mathbf{z}_{\mathrm{I}} + \frac{T_{\mathrm{f}}\dot{\mathbf{z}}_{\mathrm{I}}}{n_{\mathbf{z}}}, \\ P_{\mathbf{z},n_{\mathbf{z}}-1} = \mathbf{z}_{\mathrm{F}} - \frac{T_{\mathrm{f}}\dot{\mathbf{z}}_{\mathrm{F}}}{n_{\mathbf{z}}}, \quad P_{\mathbf{z},n_{\mathbf{z}}} = \mathbf{z}_{\mathrm{F}}, \end{aligned} \tag{34}$$

reducing the number of unknown Bézier coefficients, thus speeding up the optimization. The derivation of (34) from (31) and (33) is provided in the Appendix B.1. Then, substituting these coefficients in (31) and organizing the resulting expression gives:

$$\mathbf{z}(\tau) = F_{\mathbf{z}} + \sum_{l=2}^{n_{\mathbf{z}}-2} B_{\mathbf{z},l}(\tau) P_{\mathbf{z},l}, \tag{35}$$

with

$$F_{\mathbf{z}} = B_{\mathbf{z},0}P_{\mathbf{z},0} + B_{\mathbf{z},1}P_{\mathbf{z},1} + B_{\mathbf{z},n_{\mathbf{z}}-1}P_{\mathbf{z},n_{\mathbf{z}}-1} + B_{\mathbf{z},n_{\mathbf{z}}}P_{\mathbf{z},n_{\mathbf{z}}}. \tag{36}$$

The corresponding first and second derivatives with respect to the scaled time, $\mathbf{z}'(\tau)$ and $\mathbf{z}''(\tau)$, can be readily obtained.

4.2.2.4 Evaluation points

In order to solve for the unknown Bézier coefficients, similar to the FFS method, the EoM are evaluated at m DPs with equal time intervals within the scaled time ($\tau_i - \tau_{i-1} = 1/(m-1), i = 2, \ldots, m$):

$$\tau_1 = 0 < \tau_2 < \cdots < \tau_{m-1} < \tau_m = 1, \tag{37}$$

The constraints in Section 4.1 are satisfied only at each DP.

4.2.2.5 Compact matrix form representation

The compact matrix form representation of the position state variables and its derivatives can be written as vectors of its values at the m DPs as follows:

$$[\mathbf{z}]_{m\times 1} = [B_{\mathbf{z}}]_{m\times(n_{\mathbf{z}}-3)}[X_{\mathbf{z}}]_{(n_{\mathbf{z}}-3)\times 1} + [F_{\mathbf{z}}]_{m\times 1}, \tag{38}$$

with $[F_{\mathbf{z}}]$ a constant vector depending on $n_{\mathbf{z}}$ and on the BCs obtained using (36), $[B_{\mathbf{z}}]$ a matrix of coefficients given by:

$$[B_{\mathbf{z}}]_{m\times(n_{\mathbf{z}}-3)} = [B_{\mathbf{z},2}\ B_{\mathbf{z},3}\ \ldots\ B_{\mathbf{z},n_{\mathbf{z}}-2}]^{\top}, \tag{39}$$

and $[X_{\mathbf{z}}]$ the vector of unknown Bézier coefficients:

$$[X_{\mathbf{z}}]_{(n_{\mathbf{z}}-3)\times 1} = [P_{\mathbf{z},2}\ P_{\mathbf{z},3}\ \ldots\ P_{\mathbf{z},n_{\mathbf{z}}-2}]^{\top}. \tag{40}$$

Matrices $[B_{\mathbf{z}}]$ and $[F_{\mathbf{z}}]$ are computed offline and $[X_{\mathbf{z}}]$ results from the optimization. The compact forms of the first and second derivatives of (38) have a similar structure. Further detail is provided in Appendix B.2.

4.2.2.6 Nonlinear Programming Formulation

Given the compact matrix form (38) and corresponding derivatives, we can formulate a NLP with the unknown Bézier coefficients $[X_{\mathbf{z}}]$ and the completion time T_{f} as decision variables:

$$\begin{aligned}
&\min_{[X_{\mathbf{z}}]\forall\mathbf{z}\forall j, T_{\mathrm{f}}} \ \mathcal{J} \\
&\text{s.t.}[a_j(t)] \leq [a_{\max}], \\
&\quad 0 \leq [v_j(t)] \leq [v_{\max}], \\
&\quad [d_{ij}(t)] \geq [d_{\mathrm{s}}], \\
&\quad [y_j(t)] \leq [f_h(x_j(t))], \text{ if } h = 1, 2 \\
&\quad [y_j(t)] \geq [f_h(x_j(t))], \text{ if } h = 3, 4,
\end{aligned} \tag{41}$$

with $i = 1, \ldots, k$, $j = 1, \ldots, k$, and $i < j$. We note that (38) and its derivatives are embedded in formulation (41) through the substitution in (12)–(15).

4.2.2.7 Initialization of Decision Variables

The initialization of the unknown Bézier coefficients can be expressed in a compact form as follows:

$$[X_{\mathbf{z}}]_{(n_{\mathbf{z}}-3)\times 1} = \left([B_{\mathbf{z}}]_{n_{\mathrm{a}}\times(n_{\mathbf{z}}-3)}\right)^{-1}\left([\mathbf{z}_{\mathrm{a}}]_{n_{\mathrm{a}}\times 1} - [F_{\mathbf{z}}]_{n_{\mathrm{a}}\times 1}\right), \tag{42}$$

with n_{a} the number of DPs for the approximation and $[\mathbf{z}_{\mathrm{a}}]$ the approximated position state variables. A cubic Bézier curve can be used to approximate $[\mathbf{z}_{\mathrm{a}}]$ using the BCs. The initialization of the completion time T_{a} can be approximated using (30).

4.3 NUMERICAL RESULTS

This section presents an evaluation of the ITOP solution using the FFS and Bézier curve methods. The evaluation is carried out on a machine with an Intel Core i5-8265U CPU and

Table 1 – Settings for vehicles and intersection.

Parameter	Description	Value
N_{v}	Number of CVAD	3
$a_{\max}$	Maximum acceleration (m/s^2)	2
$v_{\max}$	Maximum speed (m/s)	15
d_{s}	Safe distance (m)	1
w_{r}	Road width (m)	11
L_{r}	Road length (m)	90

Table 2 – Scaling parameters.

Parameter	IB_1	IB_2	IB_3	IB_4
r_0	11	11	-11	-11
r_1	1	1	-1	-1
r_2	-1	1	1	-1
r_3	-11	11	11	-11

16 GB of memory in MATLAB 2018b. The NLP problems (28) and (41) are solved using the `fmincon` solver of the optimization toolbox. To compute each $\Delta v_j(t)$ in (15), we perform numerical integration of the corresponding total acceleration over time T_{f} utilizing the built-in function `trapz`. Furthermore, we compare the results obtained with the FFS method to those obtained using the method based on the Bézier curve.

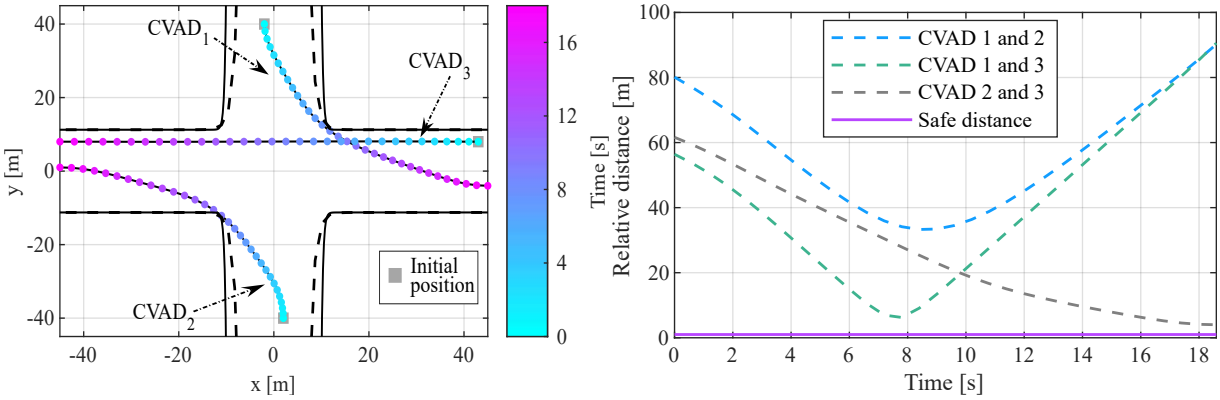
4.3.1 Scenario setup

We investigate a simple scenario with three CVAD at the intersection plaza each of which with different initial and final positions and speeds. The goal is to show that the proposed FFS and Bézier methods are able to generate near-optimal and collision-free trajectories for these three vehicles. CVAD_1 travels from north to east, CVAD_2 travels from south to west, and CVAD_3 goes straight from east to west. The center of the plaza is the origin of the CCS. The initial positions $(x_{\mathrm{i},j}, y_{\mathrm{i},j})$ of the three CVAD are $(-2, 40)$ m, $(2, -40)$ m, and $(43, 8)$ m, respectively, and the final positions $(x_{\mathrm{f},j}, y_{\mathrm{f},j})$ are, respectively, $(45, -4)$ m, $(-45, 1)$ m, and $(-45, 8)$ m. The initial speeds $(v_{\mathrm{i},x_j},\ v_{\mathrm{i},y_j})$ are $(1, -5)$ m/s, $(-1, 5)$ m/s, and $(-7, 0)$ m/s, respectively, and the final speeds $(v_{\mathrm{f},x_j}, v_{\mathrm{f},y_j})$ are, respectively, $(6, -2)$ m/s, $(-6, 2)$ m/s, and $(-8, 0)$ m/s. The required parameters for the formulation and numerical simulations are summarized in Tables 1, 2, and 3.

In order to determine appropriate values for the weighting parameters m, w_1, w_2, $N_{\mathbf{z}}$, and $n_{\mathbf{z}}$ we undertake the resolution of the NLP problems (28) and (41) with various settings. After thorough experimentation, we select $w_1 = 4$, $w_2 = 2$, $m = 30$, $N_{\mathbf{z}} = 8$, and $n_{\mathbf{z}} = 8$ as they achieve a reasonable trade-off between approximation accuracy and computational complexity for future problem instances. Further elaboration on the chosen numerical results can be found in Appendix A.4.

Table 3 – Experiment setup for three CVAD.

CVAD	Initial position (m)	Final position (m)	Initial speed (m/s)	Final speed (m/s)	Direction
1	$(-2, 40)$	$(45, -4)$	$(0, -3)$	$(4, 0)$	N-E
2	$(2, -40)$	$(-45, 1)$	$(0, 2)$	$(-4, 0)$	S-W
3	$(43, 8)$	$(-45, 8)$	$(-3, 0)$	$(-5, 0)$	E-W



(a) (b)

Figure 8 – SC1: FFS method; (a) trajectories of three CVAD and (b) safe distance between them with $d_{\mathrm{s}} = 1$ m.

4.3.2 Scenario 1 (SC1): Analysis of the trajectories obtained by the FFS method

The optimized trajectories generated by the solution of the NLP problem based on the FFS method are illustrated in Figure 8a. The labels CVAD_1–CVAD_3 identify the trajectories, colored disks indicate DPs and time, and the gray squares are the initial positions for each CVAD. The solid thick black lines show the boundary of each IB whose approximations are presented by black dashed lines. For this particular scenario, the trajectories deviate from what would be expected in a path-based method and it is clear that the CVAD follow free trajectories. Noteworthy, the followed paths seem to approach the paths of minimum distance. Figure 8b shows that the distances between every two vehicles remain above the safe distance by a large margin for this scenario. It means that the trajectories of every two CVAD do not cross simultaneously and at the same location and ensure collision avoidance. Although the continuous trajectories of CVAD_1 and CVAD_3 are in fact intersecting, as shown in Figure 8b the relative distance between them (the green dash-line) is above a safe distance (solid pink line), and therefore there will be no collision between them. The solution of the NLP problem (28) for $w_1 = 4$, $w_2 = 2$, $m = 30$ and $N_{\mathbf{z}} = 8$ resulted in a completion time, $T_{\mathrm{f}} = 18.3$ s, a total speed increment, $\Delta v = 14.6$ m/s, and an objective function value of 95.3. The computation time for generating the trajectories by using the FFS is 8.5 s.

Figures 9(a)–(c) show the acceleration in both axes (a_{x_j} and a_{y_j}), the total (absolute) acceleration (a_j), and the acceleration ($\hat{a}_j$) for each of the three CVAD, with $j = 1, 2, 3$. Figures 9(d)–(f) show the speed in both axis (v_{x_j} and v_{y_j}), and the total speed (v_j) for the same three vehicles. As shown in the figures, the total acceleration profile and also speed profile of each

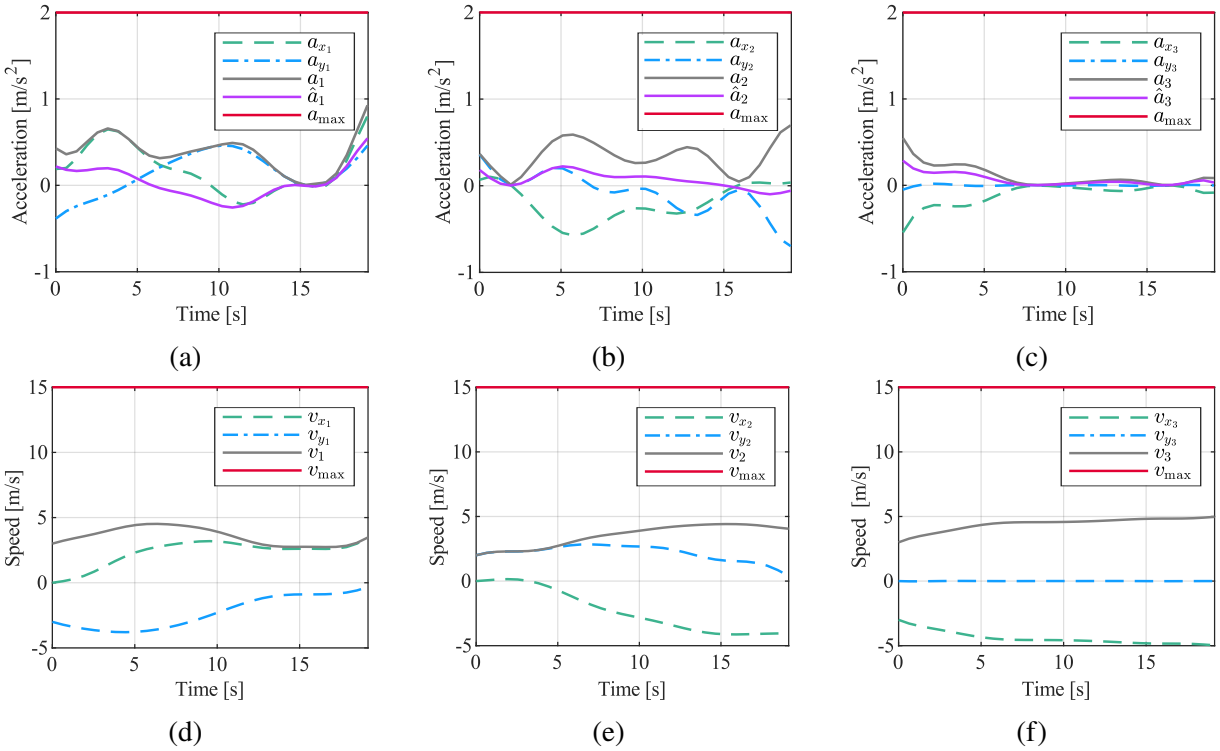


Figure 9 – SC1: FFS method; (a)–(c) acceleration profiles and (d)–(f) speed profiles of CVAD j, where j takes values 1, 2, and 3.

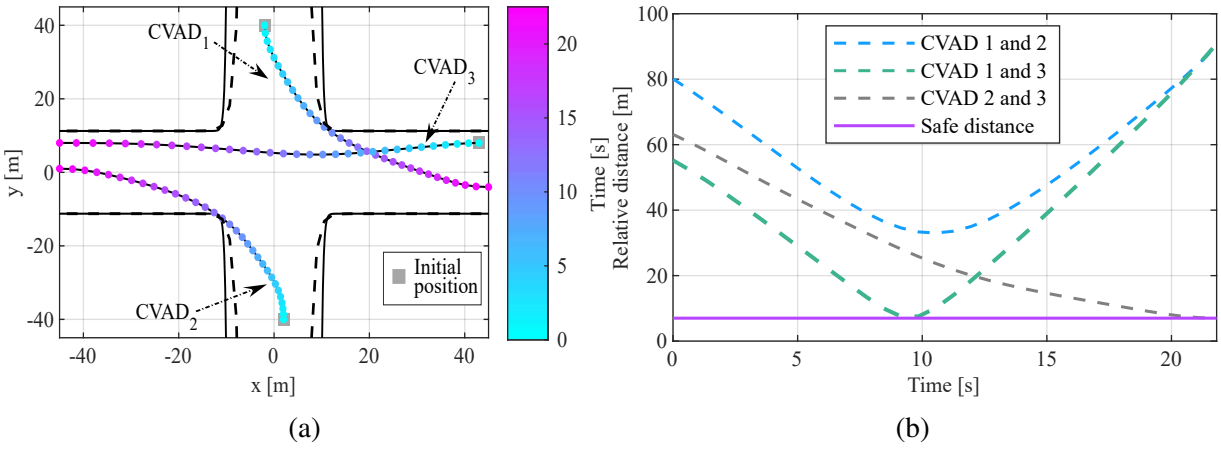
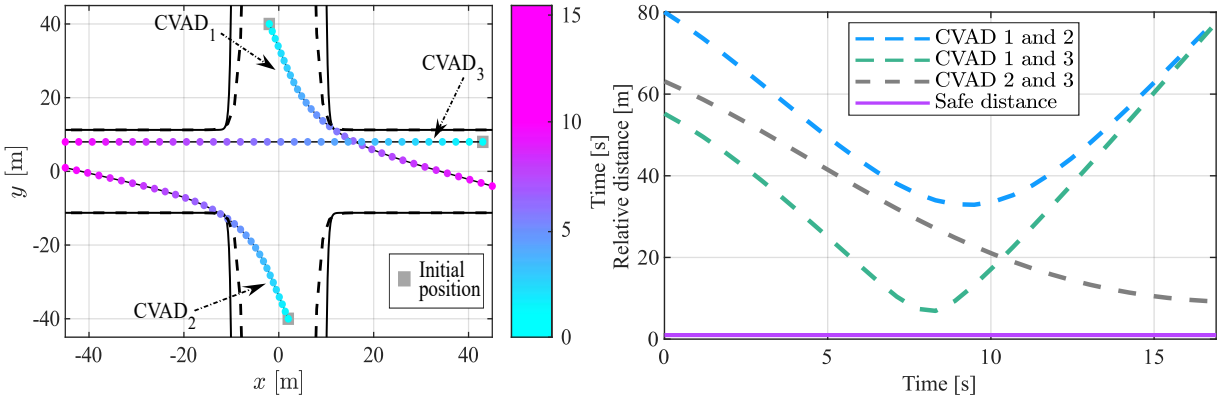


Figure 10 – SC1: FFS method; (a) trajectories of three CVAD and (b) safe distance between them with $d_{\mathrm{s}} = 7$ m.

vehicle are far below the maximum total acceleration and maximum speed values, respectively, which satisfies the existing constraints at the plaza. In addition, the profiles are smooth, as expected due to the minimization of Δv. The lower weight on T_{f} and the low initial and final speeds contribute to the low speeds observed.

To evaluate the effectiveness of the collision avoidance constraint, we present in Figure 10 the numerical results of three CVAD based on a value of d_{s} that differ from the one in Figure 8. We select a relatively large value of $d_{\mathrm{s}} = 7$ m to emphasize the impact of d_{s} on the results. In this case, the completion time is obtained as $T_{\mathrm{f}} = 21.3$ s. We observe in Figure 10a a slightly different behaviour of CVAD$_3$. In particular, CVAD$_3$ is not allowed to travel through a straight



(a) (b)

Figure 11 – SC2: Bézier method; (a) trajectories of three CVAD and (b) safe distance between them with $d_{\mathrm{s}} = 1$ m.

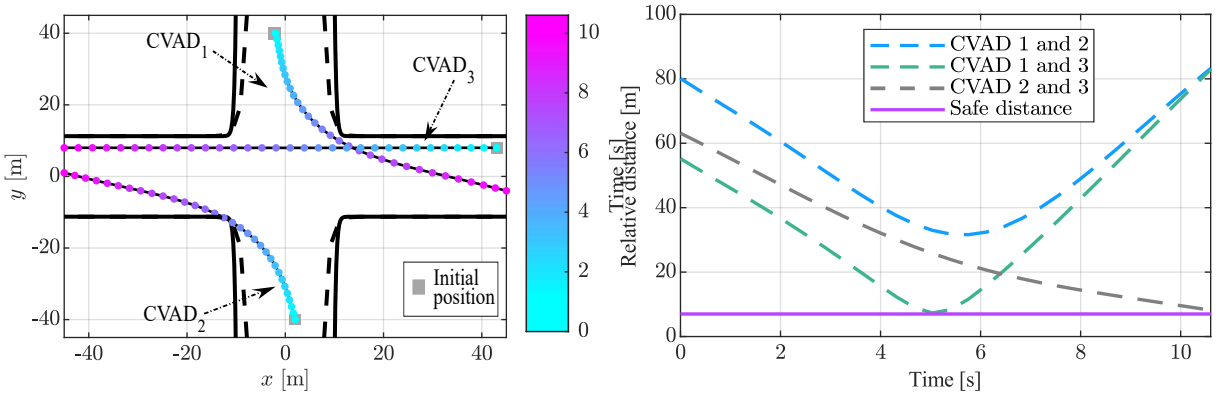
path (compare to Figure 8a). It can also be seen in Figure 10b that the collision avoidance constraint avoid CVAD_1 and CVAD_3 to collide in the time interval [9, 10.8] seconds, and subsequently, the distance between the vehicles (green dashed line) does not go below the safe distance line (solid purple line). We conclude that the smaller d_{s} leads to faster trips while in contrast, a large d_{s} can lead to a conservative solution.

4.3.3 Scenario 2 (SC2): Analysis of the trajectories obtained by the Bézier method

The optimized trajectories generated by the solution of the NLP problem based on the Bézier method are illustrated in Figure 11a. Figure 11b shows the distances between every two CVAD. It can be observed that, at all times, a minimum safe distance is maintained between all CVAD, accordingly, the vehicles distances remain above d_{s} by a large margin for this scenario. The solution of the NLP problem (41) for $w_1 = 4$, $w_2 = 2$, $m = 30$ and $n_{\mathbf{z}} = 8$ resulted in a completion time, $T_{\mathrm{f}} = 14.5$ s, a total speed increment, $\Delta v = 13.4$ m/s, and an objective function value of 90.8. The computation time for generating the trajectories by using the Bézier method is 4.5 s.

We experimented with different combinations of values for m, w_1, w_2 and $n_{\mathbf{z}}$. Varying the values of w_1 and w_2 had more influence on the total speed increment than the completion time. When the number of DPs (m) is increased there is an expected increase in computation time. Despite the corresponding increase in total speed increment, completion times also increase, suggesting that worse local minima are found for higher values of m, i.e., trajectories in longer paths result. Finally, increasing $n_{\mathbf{z}}$ also increases the computation time without sensible improvements in the other measures. Small values of m and $n_{\mathbf{z}}$ may result in better values of Δv, T_{f}, and computation time. However, the trajectories might not be smooth and may also lead to infeasible instances of the NLP problem. Similar results to those obtained with the FFS method, as shown in Appendix A.4, were observed.

To evaluate the efficacy of the collision avoidance constraint, we present in Figure 12 the numerical results of three CVAD based on a larger value of safe distance, $d_{\mathrm{s}} = 7$ m. We observe



(a) (b)

Figure 12 – SC2: Bézier method; (a) trajectories of three CVAD using Bézier method; and (b) distance between them with $d_{\mathrm{S}} = 7$ m.

in Figure 12a a slightly different behavior of CVAD_1, which keeps more distance from the corner of the IB (compared to Figure 11a) due to the activation of collision avoidance constraints. It can also be seen in Figure 12b that the collision avoidance constraint avoids the collision between CVAD_1 and CVAD_3 at around $t = 5$ s, and subsequently, the distance between the vehicles (green dashed line) does not go below the safe distance line (solid purple line).

4.3.4 Scenario 3 (SC3): Comparative Analysis; FFS vs. Bézier Methods

In this section, a comparison is made between the results obtained using the FFS and Bézier curve methods. The optimized trajectories generated by the solution of the NLP based on the Bézier and FFS methods are illustrated in Figure 13. The colored disks corresponding to the cool colormap indicate DPs and time of the Bézier method (T_{B}), and the colored disks corresponding to the warm colormap indicates DPs and time of the FFS method (T_{F}). The gray squares are the initial positions for each CVAD. The solid thick black lines show the boundary of each IB whose approximations are presented by black dashed lines. For this particular scenario,

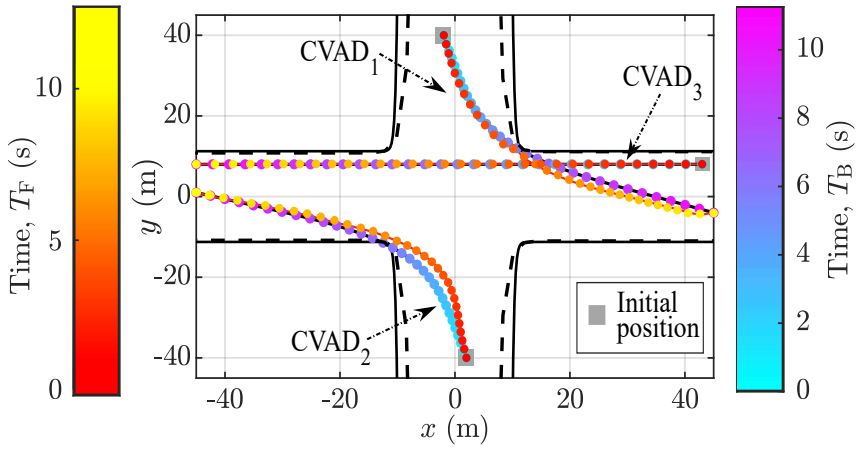


Figure 13 – SC3: FFS vs. Bézier; trajectories of three CVAD using Bézier method (T_{B}) and FFS method (T_{F}) with $d_{\mathrm{S}} = 1$ m.

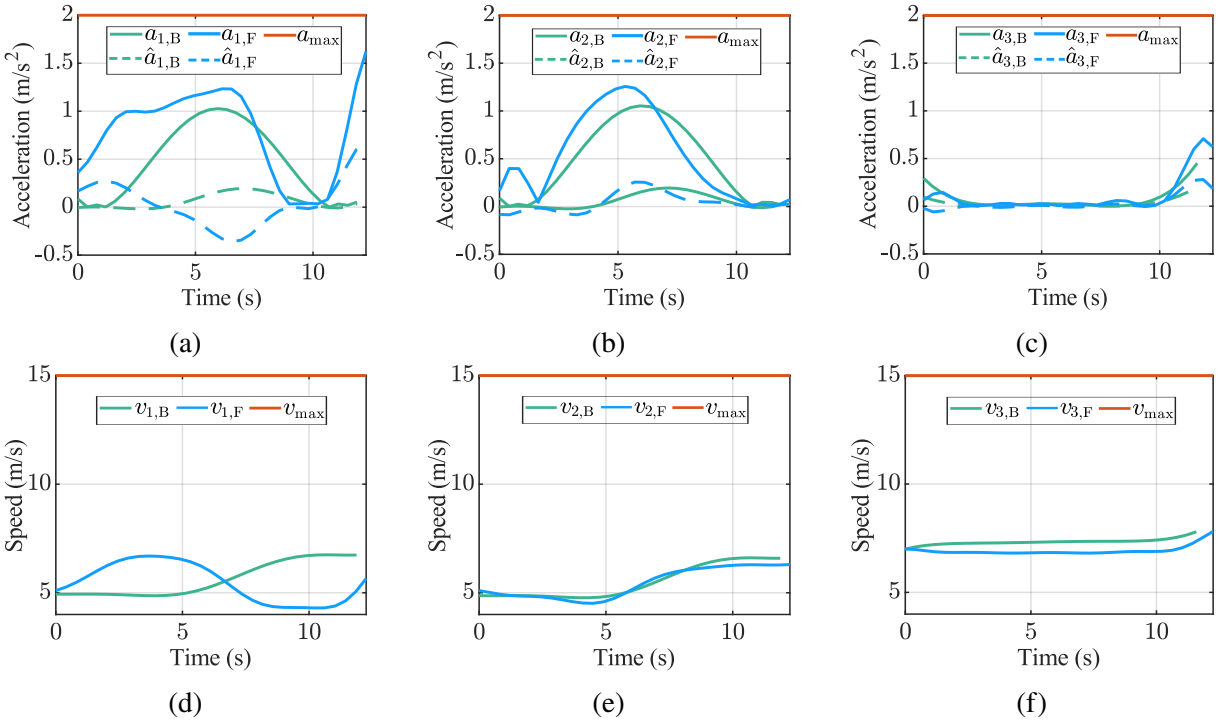


Figure 14 – SC3: FFS vs. Bézier; (a)–(c) acceleration profiles and (d)–(f) speed profiles of CVAD j, where j takes values 1, 2, and 3.

the trajectories of both methods deviate from what would be expected in a path-based method and it is clear that the CVAD follow free trajectories. Noteworthy, the followed paths of the Bézier method seem to approach the paths of minimum distance compared to the FFS method.

Figures 14(a), (b), and (c) show the total (absolute) acceleration (a_j), and the acceleration ($\hat{a}_j$) for each of the three CVAD, with $j = 1, 2, 3$, for both Bézier and FFS methods. Figures 14(d), (e), and (f) show the speed (v_j) for the same three vehicles. The total acceleration and speed profiles of each vehicle are far below the maximum total acceleration and maximum speed values, respectively, which satisfies the existing constraints at the plaza. In addition, the profiles of both methods are smooth, as expected due to the minimization of Δv. However, the acceleration and speed profiles of the trajectories generated with the FFS method exhibit more variation compared to the ones with the Bézier method. Accordingly, we can conclude that the Bézier method is capable of providing more comfortable vehicle movement with less computation time.

The numerical results derived from the solution of the NLP problem (28) for the FFS method and NLP problem (41) for the Bézier method is presented in Table 4. The table compares the total speed increment (Δv), completion time (T_f), objective function ($\mathcal{J}$), and computation time (T_c) of three CVAD. Notably, the completion time is defined as the duration required for all CVAD to cross the intersection plaza and reach their respective destinations. As seen in Table 4, the computation time of the Bézier method is lower than with the FFS method due to the smaller number of decision variables in the first method. Moreover, smaller total speed increment and completion time were obtained with the Bézier method.

Table 4 – SC3: numerical results of Bézier and FFS methods.

Method	Δv (m/s)	T_f (s)	$\mathcal{J}$	T_c (s)
Bézier	14.1	11.5	93.5	4.3
FFS	18.2	12.3	97.4	7.9

4.4 CONCLUSION

The ITOP has been addressed using the FFS and Bézier representations along with discretization strategies. These methods offer notable benefits such as fast computation speed and efficient generation of feasible trajectories, but, they are not fast enough for real-time applications. Although the solutions obtained from these methods serve as excellent initial estimates for direct optimal control techniques, they are considered to be near-optimal due to their sole parameterization of the states and not the states and control signals.

In the forthcoming chapter, our objective is to address the limitations of the proposed methodologies and effectively solve the SPIC problem with improved efficiency and reliability. To achieve this, we present an enhanced formulation of the SPIC problem that, in contrast to the ITOP formulation, incorporates various real-world considerations, including:

- Vehicle dynamics: incorporate a kinematic bicycle model to emulate the actual behavior of vehicles with the flexibility to accommodate other vehicle models.
- Continuous arrival of vehicles: take into account a continuous flow of the CVAD approaching the intersection plaza.
- Separated travel time: adapt the SPIC problem formulation to assign individual travel times to each vehicle.
- Initial and final states: take the current states for initial conditions, while leaving the final conditions free.
- Improve safety and plaza boundary constraints: develop sufficiently accurate mathematical constraints to prevent potential collisions.
- Solution methodology: devise a problem-specific algorithm based on model predictive control as a solution methodology for the SPIC problem.

5 A MODEL PREDICTIVE APPROACH FOR THE SPIC PROBLEM

The aim of this chapter is to present a new formulation for the signal-free path-free intersection control (SPIC) problem and introduce a novel method and algorithm that offer an effective solution for addressing the SPIC problem. As explored in Chapter 4, the ITOP employed multiple exponential functions as algebraic equations to model the intersection plaza geometry. However, this constraint modeling approach is not efficient enough due to some difficulties such as differentiability at the origin, nonlinearity, and so on. Furthermore, the dynamics of a particle used to represent vehicle dynamics in the previous formulation are not entirely realistic for real-world scenarios. In light of these limitations, this chapter presents a novel formulation for the SPIC problem that addresses these difficulties and incorporates various real-world considerations.

In addition to the aforementioned modeling limitations, the FFS and Bézier curve methods used to convert the ITOP into a finite dimension NLP problem also have some limitations. As discussed in Chapter 4, although these methods provide fast and collision-free solutions for the ITOP, their solutions are near-optimal and are conducted offline. A common ground of these methods is that their applications are limited to an existing set of vehicles with predefined initial and final states. Furthermore, the requirement for CVAD to have identical completion times and the absence of continuous vehicle arrival render these methods impractical in real-world scenarios. Additionally, the non-convex nature of the FFS and Bézier curve methods poses challenges in convexifying them, and also their computational demand increase exponentially with the number of vehicles, resulting in prolonged computation times. Consequently, it is crucial to develop a solution strategy for the new formulation of the SPIC problem that overcomes the limitations of the methods presented in Chapter 4.

To do so, we propose an optimization-based receding horizon approach that is formulated as a nonlinear model predictive contouring control (NMPCC) (LAM; MANZIE; GOOD, 2010; LINIGER; DOMAHIDI; MORARI, 2015; SCHWARTING et al., 2018). In particular, we extend and tailor the standard NMPCC method (LAM; MANZIE; GOOD, 2010) to address the solution of the SPIC problem, which we refer to as extended NMPCC (ENMPCC). The ENMPCC method generates collision-free and optimal CVAD trajectories based on time-independent and continuously differentiable reference paths, with a focus on maximizing progress along the paths. To guarantee V2V collision avoidance, the vehicles are modeled as polytopic sets, where each set is required to maintain a minimum safe distance from the other sets. The nonlinear and non-convex nature of the SPIC problem, arising from the vehicle model and safety constraints, presents computational challenges. We employ linear time-varying (LTV) models obtained through the linearization of nonlinear functions. Additionally, we leverage duality theory (BOYD, S. P.; VANDENBERGHE, 2004) for smoothing the collision avoidance constraints and enabling the utilization of efficient solution approaches.

In Section 5.1, we describe the modeling techniques employed to characterize the intersection plaza, vehicle dynamics, reference paths, and safety constraints. Section 5.2 provides

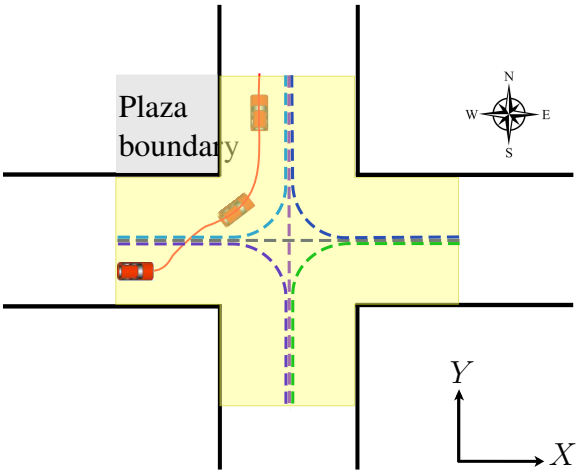


Figure 15 – A four-leg signal-free intersection as a plaza (yellow area) with one CVAD traversing on a path-free; the solid black lines mark the plaza boundaries and the dashed lines are the different reference paths.

an overview of the SPIC problem formulation and introduces the ENMPCC and LTV-ENMPCC methods. In 5.3 the SPIC simulator (SPIC-Sim), which serves as a software tool to implement the proposed methods is introduced. In Section 5.4, we present the results of numerical simulations conducted across various scenarios. Lastly Section 5.5 concludes this chapter.

5.1 PROBLEM MODELING

In this section, we present a new formulation of the SPIC problem that takes into account several real-world considerations, distinguishing it from the ITOP formulation. The SPIC problem introduces a revised model for the intersection plaza boundaries and adopts the kinematic bicycle model as the vehicle model, but, with the flexibility to include other vehicle models. Additionally, the SPIC problem incorporates additional reference paths, employs a polytopic representation for each vehicle, and formulates V2V collision avoidance constraints as a dual optimization problem. Moreover, the SPIC problem models the plaza's boundaries and the vehicle's dimensions as exact sizes with no approximation or enlargement.

5.1.1 Intersection Plaza Model

In Figure 15, the plaza is depicted as a yellow area with boundaries delimited by black lines, representing a four-leg intersection. However, different geometries can be considered for representing the plaza. The X and Y axes denote the central lines of the plaza. In the figure, a vehicle is traversing from its initial position toward its destination. Within the plaza, the vehicle is free to adjust its trajectory, shown by the solid red line, without being bound to predetermined paths or traffic lanes.

5.1.2 Vehicle Model

There are various models to describe a vehicle's dynamics, from the simple unicycle model (SICILIANO; KHATIB; KRÖGER, 2008) to sophisticated vehicle models (ULSOY; PENG; CAKMAKCI, 2012). We model each vehicle within the plaza by using a nonlinear kinematic bicycle model. The bicycle model is a widely used model for simplifying the dynamics of vehicle motion by reducing the number of wheels involved. It lumps the left and right wheels into a pair of single wheels in the middle of the vehicle's axles (RAJAMANI, 2011).

In this work, two configurations of the bicycle model are employed as vehicle models; (i) a kinematic bicycle model which has rear-wheel driving, see Figure 16, and (ii) a kinematic bicycle model which has a center of gravity (CG) driving, see Figure 17. By using Euler discretization, both models are discretized with the discrete-time index k and the time step Δt. Furthermore, each vehicle is identified by its index $i \in \mathcal{V} = \{1,2,\ldots,N_\mathrm{v}\}$ with N_v the number of vehicles.

5.1.2.1 Kinematic bicycle model with rear-wheel driving

The discretized equations of motion described by five state equations are as follows (RAJAMANI, 2011):

$$\begin{aligned} x_{i,k+1} &= x_{i,k} + \Delta t\, v_{i,k} \cos(\psi_{i,k}), \\ y_{i,k+1} &= y_{i,k} + \Delta t\, v_{i,k} \sin(\psi_{i,k}), \\ \psi_{i,k+1} &= \psi_{i,k} + \Delta t\, \frac{v_{i,k} \tan(\delta_{i,k})}{L}, \\ v_{i,k+1} &= v_{i,k} + \Delta t\, u^1_{i,k}, \\ \delta_{i,k+1} &= \delta_{i,k} + \Delta t\, u^2_{i,k}, \end{aligned} \tag{43}$$

where the i-th vehicle state vector is denoted by $\mathbf{x}_{i,k} = [x_{i,k}, y_{i,k}, \psi_{i,k}, v_{i,k}, \delta_{i,k}]^\top$, in which $x_{i,k}$ (m) and $y_{i,k}$ (m) are the longitudinal and lateral position of each CVAD in the CCS, respectively, $\psi_{i,k}$ is the orientation angle (rad), $\delta_{i,k}$ is the steering angle (rad), and $v_{i,k}$ (m/s) is the speed of the vehicle. The control input vector is denoted by $\mathbf{u}_{i,k} = [u^1_{i,k}, u^2_{i,k}]^\top$, in which $u^1_{i,k} = a_{i,k}$ (m/s^2) and $u^2_{i,k} = \dot{\delta}_{i,k}$ (rad/s) are the acceleration and steering angle rates of the vehicle, respectively. The vehicle's wheelbase distance, the distance between the front and rear wheels, is represented as L (m).

The bicycle model (43) is sensitive to large lateral accelerations, thus minimum and maximum values are imposed:

$$-0.5 l_\mathrm{v} \mu g \leq v^2_{i,k} \tan(\delta_{i,k}) \leq 0.5 l_\mathrm{v} \mu g, \tag{44}$$

with l_v the vehicle length, μ the friction coefficient, and g the gravitational acceleration. Some of the system states and both inputs are bounded to minimum and maximum values:

$$\begin{aligned} v_\mathrm{min} &\leq v_{i,k} \leq v_\mathrm{max}, \\ \delta_\mathrm{min} &\leq \delta_{i,k} \leq \delta_\mathrm{max}, \\ \mathbf{u}_\mathrm{min} &\leq \mathbf{u}_{i,k} \leq \mathbf{u}_\mathrm{max}. \end{aligned} \tag{45}$$

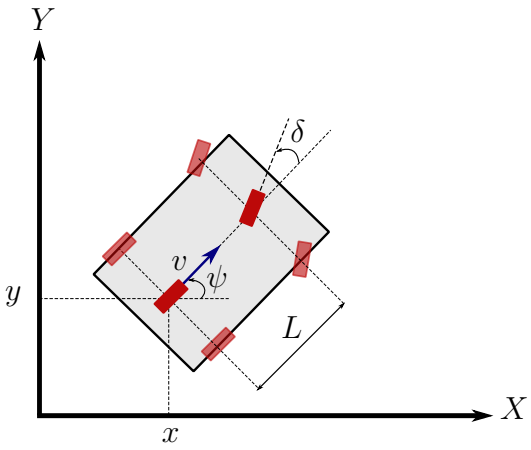


Figure 16 – Kinematic bicycle model with rear-wheel driving; the left and right wheels lumped into single wheels in the middle of the vehicle's axles; x and y the longitudinal and lateral position of each CVAD, ψ the orientation, v the speed, δ the front wheel steering angle, and L is the vehicle's wheelbase distance.

where $v_{\min}$, $v_{\max}$, $\delta_{\min}$, $\delta_{\max}$, $\mathbf{u}_{\min}$, and $\mathbf{u}_{\max}$ are the lower and upper boundaries of speed, steering angle, and control inputs, respectively.

5.1.2.2 Kinematic bicycle model with a center of gravity driving

The discretized bicycle model of a vehicle is given by (KONG et al., 2015; POLACK et al., 2017):

$$\begin{aligned}
x_{i,k+1} &= x_{i,k} + \Delta t\, v_{i,k} \cos(\psi_{i,k} + \beta_{i,k}),\\
y_{i,k+1} &= y_{i,k} + \Delta t\, v_{i,k} \sin(\psi_{i,k} + \beta_{i,k}),\\
\psi_{i,k+1} &= \psi_{i,k} + \Delta t\, \frac{v_{i,k} \sin(\beta_{i,k})}{l_{\mathrm{r},i}},\\
v_{i,k+1} &= v_{i,k} + \Delta t\, a_{i,k},\\
\beta_{i,k} &= \tan^{-1}\left(\frac{l_{\mathrm{r},i}}{l_{\mathrm{f},i} + l_{\mathrm{r},i}} \tan(\delta_{i,k})\right),
\end{aligned} \tag{46}$$

where the i-th vehicle state vector is $\mathbf{x}_{i,k} = [x_{i,k}, y_{i,k}, \psi_{i,k}, v_{i,k}]^\top$, in which $x_{i,k}$ (m) and $y_{i,k}$ (m) are the longitudinal and lateral position of each CVAD, respectively, $\psi_{i,k}$ (rad) is the orientation angle, and $v_{i,k}$ (m/s) and $\beta_{i,k}$ (rad) are the speed and the slip angle at the CG of the vehicle. The i-th vehicle control vector is $\mathbf{u}_{i,k} = [a_{i,k}, \delta_{i,k}]^\top$, in which $a_{i,k}$ (m/s^2) is the acceleration and $\delta_{i,k}$ (rad) is the front wheel steering angle of the vehicle. The distances from the front and rear axles to the CG of the i-th vehicle are $l_{\mathrm{f},i}$ (m) and $l_{\mathrm{r},i}$ (m), respectively.

To ensure compliance with the dynamic limitations of the vehicles, the following constraints are enforced for each CVAD:

$$\begin{aligned}
v_{\min} &\leq v_{i,k} \leq v_{\max},\\
\delta_{\min} &\leq \delta_{i,k} \leq \delta_{\max},\\
a_{\min} &\leq a_{i,k} \leq a_{\max},
\end{aligned} \tag{47}$$

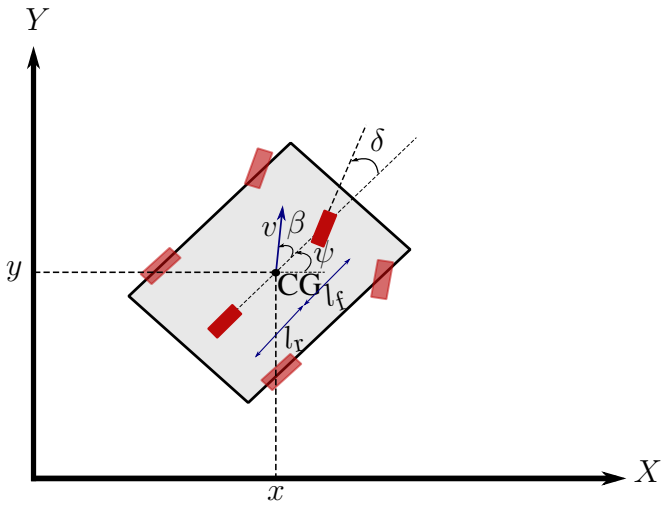


Figure 17 – Kinematic bicycle model with a center of gravity (CG) driving; the left and right wheels lumped into single wheels in the middle of the vehicle's axles; ψ the orientation, v the speed, and β the slip angle at the CG, given by x and y, respectively, the longitudinal and lateral position of each CVAD; δ the front wheel steering angle; and the distances from the front and rear axles to the CG are l_{f} and l_{r}.

where $v_{\min}$, $v_{\max}$, $\delta_{\min}$, $\delta_{\max}$, $a_{\min}$, and $a_{\max}$ are the lower and upper boundaries of speed, steering angle, and acceleration.

The kinematic bicycle models, as presented in (43) and (46), are a suitable choice for our plaza application; these models can be implemented in low-speed vehicles commonly found at intersections, allow for designing controllers for stop-and-go scenarios frequently encountered in urban driving, and are less computationally demanding than methods that employ vehicle tire models (KONG et al., 2015). However, it should be noted that model (43), with rear-wheel drive, may pose safety concerns regarding potential collisions between vehicles. This issue will be further examined and discussed in the subsequent simulation.

5.1.3 Reference Path Model

We use the road center lines as reference paths. Six different paths cover all possible movements at the intersection plaza in Figure 15. Each reference path is then followed by the vehicles according to their traversal intention at the intersection. As an example, the light blue dashed line shows the turning reference path for the vehicle approaching the plaza from the west intended to make a left turn. The other reference paths follow a similar description for the remaining directions. Note that we utilize the reference paths for computing the vehicles' progress and path error measures.

The goal is to control the motion of the CVAD along reference paths while minimizing the distance between the current position of the CVAD and the reference path, which is known as the contouring control problem (CCP) (KOREN, Yoram; LO, 1991). In the following, we review some concepts of the CCP, including the parameterization of the reference path and the definition of the path error measures based on two orthogonal components, the lateral (contouring) and

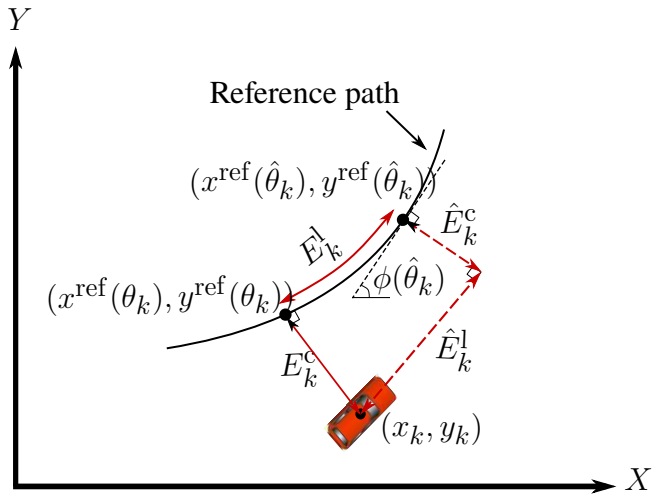


Figure 18 – A reference path corresponding to the center line of a road used for computing the vehicles' progress; lateral error (E^{c}_k) and longitudinal error (E^{l}_k), shown in solid red lines and their approximation $\hat{E}^{\text{c}}_k$ and $\hat{E}^{\text{l}}_k$ shown in dashed red lines. If $\hat{E}^{\text{l}}_k$ is zero, then E^{l}_k is also zero and therefore $\hat{\theta}_k$ is equal to θ_k.

longitudinal (lag) errors (LAM; MANZIE; GOOD, 2010).

5.1.3.1 Reference Path Parameterization

Each CVAD at position $\mathbf{p}_k = (x_k,\, y_k)$ follows a continuously differentiable and time-independent reference path $\mathbf{p}^{\text{ref}}(\theta) = (x^{\text{ref}}(\theta), y^{\text{ref}}(\theta))$, parameterized by the path parameter θ. It is assumed that the parameterization of the path satisfies $\mathrm{d}s/\mathrm{d}\theta = 1$, where s (m) determines the traveled distance by each CVAD along the path which allows us to compute the progress of each vehicle along the path. The parameterization of curves by arc length is nontrivial. However, methods like splines (WANG; KEARNEY; ATKINSON, 2002) or Bézier curves (ZHU, Y.; HAN, X.; HAN, J., et al., 2012) can approximate the arc length properly.

5.1.3.2 Lateral and Longitudinal Errors

Contouring accuracy is measured by the lateral error, that is the shortest distance of the actual position of the vehicle $\mathbf{p}_k$ from the reference path, $\mathbf{p}^{\text{ref}}(\theta)$, see Figure 18. The value of the path parameter for which the distance between $\mathbf{p}_k$ and $\mathbf{p}^{\text{ref}}(\theta)$ is minimal can be obtained by the following projection:

$$\min_{\theta} \ \|\mathbf{p}_k - \mathbf{p}^{\text{ref}}(\theta)\|_2, \tag{48}$$

where the minimizer at time step k is θ_k as shown in Figure 18. The lateral error is then given by (KOREN, Yoram; LO, 1991; LAM; MANZIE; GOOD, 2010):

$$E^{\text{c}}_k = \sin(\phi(\theta_k))(x_k - x^{\text{ref}}(\theta_k)) - \cos(\phi(\theta_k))(y_k - y^{\text{ref}}(\theta_k)), \tag{49}$$

where $\phi(\theta_k)$ is the tangent angle to the reference path with respect to the x axis and given as:

$$\phi(\theta_k) = \arctan\left(\frac{\nabla y^{\text{ref}}(\theta_k)}{\nabla x^{\text{ref}}(\theta_k)}\right), \tag{50}$$

with

$$\nabla y^{\mathrm{ref}}(\theta_k) = \mathrm{d}y^{\mathrm{ref}}(\theta)/\mathrm{d}\theta\Big|_{\theta=\theta_k}, \quad \nabla x^{\mathrm{ref}}(\theta_k) = \mathrm{d}x^{\mathrm{ref}}(\theta)/\mathrm{d}\theta\Big|_{\theta=\theta_k}.$$

However, given that (48) is an optimization problem itself, this formulation of lateral error is not appropriate for use in an online optimization problem. Then, an approximation $\hat{\theta}_k$ of θ_k was introduced with dynamics (PATRIKALAKIS; MAEKAWA, 2002; LAM; MANZIE; GOOD, 2010):

$$\hat{\theta}_{k+1} = \hat{\theta}_k + \Delta t\, \nu_k,\ \nu_k \in [0, \nu_{\max}],\ \nu_{\max} > 0, \tag{51}$$

where $\hat{\theta}_k$ is the approximated value of the path parameter at time k that can be interpreted as the vehicle progress along the reference path, ν_k is a virtual input that controls the evolution of $\hat{\theta}_k$, and $\nu_{\max}$ is a maximum value for the virtual input.

The optimal path parameter θ_k and its approximation $\hat{\theta}_k$ are linked by introducing the longitudinal error $E_k^{\mathrm{l}} = \|\theta_k - \hat{\theta}_k\|$. The longitudinal error is also approximated since θ_k is not known at the optimization time. Consequently, the approximated lateral error ($\hat{E}_k^{\mathrm{c}}$) and approximated longitudinal error ($\hat{E}_k^{\mathrm{l}}$) are defined as follows:

$$\hat{E}_k^{\mathrm{c}} = \sin(\phi(\hat{\theta}_k))(x_k - x^{\mathrm{ref}}(\hat{\theta}_k)) - \cos(\phi(\hat{\theta}_k))(y_k - y^{\mathrm{ref}}(\hat{\theta}_k)), \tag{52}$$

$$\hat{E}_k^{\mathrm{l}} = -\cos(\phi(\hat{\theta}_k))(x_k - x^{\mathrm{ref}}(\hat{\theta}_k) - \sin(\phi(\hat{\theta}_k))(y_k - y^{\mathrm{ref}}(\hat{\theta}_k)), \tag{53}$$

where $\phi(\cdot)$ is defined in (50). It can be seen in Figure 18 that $\theta_k \approx \hat{\theta}_k$ if $\hat{E}_k^{\mathrm{l}} \approx 0$.

5.1.4 V2V Collision Avoidance Constraints

Ensuring the safety of the CVAD remains a critical concern while traveling through urban roads, especially in complex intersections. Hence, the modeling of collision avoidance constraints plays a pivotal role in cooperative trajectory optimization, as they determine the resultant collision-free trajectories of the CVAD. In this context, we examine two potential approaches that are based on the geometrical representation of vehicles; one employing circular definitions (see Figure 19) and the other employing rectangular polytopes (see Figure 20). The choice of geometric representation directly impacts the algorithms that have to be utilized for solving the problem.

5.1.4.1 Circular Representation of the road region occupied by each CVAD

When using a circular approximation, the road region occupied by each CVAD is modeled as a circle of appropriate radius. Figure 19 depicts two vehicles on the road and the parameters related to collision avoidance. The thick black curves are the left and right boundaries of the road and the dashed gray curve is a reference path. The schematic of constraint on the collision between two vehicles i and j based on the kinematic bicycle models (43) and (46) is shown in Figure 19(a) and Figure 19(b), respectively. The solid circle in Figure 19(a) and Figure 19(b) shows the area for which the constraint should be satisfied. If the coordinate of vehicle i or any other vehicle enters the solid circle, the constraint would be violated.

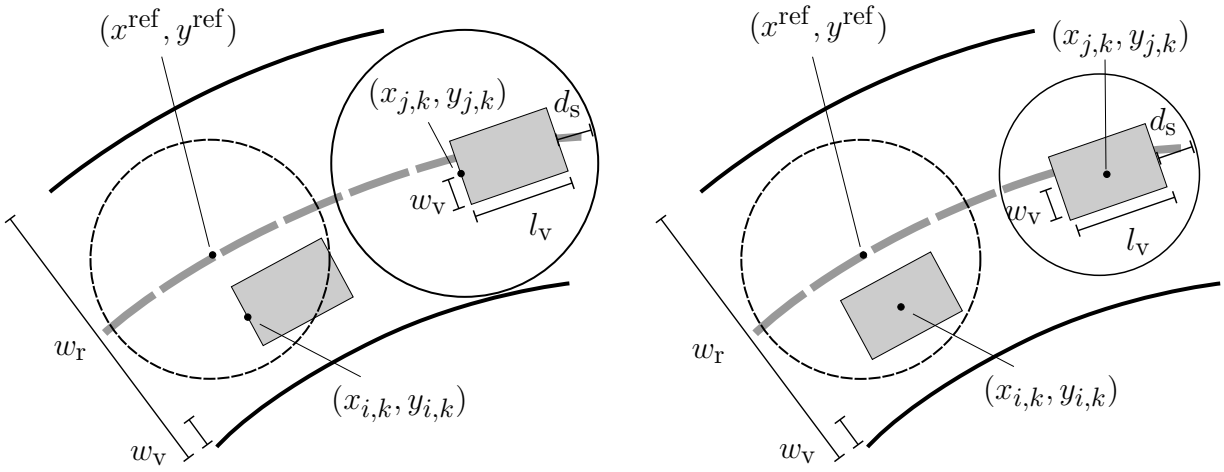


Figure 19 – Schematic of a road with two vehicles and parameters related to collision avoidance, (a) based on the bicycle model with rear-wheel driving, (43), and (b) based on the bicycle model with a center of gravity driving, (46); the dashed circle around point $(x^{\mathrm{ref}},\, y^{\mathrm{ref}})$ shows the area within which the vehicle satisfies the boundary constraint; the solid circle shows the area for which the V2V collision avoidance constraints should be satisfied.

For two circles that belong to different vehicles with center positions $\mathbf{p}_i = [x_{i,k}, y_{i,k}]^\top$ and $\mathbf{p}_j = [x_{j,k}, y_{j,k}]^\top$, the collision avoidance constraint reads as:

$$\mathrm{dist}(\mathbf{p}_i, \mathbf{p}_j) \geq d_{\mathrm{s}} + l_{\mathrm{v}}, \tag{54}$$

with $i, j \in \mathcal{V}$, $i < j$, d_{s} being the minimum safe distance between vehicles, l_{v} being the vehicle length, and $\mathrm{dist}(\mathbf{p}_i, \mathbf{p}_j)$ is the Euclidean distance between vehicles i and j and defined as:

$$\|\mathbf{p}_i - \mathbf{p}_j\|_2 \;=\; \sqrt{\left(x_{i,k} - x_{j,k}\right)^2 + \left(y_{i,k} - y_{j,k}\right)^2}. \tag{55}$$

The constraint (54) is nonlinear and non-convex and defines the collision-free region outside of a circle with the center at the j-th vehicle as shown in Figure 19. When opting for a larger number of smaller circles, a more precise representation of the vehicle can be achieved. However, this approach introduces additional constraints and increases the complexity of the problem.

5.1.4.2 Polytopic Representation of the road region occupied by each CVAD

When using a rectangular approximation, the road region occupied by each CVAD is modeled as a polytopic set $\mathcal{P}$, which is a bounded intersection of a finite number of half-spaces. This representation uses dual optimization to formulate the collision avoidance constraints. For two polytopic sets $\mathcal{P}_i$ and $\mathcal{P}_j$ that belong to different vehicles the collision avoidance constraint reads:

$$\mathrm{dist}(\mathcal{P}_i, \mathcal{P}_j) \geq d_{\mathrm{s}}. \tag{56}$$

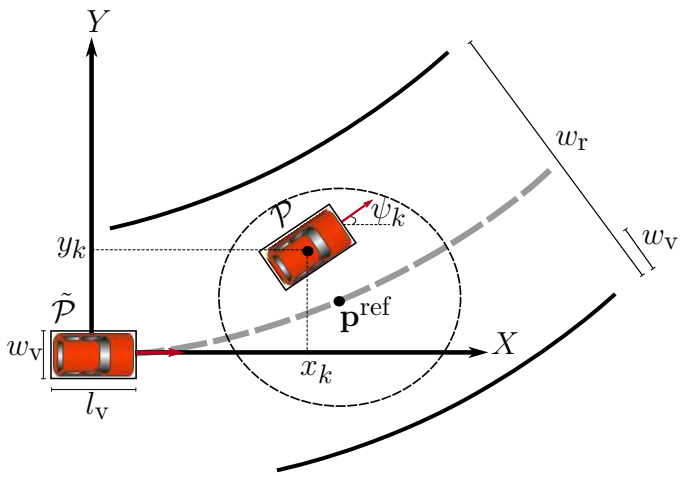


Figure 20 – Schematic of a road; the thick black curves are the left and right boundaries of the road and the dashed gray curve is a reference path; the dashed circle around point $\mathbf{p}^{\mathrm{ref}} = [x^{\mathrm{ref}}, y^{\mathrm{ref}}]^\top$ shows the area within which the vehicle satisfies the boundary constraint that forces the vehicle to remain within the road boundaries. $\mathcal{P}$ is the polytopic representation of the vehicle and its transformation from $\tilde{\mathcal{P}}$ to $\mathcal{P}$ with ψ_k as the vehicle orientation.

The initial pose of each CVAD can be mathematically described by a two-dimensional rectangular polytope $\tilde{\mathcal{P}}_i$ as shown in Figure 20 (ZHANG, X.; LINIGER; BORRELLI, 2020):

$$\tilde{\mathcal{P}}_i = \{\mathbf{p}_i \in \mathbb{R}^2 \mid \tilde{\mathbf{A}}_i \, \mathbf{p}_i \leq \tilde{\mathbf{b}}_i\}, \tag{57}$$

where

$$\begin{aligned} \tilde{\mathbf{A}}_i &= \begin{bmatrix} \mathbf{I}_{2\times 2} & -\mathbf{I}_{2\times 2} \end{bmatrix}^\top, \\ \tilde{\mathbf{b}}_i &= \begin{bmatrix} \frac{l_{\mathrm{v}}}{2} & \frac{w_{\mathrm{v}}}{2} & \frac{l_{\mathrm{v}}}{2} & \frac{w_{\mathrm{v}}}{2} \end{bmatrix}^\top, \end{aligned} \tag{58}$$

with l_{v} being the length and w_{v} the width of the CVAD. As each CVAD i moves along the road, polytope $\tilde{\mathcal{P}}_i$ undergoes affine transformations including rotation and translation. Hence, the transformed polytope $\mathcal{P}_i$ is a function of the states and defined as:

$$\mathcal{P}_i(\mathbf{x}_{i,k}) = \{\mathbf{p}_i \in \mathbb{R}^2 \mid \mathbf{A}_i(\mathbf{x}_{i,k}) \, \mathbf{p}_i \leq \mathbf{b}_i(\mathbf{x}_{i,k})\}, \tag{59}$$

where

$$\begin{aligned} \mathbf{A}_i(\mathbf{x}_{i,k}) &= \tilde{\mathbf{A}}_i \begin{bmatrix} \cos(\psi_{i,k}) & -\sin(\psi_{i,k}) \\ \sin(\psi_{i,k}) & \cos(\psi_{i,k}) \end{bmatrix}, \\ \mathbf{b}_i(\mathbf{x}_{i,k}) &= \tilde{\mathbf{b}}_i + \mathbf{A}_i(\mathbf{x}_{i,k}) \, \mathbf{p}_i. \end{aligned} \tag{60}$$

This representation is time-varying and is a function of the vehicle states. For clarity and simplicity, in the following, we consider $\mathcal{P}_i = \mathcal{P}_i(\mathbf{x}_{i,k})$, $\mathbf{A}_i = \mathbf{A}_i(\mathbf{x}_{i,k})$, and $\mathbf{b}_i = \mathbf{b}_i(\mathbf{x}_{i,k})$.

To prevent collisions, the intersection of the polytopic sets of vehicles $i,j \in \mathcal{V}$, and $i < j$, should be avoided, that is, $\mathcal{P}_i \cap \mathcal{P}_j = \emptyset$ with $\mathcal{P}_i$ and $\mathcal{P}_j$ being the i-th and j-th CVAD polytopes, respectively. This results in a non-convex problem that is difficult to solve. In the remaining of this subsection, we address this issue by reformulating $\mathcal{P}_i \cap \mathcal{P}_j = \emptyset$ to a smooth and differentiable optimization problem (ZHANG, X.; LINIGER; BORRELLI, 2020).

This optimization problem is defined as the distance between the sets $\mathcal{P}_i$ and $\mathcal{P}_j$ and is written as the following primal problem (BOYD, S. P.; VANDENBERGHE, 2004):

$$\operatorname{dist}(\mathcal{P}_i, \mathcal{P}_j) = \min_{\mathbf{p}_i, \mathbf{p}_j} \{\|\mathbf{p}_i - \mathbf{p}_j\|_2 \,|\, \mathbf{A}_i\mathbf{p}_i \leq \mathbf{b}_i, \mathbf{A}_j\mathbf{p}_j \leq \mathbf{b}_j\}, \tag{61}$$

with

$$\begin{aligned} \mathcal{P}_i &= \{\mathbf{p}_i \in \mathbb{R}^2 \,|\, \mathbf{A}_i\mathbf{p}_i \leq \mathbf{b}_i\}, \\ \mathcal{P}_j &= \{\mathbf{p}_j \in \mathbb{R}^2 \,|\, \mathbf{A}_j\mathbf{p}_j \leq \mathbf{b}_j\}. \end{aligned} \tag{62}$$

To guarantee collision avoidance, not only the intersection of these sets should be empty, but also a minimum safe distance, d_s, must be kept in any direction between them, that is, $\operatorname{dist}(\mathcal{P}_i, \mathcal{P}_j) \geq d_\mathrm{s}$. The optimization problem (61) cannot be directly applied in the general form of the optimal control problem. The reason is that an optimization problem would act as a constraint for another optimization problem. To solve this problem, the duality theory is employed as suggested in (BOYD, S. P.; VANDENBERGHE, 2004; ZHANG, X.; LINIGER; BORRELLI, 2020).

Given problem (61), where $\mathcal{P}_i$ and $\mathcal{P}_j$ are nonempty sets, the following dual problem can be solved instead of primal problem (61) (ZHANG, X.; LINIGER; BORRELLI, 2020):

$$\begin{aligned} \operatorname{dist}(\mathcal{P}_i, \mathcal{P}_j) = \max_{\boldsymbol{\lambda}_{i,j}, \boldsymbol{\lambda}_{j,i}, \mathbf{s}_{i,j}} \; & -\mathbf{b}_i^\top \boldsymbol{\lambda}_{i,j} - \mathbf{b}_j^\top \boldsymbol{\lambda}_{j,i} \\ \text{s.t.} \;\; & \mathbf{A}_i^\top \boldsymbol{\lambda}_{i,j} + \mathbf{s}_{i,j} = \mathbf{0}, \;\; \mathbf{A}_j^\top \boldsymbol{\lambda}_{j,i} - \mathbf{s}_{i,j} = \mathbf{0}, \\ & \|\mathbf{s}_{i,j}\|_2 \leq 1, \;\; -\boldsymbol{\lambda}_{i,j} \leq \mathbf{0}, -\boldsymbol{\lambda}_{j,i} \leq \mathbf{0}, \end{aligned} \tag{63}$$

with $\boldsymbol{\lambda}_{i,j}, \boldsymbol{\lambda}_{j,i}$ and $\mathbf{s}_{i,j} = \mathbf{s}_{j,i}$ being the Lagrange multiplier vectors associated with the inequality constraints and equality constraints, respectively. The derivation of (63) from (61) is provided in Appendix C. Furthermore, problem (63) can be reformulated as the following feasibility problem $\{\exists\, -\boldsymbol{\lambda}_{i,j} \leq \mathbf{0}, -\boldsymbol{\lambda}_{j,i} \leq \mathbf{0}, s_{i,j} : \; -\mathbf{b}_i^\top \boldsymbol{\lambda}_{i,j} - \mathbf{b}_j^\top \boldsymbol{\lambda}_{j,i} \geq d_\mathrm{s}, \; \mathbf{A}_i^\top \boldsymbol{\lambda}_{i,j} + \mathbf{s}_{i,j} = \mathbf{0}, \; \mathbf{A}_j^\top \boldsymbol{\lambda}_{j,i} - \mathbf{s}_{i,j} = \mathbf{0}, \; \|\mathbf{s}_{i,j}\|_2 \leq 1\}$, using the intuition that the optimal value of the dual problem, which represents the distance between $\mathcal{P}_i$ and $\mathcal{P}_j$, is constrained to be greater than d_s.

This approach provides an exact reformulation of non-differentiable collision avoidance constraints (56) into smooth, differentiable constraints by leveraging the strong duality of convex optimization. This approach offers several advantages over alternative methods. Firstly, it avoids the need to solve a mixed integer linear/quadratic programming problem, which can be computationally expensive. Additionally, it eliminates the requirement for linearization, which can restrict the solution space by imposing limitations on collision-avoidance constraints. Furthermore, this method enables the specification of a safe distance between vehicles, rather than relying on separating hyperplanes.

5.1.5 Plaza Boundary Constraints

The SPIC problem must satisfy constraints to prevent the i-th CVAD from violating the intersection boundaries. This section introduces constraints that are distinct from the plaza boundary constraints IB_1 to IB_4 discussed in section 4.1.6. These constraints can be expressed by

denoting $\mathcal{O}_r$ as a polytope set occupied by the r-th intersection boundary, where $r \in \{1, 2, \ldots, h\}$, and h is the number of intersection boundaries. The constraints require that $\mathcal{P}_i$ and $\mathcal{O}_r$ do not intersect, i.e., $\mathcal{P}_i \cap \mathcal{O}_r = \emptyset$. A similar reformulation of V2V collision avoidance constraints discussed in Section 5.1.4.2 can be derived for the intersection plaza boundary constraints. However, it would preserve non-convexity and increase complexity by adding more decision variables and constraints to the optimization problem.

Hence, this section presents a new set of constraints to ensure that vehicles stay within the intersection plaza boundaries. Each vehicle is constrained to travel only within a well-defined circle with the center point moving along the reference path $\mathbf{p}^{\mathrm{ref}} = [x^{\mathrm{ref}}(\hat{\theta}_{i,k}), y^{\mathrm{ref}}(\hat{\theta}_{i,k})]^\top$ with a diameter of the road width, w_{r}, minus the vehicle width, w_{v}, (LEVY; HADDAD, 2021). In Figure 19(a), Figure 19(b), and Figure 20, the dashed circle surrounding the point $\mathbf{p}^{\mathrm{ref}}$ shows the region within which the vehicle satisfies the boundary constraint, even though, by definition, the vehicle is on a line on the diameter of the circle perpendicular to the reference path. The plaza boundaries constraints are then defined as the following convex constraints:

$$\mathrm{dist}(\mathbf{p}_i,\, \mathbf{p}^{\mathrm{ref}}) \leq \frac{w_{\mathrm{r}} - w_{\mathrm{v}}}{2}, \tag{64}$$

with $\mathrm{dist}(\mathbf{p}_i,\, \mathbf{p}^{\mathrm{ref}})$ is the Euclidean distance between the current position of the i-th vehicle and the point $\mathbf{p}^{\mathrm{ref}}$. So, constraint (64) is defined as:

$$\sqrt{(x^{\mathrm{ref}}(\hat{\theta}_{i,k}) - x_{i,k})^2 + (y^{\mathrm{ref}}(\hat{\theta}_{i,k}) - y_{i,k})^2} \leq \frac{w_{\mathrm{r}} - w_{\mathrm{v}}}{2}. \tag{65}$$

with w_{r} being the road width.

5.2 SPIC PROBLEM FORMULATION

Based on the models and constraints introduced in Section 5.1, in this section, we introduce the SPIC problem formulation. Then, we build on a nonlinear MPC formulation suitable for solving SPIC problem, known as nonlinear model predictive contouring control (NMPCC) (LAM; MANZIE; GOOD, 2010; LINIGER; DOMAHIDI; MORARI, 2015). NMPCC, in comparison to standard nonlinear MPC, incorporates an analytical representation of a reference path that is not time-parameterized but rather based on a tailored path parameter (parameterization). The use of the lateral and longitudinal errors in the cost function of the controller is what differentiates the NMPCC controller from the standard nonlinear MPC. In particular, we propose a practical extension of the standard NMPCC (LAM; MANZIE; GOOD, 2010), denoted as extended NMPCC (ENMPCC), to address the solution of the SPIC problem.

5.2.1 SPIC in General Form

The multi-vehicle interaction at the intersection plaza can be considered as a signal-free path-free intersection control problem. we introduce the SPIC problem formulation as the

following optimization problem that computes optimal and collision-free trajectories for all the CVAD in the intersection plaza:

$$\min_{\mathbf{u}_{i,\cdot|k},\forall i} \quad \sum_{i=1}^{N_\mathrm{v}} \sum_{t=k}^{k+N_\mathrm{h}} \mathcal{J}_{i,t}(\mathbf{x}_{i,t}, \mathbf{u}_{i,t}), \tag{66a}$$

$$\text{s.t.} \quad \mathbf{x}_{i,t+1|k} - \mathbf{f}(\mathbf{x}_{i,t|k}, \mathbf{u}_{i,t|k}) = \mathbf{0}, \tag{66b}$$

$$\hat{\theta}_{i,t+1|k} = \hat{\theta}_{i,t|k} + \Delta t\, \nu_{i,t|k}, \tag{66c}$$

$$\mathbf{x}_{i,0|k} - \mathbf{x}_{i,k} = \mathbf{0},\ \hat{\theta}_{i,0|k} - \hat{\theta}_{i,k} = 0, \tag{66d}$$

$$\mathbf{x}_{i,t|k} \in \mathcal{X},\ \mathbf{u}_{i,t|k} \in \mathcal{U},\ \nu_{i,t|k} \in [0, \nu_{\max}], \tag{66e}$$

$$\mathcal{P}(\mathbf{x}_{i,t|k}) \cap \mathcal{P}(\mathbf{x}_{j,t|k}) = \emptyset, \tag{66f}$$

$$\mathrm{dist}(\mathbf{p}_{i,t|k},\, \mathbf{p}^{\mathrm{ref}}) \leq \frac{w_\mathrm{r} - w_\mathrm{v}}{2}, \tag{66g}$$

$$i,j \in \{1, \ldots, N_\mathrm{v}\},\ i < j,$$

where superscript i refers to i-th CVAD and N_h is the prediction horizon. This problem is a multi-objective optimization problem, with (66)a representing the objective function. In (66)b, the function $\mathbf{f}(\cdot)$ represents the nonlinear model of each CVAD as defined in (43) and (46). (66)c represents the vehicle's progress in which $\hat{\theta}_i$ and ν_i are, respectively, an additional state variable and an additional control variable. The state and input feasible sets are respectively denoted by $\mathcal{X}$ and $\mathcal{U}$, which show the state and control input constraints as defined in (47). Additionally, constraints (66)f and (66)g represent the V2V collision avoidance constraints and intersection plaza boundary constraints, respectively.

In practical terms, we operate under the assumption that the SPIC problem is addressed through central coordination at the intersection plaza. In this process, every vehicle within the intersection transmits its data, encompassing vehicles' current states, to the coordinator. The optimization problem is then solved by the coordinator, and the resulting trajectories are subsequently communicated back to the respective vehicles.

5.2.2 Objective Function

In the SPIC problem, we would like to minimize the lateral and longitudinal errors and maximize the vehicles' progress along the reference paths while penalizing the control inputs for each CVAD in the intersection plaza. Given $\hat{\mathbf{E}}_k = [\hat{E}_k^\mathrm{c}, \hat{E}_k^\mathrm{l}]^\top$, the objective function is:

$$\mathcal{J} = \hat{\mathbf{E}}_k^\top \mathbf{Q} \hat{\mathbf{E}}_k + \nu_k^\top r_\nu \nu_k - q_\theta \hat{\theta}_k + \mathbf{u}_k^\top \mathbf{R} \mathbf{u}_k, \tag{67}$$

with weighting matrices Q and R as:

$$Q = \begin{bmatrix} q_\mathrm{c} & 0 \\ 0 & q_\mathrm{l} \end{bmatrix},\ R = \begin{bmatrix} r_\mathrm{v} & 0 \\ 0 & r_\delta \end{bmatrix}$$

in which $q_\mathrm{c} > 0$, $q_\mathrm{l} > 0$, $r_\nu > 0$, and $q_\theta > 0$ are lateral error, longitudinal error, virtual input, and progress cost weights, respectively, and $r_\mathrm{v} > 0$ and $r_\delta > 0$ are, respectively, the weights

on the acceleration and steering angle of each vehicle. The appropriate tuning of these weights allows for a trade-off between contouring accuracy and progress along the reference paths. In this application, it is desirable to sacrifice contouring accuracy to allow vehicles to traverse the path as fast as possible and make better use of the intersection space.

5.2.3 Extended Nonlinear Model Predictive Contouring Control (ENMPCC)

The ENMPCC method involves the inclusion of additional cost terms, such as fuel consumption (due to acceleration) or driving comfort (due to steering angle), and additional reference paths to cover all possible vehicle paths at the intersection plaza. Compared to the standard formulation, we incorporate V2V constraints to prevent potential vehicle collisions. Moreover, we impose intersection boundary constraints to restrict vehicle movements within intersection boundaries. Furthermore, the standard NMPCC assumes small lateral deviations of the controlled object from the reference path which is for the case in a high-precision application (LAM et al., 2013). However, the ENMPCC approach acknowledges the need for significant deviations from the reference path. These deviations are necessary to prevent potential collisions and/or to expedite the vehicles' traverse.

By employing the objective function (67) to the problem (66), the ENMPCC problem is defined as:

$$
\begin{aligned}
\min_{\substack{\mathbf{u}_{i,\cdot|k},\\ \nu_{i,\cdot|k},\forall i}} \quad & \sum_{i=1}^{N_\mathrm{v}} \left(\sum_{t=k}^{k+N_\mathrm{h}} \hat{\mathbf{E}}_{i,t}^\top \mathbf{Q} \hat{\mathbf{E}}_{i,t} - q_\theta \hat{\theta}_{i,t} + \sum_{t=k}^{k+N_\mathrm{h}-1} \mathbf{u}_{i,t}^\top \mathbf{R} \mathbf{u}_{i,t} + \nu_{i,t}^\top r_\nu \nu_{i,t} \right) \\
\text{s.t.} \quad & \mathbf{x}_{i,t+1|k} - \mathbf{f}(\mathbf{x}_{i,t|k}, \mathbf{u}_{i,t|k}) = \mathbf{0}, \\
& \hat{\theta}_{i,t+1|k} = \hat{\theta}_{i,t|k} + \Delta t \, \nu_{i,t|k}, \\
& \mathbf{x}_{i,0|k} - \mathbf{x}_{i,k} = \mathbf{0}, \ \hat{\theta}_{i,0|k} - \hat{\theta}_{i,k} = 0, \\
& \mathbf{x}_{i,t|k} \in \mathcal{X}, \ \mathbf{u}_{i,t|k} \in \mathcal{U}, \ \nu_{i,t|k} \in [0, \nu_{\max}], \\
& \sqrt{\left(x_{i,t|k} - x_{j,t|k}\right)^2 + \left(y_{i,t|k} - y_{j,t|k}\right)^2} \geq d_\mathrm{s} + l_\mathrm{v}, \\
& \sqrt{(x^{\mathrm{ref}}(\hat{\theta}_{i,t|k}) - x_{i,t|k})^2 + (y^{\mathrm{ref}}(\hat{\theta}_{i,t|k}) - y_{i,t|k})^2} \leq \frac{w_\mathrm{r} - w_\mathrm{v}}{2}.
\end{aligned}
\tag{68}
$$

5.2.4 Linear Time-Varying (LTV) Implementation

The objective of this section is to speed up the computation time of the proposed ENMPCC method for solving the SPIC problem. Problem (68) is an NMPCC problem that should be solved online at each time step. The computational cost associated with solving this problem can be high. The linear time-varying approximation of the nonlinear kinematic bicycle model as well as of the lateral and longitudinal errors is used to construct a tractable optimization problem (FALCONE et al., 2007). Each nonlinear function is linearized around the output of the last solution of the optimal control problem over a shifted horizon, except for the first and the last steps. The state of the first step is measured. The last step was not predicted in the previous

solution. Thus, the system is simulated for one time step assuming that the input is the same as in the second to last step.

5.2.4.1 LTV Approximation of the Vehicle Model

The kinematic bicycle model (46) is approximated by first-order Taylor series expansion around the predicted state's trajectory ($\bar{\mathbf{x}}_k = [\bar{x}_k, \bar{y}_k, \bar{\psi}_k, \bar{v}_k]^\top$) and control inputs ($\bar{\mathbf{u}}_k = [\bar{a}_k, \bar{\delta}_k]^\top$) computed at the previous time step as:

$$\bar{\mathbf{x}}_{k+1} = \mathbf{A_k}\bar{\mathbf{x}}_k + \mathbf{B_k}\bar{\mathbf{u}}_k + \mathbf{d_k}, \tag{69}$$

with

$$\begin{aligned}
\mathbf{A}_k &= \begin{bmatrix} 0 & 0 & -\bar{v}\sin(\bar{\psi}+\bar{\beta}) & \cos(\bar{\psi}+\bar{\beta}) \\ 0 & 0 & \bar{v}\cos(\bar{\psi}+\bar{\beta}) & \sin(\bar{\psi}+\bar{\beta}) \\ 0 & 0 & 0 & \frac{1}{l_\mathrm{r}}\sin(\bar{\beta}) \\ 0 & 0 & 0 & 0 \end{bmatrix}, \\
\mathbf{B}_k &= \begin{bmatrix} 0 & -\Gamma\bar{v}\sin(\bar{\psi}+\bar{\beta}) \\ 0 & \Gamma\bar{v}\cos(\bar{\psi}+\bar{\beta}) \\ 0 & \frac{\bar{v}}{l_\mathrm{r}}\Gamma\cos(\bar{\beta}) \\ 1 & 0 \end{bmatrix}, \\
\mathbf{d}_k &= \dot{\bar{\mathbf{x}}} - \mathbf{A}\bar{\mathbf{x}} - \mathbf{B}\bar{\mathbf{u}}, \\
\Gamma &= \frac{l_\mathrm{r}}{(l_\mathrm{f}+l_\mathrm{r})(\cos^2(\bar{\delta}) + l_\mathrm{r}^2\sin^2(\bar{\delta})/(l_\mathrm{f}+l_\mathrm{r})^2)}.
\end{aligned} \tag{70}$$

5.2.4.2 LTV Approximation of Lateral and Longitudinal Errors

The lateral and longitudinal errors (52) and (53) are approximated by linear functions over the prediction horizon using a Taylor series expansion around $\bar{\mathbf{x}}_k$ and $\bar{\hat{\theta}}_k$ as follows:

$$\hat{E}_k^{\mathrm{c,app}} = \hat{E}_k^{\mathrm{c}}(\bar{\mathbf{x}}_k, \bar{\hat{\theta}}_k) + (\nabla\hat{E}_k^{\mathrm{c}}(\bar{\mathbf{x}}_k, \bar{\hat{\theta}}_k))^\top \begin{bmatrix} \mathbf{x}_k - \bar{\mathbf{x}}_k \\ \hat{\theta}_k - \bar{\hat{\theta}}_k \end{bmatrix}, \tag{71}$$

$$\hat{E}_k^{\mathrm{l,app}} = \hat{E}_k^{\mathrm{l}}(\bar{\mathbf{x}}_k, \bar{\hat{\theta}}_k) + (\nabla\hat{E}_k^{\mathrm{l}}(\bar{\mathbf{x}}_k, \bar{\hat{\theta}}_k))^\top \begin{bmatrix} \mathbf{x}_k - \bar{\mathbf{x}}_k \\ \hat{\theta}_k - \bar{\hat{\theta}}_k \end{bmatrix}, \tag{72}$$

with

$$
\begin{aligned}
\nabla \hat{E}_k^{\mathrm{c}}(\bar{\mathbf{x}}_k, \bar{\hat{\theta}}_k) &= \begin{bmatrix} \sin(\phi(\bar{\hat{\theta}}_k)) & -\cos(\phi(\bar{\hat{\theta}}_k)) & 0 & 0 & d_{\mathrm{p,c}} \end{bmatrix}, \\
\nabla \hat{E}_k^{\mathrm{l}}(\bar{\mathbf{x}}_k, \bar{\hat{\theta}}_k) &= \begin{bmatrix} -\cos(\phi(\bar{\hat{\theta}}_k)) & -\sin(\phi(\bar{\hat{\theta}}_k)) & 0 & 0 & d_{\mathrm{p,l}} \end{bmatrix}, \\
d_{\mathrm{p,c}} &= -\gamma \hat{E}_k^{\mathrm{l}}(\bar{\mathbf{x}}_k, \bar{\hat{\theta}}_k) - \sin(\phi(\bar{\hat{\theta}}_k)) \nabla x^{\mathrm{ref}}(\bar{\hat{\theta}}_k) \cos(\phi(\bar{\hat{\theta}}_k)) \nabla y^{\mathrm{ref}}(\bar{\hat{\theta}}_k), \\
d_{\mathrm{p,l}} &= \gamma \hat{E}_k^{\mathrm{c}}(\bar{\mathbf{x}}_k, \bar{\hat{\theta}}_k) + \cos(\phi(\bar{\hat{\theta}}_k)) \nabla x^{\mathrm{ref}}(\bar{\hat{\theta}}_k) \sin(\phi(\bar{\hat{\theta}}_k)) \nabla y^{\mathrm{ref}}(\bar{\hat{\theta}}_k), \\
\gamma &= \frac{\eta}{(1+\alpha^2)(\nabla x^{\mathrm{ref}}(\bar{\hat{\theta}}_k))^2}, \\
\alpha &= \frac{\nabla y^{\mathrm{ref}}(\bar{\hat{\theta}}_k)}{\nabla x^{\mathrm{ref}}(\bar{\hat{\theta}}_k)}, \\
\eta &= \nabla^2 y^{\mathrm{ref}}(\bar{\hat{\theta}}_k) \nabla x^{\mathrm{ref}}(\bar{\hat{\theta}}_k) - \nabla y^{\mathrm{ref}}(\bar{\hat{\theta}}_k) \nabla^2 x^{\mathrm{ref}}(\bar{\hat{\theta}}_k).
\end{aligned}
$$

Equations (71) and (72) are replaced in the objective function of the ENMPCC (68).

5.2.5 LTV-ENMPCC Formulation

The LTV-ENMPCC formulation incorporates (63) to address V2V collision avoidance constraints, and it also integrates the LTV approximation method detailed in Section 5.2.4. This approximation is applied to the ENMPCC problem formulation introduced in Section 5.2.3, resulting in the following problem:

$$
\begin{aligned}
\min_{\substack{\mathbf{u}_{i,\cdot|k}, \nu_{i,\cdot|k}, \\ \boldsymbol{\lambda}_{\cdot|k}^{i,j}, \boldsymbol{\lambda}_{\cdot|k}^{j,1}, \mathbf{s}_{\cdot|k}^{i,j}}} \quad & \sum_{i=1}^{N_{\mathrm{v}}} \left(\sum_{t=k}^{k+N_{\mathrm{h}}} \hat{\mathbf{E}}_{i,t}^{\mathrm{app}\,T} \mathbf{Q} \hat{\mathbf{E}}_{i,t}^{\mathrm{app}} - q_\theta \hat{\theta}_{i,t} + \sum_{t=k}^{k+N_{\mathrm{h}}-1} \mathbf{u}_{i,t}^\top \mathbf{R} \mathbf{u}_{i,t} + \nu_{i,t}^\top r_\nu \nu_{i,t} \right) \\
\text{s.t.} \quad & \mathbf{x}_{i,t+1|k} = \mathbf{A}_{i,t} \mathbf{x}_{i,t|k} + \mathbf{B}_{i,t} \mathbf{u}_{i,t|k} + \mathbf{d}_{i,t|k} \\
& \hat{\theta}_{i,t+1|k} = \hat{\theta}_{i,t|k} + \Delta t \, \nu_{i,t|k}, \\
& \mathbf{x}_{i,0|k} - \mathbf{x}_{i,k} = \mathbf{0}, \ \hat{\theta}_{i,0|k} - \hat{\theta}_{i,k} = 0, \\
& \mathbf{x}_{i,t|k} \in \mathcal{X}, \ \mathbf{u}_{i,t|k} \in \mathcal{U}, \ \nu_{i,t|k} \in [0, \nu_{\max}], \\
& \left(-\mathbf{b}^i (\mathbf{x}_{t|k}^i)^\top \boldsymbol{\lambda}_{t|k}^{i,j} - \mathbf{b}^j (\mathbf{x}_{t|k}^j)^\top \boldsymbol{\lambda}_{t|k}^{j,i} \right) \geq d_{\mathrm{s}}, \\
& \mathbf{A}^i (\mathbf{x}_{t|k}^i)^\top \boldsymbol{\lambda}_{t|k}^{i,j} + \mathbf{s}_{t|k}^{i,j} = \mathbf{0}, \\
& \mathbf{A}^j (\mathbf{x}_{t|k}^j)^\top \boldsymbol{\lambda}_{t|k}^{j,i} - \mathbf{s}_{t|k}^{i,j} = \mathbf{0}, \\
& -\boldsymbol{\lambda}_{t|k}^{i,j} \leq \mathbf{0}, \ -\boldsymbol{\lambda}_{t|k}^{j,i} \leq \mathbf{0}, \ \|\mathbf{s}_{t|k}^{i,j}\|_2 \leq 1, \\
& \sqrt{(x^{\mathrm{ref}}(\hat{\theta}_{i,t|k}) - x_{i,t|k})^2 + (y^{\mathrm{ref}}(\hat{\theta}_{i,t|k}) - y_{i,t|k})^2} \leq \frac{w_{\mathrm{r}} - w_{\mathrm{v}}}{2}.
\end{aligned} \tag{73}
$$

The complexity associated with problem (73) is reduced through the use of the LTV approximation of the nonlinear terms present in the bicycle model (46) and the objective function (67). Although LTV approximation of the V2V constraints can be a potential approach, for safety-critical constraints such as (66)f, it is crucial to employ advanced methods that better

capture the V2V constraints (LI, S. E. et al., 2020). Accordingly, we mitigated the complexity of the V2V constraints by leveraging the duality theory as discussed in Section 5.1.4.2. By this technique the non-differentiable collision avoidance constraints (66)f are reformulated into smooth and differentiable algebraic inequalities (BOYD, S. P.; VANDENBERGHE, 2004).

5.2.6 SPIC: Algorithm Pseudo Code

In this section, we present the SPIC algorithm to solve problem (73) and to simulate an intersection plaza in which CVAD continuously approach the intersection. The proposed algorithm leverages the ENMPCC method to efficiently compute optimal trajectories for each CVAD while considering various constraints. The SPIC algorithm is designed to handle dynamic scenarios in a random order, making it an ideal candidate for coordinating CVAD at the intersections where the vehicles are expected to interact with each other and with the intersection infrastructure. The main steps of the algorithm are summarized in Algorithm 1 while its flowchart is illustrated in Figure 21.

The algorithm starts assuming that no CVAD is in the Plaza. The algorithm then enters a loop and waits for the arrival of a CVAD at the Plaza. As soon as a new CVAD arrives, its optimal inputs and state predictions are estimated according to its initial conditions, and then the number of CVAD, N_{v}, is incremented. These inputs and state predictions are used to compute the LTV approximation problem (73). For each CVAD, the algorithm linearizes the lateral and longitudinal errors and the nonlinear dynamics around the estimated optimal inputs and state predictions. Then, the CVAD is included in the instantiated NLP problem (73). The problem is solved and the optimal inputs for each CVAD are obtained. The algorithm then updates the state and progress of each CVAD using the computed optimal inputs and shifts the optimal inputs and states to estimate the next optimal inputs and state predictions. Finally, if a CVAD leaves the intersection, it is removed from the NLP problem in the next instance, and the number of CVAD in the system is decremented. The algorithm repeats this process for the remaining CVAD until the desired stopping criterion is reached.

5.3 THE SPIC SIMULATOR

This section provides a high-level overview of the SPIC simulator (SPIC-Sim) architecture. The SPIC simulator is a simulation modeling tool that allows for the modeling of flexible utilization of urban intersections, enabling the investigation of both conventional (path-based or lane-based) and unconventional (path-free or lane-free) traffic control scenarios. Figure 22 provides a high-level overview of the SPIC-Sim architecture, highlighting its key layers and components. As observed in the figure, the primary layers of the SPIC-Sim are the user interface (UI), the simulator, the controller, and the optimizer, each of which consists of different tools and components. The layered architecture of the SPIC-Sim leads to a modular system that can be appropriately extended by new features and algorithms. In the following, we describe each

Algorithm 1 SPIC Algorithm

Input: $N_{\mathrm{v}} = 0$, N_{h}, $\mathbf{x}_i(0)\ \forall i = 1,\ldots,N_{\mathrm{v}}$, $\mathbf{Q}$, $\mathbf{R}$

for $k = 0,1,\ldots$ **do**
  if i-th CVAD arrived at the Plaza **then**
    $N_{\mathrm{v}} \leftarrow N_{\mathrm{v}} + 1$
    Compute $\bar{\mathbf{u}}_{i,k}$ and $\bar{\nu}_{i,k}$ by using Procedure 1 introduced by Lam, Manzie, and Good (2010).
    Compute $\bar{\mathbf{x}}_{i,k}$ and $\bar{\hat{\theta}}_{i,k}$ by applying $\bar{\mathbf{u}}_{i,k}$ and $\bar{\nu}_{i,k}$ to (46) and (51), respectively.
  end if
  for all $i = 1,\ldots,N_{\mathrm{v}}$ **do**
    Linearize $\hat{E}_k^{\mathrm{c}}$ and $\hat{E}_k^{\mathrm{l}}$ around $\bar{\mathbf{x}}_{i,k}$ and $\bar{\hat{\theta}}_{i,k}$, using (71) and (72).
    Linearize the nonlinear dynamics of the i-th CVAD (46), around $\bar{\mathbf{u}}_{i,k}$ and $\bar{\mathbf{x}}_{i,k}$ by using (69).
    Include the i-th CVAD in the new instance of the optimization problem (73).
  end for
  Solve the NLP optimization problem (73) and obtain $\mathbf{u}^*_{i,k}$, $\nu^*_{i,k}$.
  for all $i = 1,\ldots,N_{\mathrm{v}}$ **do**
    Update i-th CVAD states by applying the first element of $\mathbf{u}^*_{i,k}$ to the nonlinear dynamics (46).
    Update i-th CVAD progress by applying the first element of $\nu^*_{i,k}$ to (51).
    Compute $\bar{\mathbf{u}}_{i,k}$ and $\bar{\nu}_{i,k}$ by shifting optimal states and inputs predictions $\mathbf{u}^*_{i,k}$ and $\nu^*_{i,k}$.
    Compute $\bar{\mathbf{x}}_{i,k}$ and $\bar{\hat{\theta}}_{i,k}$ by applying $\bar{\mathbf{u}}_{i,k}$ and $\bar{\nu}_{i,k}$ to (46) and (51), respectively.
  end for
  if a CVAD left the Plaza **then**
    Delete the CVAD from the NLP optimization problem (73).
    $N_{\mathrm{v}} \leftarrow N_{\mathrm{v}} - 1$
  end if
end for

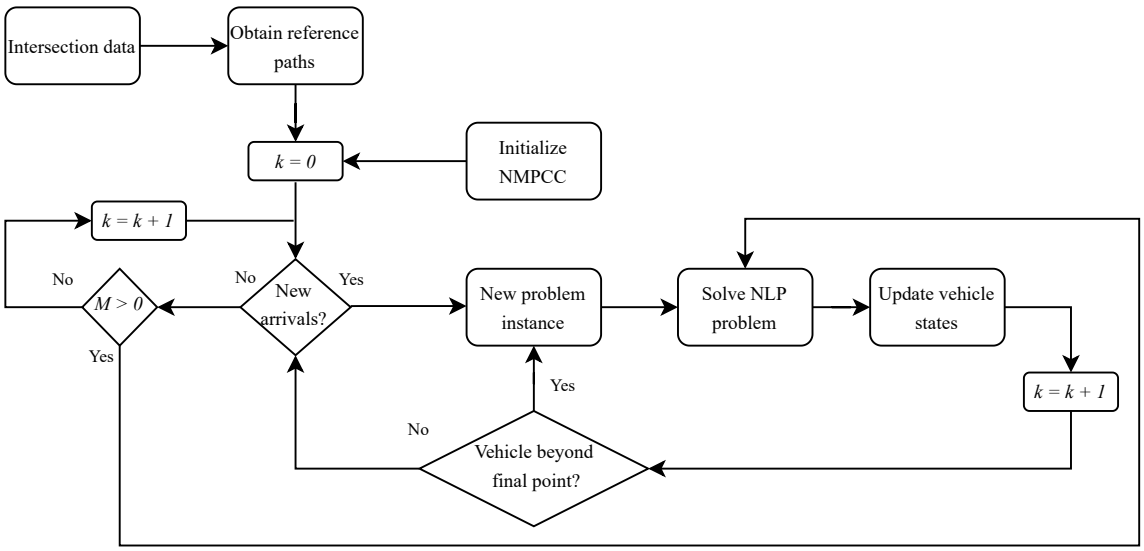


Figure 21 – A flowchart of the proposed SPIC algorithm.

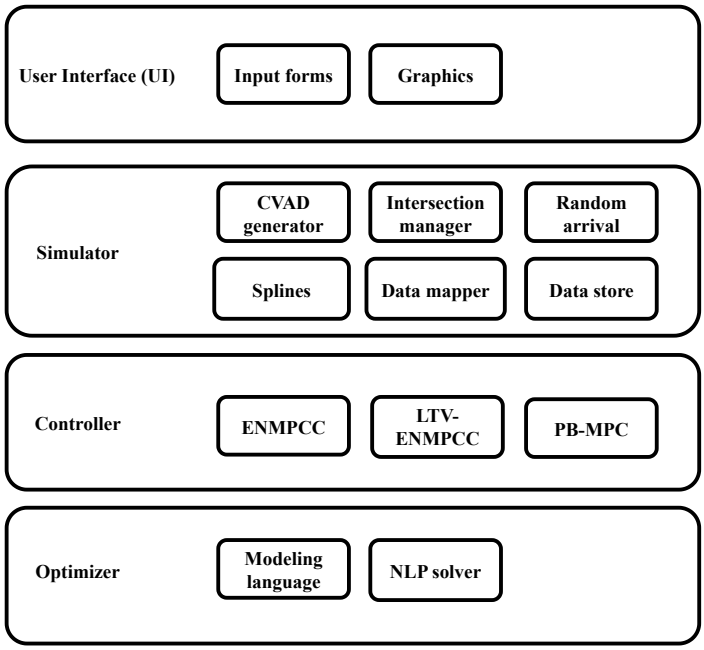


Figure 22 – The SPIC simulator architecture, main layers, and components.

layer of the SPIC-Sim and its main components.

5.3.1 User Interface

The UI component is the primary interface through which end-users interact with and configure the SPIC problem and algorithms. The UI layer encompasses key elements, including *Input forms*, and *Graphics*. The Input forms component consists of various forms and input handlers to configure the problem and algorithms via a graphical interface. Parameters such as vehicle height and width, prediction horizon, simulation time, etc can be defined by the input forms component. On the other hand, the graphics component is employed to create various visualizations and graphical elements like charts and animations.

5.3.2 Simulator

The simulator layer of the SPIC-Sim consists of multiple components such as the *CVAD generator*, *Intersection manager*, *Random arrival*, *Splines*, *Data mapper*, and *Data store*. The role of the CVAD generator component is to generate a new vehicle instance and introduce it to the intersection manager component when a vehicle enters the intersection. The intersection manager stands as a main component with the task of overseeing CVAD operations at an intersection. The specific time when a vehicle enters the intersection is determined randomly by the random arrival component. Upon the arrival of a new vehicle at the intersection, the intersection manager gathers and classifies the vehicle instance according to its approach direction. Subsequently, the vehicles are assigned suitable reference paths based on their origin and intended destination. These reference paths are created through the utilization of the spline component and are integral to the proposed methods. Additionally, the data mapper serves as entities responsible for transforming

vehicle objects into a compatible format to be used by the controller layer. Lastly, the data store component holds the simulator data intended for transmission to the UI layer, where it can be utilized for visual representations.

5.3.3 Controller

The controller layer is the main dependency of the simulator layer and interfaces the main control strategies implemented in this thesis, including *ENMPCC*, *LTV-ENMPCC*, and *PB-MPC*. The ENMPCC component applies the ENMPCC algorithm without using LTV approximation. On the other hand, the LTV approximation is employed within the LTV-ENMPCC component, where linear approximations of the objective function and system dynamics are used. Meanwhile, the PB-MPC component's responsibility is implementing a path-based MPC approach, utilizing a standard MPC method to address the conventional path-based (lane-based) driving approaches. Hence, the controller layer offers a variety of control strategies for the simulator layer to choose from, depending on user inputs. Once selected, the chosen control strategy remains in use for the entire duration of the simulation.

5.3.4 Optimizer

The central component of the SPIC-Sim is the optimizer layer, responsible for building and solving the optimization problem specified by the controller layer. The key elements within this layer include the *Modeling language* for constructing optimization models and the *NLP solver* for solving the associated optimization problems. In the present SPIC-Sim implementation, the Casadi library is utilized for building optimization problems, and IPOPT is employed to solve these optimization problems.

5.4 PRESENTATION AND DISCUSSION OF RESULTS

In this section, we assess the effectiveness of the proposed ENMPCC and LTV-ENMPCC methods in addressing the SPIC problem through various scenarios. We consider different vehicle models and incorporate different V2V collision avoidance modeling techniques to comprehensively evaluate the proposed method. Furthermore, results from a comparison with automated vehicles based on conventional path-based (lane-based) driving are shown. The comparison aims to demonstrate the feasibility and benefits of employing the SPIC algorithm for optimal and collision-free coordination of the CVAD at the intersection plaza. The simulations are implemented in the Python programming language (PYTHON SOFTWARE FOUNDATION, 2022) with CasADi, an open-source software package for nonlinear optimization and algorithmic differentiation (ANDERSSON et al., 2019). To solve the corresponding NLP problem (73), we used IPOPT, an open-source solver for large-scale nonlinear optimization (WÄCHTER; BIEGLER, 2006). The splines for obtaining the reference paths were computed using the Python *spline* command.

Table 5 – Settings for vehicles, intersection, and optimization problem

Parameter	Description	Value	Unit
Δt	Sampling time	0.2	(s)
$v_{\max}$	Maximum speed	10 or 20	(m/s)
$a_{\max}$	Maximum acceleration	2 or 5	(m/s^2)
$\delta_{\max}$	Maximum steering angle	0.52	(rad)
$\dot{\delta}_{\max}$	Maximum steering angle rate	2.09	(rad/s)
f_{v}	Vehicle flow	1200	(veh/h)
g	Gravitational acceleration	9.8	(m/s^2)
d_{s}	Safe distance	1	(m)
L	Vehicle's wheelbase distance	2.52	(m)
w_{v}	Vehicle width	1.40	(m)
l_{v}	Vehicle length	2.60	(m)
w_{r}	Road width	10	(m)
l_{f}	Distance from front axle to the vehicle CG	1.03	(m)
l_{r}	Distance from rear axle to the vehicle CG	1.49	(m)
N_{h}	Prediction horizon	30	(steps)
$\mathbf{Q}$	ENMPCC cost weight, $\mathrm{diag}(q_{\mathrm{c}}, q_{\mathrm{l}})$	$\mathrm{diag}(0.005, 5)$	-
$\mathbf{R}$	Control cost weight, $\mathrm{diag}(r_{\mathrm{v}}, r_{\delta})$	$\mathrm{diag}(5, 0.5)$	-
q_θ	Progress cost weight	1	-
r_ν	Virtual input weight	0.02	-
$q_{\mathrm{x}}, q_{\mathrm{y}}$	PB-MPC cost weights	10	-
N_{v}	Number of CVAD	28	-
N_{l}	Number of lanes	2	-
h	Number of intersection plaza boundaries	4	-
μ	Friction coefficient	1	-

5.4.1 Scenario Setup

We consider a challenging traffic scenario with the coordination of multiple vehicles entering the intersection plaza continuously. The simulations were conducted for a four-leg intersection illustrated in Figure 15. This simple intersection geometry enables a clear presentation of the method, including straight and turning movements, and two-way driving. Three simulation scenarios are considered.

In the first Scenario (SC1), we consider a simple case study in which the intersection plaza consists of two approaches; one from the south and the other from the west. In the first approach, vehicles enter the plaza from the south and have the option to either turn right or continue straight. On the other hand, in the second approach, vehicles enter from the west and can choose to either turn left or go straight. The four reference paths that can be followed by each CVAD based on their entry and exit directions are indicated by the light-blue, gray, purple, and green dashed lines in Figure 15. Scenario 1a (SC1a) and Scenario 1b (SC1b) evaluate the proposed ENMPCC method in (68) by considering two different kinematic bicycle models (43) and (46), respectively.

In the second Scenario (SC2), we consider a more complex case study in which the CVAD can approach the intersection plaza from four possible origins, namely south, north, west, or east, and can take one of three actions, turn left or right or go straight, i.e., toward any of the three remaining destinations. The six possible reference paths that each CVAD can follow

based on their entry and exit directions are displayed as colored dashed lines in Figure 15. SC2 implements the SPIC algorithm based on the LTV-ENMPCC method in (73) as described in Section 5.2.6.

Finally, in Scenario 3 (SC3) a conventional path-based driving of the CVAD is applied in the same conditions as Scenario 2. In this scenario, we use the same SPIC problem formulation however, we use a standard tracking MPC objective function to provide a conventional path-based trajectory tracking problem (PB-MPC) as:

$$\mathcal{J} = \sum_{i=1}^{N_{\mathrm{v}}} \left(\sum_{t=k}^{k+N_{\mathrm{h}}} q_{\mathrm{x}}(x_{i,t} - x^{\mathrm{ref}})^2 + q_{\mathrm{y}}(y_{i,t} - y^{\mathrm{ref}})^2 + \sum_{t=k}^{k+N_{\mathrm{h}}-1} \mathbf{u}_{i,t}^{\top} \mathbf{R} \mathbf{u}_{i,t} \right) \tag{74}$$

where q_{x} and q_{y} are suitable cost weights of the PB-MPC. We note that the reference paths are appropriately shifted to make a path-based scenario and no deviation from the paths is allowed. In this scenario, the analysis is carried out on a two-lane intersection, where each vehicle is initially located in the center of each lane.

For each scenario, simulations of 900 time steps (180 s) with flows from 300 to 2700 veh/h/approach in steps of 600 veh/h/approach were tested. For each of these fifteen cases, seven replications with different seeds were run to obtain mean estimates of the performance metrics with 95% confidence of an error below 10% (LAW, 2013).

The arrival intervals between vehicles are generated randomly from a shifted negative exponential distribution (FHWA, 2022). The arrival position along the road width at the entry point and the arrival speed around 4 m/s for each CVAD are randomly selected by sampling a uniform distribution. When exiting the plaza, the CVAD leave at either $x = 60$ m or $y = 60$ m, depending on the road used. We select $N_{\mathrm{h}} = 30$ to achieve a reasonable trade-off between computational complexity and V2V collision avoidance constraints accuracy for future problem instances. Table 5 lists the additional essential parameters pertaining to vehicles, intersection geometry, and the optimization problem.

5.4.2 Scenario 1a (SC1a): The ENMPCC method with kinematic bicycle model (43)

The aim of this scenario is to assess the effectiveness of the proposed ENMPCC method for solving the SPIC problem (66). The controller utilizes the kinematic bicycle model (43) to represent the vehicles. To ensure safety, a circular representation is used to model the road region occupied by each vehicle, and constraint (54) is implemented to prevent V2V collisions. Moreover, constraint (64) is applied to keep the vehicles within the boundaries of the plaza. To incorporate these constraints into the SPIC problem formulation, the respective equations, namely (43), (44)–(45), (54), and (64), are substituted into the corresponding constraints in problem (68).

The six snapshots of the optimal and collision-free trajectories of CVAD, obtained through the solution of ENMPCC in (68), are illustrated in Figure 23 at time steps 125, 155, 235, 275, 365, and 435. In the figures, the two gray dashed lines represent the reference paths

for vehicles moving straight through the intersection plaza, while the two gray dot-dash lines show the reference paths for vehicles making left or right turns. We can also see the trajectory of

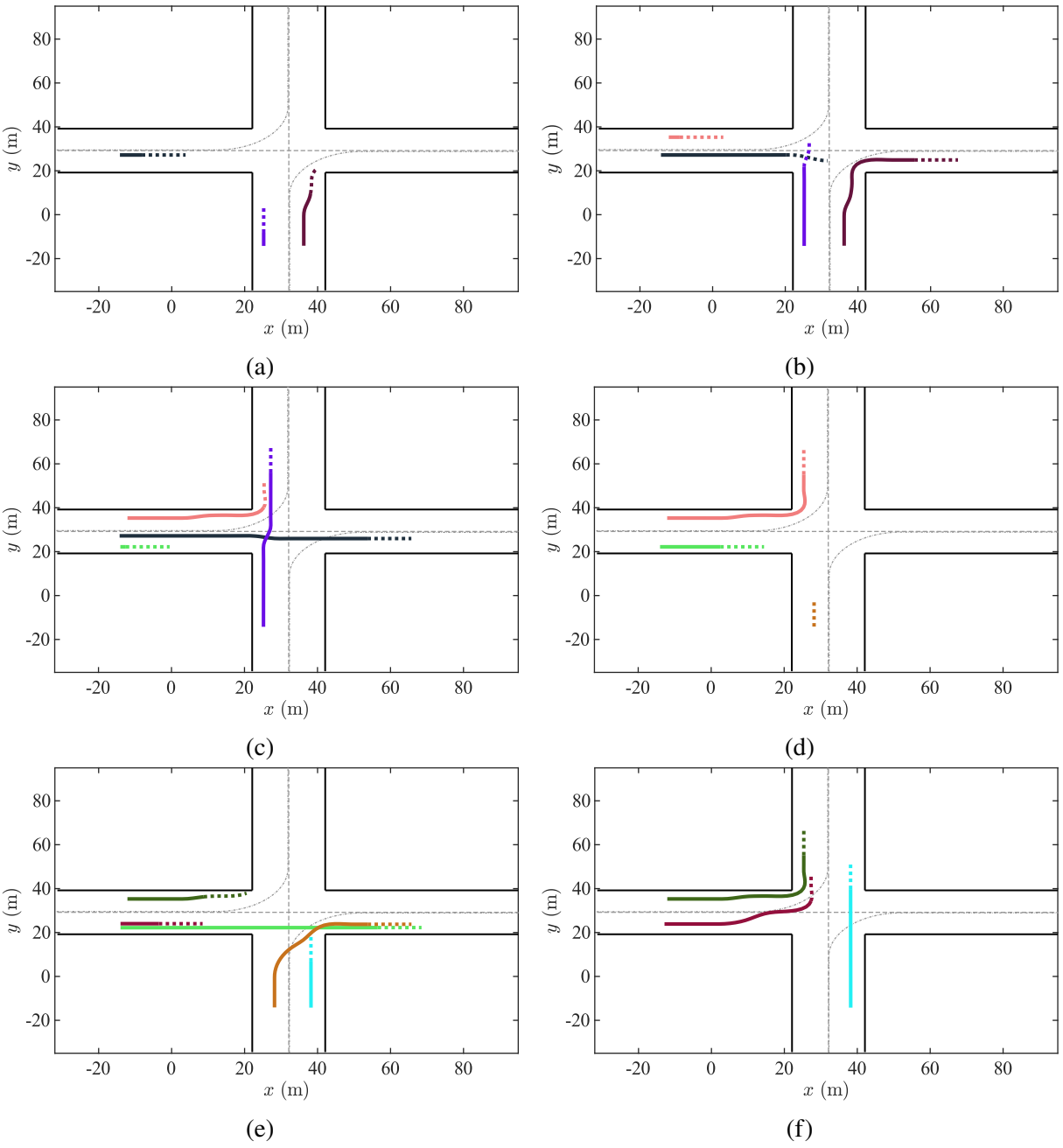


Figure 23 – SC1a: snapshots of the trajectories of the vehicles at time steps 125, 155, 235, 275, 365, and 435 with a flow of 1200 veh/h/approach.

the vehicle until the snapshot time instant by the colored solid line and its prediction for the next steps by the dotted lines of the same color. New arrivals are included in the problem, for example, the red vehicle from Figure 23(a) to Figure 23(b). Once the vehicle is beyond the final point, it is removed from the figures, for example, the brown vehicle from Figure 23(b) to Figure 23(c).

As can be seen in the figure, the vehicles are able to drive without being bound to the usual path-related road lanes and horizontal markings. Thus, the trajectories may deviate from what would be expected in a path-based method. For example, the vehicle with the light brown

trajectory in Figure 23(e) and the vehicle with the garnet trajectory in Figure 23(g) are suiting examples. The effectiveness of the collision avoidance constraint and the role of the prediction horizon can be seen in Figure 23(b). The black and purple vehicles were still in straight paths when a potential collision was predicted, as shown by the curved dotted lines in each predicted trajectory. The deviation is confirmed by the executed path seen in 23(c).

The kinematic bicycle model (43) with rear-wheel driving may cause potential challenges related to the collision between every two vehicles as illustrated in Figure 19(a). This arises due to the model's use of the rear wheel as the reference point for vehicle coordinates, which necessitates a careful selection of the parameter d_{s}. To address this concern, in the next scenario, we assess the ENMPCC method while employing the kinematic bicycle model (46) with a center of gravity driving.

5.4.3 Scenario 1b (SC1b): The ENMPCC method with kinematic bicycle model (46)

In this scenario, we consider the same conditions as in Scenario 1a, but with a difference in the employed kinematic bicycle model. Specifically, we adopt the kinematic bicycle model (46) as the vehicle model for this scenario. Accordingly, we incorporate constraints (46), (47), (54), and (64) into the corresponding constraints of the SPIC problem (68).

Figure 24 displays a sequence of six snapshots of the optimal trajectories of CVAD entering the plaza. These snapshots correspond to the same time steps as SC1a. These trajectories are obtained by solving optimization problem (68), utilizing the kinematic bicycle model illustrated in Figure 17. The trajectories of the red, green, and purple vehicles in Figure 24(c), along with the vehicle exhibiting the light blue trajectory in Figure 24(e), serve as fitting examples of path-free behavior of CVAD. New vehicles arriving are treated as new instances of the problem. For example, vehicle 4 with red trajectory from Figure 24(a) to Figure 24(b). Moreover, once the vehicle is beyond the final point, it is removed in the next instance of the problem, for example, vehicle 1 with the gray trajectory is removed from Figure 24(b) to Figure 24(c). As in SC1a, in Figure 24(b), one can seen the influence of the collision avoidance constraints. Notably, the green and purple vehicles initially followed straight paths but deviated due to predicted collisions, shown by curved lines in their trajectories.

To better assess the proposed method, Figures 25(a) to 25(e) show the speed, acceleration, orientation, steering angle, and progress profiles for six vehicles of interest (vehicles 2, 3, 5, 6, 8, and 9). To plot the figures, we assumed that vehicles have constant profiles, and thus, zero acceleration profiles before entering and after leaving the intersection plaza. Figure 25(f) shows the distances between six pairs of the vehicles of interest (vehicles 2-3, 2-4, 5-6, 7-8, 8-9, and 9-10). To plot the figure, we considered the time steps that every two vehicles are inside the intersection plaza.

To interpret the figures, we considered vehicles 2 and 3 as an example. Vehicle 3 is seen to reduce its speed and acceleration (Figures 25(a)-(b)). On the other hand, vehicle 2 is seen to increase its speed and acceleration (Figures 25(a)-(b)). Meanwhile, both vehicles are seen to

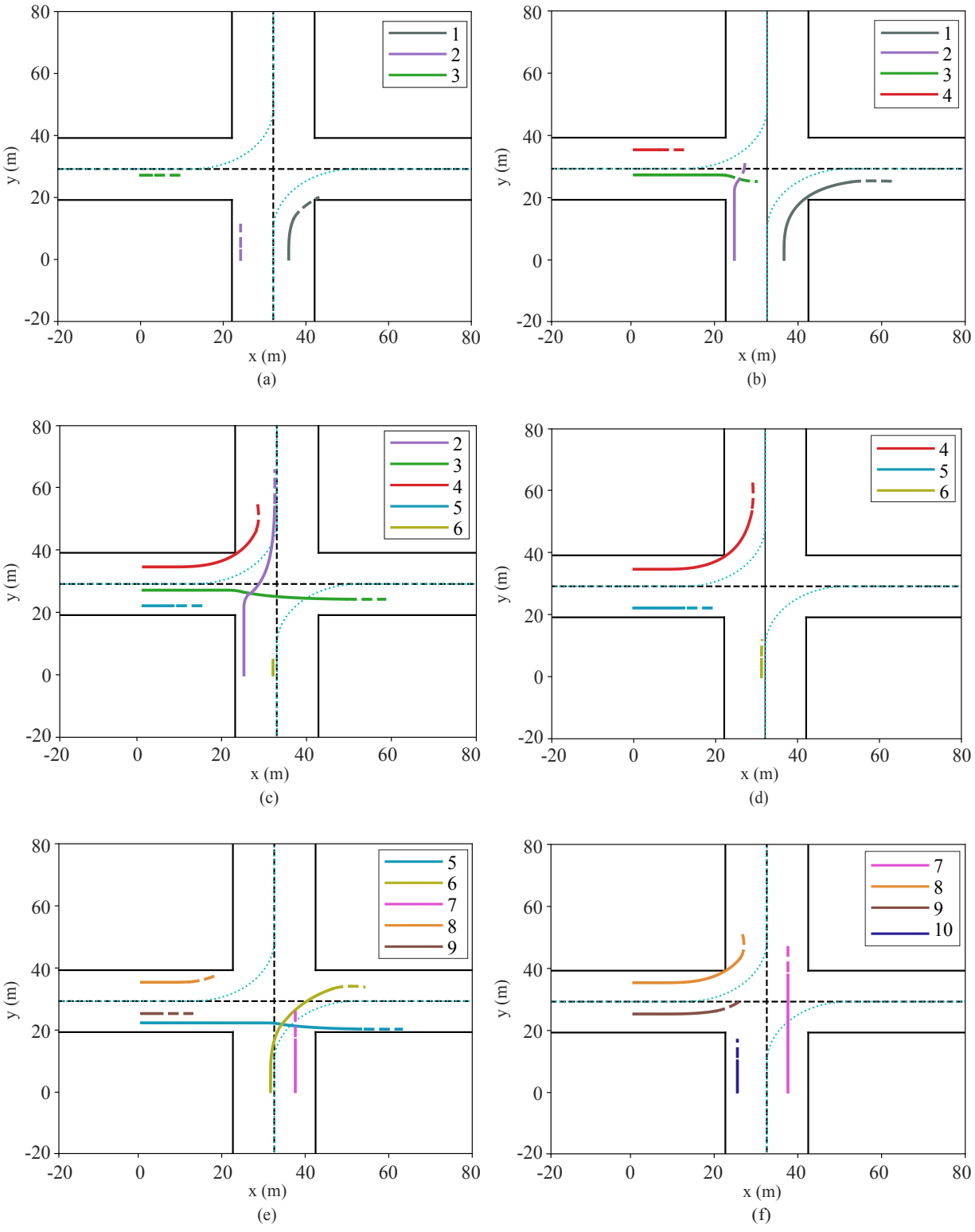


Figure 24 – SC1b: snapshots of the trajectories of the vehicles at time steps 125, 155, 235, 275, 365, and 435 with a flow of 1200 veh/h/approach.

steer to the right and then to the left (Figure 25(c)) with a corresponding temporary change in orientation (Figure 25(d)). Indeed, in Figure 24(c) one can see that vehicles 2 and 3 go further to the right relative to their previous position so that, along with the changes in speed (Figure 25(a)),

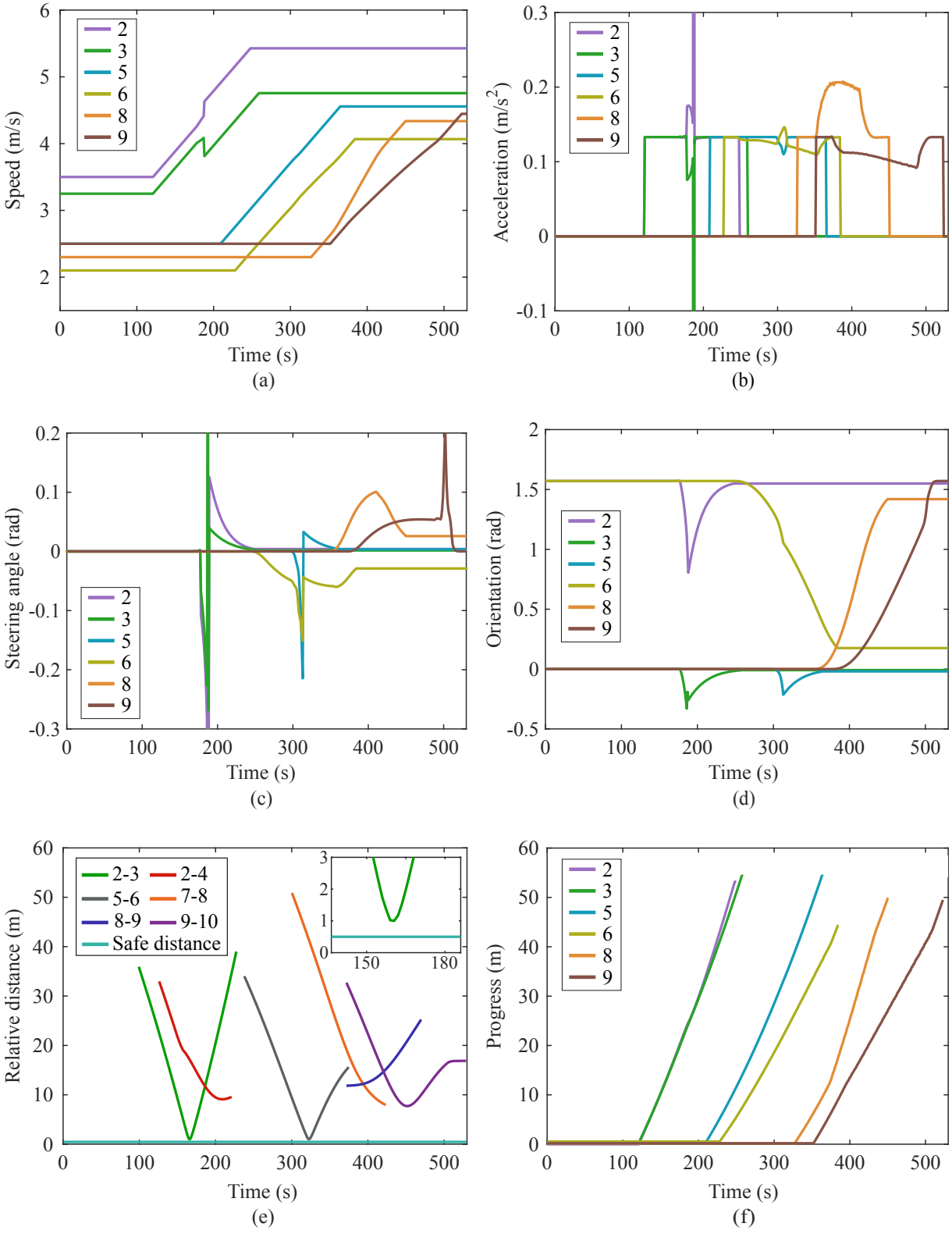


Figure 25 – SC1b: the state and control variables for six vehicles of interest (vehicles 2, 3, 5, 6, 8, and 9): (a) Speed; (b) acceleration; (c) steering angle; (d) orientation; (e) distance between every two selected vehicles; and (f) progress.

a collision is avoided. It can be seen from Figure 25(e) that the distance between vehicles 2 and 3 (the green line) decreases and then increases staying above the minimum safe distance (the

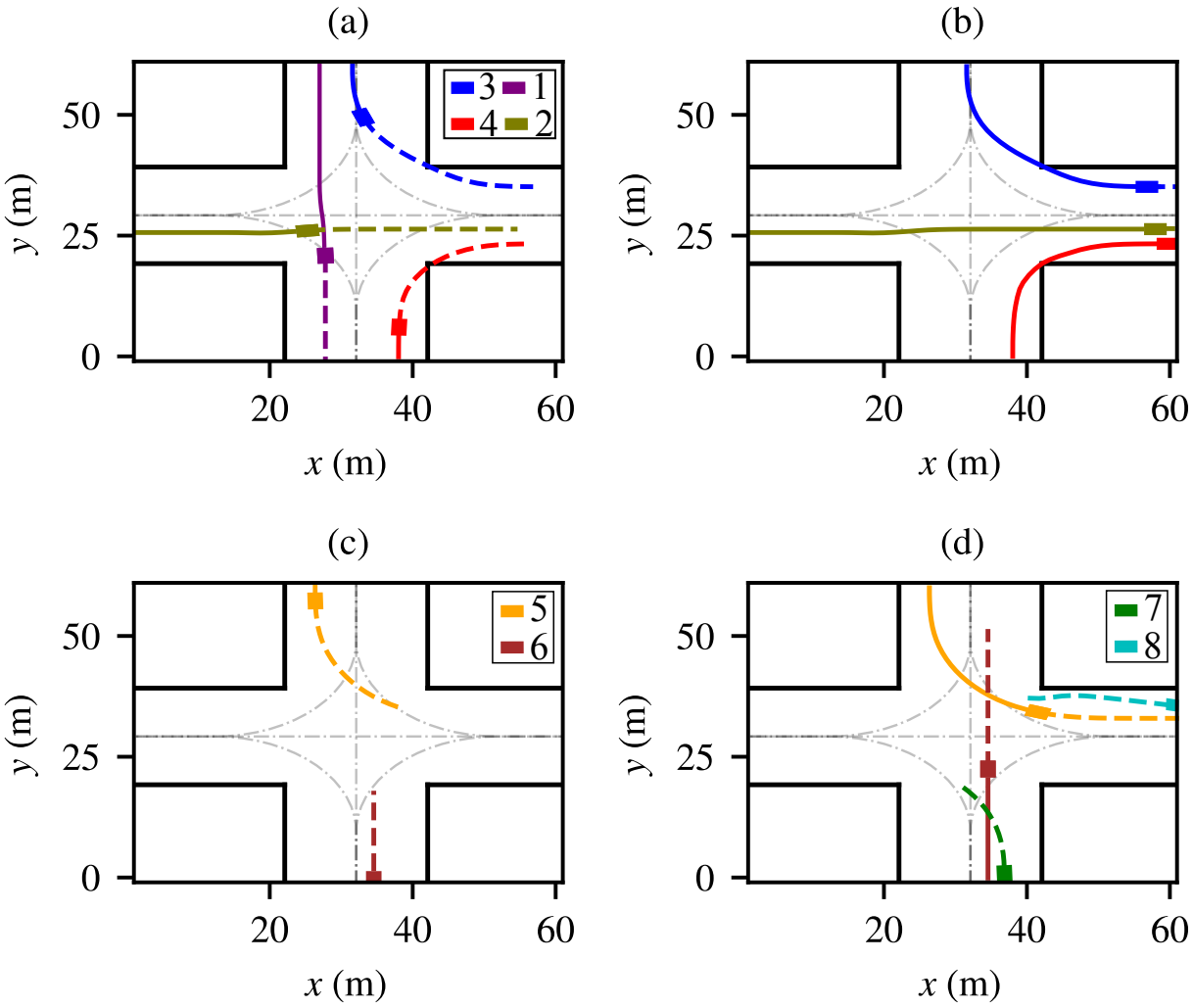


Figure 26 – SC2: snapshots of intersection plaza in time steps (a) 65, (b) 88, (c) 109, and (d) 130 operating under the LTV-ENMPCC algorithm with a flow of 1200 veh/h/approach.

light blue line). Finally, in Figure 25(f), with the proposed method, all vehicles have increasing progress profiles, which is one of this paper's goals.

5.4.4 Scenario 2 (SC2): The LTV-ENMPCC method

Figure 26 shows a sequence of four snapshots of the application of the proposed LTV-ENMPCC algorithm with time steps 65, 88, 109, and 130. The flow is 1200 veh/h/approach. In the snapshots, the six gray dash-dotted lines show the reference paths for through, left, and right turning movements. The colored solid curves show the trajectories of the CVAD until the snapshot time step. The predicted trajectories for the next steps are shown by the dashed lines of the same color. New arrivals are included in a new instance of the problem. For example, vehicles 5 and 6 from Figure 26(b) to Figure 26(d). Vehicles that traveled beyond the exit point are removed in the next instance of the problem, for example, vehicle 1 from Figure 26(a) to Figure 26(b) or vehicles 2, 3, and 4 from Figure 26(b) to Figure 26(c).

As depicted in Figure 26, the vehicles have the ability to drive without being restricted to

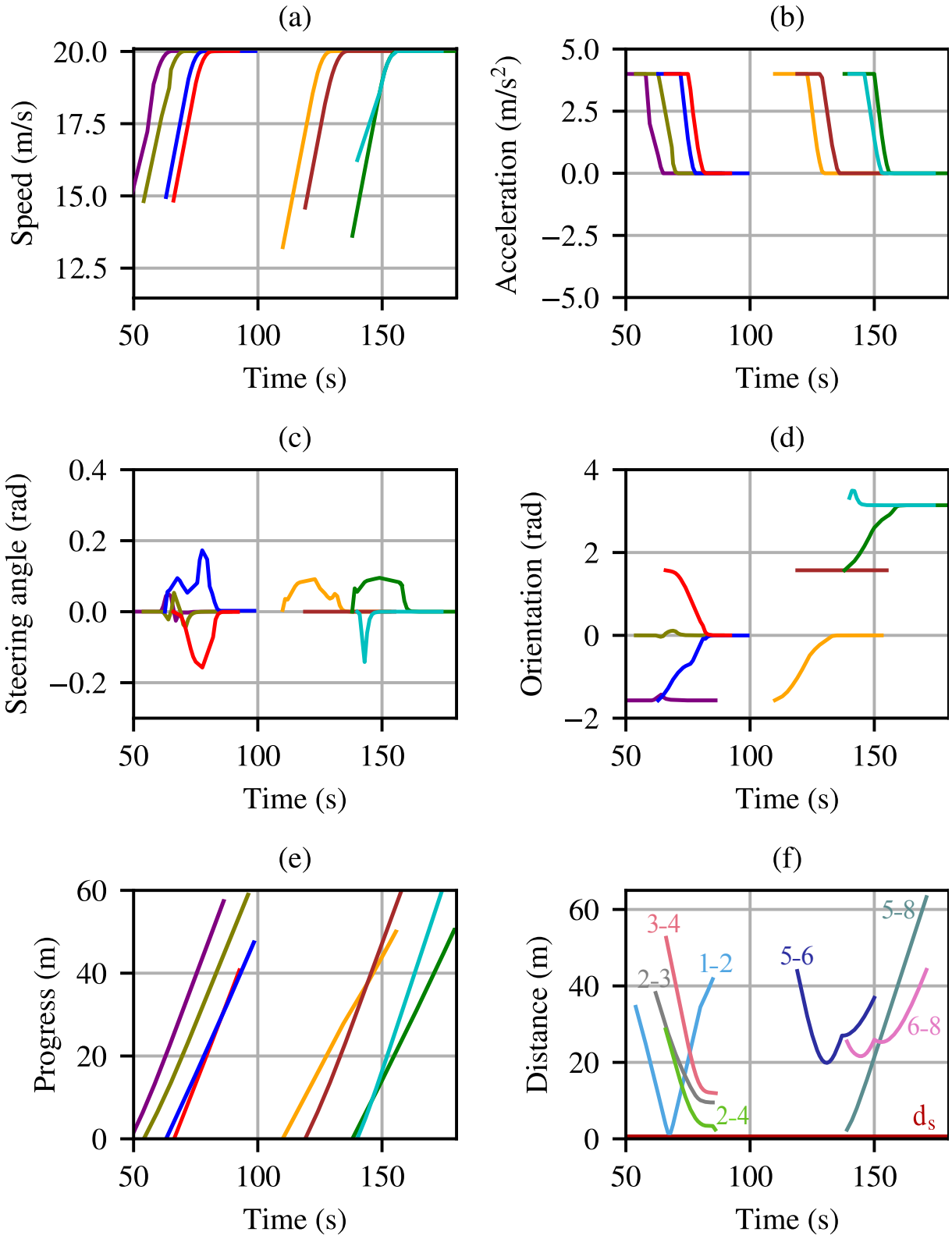


Figure 27 – SC2: states and inputs of the first eight vehicles in the intersection plaza operating under the LTV-ENMPCC algorithm with a flow of 1200 veh/h/approach.

conventional path-related road lanes and horizontal markings. As a result, their trajectories may deviate from what would typically be expected using a path-based approach, to either maximize their progress or avoid potential collisions. Notably, the trajectories of vehicles 1 and 2 shown in

Figure 26(a), the trajectories of vehicles 3 and 4 depicted in Figure 26(b), and the trajectories of vehicles 5 and 8 depicted in Figure 26(d) serve as compelling examples of such deviations. Figure 26(b) illustrates the progress maximization achieved through the utilization of path deviation, leading to more effective utilization of the intersection space.

Furthermore, Figure 26(a) provides a clear illustration of the collision avoidance constraint's effectiveness and the critical role played by the prediction horizon. Specifically, vehicles 1 and 2, traveling in straight paths, encountered a potential collision that was detected and addressed with a deviation from their original trajectories, as evidenced by the curved dashed lines in each predicted path. The executed paths, as shown in Figure 26(b), confirm the avoidance maneuver taken by these vehicles. In addition, Figure 26(d) provides evidence of the collision avoidance behavior of vehicle 8 upon entering the plaza, as it slightly deviates to the right of the road to prevent a collision with vehicle 5, which is exiting the plaza as evidenced by the curved dashed lines in the predicted path of vehicle 8.

To gain a better understanding of the snapshots, we can refer to Figure 27 which displays the speed, acceleration, steering angle, orientation, and progress profiles of the vehicles depicted in Figure 26 (vehicles 1–8) and the distances between selected vehicles. The vehicles accelerate to reach the maximum allowed speed and maintain that speed until they exit the intersection. To interpret the figures, we considered vehicle 8 as an example. Vehicle 8 is seen to exhibit a slower rate of speed increase for a short period (Figure 27(a)) and a reduction in its acceleration (Figure 27(b)). Meanwhile, the vehicle is seen to steer to the right and then to the left (Figure 27(c)) with corresponding temporary changes in their orientation (Figure 27(d)). In addition, in Figure 27(e) all vehicles have increasing progress profiles, which is one of this work's goals. Finally, Figure 27(f) illustrates the distances between seven pairs of vehicles. As an example, it can be seen that the distance between vehicles 1 and 2 (the light-blue line) decreases and then increases staying above the minimum safe distance (the red line) to avoid a potential collision as shown in Figures 26(a)-(b).

The obtained results demonstrate that the proposed method is able to produce collision-free multi-vehicle interaction and has the reasonable capability to resolve vehicle conflicts in challenging intersection scenarios.

5.4.5 Scenario 3 (SC3): Conventional Path-based Approach

In this scenario, we conduct an analysis on an intersection with a two-lane road, where each vehicle is initially positioned in the center of each lane as depicted in Figure 28(a). The intersection parameters and the vehicle parameters used in the controller remain unchanged from those employed in Scenario 2.

Figure 28 depicts four snapshots of the intersection plaza at the same time steps of the previous scenario and for the same seed. The corresponding states and control inputs of the vehicles, as well as the distance between selected vehicles in Figure 28, are presented in Figure 29. All vehicles are seen to increase their speed and acceleration once they enter the intersection.

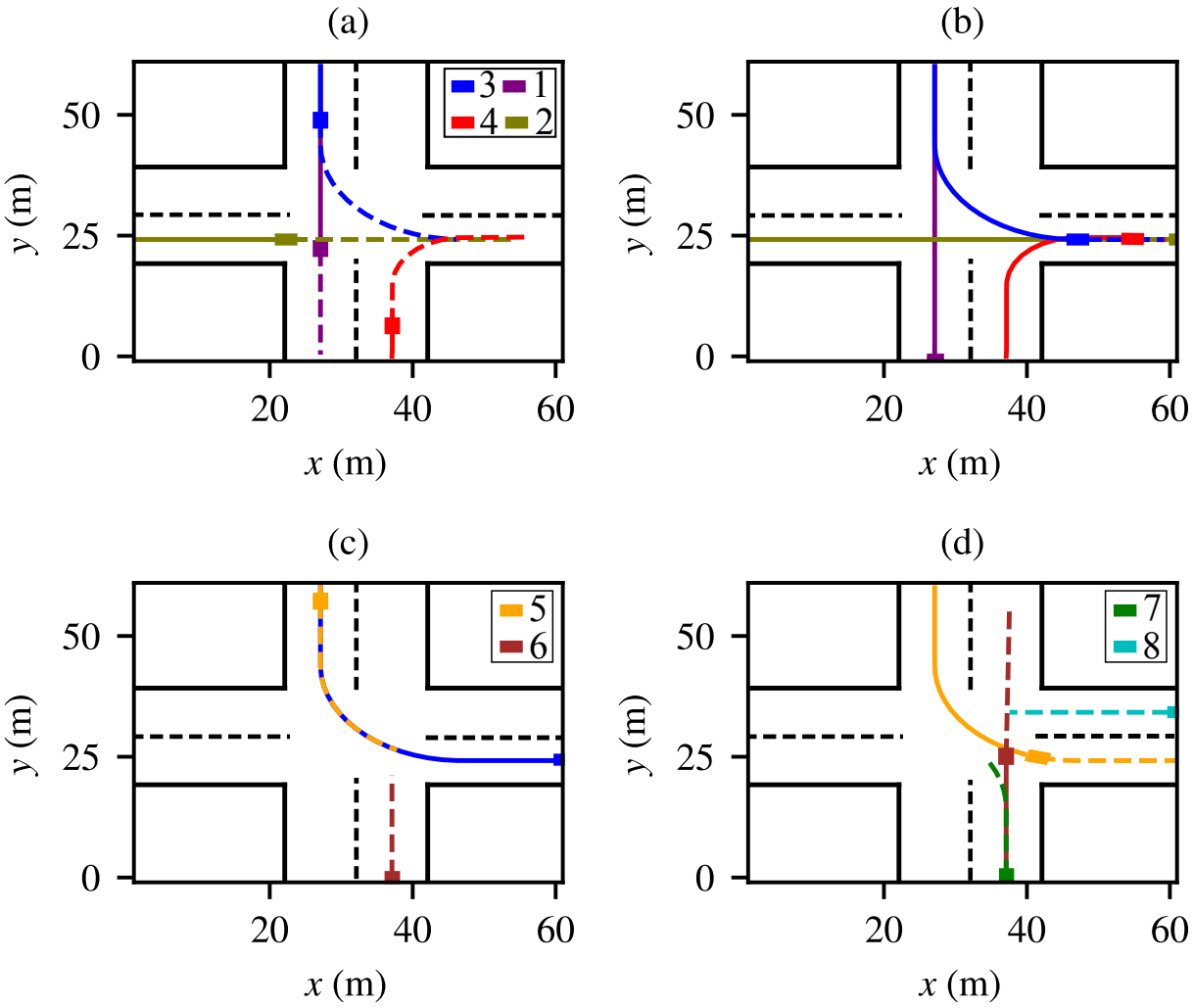


Figure 28 – SC3: snapshots of intersection plaza in time steps (a) 65, (b) 88, (c) 109, and (d) 130 operating under the conventional path-based control with a flow of 1200 veh/h/approach.

However, vehicles 1, 2, 3, 7, and 8 are seen to decrease their speed and acceleration for a short period to avoid a possible collision with other vehicles before accelerating and reaching the maximum allowed speed. This can potentially lead to vehicle delays and result in impaired traffic performance.

Vehicles 2 and 8 show a considerable reduction in their speed and acceleration to prevent a possible collision with vehicles 1 and 6, respectively, as illustrated in Figures 29(a)–(b). In contrast, when using the LTV-ENMPCC algorithm, we see a smoother behavior of the vehicles, as depicted in Figures 27(a)–(b), which is attributed to the path-free solution and the possibility of full use of the intersection space. This can affect the total time spent (TTS).

Furthermore, the distance traveled by the vehicles shown in Figure 29(e) is longer compared to the path-free results illustrated in Figure 27(e). This is because, in the path-based approach, the vehicles travel along the predefined reference paths and are not allowed to deviate from the paths, which can lead to inefficient use of the intersection space. To provide an illustration, consider vehicle 3 as an example. The LTV-ENMPCC algorithm generates a shorter path for

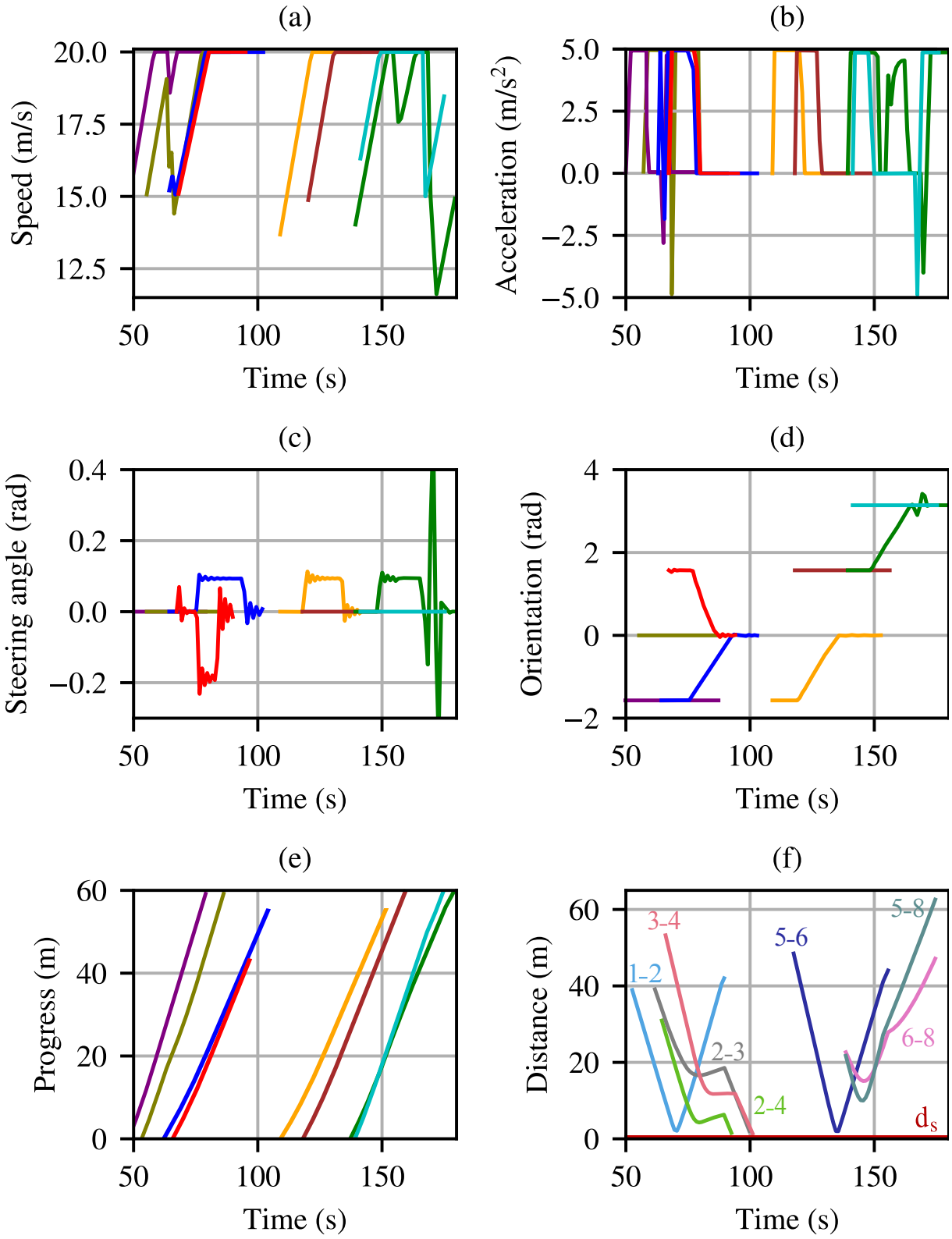


Figure 29 – SC3: states and inputs of the first eight vehicles in the intersection plaza operating under the conventional path-based control with a flow of 1200 veh/h/approach.

vehicle 3 resulting in less traveled distance to exit the intersection as shown in Figure 26(b) and Figure 27(e). In the path-based scenario, however since vehicle 3 is not allowed to deviate from the predefined reference path, it traverses more distance to exit the intersection leading to

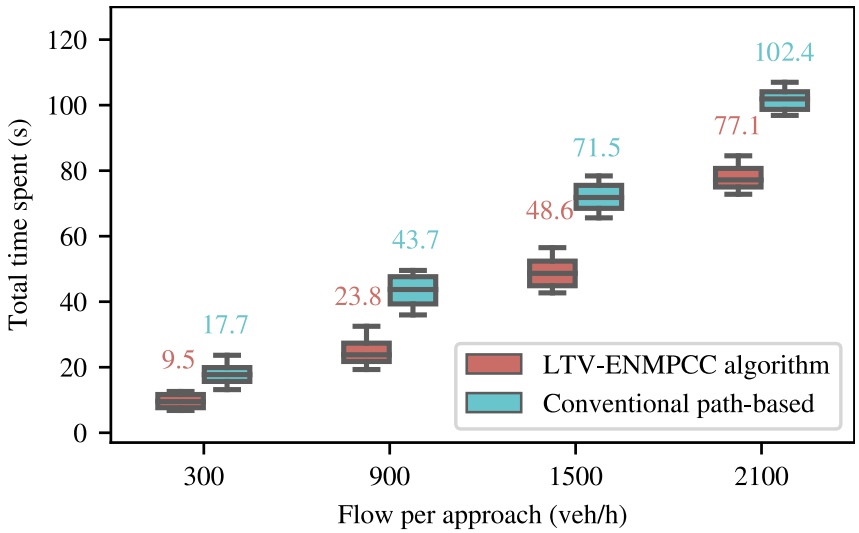


Figure 30 – SC3: total time spent of SC2 (LTV-ENMPCC algorithm) and SC3 (conventional path-based) for flows from 300 to 2100 veh/h/approach. The numbers above each box provide the median values.

longer vehicle progress as shown in Figure 28(b) and Figure 29(e). In general, the conventional path-based approach can lead to longer travel distances for vehicles, as it does not allow them to take advantage of all the available space in the intersection.

5.4.6 Total Time Spent (TTS)

Figure 30 shows the TTS for both the proposed method and the conventional path-based driving for various flow rates, ranging from 300 to 2100 veh/h/approach. By comparing the boxes within each flow rate category, one can observe that the proposed SPIC method generally results in a lower TTS. At all flow rates, the proposed method allows CVAD to follow free-path driving, resulting in faster TTS. However, path-based driving may take longer due to tracking predefined paths and decelerating to avoid collisions, as shown in Figs. 27(a)–(b) and 29(a)–(b). As a result, the LTV-ENMPCC algorithm yields a better performance and constitutes a more promising solution to enhance the overall efficiency of vehicular traffic at urban intersections.

5.4.7 Fuel Consumption

Finally, Table 6 presents the evaluation of the average fuel consumption of both LTV-ENMPCC method and the conventional path-based approach. In this study, we adopt the fuel consumption model presented in Akçelik (1983), which assumes that a vehicle can be in four operational modes; idling, cruising at a constant speed, accelerating, and decelerating. The model uses different formulas to calculate the fuel consumption for each mode, and the parameters of the model are different for different vehicle engines. In the simulations, we use the parameters for the Honda Civic engine 1.8 L vehicle obtained from (MARINHO et al., 2018). As can be seen from the table, the LTV-ENMPCC algorithm demonstrates lower fuel consumption than the conventional path-based method at all flow rates. The differences between the methods become

Table 6 – SC3: average fuel consumption (ml/s) of SC2 (LTV-ENMPCC algorithm) and SC3 (conventional path-based)

Method \ Flow	300	900	1500	2100
LTV-ENMPCC algorithm	53.5	56.1	59.4	65.0
Conventional path-based	57.4	62.0	73.3	85.7

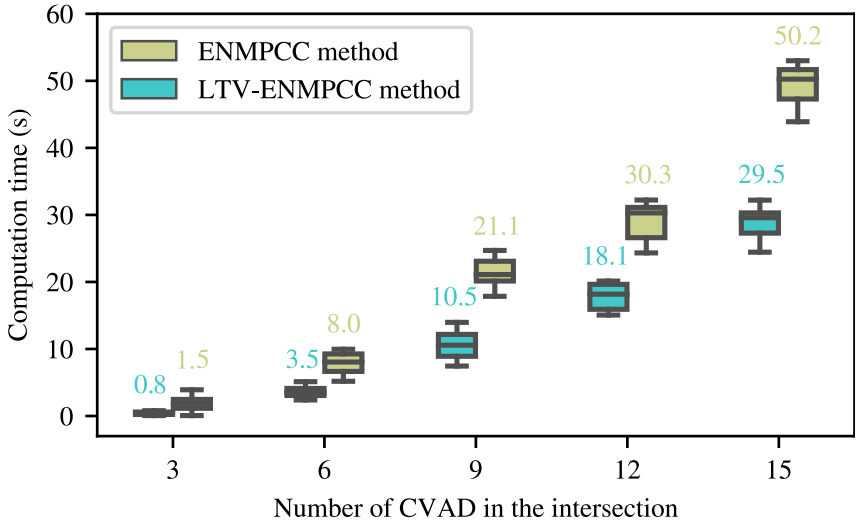


Figure 31 – The average computation time of the ENMPCC and LTV-ENMPCC methods for varying counts of CVAD numbers present in the intersection. The numbers above each box provide the median values.

more evident as the flow rate increases. This suggests that the LTV-ENMPCC algorithm might offer better fuel efficiency.

5.4.8 Computation Time

Figure 31 shows the average computation time of seven runs of the proposed ENMPCC and LTV-ENMPCC methods for different numbers of vehicles in the intersection. The figure demonstrates that the LTV-ENMPCC method speeds up the computation time compared to the proposed ENMPCC method when addressing the SPIC problem. For both methods, the computation time is relatively low when the number of vehicles is small, but it increases as the number of vehicles increases. The computation time can be improved by leveraging the separable structure of the SPIC problem and developing a distributed and parallel algorithm to exploit this structure which is considered for future work.

5.5 CONCLUSION

This chapter proposed a method based on the linear time-varying approximation of the extended nonlinear model predictive contouring control (LTV-ENMPCC) algorithm for solving the signal-free path-free intersection control (SPIC) problem. In order to promote traffic safety,

SPIC uses a polytopic representation of the CVAD and incorporates duality theory to satisfy V2V collision avoidance constraints. Additionally, a set of appropriate constraints to avoid collisions between CVAD and intersection plaza boundaries was introduced. This method enables CVAD to optimally use the intersection space while improving their travel times as demonstrated in the simulation results. Although the SPIC algorithm offers some computational advantages over previous solutions, the computation times increase exponentially with the number of vehicles. Therefore, for higher demands, the solution may not be fast enough for real-time applications. Future research will focus on enhancing the SPIC formulation and its solution algorithm.

6 CONCLUSIONS AND FUTURE WORKS

6.1 CONCLUSIONS

This thesis focused on addressing critical challenges in the development of connected vehicles under automated driving (CVAD) decision and control systems. These challenges are crucial for ensuring the safe and effective operation of CVAD at urban intersections. The thesis tackles these challenges from three different perspectives, aiming to provide practical solutions and enhance the reliability of the CVAD decision-making processes.

In Chapter 3, we introduced the signal-free path-free intersection control (SPIC) problem as a new approach to urban intersection management. The SPIC problem was designed to optimize the utilization of intersection space and enhance traffic efficiency by leveraging CVAD technology to enable path-free traversing within a signal-free intersection. Within the plaza, the intersection was transformed into a boundary-constrained free space, where vehicles are allowed to travel along the reference paths, provided that their trajectories do not lead to collisions.

In Chapter 4, we introduced the intersection trajectory optimal control problem (ITOP) as a first attempt to model and solve the SPIC problem. The ITOP simplified the problem by representing vehicle dynamics with an equation of motion of particles. It guaranteed safety by modeling the intersection plaza geometry with multiple exponential functions and avoiding collisions between vehicles using the Euclidean norm. We then transformed the ITOP problem into a nonlinear programming (NLP) optimization problem by employing finite Fourier series (FFS) and Bezier curve methods which generated near-optimal and collision-free trajectories of CVAD. Although the solutions obtained from these methods served as good initial estimates for direct optimal control methods, they showed some limitations. Motivated by the limitations of the ITOP problem, such as the non-linearity and non-differentiability of the boundary constraints, the same completion time for all CVAD, and the need for pre-defined initial and final conditions, in Chapter 5 we proposed a novel formulation of the SPIC problem that addressed these difficulties.

In Chapter 5, the SPIC problem used a kinematic bicycle model to represent the dynamics of CVAD, with the flexibility to consider other vehicle models. Each vehicle was constrained to travel within a well-defined circle that prevented it from violating the intersection boundaries. Circular and polytopic representations of the road region occupied by each CVAD were also considered to define well-posed V2V collision avoidance constraints. We proposed an extended nonlinear model predictive contouring control (ENMPCC) method to solve the SPIC problem. ENMPCC generates optimal and collision-free CVAD trajectories by solving an NLP optimization problem at each time step of the simulation. To improve the numerical performance of ENMPCC, we utilized a linear time-varying (LTV) approximation of the objective functions and the vehicle dynamics which led to better computational efficiency. Compared to the conventional path-based (lane-based) driving of CVAD, the simulation results of the SPIC problem showed that it can improve intersection space utilization, resulting in lower travel time and fuel consumption for the vehicles to traverse the intersection, with guaranteed safety.

6.2 PUBLICATIONS

The SPIC problem and the quest for a suitable solution have been pivotal in shaping the author's doctoral research, leading to the publication of four conference papers and two journal papers, all of which serve as precursors to this thesis.

6.2.1 Conference Papers

1. Elham Ahmadi, Rodrigo Castelan Carlson, Werner Kraus Junior, and Ehsan Taheri. Near-optimal coordination of vehicles at an intersection plaza using Bézier curves. In VEHICULAR 2022: The Eleventh International Conference on Advances in Vehicular Systems, Technologies and Applications, pages 6–12, Italy, IARIA, 2022.
2. Elham Ahmadi and Rodrigo Castelan Carlson. Model predictive control of multi-vehicle interaction at unsignalized intersections. In 36º Congresso Nacional de Pesquisa e Ensino em Transportes, Brasil, ANPET, 2022.
3. Rodrigo Castelan Carlson, Elham Ahmadi, Eduardo Rauh Müller, and Gabriel Wendel Santos da Silva. Optimal coordination of connected vehicles under automated driving at path-free signal-free urban intersections. In 36º Congresso Nacional de Pesquisa e Ensino em Transportes, Brasil, ANPET, 2022.
4. Elham Ahmadi, Rodrigo Castelan Carlson, Werner Kraus Junior, and Ehsan Taheri. Near-optimal coordination of vehicles at an intersection plaza using finite Fourier series. In 35º Congresso Nacional de Pesquisa e Ensino em Transportes, Brasil, ANPET, 2021.
5. Roberto Willoughby Maya, Elham Ahmadi, and Rodrigo Castelan Carlson. On-ramp merging in lane-free freeways using the SPIC algorithm. In 37º Congresso Nacional de Pesquisa e Ensino em Transportes, Brasil, ANPET (Accepted), 2023.

6.2.2 Journal Papers

1. Elham Ahmadi, Alireza Olama, Rodrigo Castelan Carlson, and Eduardo Camponogara. Signal-free path-free intersection control. IEEE Transactions on Intelligent Transportation Systems (Under Review). IEEE, 2023.
2. Elham Ahmadi and Rodrigo Castelan Carlson. Model predictive control of connected vehicles under automated driving at path-free signal-free intersections. Transportes (Accepted), 2023.

6.3 FUTURE WORKS

The results presented in this thesis demonstrate the potential of the SPIC and the proposed methods in solving the trajectory optimization problem of CVAD at signal-free and path-free

urban intersections. In this section, we provide an overview of several future research directions with significant potential to be explored.

- Exploit Frenet coordinates: A prospective research direction entails exploring the application of the Frenet coordinate system (FCS) over the Cartesian coordinate system within the ENMPCC method. The FSC can lead to better path-following accuracy and tuning since it eliminates the requirement to define the lag error.
- Improve computational efficiency: The ENMPCC methods proposed in this thesis rely on solving a nonlinear optimization problem at each sampling time. The numerical complexity of this optimization problem increases exponentially with the number of vehicles making it computationally expensive to solve in real-time. Therefore, improving the computational efficiency of the proposed methods is of significant importance. One research direction that has the potential to explore and is currently under investigation is to reformulate the SPIC problem as a bilinear optimization problem that can be solved efficiently by commercial optimization solvers such as GUROBI.
- Distributed optimization: The SPIC problem can be seen as a distributed problem in which the vehicles can collaboratively solve the associated optimization problem. A promising future direction is to exploit the separability structure of the SPIC problem and develop distributed optimization algorithms in which a significant portion of the optimization problem is solved in parallel which can lead to a lower computational time.
- Different Plaza layouts: The SPIC problem can be used in different plaza layouts than urban intersections. A possible research direction is to customize and use the SPIC formulation and the proposed methods for different plaza layouts such as roundabouts, three-leg intersections, and on-ramp merging in lane-free freeways. Currently, we are investigating the application of the SPIC problem for on-ramp merging.
- Comprehensive network of intersections, extending the applicability of the proposed method beyond individual intersections. An essential focus lies in adapting the model and control approach to accommodate a mixed-traffic vehicular system, where connected vehicles under automated driving (CVAD) coexist with human-driven vehicles, thereby enhancing the efficacy of the proposed framework. Additionally, future investigations will delve into refining the methodology to address stochastic disturbances and parametric uncertainties. A critical consideration involves refining the assumed perfect communication between vehicles and infrastructure by incorporating a more realistic model of communication between vehicles into the problem. Furthermore, the research will aim to incorporate passenger comfort into the optimization problem, specifically by minimizing jerk, and assess the resultant impact on the trajectory optimization algorithms.

- Experimental and real-time implementation: Another research direction is to reform the proposed methods to be implemented in a laboratory setting, preceding real-world testing.

REFERENCES

AGGARWAL, Shubhani; KUMAR, Neeraj. Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges. **Computer Communications**, Elsevier, v. 149, p. 270–299, 2020.

AGHA-MOHAMMADI; ALI-AKBAR; CHAKRAVORTY, Suman; AMATO, Nancy M. FIRM: Feedback controller-based Information-state RoadMap-a framework for motion planning under uncertainty. In: INTERNATIONAL Conference on Intelligent Robots and Systems. San Francisco, California, USA: IEEE, 2011. P. 4284–4291.

AHMADI, Elham; CARLSON, Rodrigo Castelan; KRAUS JUNIOR, Werner; TAHERI, Ehsan. Near-optimal coordination of vehicles at an intersection Plaza using finite Fourier series. In: 35º Congresso Nacional de Pesquisa e Ensino em Transportes. online: ANPET, 2021.

AHMED, Hafiz Usman; HUANG, Ying; LU, Pan; BRIDGELALL, Raj. Technology developments and impacts of connected and autonomous vehicles: An overview. **Smart Cities**, MDPI, v. 5, n. 1, p. 382–404, 2022.

AKÇELIK, Rahmi. Progress in fuel consumption modeling for urban traffic management. In: AUSTRALIAN Road Research Board. Vermont South, Australia: Report ARR, 1983. P. 51–56.

AMOUZADI, Mahdi; ORISATOKI, Mobolaji Olawumi; DIZQAH, Arash M. Optimal lane-free crossing of CAVs through intersections. **IEEE Transactions on Vehicular Technology**, v. 72, p. 1488–1500, 2022.

ANDERSON, James M; NIDHI, Kalra; STANLEY, Karlyn D; SORENSEN, Paul; SAMARAS, Constantine; OLUWATOLA, Oluwatobi A. **Autonomous Vehicle Technology: A Guide for Policymakers**. [S.l.]: Rand Corporation, 2014.

ANDERSSON, Joel AE; GILLIS, Joris; HORN, Greg; RAWLINGS, James B; DIEHL, Moritz. CasADi – A software framework for nonlinear optimization and optimal control. **Mathematical Programming Computation**, Springer, v. 11, n. 1, p. 1–36, 2019.

BERNTORP, Karl. Path planning and integrated collision avoidance for autonomous vehicles. In: AMERICAN Control Conference. Seattle, WA, USA: ACC, 2017. P. 4023–4028.

BERTSEKAS, Dimitri. **Convex Optimization Theory**. [S.l.]: Athena Scientific, 2009. v. 1.

BETTS, John T. **Practical methods for optimal control and estimation using nonlinear programming**. [S.l.]: SIAM, 2010.

BETTS, John T. Survey of numerical methods for trajectory optimization. **Journal of guidance, control, and dynamics**, v. 21, n. 2, p. 193–207, 1998.

BOYD, Stephen P; VANDENBERGHE, Lieven. **Convex Optimization**. [S.l.]: Cambridge University Press, 2004.

BRIN, Michael; STUCK, Garrett. **Introduction to Dynamical Systems**. [S.l.]: Cambridge University Press, 2002.

BROSOWSKY, Mathis; KECK, Florian; DÜNKEL, Olaf; ZÖLLNER, Marius. Sample-specific output constraints for neural networks. In: 8. PROCEEDINGS of the AAAI Conference on Artificial Intelligence. online: AAAI, 2021. v. 35, p. 6812–6821.

BROSSETTE, Stanislas; WIEBER, Pierre-Brice. Collision avoidance based on separating planes for feet trajectory generation. In: 17TH International Conference on Humanoid Robotics (Humanoids). Birmingham, UK: IEEE, 2017. P. 509–514.

BUEHLER, Martin; IAGNEMMA, Karl; SINGH, Sanjiv. **The DARPA Urban Challenge: Autonomous Vehicles in City Traffic**. [S.l.]: Springer, 2009. v. 56.

CHEN, Lei; ENGLUND, Cristofer. Cooperative intersection management: A survey. **IEEE Transactions on Intelligent Transportation Systems**, v. 17, n. 2, p. 570–586, 2016.

CHOE, Ronald. **Distributed Cooperative Trajectory Generation for Multiple Autonomous Vehicles using Pythagorean Hodograph Bézier Curves**. 2017. PhD thesis – University of Illinois at Urbana-Champaign.

CHU, Jing. **Dynamics, Distributed Control and Autonomous Cluster Operations of Fractionated Spacecraft**. 2015. PhD thesis – Delft University of Technology.

CONWAY, Bruce A. A survey of methods available for the numerical optimization of continuous dynamic systems. **Journal of Optimization Theory and Applications**, Springer, v. 152, n. 2, p. 271–306, 2012.

D'ASPREMONT, Alexandre; BOYD, Stephen. Relaxations and randomized methods for nonconvex QCQPs. **EE392o Class Notes, Stanford University**, Citeseer, v. 1, p. 1–16, 2003.

DACIC, Dragan B; NESIC, Dragan; KOKOTOVIC, Petar V. Path-following for nonlinear systems with unstable zero dynamics. **IEEE transactions on automatic control**, v. 52, n. 3, p. 481–487, 2007.

DAČIĆ, Dragan B; KOKOTOVIĆ, Petar V. Path-following for linear systems with unstable zero dynamics. **Automatica**, Elsevier, v. 42, n. 10, p. 1673–1683, 2006.

DAM, Floris van. **Distributed Collision Free Trajectory Optimization for the Reconfiguration of a Spacecraft Formation**. 2019. PhD thesis – Delft University of Technology.

DRESNER, Kurt; STONE, Peter. A multiagent approach to autonomous intersection management. **Journal of Artificial Intelligence Research**, v. 31, p. 591–656, Mar. 2008.

DRESNER, Kurt; STONE, Peter. Multiagent traffic management: A reservation-based intersection control mechanism. **New York**, p. 8, 2004.

EL KHALICK M., A.; UCHIYAMA, Naoki. Discrete-time model predictive contouring control for biaxial feed drive systems and experimental verification. **Mechatronics**, Elsevier, v. 21, n. 6, p. 918–926, 2011.

ERICSON, Christer. **Real-time Aollision Detection**. [S.l.]: CRC Press, 2004.

EVERETT, Michael; CHEN, Yu Fan; HOW, Jonathan P. Motion planning among dynamic, decision-making agents with deep reinforcement learning. In: INTERNATIONAL Conference on Intelligent Robots and Systems (IROS). Madrid, Spain: IEEE, 2018. P. 3052–3059.

FALCONE, Paolo; BORRELLI, Francesco; ASGARI, Jahan; TSENG, Hongtei Eric; HROVAT, Davor. Predictive active steering control for autonomous vehicle systems. **IEEE Transactions on Control Systems Technology**, v. 15, n. 3, p. 566–580, 2007.

FAN, Zichen; HUO, Mingying; XU, Song; ZHAO, Jun; QI, Naiming. Fast cooperative trajectory optimization for close-range satellite formation using Bézier shape-based method. **IEEE Access**, v. 8, p. 30918–30927, 2020.

FAROUKI, Rida T. **Pythagorean—hodograph Curves**. [S.l.]: Springer, 2008.

FAULWASSER, Timm; KERN, Benjamin; FINDEISEN, Rolf. Model predictive path-following for constrained nonlinear systems. In: PROCEEDINGS of the 48th IEEE Conference on Decision and Control held jointly with the 28th Chinese Control Conference. Shanghai, China: IEEE, 2009. P. 8642–8647.

FHWA. **Intersection safety**. 2021. Available from: `https://safety.fhwa.dot.gov/intersection/about/index.cfm`. Visited on: 30 Sept. 2010.

FHWA. **Traffic Analysis Toolbox Volume III: Guidelines for Applying Traffic Microsimulation Modeling Software**. Available from: `https://ops.fhwa.dot.gov/trafficanalysistools/tat_vol3/sectapp_a.htm`. Visited on: 31 Aug. 2022.

GALCERAN, Enric; CARRERAS, Marc. A survey on coverage path planning for robotics. **Robotics and Autonomous systems**, Elsevier, v. 61, n. 12, p. 1258–1276, 2013.

GERDTS, Matthias; HENRION, René; HÖMBERG, Dietmar; LANDRY, Chantal. Path planning and collision avoidance for robots, 2011.

GILBERT, Elmer; KOLMANOVSKY, Ilya. Nonlinear tracking control in the presence of state and control constraints: a generalized reference governor. **Automatica**, Elsevier, v. 38, n. 12, p. 2063–2073, 2002.

GONDELACH, David J; NOOMEN, Ron. Hodographic-shaping method for low-thrust interplanetary trajectory design. **Journal of Spacecraft and Rockets**, American Institute of Aeronautics and Astronautics, v. 52, n. 3, p. 728–738, 2015.

HAGENAARS, HL; IMURA, Jun-ichi; NIJMEIJER, Henk. Approximate continuous-time optimal control in obstacle by time/space discretization of non-convex constraints. In: INTERNATIONAL Conference on Control Applications, Chicago, USA: IEEE, 2004. v. 2, p. 878–883.

HAUSER, John; HINDMAN, Rick. Maneuver regulation from trajectory tracking: Feedback linearizable systems. **IFAC Proceedings Volumes**, Elsevier, v. 28, n. 14, p. 595–600, 1995.

HE, Zhengbing; ZHENG, Liang; LU, Lili; GUAN, Wei. Erasing lane changes from roads: A design of future road intersections. **IEEE Transactions on Intelligent Vehicles**, v. 3, n. 2, p. 173–184, 2018.

HONG, Joey; SAPP, Benjamin; PHILBIN, James. Rules of the road: Predicting driving behavior with a convolutional model of semantic interactions. In: CONFERENCE on Computer Vision and Pattern Recognition. California, USA: IEEE, 2019. P. 8454–8462.

HUANG, Sunan; TEO, Rodney Swee Huat; TAN, Kok Kiong. Collision avoidance of multi unmanned aerial vehicles: A review. **Annual Reviews in Control**, Elsevier, v. 48, p. 147–164, 2019.

INTERNATIONAL, SAE. **J3016, Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-road Motor Vehicles**. [S.l.: s.n.], 2021.

KANJANAWANISHKUL, Kiattisin; HOFMEISTER, Marius; ZELL, Andreas. Smooth reference tracking of a mobile robot using nonlinear model predictive control. In: 4TH European Conference on Mobile Robots. Croatia: ECMR, 2009. P. 161–166.

KATRAKAZAS, Christos; QUDDUS, Mohammed; CHEN, Wen-Hua; DEKA, Lipika. Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions. **Transportation Research Part C: Emerging Technologies**, Elsevier, v. 60, p. 416–442, 2015.

KELLY, Matthew. An introduction to trajectory optimization: How to do your own direct collocation. **SIAM Review**, SIAM, v. 59, n. 4, p. 849–904, 2017.

KEVICZKY, TamÁs; BORRELLI, Francesco; FREGENE, Kingsley; GODBOLE, Datta; BALAS, Gary J. Decentralized receding horizon control and coordination of autonomous

vehicle formations. **IEEE Transactions on Control Systems Technology**, v. 16, n. 1, p. 19–33, 2008.

KIM, H Jin; SHIM, David H; SASTRY, Shankar. Nonlinear model predictive tracking control for rotorcraft-based unmanned aerial vehicles. In: AMERICAN control conference. Anchorage, Alaska, USA: IEEE, 2002. v. 5, p. 3576–3581.

KONG, Jason; PFEIFFER, Mark; SCHILDBACH, Georg; BORRELLI, Francesco. Kinematic and dynamic vehicle models for autonomous driving control design. In: IEEE Intelligent Vehicles Symposium. Seoul, South Korea: IEEE, 2015. P. 1094–1099.

KOREN, Y; LO, Ch-Ch. Advanced controllers for feed drives. **CIRP Annals**, Elsevier, v. 41, n. 2, p. 689–698, 1992.

KOREN, Yoram. Control of machine tools, 1997.

KOREN, Yoram; LO, Ch-Ch. Variable-gain cross-coupling controller for contouring. **CIRP Annals**, Elsevier, v. 40, n. 1, p. 371–374, 1991.

LAM, Denise. **A Model Predictive Approach to Optimal Path-Following and Contouring Control**. 2012. PhD thesis – University of Melbourne, Department of Mechanical Engineering.

LAM, Denise; MANZIE, Chris; GOOD; C., Malcolm. Model predictive contouring control for biaxial systems. **IEEE Transactions on Control Systems Technology**, v. 21, n. 2, p. 552–559, 2013.

LAM, Denise; MANZIE, Chris; GOOD, Malcolm. Model predictive contouring control. In: 49 Conference on Decision and Control (CDC). Atlanta, Georgia, USA: IEEE, 2010. P. 6137–6142.

LATTARULO, Ray; GONZÁLEZ, Leonardo; MARTÍ, Enrique; MATUTE, José; MARCANO, Mauricio; PÉREZ, Joshue. Urban motion planning framework based on n-Bézier curves considering comfort and safety. **Journal of Advanced Transportation**, Hindawi, v. 2018, 2018.

LAW, Averill M. **Simulation Modeling and Analysis**. 5. ed. Dubuque: McGraw-Hill Education, 2013. ISBN 978-0-07-340132-4.

LEE, Joyoung; PARK, Byungkyu. Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment. **IEEE Transactions on Intelligent Transportation Systems**, v. 13, n. 1, p. 81–90, 2012.

LEVIN, Michael W; REY, David. Conflict-point formulation of intersection control for autonomous vehicles. **Transportation Research Part C: Emerging Technologies**, Elsevier, v. 85, p. 528–547, 2017.

LEVY, Rotem; HADDAD, Jack. Cooperative path and trajectory planning for autonomous vehicles on roads without lanes: A laboratory experimental demonstration. **Transportation Research Part C: Emerging Technologies**, Elsevier, v. 144, p. 103813, 2022.

LEVY, Rotem; HADDAD, Jack. Path and trajectory planning for autonomous vehicles on roads without lanes. In: INTERNATIONAL Intelligent Transportation Systems Conference (ITSC). Indianapolis, IN, USA: IEEE, 2021. P. 3871–3876.

LEWIS, Frank L; VRABIE, Draguna; SYRMOS, Vassilis L. **Optimal Control**. [S.l.]: John Wiley & Sons, 2012.

LI, Bai; ZHANG, Youmin; ACARMAN, Tankut; OUYANG, Yakun; YAMAN, Cagdas; WANG, Yaonan. Lane-free autonomous intersection management: A batch-processing framework integrating reservation-based and planning-based methods. In: INTERNATIONAL Conference on Robotics and Automation (ICRA). China: IEEE, 2021. P. 7915–7921.

LI, Bai; ZHANG, Youmin; JIA, Ning; PENG, Xiaoyan. Autonomous intersection management over continuous space: A microscopic and precise solution via computational optimal control. **IFAC-PapersOnLine**, v. 53, n. 2, p. 17071–17076, Jan. 2020.

LI, Bai; ZHANG, Youmin; ZHANG, Yue; JIA, Ning; GE, Yuming. Near-optimal online motion planning of connected and automated vehicles at a signal-free and lane-free intersection. In: INTELLIGENT Vehicles Symposium (IV). Fisciano, Italy: IEEE, 2018. P. 1432–1437.

LI, Nan; OYLER, Dave W; ZHANG, Mengxuan; YILDIZ, Yildiray; KOLMANOVSKY, Ilya; GIRARD, Anouck R. Game theoretic modeling of driver and vehicle interactions for verification and validation of autonomous vehicle control systems. **IEEE Transactions on Control Systems Technology**, v. 26, n. 5, p. 1782–1797, 2017.

LI, Shengbo Eben; WANG, Zhitao; ZHENG, Yang; SUN, Qi; GAO, Jiaxin; MA, Fei; LI, Keqiang. Synchronous and asynchronous parallel computation for large-scale optimal control of connected vehicles. **Transportation Research Part C: Emerging Technologies**, Elsevier, v. 121, p. 102842, 2020.

LINIGER, Alexander; DOMAHIDI, Alexander; MORARI, Manfred. Optimization-based autonomous racing of 1: 43 scale RC cars. **Optimal Control Applications and Methods**, Wiley Online Library, v. 36, n. 5, p. 628–647, 2015.

LIU, Yuanchang; BUCKNALL, Richard. A survey of formation control and motion planning of multiple unmanned vehicles. **Robotica**, Cambridge University Press, v. 36, n. 7, p. 1019–1047, 2018.

MAC, Thi Thoa; COPOT, Cosmin; TRAN, Duc Trung; DE KEYSER, Robin. Heuristic approaches in robot path planning: A survey. **Robotics and Autonomous Systems**, Elsevier, v. 86, p. 13–28, 2016.

MAKANTASIS, Konstantinos; PAPAGEORGIOU, Markos. Motorway path planning for automated road vehicles based on optimal control methods. **Transportation Research Record**, SAGE Publications Sage CA: Los Angeles, CA, v. 2672, n. 19, p. 112–123, 2018.

MALEKZADEH, Milad; PAPAMICHAIL, Ioannis; PAPAGEORGIOU, Markos. Linear–quadratic regulators for internal boundary control of lane-free automated vehicle traffic. **Control Engineering Practice**, Elsevier, v. 115, p. 104912, 2021.

MARINHO, Arthur Theodoro; ECCEL, Renan Artur Lopes; CARLSON, Rodrigo Castelan; JUNIOR, Werner Kraus. Calibração do modelo de consumo de combustível do simulador AIMSUN com dados coletados via OBD. **Transportes**, v. 26, n. 2, p. 139–154, 2018.

MCNAB, Robert J; TSAO, Tsu-Chin. Receding time horizon linear quadratic optimal control for multi-axis contour tracking motion control. **J. Dyn. Sys., Meas., Control**, v. 122, n. 2, p. 375–381, 2000.

MELSON, Christopher; MA, Jiaqi. Perception, awareness, and importance of preparing for connected and automated vehicle (CAV) technologies: A survey of Louisiana organizations. In: TRAN-SET 2020. Tennessee, USA: American Society of Civil Engineers Reston, VA, 2021. P. 18–27.

MINGUEZ, Javier; LAMIRAUX, Florant; LAUMOND, Jean-Paul. Motion planning and obstacle avoidance. **Springer handbook of robotics**, Springer, p. 1177–1202, 2016.

MINGYING, HUO; ZICHEN, FAN; NAIMING, QI; ZHIGUO, SONG; XIN, SHI. Fast cooperative trajectory optimization and test verification for close-range satellite formation using Finite Fourier Series method. **Chinese Journal of Aeronautics**, Elsevier, v. 33, n. 8, p. 2224–2229, 2020.

MIRHELI, Amir; TAJALLI, Mehrdad; HAJIBABAI, Leila; HAJBABAIE, Ali. A consensus-based distributed trajectory control in a signal-free intersection. **Transportation research part C: emerging technologies**, Elsevier, v. 100, p. 161–176, 2019.

MORGAN, Daniel James. **Guidance and Control of Swarms of Spacecraft**. [S.l.]: University of Illinois at Urbana-Champaign, 2015.

MÜLLER, Eduardo Rauh. **Optimal Arrival Time Scheduling of Automated Vehicles at Intersections**. 2018. Doctoral dissertation – Post-graduate Program in Automation and Systems Engineering, Federal University of Santa Catarina, Florianópolis, SC, Brazil.

MÜLLER, Eduardo Rauh; CARLSON, Rodrigo Castelan; KRAUS, Werner. Time optimal scheduling of automated vehicle arrivals at urban intersections. In: 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC). Rio de Janeiro, Brasil: IEEE, 2016. P. 1174–1179.

NAGARIYA, Akhil; SARIPALLI, Srikanth. An iterative lqr controller for off-road and on-road vehicles using a neural network dynamics model. In: INTELLIGENT Vehicles Symposium (IV). Las Vegas, NV, United States: IEEE, 2020. P. 1740–1745.

NAIR, Siddharth H.; TSENG, Eric H.; BORRELLI, Francesco. Collision avoidance for dynamic obstacles with uncertain predictions using model predictive control. In: 61ST Conference on Decision and Control (CDC). Cancún, Mexico: IEEE, 2022. P. 5267–5272.

NETO, Ninos; JOANNIS, Georgios. **Análise do Gerenciamento de Veículos Automatizados com Exploração Livre da Área da Interseção**. 2021. Master thesis – Post-graduate Program in Automation and Systems Engineering, Federal University of Santa Catarina, Florianópolis, SC, Brazil.

PADEN, Brian; ČÁP, Michal; YONG, Sze Zheng; YERSHOV, Dmitry; FRAZZOLI, Emilio. A survey of motion planning and control techniques for self-driving urban vehicles. **IEEE Transactions on intelligent vehicles**, v. 1, n. 1, p. 33–55, 2016.

PAPAGEORGIOU, Markos; DIAKAKI, Christina; DINOPOULOU, Vaya; KOTSIALOS, Apostolos; WANG, Yibing. Review of road traffic control strategies. **Proceedings of the IEEE**, v. 91, n. 12, p. 2043–2067, 2003.

PAPAGEORGIOU, Markos; MOUNTAKIS, Kyriakos-Simon; KARAFYLLIS, Iasson; PAPAMICHAIL, Ioannis; WANG, Yibing. Lane-free artificial-fluid concept for vehicular traffic. **Proceedings of the IEEE**, IEEE, v. 109, n. 2, p. 114–121, 2021.

PARK, Sangjun; RAKHA, Hesham. Continuous flow intersections: A safety and environmental perspective. In: 13TH International Conference on Intelligent Transportation Systems. Madeira Island, Portugal: IEEE, 2010. P. 85–90.

PATEL, Rushen B; GOULART, Paul J. Trajectory generation for aircraft avoidance maneuvers using online optimization. **Journal of Guidance, Control, and Dynamics**, v. 34, n. 1, p. 218–230, 2011.

PATRIKALAKIS, Nicholas M; MAEKAWA, Takashi. **Shape Interrogation for Computer Aided Design and Manufacturing**. [S.l.]: Springer, 2002. v. 15.

PETROPOULOS, Anastassios E; LONGUSKI, James M. Shape-based algorithm for the automated design of low-thrust, gravity assist trajectories. **Journal of Spacecraft and Rockets**, v. 41, n. 5, p. 787–796, 2004.

PETROPOULOS, Anastassios Evangelos. **A Shape-Based Approach to Automated, Low-Thrust, Gravity-Assist Trajectory Design.** [S.l.]: Purdue University, 2001.

POLACK, Philip; ALTCHÉ, Florent; D'ANDRÉA-NOVEL, Brigitte; LA FORTELLE, Arnaud de. The kinematic bicycle model: A consistent model for planning

feasible trajectories for autonomous vehicles? In: IEEE Intelligent Vehicles Symposium. California, USA: IEEE, 2017. P. 812–818.

PYTHON SOFTWARE FOUNDATION. **Python 3.8.13 Documentation**. https://www.python.org/: [s.n.], 2022.

RAJAMANI, Rajesh. **Vehicle Dynamics and Control**. USA: Springer Science & Business Media, 2011.

RAO, Anil V. A survey of numerical methods for optimal control. **Advances in the Astronautical Sciences**, Univelt, Inc., v. 135, n. 1, p. 497–528, 2009.

RIOS-TORRES, Jackeline; MALIKOPOULOS, Andreas A. A Survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps. **IEEE Transactions on Intelligent Transportation Systems**, v. 18, n. 5, p. 1066–1077, 2017.

ROCA, Damian; MILITO, Rodolfo; NEMIROVSKY, Mario; VALERO, Mateo. Advances in the hierarchical emergent behaviors (HEB) Approach to autonomous vehicles. **IEEE Intelligent Transportation Systems Magazine**, v. 12, n. 4, p. 57–65, 2020.

ROCA, Damian; NEMIROVSKY, Daniel; NEMIROVSKY, Mario; MILITO, Rodolfo; VALERO, Mateo. Emergent behaviors in the internet of things: The ultimate ultra-large-scale system. **IEEE micro**, v. 36, n. 6, p. 36–44, 2016.

ROESS, Royer P. **Traffic Engineering**. [S.l.]: United States of Anerica, 2004.

ROMERO, Angel; SUN, Sihao; FOEHN, Philipp; SCARAMUZZA, Davide. Model predictive contouring control for time-optimal quadrotor flight. **IEEE Transactions on Robotics**, 2022.

ROSSITER, JA. **A First Course in Predictive Control**. [S.l.]: CRC Press, 2018.

SAERENS, Bart; DIEHL, Moritz; VAN DEN BULCK, Eric. Optimal control using Pontryagin's maximum principle and dynamic programming. In: AUTOMOTIVE Model Predictive Control. [S.l.]: Springer, 2010. P. 119–138.

SAMSON, Claude. Path following and time-varying feedback stabilization of a wheeled mobile robot. In: INTERNATIONAL Conference on Control, Automation, Robotics and Vision. Singapore: ICARCV, 1992. v. 13.

SCHEPPERLE, Heiko; BÖHM, Klemens. Valuation-aware Traffic Control: The Notion and the Issues. In: MULTI-AGENT systems for traffic and transportation engineering. [S.l.]: IGI Global, 2009. P. 218–239.

SCHWARTING, Wilko; ALONSO-MORA, Javier; PAULL, Liam; KARAMAN, Sertac; RUS, Daniela. Safe nonlinear trajectory generation for parallel autonomy with a dynamic

vehicle model. **IEEE Transactions on Intelligent Transportation Systems**, v. 19, n. 9, p. 2994–3008, 2017.

SCHWARTING, Wilko; ALONSO-MORA, Javier; PAULL, Liam; KARAMAN, Sertac; RUS, Daniela. Safe nonlinear trajectory generation for parallel autonomy with a dynamic vehicle model. **IEEE Transactions on Intelligent Transportation Systems**, v. 19, n. 9, p. 2994–3008, 2018.

SCHWENZER, Max; AY, Muzaffer; BERGS, Thomas; ABEL, Dirk. Review on model predictive control: An engineering perspective. **The International Journal of Advanced Manufacturing Technology**, Springer, v. 117, n. 5-6, p. 1327–1349, 2021.

SCHWUNG, Michael; LUNZE, Jan. Networked event-based collision avoidance of mobile objects with trajectory planning based on Bézier curves. **European Journal of Control**, Elsevier, v. 58, p. 327–339, 2021.

SEKERAN, Maya; ROSTAMI-SHAHRBABAKI, Majid; SYED, Arslan Ali; MARGREITER, Martin; BOGENBERGER, Klaus. Lane-free traffic: History and state of the art. In: 25TH International Conference on Intelligent Transportation Systems. China: ITSC, 2022. P. 1037–1042.

SHARMA, Omveer; SAHOO, Nirod C.; PUHAN, Niladri B. Recent advances in motion and behavior planning techniques for software architecture of autonomous vehicles: A state-of-the-art survey. **Engineering Applications of Artificial Intelligence**, Elsevier, v. 101, p. 104211, 2021.

SHIRAZI, Abolfazl; CEBERIO, Josu; LOZANO, Jose A. Spacecraft trajectory optimization: A review of models, objectives, approaches and solutions. **Progress in Aerospace Sciences**, Elsevier, v. 102, p. 76–98, 2018.

SHLADOVER, Steven E. Connected and automated vehicle systems: Introduction and overview. **Journal of Intelligent Transportation Systems**, Taylor & Francis, v. 22, n. 3, p. 190–200, 2018.

SICILIANO, Bruno; KHATIB, Oussama; KRÖGER, Torsten. **Springer Handbook of Robotics**. Ed. by Bruno Siciliano and Oussama Khatib. [S.l.]: Springer, 2008. v. 200.

SIEGEL, Joshua E.; ERB, Dylan C.; SARMA, Sanjay E. A survey of the connected vehicle landscape—architectures, enabling technologies, applications, and development areas. **IEEE Transactions on Intelligent Transportation Systems**, v. 19, n. 8, p. 2391–2406, 2018.

STEVANOVIC, Aleksandar; MITROVIC, Nikola. Impact of conflict resolution parameters on combined alternate-directions lane assignment and reservation-based intersection control. **European transport research review**, SpringerOpen, v. 12, n. 1, p. 1–10, 2020.

SUNDAR, Satish; SHILLER, Zvi. Time-optimal obstacle avoidance. In: INTERNATIONAL Conference on Robotics and Automation. Japan: IEEE, 1995. v. 3, p. 3075–3080.

SUSANU, Mara; DUMUR, Didier. Hierarchical predictive control within an open architecture virtual machine tool. **CIRP annals**, Elsevier, v. 55, n. 1, p. 389–392, 2006.

TAHERI, Ehsan; ABDELKHALIK, Ossama. Initial three-dimensional low-thrust trajectory design. **Advances in Space Research**, v. 57, n. 3, p. 889–903, 2016.

TAHERI, Ehsan; ABDELKHALIK, Ossama. Shape based approximation of constrained low-thrust space trajectories using Fourier series. **Journal of Spacecraft and Rockets**, v. 49, n. 3, p. 535–546, 2012.

TAZAKI, Yuichi; MUROOKA, Masaki. A survey of motion planning techniques for humanoid robots. **Advanced Robotics**, Taylor & Francis, v. 34, n. 21-22, p. 1370–1379, 2020.

TYPALDOS, Panagiotis; PAPAGEORGIOU, Markos; PAPAMICHAIL, Ioannis. Optimization-based path-planning for connected and non-connected automated vehicles. **arXiv preprint arXiv:2104.06778**, 2021.

ULSOY, A. Galip; PENG, Huei; CAKMAKCI, Melih. **Automotive Control Systems**. [S.l.]: Cambridge University Press, 2012.

VAGALE, Anete; OUCHEIKH, Rachid; BYE, Robin T; OSEN, Ottar L; FOSSEN, Thor I. Path planning and collision avoidance for autonomous surface vehicles I: a review. **Journal of Marine Science and Technology**, Springer, p. 1–15, 2021.

VIRDI, Jaskaran. **Using Deep Learning to Predict Obstacle Trajectories for Collision Avoidance in Autonomous Vehicles**. [S.l.]: University of California, San Diego, 2018.

WÄCHTER, Andreas; BIEGLER, Lorenz T. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. **Mathematical Programming**, Springer, v. 106, n. 1, p. 25–57, 2006.

WANG, Hongling; KEARNEY, Joseph; ATKINSON, Kendall. Arc-length parameterized spline curves for real-time simulation. In: PROC. 5th International Conference on Curves and Surfaces. France: Curves and Surfaces, 2002. v. 387396.

YOUAKIM, Dina; RIDAO, Pere. Motion planning survey for autonomous mobile manipulators underwater manipulator case study. **Robotics and Autonomous Systems**, Elsevier, v. 107, p. 20–44, 2018.

YU, Chunhui; FENG, Yiheng; LIU, Henry X; MA, Wanjing; YANG, Xiaoguang. Corridor level cooperative trajectory optimization with connected and automated vehicles. **Transportation Research Part C: Emerging Technologies**, Elsevier, v. 105, p. 405–421, 2019.

ZENG, Kui; GENG, Yunhai; WU, Baolin. Shape-based analytic safe trajectory design for spacecraft equipped with low-thrust engines. **Aerospace Science and Technology**, Elsevier, v. 62, p. 87–97, 2017.

ZHANG, Xiaojing; LINIGER, Alexander; BORRELLI, Francesco. Optimization-based collision avoidance. **IEEE Transactions on Control Systems Technology**, v. 29, n. 3, p. 972–983, 2020.

ZHANG, Yue; CASSANDRAS, Christos G. An impact study of integrating connected automated vehicles with conventional traffic. **Annual Reviews in Control**, Elsevier, v. 48, p. 347–356, 2019.

ZHAO, Jiang; ZHOU, Rui. Reentry trajectory optimization for hypersonic vehicle satisfying complex constraints. **Chinese Journal of Aeronautics**, Elsevier, v. 26, n. 6, p. 1544–1553, 2013.

ZHU, Feng; UKKUSURI, Satish V. A linear programming formulation for autonomous intersection control within a dynamic traffic assignment and connected vehicle environment. **Transportation Research Part C: Emerging Technologies**, Elsevier, v. 55, p. 363–378, 2015.

ZHU, Yuanpeng; HAN, Xuli; HAN, Jing, et al. Quartic trigonometric Bézier curves and shape preserving interpolation curves. **Journal of Computational Information Systems**, v. 8, n. 2, p. 905–914, 2012.

APPENDIX A – FINITE FOURIER SERIES METHOD DERIVATION

In the intersection trajectory optimal control problem (ITOP), there are eight boundary conditions (BCs), the initial and final positions and speeds, that need to be satisfied with respect to scaled time τ. Using the finite Fourier series (FFS) representation of states, (22), and by taking its derivative, the following equation can be constructed:

$$\mathbf{z}'(\tau) = \sum_{n=1}^{N_\mathbf{z}} \left(-\pi n b_n^\mathbf{z} \sin(\pi n\tau) + \pi n c_n^\mathbf{z} \cos(\pi n\tau)\right). \tag{75}$$

A.1 EXPRESSING SOME FINITE FOURIER SERIES COEFFICIENTS USING BOUNDARY CONDITIONS

Because the first eight coefficients are the ones obtained for $n = 1{,}2$, (22) and (75) can be rewritten as:

$$\begin{aligned}\mathbf{z}(\tau) = &\frac{b_0^\mathbf{z}}{2} + b_1^\mathbf{z}\cos(\pi\tau) + c_1^\mathbf{z}\sin(\pi\tau) + b_2^\mathbf{z}\cos(2\pi\tau) + c_2^\mathbf{z}\sin(2\pi\tau)\\ &+ \sum_{n=3}^{N_\mathbf{z}} \left(b_n^\mathbf{z}\cos(\pi n\tau) + c_n^\mathbf{z}\sin(\pi n\tau)\right),\end{aligned} \tag{76}$$

$$\begin{aligned}\mathbf{z}'(\tau) = &-\pi b_1^\mathbf{z}\sin(\pi\tau) + \pi c_1^\mathbf{z}\cos(\pi\tau) - 2\pi b_2^\mathbf{z}\sin(2\pi\tau) + 2\pi c_2^\mathbf{z}\cos(2\pi\tau)\\ &+ \sum_{n=3}^{N_\mathbf{z}} \left(-\pi n b_n^\mathbf{z}\sin(\pi n\tau) + \pi n c_n^\mathbf{z}\cos(\pi n\tau)\right).\end{aligned} \tag{77}$$

By replacing the BCs (24) in Eqs. (76) and (77) we obtain:

$$\mathbf{z}_\mathrm{I} = \frac{b_0^\mathbf{z}}{2} + b_1^\mathbf{z} + b_2^\mathbf{z} + \sum_{n=3}^{N_\mathbf{z}} b_n^\mathbf{z}, \tag{78}$$

$$\mathbf{z}_\mathrm{F} = \frac{b_0^\mathbf{z}}{2} - b_1^\mathbf{z} + b_2^\mathbf{z} + \sum_{n=3}^{N_\mathbf{z}} (-1)^n b_n^\mathbf{z}, \tag{79}$$

$$\mathbf{z}'_\mathrm{I} = \pi\left(c_1^\mathbf{z} + 2c_2^\mathbf{z} + \sum_{n=3}^{N_\mathbf{z}} n c_n^\mathbf{z}\right), \tag{80}$$

$$\mathbf{z}'_\mathrm{F} = \pi\left(2c_2^\mathbf{z} - c_1^\mathbf{z} + \sum_{n=3}^{N_\mathbf{z}} (-1)^n n c_n^\mathbf{z}\right). \tag{81}$$

Rewriting (78)–(81) with the four coefficients on the left hand side results in two linear systems:

$$\begin{cases} b_1^\mathbf{z} + b_2^\mathbf{z} = \mathbf{z}_\mathrm{I} - \frac{b_0}{2} - \sum_{n=3}^{N_\mathbf{z}} b_n^\mathbf{z} \\ b_2^\mathbf{z} - b_1^\mathbf{z} = \mathbf{z}_\mathrm{F} - \frac{b_0^\mathbf{z}}{2} - \sum_{n=3}^{N_\mathbf{z}} (-1)^n b_n^\mathbf{z} \end{cases}$$
$$\begin{cases} c_1^\mathbf{z} + 2c_2^\mathbf{z} = \frac{1}{\pi}\dot{\mathbf{z}}_\mathrm{I} - \sum_{n=3}^{N_\mathbf{z}} n c_n^\mathbf{z} \\ 2c_2^\mathbf{z} - c_1^\mathbf{z} = \frac{1}{\pi}\dot{\mathbf{z}}_\mathrm{F} - \sum_{n=3}^{N_\mathbf{z}} (-1)^n n c_n^\mathbf{z}. \end{cases} \tag{82}$$

The solution of these systems gives eight FFS coefficients as presented in (25).

A.2 FINITE FOURIER SERIES COMPACT MATRIX FORM

Replacing back the calculated Fourier coefficients (25) in (76) and organizing the resulting expression gives:

$$\mathbf{z}(\tau) = F_{\mathbf{z}} + C_{b_0^{\mathbf{z}}} b_0^{\mathbf{z}} + \sum_{n=3}^{N_{\mathbf{z}}} (C_{b_n^{\mathbf{z}}} b_n^{\mathbf{z}} + C_{c_n^{\mathbf{z}}} c_n^{\mathbf{z}}) \tag{83}$$

with

$$\begin{aligned} F_{\mathbf{z}} = & \frac{1}{2}(\mathbf{z}_{\mathrm{I}} - \mathbf{z}_{\mathrm{F}})\cos(\pi\tau) + \frac{1}{2\pi}(\dot{\mathbf{z}}_{\mathrm{I}} - \dot{\mathbf{z}}_{\mathrm{F}})\sin(\pi\tau) \\ & + \frac{1}{2}(\mathbf{z}_{\mathrm{I}} + \mathbf{z}_{\mathrm{F}})\cos(2\pi\tau) + \frac{1}{4\pi}(\dot{\mathbf{z}}_{\mathrm{I}} + \dot{\mathbf{z}}_{\mathrm{F}})\sin(2\pi\tau) \end{aligned} \tag{84}$$

$$C_{b_0^{\mathbf{z}}} = \frac{1}{2}(1 - \cos(2\pi\tau)) \tag{85}$$

$$C_{b_n^{\mathbf{z}}} = \begin{cases} \cos(n\pi\tau) - \cos(\pi\tau) & \text{when } n \text{ is odd,} \\ \cos(n\pi\tau) - \cos(2\pi\tau) & \text{when } n \text{ is even,} \end{cases} \tag{86}$$

$$C_{c_n^{\mathbf{z}}} = \begin{cases} \sin(n\pi\tau) - n\sin(\pi\tau) & \text{when } n \text{ is odd,} \\ \sin(n\pi\tau) - \frac{1}{2}n\sin(2\pi\tau) & \text{when } n \text{ is even.} \end{cases} \tag{87}$$

The corresponding first derivative with respect to the scaled time is:

$$\mathbf{z}'(\tau) = F'_{\mathbf{z}} + C'_{b_0^{\mathbf{z}}} b_0^{\mathbf{z}} + \sum_{n=3}^{N_{\mathbf{z}}} (C'_{b_n^{\mathbf{z}}} b_n^{\mathbf{z}} + C'_{c_n^{\mathbf{z}}} c_n^{\mathbf{z}}) \tag{88}$$

with

$$\begin{aligned} F'_{\mathbf{z}} = & -\frac{\pi}{2}(\mathbf{z}_{\mathrm{I}} - \mathbf{z}_{\mathrm{F}})\sin(\pi\tau) + \frac{T}{2}(\dot{\mathbf{z}}_{\mathrm{I}} - \dot{\mathbf{z}}_{\mathrm{F}})\cos(\pi\tau) \\ & - \pi(\mathbf{z}_{\mathrm{I}} + \mathbf{z}_{\mathrm{F}})\sin(2\pi\tau) + \frac{T}{2}(\dot{\mathbf{z}}_{\mathrm{I}} + \dot{\mathbf{z}}_{\mathrm{F}})\cos(2\pi\tau) \end{aligned} \tag{89}$$

$$C'_{b_0^{\mathbf{z}}} = \pi\sin(2\pi\tau) \tag{90}$$

$$C'_{b_n^{\mathbf{z}}} = \begin{cases} -n\pi\sin(n\pi\tau) + \pi\sin(\pi\tau) & \text{when } n \text{ is odd,} \\ -n\pi\sin(n\pi\tau) + 2\pi\sin(2\pi\tau) & \text{when } n \text{ is even,} \end{cases} \tag{91}$$

$$C'_{c_n^{\mathbf{z}}} = \begin{cases} n\pi\cos(n\pi\tau) - n\pi\cos(\pi\tau) & \text{when } n \text{ is odd,} \\ n\pi\cos(n\pi\tau) - n\pi\cos(2\pi\tau) & \text{when } n \text{ is even.} \end{cases} \tag{92}$$

and the second derivative is:

$$\mathbf{z}''(\tau) = F''_{\mathbf{z}} + C''_{b_0^{\mathbf{z}}} b_0^{\mathbf{z}} + \sum_{n=3}^{N_{\mathbf{z}}} (C''_{b_n^{\mathbf{z}}} b_n^{\mathbf{z}} + C''_{c_n^{\mathbf{z}}} c_n^{\mathbf{z}}) \tag{93}$$

with

$$\begin{aligned} F''_{\mathbf{z}} =& -\frac{\pi^2}{2}(\mathbf{z}_{\mathrm{I}} - \mathbf{z}_{\mathrm{F}})\cos(\pi\tau) - \frac{\pi T}{2}(\dot{\mathbf{z}}_{\mathrm{I}} - \dot{\mathbf{z}}_{\mathrm{F}})\sin(\pi\tau) \\ & - 2\pi^2(\mathbf{z}_{\mathrm{I}} + \mathbf{z}_{\mathrm{F}})\cos(2\pi\tau) - \pi T(\dot{\mathbf{z}}_{\mathrm{I}} + \dot{\mathbf{z}}_{\mathrm{F}})\sin(2\pi\tau) \end{aligned} \tag{94}$$

$$C''_{b_0^{\mathbf{z}}} = 2\pi^2\cos(2\pi\tau) \tag{95}$$

$$C''_{b_n^{\mathbf{z}}} = \begin{cases} -(n\pi)^2\cos(n\pi\tau) + \pi^2\cos(\pi\tau) & \text{when } n \text{ is odd,} \\ -(n\pi)^2\cos(n\pi\tau) + 4\pi^2\cos(2\pi\tau) & \text{when } n \text{ is even,} \end{cases} \tag{96}$$

$$C''_{c_n^{\mathbf{z}}} = \begin{cases} -(n\pi)^2\sin(n\pi\tau) + n\pi^2\sin(\pi\tau) & \text{when } n \text{ is odd,} \\ -(n\pi)^2\sin(n\pi\tau) + 2n\pi^2\sin(2\pi\tau) & \text{when } n \text{ is even.} \end{cases} \tag{97}$$

Finally, the compact matrix form representation for the position state variables and its derivatives (speed state variables and accelerations) already incorporating the coefficients from the BCs can be used. Considering bracket $[\cdot]$ as vectors representation, we can write the positions state variables and its first and second derivatives as:

$$\begin{aligned} [\mathbf{z}]_{m\times 1} =& [A_{\mathbf{z}}]_{m\times(2N_{\mathbf{z}}-3)}[X_{\mathbf{z}}]_{(2N_{\mathbf{z}}-3)\times 1} + [F_{\mathbf{z}}]_{m\times 1}, \\ [\mathbf{z}']_{m\times 1} =& [A_{\mathbf{z}'}]_{m\times(2N_{\mathbf{z}}-3)}[X_{\mathbf{z}}]_{(2N_{\mathbf{z}}-3)\times 1} + [F'_{\mathbf{z}}]_{m\times 1}, \\ [\mathbf{z}'']_{m\times 1} =& [A_{\mathbf{z}''}]_{m\times(2N_{\mathbf{z}}-3)}[X_{\mathbf{z}}]_{(2N_{\mathbf{z}}-3)\times 1} + [F''_{\mathbf{z}}]_{m\times 1}, \end{aligned} \tag{98}$$

with

$$\begin{aligned} [X_{\mathbf{z}}]_{(2N_{\mathbf{z}}-3)\times 1} =& [b_0^{\mathbf{z}}\ b_3^{\mathbf{z}}\ c_3^{\mathbf{z}}\ \dots\ b_{N_{\mathbf{z}}}^{\mathbf{z}}\ c_{N_{\mathbf{z}}}^{\mathbf{z}}]^\top, \\ [A_{\mathbf{z}}]_{m\times(2N_{\mathbf{z}}-3)} =& [C_{b_0^{\mathbf{z}}}\ C_{b_3^{\mathbf{z}}}\ C_{c_3^{\mathbf{z}}}\ \dots\ C_{b_{N_{\mathbf{z}}}^{\mathbf{z}}}\ C_{c_{N_{\mathbf{z}}}^{\mathbf{z}}}]_{m\times(2N_{\mathbf{z}}-3)}, \\ [A_{\mathbf{z}'}]_{m\times(2N_{\mathbf{z}}-3)} =& [C'_{b_0^{\mathbf{z}}}\ C'_{b_3^{\mathbf{z}}}\ C'_{c_3^{\mathbf{z}}}\ \dots\ C'_{b_{N_{\mathbf{z}}}^{\mathbf{z}}}\ C'_{c_{N_{\mathbf{z}}}^{\mathbf{z}}}]_{m\times(2N_{\mathbf{z}}-3)}, \\ [A_{\mathbf{z}''}]_{m\times(2N_{\mathbf{z}}-3)} =& [C''_{b_0^{\mathbf{z}}}\ C''_{b_3^{\mathbf{z}}}\ C''_{c_3^{\mathbf{z}}}\ \dots\ C''_{b_{N_{\mathbf{z}}}^{\mathbf{z}}}\ C''_{c_{N_{\mathbf{z}}}^{\mathbf{z}}}]_{m\times(2N_{\mathbf{z}}-3)}, \end{aligned} \tag{99}$$

and $[F_{\mathbf{z}}]_{m\times 1}$, $[F'_{\mathbf{z}}]_{m\times 1}$, and $[F''_{\mathbf{z}}]_{m\times 1}$ obtained from (84), (89), and (94), and $[A_{\mathbf{z}}]_{m\times 1}$, $[A'_{\mathbf{z}}]_{m\times 1}$, and $[A''_{\mathbf{z}}]_{m\times 1}$ obtained from (85), (86), (87), (90), (91), (92), and (95), (96), (97), respectively.

The "F" terms ($F_{\mathbf{z}}$,$F'_{\mathbf{z}}$, and $F''_{\mathbf{z}}$) depend on the boundary information at the position and speed level. Therefore, the speed-level and acceleration-level boundary conditions are updated since they are multiplied by T_{f} and T_{f}^2, respectively. In general, the terms that depend on the boundary conditions are updated at every iteration if T_{f} happens to change during iterations. From this analysis, it results that fixed-time problems are easier and faster to solve. Moreover, for time-free problems, the rest of the structure remains intact, and it provides additional flexibility in the case of intersection management.

A.3 FINITE FOURIER SERIES UNKNOWN COEFFICIENT INITIALIZATION

In the initialization of the unknown coefficients of the FFS method in equation (29), the cubic polynomials can be used to approximate the position state variables $[\mathbf{z}_{\mathrm{a}}]$ at the

Table 7 – Numerical results for different values of w_1 and w_2 ($m = 30$ and $N_{\mathbf{z}} = 8$)

w_1	w_2	Total speed increment Δv [m/s]	Completion time T_f [s]	Objective function $\mathcal{J}$	Computation time [s]
4	0	12.7	27.6	51.1	10.1
0	4	40.7	12.9	51.8	6.4
4	4	15.2	19.3	138.2	9.3
4	2	14.6	18.3	95.3	8.5
2	4	19.4	17.9	110.4	11.5

Table 8 – Numerical results for different values of m ($w_1 = 4$, $w_2 = 2$, and $N_{\mathbf{z}} = 8$)

m	Total speed increment Δv [m/s]	Completion time T_f [s]	Objective function $\mathcal{J}$	Computation time [s]
10	13.2	17.8	88.4	6.6
30	14.6	18.3	95.3	8.5
50	16.2	19.4	103.6	12.4
70	16.9	22.2	112.1	18.7
90	18.1	28.7	129.8	35.9

approximation DPs as:

$$\mathbf{z}_\mathrm{a}(\tau) = \; a_{\mathbf{z}}\tau^3 + b_{\mathbf{z}}\tau^2 + c_{\mathbf{z}}\tau + d_{\mathbf{z}}, \tag{100}$$

for which the coefficients can be obtained from the BCs as follows:

$$\begin{aligned} a_{\mathbf{z}} =& T\dot{\mathbf{z}}_\mathrm{I} + T\dot{\mathbf{z}}_\mathrm{F} + 2(\mathbf{z}_I - \mathbf{z}_F), \\ b_{\mathbf{z}} =& 3(\mathbf{z}_F - \mathbf{z}_I) - 2T\dot{\mathbf{z}}_\mathrm{I} - T\dot{\mathbf{z}}_\mathrm{F}, \\ c_{\mathbf{z}} =& T\dot{\mathbf{z}}_\mathrm{I}, \\ d_{\mathbf{z}} =& \mathbf{z}_I. \end{aligned} \tag{101}$$

A.4 FINITE FOURIER SERIES WEIGHTING PARAMETER SELECTION

In order to find suitable values for m, w_1, w_2, and $N_{\mathbf{z}}$, we solve the NLP problem (28) for different values. Selected numerical results are reported in Tables 7–9. Table 7 presents the results with respect to different values of w_1 and w_2. It is clear in the table the trade-off between completion time and total speed increment (see (14)). We choose $w_1 = 4$ and $w_2 = 2$ for subsequent problem instances because of the relatively low computation time and balance between Δv and T_f.

In Table 8 the results for different numbers of the DPs (m) are reported. As expected, increasing the number of DPs results in increased computation time. Despite the corresponding increase in total speed increment, completion times also increase, suggesting that worse local minima are found for higher values of m, i.e., trajectories in longer paths result. We choose $m = 30$ for subsequent problem instances, sufficiently large to avoid collision in between DPs.

The results for a different number of FFS terms ($N_{\mathbf{z}}$) are presented in Table 9. Increasing $N_{\mathbf{z}}$ increases the computation time without sensible improvements in the other measures. Slight changes in completion time are observed despite the lower total speed increment, indicating that shorter paths are enabled by the increased flexibility in function generation. We choose

Table 9 – Numerical results for different values of $N_{\mathbf{z}}$ ($w_1 = 4$, $w_2 = 2$, and $m = 30$)

$N_{\mathbf{z}}$	Total speed increment Δv [m/s]	Completion time T_{f} [s]	Objective function $\mathcal{J}$	Computation time [s]
6	15.0	17.2	96.4	4.7
8	14.6	18.2	94.9	8.5
10	15.8	19.1	101.7	10.1
12	16.5	19.1	104.2	18.8
14	15.6	18.6	99.6	32.8

$N_{\mathbf{z}} = 8$ to achieve a reasonable trade-off between approximation accuracy and computational burden. Small values of m and $N_{\mathbf{z}}$ may result in better values of Δv, T_{f}, and computation time. However, the trajectories might not be smooth and may also lead to infeasible instances of the NLP problem. On the other hand, large values result in even higher computation times and, possibly, worse local optima.

APPENDIX B – BÉZIER CURVE METHOD DERIVATION

To satisfy the boundary conditions (BCs) in (24), it is necessary to compute the derivatives of (31) and (32) with respect to the scaled time τ. The first derivative of (31) provides the speed state variables of each vehicle j, v_{x_j} and v_{y_j}, in each coordinate of the Cartesian coordinate system (CCS), given by:

$$\mathbf{z}'(\tau) = \sum_{l=0}^{n_\mathbf{z}} B'_{\mathbf{z},l}(\tau) P_{\mathbf{z},l}, \tag{102}$$

and the second derivative is:

$$\mathbf{z}''(\tau) = \sum_{l=0}^{n_\mathbf{z}} B''_{\mathbf{z},l}(\tau) P_{\mathbf{z},l}, \tag{103}$$

where

$$B'_{\mathbf{z},l}(\tau) = \begin{cases} -n_\mathbf{z}(1-\tau)^{n_\mathbf{z}-1}, & \text{if } l = 0, \\ \frac{n_\mathbf{z}!\,\tau^{l-1}(1-\tau)^{n_\mathbf{z}-l}}{(l-1)!(n_\mathbf{z}-l)!} - \frac{n_\mathbf{z}!\,\tau^{l}(1-\tau)^{n_\mathbf{z}-l-1}}{l!(n_\mathbf{z}-l-1)!}, & \text{if } l \in [1, n_\mathbf{z}-1], \\ n_\mathbf{z}\tau^{n_\mathbf{z}-1}, & \text{if } l = n_\mathbf{z}. \end{cases} \tag{104}$$

and

$$B''_{\mathbf{z},l}(\tau) = \begin{cases} n_\mathbf{z}(n_\mathbf{z}-1)(1-\tau)^{n_\mathbf{z}-2}, & \text{if } l = 0 \\ n_\mathbf{z}(n_\mathbf{z}-1)(n_\mathbf{z}-2)\tau(1-\tau)^{n_\mathbf{z}-3} - 2n_\mathbf{z}(n_\mathbf{z}-1)(1-\tau)^{n_\mathbf{z}-2}, & \text{if } l = 1 \\ \frac{n_\mathbf{z}!\,\tau^{l-2}\,(1-\tau)^{n_\mathbf{z}-l}}{(l-2)!(n_\mathbf{z}-l)!} - \frac{2n_\mathbf{z}!\,\tau^{l-1}\,(1-\tau)^{n_\mathbf{z}-l-1}}{(l-1)!(n_\mathbf{z}-l-1)!} + \frac{n_\mathbf{z}!\,\tau^{l}\,(1-\tau)^{n_\mathbf{z}-l-2}}{l!(n_\mathbf{z}-l-2)!}, & \text{if } l \in [2, n_\mathbf{z}-2] \\ n_\mathbf{z}(n_\mathbf{z}-1)(n_\mathbf{z}-2)\tau^{n_\mathbf{z}-3}(1-\tau) - 2n_\mathbf{z}(n_\mathbf{z}-1)\tau^{n_\mathbf{z}-2}, & \text{if } l = n_\mathbf{z}-1 \\ n_\mathbf{z}(n_\mathbf{z}-1)\tau^{n_\mathbf{z}-2}, & \text{if } l = n_\mathbf{z} \end{cases} \tag{105}$$

B.1 EXPRESSING SOME BÉZIER COEFFICIENTS USING BOUNDARY CONDITIONS

By replacing $\tau = 0$ and $\tau = 1$ in (32) and (104), the boundary values of $B_{n_\mathbf{z},l}(\tau)$ and $B'_{n_\mathbf{z},l}(\tau)$ can be obtained as follows:

$$
\begin{aligned}
B_{\mathbf{z},l}(\tau=0) &= \begin{cases} 1 & l=0 \\ 0 & l \in [1, n_\mathbf{z}] \end{cases} \\
B_{\mathbf{z},l}(\tau=1) &= \begin{cases} 0 & l \in [0, n_\mathbf{z}-1] \\ 1 & l=n_\mathbf{z} \end{cases} \\
B'_{\mathbf{z},l}(\tau=0) &= \begin{cases} -n_\mathbf{z} & l=0 \\ n_\mathbf{z} & l=1 \\ 0 & l \in [2, n_\mathbf{z}] \end{cases} \\
B'_{\mathbf{z},l}(\tau=1) &= \begin{cases} 0 & l \in [0, n_\mathbf{z}-2] \\ -n_\mathbf{z} & l=n_\mathbf{z}-1 \\ n_\mathbf{z} & l=n_\mathbf{z} \end{cases}
\end{aligned}
\tag{106}
$$

Given equations (24), (31), (102), and (106), one can write the following relations:

$$
\begin{aligned}
&\mathbf{z}_\mathrm{i} = P_{\mathbf{z},0}, && \mathbf{z}_\mathrm{f} = P_{\mathbf{z},n_\mathbf{z}} \\
&\dot{\mathbf{z}}_\mathrm{i} = \frac{n_\mathbf{z}}{T}(P_{\mathbf{z},1} - P_{\mathbf{z},0}), && \dot{\mathbf{z}}_\mathrm{f} = \frac{n_\mathbf{z}}{T}(P_{\mathbf{z},n_\mathbf{z}} - P_{\mathbf{z},n_\mathbf{z}-1})
\end{aligned}
\tag{107}
$$

By manipulating (107), one can obtain eight known Bezier coefficients ($P_{\mathbf{z},0}$, $P_{\mathbf{z},1}$, $P_{\mathbf{z},n_\mathbf{z}-1}$ and $P_{\mathbf{z},n_\mathbf{z}}$) as presented in (34).

B.2 BÉZIER COMPACT MATRIX FORM

The first and second derivatives of positions state variables (38) at the m discretization points (DPs) as vectors of their values can be obtained as follows:

$$
\begin{aligned}
[\mathbf{z}]_{m\times 1} &= [B_\mathbf{z}]_{m\times(n_\mathbf{z}-3)}[X_\mathbf{z}]_{(n_\mathbf{z}-3)\times 1} + [F_\mathbf{z}]_{m\times 1}, \\
[\mathbf{z}']_{m\times 1} &= [B'_\mathbf{z}]_{m\times(n_\mathbf{z}-3)}[X_\mathbf{z}]_{(n_\mathbf{z}-3)\times 1} + [F'_\mathbf{z}]_{m\times 1}, \\
[\mathbf{z}'']_{m\times 1} &= [B''_\mathbf{z}]_{m\times(n_\mathbf{z}-3)}[X_\mathbf{z}]_{(n_\mathbf{z}-3)\times 1} + [F''_\mathbf{z}]_{m\times 1},
\end{aligned}
\tag{108}
$$

with

$$[X_\mathbf{z}]_{(n_\mathbf{z}-3)\times 1} = [P_{\mathbf{z},2}\ P_{\mathbf{z},3}\ \dots\ P_{\mathbf{z},n_\mathbf{z}-2}]^T, \tag{109}$$

$$[B_\mathbf{z}]_{m\times(n_\mathbf{z}-3)} = [B_{\mathbf{z},2}\ B_{\mathbf{z},3}\ \dots\ B_{\mathbf{z},n_\mathbf{z}-2}]^T, \tag{110}$$

$$[B'_\mathbf{z}]_{m\times(n_\mathbf{z}-3)} = [B'_{\mathbf{z},2}\ B'_{\mathbf{z},3}\ \dots\ B'_{\mathbf{z},n_\mathbf{z}-2}]^T, \tag{111}$$

$$[B''_\mathbf{z}]_{m\times(n_\mathbf{z}-3)} = [B''_{\mathbf{z},2}\ B''_{\mathbf{z},3}\ \dots\ B''_{\mathbf{z},n_\mathbf{z}-2}]^T, \tag{112}$$

where $[F_\mathbf{z}]$, $[F'_\mathbf{z}]$, and $[F''_\mathbf{z}]$ are the constant vectors depending on $n_\mathbf{z}$ and on the BCs that are obtained using (36) and its derivatives.

APPENDIX C – DUAL FORMULATION DERIVATION

In order to derive (63) from (61), we can leverage the convexity of the primal problem (61) and the nonemptiness of sets $\mathcal{P}_i$ and $\mathcal{P}_j$. This leads to the establishment of strong duality, as discussed in (BOYD, S. P.; VANDENBERGHE, 2004). Consequently, we can solve the dual problem instead of the primal problem. The equivalent form of primal problem (61) is:

$$\begin{aligned} \min_{\mathbf{p}_i,\mathbf{p}_j} \ & \|\mathbf{p}_i - \mathbf{p}_j\|_2 = \min_{\mathbf{p}_i,\mathbf{p}_j,\mathbf{w}_{i,j}} \|\mathbf{w}_{i,j}\|_2 \\ \text{s.t.} \quad & \mathbf{A}_i\,\mathbf{p}_i \leq \mathbf{b}_i \\ & \mathbf{A}_j\,\mathbf{p}_j \leq \mathbf{b}_j \\ & \mathbf{p}_i - \mathbf{p}_j = \mathbf{w}_{i,j} \end{aligned} \tag{113}$$

The Lagrangian function associated with problem (113) is defined as follows:

$$\begin{aligned} L(\mathbf{p}_i, \mathbf{p}_j, \boldsymbol{\lambda}_{i,j}, \boldsymbol{\lambda}_{j,i}, \mathbf{s}_{i,j}) = \|\mathbf{w}_{i,j}\|_2 + \boldsymbol{\lambda}_{i,j}^T(\mathbf{A}_i\mathbf{p}_i - \mathbf{b}_i) + \boldsymbol{\lambda}_{j,i}^T(\mathbf{A}_j\mathbf{p}_j - \mathbf{b}_j) \\ + \mathbf{s}_{i,j}^T(\mathbf{p}_i - \mathbf{p}_j - \mathbf{w}_{i,j}), \end{aligned} \tag{114}$$

where $\boldsymbol{\lambda}_{i,j}, \boldsymbol{\lambda}_{j,i}$ and $\mathbf{s}_{i,j} = \mathbf{s}_{j,i}$ are the Lagrange multiplier vectors associated with the inequality constraints and equality constraints, respectively. Then, the Lagrangian dual function is defined as:

$$g(\boldsymbol{\lambda}_{i,j}, \boldsymbol{\lambda}_{j,i}, \mathbf{s}_{i,j}) = \inf_{\mathbf{p}_i,\mathbf{p}_j,\mathbf{w}_{i,j}} L(\mathbf{p}_i, \mathbf{p}_j, \boldsymbol{\lambda}_{i,j}, \boldsymbol{\lambda}_{j,i}, \mathbf{s}_{i,j}), \tag{115}$$

By rearranging the terms in (115), one can write the following relation:

$$\begin{aligned} g(\boldsymbol{\lambda}_{i,j}, \boldsymbol{\lambda}_{j,i}, \mathbf{s}_{i,j}) = & \inf_{\mathbf{w}_{i,j}} \left(\|\mathbf{w}_{i,j}\|_2 - \mathbf{s}_{i,j}^T\mathbf{w}_{i,j} \right) \\ & + \inf_{\mathbf{p}_i} \left((\mathbf{A}_i^T\boldsymbol{\lambda}_{i,j} + \mathbf{s}_{i,j})^T\mathbf{p}_i - \boldsymbol{\lambda}_{i,j}^T\mathbf{b}_i \right) \\ & + \inf_{\mathbf{p}_j} \left((\mathbf{A}_j^T\boldsymbol{\lambda}_{j,i} - \mathbf{s}_{i,j})^T\mathbf{p}_j - \boldsymbol{\lambda}_{j,i}^T\mathbf{b}_j \right). \end{aligned} \tag{116}$$

One can simplify and rephrase (116) by employing the definition of the conjugate function $f^*(s) = \sup_{x\in\mathrm{dom}f}(s^Tx - f(x))$ and utilizing the fact that $\inf f(x) = -\sup(-f(x))$. Consequently, we obtain:

$$\inf_x(f(x) - s^Tx) = -\sup_x(-f(x) + s^Tx) = -f^*(s).$$

For the first term of the right-hand side of (116), we have

$$\inf_{\mathbf{w}_{i,j}} \left(\|\mathbf{w}_{i,j}\|_2 - \mathbf{s}_{i,j}^T\mathbf{w}_{i,j} \right) = -\sup_{\mathbf{w}_{i,j}} \left(\mathbf{s}_{i,j}^T\mathbf{w}_{i,j} - \|\mathbf{w}_{i,j}\|_2 \right) = -f^*(\mathbf{s}_{i,j}) = -\|\mathbf{w}_{i,j}\|_2^*, \tag{117}$$

and the conjugate of $\|\mathbf{w}_{i,j}\|$ is (see example 3.26 on page 93 of the book by Stephen P Boyd and Vandenberghe (2004)):

$$f^*(\mathbf{s}_{i,j}) = \begin{cases} 0 & \|\mathbf{s}_{i,j}\|_* \leq 1 \\ \infty & \text{otherwise,} \end{cases} \tag{118}$$

with $\|\cdot\|$ is a norm on $\mathbb{R}^n$ and $\|\cdot\|_*$ is the dual norm. For the second term of the right-hand side of (116), we have:

$$\inf_{\mathbf{p}_i} \left((\mathbf{A}_i^T \boldsymbol{\lambda}_{i,j} + \mathbf{s}_{i,j})^T \mathbf{p}_i - \boldsymbol{\lambda}_{i,j}^T \mathbf{b}_i \right) = \begin{cases} -\mathbf{b}_i^T \boldsymbol{\lambda}_{i,j} & \mathbf{A}_i^T \boldsymbol{\lambda}_{i,j} + \mathbf{s}_{i,j} = 0 \\ -\infty & \text{otherwise,} \end{cases} \tag{119}$$

and similarly, for the third term of the right-hand side of (116), we have:

$$\inf_{\mathbf{p}_j} \left((\mathbf{A}_j^T \boldsymbol{\lambda}_{j,i} + \mathbf{s}_{i,j})^T \mathbf{p}_j - \boldsymbol{\lambda}_{j,i}^T \mathbf{b}_j \right) = \begin{cases} -\mathbf{b}_j^T \boldsymbol{\lambda}_{j,i} & \mathbf{A}_j^T \boldsymbol{\lambda}_{j,i} - \mathbf{s}_{i,j} = 0 \\ -\infty & \text{otherwise.} \end{cases} \tag{120}$$

Finally, by replacing (117)–(120) in (116) and forming the dual problem as:

$$\begin{aligned} &\max \ g(\boldsymbol{\lambda}_{i,j}, \boldsymbol{\lambda}_{j,i}, \mathbf{s}_{i,j}) \\ &\text{s.t.} \quad \boldsymbol{\lambda}_{i,j}, \boldsymbol{\lambda}_{j,i} \geq 0 \end{aligned} \tag{121}$$

the dual form of (61) is formed as follows:

$$\begin{aligned} \operatorname{dist}(\mathcal{P}_i, \mathcal{P}_j) = \max_{\boldsymbol{\lambda}_{i,j}, \boldsymbol{\lambda}_{j,i}, \mathbf{s}_{i,j}} \quad & -\mathbf{b}_i^T \boldsymbol{\lambda}_{i,j} - \mathbf{b}_j^T \boldsymbol{\lambda}_{j,i} \\ \text{s.t.} \quad & \mathbf{A}_i^T \boldsymbol{\lambda}_{i,j} + \mathbf{s}_{i,j} = 0, \ \mathbf{A}_j^T \boldsymbol{\lambda}_{j,i} - \mathbf{s}_{i,j} = 0, \\ & \|\mathbf{s}_{i,j}\|_2 \leq 1, \ -\boldsymbol{\lambda}_{i,j} \leq 0, -\boldsymbol{\lambda}_{j,i} \leq 0. \end{aligned} \tag{122}$$

The optimal value of the dual problem, which is the distance between $\mathcal{P}_i$ and $\mathcal{P}_j$, is constrained to be larger than d_{s}. We can utilize this intuition to reformulate problem (63) as the following feasibility problem for $-\boldsymbol{\lambda}_{i,j} \leq 0$, $-\boldsymbol{\lambda}_{j,i} \leq 0$, and s:

$$\begin{aligned} & -\mathbf{b}_i^T \boldsymbol{\lambda}_{i,j} - \mathbf{b}_j^T \boldsymbol{\lambda}_{j,i} \geq d_{\mathrm{s}} \\ & \mathbf{A}_i^T \boldsymbol{\lambda}_{i,j} + \mathbf{s}_{i,j} = 0, \\ & \mathbf{A}_j^T \boldsymbol{\lambda}_{j,i} - \mathbf{s}_{i,j} = 0, \\ & \|\mathbf{s}_{i,j}\|_2 \leq 1. \end{aligned} \tag{123}$$